

# Visual Servo Control by Optimizing Hybrid Objective Function with Visibility and Path Constraints

A.H. Abdul Hafez \*

\* Dept. of Computer Engineering, Faculty of Engineering,  
Hasan Kalyoncu University, Sahinbey, 27410 - Gaziantep, Turkey  
(e-mail: [abdul.hafez@hku.edu.tr](mailto:abdul.hafez@hku.edu.tr)).

**Abstract:** This paper presents a new stable hybrid visual servo controller to complete the task of robot arm positioning. Our method is generic hybrid method and superior to the state of the art. The objective function is designed to include the full but weighted 2D and 3D available information. The positioning task has been formulated as a minimization problem. Here, each of the 2D and 3D error functions is used to control the six degrees of freedom. The importance weights are computed to satisfy a set of constraints defined by the visual servoing process. Stability analysis is also presented for the proposed control law. The experimental evaluation is done in order to show the enhanced performance of the visual servoing process. Simulation results show that this method provides an efficient solution to the camera retreat and features visibility problems. Performance of the visual servoing system is evaluated by its ability to keep features visible in the image and the Cartesian trajectory within the robot workspace during the process.

**Keywords:** Visual Servo Control; Optimization; Hybrid Control; Visibility constraints.

## 1. INTRODUCTION

Visual servoing has become an attractive area of research, and has recently received considerable amount of attention Corck and Hutchinson (2001), Malis et al. (1999), Chaumette and Hutchinson (2006), and Chaumette and Hutchinson (2007). Visual servoing schemes use one or more cameras along with computer vision algorithms to control the position of a robot arm or mobile robot with respect to an object or a set of features of the object to be manipulated Unger et al. (2009), Kragic (2001), and Cai et al. (2013). It is used in a wide range of applications such as robot navigation Cherubini and Chaumette (2012), and Li et al. (2013), lane tracking by vehicles Cherubini et al. (2011), and Benhimane and Malis (2007), and industrial manipulation Benhimane et al. (2008).

The essence of visual servoing is to move the concerned object from the current pose to a desired pose given by current and desired images. This is essentially obtained by nullifying a task function Samson et al. (1991) or minimization of a cost function Malis (2004). Visual features are extracted from the two images and used to formulate a function of the error between the current pose and the desired one Dani et al. (2013), Kragic (2001), and Samson et al. (1991). The role of the minimization process is to regulate this error function to zero. Image features can be used directly in the definition of the error function. This leads to a formulation of the 2D error function that is minimized in the image space. These image features may also be used to formulate an error function in the pose space.

In image-based visual servoing, the input is computed in the 2-D image space. This method is robust to robot and camera calibration errors Dani et al. (2013) and Chaumette (1998). However, image-based visual servoing is known to be locally stable with a stability region that is difficult to determine. This

is in addition to the need of the depth estimates of the image features Luca et al. (2007). The main drawback of 3D visual servoing is the lack of control in the image space. This implies that the object may get out of the camera field of view Kermorgant (2013) and Chaumette (1998). In contrast, 2D visual servoing does not have any control in the Cartesian space and the camera trajectory is not predictable. A satisfactory control scheme that avoids these drawbacks Cai et al. (2013), Malis et al. (1999), and Malis and Chaumette (2002) is to use both kinds of information *i.e.*, 2D and 3D. For this reason, this control scheme is called 2 1/2D visual servoing Malis et al. (1999). A hybrid (2D-3D) cost function that contains information from the 2D image space and the 3D pose space was proposed in Abdul Hafez and Jawahar (2006a), and Abdul Hafez and Jawahar (2007). In Abdul Hafez and Jawahar (2007), minimization of this objective function is applied to the visual servoing process. In such a minimization method, the minimization process searches for a least squares solution that minimizes the 2D error from the image space and the 3D error from the pose space simultaneously. In this paper, the previous works Abdul Hafez and Jawahar (2006a) and Abdul Hafez and Jawahar (2007) are extended to the case of varying weighting factors. This allows these factors to be included in the pseudo-inverse of the Jacobean matrix that improve the performance. Stability analysis and weight computation strategy is also presented. This work is continuation to the hybrid visual servo control works presented in the literature. It is started by Malis et al. (1999); Malis and Chaumette (2002) and then Abdul Hafez and Jawahar (2006a), Abdul Hafez and Jawahar (2007), and Abdul Hafez et al. (2013). Later, Kermorgant and Chaumette proposed in Kermorgant and Chaumette (2011) a similar work to this with small difference in the computation method of weights. The difference from the current work is they assign an individual unbounded weight to each feature. However, this

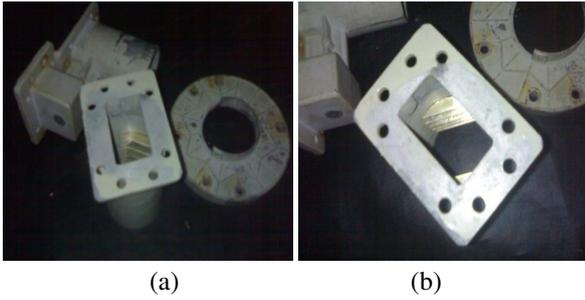


Fig. 1. The initial and desired camera poses of a positioning task given as images. Initial image in (a) and desired image in (b).

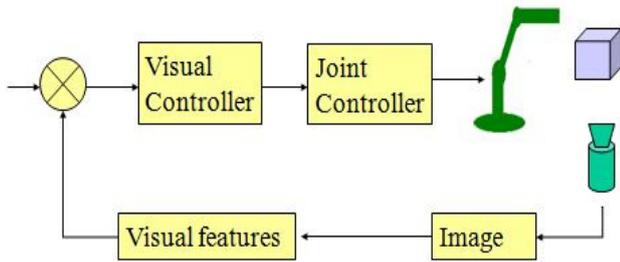


Fig. 2. Dynamic look-and-move visual servoing system. The output of the visual controller is the Cartesian velocity, which is fed as an input to internal (joint controller).

may bring stability issues due to accumulated high gain resulted from weights of more than one active features. The proposed work, presented a stable hybrid control law with time varying weights. It satisfies a few constraints like feature visibility and straight line constraints.

The experimental evaluation is done in order to show the enhanced performance of the visual servoing process. Since the visual servoing process is presented within the optimization framework, it is preferable to analyze the analogous properties of visual servoing as well as the optimization process. Two main properties are considered here, the time of convergence and the trajectory of the working point during the optimization process. Our hybrid function is defined over both image and Cartesian spaces. It is enough to evaluate the time of convergence over the Cartesian space due to the duality between the two mentioned spaces. The trajectory of the working point is evaluated by projecting it on the Cartesian space to get the camera trajectory, and on the image space to get the image feature trajectory. The performance of visual servoing system is evaluated by its ability to keep features visible in the image and the Cartesian trajectory within the robot workspace during the process.

## 2. VISUAL SERVOING AS MINIMIZATION PROBLEM

The positioning task is to move the robot end-effector from an initial pose  $P \in R^3 \times SO(3)$  to reach a desired pose  $P^*$ . These two poses are almost given, in the general case, by initial and desired images, (see Figure 1 for example). In other words, the problem is to minimize an error vector  $e(s)$  of visual features  $s(P)$  extracted from the initial/current image by finding a change in the pose vector  $\Delta P$  or the velocity vector

$V = [\nu, \omega]^T$  that incrementally minimizes a cost function  $E(s(P))$ . Viewing the problem as a nonlinear least squares minimization allows us to formulate the following cost function

$$E(s(P)) = \frac{1}{2}(s(P) - s(P^*))^T(s(P) - s(P^*)). \quad (1)$$

Formulating the positioning task as a minimization problem is equivalent to the formulation as a control problem. The control problem is that of finding a feedback control signal  $V = \frac{\Delta P}{\Delta t} = [\nu, \omega]^T$  such that the output  $s(P)$  reaches a desired output  $s(P^*)$ . Most of the visual servoing schemes follow the indirect visual servoing control method by producing the Cartesian velocity output. A block diagram of the general look-and-move (indirect) visual servoing has been shown in Figure 2.

Most of the familiar minimization methods that provides efficient solutions to wide range of optimization problems are based on the Taylor series approximation of the cost function. The cost function given in Equation (1) can be written as

$$E(s(P^*)) \approx E(s(P)) + \frac{\partial E(s(P))}{\partial P} \Delta P + \frac{1}{2} \Delta P^T \frac{\partial^2 E(s(P))}{\partial P^2} \Delta P. \quad (2)$$

Considering that the gradient of the cost function  $\frac{\partial E(s(P))}{\partial P}$  is given as

$$\frac{\partial E(s(P))}{\partial P} = (s(P) - s(P^*))^T \frac{\partial s(P)}{\partial P} = e(s)^T J(P). \quad (3)$$

The matrix  $J(P) = \frac{\partial s(P)}{\partial P}$  is the feature space Jacobean or the first order derivative of the feature vector  $s(P)$  with respect to the pose vector  $P$ . In Gauss-Newton minimization, the second order derivative is approximated by

$$\frac{\partial^2 E(s(P))}{\partial P^2} = J^T(P)J(P), \quad (4)$$

and substituting (3) and (4) in (2), the required change in the pose is thereby

$$V = \frac{\Delta P}{\Delta t} = -\lambda J^+(P)e(s), \quad (5)$$

where  $\lambda$  is a positive constant parameter that defines the step size of the minimization process. The matrix  $J^+(P) = (J^T(P)J(P))^{-1} J^T(P)$  is the pseudo-inverse of the matrix  $J$ . This method is known as the Jacobian Pseudo-inverse method and widely used in the robot control and visual servoing Espiau et al. (1992); Malis (2004). It is enough to solve for the first order approximation to perform the minimization process. Particularly, it requires the only computation of the Jacobean matrix which is done through the computations of gradient vector. Indeed, Gauss-Newton minimization method is fast and more efficient near to minimum point.

Based on the type of the visual features  $s(P^*)$  used in the minimization process, the cost function defined in Equation (1) varies from  $E_{2D}(P)$  for 2D visual features from image space and image-based visual servoing (IBVS) to  $E_{3D}(P)$  for 3D visual features from the Cartesian space and position-based visual servoing (PBVS).

### 2.1 Image-based Visual Servoing (IBVS)

In traditional IBVS, the cost function is expressed directly in the 2D image space. Thus, the 2D visual servoing is also called image-based visual-servoing. Usually, the 2D coordinates of a set of image points are considered as features, the vector  $s(P)$  becomes  $s_{2D}(P) = [x_1, y_1, \dots, x_N, y_N]^T$  while the

desired vector  $s(P^*)$  is  $s_{2D}(P^*) = [x_1^*, y_1^*, \dots, x_N^*, y_N^*]^T$ . Equation (1) may be written as

$$E_{2D}(s(P)) = \frac{1}{2} e_i(P)^T e_i(P), \quad (6)$$

where

$$e_i(P) = s_{2D}(P) - s_{2D}(P^*). \quad (7)$$

Using the Newton minimization method presented in Section 2, the velocity vector in Equation (5) can be rewritten as

$$V_i = -\lambda J_i^+(P) e_i(s). \quad (8)$$

This is the IBVS control law with respect to the cost function defined in Equation (6). The matrix  $J_i(P) = \frac{\partial(e_i(P))}{\partial P}$  is then the image Jacobian matrix or some times called the interaction matrix Wilson et al. (1996). It describes the relation between the changes or motion in the image space and the corresponding changes or motion in the Cartesian space.

For a set of features contains  $N$  image points, the Jacobian matrix is given as

$$J_i(P) = [J_{i1}^T \dots J_{ik}^T \dots J_{iN}^T]^T,$$

where  $J_{ik}$  is computed as a function of the  $k$ th image point coordinates  $x_k, y_k$  and its depth  $Z_k$  as follows

$$J_{ik} = \begin{bmatrix} -\frac{1}{Z_k} & 0 & \frac{x_k}{Z_k} & x_k y_k & -(1+x_k^2) & y_k \\ 0 & -\frac{1}{Z_k} & \frac{y_k}{Z_k} & 1+y_k^2 & -x_k y_k & -x_k \end{bmatrix}. \quad (9)$$

To avoid the Jacobian singularity, the number of features is selected such that  $N > 3$ . This corresponds to at least 4 image points. In the simplest case, we consider points as visual features. The interaction matrix for a large range of image features (straight lines, ellipses, etc.) can be found in Espiau et al. (1992).

Note that  $J_i(P)$  depends on the depth  $Z$  of each selected point feature. Thus, even if the 2D visual servoing is "model-free" we still need some knowledge about the depths of the points. Several solutions have been proposed in order to solve the problem of the estimation of the depth Abdul Hafez and Jawahar (2006b); Luca et al. (2007); Yap et al. (2011); Janabi-Sharifi and Marey (2010). An estimate of the depth can be obtained using, as in 3D visual servoing, a pose determination algorithm (if a 3D target model is available), or using a structure from known motion algorithm (if the camera motion can be measured). However, using this choice may lead the system close to, or even reach, a singularity of the interaction matrix. Furthermore, the convergence may also not be attained due to local minima reached because of the computation by the control law of unrealizable motions in the image Chaumette (1998). The pseudo-inverse  $J_i^+$  of the interaction matrix  $J_i$  is usually approximated by  $\widehat{J}_i^+$  can be estimated using different computational methods. It can be numerically estimated or computed analytically.

In general, image-based visual servoing is known to be robust not only with respect to camera but also to robot calibration errors. However, its convergence is theoretically ensured only in a region (quite difficult to determine analytically) around the desired position. Except in very simple cases, the analysis of the stability with respect to calibration errors seems to be impossible, since the system is coupled and non-linear.

## 2.2 Position-based Visual Servoing (PBVS)

The control law in position-based visual-servoing uses directly the error on the pose of the camera. Based on the number of visual features available in the image the pose can be computed regardless the availability of the model of target object. 3D visual features such as the position and orientation can be part of the feature vector  $s_{3D}(P) = [T, u\theta]^T = [T_x, T_y, T_z, u\theta]^T$ . Here,  $T$  is the translation vector and  $u\theta$  is the rotation vector represented by the rotation axis  $u$  and angle  $\theta$ . The desired features are  $s_{3D}(P^*)$ . Similarly to Equation (6) and Equation (7), we can write

$$E_{3D}(s(P)) = \frac{1}{2} e_p(P)^T e_p(P), \quad (10)$$

where

$$e_p(P) = s_{3D}(P) - s_{3D}(P^*). \quad (11)$$

Using the Newton minimization method presented in Section 2, the velocity vector given in Equation (5) can be written for the PBVS control law as

$$V_i = -\lambda J_p^+(P) e_p(s), \quad (12)$$

where the matrix

$$J_p(P) = \frac{\partial(e_p(P))}{\partial P}$$

is the Cartesian Jacobean matrix. It describes the relation between the motion or changes of the selected 3D features, and changes of the camera pose or the motion of the camera.

The translation vector  $T$  represents the translation between the current camera frame  $F_C$  and the desired one  $F_{C^*}$ . Then,  $s_{3D}(P) = [{}^C T_C, u\theta]^T$ ,  $s_{3D}(P^*) = 0_{(6 \times 1)}$ , and the Jacobian matrix is given as

$$J_p(P) = \begin{bmatrix} R_C^{C^*} & 0_{(3 \times 3)} \\ 0_{(3 \times 3)} & J_w \end{bmatrix}, \quad (13)$$

where

$$J_w = \frac{\partial(u\theta)}{\partial P} = \mathbf{I}_3 - \frac{\theta}{2} [\mathbf{u}]_{\times} + \left(1 - \frac{\text{sinc } \theta}{\text{sinc}^2 \frac{\theta}{2}}\right) [\mathbf{u}]_{\times}^2. \quad (14)$$

Here,  $\text{sinc}(x) = \frac{\sin(x)}{x}$ , and  $J_w^{-1} u\theta = u\theta$ .

Let us note that the matrices  $T$  and  $u\theta$  are computed using pose estimation algorithm Yap et al. (2011); Janabi-Sharifi and Marey (2010). The intrinsic camera parameters should be available. In addition, the CAD model of the target object should also be available. The main advantage of this approach is that it directly controls the camera trajectory in Cartesian space. Once we have the desired camera and the current camera pose, the camera displacement to reach the desired position is thus easily obtained, and the control of the robot end-effector can be performed either in open loop or, more robustly, in closed-loop. However, since there is no control in the image, the image features used in the pose estimation may leave the image, especially if the robot or the camera is coarsely calibrated. This leads to servoing failure. Also note that, if the camera is coarse calibrated, or if errors exist in the 3D model of the target, the current and desired camera poses will not be accurately estimated.

## 2.3 Hybrid 2 1/2 D Visual Servoing

Usually, hybrid methods use 2D and 3D information extracted from partial geometric reconstruction. Partial geometric reconstruction may be done based on homography matrix decomposition Malis et al. (1999) or from decomposing the essential

matrix Malis et al. (2000). Many hybrid visual servoing methods can be identified as partitioned methods Deguchi (1998); Corck and Hutchinson (2001); Kyrki et al. (2004), since they are partitioning the degrees of freedom of the robot motion. In the remaining of this Subsection, the seminal work on hybrid visual servoing Malis et al. (1999) is reviewed in addition to three examples of the partitioned methods.

*Design of the Error Function and Computing the Jacobian*  
The 2 1/2 D visual servoing is a hybrid method that partially uses the information from the 2D image space and 3D Cartesian space. The selection of partial 2D and 3D information to be involved in the 2 1/2 D visual servoing control law is done by defining the feature vectors  $s_{2D}(P)$  and  $s_{3D}(P)$  as follows Malis et al. (2000):

$$s_{2D}(P) = [x, y, \log(Z)]^T, \quad (15)$$

$$s_{3D}(P) = [C^*(u\theta)_C]^T. \quad (16)$$

Here,  $x$  and  $y$  are the normalized image coordinates of a selected image point  $m$  and  $Z$  is its 3D depth given with respect to the current camera frame. The angle  $\theta$  and axis  $u$  represent the rotation between the current and desired camera frame.

Let the desired values of these two feature vectors be

$$s_{2D}(P^*) = [x^*, y^*, \log(Z^*)]^T, \text{ and } s_{3D}(P^*) = [\mathbf{0}_{3 \times 1}]. \quad (17)$$

The error functions  $e_i(P) = s_{2D}(P) - s_{2D}(P^*)$  and  $e_p(P) = s_{3D}(P) - s_{3D}(P^*)$  are defined as follows

$$e_i(P) = [x - x^*, y - y^*, \log(Z/Z^*)]^T, \quad (18)$$

$$e_p(P) = [C^*(u\theta)_C]^T. \quad (19)$$

To compute the Jacobian matrix, let us write the time derivative of these error functions as a function of the camera screw velocity  $V$ . The time derivative of  $e_p(P) = u\theta$  can be written as

$$\begin{aligned} \frac{\partial(e_p(P))}{\partial t} &= \frac{\partial u\theta}{\partial t} = \frac{\partial(e_p(P))}{\partial P} \cdot \frac{\partial P}{\partial t} \\ &= J_p(P) \cdot V_c = [\mathbf{0}_{3 \times 3} \quad J_w] V_c. \end{aligned} \quad (20)$$

The matrix  $J_w$  is given same as Equation (14).

The time derivative of  $e_i(P)$  can be written as

$$\begin{aligned} \frac{\partial(e_i(P))}{\partial t} &= \frac{\partial(e_i(P))}{\partial P} \cdot \frac{\partial P}{\partial t} \\ &= \frac{\partial([x - x^*, y - y^*, \log(Z/Z^*)]^T)}{\partial P} \cdot \frac{\partial P}{\partial t} \\ &= J_i(P) \cdot V_c = \left[ \frac{1}{d^*} J_v \quad J_{v,w} \right] V_c. \end{aligned} \quad (21)$$

Here,  $d^*$  is unknown parameter related to the depth  $Z$  of the concern point as  $\rho = Z/d^*$ . Substituting in Equation (5) after considering that

$$J(P) = \begin{bmatrix} \frac{1}{d^*} J_v & J_{v,w} \\ \mathbf{0}_{3 \times 3} & J_w \end{bmatrix}, \quad (22)$$

One can note that the matrix  $J^{-1}(P)$  is an upper triangular square matrix. This matrix does not suffer from any singularity in the task space.

### 3. VISUAL SERVO CONTROL BY MINIMIZING HYBRID WEIGHTED OBJECTIVE FUNCTION

We proposed in Abdul Hafez and Jawahar (2006a, 2007) a hybrid (2D-3D) cost function that contains dense information

from the 2D image space and the full 3D pose information. The minimization of this objective function is applied to the visual servoing process. A close work to ours is the error function which was proposed in Malis et al. (1999); Malis and Chaumette (2002) and the one recently proposed in Kermorgant and Chaumette (2011). In Malis et al. (1999), the error function was defined as a 6-vector. The 3-vector that contains the 2D visual information from the image space is used to recover the position of the camera, where the 3-vector that contains the 3D visual information from the pose space is used to recover the orientation of the camera. This has been reviewed in Section 2.3. In contrast, current method uses both the image space information and pose space information to recover the full camera pose information, both position and orientation. Both 2D and 3D errors are concatenated in a  $(2N+6)$ -vector and minimized together. It is assumed that an  $n$  image points and 6 dimensional 3D vector (3 for translation and 3 for rotation). In such a minimization method, the minimization process searches for a least squares solution that minimizes both the 2D error from the image space and the 3D error from the pose space simultaneously.

The work presented in Kermorgant and Chaumette (2011), is identical to the one we proposed in Abdul Hafez and Jawahar (2006a, 2007), with an exception that the weights are computed and assigned in different way. These differences are focused and detailed later in this section. This paper extends the previous works Abdul Hafez and Jawahar (2006a, 2007); Kermorgant and Chaumette (2011) in such a way that: (i) the weights are assumed time variants. Indeed, the weights matrix will be augmented in the pseudo-inverse of the Jacobian in Equation (5), (ii) stability analysis and proof is presented for the hybrid task. (iii) the weights computation methods are presented as well.

#### 3.1 Control Law from Minimizing the Hybrid Function

Let us define a hybrid cost function as the weighted sum of the two  $E_{3D}(s(P))$  and  $E_{2D}(s(P))$  function as follows

$$E_h(s(P)) = \lambda_1^2 E_{2D}(s(P)) + \lambda_2^2 E_{3D}(s(P)). \quad (23)$$

Here,  $\lambda_1$  and  $\lambda_2$  are positive scalar factors that play the role of the step size of the minimization process in addition to the integration ratio between 2D and 3D spaces. The functions  $E_{3D}(s(P))$  and  $E_{2D}(s(P))$  are the PBVS and IBVS cost functions as defined in Equations (6) and (10) respectively. Recall that the features  $s_{2D}(P)$  are selected as the 2D coordinates of a set of image points. The current features are  $s_{2D}(P) = [x_1, y_1, \dots, x_N, y_N]^T$  while the desired vector  $s(P^*)$  is  $s_{2D}(P^*) = [x_1^*, y_1^*, \dots, x_N^*, y_N^*]^T$ . In contrast, the feature  $s_{3D}(P)$  is selected as the pose vector  $P$ . The current features are  $s_{3D}(P) = [T, u\theta]^T = [T_x, T_y, T_z, u\theta]^T$ . while the desired features are  $s_{3D}(P^*)$ . While minimizing this cost function, the process searches for a solution that reduce the value of the two individual functions  $E_{3D}(s(P))$  and  $E_{2D}(s(P))$ .

Consider a positioning task to be achieved by minimizing the hybrid objective function given in Equation (23). This is the weighted sum of the objective functions defined in Equations (6) and (10) as IBVS and PBVS objective functions respectively. The gradient vector of this objective function is given as

$$\begin{aligned} \frac{\partial E_h(P)}{\partial P} &= \frac{2\lambda_1 \partial \lambda_1}{\partial P} + \lambda_1^2 \frac{\partial (e_i(P)^T e_i(P))}{\partial P} \\ &+ \frac{2\lambda_2 \partial \lambda_2}{\partial P} + \lambda_2^2 \frac{\partial (e_p(P)^T e_p(P))}{\partial P} \end{aligned} \quad (24)$$

The partial derivative of the weights can be written as  $\frac{\partial \lambda_i}{\partial P} = \frac{\partial \lambda_i}{\partial t} \frac{\partial t}{\partial P}$ . The term  $\frac{\partial \lambda_i}{\partial t}$  can be neglected if we assume that the weights are varying slowly. This is a classical assumption in visual servoing literature Garcia-Aracil et al. (2005); Mansard et al. (2009); Cheah et al. (2007).

By substituting this assumption in Equation 24, the later can be simplified to

$$\begin{aligned} \frac{\partial E_h(P)}{\partial P} &= \lambda_1^2 \frac{\partial (e_i(P)^T e_i(P))}{\partial P} + \lambda_2^2 \frac{\partial (e_p(P)^T e_p(P))}{\partial P} \\ &= \lambda_1^2 e_i(P)^T \frac{\partial (e_i(P))}{\partial P} + \lambda_2^2 e_p(P)^T \frac{\partial (e_p(P))}{\partial P} \\ &= \lambda_1^2 e_i(P)^T J_i(P) + \lambda_2^2 e_p(P)^T J_p(P). \end{aligned} \quad (25)$$

The above equation can be written in a compact matrix form as

$$\frac{\partial E_h(P)}{\partial P} = [e_i(P)^T \ e_p(P)^T] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} J_i(P) & 0 \\ 0 & J_p(P) \end{bmatrix}. \quad (26)$$

Here, the matrices  $\Lambda_i = \text{diag}(\lambda_i^2)$ . Equation (26) can be written by decomposing the matrices  $\Lambda_i$  as

$$\frac{\partial E_h(P)}{\partial P} = [e_i(P)^T \ e_p(P)^T] \begin{bmatrix} \tilde{\Lambda}_1 & 0 \\ 0 & \tilde{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \tilde{\Lambda}_1 & 0 \\ 0 & \tilde{\Lambda}_2 \end{bmatrix} \begin{bmatrix} J_i(P) & 0 \\ 0 & J_p(P) \end{bmatrix}, \quad (27)$$

where the matrices  $\tilde{\Lambda}_i = \text{diag}(\lambda_i)$ .

The above formula can be shown in the form of an augmented error function as

$$\frac{\partial E_h(P)}{\partial P} = e_h(P)^T J_h(P). \quad (28)$$

Here, the error

$$e_h(P) = \tilde{\Lambda} [e_i(P)^T \ e_p(P)^T]^T$$

. The matrix  $J_h$  is  $J_h = \tilde{\Lambda} J(P)$ , where  $\tilde{\Lambda} = \begin{bmatrix} \tilde{\Lambda}_1 & 0 \\ 0 & \tilde{\Lambda}_2 \end{bmatrix}$

and  $J(P) = \begin{bmatrix} J_i(P) & 0 \\ 0 & J_p(P) \end{bmatrix}$ . Similarly to the analysis in Section 2, the velocity signal that is a function of the change in the pose is given as:

$$V = -\mu J_h^+ e_h(P), \quad (29)$$

where  $J_h^+ = (J_h^T J_h)^{-1} J_h^T$  is the pseudo inverse of the matrix  $J_h$ , and  $\mu$  is proportional constant or the step size factor.

### 3.2 Stability Analysis of the Hybrid visual control law

This subsection considers the fundamental issues related to the stability of visual servoing control law. Then, the stability of the visual servoing using the proposed hybrid function is presented. Visual servoing system can be generally considered as a non-linear control system.

The stability of such systems, the closed-loop visual servo system, can be studied as the stability of non-linear system using Lyapunov theory. Let us consider a candidate Lyapunov function defined as

$$\mathcal{L} = \frac{1}{2} (s(P) - s(P^*))^T (s(P) - s(P^*)) = \frac{1}{2} e(P)^T e(P). \quad (30)$$

The first order derivative of this function is

$$\frac{\partial \mathcal{L}}{\partial t} = -\lambda e(P)^T J(P) \widehat{J(P)^+} e(P) \quad (31)$$

From the automatic control theory Isidori (1995); Khalil (2002), we can consider that the global asymptotic stability of a system to which the above Lyapunov function  $\mathcal{L}$  is defined, is obtained when the following sufficient condition is ensured:

$$J(P) \widehat{J(P)^+} > 0. \quad (32)$$

In general, if the number of the features  $N$  is equal to the number of degrees of freedom of the robot, the matrices  $J(P)$  and  $\widehat{J(P)^+}$  are of full rank 6, and the condition Equation (32) is ensured. This is subject to good approximation of the matrix  $\widehat{J(P)^+}$ . For more details about stability analysis of the two visual servoing controllers, IBVS and PBVS, readers are referred to Chaumette and Hutchinson (2006).

Let us consider the different possibilities for the values of the weights, *i.e.* the different possible values of the factors  $\lambda_1$  and  $\lambda_2$ . These include the following cases:

- (1) The first case is  $\lambda_1 = 1$  and  $\lambda_2 \geq 0$ . This case is reduced to pure IBVS when  $\lambda_2 = 0$ . Let us remember that IBVS has the property of producing a straight line trajectories in the image space.
- (2) The second case is  $\lambda_2 = 1$  and  $\lambda_1 \geq 0$ . This case is reduced to pure PBVS when  $\lambda_1 = 0$ . Let us remember that PBVS has the property of producing a straight line trajectories in the Cartesian space.
- (3) The third case is  $\lambda_1 > 0$  and  $\lambda_2 > 0$ . This is the general hybrid visual servoing algorithm. The algorithm here shows the properties of both IBVS and PBVS based on the values of the factors  $\lambda_1$  and  $\lambda_2$ .

However, some of these cases are published in the literature in different works by different authors. We originally proposed the third case in Abdul Hafez and Jawahar (2007). Later on to Abdul Hafez and Jawahar (2007), a work that is similar to the second case is proposed by Kermorgant and Chaumette proposed in Kermorgant and Chaumette (2011). The differences of the later work to our work is explained and discussed in Section 4.2.

Recollecting Equation (29) and considering the above first and second case, this equation can be written in the first case, particularly when  $\lambda_2 = 0$  as:

$$\begin{aligned} V &= -\mu \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} J_i(P) & 0 \\ 0 & J_p(P) \end{bmatrix}^+ \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_i(P) \\ e_p(P) \end{bmatrix} \\ &= -\mu J_i(P)^+ e_i(P), \end{aligned} \quad (33)$$

and in the second case, particularly when  $\lambda_1 = 0$  as

$$\begin{aligned} V &= -\mu \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} J_i(P) & 0 \\ 0 & J_p(P) \end{bmatrix}^+ \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} e_i(P) \\ e_p(P) \end{bmatrix} \\ &= -\mu J_p(P)^+ e_p(P). \end{aligned} \quad (34)$$

In these two cases, classical stability analysis can be done following the methodology given in Section 3.2 and detailed in Chaumette and Hutchinson (2006).

In the third case, we assume each factor  $\lambda_i$  varies smoothly with respect to time. We have here the general case described in Equation (29). We originally proposed this case in Abdul Hafez and Jawahar (2007). Here, the stability analysis is reconsidered. However, we follow discussion presented in subsection 3.2

to show the local stability of the system. Local stability is guaranteed as long as the matrices  $J_p(P)^+$  and  $J_i(P)^+$  are good approximations of the inverse Jacobean of the matrices  $J_p(P)$  and  $J_i(P)$  respectively. This means that

$$J_h(p)J_h(p)^+ > 0. \quad (35)$$

The factors  $\lambda_1 > 0$  and  $\lambda_2 > 0$  can help simplifying the stability analysis since they are always positive. This ensures the positive definite condition of the matrix given above in Eq. (35). One thing to be noticed is that the configuration may reach a potential local minima when  $\Lambda e_h(p) \in J_h(p)^T$ . This case of potential local minima is described in Chaumette (1998).

#### 4. HANDLING THE CONSTRAINTS BY THE IMPORTANCE FACTORS $\lambda_1$ AND $\lambda_2$

The weighting factor  $\lambda_1$ , is assigned to 2D features, based on an error function that measures the possibility of a feature to get out of the camera field of view. This means whenever a 2D feature approaches the image border, the value of the weighting factor  $\lambda_1$  will be increased. Higher value of  $\lambda_1$  allows the behavior of IBVS to dominate. Similarly, the value of weighting factor  $\lambda_2$  are computed based on an error function in the robot joint space. Next subsection shows the computation of these weights  $\omega_{2D}$  and  $\omega_{3D}$  in the third *i.e.* the general case then, in the later subsection, the case 1 and case 2, we call the special cases, are presented.

##### 4.1 The general case

As we have seen in the previous section, case one is reduced to IBVS and case two is reduced to PBVS. The computation of the factors  $\lambda_1$  and  $\lambda_2$  in the third case *i.e.* the general case will be done based on this fact. The computation is done in such a way that whenever the behaviour of IBVS is required, larger value is given to  $\lambda_1$ . Similarly, whenever the behaviour of PBVS is required, larger value is given to  $\lambda_2$ . We explain below the method used to compute these importance factors  $\lambda_1$  and  $\lambda_2$ . In the remaining of this subsection, we call  $\lambda_1$  as the importance factor for IBVS, and call  $\lambda_2$  as the importance factor for PBVS.

The importance weight of the PBVS is inversely proportional to the ability of following the straight line camera path. Hence, the importance factor controller of position-based with respect to one arm joint can be written to be inversely proportional to the end-effector deviation from the straight line path as follows:

$$\omega_{3D} = \frac{1}{\sqrt{2\pi}\sigma_r} \exp\left[-\frac{r^2}{2\sigma_r^2}\right]. \quad (36)$$

Here,  $\sigma_r$  is a selected threshold, and  $r$  is the distance of the end-effector to the straight line path. A plot of the importance weights of position-based basic controller with respect to the distance to the image border is illustrated in Fig. 3.

The importance weight of the IBVS is proportional to the weakness of the performance of position-based visual servoing. The performance of position-based vision control is measured by the ability of keeping the point features  $(u^i, v^i)$  visible in the camera field of view. The error in the performance of position-based vision control can be measured as a function of the distance of the  $i$ th point to the nearest image border. Let the parameter  $\{d_t^i\}_{i=1}^N$  be the distance of the  $i$ th point to the nearest image border at time  $t$ , where

$$d^i = \min\{u^i - m_0, v^i - n_0, m_0 - u^i, n_0 - v^i\}, \quad (37)$$

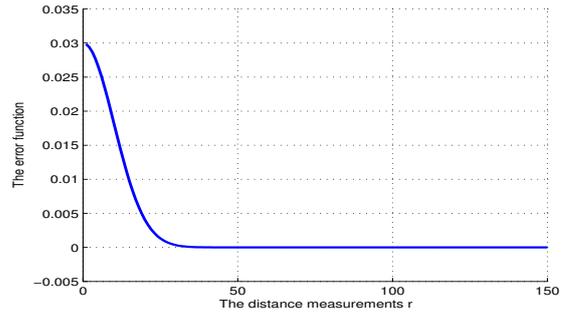


Fig. 3. The importance function of the 3D visual information. The importance factor is a function of the distance  $r$ . When the working point is far from the straight line, the factor is zero value.

where  $m_0$  and  $n_0$  are margins to the image borders, and  $N$  is the number of image points. Hence, the importance factor controller of image-based with respect to one image point can be written to be inversely proportional to this distance as follows:

$$\omega_t^i = \frac{1}{\sqrt{2\pi}\sigma_d} \exp\left[-\frac{d^{i(2)}}{2\sigma_d^2}\right]. \quad (38)$$

Here,  $\sigma_d$  is a selected threshold such that only image points within a minimum distance to its image border will contribute to the importance weight. We get the following importance weight for  $i$ th image point. Finally, the importance weight image-based vision control algorithm is given as

$$\omega_{2D} = \max_i\{\omega_t^i\}_{i=1}^N. \quad (39)$$

A plot that shows the image borders along with the margins  $m_0$  and  $n_0$  to compute the distance to the image border is illustrated in Fig. 4.

One thing that we may note from curves of the importance factors is that each one is bounded with a maximum value, but this is not the same in the first two cases. The two values of the factors are normalized to play the role of balancing the amount of 2D and 3D information need to be integrated and hence contribute to the velocity vector of the camera computed by the visual controller.

##### 4.2 The special reduced cases

The discussion here in this subsection is about the other first and second cases. In case one, it is assumed that IBVS is the default controller. This means that the trajectories in the image space are a straight lines. In addition, since the 3D trajectories is a complex 3D curve, arm joints may reach its limits. Whenever a joint approaches its limit, a higher value is assigned to  $\lambda_2$ . Similarly, PBVS is assumed to be the default controller in case two. The 3D trajectory is straight line while the image trajectories may move out of the camera field of view. Whenever an image feature approaches the image border, a higher value is assigned to  $\lambda_1$ . The work presented later by Kermorgant and Chaumette proposed in Kermorgant and Chaumette (2011) is similar to the case two of our work in the sense that it assumes the default scheme is the PBVS that produces a straight line 3D trajectory. It is different in the sense that in Kermorgant and Chaumette (2011) a different weight is assigned to every different image feature, and the weights are unbounded hence they can grow up to infinity.

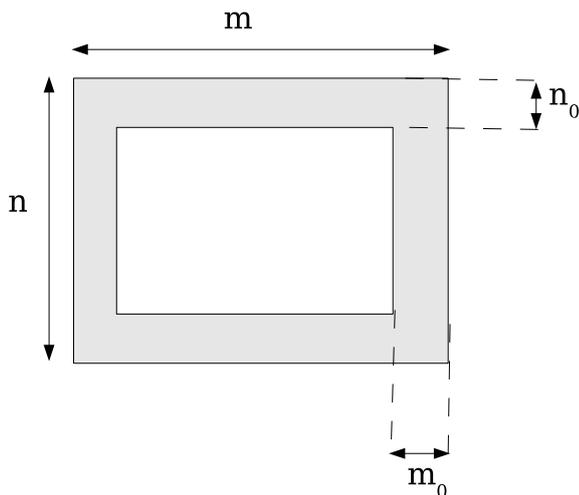


Fig. 4. The margins to the image borders that affect the values of the importance factor of the 2D visual information.

The work recently proposed in Kermorgant and Chaumette (2011) presents a different weighting strategy in a similar framework. This strategy assumes that  $\Lambda_2 = \text{diag}(\lambda_2) = I_6$ . In other words, the importance factor of 3D information is  $\lambda_2 = 1$ . In contrast,  $\Lambda_1 = \text{diag}(\lambda_{1i})$ , where  $\lambda_{1i}$  is the importance factor of the 2D information associated with the image feature  $(u^i, v^i)$ . In this case, whenever a 2D feature  $(u^i, v^i)$  is going to leave the camera FoV, the importance factor  $\lambda_{1i}$  associated with it is increased. To bring more attention to the feature near the border, the weight  $\lambda_1$  will go to infinity when the image feature go near to image border. Briefly, the servoing process starts by default as PBVS, and start injecting 2D features  $(u^i, v^i)$  whenever it approaches any of the image borders. Due to the effect of measurement errors and other types of noise, the contribution of the concern feature to the velocity signal may not be enough in this case to bring the feature back into the image toward its desired position. Another issue to be noted in this work is that the weights of image features are going to infinity when many features are very near the border. This may gives large amount of contribution to IBVS scheme, hence the arm may reaches its limits. In addition, the high gain which is earned in such situation may cause stability problem to the servoing system.

Another close work but previous to ours is the 2 1/2 D visual servoing error function that was proposed in Malis et al. (1999) and has been reviewed in Subsection 2.3 from this paper. In Malis et al. (1999), the features are selected as a 6-vector. The first 3-vector is  $s_{2D}(P) = [x, y, \log(Z)]^T$  that contains the 2D visual information from the image space. These features are used to recover the position of the camera, where the another 3-vector is  $s_{3D}(P) = [C^*(u\theta)_C]^T$  that contains the 3D visual information from the pose space is used to recover only the orientation of the camera. Let us rewrite the weighting matrices in the control law given in Equation (29) as  $\tilde{\Lambda}_1 = \text{diag}(\lambda_{1i})$ , where  $i = 1, \dots, 2N$ ; and  $\tilde{\Lambda}_2 = \text{diag}(\lambda_{2j}) = I_6$ , where  $j = 1, \dots, 6$ . We find that the 2 1/2 D visual servoing is fitting within our frame work by assigning  $\{\lambda_{1i} = 0\}_{i=3}^{2N}$  and  $\{\lambda_{2j} = 0\}_{j=1}^3$ .

In contrast, our method uses each of the full image space information and pose space information to recover the full

	$T_x$	$T_y$	$T_z$	$R$	$P$	$Y$
Initial pose 1st task	0.0	-1.7	0.0	0.0	0.0	-1.57
Desired pose 1st task	0.0	-1.7	0.0	0.0	0.78	-1.57
Initial pose 2nd task	2.15	-2.09	0.0	0.8	-0.6	-1.57
Desired pose 2nd task	0.0	-1.5	0.0	0.0	0.0	-1.57
Object pose	0.0	0.0	0.0	0.0	0.0	-1.57

Table 1. The initial and desired camera pose, and object frame pose with respect to the world reference frame.

	$X$	$Y$	$Z$		$X$	$Y$	$Z$
$P_1$	0.0	0.0	0.0	$P_2$	0.25	0.25	0.0
$P_3$	-0.25	0.25	0.0	$P_4$	-0.25	-0.25	0.0
$P_5$	-0.25	0.25	-0.2	$P_6$	-0.2	-0.2	0.1

Table 2. The 3D points coordinates in the object frame.

camera pose *i.e.*, both position and orientation. The 2D features are  $s_{2D}(P) = [x_1, y_1, \dots, x_N, y_N]^T$ , *i.e.*, the coordinates of the considered image points. The 3D features  $s_{3D}(P) = [T, u\theta]^T = [T_x, T_y, T_z, u\theta]^T$ , *i.e.*, the pose vector. Both 2D  $s_{2D}(P)$  and 3D features  $s_{3D}(P)$  are concatenated into a 12-vector and minimized together. In such a minimization method, the minimization process searches for a least squares solution that minimizes the 2D error from the image space and the 3D error from the pose space simultaneously. This produces an optimal performance in both spaces.

## 5. RESULTS AND DISCUSSION

We present simulation experiments where the proposed methods are compared to previous method like IBVS, PBVS, and to hybrid methods, namely 21/2D visual servoing. These methods in addition to our proposed method are implemented in a simulation framework. The simulation assumes a perspective camera model with  $1000m$  focal length and unit aspect ratio. The servoing target object consists of six non-planar points. The object pose, and initial and desired camera pose with respect to a world reference frame are given in Table 1. The object point coordinates are given in Table 2.

As we have mentioned in the introduction, the Cartesian camera path and the feature image trajectory reflect the behavior of the working point of the optimization process. Through the experiments done in this section, we evaluate the camera path resulting from optimizing our hybrid objective function, showing that it is improved with respect to the basic visual servoing algorithms as well as with respect to other state of the art hybrid visual servoing method. This can be noted where our algorithm produces much shorter camera path. The same discussion can be applied to the image trajectory, which it is nothing but the projection of the working point to the image space. Similarly, we are interested in producing an image trajectory that is within the image area.

The experiments performs a comparison through two positioning tasks. The first one is a  $\pi/4$  rad rotation error around the camera optical axis. It is the first task presented above. This task is useful to evaluate the efficiency of the proposed methods compared to previous works in the literature like IBVS and its improved version, 21/2D visual servoing. The task is achieved using our proposed method. Results are compared to IBVS and 21/2D VS. The camera retreat in each method is evaluated by the magnitude of the screw velocity  $V_z$  along the camera optical

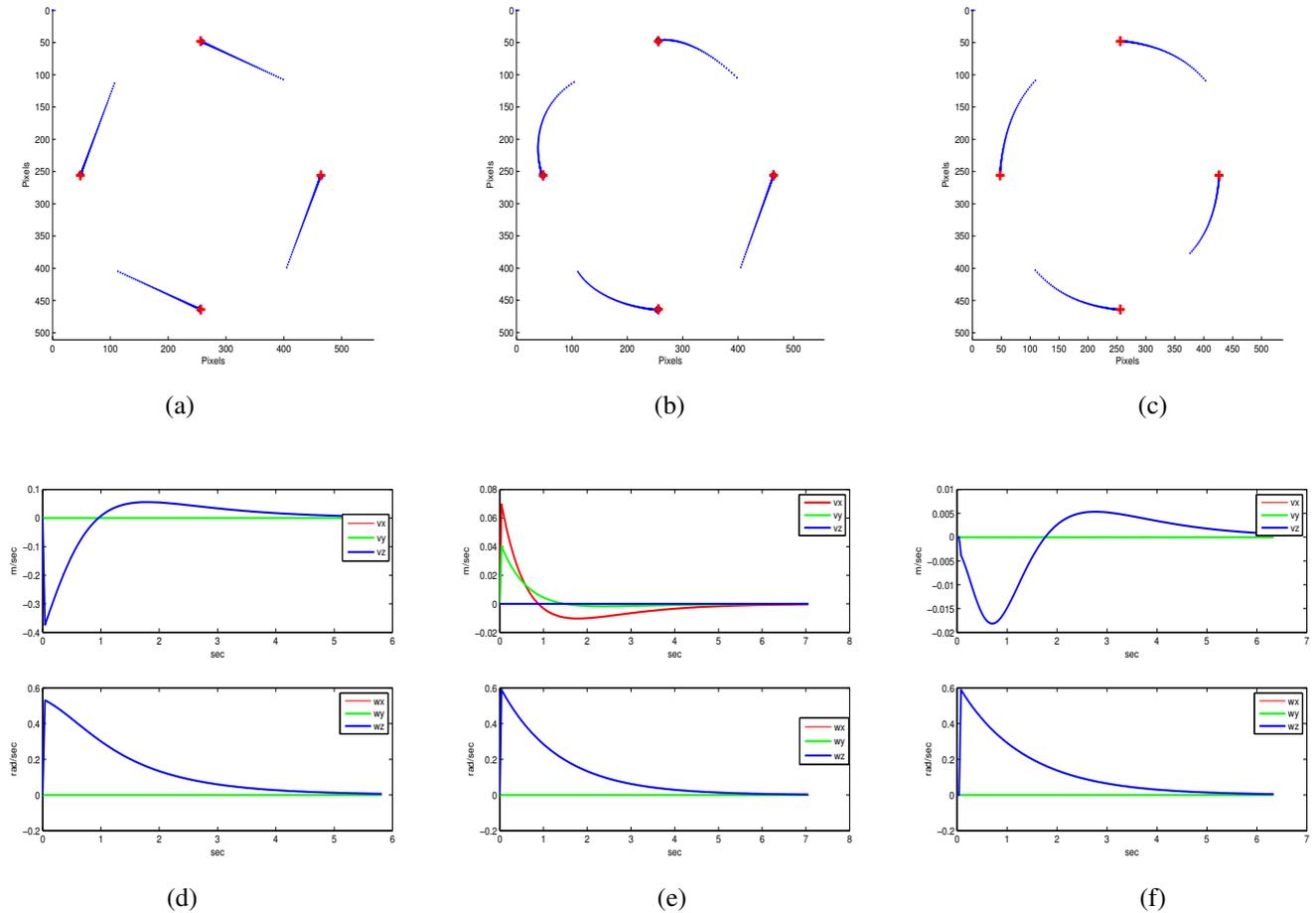


Fig. 5. The image features trajectories and the screw velocity. Image-based visual servoing in (a) and (d). 21/2D visual servoing (b) and (e), and our proposed method in (c) and (f). The desired positions of the image features are marked by +. The translational and rotational velocities are measured by m/sec and rad/sec respectively. The image-based method shows considerable amount of camera retreat, i.e. high value of velocity  $V_z$  along the axis z. In contrast, it much smaller values in case of using 21/2D and our proposed methods.

axis and finally the performance of our proposed method is compared to the IBVS and 21/2D VS methods.

The second task is a general positioning task that contains a rotation and translation errors. It is the last task presented above. This task is useful to evaluate the camera trajectory in the Cartesian space. Camera trajectories resulting from our proposed method and methods like PBVS and 21/2D VS are compared here.

### 5.1 Camera Retreat from Rotation Error

The experiments for the first task is done using our proposed, IBVS and 21/2D. The results are shown in Figure 5. In case of using IBVS, the camera retreat along the camera optical axis is demonstrated in Figures 5(a) and 5(d). These figures show the image trajectory and screw velocity respectively. A nice image trajectory was obtained while a considerable pure backward translation motion (camera retreat) was observed. The task is completed properly using 21/2D method. This is shown in Figures 5(b) and 5(e). It is clear that there is no notable translation motion while the rotation error decreased properly to zero. In other words, the task has been done without camera retreat.

The results in case of our method are shown in Figures 5(c), 5(f) respectively. The proposed method shows an image trajectory similar to the one obtained using 21/2D method. However, a moderately small camera retreat is observed.

### 5.2 Camera Trajectory in the Cartesian Space

Here, we consider the second task. This is a general task with both translational and rotational motion. First we present the results from PBVS where the camera trajectory in the Cartesian space is a straight line. This is the shortest camera path. Figures 6(a) and 6(d) show the features trajectory in the image space and camera trajectory in the Cartesian space respectively. However the image trajectory is an undesirable complex curve and some features got out of the camera field of view. In contrast, 21/2D visual servoing, as shown in Figures 6(b) and 6(e), has improved the image trajectory, but nothing about the Cartesian camera path, it is not a straight line at all.

Using our proposed method produces, as illustrated in Figures 6(c) and 6(f), a camera trajectory that is not only very close to a straight line, but also with fine image trajectory.

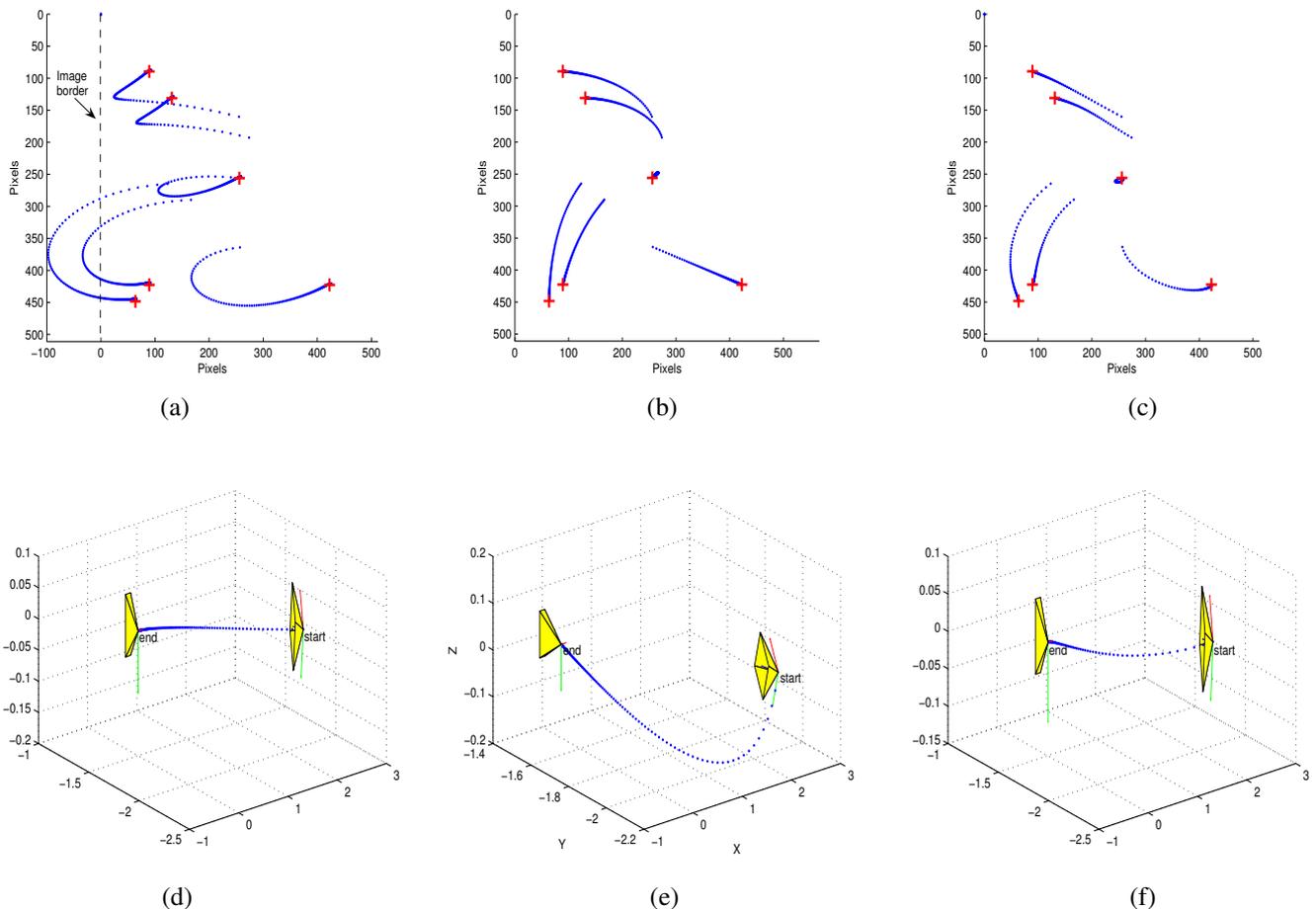


Fig. 6. The image features trajectories and the camera trajectory in the Cartesian space. PBVS visual servoing in (a) and (d), 21/2D visual servoing in (b) and (e), and our proposed method in (c) and (f). The desired positions of the image features are marked by +. The PBVS methods produces a straight line but the features leave the camera field of view. In 21/2D method one feature follows straight line in the image and remaining features' trajectory is acceptable, hence the camera trajectory is unpredictable curve. In contrast, our proposed method shows very near to the straight line camera trajectory along with image trajectories that keep the features in the field of view.

In general, our proposed methods show an improved performance in image space and Cartesian space together. In both methods, image features are less probable to leave the image while the camera performs less retreat in the Cartesian space. An improvement is observed in the trajectories in the image space for all considered points. Most of the previous hybrid methods improve the path of only one point in the image.

## 6. SUMMARY

In this paper, we presented the visual servoing problem using a minimization framework. The two traditional IBVS and PBVS methods along with previous hybrid methods like 2 1/2 D VS are also defined using the same minimization framework. The IBVS uses an error function consists of 2D information from the image, while PBVS uses an error function consists of 3D information from the Cartesian space. Our major contribution is presented here as an extended hybrid objective function. This proposed hybrid function concatenates the two individual 2D and 3D error vector in one extended vector. Here, the full 2D and 3D information are used in the process, in the general case, comparing with the earlier hybrid methods that partially use the

2D and 3D information in the visual servoing process. Stability analysis is presented. A methods for proper computations of the weights that satisfy the visibility and path constraints are shown. Experiments have been carried out within a simulation framework. The results are analyzed and a comparison of our proposed method to classical and hybrid methods is presented using some of the canonical tasks stated in the literature. Experiments show that the proposed methods outperform the state of the art method in both the image and Cartesian spaces.

## REFERENCES

- Abdul Hafez, A.H., Cervera, E., and Jawahar, C.V. (2013). Stable hybrid visual servo control by a weighted combination of image-based and position-based algorithms. *International Journal of Control and Automation*, 6(3), 149–164.
- Abdul Hafez, A.H. and Jawahar, C.V. (2006a). Improvement to the minimization of hybrid error functions for pose alignment. In *IEEE Int. Conf. on Automation, Robotics, Control, and Vision, ICARCV'06*, 1–6. Singapore.
- Abdul Hafez, A.H. and Jawahar, C.V. (2006b). Target model estimation using particle filters for visual servoing. In *IEEE Int. Conference on Pattern Recognition, ICPR'06*. Hong Kong.

- Abdul Hafez, A.H. and Jawahar, C.V. (2007). Visual servoing by optimization of a 2D/3D hybrid objective function. In *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, 1691–1696. Roma, Italia.
- Benhimane, S. and Malis, E. (2007). Homography-based 2d visual tracking and servoing. *Special Joint Issue IJCV/IJRR on Robot and Vision. Published in The International Journal of Robotics Research*, 26(7), 661–676.
- Benhimane, S., Najafi, H., Grundmann, M., Malis, E., Genc, Y., and Navab, N. (2008). Real-time object detection and tracking for industrial applications. In *International Conference on Computer Vision Theory and Applications*.
- Cai, C., Dean-Leon, E., Mendoza Gallegos, D., Somani, N., and Knoll, A. (2013). Uncalibrated 3d stereo image-based dynamic visual servoing for robot manipulators. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'13*. Tokyo, Japan.
- Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and A. Morse (eds.), *The Confluence of Vision and Control*, 66–78. LNCIS Series, No 237, Springer-Verlag.
- Chaumette, F. and Hutchinson, S. (2006). Visual servo control, part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4), 82–90.
- Chaumette, F. and Hutchinson, S. (2007). Visual servo control, part II: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1), 109–118.
- Cheah, C.C., Wang, D.Q., and Sun, Y.C. (2007). Region-reaching control of robots. *Transaction on Robotics.*, 23(6).
- Cherubini, A. and Chaumette, F. (2012). Visual navigation of a mobile robot with laser-based collision avoidance. *Int. Journal of Robotics Research*.
- Cherubini, A., Chaumette, F., and Oriolo, G. (2011). Visual servoing for path reaching with nonholonomic robots. *Robotica*, 29(7), 1037–1048.
- Corck, P. and Hutchinson, S. (2001). A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 14(4), 507–515.
- Dani, A., Panahandeh, G., Chung, S.J., and Hutchinson, S. (2013). Image moments for higher-level feature based navigation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'13*. Tokyo, Japan.
- Deguchi, K. (1998). Optimal motion control for image-based visual servoing by decoupling translation and rotation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'98*, volume 2, 705–711.
- Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3), 313–326.
- Garcia-Aracil, N., Malis, E., Aracil-Santonja, R., and Perez-Vidal, C. (2005). Continuous visual servoing despite the changes of visibility in image features. *IEEE Trans. on Robotics and Automation*, 21(6).
- Isidori, A. (1995). *Nonlinear control systems*. Springer-Verlag, New York, 3rd edition.
- Janabi-Sharifi, F. and Marey, M. (2010). A kalman-filter-based method for pose estimation in visual servoing. *IEEE Trans. on Robotics*, 26(5), 939–947.
- Kermorgant, O. (2013). Partial visibility constraint in 3d visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'13*. Tokyo, Japan.
- Kermorgant, O. and Chaumette, F. (2011). Combining ibvs and pbvs to ensure the visibility constraint. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'11*, 2849–2854.
- Khalil, H.K. (2002). *Nonlinear Systems*. Prentice Hall, 3rd edition.
- Kragic, D. (2001). *Visual servoing for manipulation: Robustness and integration issues*. Ph.D. thesis, Royal Institute of Technology, Sweden.
- Kyrki, V., Kragic, D., and Christensen, H.I. (2004). New shortest-path approaches to visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, 349–354. Sendai, Japan.
- Li, B., Fang, Y., and Zhang, X. (2013). Uncalibrated visual servoing of nonholonomic mobile robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'13*. Tokyo, Japan.
- Luca, A.D., Oriolo, G., and Giordano, P.R. (2007). On-line estimation of feature depth for image-based visual servoing schemes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1–6.
- Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, 1843–1848. New Orleans, USA.
- Malis, E. and Chaumette, F. (2002). Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Trans. on Robotics and Automation*, 18(2), 176–186.
- Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 D visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2), 238–250.
- Malis, E., Chaumette, F., and Boudet, S. (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.
- Mansard, N., Remazeilles, A., and Chaumette, F. (2009). Continuity of varying-feature-set control laws. *IEEE Trans. on Automatic Control*, 54(11), 2493–2505.
- Samson, C., Espiau, B., and Borgne, M. (1991). *Robot Control: The task Function approach*. Oxford University Press, Oxford, England.
- Unger, C., Benhimane, S., Wahl, E., and Navab, N. (2009). Efficient disparity computation without maximum disparity for real-time stereo vision. In *British Machine Vision Conference (BMVC)*. London.
- Wilson, W., Hulls, C.C.W., and Bell, G.S. (1996). Relative endeffector control using Cartesian position-based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5), 684–696.
- Yap, Jr., T.N., Li, M., Mourikis, A.I., and Shelton, C.R. (2011). A particle filter for monocular vision-aided odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1–6.