Stable Hybrid Visual Servo Control by a Weighted Combination of image-based and position-based Algorithms

A. H. Abdul Hafez[1,2], Enric Cervera[3] and C.V. Jawahar[1]

[1] *CVIT, IIIT, Hyderabad, India*
[2] *Dept. of CSE, Aurora's Technological and Research Institute, Hyderabad, India*
[3] *Dept. of CSE, Universitat Jaume-I, Castell, Spain*
ah.abdulhafez@gmail.com, jawahar@iiit.ac.in, ecervera@icc.uji.es

Abstract

In this paper, we present a novel stable hybrid vision-based robot control algorithm. This method utilizes probabilistic integration of image-based and position-based visual servo controllers to produce an improved vision-based robot control. A probabilistic framework is employed to derive the integration scheme. Appropriate probabilistic importance functions are defined for the basic two algorithms to characterize their suitability in the task. The integrated algorithm has superior performance both in image and Cartesian spaces. Experiments validate this claim.

1 Introduction

Visual servo control or what is also called visual servoing is a widely used vision-based robot control scheme. In visual servoing, the control loop of the robot is closed using the visual information. Many visual servoing algorithms have been proposed in the literature [1, 2, 3, 4, 5, 6]. They primarily differ in the selected features and the error functions to minimize. There are 2D/image-based (IBVS) and 3D/position-based (PBVS) visual servoing algorithms. Chaumette, in [7], has shown that each of these two basic categories has drawbacks or potential problems. The drawbacks could be observed either in the image space such as the inability to keep the features visible during the servoing process [8], or in the Cartesian space such as the inability to keep the arm in its Cartesian/joint space [9]. This is in addition to the local minima avoidance (convergence) problem [10, 11]. Due to the complementary limitations in these two algorithms, hybrid methods have been recently evolved to integrate the advantages and discard the drawbacks. In addition to providing accurate control signal, research on hybrid visual servoing has focused on addressing the issues mentioned above that includes: feature visibility [7, 10, 11, 12], local minima avoidance [7, 9], convergence speed [8, 13, 14], continuous control signal [8, 13, 14], and shortest camera path [11, 13, 14, 8, 10].

Many of the hybrid methods address the above mentioned issues by integrating the 2D and 3D information in the feature space [10, 11, 15] or in the action space [16, 13, 14, 8]. The 2 1/2 D visual servoing method [11] decouple the rotation from the translation considering one visible image point during the servoing process. The most comprehensive solutions are

based on potential fields [17, 9], which was originally introduced to the robotic community by Khatib in [18] as a solution to the collision avoidance problem. An algorithm which switches between image-based and position-based vision control algorithms is presented by Gans and Hutchinson [13, 14]. In their hybrid switching method, the IBVS and PBVS run independently. Depending on which constraints are satisfied, a logical decision system switches between them. Since this method switches between IBVS and PBVS in binary form, local minima may be reached. Another method for switching is presented by Chesi *et al.* [8]. It switches between elementary camera motions, mainly rotation and translation extracted by decomposing the homography matrix between the current and desired views. In these two switching methods, the control signal suffers from discontinuity when features approach the image border. They need large amount of time for convergence. Hafez *et al.* presented in [19, 16] a smooth linear combination of different visual servoing algorithms. The combining weights are computed in [19] using an error function of the weakness of the concern algorithm using the boosting algorithm from machine learning theory. Later in [12], Kermorgant and Chaumette proposed a similar work with a different computation method of weights. Another hybrid method based on potential fields [9] has also been proposed for path planning in the image space. This method introduces the visibility and robot joint limits constraints into the design of the desired trajectories. Essentially, this is a local path planning method. The local minima are not ensured to be avoidable when repulsive and attractive fields are equal.

The main contribution of this paper is a stable probabilistic framework for integration of different visual servoing algorithms. The problem is reduced to the computation of the weighted sum of the output of the individual basic algorithms. A new method for computation of weights is proposed using importance functions. These functions are defined for the individual algorithms. In the remaining of this paper, we present a background about visual servoing and its basic schemes i.e. IBVS and PBVS in Section 2. The main contribution of this paper is Section 3. It presents the probabilistic framework, the integration rule for IBVS and PBVS, the weight computation method for IBVS and PBVS, and the stability analysis of the proposed hybrid algorithm. The last section provide the experimental evaluation of the proposed control algorithm.

2 Background of Visual Servoing

2.1 Image-based visual servoing

In Image-based visual servoing, the task function is defined with respect to the error $e(s)$ in the image space, where s is the vector of the current features position and s^* is the vector of the desired one. The error using IBVS can be written as

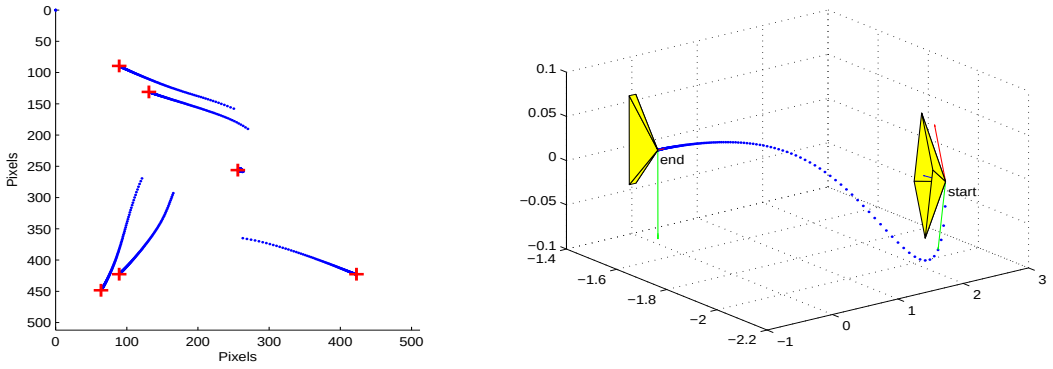
$$\dot{e}_{im}(s) = J_{im} V_{im}. \tag{1}$$

By assuming exponential convergence of the error $\dot{e} = -\lambda e$, we get

$$V_{im} = -\lambda J_{im}^{+} e_{im}(s), \tag{2}$$

where J_{im}^{+} is the pseudo-inverse of the image Jacobian matrix J_{im} [1].

It can be shown that the feature trajectory in the image space is a straight line while the camera trajectory in the Cartesian space is unpredictable. See Fig. 1 for a visualization.



(a) (b)

Figure 1. Feature trajectories in the image space in (a), and the camera trajectory in the Cartesian space in (b) for the image-based visual servoing algorithm. The desired positions of the image features are marked by +.

From any initial state, IBVS moves the image points straight towards their desired positions in the image. This is subject to the availability of a good estimate of the depth and robust image measurements. IBVS is proved to be asymptotically locally stable, but the global stability is not ensured since unpredictable image local minima and Jacobian singularity may occur at any time.

2.2 Position-based visual servoing

In position-based visual servoing, the camera velocity is defined as a function of the error between the current and desired camera pose. This error is the transformation T_C^{C*} represented as a (6×1) vector $e_p(s) = [t_C^{C*}, u\theta]^T$. The error using IBVS can be written as

$$\dot{e}_{po}(s) = J_{po} V_{po}. \tag{3}$$

By assuming exponential convergence of the error $\dot{e} = -\lambda e$, we get

$$V_{po} = -\lambda J_{po}^{-1} e_{po} \tag{4}$$

where J_{po} is the interaction matrix [1].

While PBVS minimizes the error function in the Cartesian space, the camera trajectory is a straight line, but the feature trajectory is not predictable and may get out of the camera field of view. See Fig. 2 for a visualization. The PBVS method is known to be globally asymptotically stable and does not suffer from any local minima or Jacobian singularity. The global stability is subject to an accurate estimate of the pose.

3 The Integration Framework

3.1 Probabilistic integration framework

It was shown in [16] that the integration framework of IBVS and PBVS can be formulated in probabilistic context. This deals with the velocity control signal V_i as probabilistic variable conditioned to the image measurements x.

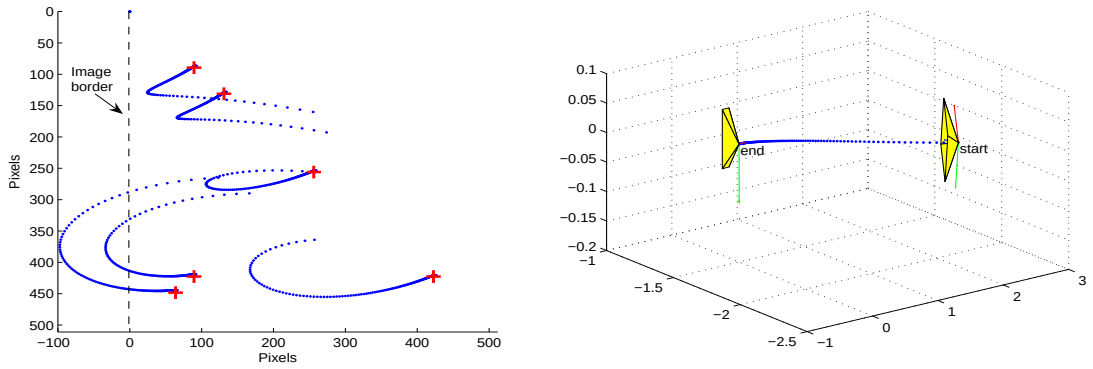


Figure 2. Feature trajectories in the image space in (a), and the camera trajectory in the Cartesian space in (b) for the position-based visual servoing algorithm. The desired positions of the image features are marked by +.

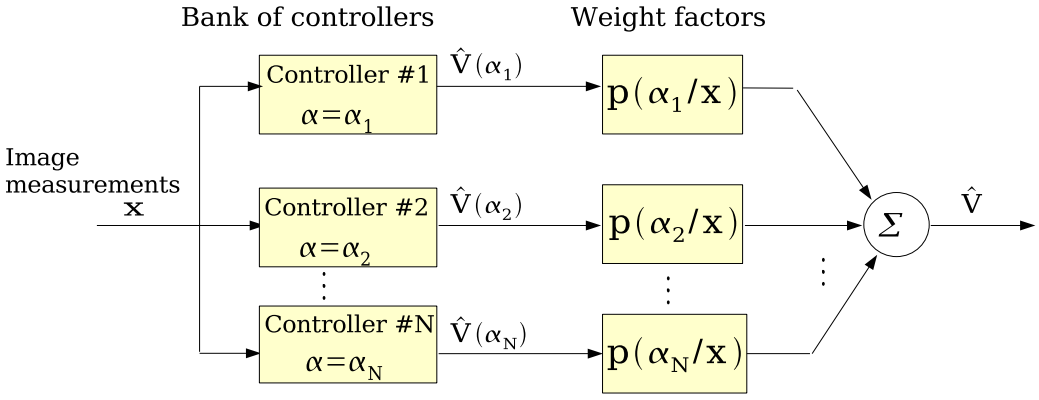


Figure 3. Weighted sum of the velocity estimates

Let $\mathcal{V} = \{V_i\}$, where $\mathcal{V} \subseteq R^6$ be a set of possible states of the velocity screw vector commanded to the robot arm controller. The probability that the velocity vector V uses the value V_i conditioned to the image measurements vector x is

$$p(V = V_i|x) = p(V_i|x). \tag{5}$$

By convention, these probabilities sum to unity as $\int_{\mathcal{V}} p(V_i|x) = 1$, and in the discrete case $\sum_{\mathcal{V}} p(V_i|x) = 1$. The conditioned expected value of the velocity vector over the universe $\mathcal{V}$ is

$$\hat{V} = E(V_i|x) = \int_{\mathcal{V}} V_i \; p(V_i|x) \; dV_i. \tag{6}$$

Here $\hat{V}$ is called the conditioned mean or expected value.

Let us define another parameter α_i of the visual servoing process. The candidature visual servoing sub-controllers which are represented in the universe $\mathcal{V}$ are represented by α_i, where $i = 1, ..., N$. Here N is the number of visual servoing schemes or the sub-controllers. In other words, we have N individual visual servoing control laws represented by the parameters α_i, where $i = 1, ..., N$. For example, an integration can be done between IBVS and PBVS control laws, then i takes the values $\{im, po\}$. It is possible to rewrite the probability function $p(V_i|x)$ as

$$p(V_i|x) = \int_{\alpha} p(V_i, \alpha|x) \; d\alpha = \int_{\alpha} p(V_i|\alpha, x) \; p(\alpha|x) \; d\alpha. \tag{7}$$

By substituting (7) in (6), we can write

$$\hat{V} = \int_{\mathcal{V}} V_i \; \int_{\alpha} p(V_i|\alpha, x) \; p(\alpha|x) \; d\alpha \; dV_i,$$

$$\hat{V} = \int_{\alpha} p(\alpha|x) \int_{\mathcal{V}} V_i \; p(V_i|\alpha, x) \; dV_i \; d\alpha. \tag{8}$$

The inner integral is the estimate of the velocity for a given α and therefore

$$\hat{V}(\alpha) = \int_{\mathcal{V}} V_i \; p(V_i|\alpha, x) \; dV_i.$$

The Eq.(8) can be rewritten as

$$\hat{V} = \int_{\alpha} p(\alpha|x) \; \hat{V}(\alpha) \; d\alpha, \tag{9}$$

and in the discrete case

$$\hat{V} = \sum_{\alpha_i} \hat{V}(\alpha_i) \; p(\alpha_i|x), \quad \text{where } \sum_{\alpha_i} p(\alpha_i|x) = 1. \tag{10}$$

Here $p(\alpha_i|x)$ is the discrete probability (importance weight) for α_i, conditioned on the image measurement x. Equation (10) says that the optimal estimate of the velocity vector is a weighted summation of the velocity values computed from each individual control law. This is shown in Fig. 3. The problem is now reduced to the computation of the velocity from each individual controller in addition to the computation of the probabilities (importance weights) using each of these individual controllers. These weights are computed using the image and Cartesian constraints to be satisfied during the process of visual servoing.

3.2 The Integration Rule of IBVS and PBVS Controllers

Two individual image-based visual servoing and position-based visual servoing control laws can be integrated using (10)

$$\hat{V} = \hat{V}(\alpha_{im})\ p(\alpha_{im}|x) + \hat{V}(\alpha_{po})\ p(\alpha_{po}|x). \tag{11}$$

Here, $\hat{V}(\alpha_{im})$ is the velocity computed using the IBVS controller given in (2). The velocity $\hat{V}(\alpha_{po})$ is computed using the PBVS controller given in (4). The probability of using each individual control law $p(\alpha_i|x)$, conditioned on the image measurement x, is determined by the image and Cartesian constraints to be satisfied during the process of visual servoing. The probabilistic weights $\omega_{im} = p(\alpha_{im}|x)$ and $\omega_{po} = p(\alpha_{po}|x)$ given in (11) can be normalized in such a way that it sums up to one, to satisfy (10). The normalized term $\frac{p(\alpha_{im}|x)}{p(\alpha_{im}|x)+p(\alpha_{po}|x)}$ can be denoted as ω and $\frac{p(\alpha_{po}|x)}{p(\alpha_{im}|x)+p(\alpha_{po}|x)}$ can be denoted as $1-\omega$. Taking into account Eq (10), we can write

$$V = \omega V_{im} + (1-\omega)V_{po}, \tag{12}$$

where $0 < \omega < 1$. In fact, the weights are computed in such a way to avoid both the case of pure IBVS when $\omega = 1$ and the case of PBVS when $\omega = 0$.

In the remaining of this paper, we consider that the image and joint measurement are deterministic. This assumption simplifies the computation of the probabilistic importance weights. These weights are simply computed as proportional to the performance of the concern control law. In other words, we can consider the importance of one controller as a function of the weakness in the performance of the remaining controllers.

3.3 Computation of the Importance weights for classical visual servoing

Here in this paper, we assume that we only have image-based and position-based visual servoing controllers. The importance weight of the IBVS is computed based on how much the performance of the PBVS is undesirable . In contrast, The importance weight of the PBVS is computed based on how much is the performance of the IBVS undesirable. We explain below the method used to compute these importance vector for both the IBVS and PBVS.

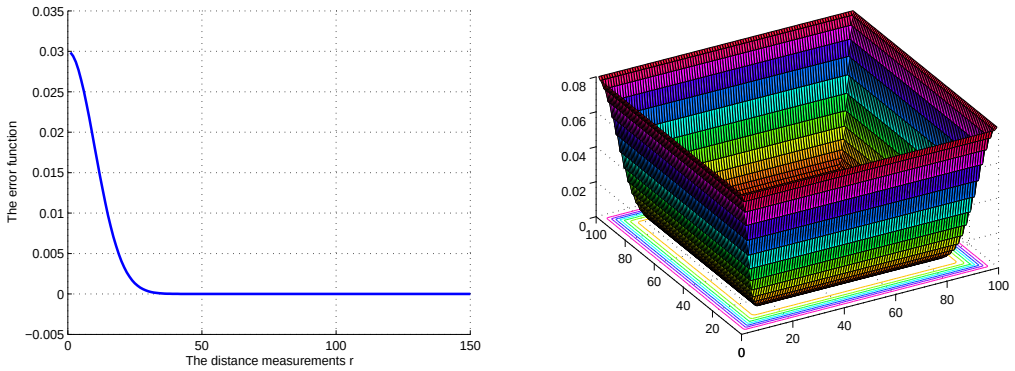
3.3.1 Weights for position-based vision control

The importance weight of the PBVS is proportional to the weakness of the performance of image-based visual servoing. The performance of image-based vision control is measured by the ability of the working point q^i of the ith arm joint to avoid the joint limits $\{q^i_{min}, q^i_{max}\}$ of the robot arm. The configuration $\mathbf{q}$ of joints of a robot arm is acceptable when

$$\forall i, q^i \in \left[q^i_{min} + \theta^i_q, q^i_{max} - \theta^i_q\right]. \tag{13}$$

Here, θ^i_q is a threshold of the ith joint. The weakness in the performance of image-based control algorithm can be measured as the distance of the working point q^i of the ith joint to its concerned joint limit θ^i. Let the parameter $\{r^i_t\}^N_{i=1}$ be the distance of the joint q^i to its threshold θ^i_q at time moment t, where

$$r^i = \min\{q^i - q^i_{min} - \theta^i_q, q^i_{max} - \theta^i_q - q^i\} \tag{14}$$



(a) (b)

Figure 4. The importance function of position-based control algorithm in (a) and image-based control algorithm in (b).

and N is the number of the joints of the arm. Hence, the importance factor controller of position-based with respect to one arm joint can be written to be inversely proportional to this distance as follows:

$$\omega_t^q = \frac{1}{\sqrt{2\pi}\sigma_q} \exp\left[-\frac{r^2}{2\sigma_q^2}\right]. \tag{15}$$

Here, σ_q is a selected threshold such that only working points within a minimum distance to its joint limits will contribute to the importance weight. We get the following importance weight for ith joint. Finally, the importance weight of the position-based vision control algorithm is given as

$$\omega_{po} = \max_q \{\omega_t^q\}_{q=1}^N. \tag{16}$$

A plot of the importance weights of position-based basic controller with respect to the distance to the image border is illustrated in Fig. 4(a).

3.3.2 Weights for image-based vision control

The importance weight of the IBVS is proportional to the weakness of the performance of position-based visual servoing. The performance of the position-based vision control is measured by the ability to keep 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 ith point to the nearest image border. Let the parameter $\{d_t^i\}_{i=1}^N$ be the distance of the ith point to the nearest image border at time t, where

$$d^i = \min\{u^i - u_{min}, v^i - v_{min}, u_{max} - u^i, v_{max} - v^i\}, \tag{17}$$

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:

$$\omega_t^i = \frac{1}{\sqrt{2\pi}\sigma_d} \exp\left[-\frac{d^{i(2)}}{2\sigma_d^2}\right]. \tag{18}$$

Here, σ_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 the ith image point. Finally, the importance weight of the image-based vision control algorithm is given as

$$\omega_{im} = \max_i \{\omega_t^i\}_{i=1}^N. \tag{19}$$

A plot of the importance weights of image-based basic controller with respect to the distance to the image border is illustrated in Fig. 4(b).

3.4 Stability Analysis of the Hybrid visual control law

Let us recall the integration rule given in 12. It can be rewritten as

$$V = \omega V_{im} + (1-\omega)V_{po} = -(\rho\omega J_{im}^+ e_{im} + \rho(1-\omega)J_{po}^+ e_{po}), \tag{20}$$

$$V = -\rho \begin{bmatrix} \omega J_{im}^+ & 0 \\ 0 & (1-\omega)J_{po}^+ \end{bmatrix} \begin{bmatrix} e_{im}(P) \\ e_{po}(P) \end{bmatrix} = -\rho \begin{bmatrix} J_{im}^+ & 0 \\ 0 & J_{po}^+ \end{bmatrix} \begin{bmatrix} \Omega & 0 \\ 0 & I-\Omega \end{bmatrix} \begin{bmatrix} e_{im}(P) \\ e_{po}(P) \end{bmatrix} \tag{21}$$

$$V = -\rho W \begin{bmatrix} J_{im}^+ & 0 \\ 0 & J_{po}^+ \end{bmatrix} W \begin{bmatrix} e_{im}(P) \\ e_{po}(P) \end{bmatrix} = -\rho J_\omega^+ e_\omega \tag{22}$$

Here, the matrix $W = \begin{bmatrix} (\Omega)^{1/2} & 0 \\ 0 & (I-\Omega)^{1/2} \end{bmatrix}$, the pseudo inverse $J_\omega^+ = WJ^+ = W \begin{bmatrix} J_{im}^+ & 0 \\ 0 & J_{po}^+ \end{bmatrix}$, and the error $e_\omega = W \begin{bmatrix} e_{im}(P) \\ e_{po}(P) \end{bmatrix}$. Substituting in general state equation of the visual servoing system $\dot{e}_\omega = J_\omega V$ we get the following equation:

$$\dot{e}_\omega = -\rho J_\omega \widehat{J_\omega^+} e_\omega. \tag{23}$$

Here, we consider the fundamental issues related to the stability of visual servoing control law. 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

$$\mathcal{L} = \frac{1}{2} e_\omega^T e_\omega. \tag{24}$$

The first order derivative of this function is

$$\frac{\partial \mathcal{L}}{\partial t} = e_\omega^T \frac{\partial e_\omega}{\partial t} = e_\omega^T \dot{e}_\omega, \tag{25}$$

$$\frac{\partial \mathcal{L}}{\partial t} = -\rho e_\omega^T J_\omega \widehat{J_\omega^+} e_\omega$$

From the automatic control theory [20, 21], 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:

$$J_\omega \widehat{J_\omega^+} > 0. \tag{26}$$

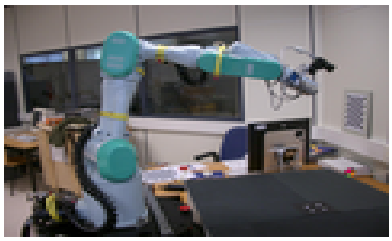


Figure 5. External view of the experimental setup.

In general, the number of the features is $(2N+6)$, where N is the number of image points. The number $(2N+6)$ is greater than the number of degrees of freedom of the robot. The matrices J_ω and $\widehat{J_\omega^+}$ are of full rank 6, and the condition Equation (26) is ensured. This is subject to good approximation of the matrix $\widehat{J_\omega^+}$. Let us note that all the elements of the matrix W are positive since they are computed from the Gaussian functions. Indeed, the condition (26) is still held even the matrix W is time variant. In addition, the elements of W are assumed to vary smoothly, hence its time derivative can be considered null. This is a common assumption in visual servoing literature [22, 23, 24]. One more thing to note is that the configuration may reach a potential local minima when $We_\omega \in J_\omega^T$. This case of potential local minima is related to image features used in the error vector. It is described in [7].

4 Experimental Evaluation

4.1 The Experimental Setup

Our experimental setup consists of a Mitsubishi PA-10 robot arm, with a Point Grey Flea camera mounted on the end-effector. The camera delivers 60 fps at VGA resolution, and it is connected through a Firewire port to the computer which controls the arm. Camera intrinsic parameters are coarsely calibrated. Though the arm has 7 DOF, only 6 of them are controlled via Cartesian velocity commands in the end-effector frame. Fig. 5. shows the external view of this experimental setup. The target used in our experiments is made of 4 white points in a 15 cm square, which is tracked with the ViSP software [25]. Both 2D and 3D visual servoing tasks are defined, and the proportional control gain is set to 0.2. In the hybrid algorithm, we set a threshold $\sigma_d = 25.0$ pixels to the image border. Similar threshold $\sigma_q = 30^\circ$ is used for the joint parameters.

4.2 Results from Positioning Tasks

Most of the robot vision algorithms work well for those task that involved simple motion. They may fail to perform specific challenging tasks like those stated in [13, 14]. We carried out the experiments for three of these challenging tasks. The first task is rotation of 90° about the camera axis, the second one is rotation of 180° about the same axis.

4.2.1 Rotation of 90° about the camera optical axis

This task consists of a 90° rotation around the camera optical axis. Such a task is troublesome for the classical 2D image-based and 3D position-based methods [7]. In 3D

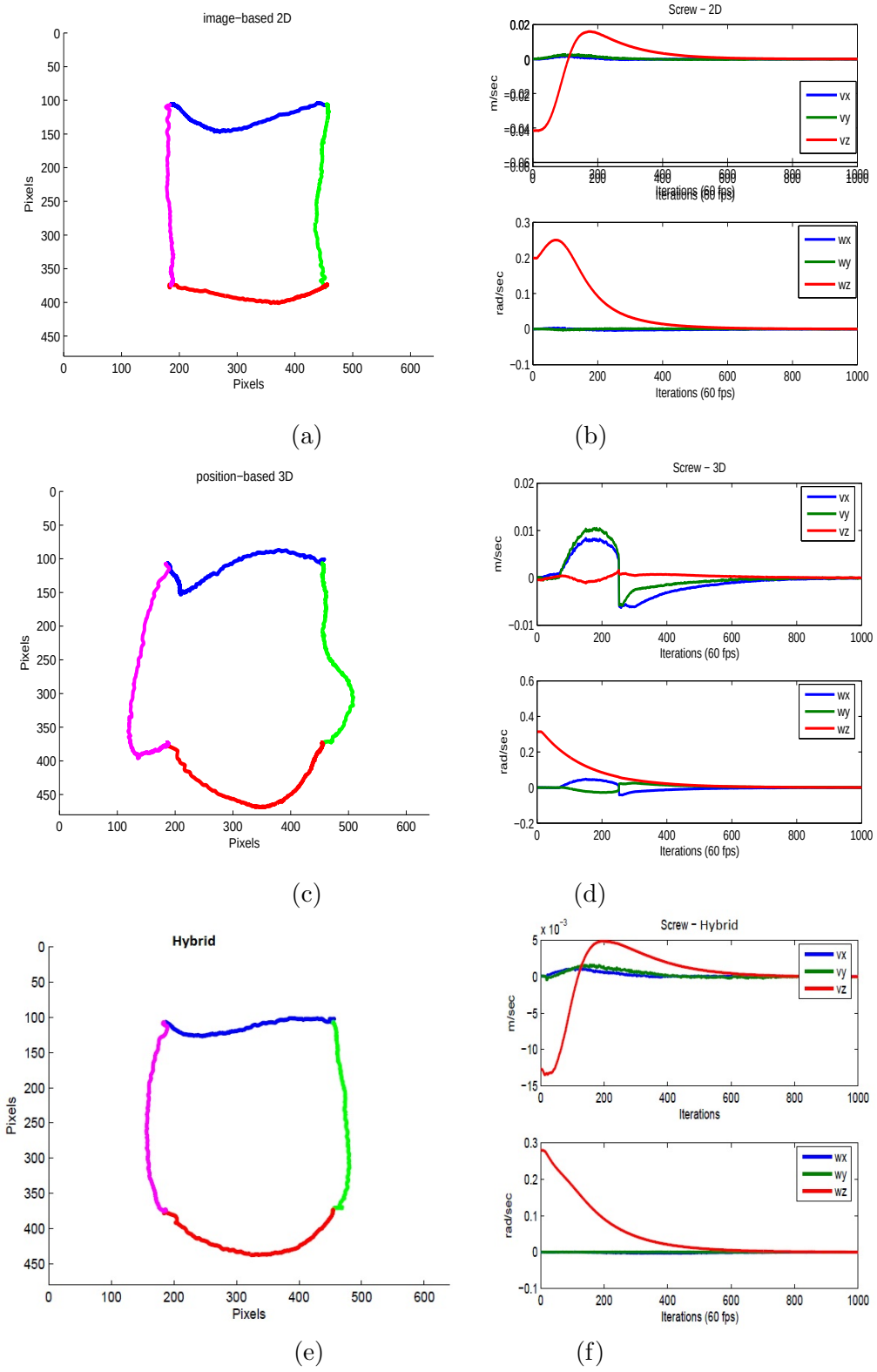


Figure 6. Trajectories in the image space in (a,c,e) and the screw velocity in (b,d,f) of the 2D image-based, 3D position-based, and hybrid vision control algorithm. The task is a 90° about the camera optical axis.

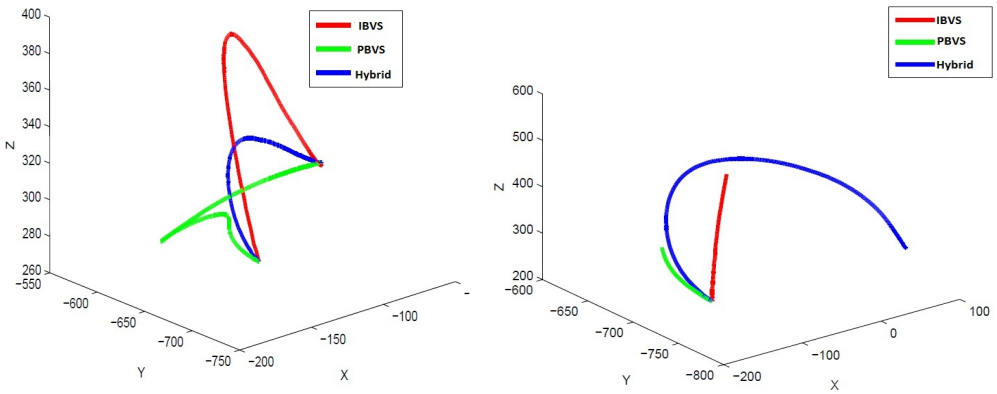


Figure 7. The end-effector trajectories using image-based in red color, position-based in green color, and hybrid in blue color. The task is a 90° about the camera optical axis in (a) and 180° about the camera optical axis in (b).

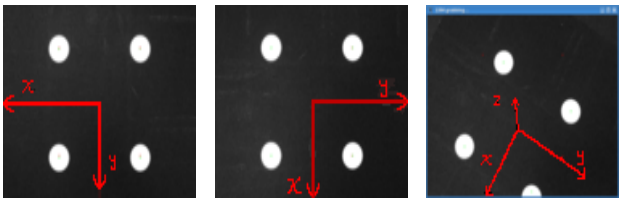


Figure 8. The initial (a) and desired (b) images of the first task: 90 degrees rotation about optical camera axis. In (c) the object target as seen by arm in the middle of the task using position-based method. The feature has partially got out of the field of view. The object frames are drown in red color.

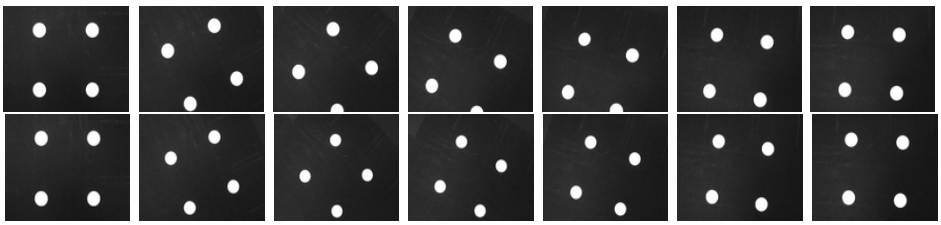


Figure 9. Sequence of images show the image feature trajectories for position-based (top) and hybrid (down) algorithm. The task is a 90° about the camera optical axis. One can not the that one point has partially got out of the field of view in the position-based method (top).

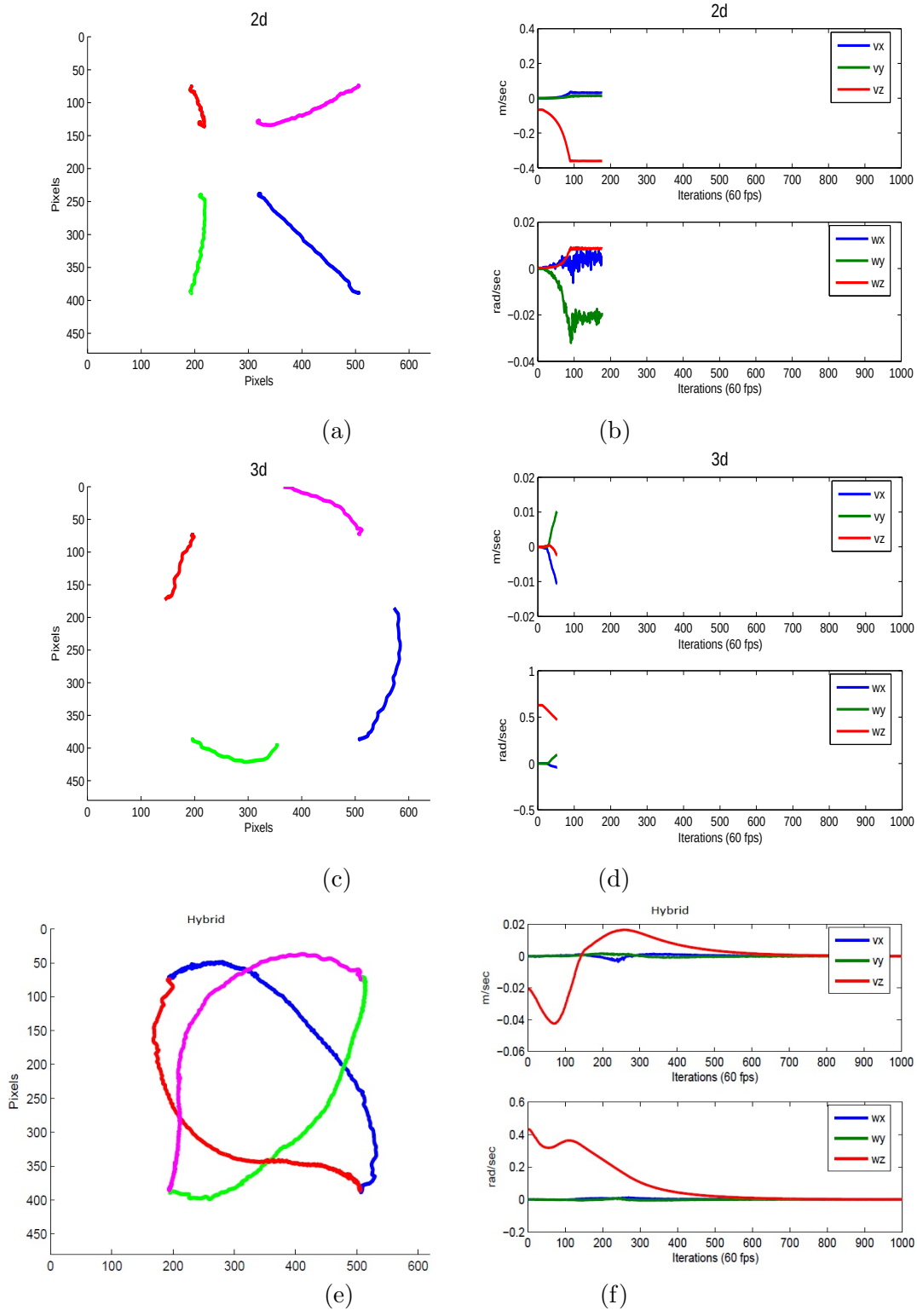


Figure 10. Trajectories in the image space in (a,c,e) and the screw velocity in (b,d,f) of the 2D image-based, 3D position-based, and hybrid vision control algorithm. The task is a 180° about the camera optical axis.

position-based method, the image features can get out of the camera view easily, while using 2D image-based method the camera retreats backward along the camera Z axis trying to perform a non-realizable motion by robot arm.

In the first case *i.e.* using image-based vision control, Fig. 6 (a) shows the image trajectories using image-based (2D) algorithm. The straight line trajectories cause the camera to retreat backward along its Z axis. The trajectory of the end-effector frame is illustrated in Fig. 7. The screw velocity in Fig. 6 (b) also shows a considerable backward translational motion along the Z axis.

In the second case, *i.e.,* position-based (3D) control, the camera trajectory is pure rotation about its optical axis. Thus, the end effector describes a part of circle while the image points describe a 90° arcs around the principal point of the image. Unfortunately, these arcs may get out of the camera field of view (FoV). The image trajectory of this case is shown in Fig. 6 (c) while the trajectory of the end-effector is shown in Fig. 7 (a), the screw velocity is shown in Fig. 6 (d).

In Fig. 8 (c), there is an image of the target object in the middle of the task using 3D position-based algorithm. One feature gets partially out of the camera FoV. The tracking continues, while the centroid of the circle representing the feature moves away from its correct position. Then the model of the square is wrongly fitted. Indeed, the 3D pose is wrong. As it is shown the red color frame That explains the strange motion of the camera frame in the middle of the task using 3D or position-based algorithm in Fig. 9.

According to the red color frame, the camera is normal to the surface, thus the Z axis should go inside the image, but it departs quite significantly. That explains the strange motion of the camera frame in the middle of the task using 3D or position-based algorithm in Figs. 6 (d) and 7 (a).

The proposed method combines both schemes. It seamlessly achieves the task while keeping the visual features in the field of view, see Fig. 6 (e). The performance in the image is similar to the performance of the case of using 2D algorithm. The end-effector is moving smoothly to the goal location, see Fig. 7 (a). End-effector trajectory is satisfactory. It also keeps the arm in its workspace.

To get better clarity about the performance of the proposed method, reader may compare the translation motion in Figs. 6 (b,d,f). One can note the considerable backward motion along Z axis using image-based (2D) algorithm, the velocity reaches 4 cm/sec. This is shown in Fig. 6 (b). In contrast, the backward motion using the 3D algorithm is up to 1 cm/sec, see Fig. 6 (d) and 7 (a), this backward motion is due to the partial absence of one feature near to the image border. Using the proposed algorithm, the backward motion is very small and not more than 1.3 cm/sec, see Fig. 6 (f). Fig. 9 shows two sequences of images as seen by the arm using each of the position-based and hybrid proposed algorithm. The top one is for the position-based (3D) algorithm. One of the features has partially got out of the field of view. The sequence at the bottom is using the hybrid algorithm. The features have been successfully kept in the field of view.

4.2.2 Rotation of 180° about the camera optical axis

This task is more troublesome for the classical image-based and position-based methods. In image-based, since the rotation is 180°, the camera retreats back to infinity. The robot arm obviously gets out its work space. As illustrated in Fig. 10 (a,b), the process stopped after around 200 iterations. In position-based, one feature, that is near to the image border, gets out of the camera field of view. The process stopped after approximately 80 iterations

as it illustrated in Fig. 10 (c,d). As it is shown in Fig. 10 (e,f), the process completed successfully using the proposed hybrid control law. The end-effector trajectories in the Cartesian space using of image-based, position-based, and hybrid algorithms are illustrated in Fig. 7 (b).

5 Conclusions

An integration framework is proposed to improve the performance of IBVS and PBVS algorithms. The constraints of the features' visibility in the image as well as the joint limits of the arm are used to define the importance functions. It is shown that the hybrid method is superior to these individual algorithms. Experiments are carried out to demonstrate the significant improvement in performance of the classical vision-based control algorithms in both image and Cartesian (joint) space.

References

[1] S. Hutchinson, G. Hager, and Cork, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct 1996.

[2] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, December 2006.

[3] ——, "Visual servo control, part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, March 2007.

[4] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, New Orleans, USA, April 2004, pp. 1843–1848.

[5] A. Dame and E. Marchand, "Mutual information-based visual servoing," *IEEE Trans. on Robotics*, vol. 27, no. 5, pp. 958–969, October 2011.

[6] B. Tamadazte, N. Lefort-Piat, and E. Marchand, "A direct visual servoing scheme for automatic nanopositioning," *IEEE/ASME Trans. on Mechatronics*, vol. 17, no. 4, pp. 728–736, August 2012.

[7] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, D. Kriegman, G. . Hager, and A. Morse, Eds. LNCIS Series, No 237, Springer-Verlag, 1998, pp. 66–78.

[8] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, "Keeping features in the field of view in eye-in-hand visual servoing: A switching approach," *IEEE Trans. on Robotics*, vol. 20, no. 5, pp. 534–549, Oct 2004.

[9] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, Aug 2002.

[10] V. Kyrki, D. Kragic, and H. I. Christensen, "New shortest-path approches to visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, Sendai, Japan, September-October 2004, pp. 349–354.

[11] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, April 1999.

[12] O. Kermorgant and F. Chaumette, "Combining ibvs and pbvs to ensure the visibility constraint," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'11*, 2011, pp. 2849–2854.

[13] N. Gans and S. A. Hutchinson, "An asymptotically stable switched system visual controller for eye in hand robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'03*, Las Vegas, October 2003, pp. 3061–3068.

[14] ——, "An experimental study of hybrid switched system approach to visual servoing," in *IEEE Int. Conf. on Robotics and Automation, ICRA'03*, Taiwan, September 2003, pp. 3061–3068.

[15] E. Cervera, A. P. D. Pobil, F. Berry, and P. Martinet, "Improving image-based visual servoing with three-dimensional features." *Int. Journal of Robotic Research*, vol. 22, no. 10-11, pp. 821–840, 2003.

[16] A. H. Abdul Hafez and C. V. Jawahar, "Probabilistic integration framework for improved visual servoing in image and Cartesian spaces," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'06*, Bejing, China, October 2006.

[17] P. Corck and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 507–515, Aug 2001.

[18] O. Khatib, "Real time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[19] A. H. Abdul Hafez, E. Cervera, and C. V. Jawahar, "Optimizing image and camera trajectory using on-line boosting," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'07*, San Diego, CA, October 2007.

[20] A. Isidori, *Nonlinear control systems*, 3rd ed. New York: Springer-Verlag, 1995.

[21] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[22] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Trans. on Robotics and Automation*, vol. 21, no. 6, December 2005.

[23] N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *IEEE Trans. on Automatic Control*, vol. 54, no. 11, pp. 2493–2505, November 2009.

[24] C. C. Cheah, D. Q. Wang, and Y. C. Sun, "Region-reaching control of robots," *Transaction on Robotics.*, vol. 23, no. 6, pp. 1260–1264, Dec. 2007.

[25] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, December 2005.