Hybrid Visual Servoing by Boosting IBVS and PBVS

A.H. Aabdul Hafez*, Enric Cervera†, and C.V. Jawahar‡
*Automatic Control Lab. College of Electrical and Electronic Engineering,
University of Aleppo, Syria. Email: hafez@research.iiit.ac.in
†Robotic Intelligence Lab. Jaume-I University of Castello,
Spain, Email: enric.cervera@icc.uji.es
‡Center for Visual Information Technology, International Institute of Information Technology,
Hyderabad, India, Email: jawahar@iiit.ac.in

***Abstract*—In this paper, we present a novel boosted robot vision control algorithm. This method utilizes on-line boosting to produce a *strong* vision-based robot control starting from two *weak* algorithms. These weak methods are image-based and position-based visual servoing algorithms. The notion of weak and strong algorithms have been presented in the context of robot vision control. Appropriate error functions are defined for the weak algorithms to evaluate their suitability in the task. The integrated algorithm has superior performance both in image and Cartesian spaces. Experiments validate this claim.**

I. 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 visual information. Visual information can be either from the 2D image space or from the 3D Cartesian pose space [1]. A typical visual servoing algorithm minimizes an error function between the current pose (position and direction) of the camera and the desired one [2]. Many visual servoing algorithms have been proposed in the literature [3], [4], [5]. They differ based on the objective function used in the minimization process. Basically, we have 2D/image-based (IBVS) and 3D/position-based (PBVS) visual servoing algorithms. Chaumette [1] has shown that each of these two basic classes of algorithms has weak points (drawbacks) or potential problems.

The weakness of each algorithm is measured either in the image space such the ability to keep the features visible during the servoing process [6], or in the Cartesian space such the ability to keep the arm in its Cartesian/joint space [7]. This is in addition to the local minima avoidance (convergence problem) [8], [9]. The image trajectory of image-based visual servoing algorithm is a straight line. This ensures the visibility of image features during the whole servoing process using this algorithm. However, the Cartesian trajectory is a curvature path and robot may get out of its workspace. In contrast, the position-based algorithm keeps the robot in the Cartesian workspace thanks to the straight line camera path produced using this algorithm. Unfortunately, the image features are not ensured to be always visible using this algorithm. Owing to the complementary properties of the weak points in these two algorithms, hybrid methods have been recently proposed to integrate the advantages and discard the drawbacks.

Recently, in addition to providing accurate control signal, research on hybrid visual servoing has focused on addressing issues like feature visibility, local minima aviodance, faster convergence, short camera path, continuous control signal, *etc*. Many of the hybrid methods address the above mentioned issues by integrating the 2D and 3D information in the feature space [8], [9], [10] or in the action space [11], [12], [13], [6].

The $21/2$D visual servoing method [9] decoupled the rotation motion from its translational part considering one visible image point during the servoing process. In this method, the homography matrix that relates image points in the current and desired views is decomposed to extract the required rotation motion. The translation is controlled using the selected visible image point. However, there is no knowledge about the camera path in the Cartesian space. The translation and the z axis rotation are controlled in [8] using the error information, which is extracted from the decomposition of the homography matrix. The rotations about x and y axes are controlled using selected image point, say the origin of the object. Even though, the two mentioned methods described in [8], [9] are globally stable, they are sensitive to noise and measurement errors like all methods based on homography estimation.

An algorithm which switches between image-based and position-based vision control algorithms is presented by Gans and Hutchinson [12], [13]. In this hybrid switching method, the IBVS and PBVS run independently. Based on satisfaction of some constraints, a logical decision system switches between them. Since this method switches between IBVS and PBVS in binary form, local minima may be reached. Another principle of switching is presented by Chesi *et al.* [6]. 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 and Jawahar [11] present a smooth linear combination of different visual servoing algorithms. The combining weights are computed using an error function of the weakness of the concern algorithm. Another hybrid method based on potential fields [7] is pro-

posed 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.

This paper aims at introducing boosting as a machine learning algorithm to produce an enhanced vision-based control algorithm. This open a new direction in research on vision-based robot control and visual servoing. The work here is motivated by the recent success of boosting algorithm in solving computer vision problems. For example, boosting has been used for face recognition [14], visual tracking [15], feature selection [16], *etc*. Most of these use Ada-boost or its variants. Ada-Boost has been carefully analyzed and tested by many researchers [17]. We use boosting for efficient positioning task in active vision systems, and demonstrate on robot vision control problem.

Here, we utilize on-line boosting to enhance the visual control from the two basic (image-based and position-based) visual servoing algorithms. Each of these weak algorithms performs well only over a subspace of the input domain. Boosting allows the design of a strong algorithm with higher performance from multiple weak algorithms. In boosting visual servoing algorithm, two independent weak robot vision control algorithms (IBVS and PBVS) are boosted to obtain a strong algorithm. Since the output of each of these algorithms is velocity control signal, the final output of the boosted algorithm is the boosted control signal derived from the independent IBVS and PBVS algorithms. This has the advantage of running the two simple IBVS and PBVS algorithms instead of running the hybrid visual servoing algorithm that has more complicated form. The problem is reduced to the weighted sum of the output of these two weak algorithms. Weights are computed from the probabilistic error functions that are defined for each one of the involved independent weak algorithms. The error function of the IBVS algorithm is defined in the joint space to prevent the arm from reaching the limits of its joints. The error function of the PBVS algorithm is defined in the image space to prevent the object from getting out of the camera field of view.

The rest of the paper is a background of boosting and visual servoing in the next section. In the third section, a probabilistic frame work for integration of different visual servoing algorithms is presented. Next section is utilizing the boosting algorithm as an implementation of the proposed integration framework. This includes a statement of the over all algorithm and the definition of the error functions (weakness metric) of the two image-based and position-based visual servoing algorithm. Experimental results and analysis are presented in the last section.

II. Background of Boosting and Visual Servoing

Boosting is a general method for improving the performance of a given algorithm by combining multiple weak hypotheses [18]. In other words, boosting transforms a set of weak algorithms into a strong one. The question about whether a weak learning algorithm, which performs just slightly better than random guessing, can be boosted into an arbitrarily accurate strong learning algorithm has been posed for the first time by Kearns and Valiant [19]. However, Schapire was the first to provide a provable polynomial-time boosting algorithm in [20].

A. Weak and strong algorithms

Weak algorithms: Let the function $y = f(x)$ that maps $R^n \longrightarrow R^m$ represents the algorithm that takes x as an input and results the output vector y. A weak vision control algorithm is defined here as follows: For a subset $X \in R^n$ of the function domain, error e is greater than an ϵ. Some algorithms compute the error e using training data as in the case of classifiers or could be computed using the actual and desired output as in the case of control algorithms based on progressive learning.

Strong algorithms: Given a set of N weak algorithms $y_i = f_i(x)$, a strong algorithm is computed as a linear combination of the N weak algorithms $y_i = f_i(x)$. This can be written as

$$Y = F(x) = \sum_{i=1}^{N} \alpha_i f_i(x) = \sum_{i=1}^{N} \alpha_i y_i. \tag{1}$$

A strong algorithm should satisfy the constraint that the error $e \leq \epsilon, \quad \forall x \in R^n$.

Let the function $e_i = p_i(y, x)$ represents the weakness in the performance of the weak algorithm. This error is increased when the performance of the weak algorithm $y_i = f_i(x)$ is less satisfactory. Weight that determines the contribution of the weak algorithm to the strong one is defined as

$$\alpha_i = \frac{1}{2} \ln(\frac{1 - e_i}{e_i}). \tag{2}$$

This way of weight computation reduces the weight value when the error increases and the performance becomes more non-satisfactory, in addition to adding an upper bound to the weight value that is useful to the process convergence.

B. Image-based and position-based visual servoing

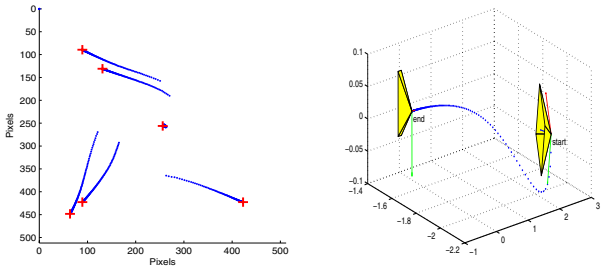
In Image-based visual servoing, the task function is defined with respect to the error $e(s)$ in the image space, where s is the vector of the current features position and s^* is the vector of the desired one. The velocity screw using IBVS is given as [3]

$$\dot{e_i}(s) = J_i V_i \tag{3}$$

$$V_i = -\lambda_i J_i^+ e_i(s), \tag{4}$$

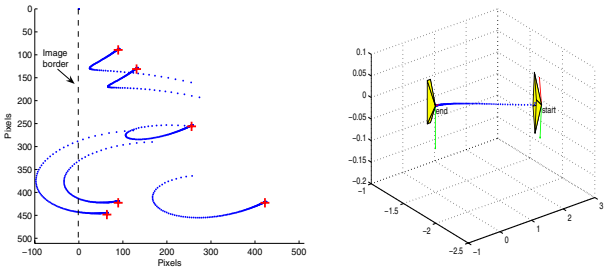
where J_i^+ is the pseudo-inverse of the Jacobian matrix J_i.

It is easy to show 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. Indeed, from any initial state, IBVS moves the image points straight toward its 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



(a) (b)

Fig. 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 +.



(a) (b)

Fig. 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 +.

global stability is not ensured since unpredictable image local minima and Jacobian singularity may occurs at any time.

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 velocity screw using PBVS is given as [3]

$$\dot{e_p}(s) = J_p V_p \tag{5}$$

$$V_p = -\lambda_p J_p^{-1} e_p \tag{6}$$

where J_p is the interaction matrix [21].

While PBVS minimize 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. However, 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.

III. Integration based using On-line Boosting

The visual data can be 2D visual data from the image space or/and 3D visual data from the Cartesian space. Using only one kind of data (2D or 3D) leads to the two traditional image-based or position-based vision control respectively. Boosted visual control algorithm consider 2D and 3D algorithms as weak algorithms. A linear combination of these two weak algorithms produce a strong algorithm with satisfactory performance. Weights which are used in the combination are computed based on error function associated with each one of these weak algorithms.

A. The overall boosting-based algorithm

The task is a robot arm positioning task with respect to a set of features or with respect to an object that contains a set of features. Equations (4) and (6) present two weak vision control algorithms that can be used to perform the said positioning task. The former has an undesired behavior in the Cartesian space while the later has an undesired behavior in the image space. On-line boosting produces a strong vision-based algorithm using the error functions defined in correspondence to each algorithm.

The general structure of the algorithm was explained in Sec II. For each of image-based vision control and position-based vision control algorithms, the error functions explained in Sections III-B and given by Eq. (12) and Eq. (15) are evaluated and used to compute the corresponding weights α_i that gives the current importance to each algorithm. The weights α_i are normalized to sum up to one

$$\bar{\alpha_i} = \left(\frac{\alpha_i}{\sum \alpha_i}\right) |_{i \in \{IBVS, PBVS\}} .$$

So, the output of the strong algorithm given in Eq. (1) can be written as

$$V = F(x) = \sum_{i \in \{IBVS, PBVS\}} \bar{\alpha_i} V_i(X). \tag{7}$$

or

$$V = \omega V_{im} + (1 - \omega) V_{po}, \tag{8}$$

where, $\bar{\alpha}_{IBVS} = \omega$ and $\bar{\alpha}_{PBVS} = 1 - \omega$.

Algorithm 1 describes details of the on-line boosted vision-based control algorithm.

B. Error function for image-based vision control

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 joints configuration $\mathbf{q}$ of a robot arm is acceptable when

$$\forall i, q^i \in \left[q^i_{min} + \theta^i_q, q^i_{max} - \theta^i_q\right]. \tag{9}$$

Here, θ^i_q is a threshold of the ith joint. The error in the performance of image-based control algorithm can be measured as a function of 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

$$r^i = \min\{q^i - q^i_{min} - \theta^i_q, q^i_{max} - \theta^i_q - q^i\} \tag{10}$$

and N is the number of the joints of the arm.

The error function of image-based visual servoing is as

$$e^q_t = \frac{1}{\sqrt{2\pi}(\sigma_q)} \exp\left[-\frac{r^2}{2\sigma^2_q}\right]. \tag{11}$$

Algorithm 1 Details of on-line boosting algorithm for improving the image-based vision control and position-based vision control algorithms

1: Input: A list of input data: The pose P_t at time t, image measurements X_t where $X = \{x_m : m \in \{1, \ldots, M\}\}$, and a set of the two IBVS and PBVS weak algorithms $V_j = f_j(X)$, where $j \in \{IBVS, PBVS\}\}$
2: Output: A strong algorithm $V = F(X)$.
3: **for** till convergence **do**
4: **for** $j \in \{IBVS, PBVS\}$ **do**
5: Compute the output of the weak vision-based control algorithm $V_j = f_j(X)$. As in Eqs. (4) and (6)
6: Compute the error function for IBVS weak algorithm $e_{t(ibvs)} = \sum_{i=1}^{N_q} e_t^i$, where e_t^i is computed as in Eq. (11)
7: Compute the error function for PBVS weak algorithm $e_{t(pbvs)} = \sum_{i=1}^{N_d} e_t^i$, where e_t^i is computed as in Eq. (14)
8: Compute the weights $\left\{\alpha_j = \frac{1}{2}\ln(\frac{1-e_j}{e_j})\right\}_{j\in\{IBVS,PBVS\}}$
9: **end for**
10: Normalize the weights α_j to $\bar{\alpha_j}$ to sum up to one.
11: Compute the output of the strong algorithm $V = F(x) = \sum_{j\in\{IBVS,PBVS\}} \bar{\alpha_j} V_j(X)$
12: **end for**

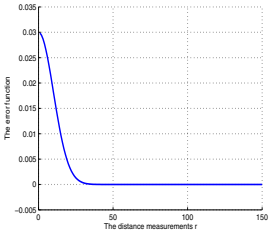


Fig. 3. (b) The error function of image-based control algorithm.

Here, σ_q is selected in such that only working points within a minimum distance to its joint limits will contribute to the error function.

The total error function of the image-based vision control algorithm is given as

$$e_{t(ibvs)} = \sum_{q=1}^{N} e_t^q. \tag{12}$$

This is the error function of the performance of the weak image-based vision control algorithm. A plot of the error with respect to the distance to the joint limit is illustrated in Fig. 3. Substituting in Eq. (2), we get the associated weight to the image-based vision control algorithm.

C. Error function for position-based vision control

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 ith point to the nearest image border or a threshold θ_I of the 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

$$d^i = \min\{u^i - u_{min}, v^i - v_{min}, u_{max} - u^i, v_{max} - v^i\}, \tag{13}$$

and N is the number of image points.

The error function of the position-based vision control algorithm with respect to one image point is given as

$$e_t^i = \frac{1}{\sqrt{2\pi}(\sigma_i)} \exp\left[-\frac{d^2}{2\sigma_i^2}\right]. \tag{14}$$

Here, σ_i is selected in such that only image points within a minimum distance to the image border will contribute to the error function.

The total error function of position-based algorithm is given as

$$e_{t(pbvs)} = \sum_{i=1}^{N} e_t^i. \tag{15}$$

This is the error function of the performance of the weak position-based vision control algorithm. A plot of the error with respect to the distance to the image border is illustrated in Fig. 4. Substituting in Eq. (2), we get the associated weight to the position-based algorithm.

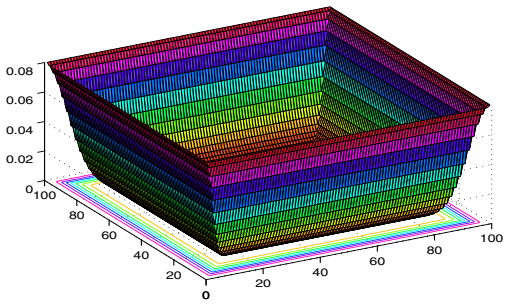


Fig. 4. The error function of the position-based vision control algorithm.

Fig. 5. External view of the experimental setup.

IV. Experimental Evaluation

A. 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

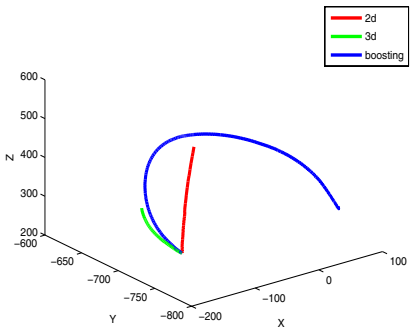


Fig. 6. The end-effector trajectories using image-based in red color, position-based in green color, and boosting-based in blue color. The task is a 180° about the camera optical axis.

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 [22]. Both 2D and 3D visual servoing tasks are defined, and the proportional control gain is set to 0.2. In the boosting algorithm, we set a threshold $\theta_I = 25.0$ pixels to the image border. Similar threshold $\theta_q = 30°$ is used for the joint parameters. The values of the parameters σ_i and σ_q are set to 50 pixels and 60 degrees respectively.

B. Results from Positioning Tasks

The most of robot vision algorithms work well for those task that involved simple motion. They have been developed in response to specific task but they fail to perform some another specific challenging tasks. Some of these challenging tasks are stated in [23]. 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, and the last one is features rotation about an axis in the object plane involved with general motion.

1) 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. 7 (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. 7 (c,d). As it is shown in Fig. 7 (e,f), the process completed successfully using the boosting-based control law.

The end-effector trajectories in the Cartesian space using of image-based, position-based, and boosting-based algorithms are illustrated in Fig. 6 (b). Fig. 8 shows the error functions

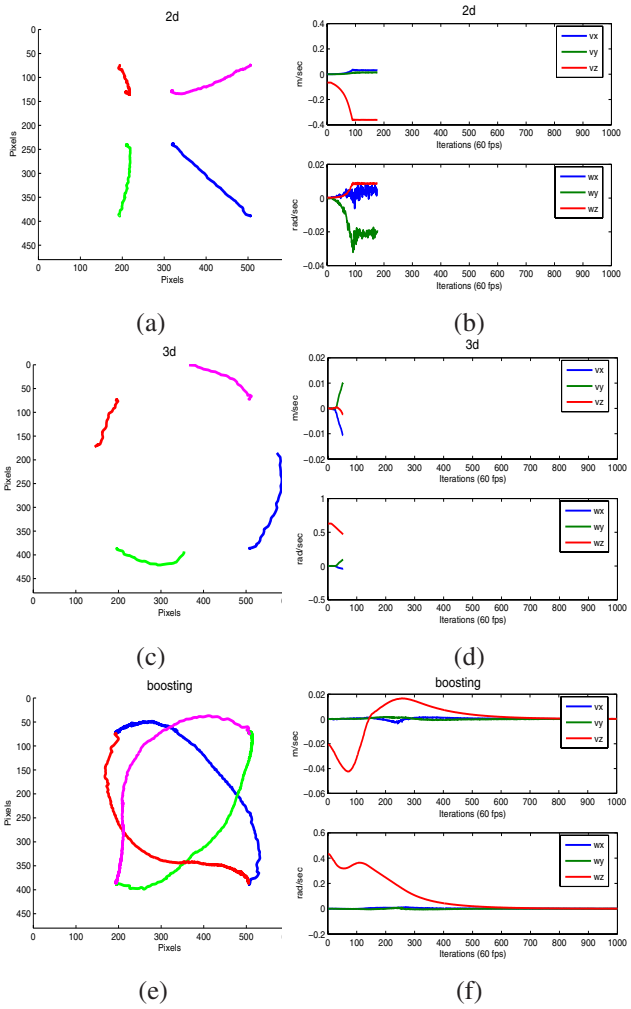


Fig. 7. Trajectories in the image space in (a,c,e) and the screw velocity in (b,d,f) of the 2D image-based, 3D position-based, and boosting-based vision control algorithm. The task is a 180° about the camera optical axis.

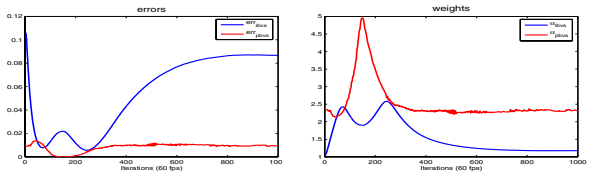


Fig. 8. The error functions of image-based and position-based weak algorithms in (a), and the corresponding weights in (b). The task is a 180° about the camera optical axis.

of each of the image-based and position-based algorithms as a boosted weak algorithms in Fig. 8 (a). the corresponding weights in Fig. 8 (b).

2) Feature point rotation/general motion: This task is to rotate the 3D feature points about an axis perpendicular to the camera optical axis and lying in the feature plane. Since the system needs to perform significant rotation and translation to nullify the task function, this task considered as one of the most strenuous tasks. More details about this task can be found in [23]. The image-based control algorithm is known

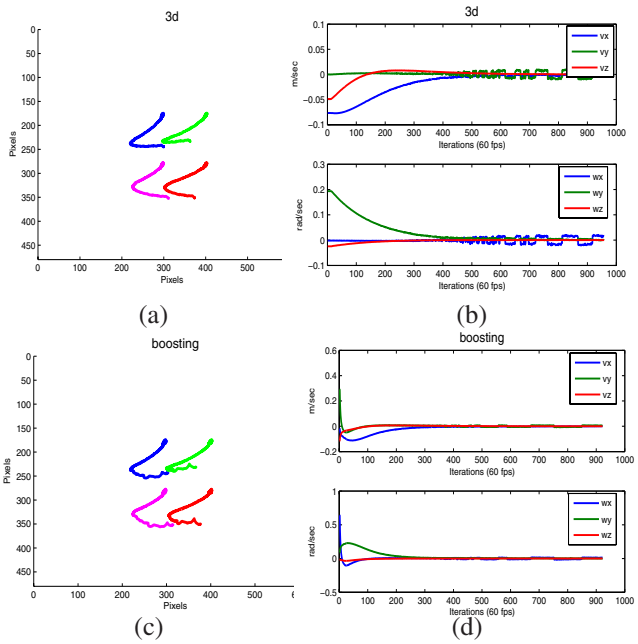


Fig. 9. Trajectories in the image space in (a) and the screw velocity in (b) of the 3D position-based vision control algorithm. The task is feature point rotation/general motion.

to preform poorly to this task [23]. In our experiments the process failed after a few iterations. Both position-based and boosting-based perform well toward this task and better than image-based. There is an advantage of boosting over position-based at the convergence phase: oscillations are smaller, due to the integration and merging with image-based. By comparing between the 3D and boosting-based kinematic screws in Figs. (9 (b,d), one could say that, in boosting, the oscillations at convergence are lowered by more than 20%.

V. Conclusion and Future Work

The on-line boosting algorithm has been used to improve the performance of each one of these two algorithms. It deals with them as weak algorithms associated with error functions that describe the weakness in the performance of the algorithm. These weak algorithms can be linearly combined to produce a strong algorithm with satisfactory performance. The constraints of the features' visibility in the image and the joint limits of the arm are used to define the error functions and compute the suitable weights.

It is shown that boosting algorithms can be generalized to much more than classifications and recognition. It can be used for vision control application. In the same time, vision control algorithms are shown to be significantly improved in both, the performance and implementation aspects. Experiments has been curried out and showed a significant improvement to the performance of the classical vision-based control algorithms in both image and Cartesian (joint) space.

Acknowledgment

The UJI Robotic Intelligence Lab is partially funded by the Spanish Ministry of Science and Education (MEC) under grant DPI2005-08203-C02-01.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] ——, "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.

[14] P. Yang, S. Shan, W. Gao, S. Li, and D. Zhang, "Face recognition using Ada-boosted gabor features," in *Automatic Face and Gesture Recognition*, 2004, pp. 356–361.

[15] S. Avidan, "Ensemble tracking," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'05*, vol. 2, 2005, pp. 494–501.

[16] H. Grabner and H. Bischof, "On-line boosting and vision," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'06*, 2006.

[17] R. Schapire, "The boosting approach to the machine learning: An overview," in *MSRI Workshope on Nonlinear Estimation and Clasification*, 2001.

[18] ——, "A breif introduction to boosting," in *International Joint Conference on Artificial Intelligence, IJCAI'99*, 1999.

[19] K. Michael, G. V. Leslie, and L. Valiant, "Cryptographic limitations on learning boolean formulae and finite automata," *Journal of the Association for Computing Machinery*, vol. 41, no. 1, p. 6795, 1994.

[20] R. Schapire, "The strength of weak learnability," in *Machine Learning*, vol. 5, 1990, pp. 197–227.

[21] W. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative endeffector control using Cartesian position-based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, October 1996.

[22] 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.

[23] N. Gans, S. Hutchinson, and P. Corcke, "Performance test for visual servo control systems, with application to partitioned approaches to visual servo control," *International Journal of Robotics Research*, vol. 22, no. 10-11, pp. 955–981, October-November 2003.