

# Uncalibrated image-based visual servoing

Àngel Santamaria-Navarro and Juan Andrade-Cetto

**Abstract**—This paper develops a new method for uncalibrated image-based visual servoing. In contrast to traditional image-based visual servo, the proposed solution does not require a known value of camera focal length for the computation of the image Jacobian. Instead, it is estimated at run time from the observation of the tracked target. The technique is shown to outperform classical visual servoing schemes in situations with noisy calibration parameters and for unexpected changes in the camera zoom. The method’s performance is demonstrated both in simulation experiments and in a ROS implementation of a quadrotor servoing task. The developed solution is tightly integrated with ROS and is made available as part of the IRI ROS stack.

## I. INTRODUCTION

Visual servo control is the problem of using computer vision data in the servo loop to control a robot. We are concerned with image-based visual servo, in which a set of image features have to reach desired image locations. This set of features usually correspond to the projected image coordinates of several 3D points on the tracked target. For a stable visual servo control law, the difference between the actual location of the observed features and their desired locations should decrease asymptotically.

The classical approach to image based visual servo requires a priori knowledge of the camera calibration parameters, basically to estimate the depth of each feature point in the visual data. There are situations however in which camera parameters are either noisy or unavailable. This situation might negatively influence the convergence of the error function along the directions corrupted by such noise.

In this paper we develop a strategy to image-based visual servo with mild assumptions about principal point and skew, and that does not require a priori knowledge of camera focal length. Instead, the focal length is re-expressed in terms of the solution of a linear system on the coordinates of a set of control points in the target reference frame, and is estimated at each iteration of the control loop.

Our approach is compared with the classical approach to image-based visual servo, both in Matlab simulations and inside a ROS quadrotor stack. The technique is used to stabilize an UAV during the landing approach maneuver.

This paper is structured as follows. In the next section a brief description about the state of the art on visual servoing, and more specifically on image-based servo is given. The

problem background formulation is explained in Section 3; and in Section 4, our new uncalibrated image Jacobian is developed and the control law is described. Section 5 contains simulations and experimental results. Conclusions are given in Section 6.

## II. RELATED WORK

Vision-based robot control systems are usually classified in three groups: position-based visual servo, image-based visual servo, and hybrid control systems [1], [2]. In position-based visual servo, the geometric model of the target is used in conjunction with visual features extracted from the image to estimate the pose of the target with respect to the camera frame. The control law is designed to reduce such pose error. For this reason, the approach is also referred as 3D visual servoing [3]. Minimizing error in pose has the disadvantage that features could easily be lost from the image during the servo loop. In image-based visual servoing on the other hand, the control law is defined directly in the image plane, minimizing the error between observed and desired image feature coordinates [4], [5]. There exist however stability and convergence problems that may occur, either because the image Jacobian becomes singular during the servoing, or because the controller falls in a local minima at points with unrealizable image motion [6].

This situation can be palliated to some extent with the use of hybrid approaches, which entail some combination of both groups of algorithms. The 2-1/2-D hybrid visual servoing scheme [7] estimates partial camera displacement at each iteration of the control law and minimizes a functional of both, the error measures in image space typical from image-based servo and a log depth ratio accounting for the rate at which the camera moves to the target. Another hybrid approach is the partitioned visual servo scheme [8], which is based on adding to the traditional image-based error function a term decoupling the motion and rotation along the  $z$  axis. To this end two new image features are introduced, one of them related to the area of the polygon being tracked.

In all image-based and hybrid approaches however, the resulting image Jacobian or interaction matrix, which relates the camera velocity with the image feature velocities, depends on a priori knowledge of intrinsic camera parameters. To do away with this dependence, one could optimize for the parameters in the image Jacobian while error in the image plane is being minimized. This is done for instance, using Gauss-Newton to minimize squared image error and non-linear least squares optimization for the image Jacobian [9], [10]; using weighted recursive least squares (RLS), not to obtain the true parameters, but instead an approximation that

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness under project PAU+ DPI2011-27510 and by the EU project ARCAS FP7-287617.

The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, Barcelona 08028, Spain, [asantamaria](mailto:asantamaria), [cetto@iri.upc.edu](mailto:cetto@iri.upc.edu).

still guarantees asymptotic stability of the control law in the sense of Lyapunov [11]; or using k-nearest neighbor regression to store previously estimated local models or previous movements, and estimating the Jacobian using local least squares (LLS) [12]. To provide robustness to outliers in the computation of the Jacobian, [13] proposes the use of an M-estimator.

This paper presents a new approach to image-based visual servo in which the computation of the image Jacobian makes mild assumptions about the camera parameters. In particular, it assumes squared pixel sizes, centered principal point, and unknown focal length. Independence of focal length makes the system robust to noise and to unexpected large variations of this parameter.

### III. BACKGROUND

Drawing inspiration on the EPnP [14] and UPnP [15] algorithms, we can formulate the focal length in terms of the relation between the camera and target frames. To this end we set a reference system attached to the target object, and define a set of four control points as a basis for this reference system. Then, one can express the 3D coordinates of each target feature as a weighted sum of the elements of this basis. Computing the pose of the object with respect to the camera resorts to computing the location of these control points with respect to the camera frame. A least squares solution for the control point coordinates albeit scale, is given by the null eigenvector of a linear system made up of all 2D to 3D perspective projection relations between the target points. Given the fact that distances between control points must be preserved, these distance constraints can be used in a second least squares computation to solve for scale and focal length.

More explicitly, the perspective projection equations for each target feature become

$$\sum_{j=1}^4 \left( a_{ij} x_j + a_{ij} (u_0 - u_i) \frac{z_j}{\alpha} \right) = 0, \quad (1)$$

$$\sum_{j=1}^4 \left( a_{ij} y_j + a_{ij} (v_0 - v_i) \frac{z_j}{\alpha} \right) = 0, \quad (2)$$

where  $\mathbf{s}_i = [u_i, v_i]^T$  are the image coordinates of the target feature, and  $\mathbf{c}_j = [x_j, y_j, z_j]^T$  are the 3D coordinates of the  $j$ -th control point in the camera frame. The terms  $a_{ij}$  are the barycentric coordinates of the  $i$ -th target feature which are constant regardless of the location of the camera reference frame, and  $\alpha$  is our unknown focal length.

These equations can be jointly expressed for all 2D-3D correspondences as a linear system

$$\mathbf{M}\mathbf{x} = \mathbf{0}, \quad (3)$$

where  $\mathbf{M}$  is a  $2n \times 12$  matrix made of the coefficients  $a_{ij}$ , the 2D points  $\mathbf{s}_i$  and the principal point; and  $\mathbf{x}$  is our vector of 12 unknowns containing both the 3D coordinates of the control points in the camera reference frame and the camera focal length, dividing the  $z$  terms:

$$\mathbf{x} = [x_1, y_1, z_1/\alpha, \dots, x_4, y_4, z_4/\alpha]^T. \quad (4)$$

Its solution lies in the null space of  $\mathbf{M}$ , and can be computed as a scaled product of the null eigenvector of  $\mathbf{M}^T \mathbf{M}$  via Singular Value Decomposition

$$\mathbf{x} = \beta \mathbf{v}, \quad (5)$$

the scale  $\beta$  becoming a new unknown<sup>1</sup>.

To solve for  $\beta$  we add constraints that preserve the distance between control points of the form

$$\|c_j - c_{j'}\|^2 = d_{jj'}^2, \quad (6)$$

where  $d_{jj'}$  is the known distance between control points  $c_j$  and  $c_{j'}$  in the world coordinate system. Substituting  $\mathbf{x}$  in the six distance constraints of Eq. 6, we obtain a system of the form

$$\mathbf{L}\mathbf{b} = \mathbf{d}, \quad (7)$$

where  $\mathbf{b} = [\beta^2, \alpha^2 \beta^2]^T$ ,  $\mathbf{L}$  is a  $6 \times 2$  matrix built from the known elements of  $\mathbf{v}$ , and  $\mathbf{d}$  is the 6-vector of squared distances between the control points. We solve this over-determined linearized system using least squares and estimate the magnitudes of  $\beta$  and  $\alpha$  by back substitution:

$$\beta = \sqrt{b_1} \quad \alpha = \sqrt{\frac{b_2}{b_1}} \quad (8)$$

For a more exhaustive explanation of this method for pose and focal length estimation we refer the reader to the above-mentioned papers.

### IV. IMAGE JACOBIAN AND CONTROL LAW

#### A. Image Jacobian

In camera coordinates, the motion of the target can be described by a translational velocity  $\mathbf{t}$  and an angular velocity  $\mathbf{\Omega}$ . The velocity of each control point in the camera frame is given by the expression

$$\dot{\mathbf{c}}_j = -\mathbf{t} - \mathbf{\Omega} \times \mathbf{c}_j \quad (9)$$

which corresponds to

$$\begin{bmatrix} \dot{x}_j \\ \dot{y}_j \\ \dot{z}_j \end{bmatrix} = \begin{bmatrix} -t_x - \omega_y z_j + \omega_z y_j \\ -t_y - \omega_z x_j + \omega_x z_j \\ -t_z - \omega_x y_j + \omega_y x_j \end{bmatrix} \quad (10)$$

Injecting Eq. 5 in Eq. 10, we obtain

$$\begin{bmatrix} \dot{x}_j \\ \dot{y}_j \\ \dot{z}_j \end{bmatrix} = \begin{bmatrix} -t_x - \omega_y \alpha \beta v_z + \omega_z \beta v_y \\ -t_y - \omega_z \beta v_x + \omega_x \alpha \beta v_z \\ -t_z - \omega_x \beta v_y + \omega_y \beta v_x \end{bmatrix}, \quad (11)$$

where  $v_x, v_y$ , and  $v_z$  are the  $x, y$ , and  $z$  components of eigenvector  $\mathbf{v}$  related to the control point  $\mathbf{c}_j$ , and whose image projection is given by

$$\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \alpha \frac{x_j}{z_j} + u_0 \\ \alpha \frac{y_j}{z_j} + v_0 \end{bmatrix}, \quad (12)$$

and its time derivative by

<sup>1</sup>In the noise-free case,  $\mathbf{M}^T \mathbf{M}$  is only rank deficient by one, but when image noise is severe  $\mathbf{M}^T \mathbf{M}$  might loose rank, and a more accurate solution can be found as a linear combination of the basis of its null space. In this work, we consider only the least squares approximation. That is, only the eigenvector associated to the smallest eigenvalue.

$$\begin{bmatrix} \dot{u}_j \\ \dot{v}_j \end{bmatrix} = \alpha \begin{bmatrix} \frac{\dot{x}_j}{z_j} - \frac{x_j \dot{z}_j}{z_j^2} & \frac{\dot{y}_j}{z_j} - \frac{y_j \dot{z}_j}{z_j^2} \\ \frac{\dot{y}_j}{z_j} - \frac{y_j \dot{z}_j}{z_j^2} & \frac{\dot{z}_j}{z_j} - \frac{z_j \dot{z}_j}{z_j^2} \end{bmatrix}. \quad (13)$$

Substituting Eqs. 5 and 11 in Eq. 13 we obtain

$$\dot{u}_j = \frac{-t_x - \alpha\beta v_z \omega_y + \beta v_y \omega_z}{\beta v_z} - \frac{v_x(-t_z - \beta v_y \omega_x + \beta v_x \omega_y)}{\alpha\beta v_z^2} \quad (14)$$

$$\dot{v}_j = \frac{-t_y - \alpha\beta v_z \omega_x + \beta v_x \omega_z}{\beta v_z} - \frac{v_y(-t_z - \beta v_y \omega_x + \beta v_x \omega_y)}{\alpha\beta v_z^2}, \quad (15)$$

which can be rewritten as

$$\dot{\mathbf{s}}_j = \mathbf{J}_j \mathbf{v}_c, \quad (16)$$

with  $\dot{\mathbf{s}}_j = [\dot{u}_j, \dot{v}_j]^T$ , the image velocities of control point  $j$ , and  $\mathbf{v}_c = [\mathbf{t}^T, \Omega^T]^T$ , the camera velocities.  $\mathbf{J}_j$  is our seeked calibration-free image Jacobian for the  $j$ -th control point, and takes the form

$$\mathbf{J}_j = \begin{bmatrix} \frac{-1}{\beta v_z} & 0 & \frac{v_x}{\alpha\beta v_z^2} & \frac{v_x v_y}{\alpha v_z^2} & \frac{-v_x^2 - \alpha^2 v_z^2}{\alpha v_z^2} & \frac{v_y}{v_z} \\ 0 & \frac{-1}{\beta v_z} & \frac{v_y}{\alpha\beta v_z^2} & \frac{v_x v_y}{\alpha v_z^2} & \frac{-v_y^2 - \alpha^2 v_z^2}{\alpha v_z^2} & \frac{-v_x}{v_z} \end{bmatrix}. \quad (17)$$

Stacking these together, we get the image Jacobian for all control points

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_4 \end{bmatrix}.$$

### B. Control Law

The aim of our image-based control schemes is to minimize the error

$$\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*, \quad (18)$$

where  $\mathbf{s}(t)$  are the current image coordinates of our set of target features, and  $\mathbf{s}^*$  are their final desired position in the image plane, computed with our initial value for  $\alpha$ . If we select  $\mathbf{s}$  to be the projection of the control points  $\mathbf{c}$ , and disregarding the time variation of  $\alpha$ , and consequently of  $\mathbf{s}^*$ , the derivative of Eq. 18 becomes

$$\dot{\mathbf{e}} = \dot{\mathbf{s}} = \mathbf{J} \mathbf{v}_c. \quad (19)$$

Assuming a holonomic platform, the camera velocities  $\mathbf{v}_c$  can be used to command the robot with an exponential decoupled decrease of the error, i.e.,  $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ ,

$$\mathbf{v}_c = -\lambda \mathbf{J}^+ \mathbf{e}, \quad (20)$$

where  $\mathbf{J}^+$  is chosen as the Moore-Penrose pseudo-inverse of  $\mathbf{J}$ , that is  $\mathbf{J}^+ = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$ .

### C. Stability analysis

To analyze the stability of this closed loop visual servo system, we will use Lyapunov analysis. Let  $\mathcal{L} = \frac{1}{2} \|\mathbf{e}(t)\|^2$  be a candidate Lyapunov function, whose derivative is given by [1]

$$\dot{\mathcal{L}} = \mathbf{e}^T \dot{\mathbf{e}} = -\lambda \mathbf{e}^T \mathbf{J} \mathbf{J}^+ \mathbf{e}. \quad (21)$$

Our system is globally asymptotic stable when the following sufficient condition holds

$$\mathbf{J} \mathbf{J}^+ > 0. \quad (22)$$

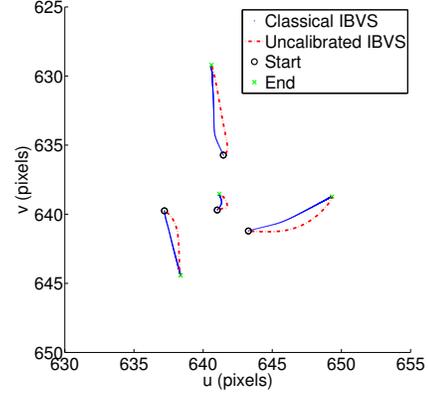


Fig. 1. Comparison of control point trajectories in the image plane under noise-free conditions.

The terms of  $\mathbf{v}$  in our Jacobian are the coordinates, albeit scale of our control points, which in turn form a basis of the original features. They are by construction linearly independent and it can be shown that with such selection of control points,  $\mathbf{J}$  has full rank 6 and thus, Ineq. 22 holds.

## V. SIMULATIONS AND EXPERIMENTS

In this section, a comparison between the classical image-based and the new uncalibrated image-based visual servoing approaches is presented, as well as an implementation of our method for an approach to land maneuver task for a quadrotor UAV.

### A. Simulations

The traditional image-based and the uncalibrated image-based approaches were simulated in Matlab-Simulink using as template the Visual Servoing Toolbox<sup>2</sup>.

Given a random set of target features, an initial camera position, and a desired final position with respect to the target, we want to compare the performance of the two algorithms. The camera is assumed to be fully controllable. That is, to have 6 dof, and the controller used is a proportional controller with a  $\lambda$  gain of 0.125. This value is chosen to be able to compare the time both algorithms take to reduce both image and Cartesian errors.

The simulation runs for 50 sec with time steps of 0.1 sec, and executes the two approaches with the same setup, except for the unknown camera calibration parameters in the uncalibrated case. Image trajectories of the control points are shown in Fig. 1. Fig. 2 shows the obtained camera trajectories, as well as the corresponding control points. The camera is indicated with a yellow tetrahedron in the initial and final locations, and the camera trajectories are shown as a concatenation of camera frames, with their axes depicted in red, green, and blue colors.

The time evolution of errors in the image plane and in Cartesian coordinates of the control points is plotted in Fig. 3. Under equal noise-free simulation conditions, both

<sup>2</sup><http://vstoolbox.sourceforge.net/>



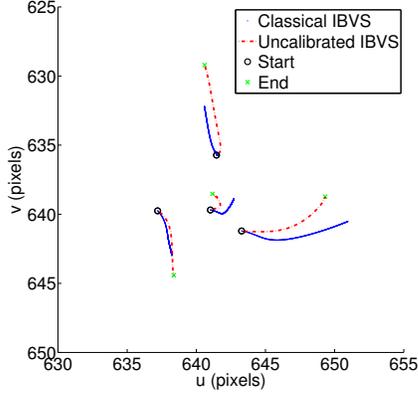


Fig. 6. Control point trajectories in the image plane with an unexpected zoom change of 20%.

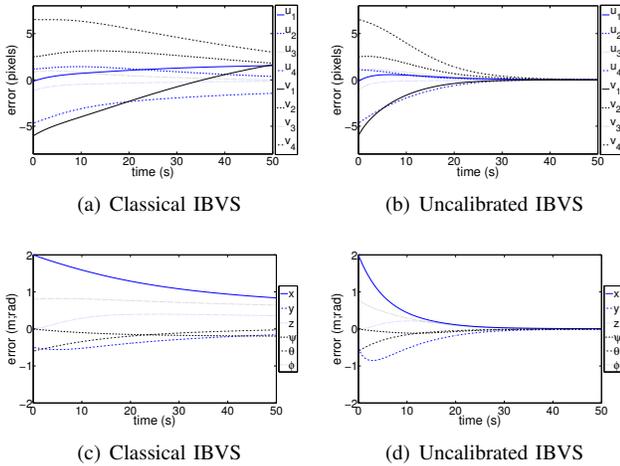


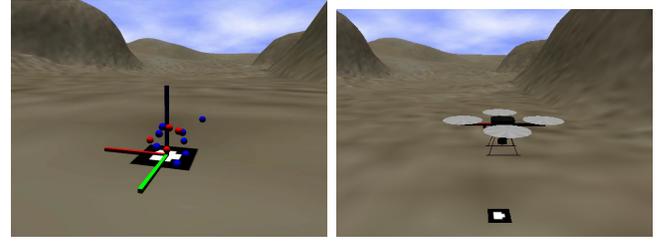
Fig. 7. Comparison of control point errors for unexpected zoom change of 20%, (a-b) as reprojections in the image plane, and (c-d) as Cartesian coordinates in the camera reference frame.

is unable to reach the desired configuration in the allotted simulation time, in contrast to the proposed approach in which the servoing task is completed without trouble. Figure 7 shows the control point error trajectories, in the image plane (a-b), and in camera centered Cartesian coordinates (c-d).

### B. Quadrotor experiments

This visual-based servo method was designed with a particular application in mind, that of maneuvering a quadrotor UAV to a desired location for robotic construction tasks. In the experiments reported here, we simulated an approach to landing maneuver by attaching a camera below the platform that observes an easily recognizable tag. The entire simulation system, including quadrotor kinematics and dynamics, was implemented in ROS using the Hector quadrotor stack [16].

As explained in the background formulation, we assume a set of randomly selected 3D feature points on the target and their 2D projections. This is achieved by attaching an ARTag



(a) Target reference frame and control points (b) Desired configuration

Fig. 8. Target and desired robot configuration for the approach to landing task. a) The ARTag, its reference frame, 3D feature points (blue), and their basis points (red). b) Desired robot configuration.

marker to the object frame, and computing the location of these random points with respect to the object frame, as well as its basis, i.e., the control points. At each iteration, the marker is detected in the scene, and the projection of the control points computed. Fig. 8 a shows the ARTag, its coordinate frame, the randomly selected feature points (blue dots), and the four control points that constitute the basis (red points). For simplicity, all other figures do not plot the feature or reference points, but simply the marker. Frame b in the same figure shows the desired robot position.

The quadrotor is teleoperated to an initial position, shown in Fig. 9 a-b. At that point, manual control is turned off and the uncalibrated image-based visual servo method activated, performing at each iteration marker detection, with the estimation of the location of the control points and the focal length. These values are used to evaluate the image Jacobian and to update the control law. The obtained velocities are used to control the UAV to its final destination as in Fig. 9 c-d.

The quadrotor is an underactuated vehicle [17], its pitch and roll are controlled by the attitude subsystem and cannot be directly actuated. Our control law does not have access to these variables and to remove from the control command, their contribution to the image error can be isolated from that of the other control variables as in [18]. Rewriting Eq. 19 as

$$\dot{\mathbf{e}} = \mathbf{J}_1 [v_x, v_y, v_z, \omega_z]^T + \mathbf{J}_2 [\omega_x, \omega_y]^T, \quad (23)$$

and rearranging terms, our control command becomes

$$[v_x, v_y, v_z, \omega_z]^T = \mathbf{J}_1^+ (\mathbf{J}_2 [\omega_x, \omega_y]^T - \lambda \mathbf{e}) \quad (24)$$

where  $\mathbf{J}_1^+$  is the Moore-Penrose pseudo-inverse of the Jacobian columns corresponding to  $[v_x, v_y, v_z, \omega_z]^T$ , and  $\mathbf{J}_2$  are the Jacobian columns corresponding to  $[\omega_x, \omega_y]^T$ . The platform gyros are used to read out values for  $\omega_x$  and  $\omega_y$ .

One last issue remains to be solved. The selection of the control gain  $\lambda$  bears a compromise between speed and stability. We must have in mind that our robot is a dynamical nonholonomic platform, and that velocity values in some directions cannot be achieved. Moreover, control commands must also guarantee that during the servoing task the camera does not lose the target object from its field of view. These two reasons suggest that low values of  $\lambda$  should be selected.

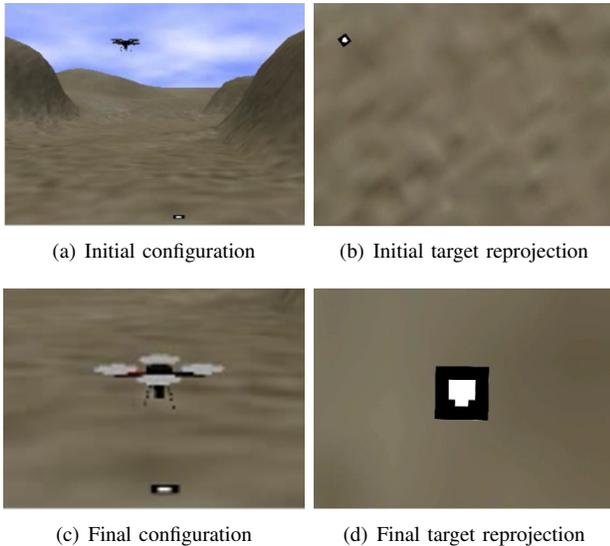


Fig. 9. Initial and final configurations for the approach to land task. The frames to the left show the robot location with respect to the target. The frames to the right show the target as seen by the camera in each case.

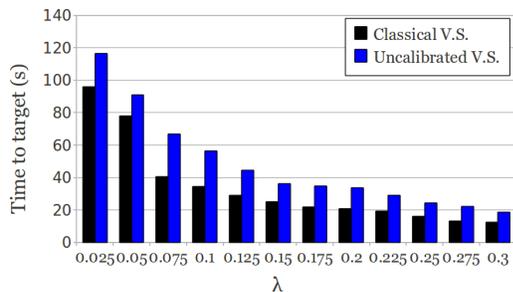


Fig. 10. Time to target comparison over the classical and the proposed approaches for different values of the control gain  $\lambda$ .

To determine an appropriate value of this parameter for our working conditions, we simulated both the calibrated and uncalibrated visual-based servoing schemes with increasing values of  $\lambda$ , and plotted the time to completion in Fig. 10. We can see that the proposed approach is slightly slower than the original approach. An optimal selection of this parameter is an issue of further research.

## VI. CONCLUSIONS

This paper presents a new visual-based uncalibrated visual servoing method. Target features are parameterized with their barycentric coordinates, and the basis of these coordinates is used to define a set of control points. A method is given to recover the coordinates of these control points and also of the camera focal length. With these, a new image Jacobian is derived which is guaranteed by construction to be of full rank. This guarantees asymptotic stability of the control law regardless of the target point selection, as long as planar configurations are avoided.

The technique is demonstrated in both Matlab and ROS simulations. All our code is available for download at

[www.ros.org/wiki/kinton\\_apps](http://www.ros.org/wiki/kinton_apps). A video of the method at work is also available in the same page.

We can think of two avenues for further research. On the one hand, servoing is constrained in the approach, as presented, to work only on the control points. To make it generic to any point in the target, their coordinates must be expressed using their barycentric coordinate coefficients. This situation entails very long terms in the Jacobian and we are thinking on ways to make them simpler. Secondly, the true dynamic behavior of the UAV must be taken into account in the design of the control law. There are some motion directions which are unattainable, and this might lead to instability. We leave these two topics as our current research interests.

## REFERENCES

- [1] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *Robot. Automat. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.
- [2] —, "Visual servo control. II. Advanced approaches," *Robot. Automat. Mag.*, vol. 14, no. 1, pp. 109–118, 2007.
- [3] S. Hutchinson, G. D. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, 1996.
- [4] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic visual servo control of robots: An adaptive image-based approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, St. Louis, Mar. 1985, pp. 662–668.
- [5] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, 1992.
- [6] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, ser. Lect. Notes Control Info. Sci., vol. 237, 1998, pp. 66–78.
- [7] E. Malis, F. Chaumette, and S. Boudet, "2-1/2-D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, 1999.
- [8] P. Corke and S. Hutchinson, "A new hybrid image-based visual servo control scheme," in *Proc. IEEE Int. Conf. Decision Control*, vol. 3, Sydney, Dec. 2000, pp. 2521–2526.
- [9] J. Piepmeyer, B. Gumpert, and H. Lipkin, "Uncalibrated eye-in-hand visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Washington, May 2002, pp. 568–573.
- [10] J. Piepmeyer, G. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Trans. Robot. Autom.*, vol. 20, no. 1, pp. 143–147, 2004.
- [11] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, 1994, pp. 186–193.
- [12] A. Farahmand, A. Shademan, and M. Jagersand, "Global visual-motor estimation for uncalibrated visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, Nov. 2007, pp. 1969–1974.
- [13] A. Shademan, A. Farahmand, and M. Jagersand, "Robust Jacobian estimation for uncalibrated visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, May 2010, pp. 5564–5569.
- [14] F. Moreno-Noguer, V. Lepetit, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vision*, vol. 81, no. 2, pp. 155–166, 2009.
- [15] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer, "Exhaustive linearization for robust camera pose and focal length estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, to appear.
- [16] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk, "Comprehensive simulation of quadrotor UAVs using ROS and Gazebo," in *Proc. 3rd Int. Conf. Simul. Model. Program. Auton. Robot.*, ser. Lect. Notes Comput. Sci., vol. 7628, Tsukuba, Japan, Nov. 2012, pp. 400–411.
- [17] T. Hamel and R. Mahony, "Image based visual servo control for a class of aerial robotic systems," *Automatica*, vol. 43, no. 11, pp. 1975–1983, 2007.
- [18] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *Robot. Automat. Mag.*, vol. 19, no. 3, pp. 20–32, 2012.