A comparison between Position Based and Image Based Visual Servoing on a 3 DOFs translating robot

Giacomo Palmieri[1], Matteo Palpacelli[2], Massimiliano Battistelli[2]

[1]*Università degli Studi e-Campus, Italy*
E-mail: *giacomo.palmieri@uniecampus.it*

[2]*Dipartimento di Ingegneria Industriale e Scienze Matematiche,*
Università Politecnica delle Marche, Italy
E-mail: *m.palpacelli@univpm.it* *m.battistelli@univpm.it*

Keywords: Position Based Visual Servoing, Image Based Visual Servoing.

SUMMARY. Two Visual Servoing controls respectively of the Position Based and Image Based type have been developed to govern a parallel translating manipulator with an eye-in-hand configuration. The robot must be able to reach and grasp a special target randomly positioned in the workspace; the control must be adaptive to compensate motions of the target in the 3D space. The implemented trajectory planning strategy ensures the continuity of the velocity vector in the Cartesian space for both PBVS and IBVS controls whereas a replanning event is needed. A comparison between the two approaches is given in terms of accuracy, fastness and stability in relation to the robot peculiar characteristics.

1 INTRODUCTION

Visual servoing is the use of computer vision to control the motion of a robot; two basic approaches can be identified [1-3]: *Position-Based Visual Servo* (PBVS), in which vision data are used to reconstruct the 3D pose of the robot and a kinematic error is generated in the Cartesian space and mapped to actuators commands [4-5]; *Image-Based Visual Servo* (IBVS), in which the error is generated directly from image plane features [6-9]. The principal advantage of using position-based control is the chance of defining tasks in a standard Cartesian frame. On the other hand, the control law depends on the vision system calibration parameters, and can become widely sensitive to calibration errors. On the contrary, the image-based control is less sensitive to calibration errors; however, it is required the online calculation of the image Jacobian, that is a quantity depending on the distance between the target and the camera and it is difficult to evaluate. A control in the image plane results also strongly nonlinear and coupled when mapped on the joint space of the robot and may be cause of problems due to the passage through points which are singular for the kinematics of the manipulator [1].

Visual servo systems can also be classified on the basis of their architecture in the following two categories [1]: the vision system provides an external input to the joint closed loop control of the robot that stabilizes the mechanism (*dynamic look and move*); the vision system is directly used in the control loop to compute joints inputs, thus stabilizing autonomously the robot (*direct visual servo*). In general most applications are of the dynamic look and move type; one of the reasons is the difference between vision systems and standard servo control loops rates. The low frequency imposed by vision might cause problems on the stability of the manipulator control system, especially in cases where several DOFs are involved.

The aim of this work is to develop and compare two different Visual Servoing controls, respectively of the Position Based and Image Based type. They are used to govern a parallel translating manipulator with an eye-in-hand configuration. The robot must be able to reach and grasp a special target randomly positioned in the workspace; the control must be adaptive to compensate motions of the target in the 3D space. The use of Bézier curves in the trajectory planning algorithms ensures the continuity of the velocity vector in the Cartesian space for both PBVS and IBVS controls whereas a replanning event is needed. A dynamic look and move philosophy has been adopted to conjugate the low frame rate of the vision system (lower than 30 Hz) with the high control rate of the joint servo (about 1000 Hz). The robot, which is called ICaRo, is a research prototype designed and realized at the Department of Mechanics of the Polytechnic University of Marche [10]; the end-effector is in-parallel actuated by 3 limbs whose kinematic structure only allows pure translations in the 3D space. The workspace is a cube of 0.6 m edge free of singular points. The eye-in-hand configuration has been adopted for the installation of the vision system. The optical axis of the camera is aligned with the end effector vertical axis. The robot is managed by a central control unit, a DS1103 real-time board by dSPACE. The vision data, after image acquisition and preliminary processing made by the CVS real-time hardware of National Instruments, are sent via a serial interface to the central unit that performs the control algorithms.

2 PBVS CONTROL

Using a position based method, a 3D camera calibration is required in order to map the 2D data of the image features to the Cartesian space data. This is to say that intrinsic and extrinsic parameters of the camera must be evaluated. *Intrinsic parameters* depend exclusively on the optical characteristics, e.g. lens and CCD sensor properties. The calibration of intrinsic parameters can be operated offline in the case that optical setup is fixed during the operative tasks of the robot [11]. *Extrinsic parameters* indicate the relative pose of the camera reference system $O_c - x_c y_c z_c$ with respect to a generic world reference system. It is assumed that the world reference system is a system $O_o - x_o y_o z_o$ that is fixed with the target object, so that the extrinsic parameters give directly the pose of the camera with respect to the target. The extrinsic parameters matrix coincides with the homogeneous transformation between the camera and the object reference systems:

$$\mathbf{T}_o^c = \begin{bmatrix} \mathbf{R}_o^c & \mathbf{t}_o^c \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{1}$$

The object coordinate system is attached to a pattern of four coplanar white circles painted on the top surface of the target itself. The aim is to determine the extrinsic parameters, once the geometry of the optical pattern and the pixel coordinates of the centroids of the circles projected onto the image plane are known. The problem of determining the pose, knowing the correspondence of *n* points in the world and camera frame reference systems, is typical in photogrammetry (*PnP* problem), and it is proved that for 4 coplanar points the solution is unique [12].

The global architecture of the PBVS control is shown in figure 1; two separate loops can be seen: the inner loop realizes a standard PD control in the operative space with gravity compensation ($\mathbf{g}$ term) by exploiting the information from encoders; the camera, working as an exteroceptive sensor, closes an outer loop where visual information is processed in order to plan and replan in real time the desired trajectory of the robot. A FIR digital filter is used to reduce small oscillations of the signal that deeply affect the control stability.

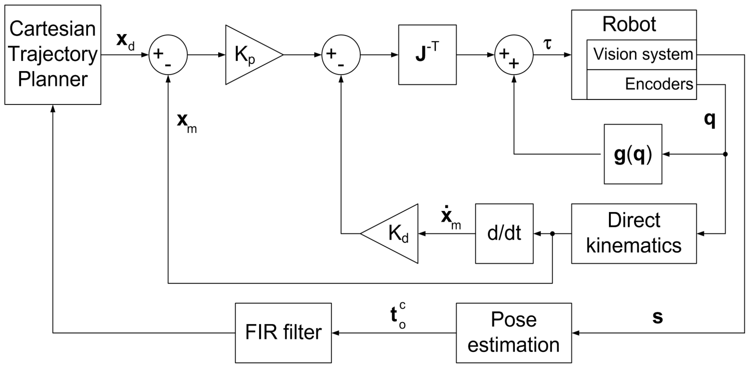

Figure 1: dynamic look and move PBVS control.

A control in the Cartesian space is realized imposing the following requirements to the planned motions: starting from an arbitrary status of the end-effector (position and velocity), the continuity of position and velocity must be ensured; the final point must be reached with a vertical tangent in order to facilitate the grasping of the target object.

To this purpose a third order Bézier curve has been adopted [13]. This curve is defined by 4 points in the 3D space and has the fundamental property of being tangent to the extremities of the polyline defined by the four points (figure 2a). The parametric formulation of a cubic Bézier curve is:

$$\mathbf{B}(s) = \mathbf{P}_0\left(1-s^3\right) + 3\mathbf{P}_1 s(1-s)^2 + 3\mathbf{P}_2 s^2(1-s) + \mathbf{P}_3 s^3 \qquad s \in [0,1] \tag{2}$$

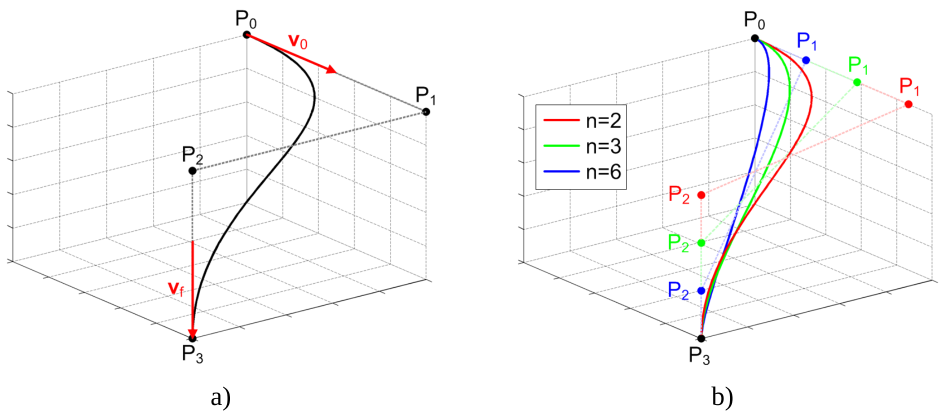

Figure 2: a) 3rd order Bézier curve in a 3D space; b) effect of the parameter n.

In the trajectory planner algorithm $\mathbf{P}_0$ is the current position of the end-effector, while $\mathbf{P}_3$ is the target point obtained from the vision system. Knowing the current velocity direction $\hat{\mathbf{v}}_0$ and the desired final velocity direction $\hat{\mathbf{v}}_f$, $\mathbf{P}_1$ and $\mathbf{P}_2$ are defined as

$$\mathbf{P}_1 = \mathbf{P}_0 + \frac{\left\|\mathbf{P}_3 - \mathbf{P}_0\right\|}{n}\hat{\mathbf{v}}_0 \qquad \mathbf{P}_2 = \mathbf{P}_3 - \frac{\left\|\mathbf{P}_3 - \mathbf{P}_0\right\|}{n}\hat{\mathbf{v}}_f \tag{3}$$

where n is a tunable parameter that influences the curvature of the trajectory (figure 2b).

Since $\mathbf{P}_3$ is continuously estimated during the motion of the robot, variations of its position are compensated by a continuous trajectory replanning. A trapezoidal velocity profile is defined by a temporal planning algorithm that is able to modify the shape of the profile if the total length of the planned trajectory changes during the motion.

3 IBVS CONTROL

In the image-based visual servo control the error signal is defined directly in terms of image feature parameters. In the present application the optical target is an annulus on the top surface of a cylinder. The image features are the coordinates of the centre and the major diameter of the ellipse resulting from the projection of the annulus on the image plane; they are collected in the feature space vector $\mathbf{s} = \{u, v, d\}^T$ expressed in pixel. The relation between feature space and Cartesian space is found in the interaction matrix $\mathbf{L_e}$ [12]: being $\dot{\mathbf{x}} = \{\dot{x}, \dot{y}, \dot{z}\}^T$ the Cartesian velocity and $\dot{\mathbf{s}} = \{\dot{u}, \dot{v}, \dot{d}\}^T$ the feature space velocity, it is

$$\dot{\mathbf{s}} = \mathbf{L_e}\dot{\mathbf{x}} \tag{4}$$

Here the interaction matrix is defined as

$$\mathbf{L_e} = \begin{bmatrix} f/z & 0 & -u/z \\ 0 & f/z & -v/z \\ 0 & 0 & -fD/z^2 \end{bmatrix} \tag{5}$$

where f is the focal length (assumed equal in u and v directions) expressed in pixel and D is the metric value of the annulus diameter in the Cartesian space. It arises from (5) that the interaction matrix is a function of image features u, v and of the Cartesian variable z. The estimation of z is performed by a linear interpolation on the parameter d; the interpolation law is obtained in a preliminary calibration step. The inversion of equation (4) is performed at each time step of the control by means of the Moore-Penrose pseudoinverse, so that

$$\dot{\mathbf{x}} = \mathbf{L_e^+}\dot{\mathbf{s}} \tag{6}$$

The trajectory planner operates in this case on a 3D hybrid space defined by the two image coordinates u and v, and the vertical Cartesian direction z. In analogy with the PBVS control, it is defined a Bézier curve according to equation (2) trough points which belong to the hybrid space $\{u, v, z\}$. Such kind of strategy allows for the continuity of the velocity in the 3Dspace even in the case of motion of the target. In the hybrid space the variable z is chosen instead of d in order to plan conveniently the vertical speed of the robot. In fact, introducing the relation

$$d = \frac{fD}{z} \tag{7}$$

it results for the velocities:

$$\dot{d} = -fD\frac{\dot{z}}{z^2} \tag{8}$$

Thus, if a constant $\dot{d}$ is planned, the resulting vertical speed $\dot{z}$ would not be constant and tends to zero when the camera approaches the target. It follows that a better control is achievable planning directly in the z variable.

The scheme of the implemented IBVS control is shown in figure3. As in PBVS it is possible to individuate the inner and outer loops. In the outer loop the pose estimation block disappears and the image features are directly the input of the trajectory planning block. Here the real time estimation of $\mathbf{L}_\mathbf{e}^+$ is required in order to map the trajectory from the feature to the Cartesian space.

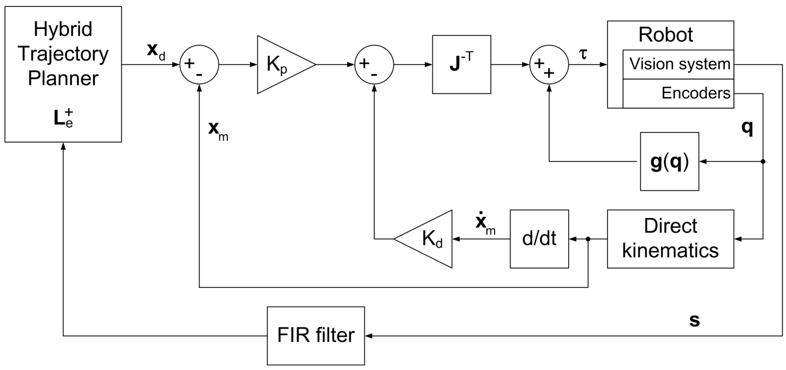

Figure 3: dynamic look and move IBVS control.

4 EXPERIMENTAL RESULTS

A series of tests have been performed in order to compare the PB and IB approaches at different speeds and in cases of fixed or moving target. The starting position of the robot allows the camera to frame a wide area of the workspace; when the target is recognized by the vision system, the controller starts to govern the robot till the target is reached.

Figure 4 shows the results of the slow tests ($v_{max} \simeq 10^{-1}$ m/s) with a fixed target. The plots on the left side are referred to the PBVS, while IBVS results are on the right column; the position of the target is the same for both tests. The trajectories on the image plane of the image features and of the corresponding centroid are represented on the top of the figure; such features are the four coplanar circles for the PB and the radius of the annulus for the IB. The Cartesian position and velocity are plotted respectively in the middle and at the bottom of the figure; lines refer to measured entities, while circles show the planned trajectory and velocity profile. It results that the

IB control has a slightly higher accuracy in centering the target. Further on, the IBVS shows a higher initial peak in x and y components of velocity, which allows for quickly compensating the offset between camera and target in the horizontal plane.

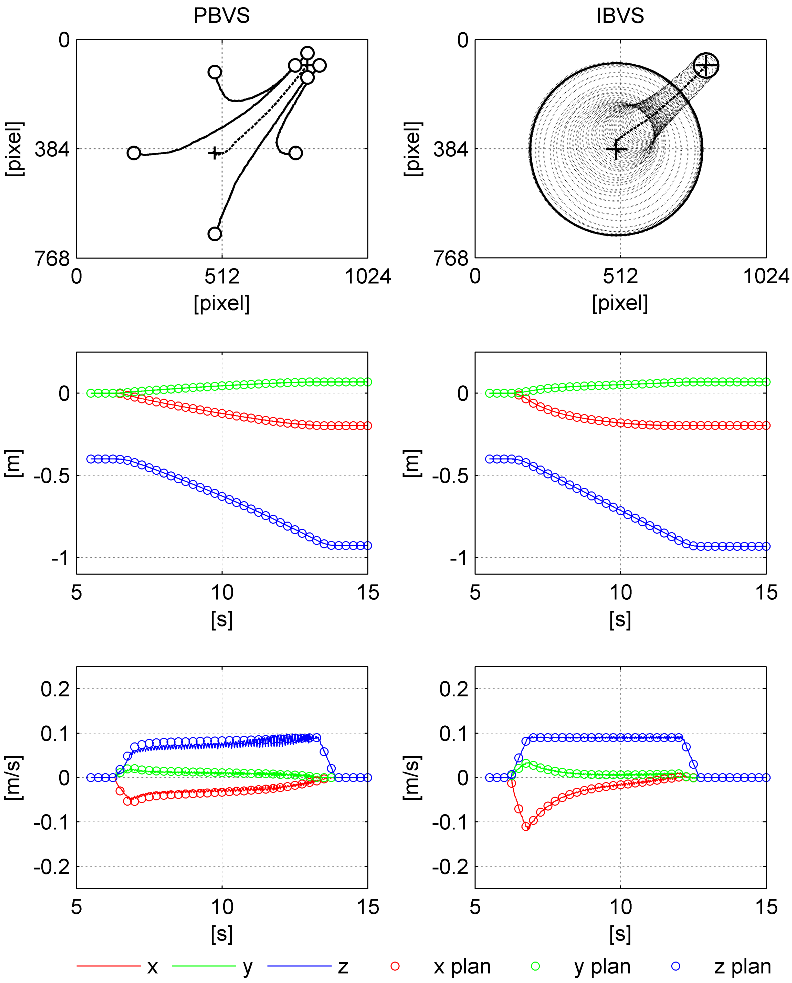

Figure 4: slow tests ($v_{max} \simeq 10^{-1}$ m/s) with fixed target.

The same considerations are even more evident in fast tests where the maximum velocity of the robot is twice than in the case of slow tests ($v_{max} \simeq 2 \cdot 10^{-1}$ m/s). Results are plotted in figure 5 in analogy with figure 4. Against a strong decrease of the time required to reach the target, in both PB an IB controls it is not noticed a loss of accuracy.

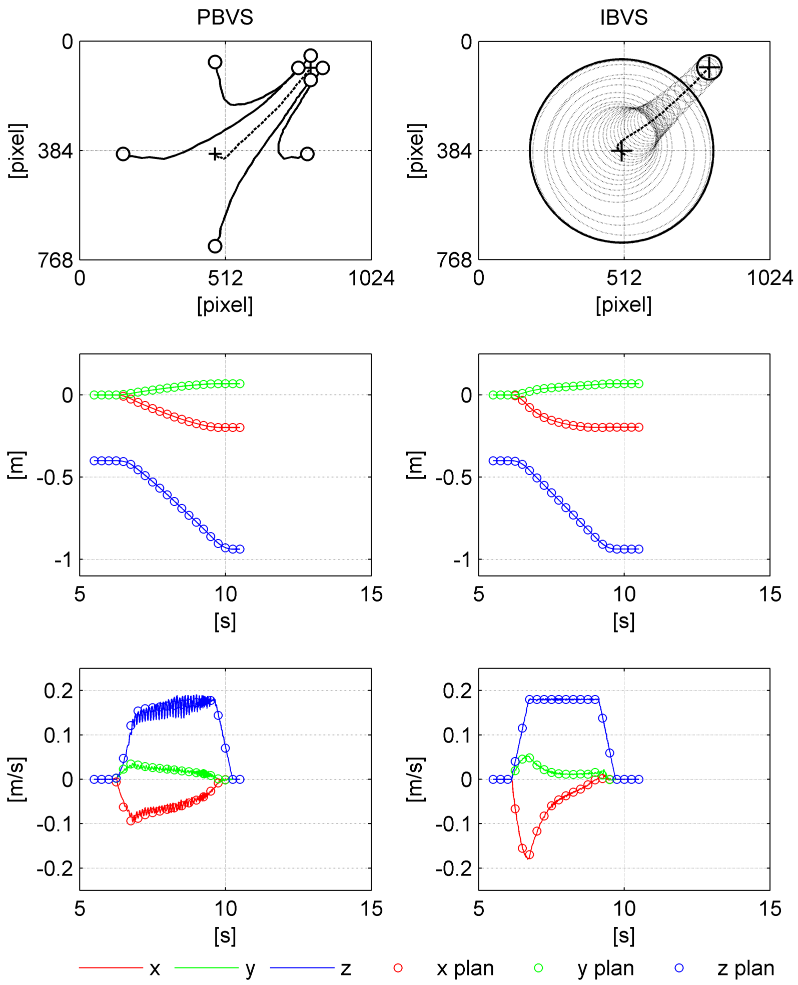

Figure 5: fast tests (v_{max}≃$2{\cdot}10^{-1}$ m/s) with fixed target.

A further experimental evaluation is here described to prove the ability of the control in compensating a generic motion of the target: once the visual control has started, the object is moved by means of a conveyor belt; the controller is able to replan continuously the trajectory ensuring the continuity of the velocity and quickly compensating the displacement imposed to the target. The original (referred to the PB algorithm) and replanned trajectories are plotted in figure 6 in orthographic and axonometric views. The initial higher velocity on the horizontal plane is once again evident for the IBVS with the result of a longer Cartesian trajectory; dealing with the accuracy, no relevant differences between PBVS and IBVS are appreciated.

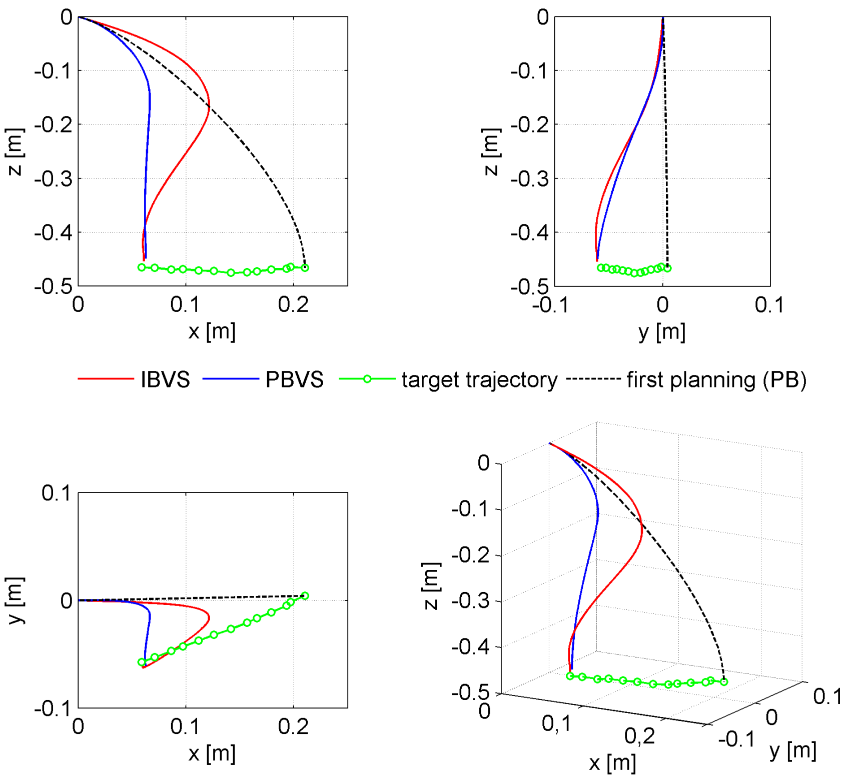

Figure 6: IBVS and PBVS tests with moving target.

5 CONCLUSIONS

Two dynamic look and move visual servos, respectively of the Position Based and Image Based type, have been implemented to govern a parallel translating robot. Both controls are able to detect a target in the 3D workspace and to compensate a generic motion of the target itself. A continuous replanning algorithm defines the trajectory of the end-effector ensuring the continuity of the velocity vector even if sudden displacements are imposed to the target; trajectories are planned using cubic Bézier curves that give a smooth spatial path and allow to impose the final direction of the velocity in order to facilitate the grasping task; such curves are planned in the Cartesian space for the PBVS, while a hybrid space is conveniently defined for the IBVS.

Experimental results demonstrate that the accuracies of the two proposed controls are substantially comparable. Nevertheless, the IBVS shows a higher sensitivity to horizontal displacement proved by initial velocity peaks on x and y directions. Both methods requires the knowledge of the geometric model of the optical target. The PB approach requires also the 3D calibration of the camera in order to estimate the intrinsic parameters used in the *PnP* algorithm.

The sensitivity on calibration errors is one of the main drawbacks of the PBVS. Further on, even if the solution to the *PnP* problem returns the homogeneous transformation matrix $\mathbf{T}_o^c$, only the vector $\mathbf{t}_o^c$ is then a useful information for a pure translation robot.

These reasons make the IBVS preferable for the studied application: the method is simple, accurate and does not require a camera calibration; also typical problems of stability of IB controls are avoided since the robot is free of singular points inside its workspace..

References

[1] Hutchinson, S., Hager, G.D. and Corke, P.I., "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, **12(5)**, 651-670 (1996).

[2] Corke, P.I. and Hutchinson, S., "Real-time vision, tracking and control," in *Proc. of the 2000 IEEE International Conference on Robotics & Automation*, 622-629 (2000).

[3] Chaumette, F. and Hutchinson, S., "Visual servo control, part I: basic approaches," *IEEE Robotics & Automation Magazine*, **13(4)**, 82-90 (2006).

[4] Wilson, W.J., Williams Hulls, C.C. and Bell, G.S., "Relative end-effector control using Cartesian position based visual servoing," *IEEE Trans. on Robotics and Automation*, **12(5)**, 684-696 (1996).

[5] Gans, N.R., Dani, A.P. and Dixon, W.E., "Visual servoing to an arbitrary pose with respect to an object given a single known length," in *Proc. of the 2008 American Control Conference*, 1261-1267 (2008).

[6] Weiss, L.E., Sanderson, A.C. and Neuman, C.P., "Dynamic visual servo control of robots: An adaptive image-based approach," in *Proc. of the 1985 IEEE International Conference on Robotics and Automation 2*, 662- 668 (1985).

[7] Hashimoto, K., Kimoto, T., Ebine, T. and Kimura, H., "Manipulator control with image-based visual servo," in *Proc. of the 1991 IEEE Intemational Conference on Robotics and Automation*, 2267-2272 (1991).

[8] Qi, Z. and McInroy, J.E., "Improved image based visual servoing with parallel robot," *Journal of Intelligent & Robotic Systems*, **53**, 359-379 (2008).

[9] Fioravanti, D., Allotta, B. and Rindi, A., "Image based visual servoing for robot positioning tasks," *Meccanica*, **43**, 291-305 (2008).

[10] Callegari, M. and Palpacelli, M.C., "Prototype design of a translating parallel robot," *Meccanica*, **43(2)**, 133-151 (2008).

[11] Heikkilä, J., Silvén, O., "A Four-step Camera Calibration Procedure with Implicit Image Correction," in *Proc. CVPR Computer Vision and Pattern Recognition, IEEE Computer Society*, 1106-1112 (1997).

[12] Siciliano, B. and Khatib, O., *Springer Handbook of Robotics*, Springer (2008).

[13] Sederberg, T.W., *Computer aided geometric design*, BYU, Computer Aided Geometric Design Course Notes (2011).