Encyclopedia of Systems and Control, 1188-1194, Springer-Verlag, 2015.

Robot Visual Control

François Chaumette, Inria, Rennes, France

Abstract

This article describes the basic concepts of vision-based control, that is, the use of visual data to control the motions of a robotics system. It describes the modeling steps allowing the design of kinematics control schemes. Applications are also described.

Keywords

Robot control, Visual servoing, Jacobian, Kinematics

Introduction

Visual control, also named visual servoing, refers to the use of computer vision data as input of real-time closed-loop control schemes to control the motion of a dynamic system, a robot typically [Hutchinson et al, 1996], [Chaumette & Hutchinson, 2008]. It can be seen as sensor-based control from a vision sensor and relies on techniques from image processing, computer vision, and control theory.

An iteration of the control scheme consists of the following steps:

- acquire an image;
- extract some useful image measurements;
- compute the current value of the visual features used as inputs of the control scheme;
- compute the error between the current and the desired values of the visual features;
- update the control outputs, which are usually the robot velocity, to regulate that error to zero, *i.e.* , to minimize its norm.

For instance, for the first example depicted on Figure 1, the image processing part consists in extracting and tracking the center of gravity of the moving people, the visual features are composed of the two Cartesian coordinates of this center of gravity, and the control scheme computes the pan and tilt velocities so that the center of gravity is as near as possible of the image center despite the unknown motion of the people. In the second example where a camera mounted on a six degrees of freedom robot arm is considered, the image measurements are a set of segments that are tracked in the image sequence. From these measurements and the knowledge of the 3D object model, the pose from the camera to the object is estimated and used as visual features. The control scheme now computes the six components of the robot velocity so that this pose reaches a particular desired value corresponding to the object position depicted in blue on the images.

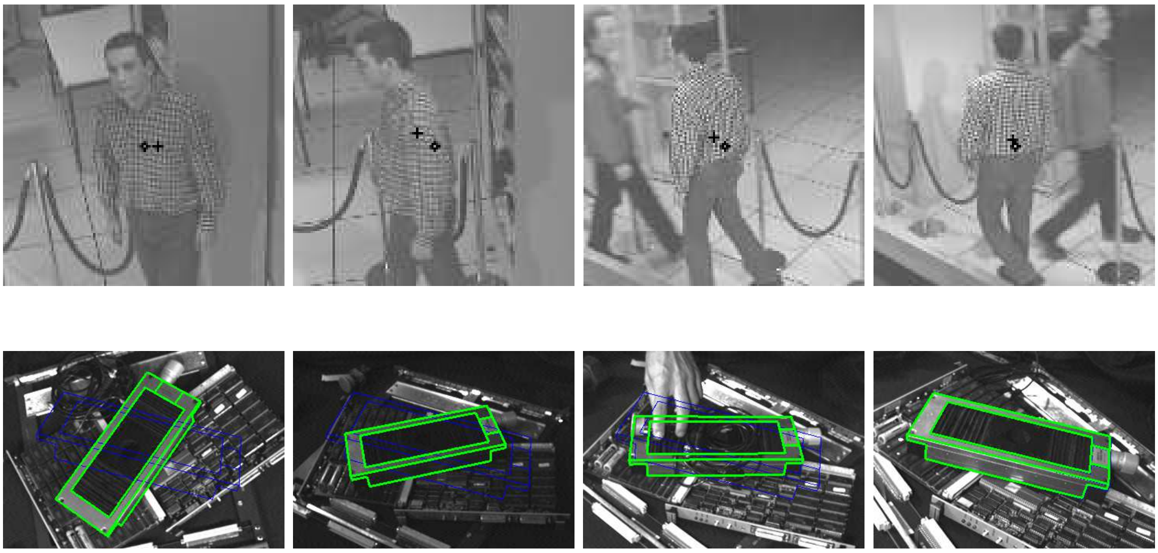

Fig. 1. Few images acquired during two visual servoing tasks: on the top, pedestrian tracking using a pan-tilt camera; on the bottom, controlling the 6 degrees of freedom of an eye-in-hand system so that an object appears at a particular position in the image (shown in blue).

Basic Theory

Main if not all visual servoing tasks can be expressed as the regulation to zero of an error $\mathbf{e}(t)$ which is defined by

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(\mathbf{r}(t)), \mathbf{a}) - \mathbf{s}^*(t). \qquad (1)$$

The parameters in (1) are defined as follows [Chaumette & Hutchinson, 2008] The vector $\mathbf{m}(\mathbf{r}(t))$ is a set of image measurements (*e.g.* the image coordinates of interest points, or the area, the center of gravity and other geometric characteristics of an object). These image measurements depend on the pose $\mathbf{r}(t)$ between the camera and the environment. They are used to compute a vector $\mathbf{s}(\mathbf{m}(\mathbf{r}(t)), \mathbf{a})$ of visual features, in which $\mathbf{a}$ is a set of parameters that represent potential additional knowledge about the system (*e.g.* coarse camera intrinsic parameters or 3D model of objects). The vector $\mathbf{s}^*(t)$ contains the desired value of the features, which can be either constant in the case of a fixed goal, or varying if the task consists in following a specified trajectory.

Visual servoing schemes mainly differ in the way that the visual features are designed. As represented on Figure 2, the two most classical approaches are named image-based visual servoing (IBVS), in which $\mathbf{s}$ consists of a set of 2D parameters that are directly expressed in the image [Weiss et al, 1987], [Espiau et al, 1992], and pose-based visual servoing (PBVS), in which $\mathbf{s}$ consists of a set of 3D parameters related to the pose between the camera and the target [Weiss et al, 1987], [Wilson et al, 1996]. In that case, the 3D parameters have to be estimated from the image measurements either through a pose estimation process using the knowledge of the 3D target model, or through a partial pose estimation process using the properties of the epipolar geometry between the current and the desired images, or finally through a triangulation process if a stereovision system is considered. Inside IBVS and PBVS approaches, many possibilities exist depending on the choice of the features. Each choice will induce a particular behavior of the system. There also exist hybrid approaches, named 2-1/2D visual servoing, which combine 2D and 3D parameters in $\mathbf{s}$ in order to benefit from the advantages of IBVS and PBVS while avoiding their respective drawbacks [Malis et al, 1999].

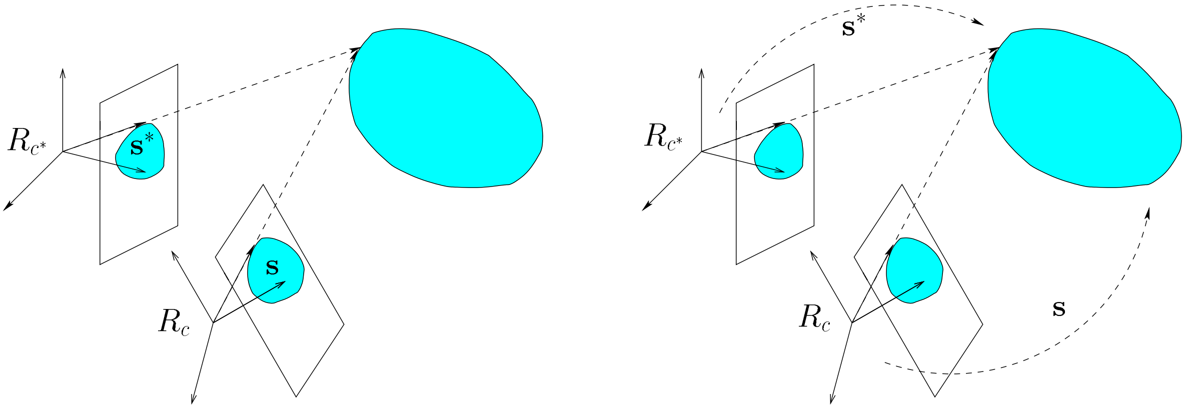

Fig. 2. If the goal is to move the camera from frame R_c to the desired frame R_{c^*}, two main approaches are possible: IBVS on the left, where the features $\mathbf{s}$ and $\mathbf{s}^*$ are expressed in the image, and PBVS on the right, where the features $\mathbf{s}$ and $\mathbf{s}^*$ are related to the pose between the camera and the observed object.

The design of the control scheme is based on the link between the time variation of the features and the robot control inputs, which are usually the velocity of the robot joints $\mathbf{q}$. This relation is given by

$$\dot{\mathbf{s}} = \mathbf{J_s}\,\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \qquad (2)$$

where $\mathbf{J_s}$ is the features Jacobian matrix, defined from the equation above similarly as the classical robot Jacobian. For an eye-in-hand system (see the left part of Figure 3), the term $\frac{\partial \mathbf{s}}{\partial t}$ represents the time variation of $\mathbf{s}$ due to a potential object motion, while for an eye-to-hand system (see the right part of Figure 3) it represents the time variation of $\mathbf{s}$ due to a potential sensor motion.

As for the features Jacobian, in the eye-in-hand configuration, it can be decomposed as [Chaumette & Hutchinson 2008]

$$\mathbf{J_s} = \mathbf{L_s}\,{}^{c}\mathbf{V}_n\,\mathbf{J}(\mathbf{q}) \qquad (3)$$

where

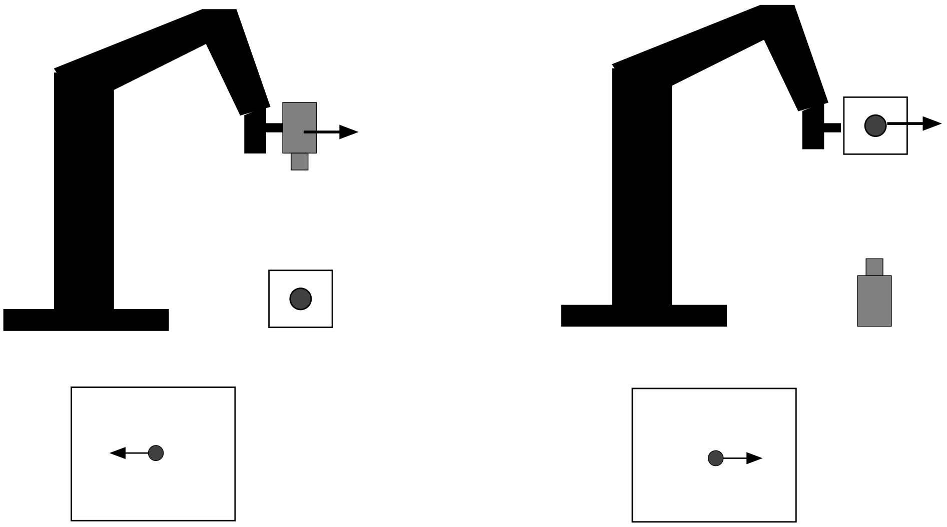

Fig. 3. In visual servoing, the vision sensor can be either mounted on the robot (eye-in-hand configuration) or observing it (eye-to-hand configuration). For the same robot motion, the motion produced in the image will be opposite from one configuration to the other.

- $\mathbf{J}(\mathbf{q})$ is the robot Jacobian such that $\mathbf{v}_n = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ where $\mathbf{v}_n$ is the robot end effector velocity;
- ${}^c\mathbf{V}_n$ is the spatial motion transform matrix from the vision sensor to the end effector. It is given by
$$ {}^c\mathbf{V}_n = \begin{bmatrix} {}^c\mathbf{R}_n & [{}^c\mathbf{t}_n]_\times \, {}^c\mathbf{R}_n \\ \mathbf{0} & {}^c\mathbf{R}_n \end{bmatrix} \tag{4} $$
where ${}^c\mathbf{R}_n$ and ${}^c\mathbf{t}_n$ are respectively the rotation matrix and the translation vector between the sensor frame and the end effector frame, and where $[{}^c\mathbf{t}_n]_\times$ is the skew symmetric matrix associated to ${}^c\mathbf{t}_n$. Matrix ${}^c\mathbf{V}_n$ is constant when the vision sensor is rigidly attached to the end effector, which is usually the case. Thanks to the robustness of closed-loop control schemes, a coarse approximation of ${}^c\mathbf{R}_n$ and ${}^c\mathbf{t}_n$ is sufficient in practice to get an estimation of ${}^c\mathbf{V}_n$. If needed, an accurate estimation is possible through classical hand-eye calibration methods.
- $\mathbf{L_s}$ is the interaction matrix of $\mathbf{s}$ defined such that $\mathbf{s} = \mathbf{L_s}\mathbf{v}$ where $\mathbf{v}$ is the relative velocity between the camera and the environment.

In the eye-to-hand configuration, the features Jacobian $\mathbf{J_s}$ is composed of

[Chaumette & Hutchinson, 2008]

$$\mathbf{J_s} = -\mathbf{L_s}\, {}^c\mathbf{V}_f\, {}^f\mathbf{V}_n\, \mathbf{J}(\mathbf{q}) \tag{5}$$

where

- ${}^f\mathbf{V}_n$ is the spatial motion transform matrix from the robot reference frame to the end effector frame. It is known from the robot kinematics model.
- ${}^c\mathbf{V}_f$ is the spatial motion transform matrix from the camera frame to the reference frame. It is constant as long as the camera does not move. In that case, similarly as for the eye-in-hand configuration, a coarse approximation of ${}^c\mathbf{R}_f$ and ${}^c\mathbf{t}_f$ is usually sufficient to get an estimation of ${}^c\mathbf{V}_f$.

A lot of works have concerned the modeling of the visual features and the determination of the analytical form of the interaction matrix. To give just an example, in the case of an image point with normalized Cartesian coordinates $\mathbf{x} = (x, y)$ and whose 3D corresponding point has depth Z, its interaction matrix is given by [Espiau et al, 1992]

$$\mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{6}$$

where the three first columns contain the elements related to the three components of the translational velocity, and where the three last columns contain the elements related to the three components of the rotational velocity.

By just changing the parameters representing the same image point, that is by using the cylindrical coordinates defined by $\boldsymbol{\gamma} = (\rho, \theta)$ with $\rho = \sqrt{x^2+y^2}$ and $\theta = \text{Arctan}(y/x)$, the interaction matrix of these parameters has a completely different form [Chaumette & Hutchinson, 2008]:

$$\mathbf{L}_{\boldsymbol{\gamma}} = \begin{bmatrix} -c/Z & -s/Z & \rho/Z & (1+\rho^2)s & -(1+\rho^2)c & 0 \\ s/(\rho Z) & -c/(\rho Z) & 0 & c/\rho & s/\rho & -1 \end{bmatrix} \tag{7}$$

where $c = \cos\theta$ and $s = \sin\theta$. This implies that using the Cartesian coordinates or the cylindrical coordinates as visual features will induce a different behavior, that is a different robot trajectory and a different trajectory of the point in the image.

Currently, the analytical form of the interaction matrix is available for most classical geometrical primitives, such as segments, straight lines, ellipses, moments related to planar objects of any shape [Chaumette, 2004], and also coordinates of 3D points and pose parameters. Methods also exist to estimate off-line or online a numerical value of the interaction matrix. Omnidirectional vision sensors, the coupling between a camera and structured light, and even 2D echographic probes have also been studied. A large variety of visual features is thus available for many vision sensors.

Once the modeling step has been performed, the design of the control scheme can be quite simple. The most classical control scheme has the following form [Chaumette & Hutchinson, 2008]

$$\dot{\mathbf{q}} = -\lambda \widehat{\mathbf{J}_{\mathbf{s}}}^{+}(\mathbf{s}-\mathbf{s}^*) + \widehat{\mathbf{J}_{\mathbf{s}}}^{+}\frac{\partial \mathbf{s}^*}{\partial t} - \widehat{\mathbf{J}_{\mathbf{s}}}^{+}\widehat{\frac{\partial \mathbf{s}}{\partial t}} \tag{8}$$

where λ is a positivive gain tuning the rate of convergence of the system and $\widehat{\mathbf{J}_{\mathbf{s}}}^{+}$ is the Moore-Penrose pseudo inverse of an approximation or an estimation of the features Jacobian. The exact value of all its elements is indeed generally unknown since it depends of the intrinsic and extrinsic camera parameters, as well as of some 3D parameters such as the depth of the point in Equations (6) and (7).

The second term of the control scheme anticipates for the variation of $\mathbf{s}^*$ in the case of a varying desired value. The third term compensates as much as possible a possible target motion in the eye-in-hand case and a possible camera motion in the eye-to-hand case. They are both null in the case of a fixed desired value and a motionless target or camera. They try to remove the tracking error in the other cases.

Following the Lyapunov theory, the stability of the system can be studied [Chaumette & Hutchinson 2008]. Generally, visual servoing schemes can be demonstrated to be locally asymptotically stable (*i.e.* , the robot will converge if it starts from a local neighborhood of the desired pose) if the errors introduced in $\widehat{\mathbf{J}_{\mathbf{s}}}$ are not too strong. Some particular visual servoing schemes can be demonstrated to be globally asymptotically stable (*i.e.* , the robot will converge whatever its initial pose) under similar conditions.

Finally, when the visual features do not constrain all the robot degrees of freedom, it is possible to combine the visual task with supplementary tasks such as, for instance, joint limits avoidance or the visibility constraint (to be sure that the target considered will always remain in the camera field of view). In that case, the redundancy framework can be applied and the error to be regulated to zero has the following form:

$$\mathbf{e} = \widehat{\mathbf{J_s}}^{+}(\mathbf{s} - \mathbf{s}^*) + (\mathbf{I} - \widehat{\mathbf{J_s}}^{+}\widehat{\mathbf{J_s}})\ \mathbf{e}_2 \tag{9}$$

where $(\mathbf{I} - \widehat{\mathbf{J_s}}^{+}\widehat{\mathbf{J_s}})$ is a projection operator on the null space of the visual task so that the supplementary task $\mathbf{e}_2$ will be achieved at best under the constraint that the visual task is realized [Espiau et al, 1992]. A similar control scheme to (8) is now given by

$$\dot{\mathbf{q}} = -\lambda\ \mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}} \tag{10}$$

This scheme has for instance been applied for the first example depicted on Figure 4 where the rotational motion of the mobile robot is controlled by vision, while its translational motion is controlled by the odometry to move at a constant velocity.

Applications

Potential applications of visual servoing are numerous. It can be used as soon as a vision sensor is available and a task is assigned to a dynamic system to control its motion. A non-exhaustive list of examples is (see Figure 4):

- the control of a pan-tilt-zoom camera, as illustrated in Figure 1 for the pan-tilt case;
- grasping using a robot arm;
- locomotion and dexterous manipulation with a humanoid robot;
- micro or nano manipulation of MEMS or biological cells;
- pipe inspection by an underwater autonomous vehicle;
- autonomous navigation of a mobile robot in indoor or outdoor environment;
- aircraft landing;
- autonomous satellite rendezvous;
- biopsy using ultrasound probes or heart motion compensation in medical robotics.
- virtual cinematography in animation.

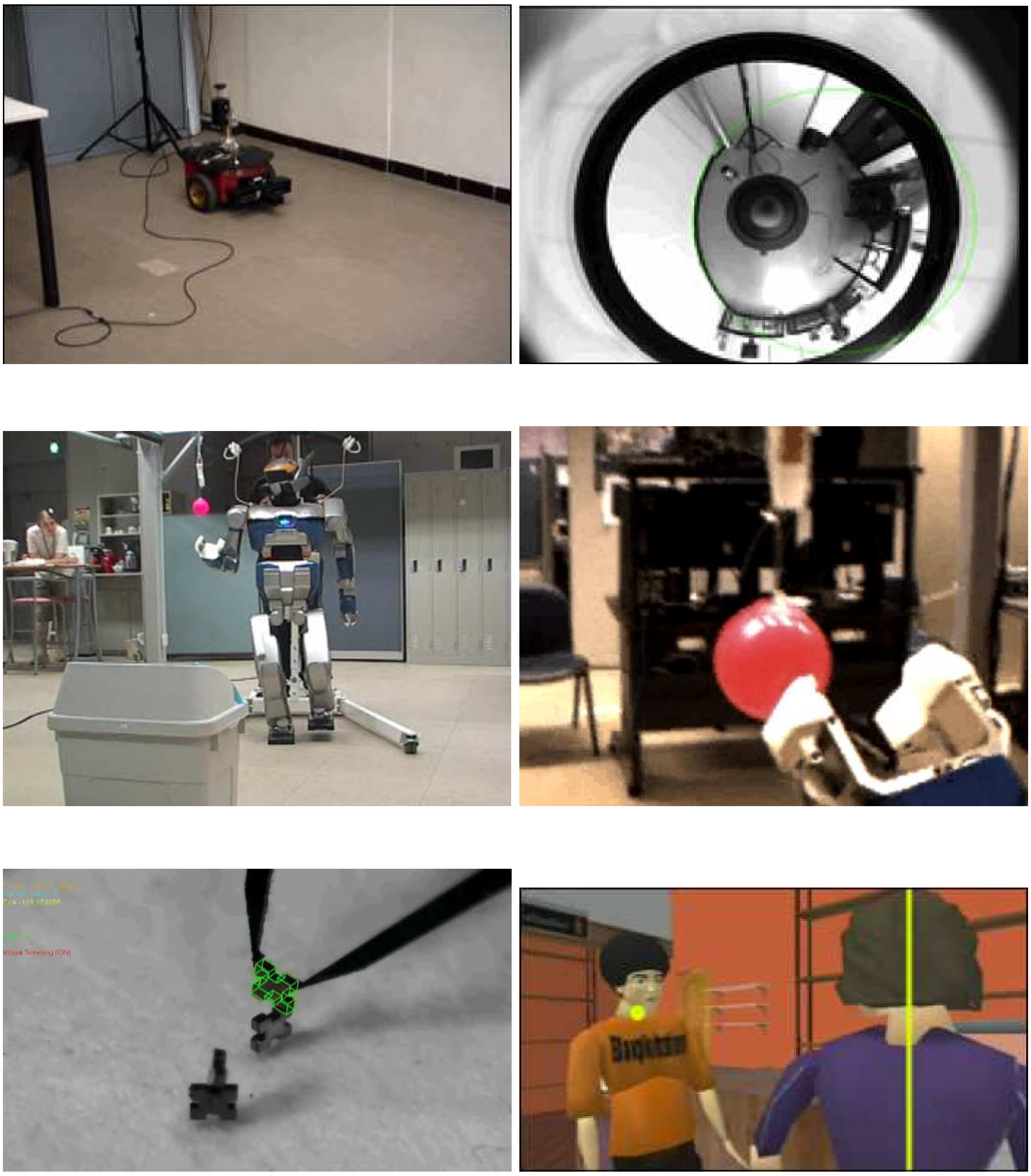

Fig. 4. Few applications of visual servoing: navigation of a mobile robot to follow a wall using an omnidirectional vision sensor (top line), grasping a ball with a humanoid robot (middle line), assembly of MEMS and film of a dialogue within the constraints of a script in animation (bottom line).

Summary and Future Directions

Visual servoing is basically a nonlinear control problem. Several modeling works have been realized to design visual features so that the control problem is transformed as much as possible to a linear control problem, leading to better stability properties. On one hand, improvements on this topic are still expected. On the other hand, the design of advanced control schemes, such as optimal control or model predictive control, is an-other way to make improvements. Then, taking into account dynamic constraints, such as nonholonomic constraints or underactuated systems, also necessitates the design of specific control laws.

Recommended Reading

In addition to the classical tutorial [Hutchinson et al, 1996] and the most recent one [Chaumette & Hutchinson, 2008], the books [Corke, 1997], [Corke 2011] and the collection of papers [Hashimoto, 1993], [Kriegman et al, 1998], and [Chesi et al, 2010] provide a good overview of past and recent works in the field. The other references below cited in text present the main pioneering contributions in visual servoing.

References

1. Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. IEEE Transactions on Robotics, 20(4):713-723.
2. Chaumette, F., Hutchinson, S. (2008). Visual servoing and visual tracking. In: Siciliano, B. and Khatib, O. (eds) Handbook of Robotics. Chapter 24, pp. 563-583, Springer, 2008.
3. Chesi, G., Hashimoto, K., Eds. (2010). Visual servoing via advanced numerical methods. LNCIS, vol. 401. Berlin: Springer.
4. Corke, P. (1997). Visual control of robots: High-performance visual servoing. New York: John Wiley & Sons.
5. Corke, P. (2011). Robotics, Vision and Control. Springer Tracts in Advanced Robotics, vol. 73 Berlin: Springer.
6. Espiau, B., Chaumette, F., Rives, P. (1992). A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation, 8(3),313–326.
7. Hashimoto, K., Ed. (1993). Visual servoing: real-time control of robot manipulators based on visual sensory feedback. Singapore: World Scientific Publishing.
8. Hutchinson, S., Hager, G., Corke, P. (1996). A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12(5), 651–670.
9. Kriegman, D., Hager, G., Morse, S., Eds. (1998) The confluence of vision and control. LNCIS, vol. 237. London: Springer.

10. Malis, E., Chaumette, F., Boudet, S. (1999). 2-1/2D visual servoing. IEEE Transactions on Robotics and Automation, 15(2), 238–250.

11. Weiss, L., Sanderson, A., Neuman, C. (1987). Dynamic sensor-based control of robots with visual feedback. IEEE Journal on Robotics and Automation, 3(5), 404–417.

12. Wilson, W., Hulls, C., Bell, G. (1996). Relative end-effector control using cartesian position-based visual servoing. IEEE Transactions on Robotics and Automation, 12(5), 684–696.