Visual Servoing

François Chaumette

Visual Servoing

François Chaumette, Inria, Univ Rennes, CNRS, IRISA

Synonyms

- Vision-based control
- Visual feedback

Definition

Visual servoing refers to the use of visual data as input of real-time closed-loop control schemes for controlling the motion of a dynamic system, a robot typically. It can be defined as sensor-based control from a vision sensor and relies on techniques from image processing, computer vision, and control theory.

Background

Basically, visual servoing consists in using the data provided by one or several cameras so that a dynamic system achieves a task specified by a set of visual constraints [12,3]. Such systems are usually robot arms or mobile robots, but can also be virtual robots, or even a virtual camera. A large variety of positioning tasks, or target tracking tasks, can be considered by controlling from one to all the degrees of freedom (DoF) of the system. Whatever the sensor configuration, which can vary from one on-board camera located on the robot end-effector to several free-standing cameras, a set of visual features has to be selected at best from the available image measurements, allowing to control the desired DoF. A control law has then to be designed so that these visual features reach a desired value, defining a correct achievement of the task. A desired planned trajectory can also be tracked. The control principle is to regulate the error between the current and desired values of the visual features to zero, or, in other terms, to minimize an objective function from which Lyapunov-based stability analysis can be performed. With a vision sensor providing 2D measurements, potential visual features are numerous, since 2D data (coordinates of particular points in the image, parameters related to geometrical shapes, intensity levels of set of pixels,...) as well as 3D data provided by a localization algorithm exploiting the extracted 2D measurements can be considered.

Typically, an iteration of the control scheme consists of the following successive 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 Fig. 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 camera 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 is mounted on a six DoF robot arm, the image measurements are the four segments that form the contour of a rectangular object. These segments are tracked in the image sequence acquired during the robot motion. The visual features are selected from the corresponding straight lines (depicted in red). The control scheme now computes the six components of the robot velocity so that these four straight lines reach particular positions (depicted in green).

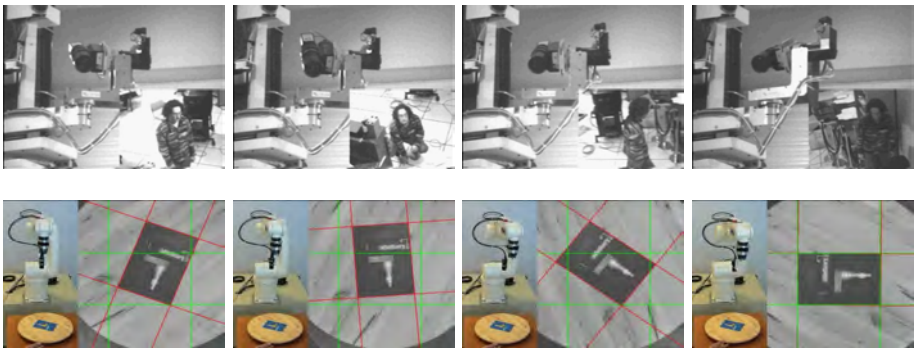

Fig. 1. Two examples of visual servoing tasks: on the top, pedestrian tracking using a pan-tilt camera; on the bottom, controlling the 6 degrees of freedom of an eye-in-hand system so that an object appears at a particular position in the image.

Theory

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

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

The parameters in (1) are defined as follows [3]: the vector $\mathbf{m}(\mathbf{r}(t))$ is a set of image measurements (*e.g.,* the image coordinates of 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, this pose varying with time t. 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 Fig. 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 [19,10], 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 [19,20]. 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 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 [13].

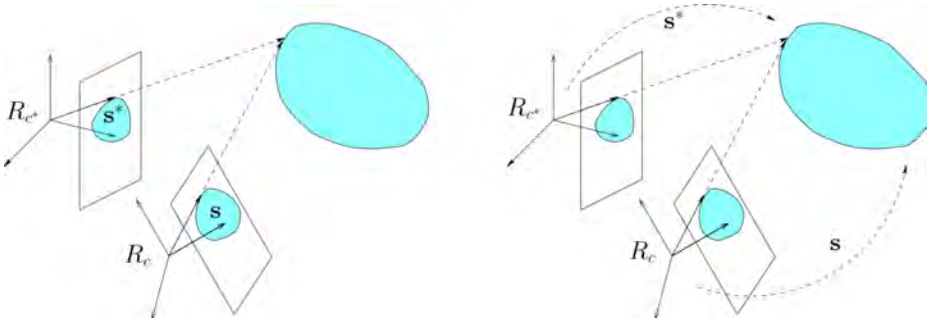

Fig. 2. If the goal is to move the camera from frame R_c to the desired frame R_{c^*}, two main approaches are possible: IBVS on the left, where the features $\mathbf{s}$ and $\mathbf{s}^*$ are expressed in the image, and PBVS on the right, where the features $\mathbf{s}$ and $\mathbf{s}^*$ are related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

The features Jacobian

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

$$\dot{\mathbf{s}} = \mathbf{J}_\mathbf{s}\, \dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \tag{2}$$

where $\mathbf{J}_\mathbf{s}$ 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 Fig. 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 Fig. 3) it represents the time variation of $\mathbf{s}$ due to a potential sensor motion.

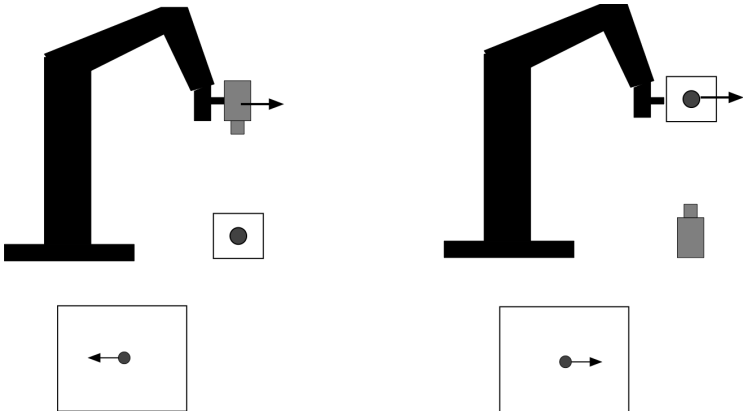

Fig. 3. In visual servoing, the vision sensor can be either mounted near the robot end-effector (eye-in-hand configuration) or outside and observing the end-effector (eye-to-hand configuration). For the same robot motion, the motion produced in the image will be opposite from one configuration to the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

As for the features Jacobian, in the eye-in-hand configuration, it can be decomposed as [3]

$$\mathbf{J_s} = \mathbf{L_s}\ {}^c\mathbf{V}_e\ \mathbf{J}(\mathbf{q}) \tag{3}$$

where

- $\mathbf{L_s}$ is the interaction matrix of $\mathbf{s}$ defined such that

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v} \tag{4}$$

  where $\mathbf{v} \in se_3$ is the relative velocity between the camera and the environment expressed in the camera frame. More details on how to determine this matrix are given below.
- ${}^c\mathbf{V}_e$ is the spatial motion transform matrix from the vision sensor to the end-effector. It is given by

$${}^c\mathbf{V}_e = \begin{bmatrix} {}^c\mathbf{R}_e & [{}^c\mathbf{t}_e]_\times\ {}^c\mathbf{R}_e \\ \mathbf{0} & {}^c\mathbf{R}_e \end{bmatrix} \tag{5}$$

  where ${}^c\mathbf{R}_e$ and ${}^c\mathbf{t}_e$ are respectively the rotation matrix and the translation vector between the sensor frame and the end-effector frame, and where $[{}^c\mathbf{t}_e]_\times$ is the skew symmetric matrix associated to ${}^c\mathbf{t}_e$. Matrix ${}^c\mathbf{V}_e$ 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 with respect to calibration errors, a coarse approximation of ${}^c\mathbf{R}_e$ and ${}^c\mathbf{t}_e$ is generally sufficient in practice to serve as a satisfactory estimation of ${}^c\mathbf{V}_e$ to be injected in the control law. If needed, an accurate estimation is possible through classical hand-eye calibration methods.
- $\mathbf{J}(\mathbf{q})$ is the robot Jacobian such that $\mathbf{v}_e = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ where $\mathbf{v}_e$ is the robot end-effector velocity.

In the eye-to-hand configuration, the features Jacobian $\mathbf{J_s}$ is composed of [3]

$$\mathbf{J_s} = -\mathbf{L_s}\ {}^c\mathbf{V}_f\ {}^f\mathbf{V}_e\ \mathbf{J}(\mathbf{q}) \tag{6}$$

where

- ${}^f\mathbf{V}_e$ 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.

The interaction matrix

A lot of works have concerned the modeling of various visual features $\mathbf{s}$ and the determination of the analytical form of their interaction matrix $\mathbf{L_s}$. 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 in the camera frame, the interaction matrix $\mathbf{L_x}$ of $\mathbf{x}$ is given by [10]

$$\mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{7}$$

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 $\theta = \mathrm{Arctan}(y/x)$, the interaction matrix of these parameters has a completely different form [3]:

$$\mathbf{L}_{\boldsymbol{\gamma}} = \begin{bmatrix} -\cos\theta/Z & -\sin\theta/Z & \rho/Z & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \\ \sin\theta/(\rho Z) & -\cos\theta/(\rho Z) & 0 & \cos\theta/\rho & \sin\theta/\rho & -1 \end{bmatrix} \tag{8}$$

This implies that using the Cartesian coordinates or the cylindrical coordinates as visual features will induce a different behavior, that is, a different trajectory of the point in the image and, consequently, a different robot trajectory. The main objective in designing a visual servoing control scheme is thus to select the best set of visual features in terms of stability, global behavior (adequate trajectories

both in the image plane and 3D space), and robustness to noise and to modeling and calibration errors from the task to be achieved, the environment observed, and the available image measurements. All these aspects can be studied from the interaction matrix of the potential visual features.

Currently, the analytical form of the interaction matrix is available for most basic features resulting from the perspective projection of simple geometrical primitives such as circles, spheres, and cylinders [10]. It is also available for image moments related to planar and almost-planar objects of any shape [2], as well as for features selected from the epipolar geometry [18] and, of course, also for coordinates of 3D points, parameters of 3D geometrical primitives, and pose parameters, assuming these features are perfectly estimated.

In the recent years, following the seminal works of [16], a new trend has concerned the use of direct image content as input of the control scheme [4]. The main objective of these works is to avoid the extraction, tracking and matching of geometrical measurements, such as points of interest or edges, so that the system is extremely accurate and robust with respect to image processing errors. The basic idea is to consider the intensity of a set of pixels as visual features ($\mathbf{s} = \mathbf{I}$). From the classical assumption in computer vision stating that the intensity level of a moving point does not change (*i.e.,* $I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{dx}, t + dt)$), it is possible to determine the interaction matrix corresponding to the intensity level of a pixel:

$$\mathbf{L}_I = -[\frac{\partial I}{\partial x} \ \frac{\partial I}{\partial y}] \, \mathbf{L}_{\mathbf{x}} \tag{9}$$

where $[\frac{\partial I}{\partial x} \ \frac{\partial I}{\partial y}]$ is the spatial gradient of the intensity along the x and y directions. Proceeding so leads to control a highly nonlinear system, with the drawback of a relatively small convergence domain, and, in general, not expected robot trajectory, plus the potential issue of robustness with respect to lighting variations. That is why the idea of direct photometric visual servoing has been expended by either considering other objective functions than $\|\mathbf{I} - \mathbf{I}^*\|$, such as the mutual information between the current and desired images [7], or other global image representations [9], or by designing photo-geometric visual features [1].

All the works mentioned above have considered a classical vision sensor modeled by a perspective projection. It is possible to generalize the approach to any sort of sensors, such as omnidirectional cameras, RGB-D sensors, the coupling between a camera and structured light, and even 2D echographic probes. A large variety of visual features is thus available for many vision sensors.

Finally, methods also exist to estimate off-line or online a numerical value of the interaction matrix, by using neural networks for instance, or the Broyden update [3]. These methods are useful when the analytical form of the interaction matrix cannot be determined, but any a priori analysis of the properties of the system is unfortunately impossible.

Control

Once the modeling step has been performed, the design of the control scheme can be quite simple for holonomic robots. The most basic control scheme has

the following form [3]

$$\dot{\mathbf{q}} = -\lambda \, \widehat{\mathbf{J}_\mathbf{s}}^{+} \, \mathbf{e} + \widehat{\mathbf{J}_\mathbf{s}}^{+} \frac{\partial \mathbf{s}^*}{\partial t} - \widehat{\mathbf{J}_\mathbf{s}}^{+} \widehat{\frac{\partial \mathbf{s}}{\partial t}} \tag{10}$$

where, in the first feedback term, $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ as defined in Eq. (1), λ is a positive (possibly varying) gain tuning the time-to-convergence of the system, and $\widehat{\mathbf{J}_\mathbf{s}}^{+}$ is the Moore-Penrose pseudoinverse 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 Eqs. (7) and (8). Methods for estimating these 3D parameters exist, either using the knowledge of the robot motion [8], or the knowledge of the 3D object model when it is available, or, up to a scalar factor, from partial pose estimation using the properties of the epipolar geometry between the current and the desired images [13].

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 serve as feedforward terms for removing the tracking error in the other cases [5].

Following the Lyapunov theory, the stability of the system can be studied [3]. 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. This is for instance the case for the pan-tilt camera control depicted on Fig. 1, for PBVS assuming the 3D parameters involved are perfectly estimated, and for well-designed IBVS schemes.

Finally, when the visual features do not constrain all the DoF, 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 new error to be regulated to zero has the following form:

$$\mathbf{e}_n = \widehat{\mathbf{J}_\mathbf{s}}^{+} \, \mathbf{e} + (\mathbf{I} - \widehat{\mathbf{J}_\mathbf{s}}^{+} \widehat{\mathbf{J}_\mathbf{s}}) \, \mathbf{e}_2 \tag{11}$$

where $(\mathbf{I} - \widehat{\mathbf{J}_\mathbf{s}}^{+} \widehat{\mathbf{J}_\mathbf{s}})$ is a projection operator on the null space of the visual task $\mathbf{e}$ so that the supplementary task $\mathbf{e}_2$ will be achieved at best under the constraint that it does not perturb the visual task. A similar control scheme to (10) is now given by

$$\dot{\mathbf{q}} = -\lambda \, \mathbf{e}_n - \widehat{\frac{\partial \mathbf{e}_n}{\partial t}} \tag{12}$$

This scheme has for instance been applied for the navigation task depicted on Fig. 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.

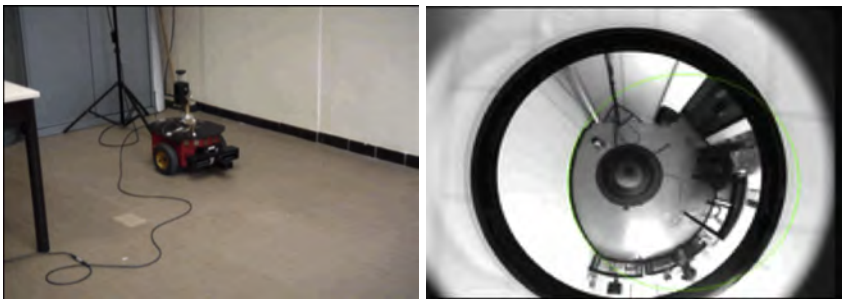

Fig. 4. Navigation of a mobile robot to follow a wall using an omnidirectional vision sensor. The 3D straight line at the bottom of the wall projects as a circle in the image (depicted in green). This circle does not move if the robot follows the wall while it moves if the robot is not correctly oriented.

Any other more advanced control strategy can be applied such as optimal control [17], coupling path planning and visual servoing [15], model predictive control or quadratic programming when visual tasks and visual constraints have to be simultaneously handled with other tasks and constraints. Particular care has to be considered for underactuated and nonholonomic systems for which adequate control laws have to be designed[11,14]

Application

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. A non-exhaustive list of examples is:

- 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.

Open problems

Visual servoing is a mature area. It is basically a nonlinear control problem for which numerous modeling works have been achieved to design visual features so that the control problem is transformed as much as possible to a linear control problem. On one hand, improvements on this topic are still expected for instantiating this general approach to particular applications. On the other hand, designing new control strategies is another direction for improvements, especially when supplementary data coming from other sensors (force, tactile, proximity sensors) are available. Finally, the current expansion of deep learning may rejuvenate the field especially for the dense direct methods that use the same input and end-to-end approach.

Cross References

- Active Sensor (Eye) Movement Control
- Hand-Eye Coordination
- Camera Pose
- Hand-Eye Calibration
- Omnidirectional Vision
- Pan-Tilt-Zoom Camera

References

[1] Bakthavatchalam M, Tahri O, Chaumette F (2018). A direct dense visual servoing approach using photometric moments. IEEE Transactions on Robotics, 34(5):1226–1239.

[2] Chaumette F (2004). Image moments: a general and useful set of features for visual servoing. IEEE Transactions on Robotics, 20(4):713–723.

[3] Chaumette F, Hutchinson S, Corke P (2016). Visual servoing. In Handbook of Robotics, 2nd edition, Chapter 34, pp. 841–866, Springer.

[4] Collewet C, Marchand E (2011). Photometric visual servoing. IEEE Transactions on Robotics, 27(4):828–834

[5] Corke P, Good M (1996). Dynamic effects in visual closed-loop systems. IEEE Transactions on Robotics and Automation, 12(5):671–683.

[6] Crombez N, Mouaddib EM, Caron G, Chaumette F (2019). Visual servoing with photometric Gaussian mixtures as dense features. IEEE Transactions on Robotics, 35(1):49–63.

[7] Dame A, Marchand E (2011). Mutual information-based visual servoing. IEEE Transactions on Robotics, 27(5):958–969.

[8] De Luca A, Oriolo G, Robuffo Giordano P (2008). Feature depth observation for image-based visual servoing: Theory and experiments, International Journal of Robotics Research, 38(4):422–450. 27(10):1093–1116.

[9] Duflot LA, Reisenhofer R, Tamadazte B, Andreff N, Krupa A (2019). Wavelet and shearlet-baed image representations for visual servoing, International Journal of Robotics Research, 38(4):422-450.

[10] 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.

[11] Hamel T, Mahony R (2002). Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach. IEEE Transactions on Robotics and Automation, 18(2):187–198.
[12] Hutchinson S, Hager G, Corke P (1996). A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12(5): 651–670.
[13] Malis E, Chaumette F (2000). 2-1/2D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. International Journal of Computer Vision, 37(1):79–97.
[14] Mariottini GL, Oriolo G, Prattichizo D (2007). Image-based visual servoing for nonholonomic mobile robots using epipolar geometry, IEEE Transactions on Robotics, 23(1):87–100.
[15] Mezouar Y, Chaumette F (2002). Path planning for robust image-based control. IEEE Transactions on Robotics, 22(10):781–804.
[16] Nayar S, Nene S, Murase H (1996). Subspace methods for robot vision. IEEE Transactions on Robotics and Automation, 12(5):750–758.
[17] Nelson B, Khosla P (1995). Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance. International Journal of Robotics Research, 14(3):225-269.
[18] Silveira G, Malis E (2012). Direct visual servoing: Vision-based estimation and control using only nonmetric information. IEEE Transactions on Robotics, 28(4):974–980.
[19] 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.
[20] 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.