A brief introduction to visual servoing

Gabino de Diego Salas

***Abstract*—This article describes the fundamentals of visual servoing. The first part is dedicated to the theoretical concepts and the mathematical formulae which applies. Then, a real task is solved using two vision control techniques: Position-based and image-based control, which are considered to be the most popular. This paper studies them in tasks of alignment and placing.**

I. Introduction

The presence of the robots is a fact nowadays. In most environments, there are machines in charge of tasks that previously were developed by humans. The advance in robotic technology has permitted more accuracy and resources savings when executing the task assigned. Such advance, together with the improvement and the introduction of sensors in the robots, makes them better at work because provides a better control in robotic actions.

Vision could be considered as one of the sensors attached to a robot. It is very useful to collect information related to the environment of the machine. When firstly introduced, the algorithm used was *looking and moving*. Thus, the result of the movement in terms of accuracy, depends directly on the accuracy of the sensor. Today, in order to increase the performance of the robot operations, some feedback has been introduced in the interaction between the visual sensor and the robot joints. This yields a closed-loop position control for robot end-effector. This is referred to as visual servoing.

Visual servoing is defined[1] as the operation of controlling a robot to manipulate its environment using vision. It is, thus, opposed to observe the world passively or actively. In order to implement a visual servoing system, knowledge in many areas is needed. Concepts related to image processing, kinematics, dynamics, control theory and real-time computing must be well known to be able to understand (and produce) a robot with this kind of control.

The remainder of the article is structured as follows: Firstly, a description explaining the variety of techniques used for visual servoing. Secondly, a brief introduction to how those techniques could be applied and implemented in real robot systems and the results obtained. The last section is dedicated to summarize and conclude the article.

II. Description

This section is dedicated to clarify some concepts related to motion tracking and visual servoing. The first part describes basic concepts which should be clear when first approaching to this technology. After the explanation of those concepts, an overview of servoing systems is provided.

A. Basic concepts

The control of the motion based in the information collected (specially using vision) is a problem that requieres knowledge in several disciplines[1]. Here, the principles of coordinates and velocity calculation are briefly described: Images formation and robot/camera configuration.

1) Coordinate transformation: Firstly, to introduce the coordinate transformation, it is necessary to define what the task space of a robot is. Formally, it is the set of positions and orientations that the robot tool can attain. Hence, that space is the region of the world that could be touched and watched by the robot.

When a robot needs to realize a task, it is necessary to use one or more coordinate frames. The robot may supply information regarding the spatial location of the object being tracked and may also have to provide information to some end-effector to produce an action. Because of the difference in the position of the two mechanism (camera and end-effector), they both need the information relative to their own coordinates, and thus, shall be translated into the corresponding geometry.

In mathematical language[1], a point P with respect to a coordinate frame x is denoted by ${}^{x}P$. Given two frames, x and y, it would be interesting to locate P in y. To make the transformation, a rotation matrix is defined. ${}^{x}R_{y}$ represents the orientation of the frame y with respect to the frame x. The frames could also be different in the origin. Thus, the vector ${}^{x}t_{y}$ represents the location of the frame y with respect to the frame x. Finally, the pose of a frame is defined as its position and orientation, and denoted by ${}^{x}x_{y} = ({}^{x}R_{y},{}^{x}t_{y})$. If x is not specified, the world coordinate frame is assumed.

As mentioned before, there is a need to translate between the coordinate frame of the visual sensor and the end-effector one. This transformation is realized as follows: Given a ${}^{y}P$ (the coordinates of a point P relative to frame y) and a ${}^{x}x_{y} = ({}^{x}R_{y},{}^{x}t_{y})$, ${}^{x}P$ can be obtianed:

$$^{x}P =^{x} R_{y}^{y}P +^{x} t_{y} \tag{1}$$

$$=^{x} x_{y} o^{y} P \tag{2}$$

2) Velocity of a rigid object: Another important aspect of motion tracking and visual servoing is velocity estimation of the objects in the workspace. For example, it would be interesting to know how the end-effector of a robot is moving.

The motion of an object could be separated in two components: Angular and translational velocity. The former is represented by $\Omega(t) = [\omega_x(t), \omega_y(t), \omega_z(t)]^T$ and the latter is denoted by $T(t) = [T_x(t), T_y(t), T_z(t)]^T$. Given a rigid point P, in order to calculate its derivatives of the coordinates (respect to base coordinates), the following relations should be applied:

$$\dot{P} = \Omega \times P + T \tag{3}$$

Both T and Ω define the velocity screw (velocity variation) of the robot moving parts:

$$\dot{r} = \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4}$$

As occurred with the position, for some applications, it is also interesting to translate the motion of an object into another coordinate frame. Given the velocity, for example, of the end-effector in its coordinates ${}^{e}\dot{r} = [{}^{e}T;{}^{e}\Omega]$, the equivalent expression in base coordinates could be as follows:

$$\dot{r} = \begin{bmatrix} \Omega \\ T \end{bmatrix} = \begin{bmatrix} R_e^e\Omega \\ R_e^eT -^e \Omega \times T \end{bmatrix} \tag{5}$$

B. Image formation

On motion tracking, one of the inputs of the control system of the robot is the information provided by the vision system. In order to use such information efficiently, knowledge of imaging formation is required.

The cameras of the system are in charge of the task of capturing the information of the environment and process it in oder to extract as useful information as possible. Each camera contains a lens that forms 2D projection of the scene on the image plane where the sensor is located. However, the depth information is lost when projecting[1]. There are several techniques to infer this parameter based on the use of more than one camera or using the additional knowledge related to the geometry of the object being tracked.

Basically, there are three methods to obtain the projection of a 3D object in the plane defined by the lens of the camera. Assuming λ to be the focal distance and given a point ${}^{c}P = [x, y, z]^T$, the results using the different methods are the following.

- **Perspective projection**: Using this technique to model the projection $p = [u, v]^T$ of the point P, the result obtained is:

$$\pi(x, y, z) = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{\lambda}{z} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6}$$

- **Scaled orthographic projection**: This technique is used as an alternative of the perspective projection. Due to the nonlinear mapping of the latter, it introduces more complexity. However, many cases could be approximated easier by linear mapping. Therefore, the result obtained is:

$$\begin{bmatrix} u \\ v \end{bmatrix} = s \begin{bmatrix} x \\ y \end{bmatrix} \tag{7}$$

  where s is a fixed scale factor.

- **Affine Projection**: This technique was produced as a result of generalizing the former method. The result is the following:

$$\begin{bmatrix} u \\ v \end{bmatrix} = A^{c}P + c \tag{8}$$

  where A is an arbitrary 2×3 matrix and c an arbitrary vector.

An important concept of image formation is the image feature[1]. It is defined as any structural feature that can be extracted from an image (e. g., an edge or a corner). From image features, image feature parameters could be extracted. The latter are any real-valued quantity that can be calculated from one or more image features. The main characteristic of a feature is that it could be located with no ambiguity in different views of the scene. Such parameters are identified in order to use them in servo control tasks to improve alignment and to reduce the possible error in position estimations.

Finally, to conclude with this section, it is interesting to outline one critical problem in robotics: Camera placing[2]. It determines the quality of the information that can be collected from the formed image. The position of the camera should be studied carefully in order to capture the most helpful images so as to complete the task.

Basically, depending on where the camera is placed, two different configurations are possible: Eye-in-hand systems and fixed camera systems. In the eye-in-hand systems, the camera is mounted on the robot's end-effector. The consequence is that there is relationship between the pose of the camera and the one of the end-effector. This relationship is known because both systems move in the same manner. On the other hand, in the fixed camera configuration, the image of the target is independent of the robot motion.

C. Robot configuration for motion tracking

The next section is dedicated to take an overview of the different kind of motion tracking systems[1]. They differ in the manner they perform the servo control and how they process the information of the environment.

A first classification of the systems can be done according to the control architecture used, since the visual system is used to provide set-points inputs to the join-level controller (in charge of the servo movements). The main advantage is that offers the possibility of use the own feedback (together with the inputs) to internally stabilize the robot. On the other hand, the direct visual servo systems use a visual servo controller, which uses only the visual information to stabilize. Actually, almost all motion trackers use a joint-level controller (thus the first architecture).

The second major classification of systems distinguishes position-based control from image-based control. The former tries to estimate the pose of the target using information from the image features together with a geometric model of the target and the camera model. Image-based controller, however, only uses the image features to track the object.

The main advantage is the computing time reduction but the image feature extraction should be very accurate. Some authors include here another system: 2 1/2 D visual servo systems, which could be described as a combination of the presented systems: Image and pose are used in order to reduce error in location estimation.

Finally, to end with this section, there is a third criteria to classify the servo architectures: Endpoint open-loop (EOL) and endpoint closed-loop (ECL). The difference between them is that ECL systems observe the target position and the location of the end-effector. On the contrary, EOL systems must operate only with the information of target coordinates.

III. Experiments

The following section consists of applying the previous concepts to a real problem. Two servoing architectures has been chosen: Position-based and image-based control, in order to use them in a robot, dedicated to manipulation tasks. The goals and the problems encountered are described in this section.

A. Position-based control

Position-based visual servoing[2] is normally referred to as 3D servoing control since image measurements are used to determine the pose of the target with respect to a camera or some common world frame.

There are two main architectures when using a position-based control system. First of them, the use of a mobile camera which controls its own position in order to reach the desired object. Normally, this kind of cameras are attached to a manipulator. The second possibility is a fixed camera which tracks a moving object, manipulated by an end-effector controlled by the information provided by the camera. Thus, it is necessary to know the transformation between the coordinate systems of both frames.

Therefore, position-based control systems need additional information in order to complete its task.

- A model of the target is required to estimate the pose.
- Calibration is needed for an accurate positioning and for an estimation of the desired velocity screw of the robot.

The required knowledge of the two parameters is the main disadvantage of this systems.

- **Align and track**: The aim is to demonstrate how the robot is capable of aligning its end-effector to a predefined reference position with respect to the target object. The goal is to achieve it whenever starting from an arbitrary position.
  In order to complete the task assigned successfully, the velocity screw of the robot needs to be estimated: $\dot{r} = Ke$, defined in the end-effector coordinate frame. The error matrix is defined as follows
  $$\mathbf{e} = \begin{bmatrix} \Delta^R t_G \\ \Delta^R \theta_G \end{bmatrix} \tag{9}$$
  where $\Delta^R \theta_G$ and $\Delta^R \theta_G$ are the translational and the rotational differentials that separate the actual point of the end-effector and the desired point.
  Thus, the changes in the velocity screw are dependent on the translational and rotational moves needed to be done. Hence, starting at the definition of $\Delta^R t_G =^R t_G +^R t^*_G$ and $\Delta^R \theta_G =^R \theta_G +^R \theta^*_G$ and applying the necessary transformations using the translating matrices[1] for translational and the rotational movement, the algorithm ends when the error is reduced to cero.

In applications like the one presented, the main advantage is that the camera (and the robot) trajectory is controlled directly in the cartesian coordinates, which allows easier trajectory planning. However, especially in the case of eye-in-hand camera configurations, the information must be extracted only from the image, and thus an accurate calibration of the camera is needed. This limitation could be reduced (even eliminated) implementing an ECL system.

B. Image-based control

Image-based visual servoing moves image plane features f^c to a set of desired locations f^*, based on robot velocity screw estimation $\dot{r}$. Image-based visual servoing control involves the computation of the image Jacobian[1] or the interaction matrix, which represents the differential relationship between the scene frame and the camera frame (where either the scene or the camera frame is usually attached to the robot).

The test made in the laboratory with this servoing control technique[2] was developed with a robot equipped with a binocular camera, which provided two images of the scene. Several tasks were ask to be done. They were considered to be completed when the difference $e(f) = f^c - f^*$ is cero (the desired position match with the image feature):

- Insertion: The objetive is to place a screwdriver in a hole with a diameter which is approximately 5mm. According to the image provided by the camera, $e_l(f) = f_l - f^*_l$ and $e_r(f) = f_r - f^*_r$ should be minimized to complete the task with success. Due to the vision system, the image Jacobian presents four rows and six columns. The reason is that there is a correlation between the two images, since y coordinates are the same in both images. This problem is solved assuming that no rotation movement is required and only translational degrees of freedom have to be controlled (the plane of the screwdriver and the hole are ortogonal). Thus, velocity screw is estimated. Since the relationship between the end-effector and the tip of the screwdriver remains constant, this is an example of an endpoint closed loop system.
- Grasping: When executing this task, the control is generated in the same manner (only translational movement). Again, this is an example of a endpoint closed loop system.
- Placing: The third task consists of the alignment of the wheels of a toy car with the road. Here, two task

[1] The pose between the camera and the robot is estimated off-line

are simultaneously performed: Point-to-line and point-to-point positioning. The algorithm tries to situate one of the four wheels in the road (point-to-point positioning): A point in the road is selected and the first task is to align it with the wheel. Then, using a rotational movement, the remaining wheels (of the other axle) are placed aligned with the road.

IV. Conclusions and future work

Visual servoing techniques have been presented during this article. Especially, it was focused on position-based and image-based systems. They mainly differ in terms of information used to produce movement. The latter only uses the information provided by the image obtained with sensors. The position-based system, on the contrary, uses also target and camera models in order to estimate the next step of the end-effector. Normally, image-based control is used in servoing tasks and motion tracking because is easier to implement and eliminates the need of creating the models and reduces the time of computing (less processes should be performed). It is also independent of calibration errors. However, its main weaknesses are no-linearity in captured images and image features correlation.

Taking a look at the future in visual servoing and motion traking, there are many interesting researches running. For example, issues like application of neural networks in control loops, camera calibration and positioning problem, as well as many other interesting lines are being investigated.

References

[1] *A tutorial on visual servo control*. Seth Hutchinson, Greg Hager, Peter Corke. *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp. 615-670. October 1996.

[2] *Survey on visual servoing for manipulation*. Danica Kragic and Henri I Christensen. Technical report ISRN KTH/NA/P–02/01–SE. CVAP259. University of Stockholm. January 2002.

[3] *Complex object tracking by visual servoing based on 2D image motion*. Armel Crétual, Franois Chaumette, Patrick Bouthemy. In *IAPR International Conference on Pattern Recognition. ICPR98*. Volume 2 - Pages 1251-1254 - Brisbane - Australia - August 1998.

[4] *Robust image-based visual servoing system using a redundant architecture*. Rafael Aracil, N. García, C. Pérez, L. Payé, L. M. Jiménez, O. Reinoso. In *16 IFAC World Congress*. Prague 2005.