VISUAL SERVO CONTROL

Robotics Planning and Navigation

What is Visual Servo?

- Visual servo (VS) control refers to the use of **computer vision data** to **control the motion** of a robot.
- Vision data can we acquired from:
  - Camera mounted on a manipulator.
  - Camera fixed in a scene.

In this lecture we will focus primarily on the former, so-called eye-in-hand case.

Basic Components

The aim of all visual servo algorithms is to minimize error *e(t)* defined by

$$\boxed{\boldsymbol{e(t) = s(m(t), a) - s^{*}}}$$

m(t) - image measurements

a - additional camera information.

s(m(t); a) – visual features

*s** - the desired values of the features

Classification of Visual Servo

1. IBVS (Image-based visual servoing)
    - Error is computed directly on the values of the features extracted on the 2D image plane, without going through a 3D reconstruction.
    - ***s*** consists of a set of features that are immediately available in the image
2. PBVS (Position-based visual servoing)
    - Information extracted from images (features) is used to reconstruct the current 3D pose (pose/orientation) of an object.
    - ***s*** consists of a pose, which must be estimated from image measurements.

Velocity Controller

$$\boxed{\boldsymbol{s' = L_s v_c}}$$

$\boldsymbol{v}_c = (v_c, w_c)$ - the spatial velocity of the camera.

s' - time variation of the features.

L_s - Interaction Matrix

Interaction Matrix calculation (IBVS)

Projection of 3D world-point to image plane with normalized coordinates:

$$\begin{cases} x &= X/Z = (u - c_u)/f\alpha \\ y &= Y/Z = (v - c_v)/f \end{cases} \quad \text{eq.1}$$

X = *(X; Y; Z)* – 3D world point,

x = *(x; y)* – normalized image coordinates

(u, v) - coordinates of the image point expressed in pixel units

(c_u, c_v) - coordinates of the principal point,

f - focal length

α - ratio of the pixel dimensions.

Camera velocity: $\boldsymbol{v_c} = (v_x; v_y; v_z)$ and $\boldsymbol{w_c} = (w_x, w_y, w_z)$.

Taking derivative of the projection equation:

$$\begin{cases} \dot{x} &= \dot{X}/Z - X\dot{Z}/Z^2 = (\dot{X} - x\dot{Z})/Z \\ \dot{y} &= \dot{Y}/Z - Y\dot{Z}/Z^2 = (\dot{Y} - y\dot{Z})/Z \end{cases} \quad \text{eq.2}$$

We can relate the velocity of the 3-D point to the camera spatial velocity using the well-known equation:

$$\dot{\boldsymbol{X}} = -\boldsymbol{v}_c - \omega_c \times \boldsymbol{X} \Leftrightarrow \begin{cases} \dot{X} = -v_x - \omega_y Z + \omega_z Y \\ \dot{Y} = -v_y - \omega_z X + \omega_x Z \\ \dot{Z} = -v_z - \omega_x Y + \omega_y X \end{cases} \quad \text{eq.3}$$

Inserting eq.3 into eq.2, and grouping terms.

$$\begin{cases} \dot{x} = -v_x/Z + xv_z/Z + xy\omega_x - (1 + x^2)\omega_y + y\omega_z \\ \dot{y} = -v_y/Z + yv_z/Z + (1 + y^2)\omega_x - xy\omega_y - x\omega_z \end{cases}$$

Which can be written as: $\dot{\boldsymbol{x}} = \mathbf{L}_{\boldsymbol{x}} \boldsymbol{v}_c$

$$\mathbf{L}_{\boldsymbol{x}} = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix}$$

Interaction Matrix calculation (IBVS)

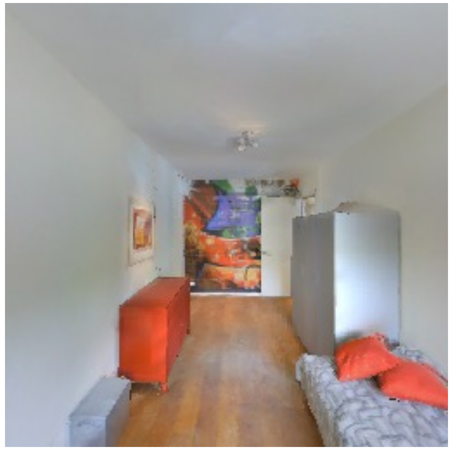
In the matrix L_x, the value Z is the depth of the point relative to the camera frame. Therefore, any control scheme that uses this form of the interaction matrix must estimate or approximate the value of **Z.**

Similarly, the camera intrinsic parameters are involved in the computation of x and y.

$$\mathbf{L}_{\boldsymbol{x}} = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix}$$

Example of IBVS

Initial Image



Target Image