VISUAL SERVOING
(IBVS)

- Lecture Notes: Teresa Vidal Calleja

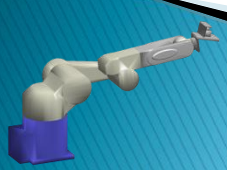


Image Based VS (IBVS)

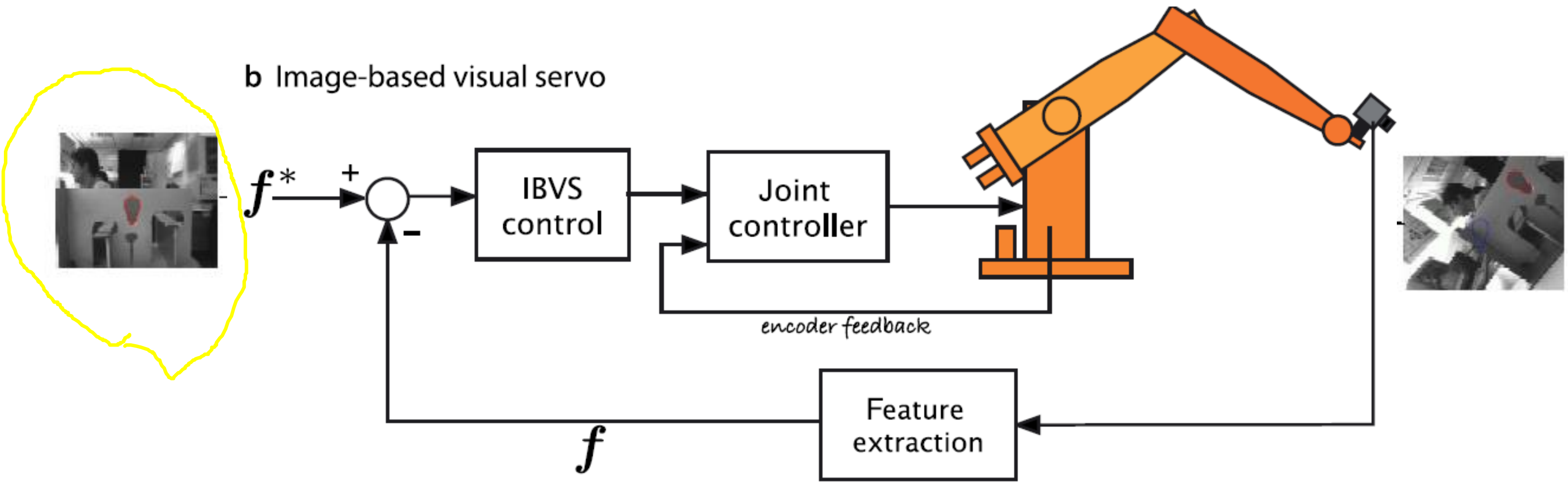
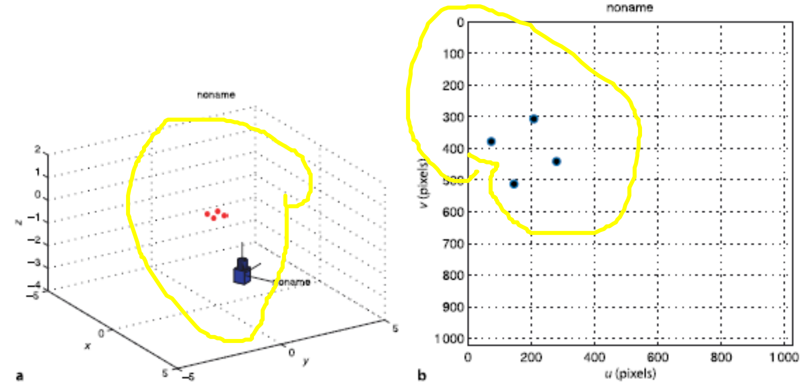
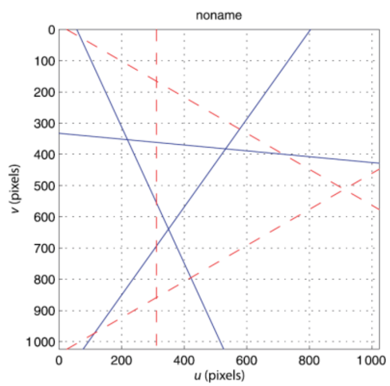


Image Features

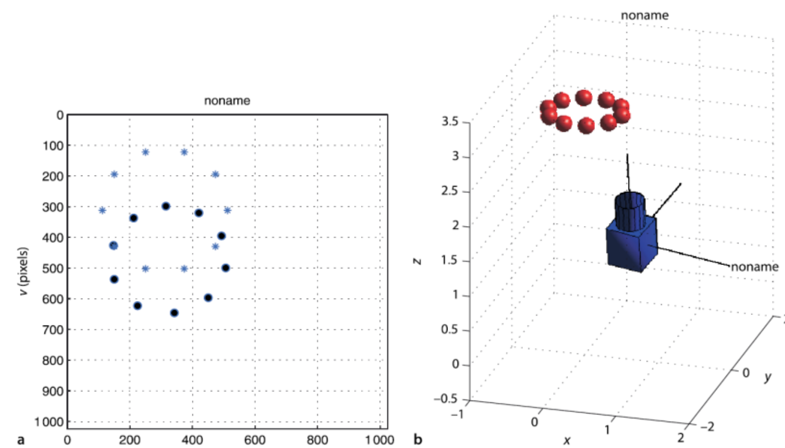
- Points



- Lines

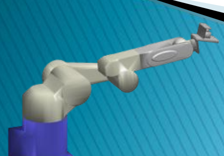


- Circles



IBVS

- Control problem is expressed in image coordinates
- The task is to move the image features $\boldsymbol{p}$ to a desire position $\boldsymbol{p}^*$
- Moving image features implicitly changes the pose



IBVS

- From the perspective projection model

$$\boldsymbol{p} = \mathcal{P}(\boldsymbol{P}, \boldsymbol{K}, \xi_C)$$

derivative wrt ξ

- Image feature's motion model

$$\boxed{\dot{\boldsymbol{p}} = J_p(\boldsymbol{P}, \boldsymbol{K}, \xi_C)\boldsymbol{\nu}}$$

Image Jacobian

(Interaction Matrix)

$$\boldsymbol{\nu} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$$

$$\boldsymbol{\nu} = (v, \omega)$$

- How do we move the camera to take image features to a desire image position?

IBVS

- Given the 3D point in camera frame $\boldsymbol{P} = (X, Y, Z)$

$$\dot{\boldsymbol{P}} = -\boldsymbol{\omega} \times \boldsymbol{P} - \boldsymbol{v}$$

$$\begin{aligned} \dot{X} &= Y\omega_z - Z\omega_y - v_x \\ \dot{Y} &= Z\omega_x - X\omega_z - v_y \\ \dot{Z} &= X\omega_y - Y\omega_x - v_z \end{aligned}$$

- Projecting into the image

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

IBVS

- Through the normalised image-plane coord.

$$u = \frac{f}{\rho_u} x + u_0,\ v = \frac{f}{\rho_v} y + v_0$$

$$x = \frac{\rho_u}{f}\bar{u},\ y = \frac{\rho_v}{f}\bar{v} \qquad \dot{x} = \frac{\rho_u}{f}\dot{\bar{u}},\ \dot{y} = \frac{\rho_v}{f}\dot{\bar{v}}$$

$$\boxed{\dot{\boldsymbol{p}} = \boldsymbol{J}_p(\boldsymbol{P}, \boldsymbol{K}, \xi_C)\boldsymbol{\nu}}$$

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{v}} \end{pmatrix} = \underbrace{\begin{pmatrix} -\frac{f}{\rho_u Z} & 0 & \frac{\bar{u}}{Z} & \frac{\rho_u \bar{u}\bar{v}}{f} & -\frac{f^2+\rho_u^2\bar{u}^2}{\rho_u f} & \bar{v} \\ 0 & -\frac{f}{\rho_v Z} & \frac{\bar{v}}{Z} & \frac{f^2+\rho_v^2\bar{v}^2}{\rho_v f} & -\frac{\rho_v \bar{u}\bar{v}}{f} & -\bar{u} \end{pmatrix}}_{J_p} \begin{pmatrix} \nu_x \\ \nu_y \\ \nu_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

Image Jacobian

IBVS Control Law

Motion model for 1 point

$$\begin{pmatrix} \dot{u}_1 \\ \dot{v}_1 \end{pmatrix} = J_{p_1} \nu$$

- Camera velocity control

$$\nu = \lambda \begin{pmatrix} J_1 \\ \vdots \\ J_N \end{pmatrix}^{+} (p^* - p)$$

Motion model for 2 points

$$\begin{pmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \end{pmatrix} = \begin{pmatrix} J_{p_1} \\ J_{p_2} \end{pmatrix} \nu$$

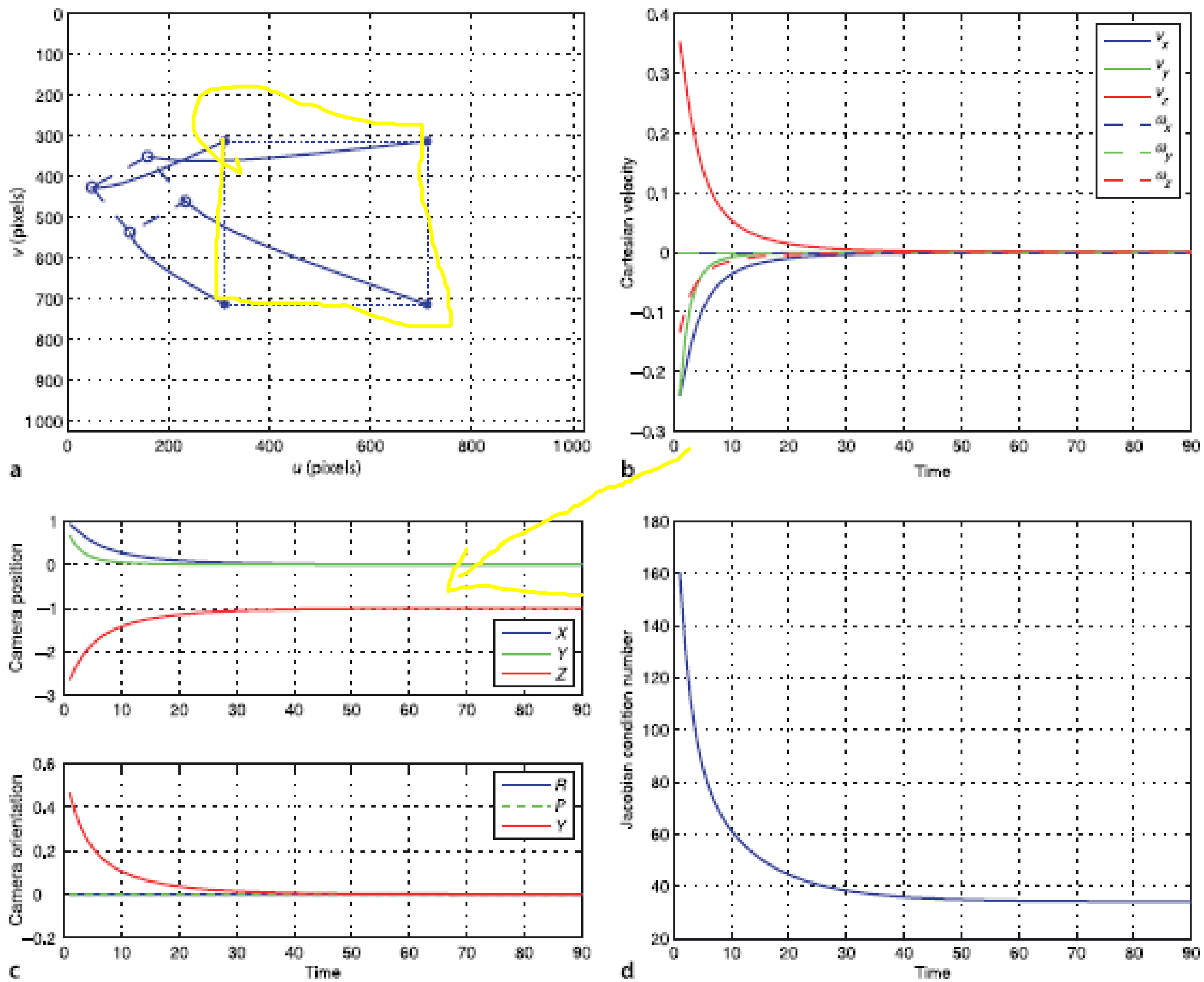
- Next camera pose $\xi_C\langle k+1\rangle = \xi_C\langle k\rangle \oplus \Delta^{-1}(\nu\langle k\rangle)$

- Robot joint velocities

$$\dot{q} = J(q)^{-1} \nu$$

Robot Jacobian

IBVS Example



IBVS Remarks

- Positioning accuracy of the system is less sensitive to camera calibration errors

- A depth estimation or approximation is necessary in the design of control law

- The convergence is ensured only in a neighborhood of desired position

d2D/dt Visual Servoing

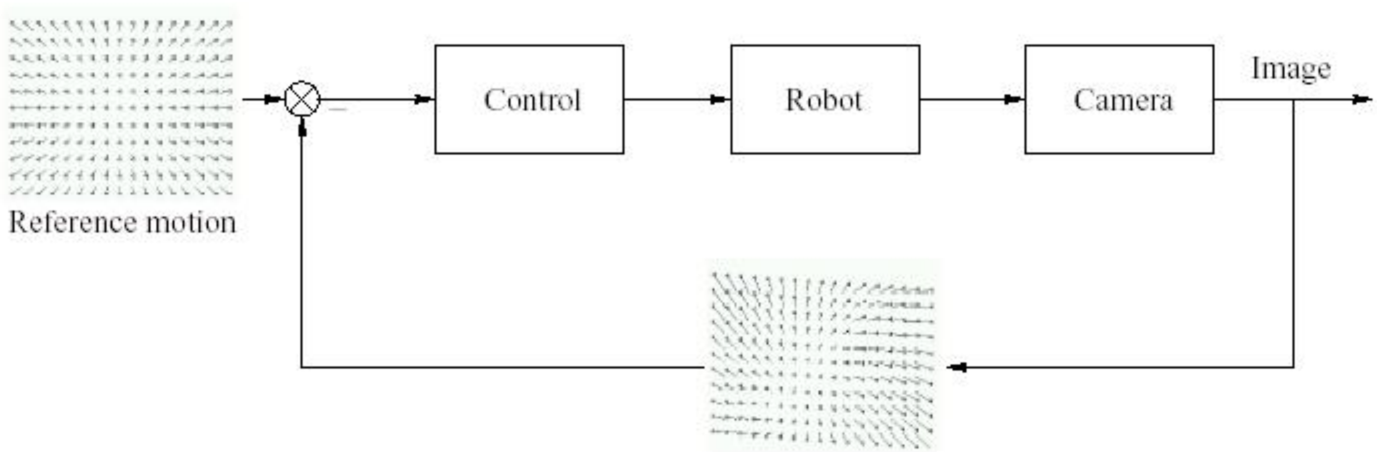


Figure 10: $\frac{d2D}{dt}$ visual servoing