An Introduction to Visual Servo Control

with application to robotic manipulators

Dr. George C. Karras and
Prof. Kostas J. Kyriakopoulos
Control Systems Lab,
School of Mechanical Engineering,
National Technical University of Athens

What is Visual Servo Control ?

Definition:

"Visual Servo Control is the use of computer vision data to control the motion of a robot"

The vision data may be acquired:

•From a camera that is mounted directly on a robot manipulator or on a mobile robot: Eye - in hand camera system.

•The camera can be fixed in the workspace so that it can observe the robot motion from a stationary configuration.

•Several cameras mounted on pan-tilt heads observing the robot motion.

Visual Servo Control is a blend of:

✓**Image Processing**
✓**Computer Vision**

Feature Extraction
(Out of the Scope of this course)

✓**Robotics Theory**
✓**Control Theory**

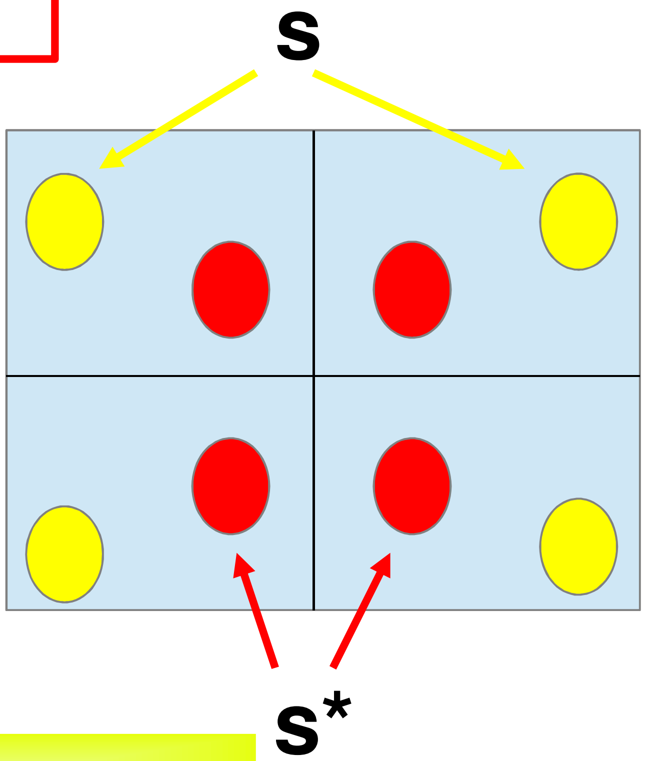
Design of control laws for robotic manipulators

The Basic Components of Visual Servoing

The aim of all vision-based control schemes is to minimize an error e(t), which is typically defined by:

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

- **The vector m(t) is a set of image measurements** (e.g., the image coordinates of interest points or the image coordinates of the centroid of an object).
- **These image measurements are used to compute a vector of k visual features, s(m(t), a).**
- **a is a set of parameters that represent potential additional knowledge about the system** (e.g., coarse camera intrinsic parameters or 3-D models of objects).
- **The vector s* contains the desired values of the features.**



NOTES:

1. **Fixed goal pose:** s* is constant over time
2. **Motionless target:** changes in **s** depend only on camera motion
3. **Camera with six degrees of freedom** (6 DOF)

The Basic Components of Visual Servoing

Example:

Consider a camera with the following parameters:

- **Focal length** f
- **Pixel size** [ax, ay]
- **Principal point** [cu, cv]
- **Image size** (resolution): width x height

→ **CAMERA INTRINSIC PARAMETERS**

Suppose we have a target which consists of four points that form a square of side length 0.5 m that lies in the xy somewhere in the **world frame**

→ **OBJECT 3D MODEL**

vector a

If we place the camera at :

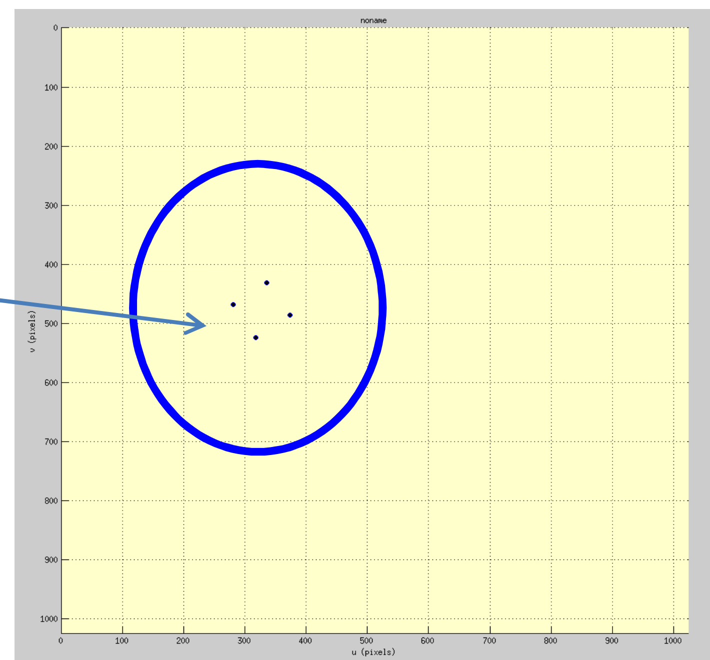
$$\mathbf{t}_{\mathbf{c}_0} = \begin{bmatrix} 1 & 1 & -3 \end{bmatrix}^T$$

$$\mathbf{r}_{\mathbf{c}_0} = \begin{bmatrix} 0 & 0 & 0.6 \end{bmatrix}^T$$

wrt to the target, the four points we will lay inside the image plane:

$$\mathbf{p}_0 = \begin{bmatrix} \mathbf{p}_{0_1} & \mathbf{p}_{0_2} & \mathbf{p}_{0_3} & \mathbf{p}_{0_4} \end{bmatrix}^{\mathbf{T}}$$

$$\mathbf{p}_0 = \begin{bmatrix} 280 & 318 & 373 & 335 \\ 469 & 524 & 486 & 431 \end{bmatrix}$$



The Basic Components of Visual Servoing

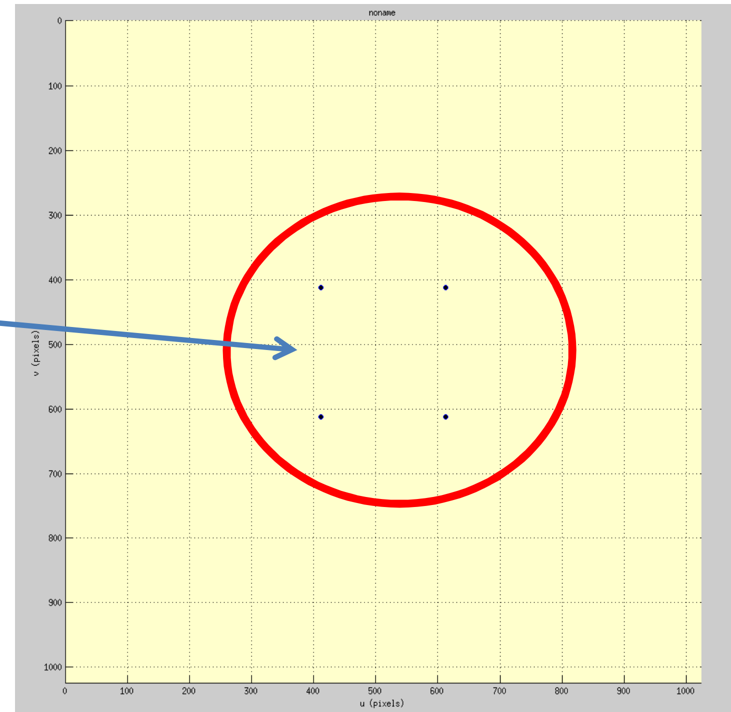
Example (continue):

If want our visual servoing scheme to place the camera at:

$$\mathbf{t}^{*}_{\mathbf{c}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{T}$$

$$\mathbf{r}^{*}_{\mathbf{c}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{T}$$

Or equivalently place the target at the centre of the camera/image



This means that the four features of the target will be actually placed at the following positions inside the image space:

$$\mathbf{p}^{*}_{c} = \begin{bmatrix} 412 & 412 & 612 & 612 \\ 412 & 612 & 612 & 412 \end{bmatrix}$$

We can define the following errors:

Cartesian Space:

$$\mathbf{e}_{\mathbf{t}_{\mathbf{c}}} = \mathbf{t}_{\mathbf{c}_0} - \mathbf{t}^{*}_{\mathbf{c}}$$

$$\mathbf{e}_{\mathbf{t}_{\mathbf{c}}} = \begin{bmatrix} 1 & 1 & -3 \end{bmatrix}^{T} - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{T}$$

$$\mathbf{e}_{\mathbf{t}_{\mathbf{c}}} = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix}^{T}$$

$$\mathbf{e}_{\mathbf{r}_{\mathbf{c}}} = \mathbf{r}_{\mathbf{c}_0} - \mathbf{r}^{*}_{\mathbf{c}}$$

$$\mathbf{e}_{\mathbf{r}_{\mathbf{c}}} = \begin{bmatrix} 0 & 0 & 0.6 \end{bmatrix}^{T} - \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{T}$$

$$\mathbf{e}_{\mathbf{r}_{\mathbf{c}}} = \begin{bmatrix} 0 & 0 & 0.6 \end{bmatrix}^{T}$$

Image Space:

$$\mathbf{e}_{\mathbf{p}} = \mathbf{p}_0 - \mathbf{p}^{*}$$

$$\mathbf{e}_{\mathbf{p}} = \begin{bmatrix} -132 & -94 & -239 & -277 \\ 57 & -88 & -126 & 19 \end{bmatrix}$$

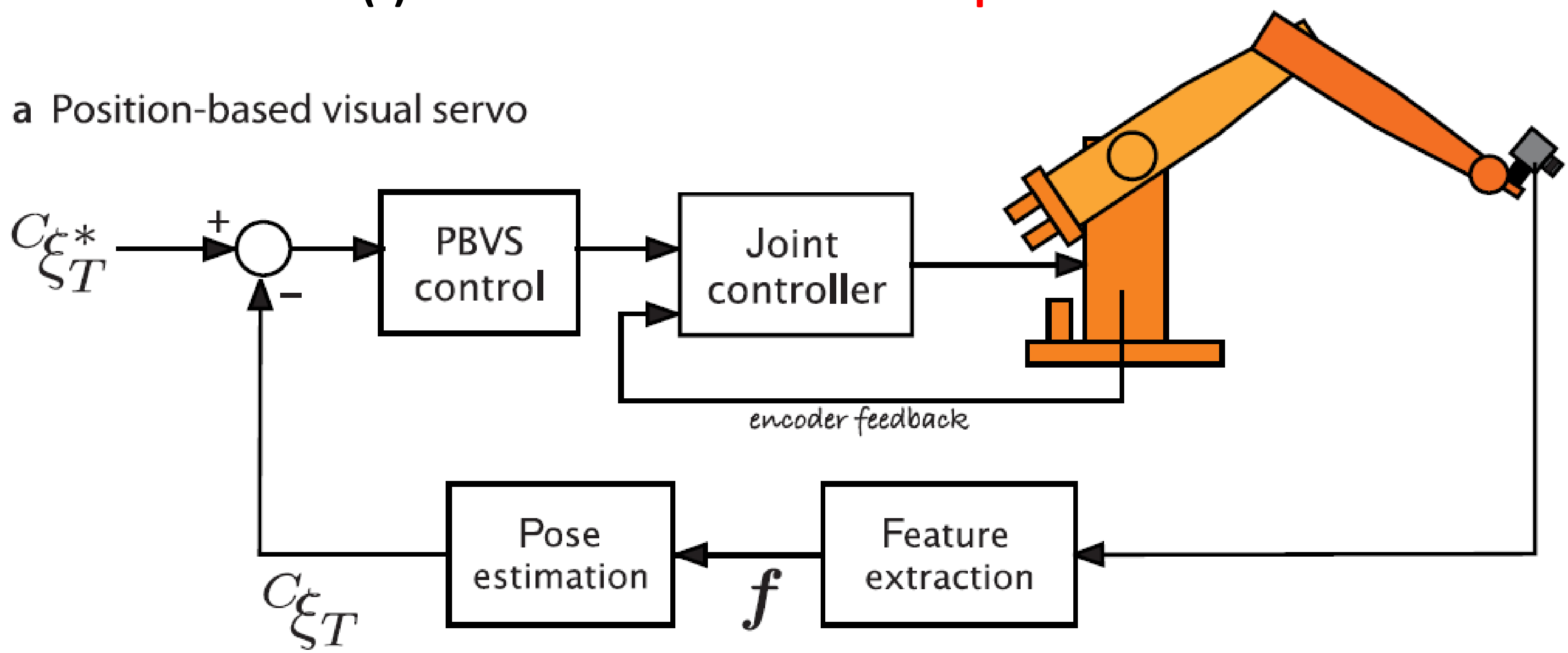
The Basic Components of Visual Servoing

We can distinguish <u>two different</u> visual servo control approaches:

<u>Position-based visual servo control (PBVS):</u>

• **s** consists of a set of 3-D parameters, which must be estimated from image measurements.

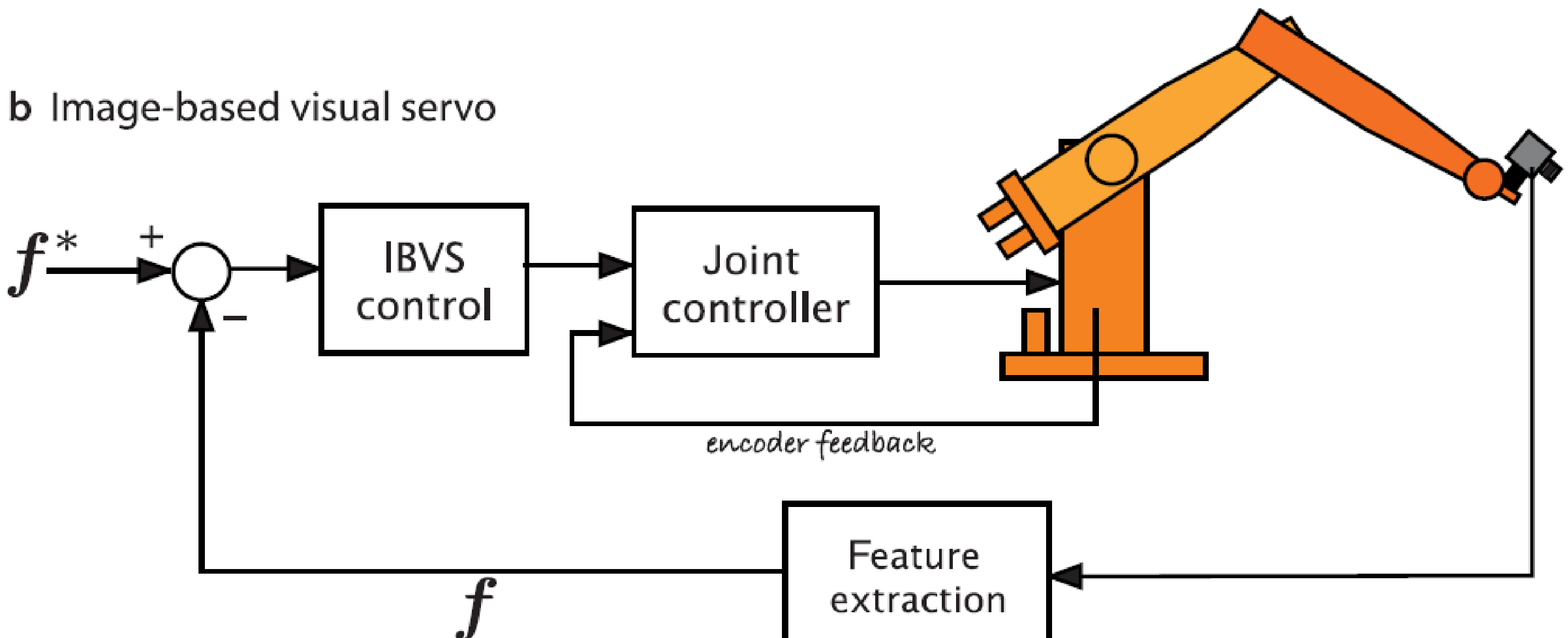
• The control error **e(t)** is defined in the **Cartesian Space**



The Basic Components of Visual Servoing

Image-based visual servo control (IBVS):

- **s** consists of a set of features that are immediately available in the image data.
- The control error **e(t)** is defined in the **Image Space**



The Basic Components of Visual Servoing

Once s has been defined we may proceed to the design of a velocity controller:

Let the spatial velocity of the camera be denoted by: $\mathbf{v}_c = (v_c, \boldsymbol{\omega}_c)$

We can derive the following relationship: $\boxed{\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}_c}$

$\mathbf{L_s} \in \mathbb{R}^{k\times 6}$ is named the interaction matrix (or Image Jacobian) related to **s**.

Camera velocity and time variation of the error: $\boxed{\dot{\mathbf{e}} = \mathbf{L_e}\mathbf{v}_c}$ $\boxed{\mathbf{L_e} = \mathbf{L_s}}$

Ensure exponential decrease of the error: $\dot{\mathbf{e}} = -\lambda\mathbf{e}$

Design the following velocity control scheme for the camera:

$$\boxed{\mathbf{v}_c = -\lambda\mathbf{L_e^+}\mathbf{e}}$$

Vc the input to the robot controller, λ>0 positive gain

$\mathbf{L_e^+} \in \mathbb{R}^{6\times k}$ The Moore- Penrose pseudo inverse of **Le**

The Basic Components of Visual Servoing

Generic Control Law for the Camera Motion:

$$\boxed{\mathbf{v}_c = -\lambda \mathbf{L}_{\mathbf{e}}^{+} \mathbf{e}}$$

$$\mathbf{L}_{\mathbf{e}}^{+} = (\mathbf{L}_{\mathbf{e}}^{\top} \mathbf{L}_{\mathbf{e}})^{-1} \mathbf{L}_{\mathbf{e}}^{\top}$$

When Le is of full rank 6

When k = 6 , if $|\mathbf{L}_{\mathbf{e}}| \neq 0$ it is possible to invert Le, giving the control law:

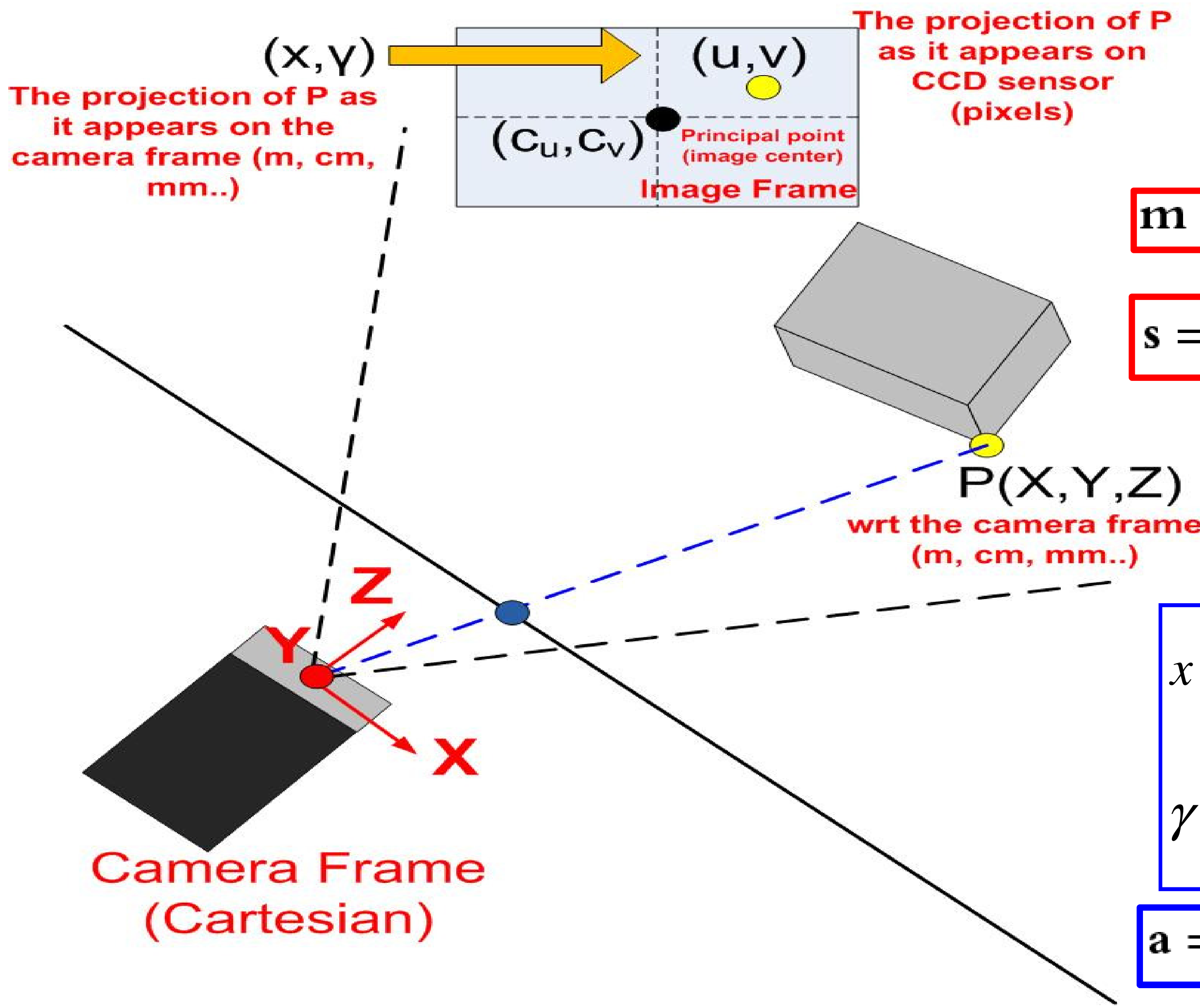
$$\boxed{\mathbf{v}_c = -\lambda \mathbf{L}_{\mathbf{e}}^{-1} \mathbf{e}}$$

Le is very difficult to obtain in real systems...

An approximation of Le is usually known..., so:

$$\widehat{\mathbf{L}_{\mathbf{e}}^{+}} \Rightarrow \boxed{\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_{\mathbf{e}}^{+}} \mathbf{e}}$$

Image Based Visual Servo Control



$$\boxed{\mathbf{m} = (u,\ v)}$$

$$\boxed{\mathbf{s} = \mathbf{x} = (x,\ y)}$$

$$x = \frac{X}{Z} = \frac{u - c_u}{fa}$$

$$y = \frac{Y}{Z} = \frac{v - c_v}{fa}$$

$$\boxed{\mathbf{a} = (c_u,\ c_v,\ f, \alpha)}$$

Image Based Visual Servo Control

The Interaction Matrix

$$\boxed{\begin{aligned} x &= \frac{X}{Z} = \frac{u - c_u}{fa} \\ y &= \frac{Y}{Z} = \frac{v - c_v}{fa} \end{aligned}}$$

Projection Equations
(central projection perspective model)

Taking the time derivative:

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

Relate the velocity of the 3D point to the camera spatial velocity:

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

Image Based Visual Servo Control

The Interaction Matrix

Combine the previous equations:

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

which can be written: $\boxed{\dot{\mathbf{x}} = \mathbf{L_x v}_c}$

Where the interaction matrix related to x is given by:

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

Z is the depth of the point relative to the camera frame and must be estimated or approximated!!!

The Interaction Matrix

To control the 6 DOF, at least three points are necessary (i.e we require k to be greater than 6).
If we use the feature vector: **x = (x1,x2,x3)**, by merely stacking interaction matrices for three points we obtain:

$$\mathbf{L_x} = \begin{bmatrix} \mathbf{L}_{\mathbf{x}_1} \\ \mathbf{L}_{\mathbf{x}_2} \\ \mathbf{L}_{\mathbf{x}_3} \end{bmatrix}$$

In this case, there will exist some configurations for which Lx is singular.

For these reasons, more than three points are usually considered!!!

Image Based Visual Servo Control

Approximating the Interaction Matrix

How to construct the estimate $\widehat{\mathbf{L}_{\mathbf{e}}^{+}}$?

Possible choices:

1. $\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \mathbf{L}_{\mathbf{e}}^{+}$ if the current depth Z of each point is available
2. $\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \mathbf{L}_{\mathbf{e}^*}^{+}$ where $\mathbf{L}_{\mathbf{e}^*}$ is the value of $\mathbf{L}_{\mathbf{e}}$ for the desired position $\mathbf{e} = \mathbf{e}^* = 0$

   only the desired depth of each point has to be set : for i features each desired Z^*_i
3. $\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = 1/2(\mathbf{L}_{\mathbf{e}} + \mathbf{L}_{\mathbf{e}^*})^{+}$ **the current depth of each point must also be available**

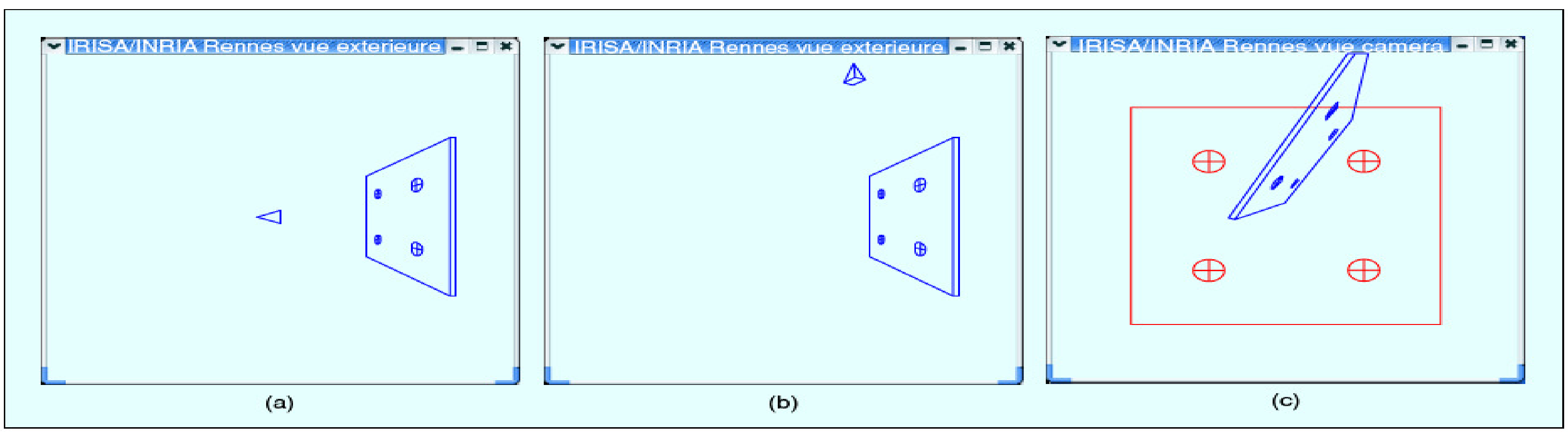


Figure 1. An example of positioning task: (a) the desired camera pose with respect to a simple target, (b) the initial camera pose, and (c) the corresponding initial and desired image of the target.

Image Based Visual Servo Control

Approximating the Interaction Matrix

Example:

Consider a camera with the following parameters:

- **Focal length** f = 0.008
- **Pixel size** [α_x, α_y] = [102400 , 102400]
- **Principal point** [c_u, c_v] = [512, 512]
- **Image size** (resolution): 1024x1024

Four Image Points

$$\mathbf{p_0} = \begin{bmatrix} \mathbf{p}_{0_1} & \mathbf{p}_{0_2} & \mathbf{p}_{0_3} & \mathbf{p}_{0_4} \end{bmatrix}^{\mathbf{T}}$$

$$\mathbf{p_0} = \begin{bmatrix} 280 & 318 & 373 & 335 \\ 469 & 524 & 486 & 431 \end{bmatrix}$$

Their respective Depths:

$$p_{0_{1_Z}} = 1.0m,\ p_{0_{2_Z}} = 1.05m,\ p_{0_{3_Z}} = 1.0m,\ p_{0_{4_Z}} = 0.95m$$

For each image point we compute:

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

Image Based Visual Servo Control

Approximating the Interaction Matrix

Example (continued):

For p_{o_1} :

$$x_{p_{0_1}} = \frac{u_{c_{p01}} - c_u}{fa_x} = \frac{280 - 512}{819.2} = -0.283$$

$$\gamma_{p_{0_1}} = \frac{v_{c_{p01}} - c_v}{fa_y} = \frac{469 - 512}{819.2} = -0.052$$

$$L_{p_{0_1}} = \begin{bmatrix} -1 & 0 & -0.283 & 0.0147 & -1.080 & -0.052 \\ 0 & -1 & -0.052 & 1.0027 & -0.0147 & 0.283 \end{bmatrix}$$

Repeat the same procedure for the rest of the feature points:

$$p_{o_2}, p_{o_3}, p_{o_4}$$

Stack the Interaction Matrices:

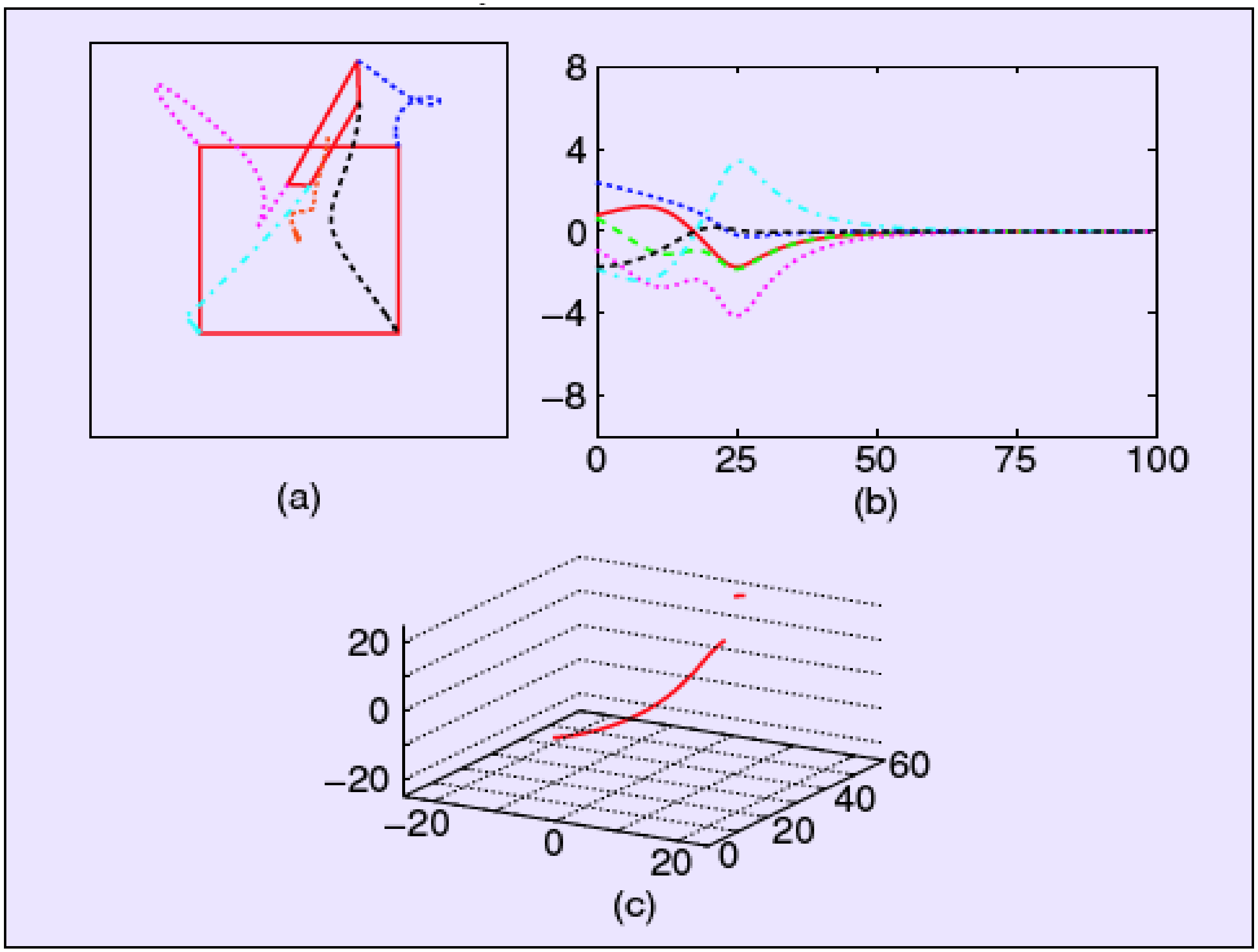
$$L_{p_0} = \begin{bmatrix} L_{p_{0_1}} \\ L_{p_{02}} \\ L_{p_{03}} \\ L_{p_{04}} \end{bmatrix}$$

Using 4 points the dimension is 8x6

Image Based Visual Servo Control

Approximating the Interaction Matrix

$$\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \mathbf{L}_{\mathbf{e}^*}^{+}$$

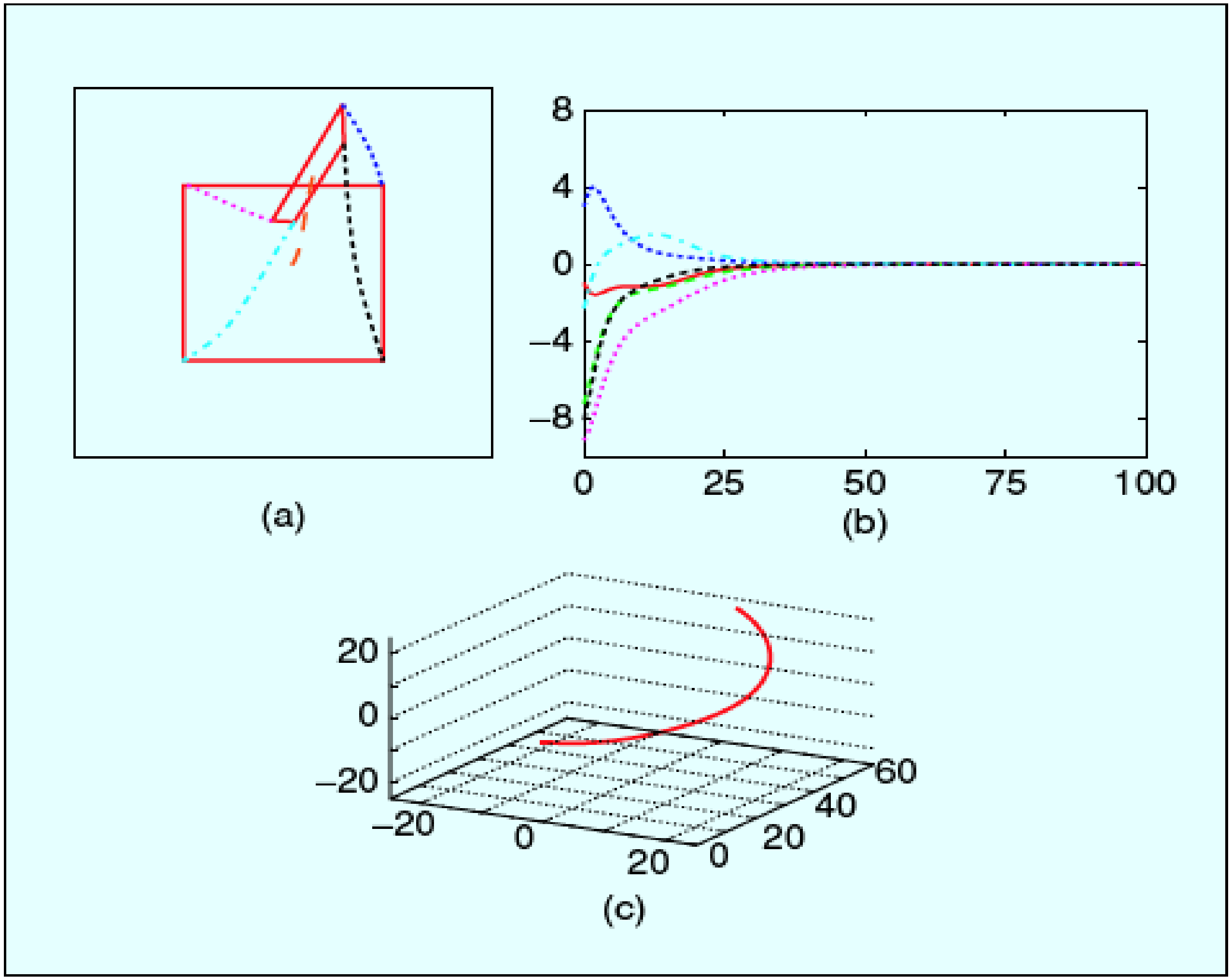


(a) (b) (c)

Image Based Visual Servo Control

Approximating the Interaction Matrix

$$\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \mathbf{L}_{\mathbf{e}}^{+}$$



(a) (b) (c)

Image Based Visual Servo Control

Approximating the Interaction Matrix

$$\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = 1/2(\mathbf{L}_{\mathbf{e}} + \mathbf{L}_{\mathbf{e}^*})^{+}$$

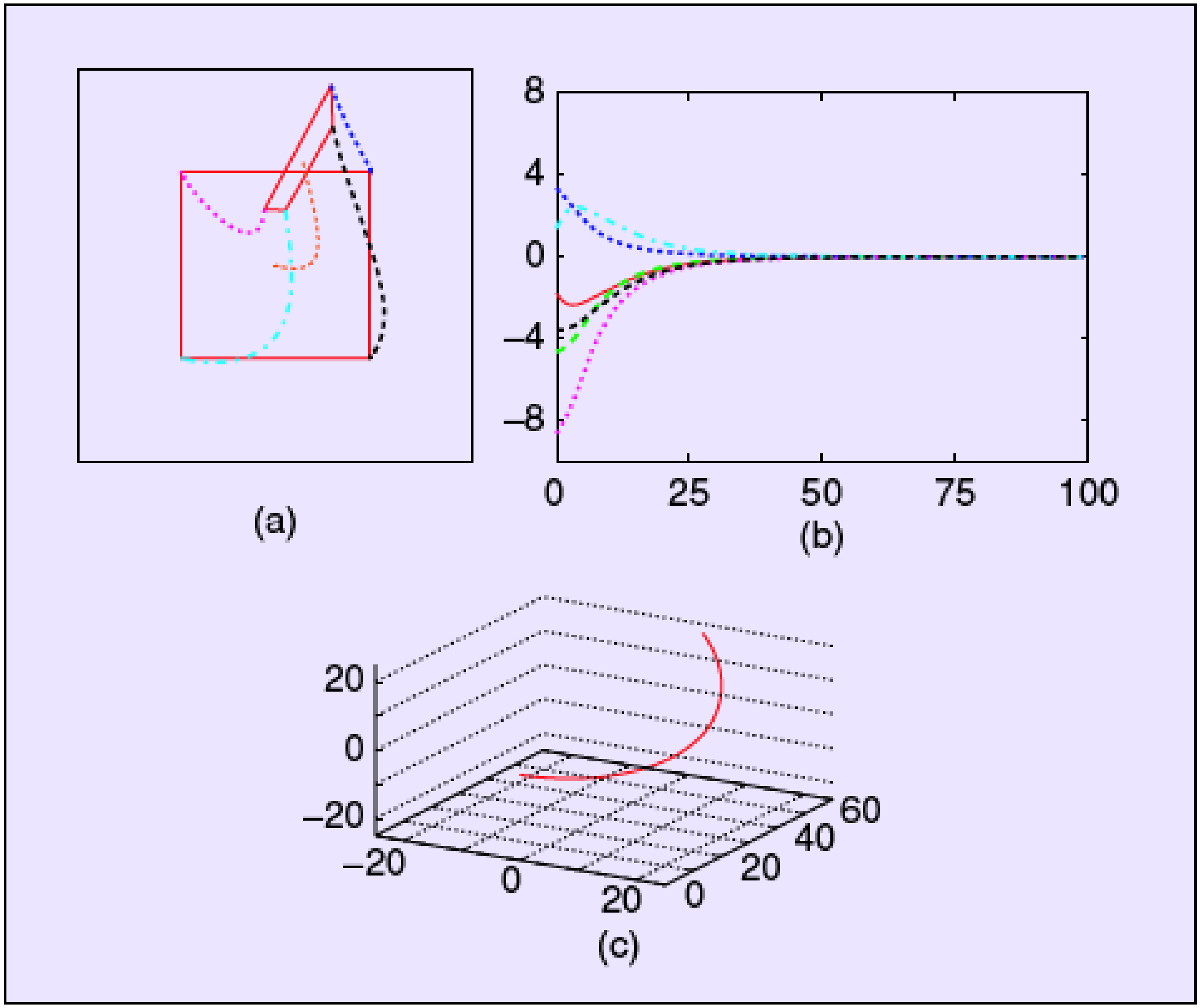


Image Based Visual Servo Control

Generic Example of the Control Design:

Consider a camera mounted on the end effector of a n DoF robotic manipulator (eye in hand) with the following intrinsic parameter vector: $\mathbf{a}=[f,a_x,a_y,c_u,c_v]$
The camera observes four features inside the image frame:

$$\mathbf{m_1}=[u_1,v_1]^T,\mathbf{m_2}=[u_2,v_2]^T,\mathbf{m_3}=[u_3,v_3]^T,\mathbf{m_4}=[u_4,v_4]^T$$

We need to design an IBVS Control Scheme in order to place the four features in the desired positions:

$$\mathbf{m}_1^*=[u_1^*,v_1^*]^T,\mathbf{m}_2^*=[u_2^*,v_2^*]^T,\mathbf{m}_3^*=[u_3^*,v_3^*]^T,\mathbf{m}_4^*=[u_4^*,v_4^*]^T$$

First we calculate $\mathbf{s}=\mathbf{m}(t,\mathbf{a})=[\mathbf{s_1},\mathbf{s_2},\mathbf{s_3},\mathbf{s_4}]^{\mathbf{T}}$

For i feature: $\mathbf{s_i}=[x_i,\gamma_i],x_i=\dfrac{u_i-c_u}{fa_x},\gamma_i=\dfrac{v_i-c_v}{fa_y}$

And the respective desired features: $\mathbf{s}_{\mathbf{i}}^*=[x_i^*,\gamma_i^*],x_i^*=\dfrac{u_i^*-c_u}{fa_x},\gamma_i^*=\dfrac{v_i^*-c_v}{fa_y}$

Thus, the error for i feature is defined: $e_i=[e_{x_i},e_{\gamma_i}],e_{x_i}=x_i-x_i^*,e_{\gamma_i}=\gamma_i-\gamma_i^*$

Image Based Visual Servo Control

Generic Example of the Control Design (continued):

The Interaction Matrix or Image Jacobian will be of the form:

Analytical:

$$\mathbf{L_x} = \begin{bmatrix} -\frac{1}{Z_1} & 0 & \frac{x_1}{Z_1} & x_1\gamma_1 & -(1+x_1)^2 & \gamma_1 \\ 0 & -\frac{1}{Z_1} & \frac{\gamma_1}{Z_1} & 1+\gamma_1^2 & -x_1\gamma_1 & -x_1 \\ -\frac{1}{Z_2} & 0 & \frac{x_2}{Z_2} & x_2\gamma_2 & -(1+x_2)^2 & \gamma_2 \\ 0 & -\frac{1}{Z_2} & \frac{\gamma_2}{Z_2} & 1+\gamma_2^2 & -x_2\gamma_2 & -x_2 \\ -\frac{1}{Z_3} & 0 & \frac{x_3}{Z_3} & x_3\gamma_3 & -(1+x_3)^2 & \gamma_3 \\ 0 & -\frac{1}{Z_3} & \frac{\gamma_3}{Z_3} & 1+\gamma_3^2 & -x_3\gamma_3 & -x_3 \\ -\frac{1}{Z_4} & 0 & \frac{x_4}{Z_4} & x_4\gamma_4 & -(1+x_4)^2 & \gamma_4 \\ 0 & -\frac{1}{Z_4} & \frac{\gamma_4}{Z_4} & 1+\gamma_4^2 & -x_4\gamma_4 & -x_4 \end{bmatrix}$$

$$\mathbf{L_x} = \begin{bmatrix} \mathbf{L_{x_1}} \\ \mathbf{L_{x_2}} \\ \mathbf{L_{x_3}} \\ \mathbf{L_{x_4}} \end{bmatrix} = \mathbf{L_e}$$

Image Based Visual Servo Control

Generic Example of the Control Design (continued):

$$\underbrace{\dot{\mathbf{s}}}_{8x1} = \mathbf{L}_x \mathbf{v}_c \Rightarrow \dot{\mathbf{e}} = \underbrace{\mathbf{L}_e}_{8x6} \underbrace{\mathbf{v}_c}_{6x1}$$

If **Le** is of full rank 6, we can use the Moore-Penrose pseudo inverse:

$$\mathbf{L}_e^{\dagger} = \left(\mathbf{L}_e^T \mathbf{L}_e\right)^{-1} \mathbf{L}_e^T$$

And then we proceed with the design of a velocity control scheme for the camera:

The Camera Velocity Screw

$$\boxed{\underbrace{\mathbf{v}_c}_{6x1} = -\lambda \underbrace{\mathbf{L}_e^{\dagger}}_{6x8} \underbrace{\mathbf{e}}_{8x1}, \quad \lambda > \mathbf{O}_{6x6}}$$

Image Based Visual Servo Control

Generic Example of the Control Design (continued):

We now proceed to the design of a velocity control scheme with respect to the base of the robotic manipulator.

In most cases the end effector frame and the camera frame do not coincide!!!

$$\mathbf{v}_c = \begin{bmatrix} {}^c\mathbf{T}_c \\ {}^c\boldsymbol{\Omega}_c \end{bmatrix}$$

The IBVS scheme calculates the velocity screw of the camera with respect to the camera frame!

We need to express the velocity screw of the camera with respect to the end effector frame. Suppose that the trasformation between the camera frame and the end-effector frame are described by:

$${}^e\mathbf{R}_c, {}^e\mathbf{t}_c$$

Then the velocity screw of the camera with respect to the end-effector frame will be given by:

$${}^e\mathbf{v}_c = \begin{bmatrix} {}^e\mathbf{T}_c \\ {}^e\boldsymbol{\Omega}_c \end{bmatrix} = \begin{bmatrix} {}^e\mathbf{R}_c\,{}^c\mathbf{T}_c - {}^e\mathbf{R}_c\,{}^c\boldsymbol{\Omega}_c \otimes {}^e\mathbf{t}_c \\ {}^e\mathbf{R}_c\,{}^c\boldsymbol{\Omega}_c \end{bmatrix}$$

Image Based Visual Servo Control

Generic Example of the Control Design (continued):

We need now to transform the velocity screw of the camera from the end – effector frame to the base frame of the manipulator, using the following Rotation Matrix and Translation Vector: ${}^{B}\mathbf{R}_{e}, {}^{B}\mathbf{t}_{e}$

$$ {}^{B}\mathbf{v}_{c} = \begin{bmatrix} {}^{B}\mathbf{T}_{c} \\ {}^{B}\mathbf{\Omega}_{c} \end{bmatrix} = \begin{bmatrix} {}^{B}\mathbf{R}_{e}\,{}^{e}\mathbf{T}_{c} - {}^{B}\mathbf{R}_{e}\,{}^{e}\mathbf{\Omega}_{c} \otimes {}^{B}\mathbf{t}_{e} \\ {}^{B}\mathbf{R}_{e}\,{}^{e}\mathbf{\Omega}_{c} \end{bmatrix} $$

Suppose that the camera is mounted on the end effector of a 6 DOF robotic manipulator. The differential kinematics of the arm are of the following form:

$$ \underbrace{{}^{B}\mathbf{v}_{c}}_{6x1} = \underbrace{\mathbf{J}(\mathbf{q})}_{6x6}\underbrace{\dot{\mathbf{q}}}_{6x1} $$

If **J(q)** if full rank:

$$ \boxed{\dot{\mathbf{q}} = -\lambda \mathbf{J}^{-1}(\mathbf{q})\,{}^{B}\mathbf{v}_{c}} $$

Control Law in the Joint Space

Image Based Visual Servo Control

Robotic Manipulator with less than 6 DOF

In the case where the robotic manipulator available for the visual servoing task has less than 6 DOF, the control design is directly addressed to the joint space of the robot

In the joint space the system equations are of the following form:

$$\dot{\mathbf{s}} = \mathbf{J}_s \dot{\mathbf{q}} + \frac{\partial s}{\partial t}, \quad \mathbf{J}_s \in \Re^{kxn}$$

Feature Jacobian Matrix , which can be linked to the interaction matrix and n the number of the robot joints

Time variation of s due to a potential object motion. **If the object is stationary this term is equal to zero**

Robot Jacobian Matrix Expressed in the end-effector frame

$$\boxed{\mathbf{J}_s = \mathbf{L}_s {}^{N}\mathbf{V}_C \mathbf{J}(\mathbf{q})}$$

Feature Jacobian Matrix

${}^{N}\mathbf{V}_c$ Is the spatial motion transform matrix from the vision frame (camera frame) to the end effector frame. It is usually a constant matrix, as soon as the vision sensor sensor is rigidly attached to the end-effector

Image Based Visual Servo Control

Robotic Manipulator with less than 6 DOF

We proceed to the design of a control scheme expressed in the joint space:

$$\boxed{\dot{\mathbf{q}} = -\lambda \hat{\mathbf{J}}_{\mathbf{e}}^{\dagger}\mathbf{e} - \hat{\mathbf{J}}_{\mathbf{e}}^{\dagger}\frac{\partial \mathbf{e}}{\partial t}} \qquad \hat{\mathbf{J}}_{s}^{\dagger} = \hat{\mathbf{J}}_{e}^{\dagger}$$

If k=n, considering the Lyapunov function $L = \frac{1}{2}\left\|\mathbf{e}(t)\right\|^2$ a sufficient condition to ensure the global asymptotic stability is given by:

$$\mathbf{J}_{\mathbf{e}}\hat{\mathbf{J}}_{\mathbf{e}}^{\dagger} > 0$$

If k>n, we obtain :

$$\hat{\mathbf{J}}_{\mathbf{e}}^{\dagger}\mathbf{J}_{\mathbf{e}} > 0$$

To ensure the local asymptotic stability of the system

Let us note that, even if the robot has six degrees of freedom, it is generally not equivalent to first compute $\mathbf{v}_c$ and then deduce $\dot{\mathbf{q}}$ **using the robot inverse Jacobian,** and to compute directly $\dot{\mathbf{q}}$. **Indeed, it may occur** that the robot Jacobian ***J(q) is singular while the feature Jacobian* Js is not (that may occur when *k < n).***

Image Based Visual Servo Control

Robotic Manipulator with less than 6 DOF

Example with m = 4 features and n = 2 DOF planar manipulator:

Feature Jacobian:

$$\mathbf{L_x} = \begin{bmatrix} \mathbf{L}_{x_1} \\ \mathbf{L}_{x_2} \\ \mathbf{L}_{x_3} \\ \mathbf{L}_{x_4} \end{bmatrix}, \mathbf{L_e} = \mathbf{L_x}$$

Spatial motion transform matrix from the vision frame to the end effector frame :

$${}^{N}\mathbf{V}_c = \begin{bmatrix} {}^{N}\mathbf{R}_c & \left[{}^{N}\mathbf{t}_C\right]_x {}^{N}\mathbf{R}_c \\ \mathbf{0} & {}^{N}\mathbf{R} \end{bmatrix}$$

Feature Jacobian Matrix in Joint Space:

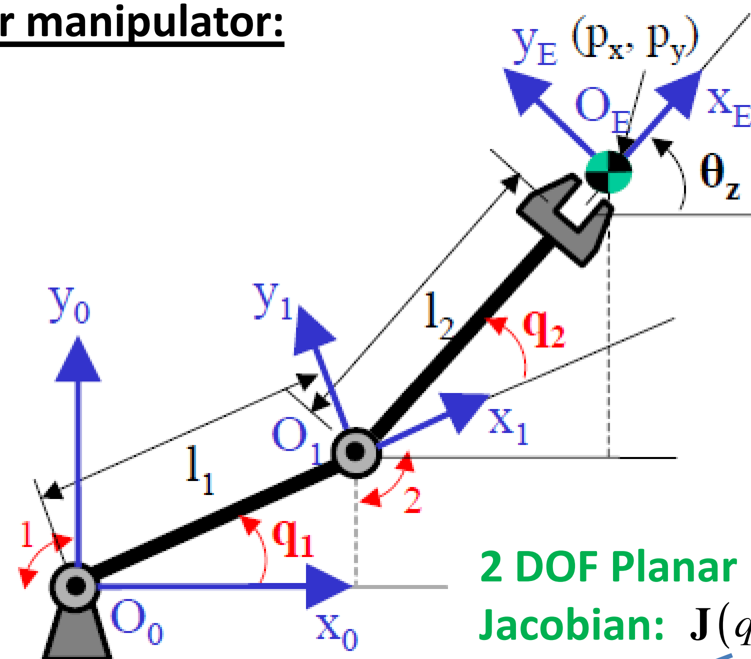
$$\boxed{\mathbf{J_s = L_s {}^{N}V_C J(q)}} \qquad \mathbf{J_s = J_e}$$

Control Scheme in Joint Space:

$$\boxed{\dot{\mathbf{q}} = -\lambda \hat{\mathbf{J}}_e^{\dagger} \mathbf{e} - \hat{\mathbf{J}}_e^{\dagger} \cancel{\frac{\partial \mathbf{e}}{\partial t}}}$$

0 Stationary target

$$\hat{\mathbf{J}}_e^{\dagger} = \left(\mathbf{J}_e^T \mathbf{J}_e\right)^{-1} \mathbf{J}_e^T$$

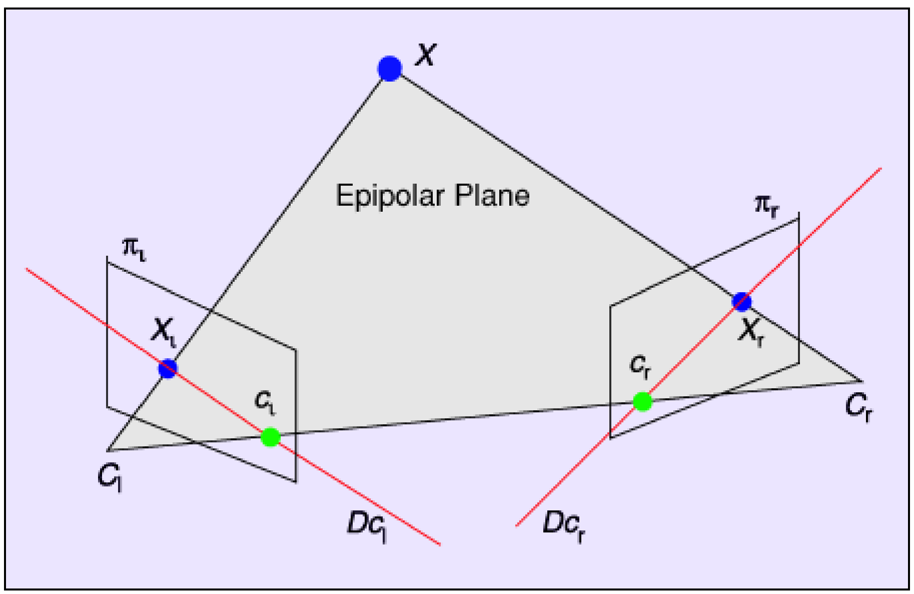


2 DOF Planar Robot Jacobian: $\mathbf{J}(q_1, q_2)$

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} -(l_1 s_1 + l_2 s_{12}) & -l_2 s_{12} \\ (l_1 c_1 + l_2 c_{12}) & l_2 c_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

Image Based Visual Servo Control

IBVS with a Stereo Vision System



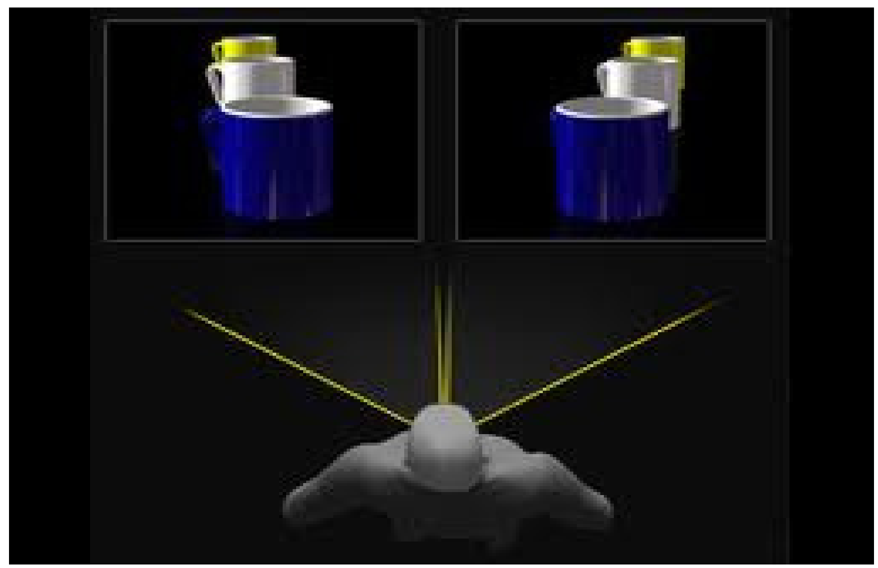


Image Based Visual Servo Control

IBVS with a Stereo Vision System

If a stereo vision system is used, a 3-D point is visible in both left and right images:

$$\mathbf{s} = \mathbf{x}_s = (\mathbf{x}_l, \mathbf{x}_r) = (x_l, y_l, x_r, y_r)$$

care must be taken when constructing the corresponding interaction matrix since the form given is expressed in either the left or right camera frame:

$$\begin{cases} \dot{\mathbf{x}}_l & = & \mathbf{L}_{\mathbf{x}_l} \mathbf{v}_l \\ \dot{\mathbf{x}}_r & = & \mathbf{L}_{\mathbf{x}_r} \mathbf{v}_r, \end{cases}$$

By choosing a sensor frame rigidly linked to the stereo vision system, we obtain:

$$\dot{\mathbf{x}}_s = \begin{bmatrix} \dot{\mathbf{x}}_l \\ \dot{\mathbf{x}}_r \end{bmatrix} = \mathbf{L}_{\mathbf{x}_s} \mathbf{v}_s$$

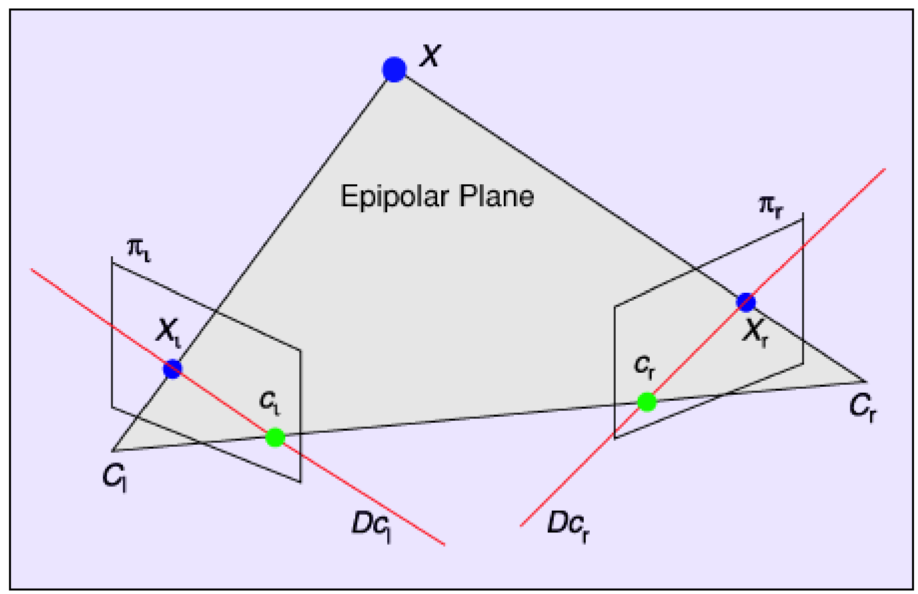


Image Based Visual Servo Control

IBVS with a Stereo Vision System

$$\dot{\mathbf{x}}_s = \begin{bmatrix} \dot{\mathbf{x}}_l \\ \dot{\mathbf{x}}_r \end{bmatrix} = \mathbf{L}_{\mathbf{x}_s} \mathbf{v}_s$$

where the interaction matrix related to $\mathbf{x}_s$ can be determined using the spatial motion transform matrix **V** to transform velocities expressed in the left or right cameras frames to the sensor frame.

$$\mathbf{V} = \begin{bmatrix} \mathbf{R} & [\mathbf{t}]_\times \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}$$

where $[\mathbf{t}]_x$ is the skew symmetric matrix associated to the vector **t** and where **(R, t) ∈ SE (3)** is the rigid body transformation from camera to sensor frame.

$$\mathbf{L}_{\mathbf{x}_s} = \begin{bmatrix} \mathbf{L}_{\mathbf{x}_l} \, {}^l\mathbf{V}_s \\ \mathbf{L}_{\mathbf{x}_r} \, {}^r\mathbf{V}_s \end{bmatrix}$$

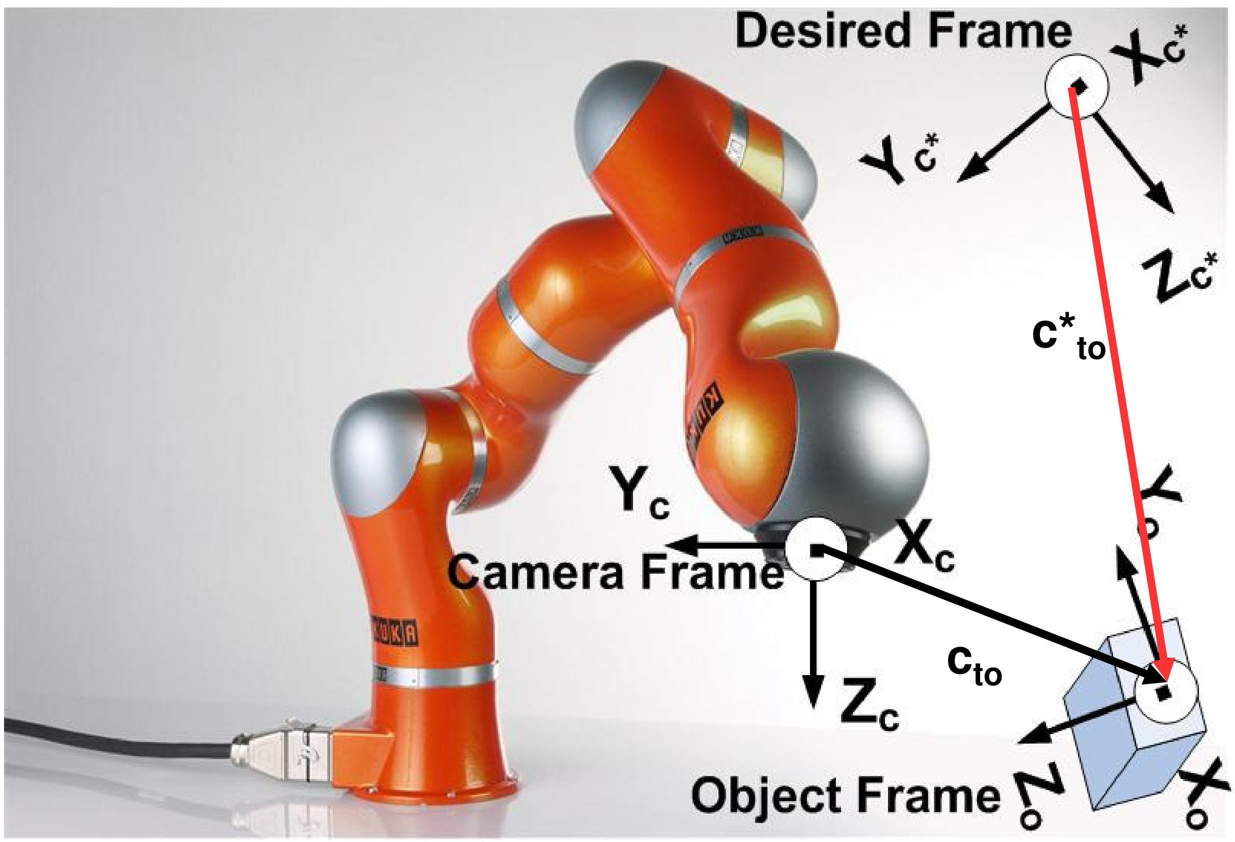
$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \quad [t]_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

Note that $\mathbf{Lx_s} \in \mathbf{R}^{4\times 6}$ is always of rank 3 because of the epipolar constraint that links the perspective projection of a 3-D point in a stereo vision system. To control the 6 DOF of the system, it is necessary to consider at least three points, the rank of the interaction matrix by considering only two points being equal to 5.

Position Based Visual Servo Control

Position-based control schemes (PBVS) use the pose of the camera with respect to some reference coordinate frame to define s.

Computing that pose from a set of measurements in one image necessitates the camera intrinsic parameters and the 3-D model of the object observed to be known (parameter vector a).



We consider three coordinate frames:

- The **current camera frame Fc**
- The desired camera frame F c∗
- The object reference frame Fo

The coordinate vectors $\mathbf{c}_{to}$ and $\mathbf{c}^*_{to}$ give the coordinates of the origin of the object frame expressed relative to the **current camera frame** and relative to the **desired camera frame**, respectively.

Position Based Visual Servo Control

Let **R** = $^{c*}\mathbf{R}_c$ be the rotation matrix that gives the orientation of the current camera frame relative to the desired frame.

We can define **s** to be (**t**, θ**u**), in which **t** is a translation vector, and θ**u** gives the angle/axis parameterization for the rotation.
We now discuss two choices for **t** and give the corresponding control laws.

If **t** is defined relative to the object frame **Fo** , we obtain:

$$\boxed{\mathbf{s} = ({}^{c}\mathbf{t}_{o}, \theta\mathbf{u}),\ \mathbf{s}^* = ({}^{c^*}\mathbf{t}_{o}, 0),\ \text{and}\ \mathbf{e} = ({}^{c}\mathbf{t}_{o} - {}^{c^*}\mathbf{t}_{o}, \theta\mathbf{u})}$$

In this case, the interaction matrix related to **e** is given by:

$$\boxed{\mathbf{L_e} = \begin{bmatrix} -\mathbf{I}_3 & [{}^{c}\mathbf{t}_{o}]_\times \\ 0 & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix}}$$

Position Based Visual Servo Control

$$\mathbf{L_e} = \begin{bmatrix} -\mathbf{I}_3 & [{}^c\mathbf{t}_o]_\times \\ 0 & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix}$$

in which **I3x3** is the 3 × 3 identity matrix and **Lθu** is given by:

$$\mathbf{L}_{\theta\mathbf{u}} = \mathbf{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\operatorname{sinc}\theta}{\operatorname{sinc}^2\frac{\theta}{2}}\right)[\mathbf{u}]_\times^2$$

where sinc x is the sinus cardinal defined such that

x sinc x = sin x and sinc 0 = 1

Following the developments presented at the beginning of the presentation, we obtain the control scheme:

$$\mathbf{v}_c = -\lambda\widehat{\mathbf{L}_\mathbf{e}^{-1}}\mathbf{e}$$

since the dimension k of s is 6, which is the number of camera degrees of freedom.

By setting:

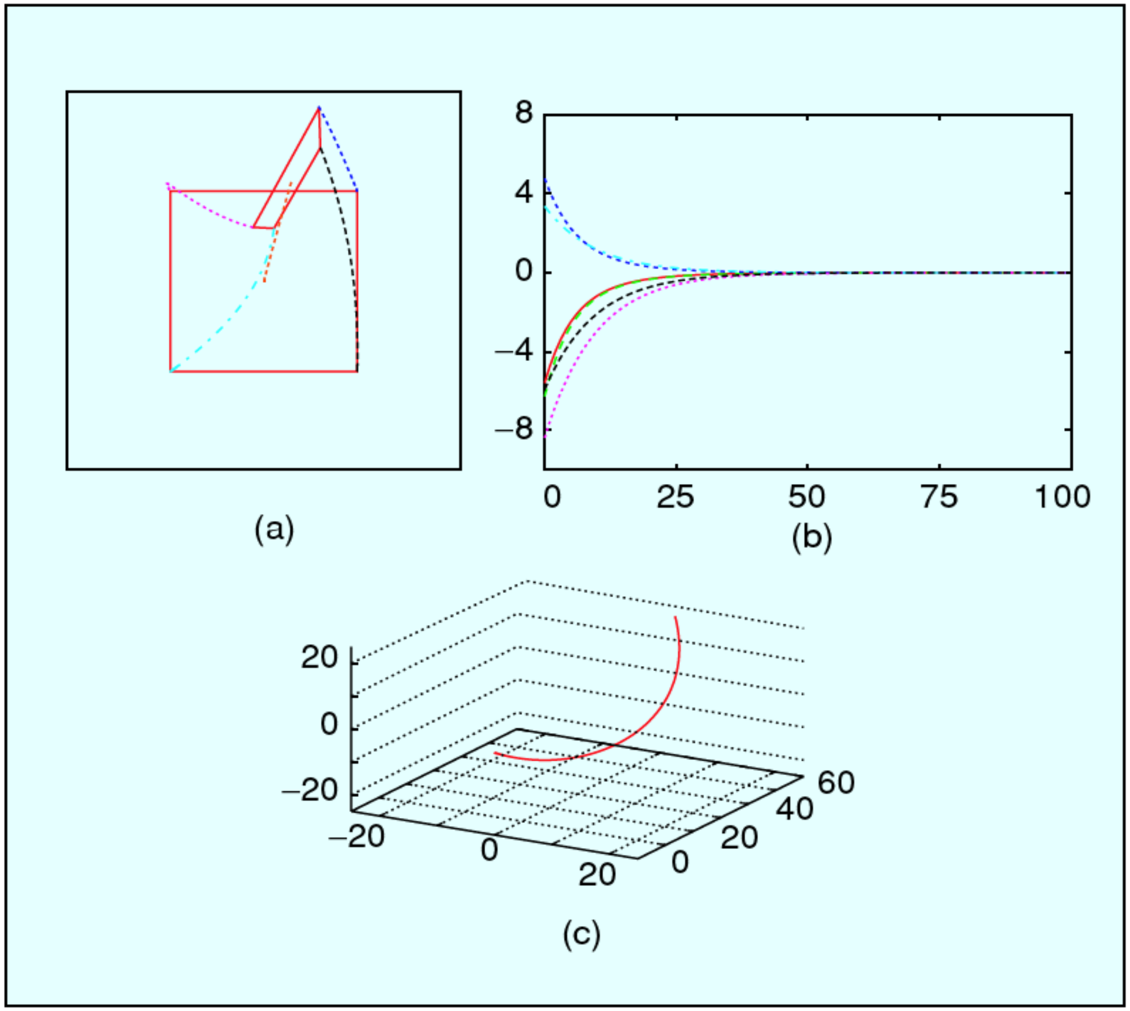
$$\widehat{\mathbf{L}_\mathbf{e}^{-1}} = \begin{bmatrix} -\mathbf{I}_3 & [{}^c\mathbf{t}_o]_\times \mathbf{L}_{\theta\mathbf{u}}^{-1} \\ 0 & \mathbf{L}_{\theta\mathbf{u}}^{-1} \end{bmatrix}$$

$$\begin{cases} \boldsymbol{v}_c & = & -\lambda\left(\left({}^{c^*}\mathbf{t}_o - {}^c\mathbf{t}_o\right) + [{}^c\mathbf{t}_o]_\times \theta\mathbf{u}\right) \\ \boldsymbol{\omega}_c & = & -\lambda\theta\mathbf{u}, \end{cases}$$

$$\mathbf{L}_{\theta\mathbf{u}}^{-1}\theta\mathbf{u} = \theta\mathbf{u}$$

Position Based Visual Servo Control

This PBVS scheme causes the rotational motion to follow a geodesic with an exponential decreasing speed and so that the translational parameters involved in **s** decrease with the same speed. This explains the nice exponential decrease of the camera velocity components In the Example. Furthermore, the trajectory in the image of the origin of the object frame follows a pure straight line (here the center of the four points has been selected as this origin). On the other hand, the camera trajectory does not follow a straight line.



Position Based Visual Servo Control

Another PBVS scheme can be designed by using **s** = ($^{c*}\mathbf{t}_c$, θ**u**).
In that case, we have **s**∗ = 0, **e** = **s**, and:

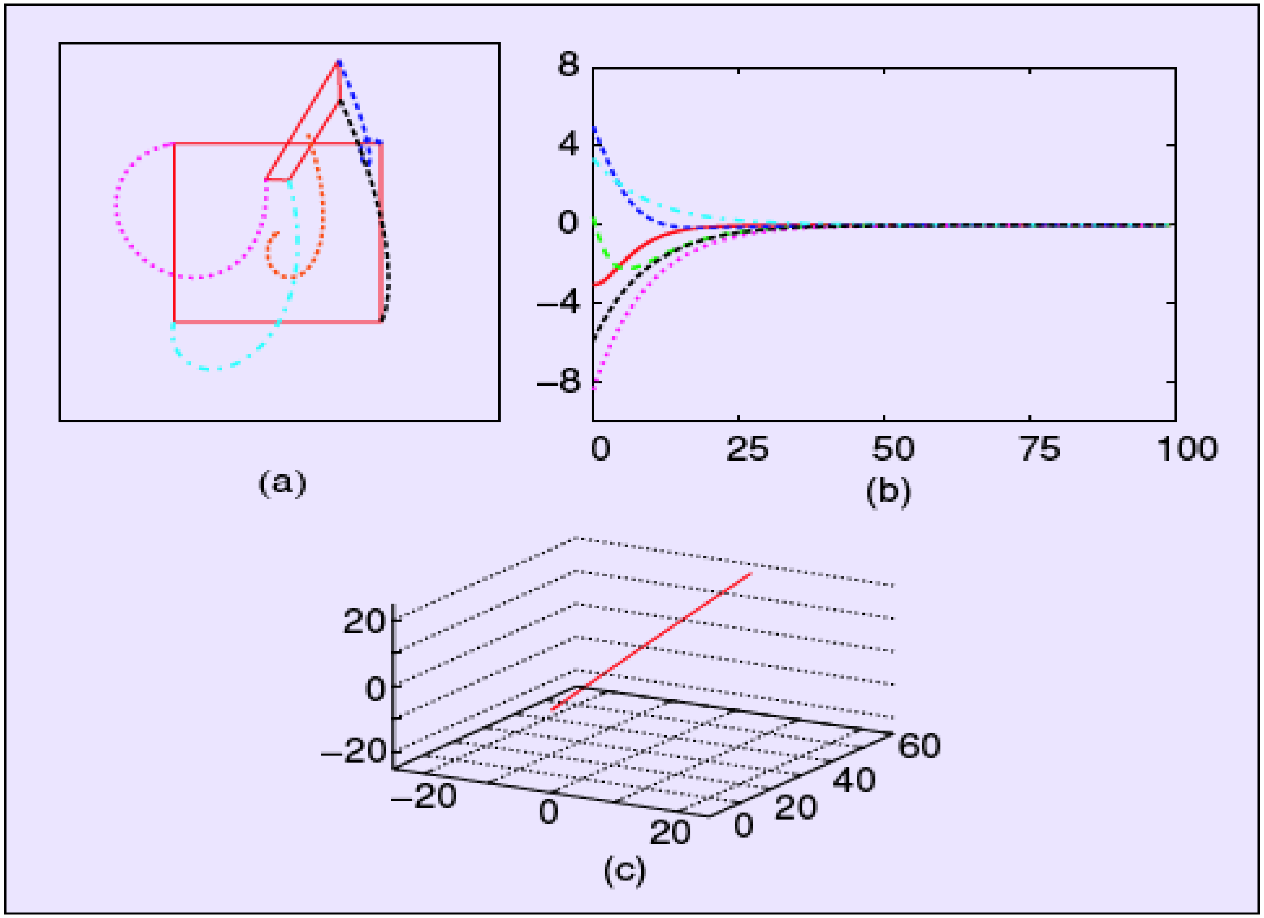
$$\boxed{\mathbf{L_e} = \begin{bmatrix} \mathrm{R} & 0 \\ 0 & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix}}$$

Note the decoupling between translational and rotational motions,
which allows us to obtain a simple control scheme:

$$\boxed{\begin{cases} \boldsymbol{v}_c & = & -\lambda \mathbf{R}^\top\, {}^{c^*}\mathbf{t}_c \\ \boldsymbol{\omega}_c & = & -\lambda\, \theta \mathbf{u}. \end{cases}}$$

Position Based Visual Servo Control

In this case, as can be seen in the Example, if the pose parameters involved are perfectly estimated, the camera trajectory is a pure straight line, while the image trajectories are less satisfactory than before. Some particular configurations can even be found so that some points leave the camera field of view.



(a) (b) (c)

Stability Analysis

We consider the fundamental issues related to **the stability of the visual servo controllers**.
To assess the stability of the closed-loop visual servo systems, we will use **Lyapunov analysis**.
In particular, consider the candidate Lyapunov function defined by **the squared error norm**:

$$\boxed{\mathcal{L} = 1/2\|\mathbf{e}(t)\|^2} \Rightarrow \boxed{\begin{aligned}\dot{\mathcal{L}} &= \mathbf{e}^\top \dot{\mathbf{e}} \\ &= -\lambda \mathbf{e}^\top \mathbf{L_e} \widehat{\mathbf{L_e^+}} \mathbf{e}\end{aligned}}$$

The global asymptotic stability of the system is thus obtained when the following sufficient condition is ensured:

$$\boxed{\mathbf{L_e} \widehat{\mathbf{L_e^+}} > 0}$$

If the number of features is equal to the number of camera degrees of freedom (i.e., k = 6), and if the features are chosen and the control scheme designed so that $\mathbf{L_e}$ and $\widehat{\mathbf{L_e^+}}$ are of full rank 6, then the above condition is ensured if the approximations involved in $\mathbf{L_e}$ and $\widehat{\mathbf{L_e^+}}$ are not too coarse.

Stability Analysis of Image Based Visual Servo Control

To study **local asymptotic stability** when k > 6, let us first define a new error:

$$\boxed{\mathbf{e}' = \widehat{\mathbf{L}_{\mathbf{e}}^{+}}\mathbf{e}} \Rightarrow \boxed{\begin{aligned} \dot{\mathbf{e}}' &= \widehat{\mathbf{L}_{\mathbf{e}}^{+}}\dot{\mathbf{e}} + \dot{\widehat{\mathbf{L}_{\mathbf{e}}^{+}}}\mathbf{e} \\ &= (\widehat{\mathbf{L}_{\mathbf{e}}^{+}}\mathbf{L}_{\mathbf{e}} + \mathbf{O})\mathbf{v}_c \end{aligned}}$$

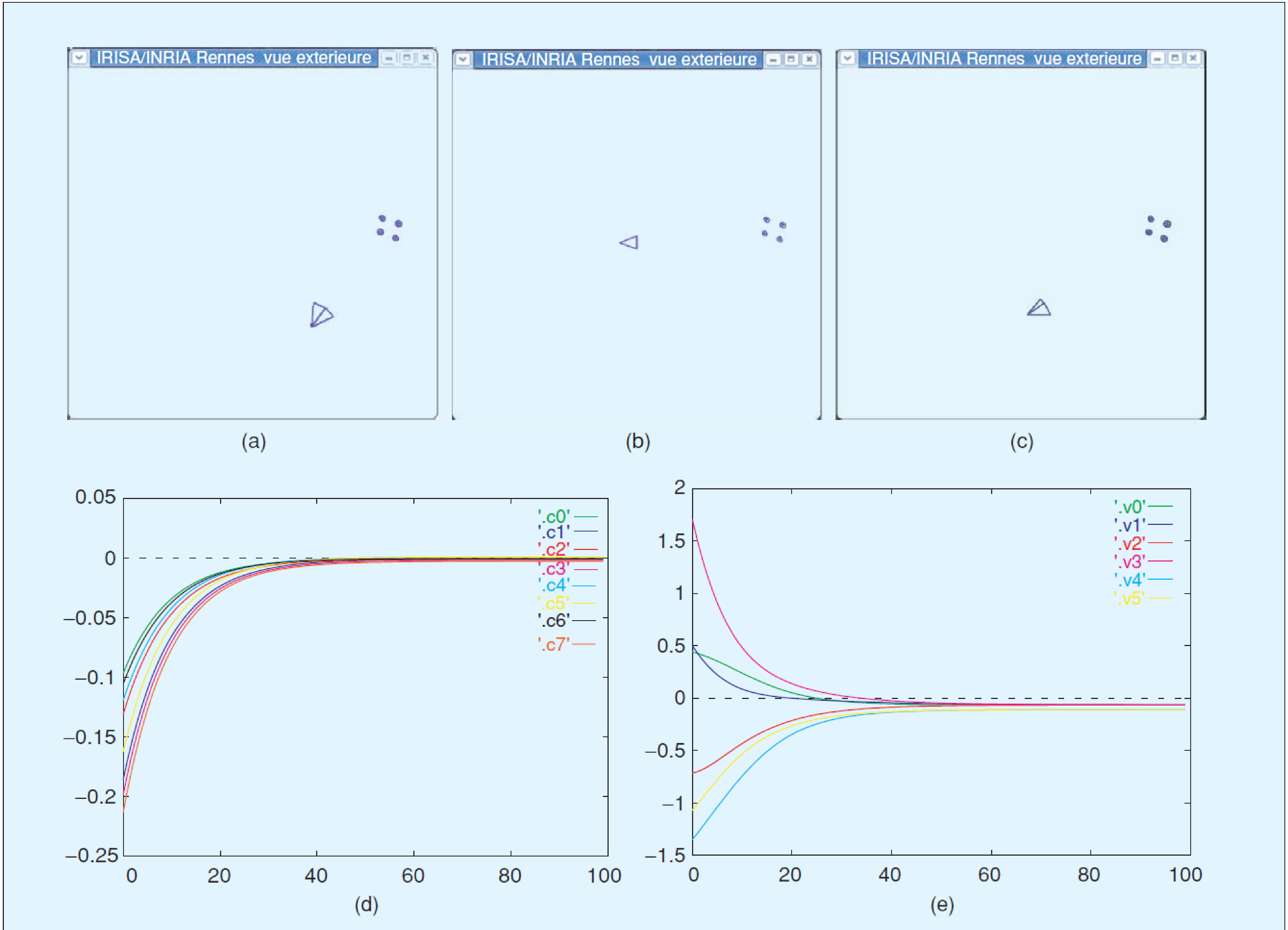
where **O** ∈ $\mathbf{R}_{6\times6}$ is equal to 0 when **e** = 0, whatever the choice of $\mathbf{Le}^{+}$.
Using the velocity control scheme , we obtain:

$$\boxed{\dot{\mathbf{e}}' = -\lambda(\widehat{\mathbf{L}_{\mathbf{e}}^{+}}\mathbf{L}_{\mathbf{e}} + \mathbf{O})\mathbf{e}'} \xrightarrow{\mathbf{O} = 0} \boxed{\dot{\mathbf{e}}' = -\lambda\widehat{\mathbf{L}_{\mathbf{e}}^{+}}\mathbf{L}_{\mathbf{e}}\mathbf{e}'}$$

which is known to be locally asymptotically stable in a neighborhood of **e = e* = 0** if:

$$\boxed{\widehat{\mathbf{L}_{\mathbf{e}}^{+}}\mathbf{L}_{\mathbf{e}} > 0}$$

$$\widehat{\mathbf{L}_{\mathbf{e}}^{+}}\mathbf{L}_{\mathbf{e}} \in \mathbb{R}^{6\times 6}$$


IRISA/INRIA Rennes vue exterieure
IRISA/INRIA Rennes vue exterieure
IRISA/INRIA Rennes vue exterieure
(a)
(b)
(c)
0.05
0
−0.05
−0.1
−0.15
−0.2
−0.25
0
20
40
60
80
100
'.c0'
'.c1'
'.c2'
'.c3'
'.c4'
'.c5'
'.c6'
'.c7'
(d)
2
1.5
1
0.5
0
−0.5
−1
−1.5
0
20
40
60
80
100
'.v0'
'.v1'
'.v2'
'.v3'
'.v4'
'.v5'
(e)

Stability Analysis of Position Based Visual Servo Control

$$\boxed{\mathbf{L}_{\theta\mathbf{u}} = \mathbf{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\operatorname{sinc}\theta}{\operatorname{sinc}^2\frac{\theta}{2}}\right)[\mathbf{u}]_\times^2}$$

Since **Lθu** is nonsingular when

$$\theta \neq 2\,k\pi$$

We obtain the global asymptotic stability of the system since **Le Le-^{1}= I$_6$** , under the strong hypothesis that **all the pose parameters are perfect**.

This is true for both PBVS methods presented previously, since the interaction matrices are full rank when **Lθu** is nonsingular.

$$\boxed{\mathbf{L}_\mathbf{e} = \begin{bmatrix} -\mathbf{I}_3 & [{}^c\mathbf{t}_o]_\times \\ 0 & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix}}$$

$$\boxed{\mathbf{L}_\mathbf{e} = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix}}$$

Stability Analysis of Position Based Visual Servo Control

With regard to robustness, feedback is computed using estimated quantities that are a function of the image measurements and the system calibration parameters. For the first PBVS method (the analysis for the second method is analogous), the interaction matrix corresponds to perfectly estimated pose parameters, while the real one is unknown since the estimated pose parameters may be biased due to calibration errors, or inaccurate and unstable due to noise .

The true positivity condition should be in fact written:

Even small errors in computing the points position in the image can lead to pose errors that can impact significantly the accuracy and the stability of the system. Example:

$$\boxed{\mathbf{L}_{\widehat{\mathbf{e}}}\widehat{\mathbf{L}_{\mathbf{e}}^{-1}} > 0}$$

$$\widehat{\mathbf{L}_{\mathbf{e}}^{-1}} = \begin{bmatrix} -\mathbf{I}_3 & [{}^c\mathbf{t}_o]_\times \mathbf{L}_{\theta\mathbf{u}}^{-1} \\ \mathbf{0} & \mathbf{L}_{\theta\mathbf{u}}^{-1} \end{bmatrix}$$

$\mathbf{L}_{\widehat{\mathbf{e}}}$ **Is unknown**

(a) (b) (c)

This presentation was based on:

1. ***Visual Servo Control Part I: Basic Approaches*, Francois Chaumette and Seth Hutchinson, IEEE Robotics and Automation Magazine 13, 4 (2006), 82-90**

2. ***Visual Servo Control Part II: Advanced Approaches*, Francois Chaumette and Seth Hutchinson, IEEE Robotics and Automation Magazine 14, 1 (2007), 109-118**