Visual Servoing

François Chaumette
Inria, IRISA, Rennes

Overview

- Introduction
- Modeling
- Control
- Applications and other issues

Introduction

Goal: to position a robot wrt. an object from the data given by a camera

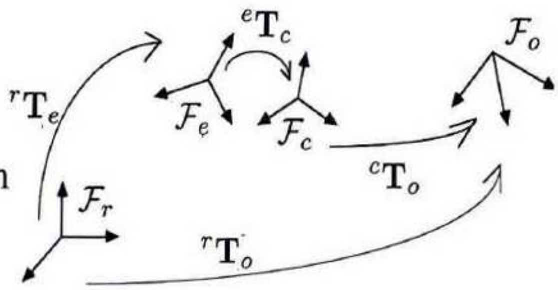
For a given desired pose ${}^{c^*}\mathbf{T}_o$

- Compute the displacement to be done: ${}^{c}\mathbf{T}_{c^*} = {}^{c}\mathbf{T}_o\, {}^{c^*}\mathbf{T}_o^{-1}$

  where ${}^{c}\mathbf{T}_o$ is known from pose estimation

- Expressed in the robot frame, this displacement is given by:

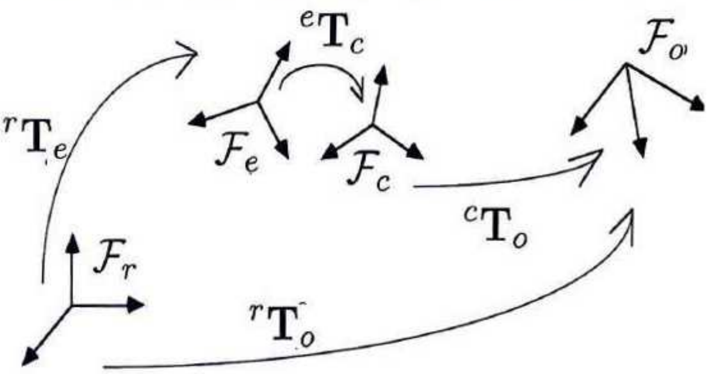
$$ {}^{e}\mathbf{T}_{e^*} = {}^{e}\mathbf{T}_c\, {}^{c}\mathbf{T}_o\, {}^{c^*}\mathbf{T}_o^{-1}\, {}^{e}\mathbf{T}_c^{-1} $$



- ${}^{e}\mathbf{T}_c$ needed to be known: hand-eye calibration

Hand-eye calibration

How to estimate the pose of a camera wrt. a robot?



$$ \begin{aligned} {}^{e}\mathbf{T}_c &= {}^{e}\mathbf{T}_r\, {}^{r}\mathbf{T}_o\, {}^{o}\mathbf{T}_c \\ &= {}^{r}\mathbf{T}_e^{-1}\, {}^{r}\mathbf{T}_o\, {}^{c}\mathbf{T}_o^{-1} \end{aligned} $$

- ${}^{e}\mathbf{T}_c$: the (constant) pose we are looking for
- ${}^{r}\mathbf{T}_e$: obtained from robot geometric model
- ${}^{c}\mathbf{T}_o$: obtained from pose estimation
- ${}^{r}\mathbf{T}_o$: constant but unknown

Idea: ${}^{r}T_o$ is the same whatever the robot/camera pose

$$ \begin{aligned} {}^{r}\mathbf{T}_o &= {}^{r}\mathbf{T}_{e_i}\, {}^{e_i}\mathbf{T}_{c_i}\, {}^{c_i}\mathbf{T}_o &= {}^{r}\mathbf{T}_{e_j}\, {}^{e_j}\mathbf{T}_{c_j}\, {}^{c_j}\mathbf{T}_o \\ &= {}^{r}\mathbf{T}_{e_i}\, {}^{e}\mathbf{T}_c\, {}^{c_i}\mathbf{T}_o &= {}^{r}\mathbf{T}_{e_j}\, {}^{e}\mathbf{T}_c\, {}^{c_j}\mathbf{T}_o \end{aligned} $$

since ${}^{e}\mathbf{T}_c = {}^{e_i}\mathbf{T}_{c_i} = {}^{e_j}\mathbf{T}_{c_j}$

Hand-eye calibration

$$\begin{aligned} {}^r\mathbf{T}_{e_i}\ {}^e\mathbf{T}_c\ {}^{c_i}\mathbf{T}_o &= {}^r\mathbf{T}_{e_j}\ {}^e\mathbf{T}_c\ {}^{c_j}\mathbf{T}_o \\ \Leftrightarrow\quad {}^r\mathbf{T}_{e_j}^{-1}\ {}^r\mathbf{T}_{e_i}\ {}^e\mathbf{T}_c &= {}^e\mathbf{T}_c\ {}^{c_j}\mathbf{T}_o\ {}^{c_i}\mathbf{T}_o^{-1} \\ \Leftrightarrow\quad {}^{e_j}\mathbf{T}_{e_i}\ {}^e\mathbf{T}_c &= {}^e\mathbf{T}_c\ {}^{c_j}\mathbf{T}_{c_i} \\ \Leftrightarrow\quad \mathbf{A}\,\mathbf{X} &= \mathbf{X}\,\mathbf{B} \end{aligned}$$

Then, decompose the rotation and translation part:

$$\begin{cases} {}^{e_j}\mathbf{R}_{e_i}\ {}^e\mathbf{R}_c = {}^e\mathbf{R}_c\ {}^{c_j}\mathbf{R}_{c_i} \\ ({}^{e_j}\mathbf{R}_{e_i} - \mathbf{I}_3)\ {}^e\mathbf{t}_c = {}^e\mathbf{R}_c\ {}^{c_j}\mathbf{t}_{c_i} - {}^{e_j}\mathbf{t}_{e_i} \end{cases}$$

Once ${}^e\mathbf{R}_c$ is known, ${}^e\mathbf{t}_c$ is obtained by solving a simple linear system

$({}^{e_j}\mathbf{R}_{e_i} - \mathbf{I}_3)$ is of rank 2 ; at least a third orientation k is necessary

Hand-eye calibration

$$\begin{aligned} {}^e\mathbf{R}_c &: \theta\mathbf{u} \\ {}^{e_j}\mathbf{R}_{e_i} &: \theta_e\mathbf{u}_e \\ {}^{c_j}\mathbf{R}_{c_i} &: \theta_c\mathbf{u}_c \end{aligned}$$

Thanks to rotation properties

${}^{e_j}\mathbf{R}_{e_i}\ {}^e\mathbf{R}_c = {}^e\mathbf{R}_c\ {}^{c_j}\mathbf{R}_{c_i}$ equivalent to a linear system

$$[\theta_c\mathbf{u}_c + \theta_e\mathbf{u}_e]_\times \tan\frac{\theta}{2}\mathbf{u} = \theta_c\mathbf{u}_c - \theta_e\mathbf{u}_e$$

$[\mathbf{v}]_\times$: anti-symetrix matrix of **v** such that $[\mathbf{v}]_\times\,\mathbf{u} = \mathbf{v}\times\mathbf{u}$
of rank 2 ; at least a third orientation k is necessary

Once $\theta\mathbf{u}$ is known, ${}^e\mathbf{R}_c$ is known (Rodrigues formula)

$${}^e\mathbf{R}_c = \cos\theta\,\mathbf{I}_3 + \sin\theta\,[\mathbf{u}]_\times + (1-\cos\theta)\,\mathbf{u}\mathbf{u}^\top$$

How to go from R to $\theta\mathbf{u}$

From Rodrigues formula

$$\mathbf{R} = \cos\theta\, \mathbf{I}_3 + \sin\theta\, [\mathbf{u}]_\times + (1-\cos\theta)\mathbf{u}\mathbf{u}^\top$$

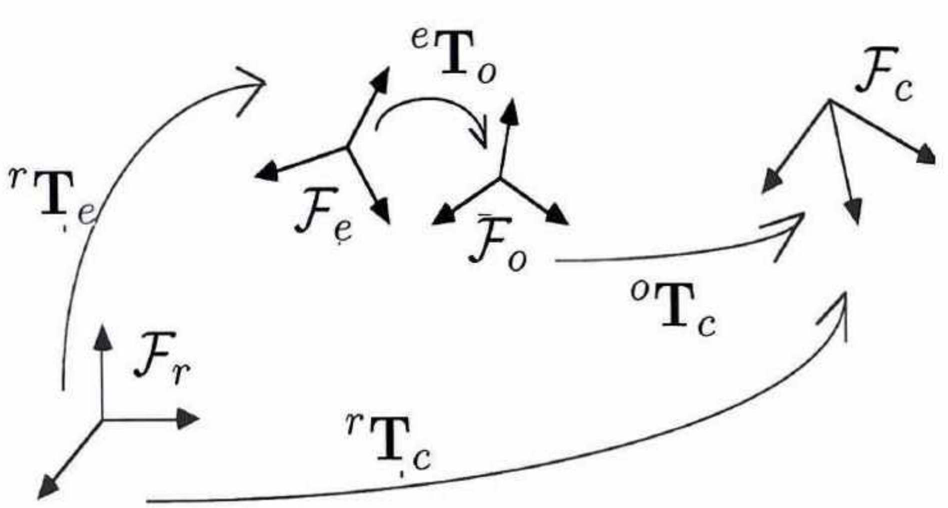
we easily obtain

$$\begin{cases} \theta &= \mathrm{Arccos}\left(\mathrm{tr}\,\mathbf{R} - 1\right)/2 \\ [\theta\mathbf{u}]_\times &= \frac{1}{2\,\mathrm{sinc}\,\theta}\left(\mathbf{R} - \mathbf{R}^\top\right) \end{cases}$$

where $\mathrm{sinc}\,\theta = \dfrac{\sin\theta}{\theta}$ $\qquad$ $(\,\mathrm{sinc}\,0 = 1)$

For $\theta = \pi$ $\quad \begin{cases} u_x &= \sqrt{(1+r_{11})/2} \\ u_y &= \sqrt{(1+r_{22})/2} \\ u_z &= \sqrt{(1+r_{33})/2} \end{cases}$

Hand-eye calibration: eye-to-hand configuration



Same problem: $^r\mathbf{T}_c \;=\; {}^r\mathbf{T}_{e_i}\,{}^e\mathbf{T}_o\,{}^c\mathbf{T}_{o_i}^{-1} \;=\; {}^r\mathbf{T}_{e_j}\,{}^e\mathbf{T}_o\,{}^c\mathbf{T}_{o_j}^{-1}$

Same resolution

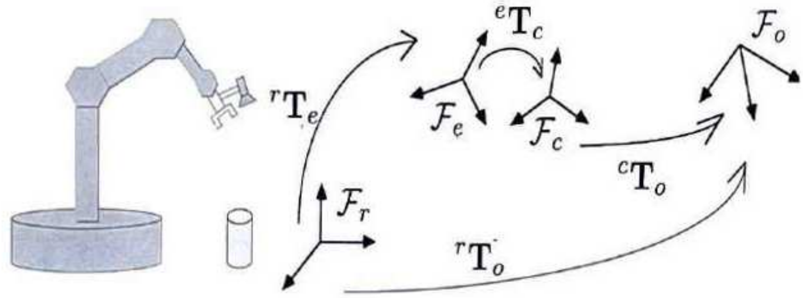
Open loop robot positioning

For a given desired pose ${}^{c^*}\mathbf{T}_o$

- Compute only once the displacement to be done

$${}^{c}\mathbf{T}_{c^*} = {}^{c}\mathbf{T}_o \, {}^{c^*}\mathbf{T}_o^{-1}$$

or $${}^{e}\mathbf{T}_{e^*} = {}^{e}\mathbf{T}_c \, {}^{c}\mathbf{T}_o \, {}^{c^*}\mathbf{T}_o^{-1} \, {}^{e}\mathbf{T}_c^{-1}$$



Advantages:

- Only one image to be processed and one very fast displacement to be achieved if the full system is perfectly calibrated

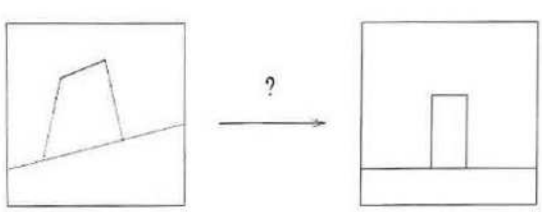
Drawbacks:

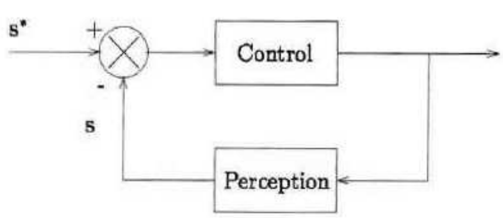
- Camera + hand-eye calibration + 3D localization needed
- Not robust to modeling and calibration errors

Better approach: closed-loop sensor-based control: visual servoing

What is visual servoing?

Vision-based closed loop control of a dynamic system





Advantages:

- Positioning accuracy
- Robustness with respect to modeling & calibration errors
- Reactive to changes (target tracking)
- Alternative to SLAM: achieve a task with the minimal information required

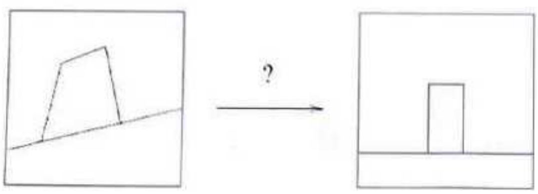
Drawbacks:

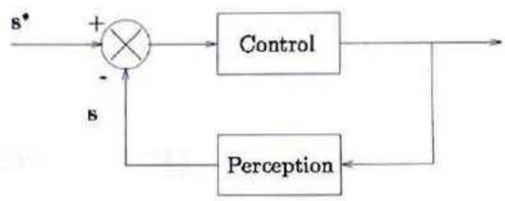
- Need many images to be processed

What is visual servoing?

Usual steps:

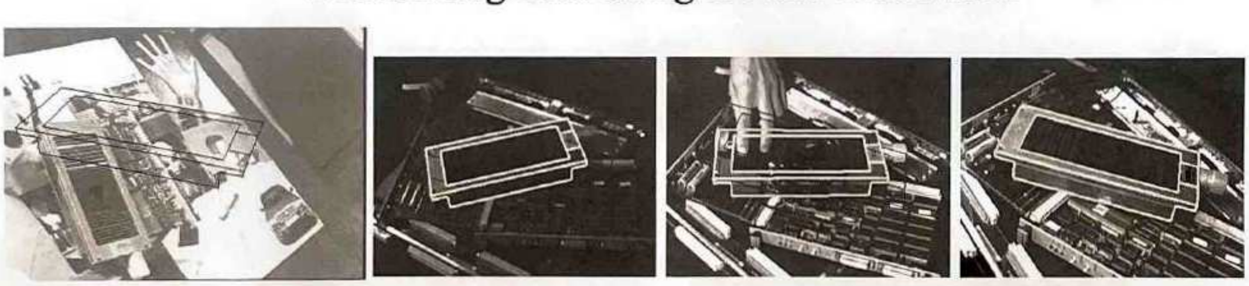
- extract visual measurements near video rate (matching then tracking)
- design adequate visual features from the available measurements: i.e., select *k* visual features to control *m* dof $(k \geq m, m \leq 6)$
- design a control scheme to regulate the error $(\mathbf{s} - \mathbf{s}^*)$ to 0
- taking into account the system and environment constraints for an adequate system behavior (stability, robustness, ...)





Pedestrian tracking using a pan/tilt camera

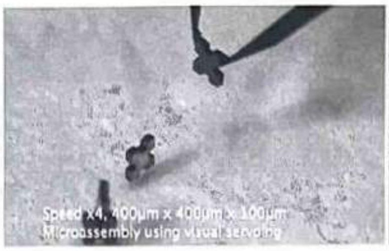
Positioning task using a 6 dof robot arm



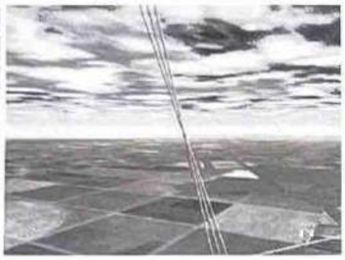
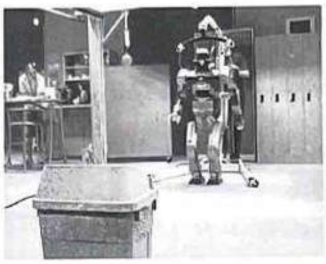
A wide spectrum of applications

Just need a camera and a robot

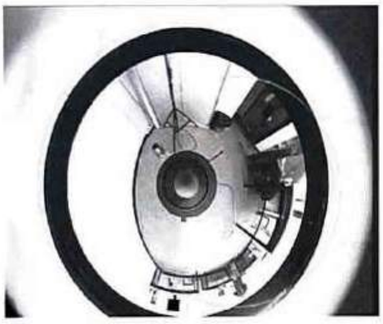
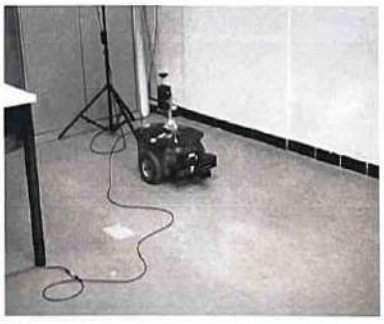






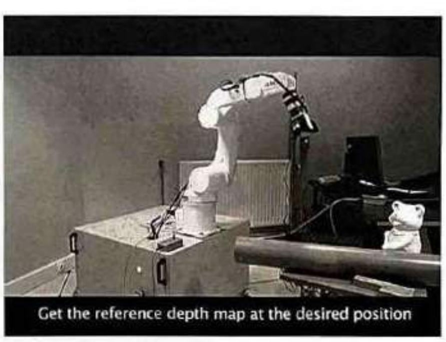




Whatever sort of vision sensor

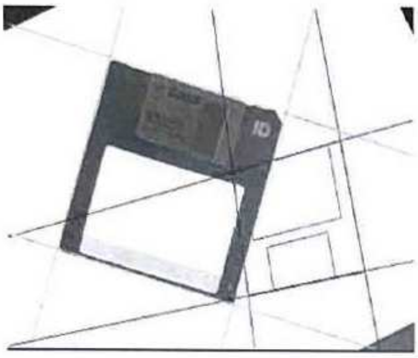
Omnidirectional camera

2D US probe



RGB-D sensor

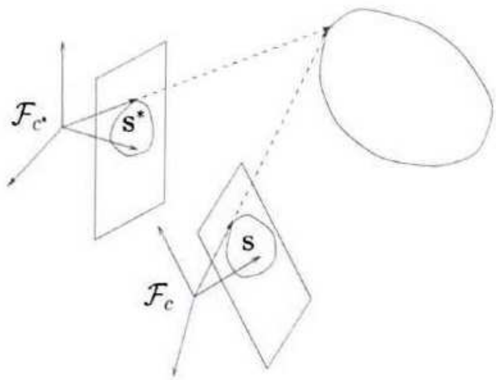
Just need a camera



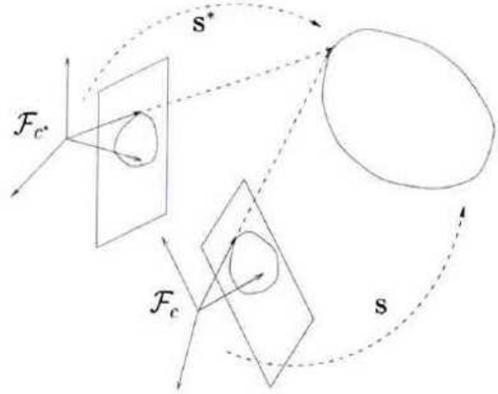
Pose estimation / 3D tracking can be formulated as Virtual Visual Servoing

Just need a computer

IBVS / PBVS



2D visual features



3D visual features

Modeling: same principle in all cases

Visual features: $\mathbf{s} = \mathbf{s}(\mathbf{p}(t)) \quad \Rightarrow \quad \dot{\mathbf{s}} = \mathbf{L_s}\, \mathbf{v}$ where:

- $\mathbf{L_s}$ = interaction matrix (similar to a Jacobian)
- $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega}) \in se_3$ = instantaneous camera velocity
  with 3 translational and 3 rotational components

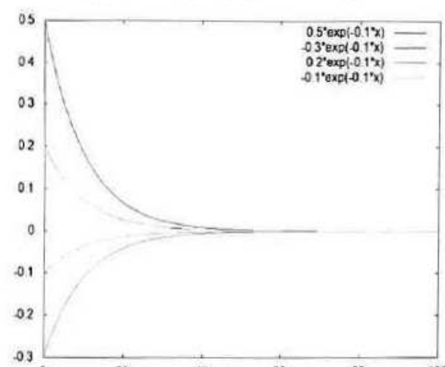
Expected behavior for positioning tasks

Basic behavior:

exponential decoupled decrease of the error $\mathbf{e}(t) = (\mathbf{s}(t) - \mathbf{s}^*)$

$$\mathbf{e}(t) = \mathbf{e}(t_0) \exp -\lambda t$$

$$\Rightarrow \dot{\mathbf{e}} = -\lambda \mathbf{e} \Leftrightarrow \dot{\mathbf{s}} = -\lambda(\mathbf{s} - \mathbf{s}^*)$$



Straight line trajectory in the feature space:

- For instance, if $\mathbf{s} = (x, y)$:

$$\begin{cases} x(t) - x^* = (x(t_0) - x^*) \exp -\lambda t \\ y(t) - y^* = (y(t_0) - y^*) \exp -\lambda t \end{cases} \Rightarrow \frac{x(t) - x^*}{x(t_0) - x^*} = \frac{y(t) - y^*}{y(t_0) - y^*}$$

$$\Rightarrow \frac{1}{x(t_0) - x^*} x(t) - \frac{1}{y(t_0) - y^*} y(t) + \frac{y^*}{y(t_0) - y^*} - \frac{x^*}{x(t_0) - x^*} = 0$$

- Same if $\mathbf{s} = (X, Y, Z)$ or whatever.

Principle of the control law for positioning tasks

If we would like $\dot{\mathbf{s}} = -\lambda(\mathbf{s} - \mathbf{s}^*)$ (exponential decoupled decrease)

From $\dot{\mathbf{s}} = \mathbf{L_s}\,\mathbf{v}$, we get

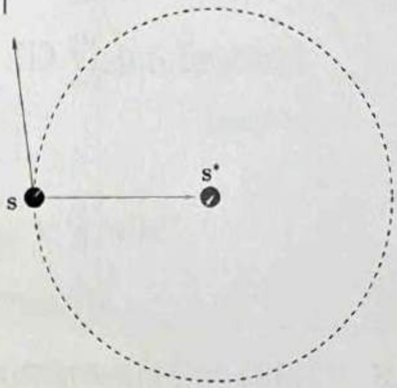
$$\mathbf{v} = -\lambda\,\widehat{\mathbf{L_s}}^{+}(\mathbf{s} - \mathbf{s}^*) \text{ with } \widehat{\mathbf{L}}_{\mathbf{s}(\mathbf{s},\mathbf{p},\mathbf{a})}$$

Closed-loop system: $\dot{\mathbf{s}} = \mathbf{L_s}\,\mathbf{v} = -\lambda\,\mathbf{L_s}\widehat{\mathbf{L_s}}^{+}(\mathbf{s} - \mathbf{s}^*)$

Lyapunov stability analysis: $\mathcal{L} = \frac{1}{2}\|\mathbf{s} - \mathbf{s}^*\|^2$

$$\dot{\mathcal{L}} = -\lambda(\mathbf{s} - \mathbf{s}^*)^\top\mathbf{L_s}\widehat{\mathbf{L_s}}^{+}(\mathbf{s} - \mathbf{s}^*)$$

- if $\mathbf{L_s}\widehat{\mathbf{L_s}}^{+} = \mathbf{I}$, perfect behavior
- if $\mathbf{L_s}\widehat{\mathbf{L_s}}^{+} > 0$, $\|\mathbf{s} - \mathbf{s}^*\|$ decreases
- if $\mathbf{L_s}\widehat{\mathbf{L_s}}^{+} < 0$, $\|\mathbf{s} - \mathbf{s}^*\|$ increases...



Sufficient condition for stability: $\mathbf{L_s}\widehat{\mathbf{L_s}}^{+} > 0$

Pseudo inverse $\mathbf{A}^+$ of $\mathbf{A}$

The $m \times k$ pseudo inverse $\mathbf{A}^+$ of any $k \times m$ matrix $\mathbf{A}$ is the only one matrix such that:

$$\begin{cases} \mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A} \\ \mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+ \\ (\mathbf{A}\mathbf{A}^+)^\top = \mathbf{A}\mathbf{A}^+ \\ (\mathbf{A}^+\mathbf{A})^\top = \mathbf{A}^+\mathbf{A} \end{cases}$$

Least squares problem:

If we look for $\mathbf{x}$ such that $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$ is minimal,
$\hat{\mathbf{x}} = \mathbf{A}^+\mathbf{b}$ is the solution such that $\|\hat{\mathbf{x}}\|$ is minimal.

Pseudo inverse $\mathbf{A}^+$ of A

In practice, let r = rank (A)

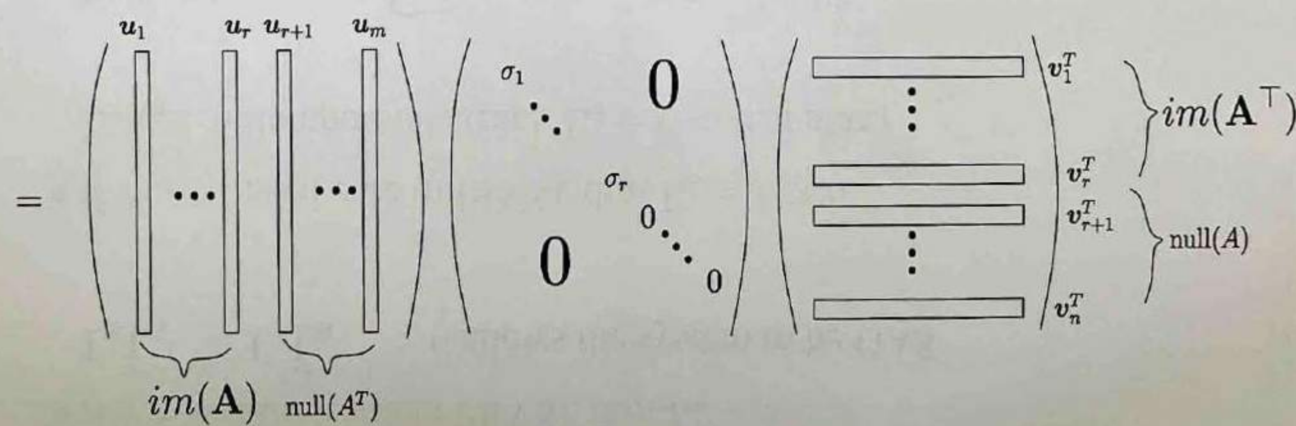
- If $k = m = r$, $\mathbf{A}^+ = \mathbf{A}^{-1}$
- If $k > m$ and $r = m$, $\mathbf{A}^+ = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top$
  In that case $\mathbf{A}^+\mathbf{A} = \mathbf{I}_m$ (left inverse)
- If $m > k$ and $r = k$, $\mathbf{A}^+ = \mathbf{A}^\top \left(\mathbf{A}\mathbf{A}^\top\right)^{-1}$
  In that case $\mathbf{A}\mathbf{A}^+ = \mathbf{I}_k$ (right inverse)
- In general, use of the Singular Value Decomposition: $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$
  where $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_k$, $\mathbf{V}\mathbf{V}^\top = \mathbf{I}_m$
  and $\mathbf{S}$ has only r values $\sigma_i \neq 0$ on its diagonal

$\mathbf{A}^+ = \mathbf{V}\mathbf{S}^+\mathbf{U}^\top$ where $\mathbf{S}^+$ has only r values $1/\sigma_i \neq 0$ on its diagonal

SVD decomposition

$$\underset{m\times n}{\mathbf{A}} = \underset{m\times m}{\mathbf{U}} \; \underset{m\times n}{\mathbf{S}} \; \underset{n\times n}{\mathbf{V}^\top}$$

$$\mathbf{V}^\top\mathbf{V} = \mathbf{I}_n \Rightarrow \begin{cases} \mathbf{v}_i^\top \mathbf{v}_i = 1, \forall i \\ \mathbf{v}_i^\top \mathbf{v}_j = 0, \forall i \neq j \end{cases}$$



Stability analysis with $\mathbf{v} = -\lambda\, \widehat{\mathbf{L_s}}^{+}(\mathbf{s}-\mathbf{s}^*)$: $\quad \mathbf{L_s}\widehat{\mathbf{L_s}}^{+} > 0$?

- if $k = 6$ (usual case in PBVS), dim $\mathbf{L_s} = 6 \times 6$
  $\mathbf{L_s}\widehat{\mathbf{L_s}}^{+} = \mathbf{L_s}\widehat{\mathbf{L_s}}^{-1} > 0$ allows the system to be GAS

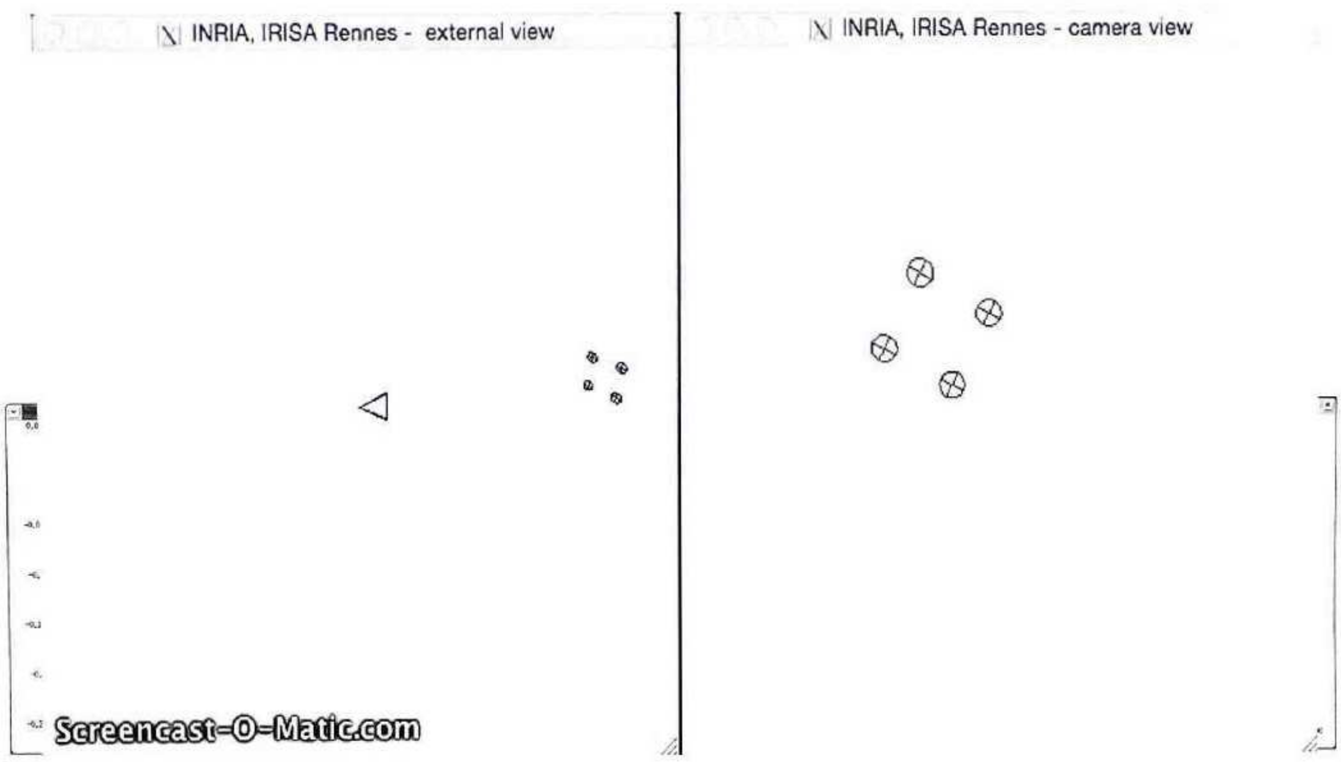
- if $k > 6$ (usual case in IBVS), dim $\mathbf{L_s} = k \times 6$
  $\mathbf{L_s}\widehat{\mathbf{L_s}}^{+} > 0$ impossible (rank $\mathbf{L_s} = 6 < k$ at max)

  By looking at $\mathbf{e} = \widehat{\mathbf{L_s}}^{+}(\mathbf{s}-\mathbf{s}^*)$ and $\mathcal{L} = \frac{1}{2}\|\mathbf{e}\|^2$

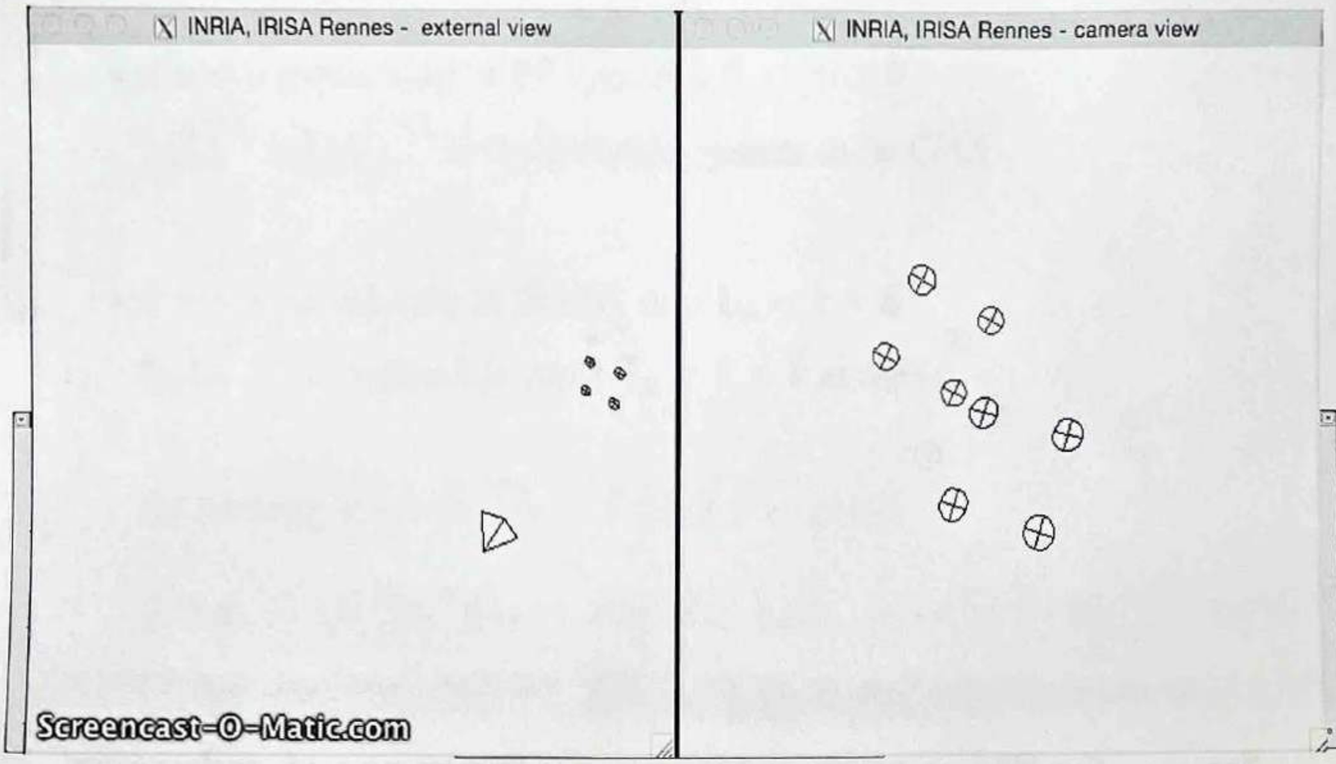
  $\dot{\mathcal{L}} = \mathbf{e}^\top\dot{\mathbf{e}} \approx \mathbf{e}^\top\widehat{\mathbf{L_s}}^{+}\mathbf{L_s}\mathbf{v} = -\lambda\mathbf{e}^\top\widehat{\mathbf{L_s}}^{+}\mathbf{L_s}\widehat{\mathbf{L_s}}^{+}(\mathbf{s}-\mathbf{s}^*) = -\lambda\mathbf{e}^\top\widehat{\mathbf{L_s}}^{+}\mathbf{L_s}\,\mathbf{e}$

  $\widehat{\mathbf{L_s}}^{+}\mathbf{L_s} > 0$ allows the system to be LAS (because of $\approx$)

Example 1: reaching a local minimum using $\widehat{\mathbf{L_s}}^{+} = \mathbf{L_s}^{+}$



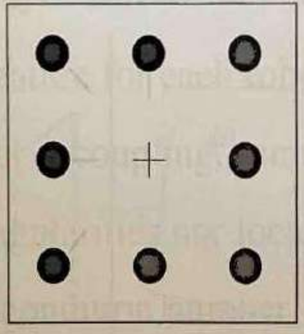
Example 2: reaching the global minimum using $\widehat{\mathbf{L_s}^+} = \mathbf{L_s^+}_{|s=s^*}$



Example 3: reaching a singularity of $\mathbf{L_s}$

Example : rotation of 180° around the optical axis
s composed of image points

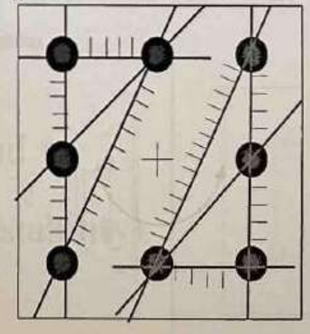
Cartesian coordinates



Using $\mathbf{L_s^+}$

At singularity, rank $\mathbf{L_s} = 2$

Cylindrical coordinates

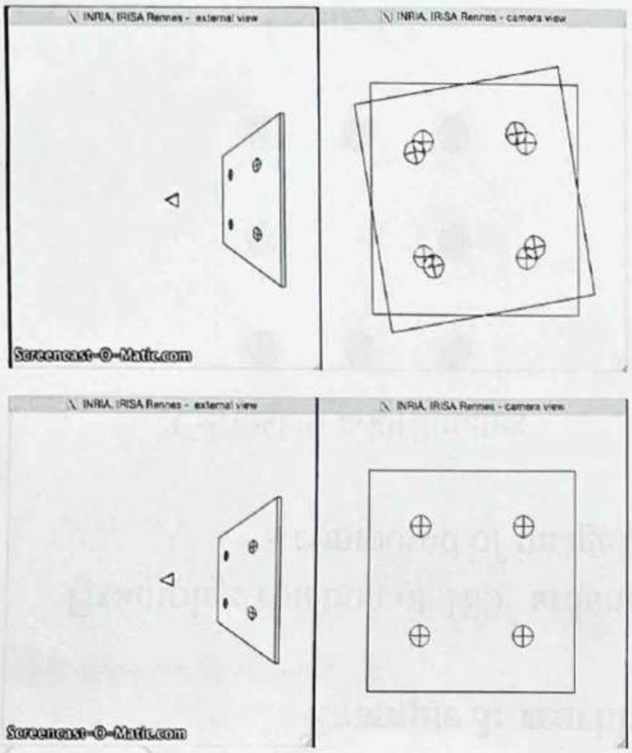


Using $\mathbf{L_s^+}$ or $\mathbf{L_s^+}_{|s=s^*}$

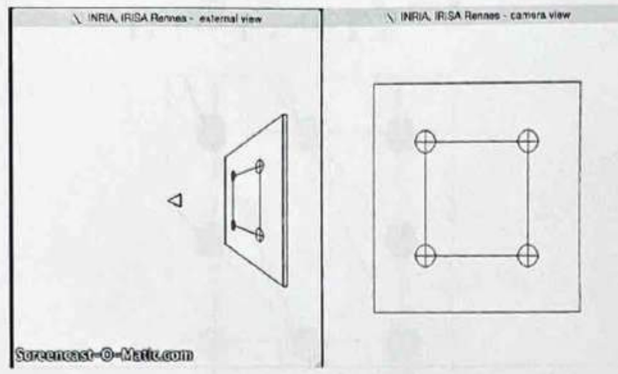
Perfect behavior

What are the best features?

Bad choice



Perfect choice (for this configuration)



Main goal: select adequate s for a given task

At least : select s such that rank $\mathbf{L_s} = m$ around s*

At most (yet a dream for 6 dof!) : select s such that $\mathbf{L_s} = \mathbf{I}_6$

- one feature for each robot dof
- perfect decoupling, same behavior of s and v
- no singularities nor local minima (global stability)
- ideal condition number
- control of a linear system

⇒ 1) Modeling issues

▷ Basics

▷ 3D visual features

▷ 2D visual features

2) Control issues

▷ Control of visual tasks - Target tracking

▷ Classification of the visual tasks

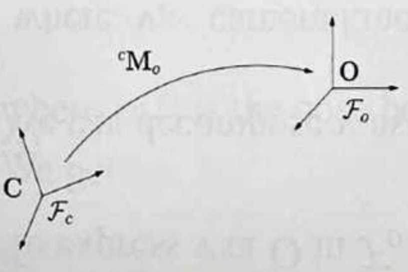
▷ Hybrid tasks

3) Applications and other issues

Change of frames

pose $\mathbf{p} \in SE_3$

$$\mathbf{X}_c = {}^c\mathbf{R}_o\,\mathbf{X}_o + {}^c\mathbf{t}_o$$



$\mathbf{X}_c$: coordinates of $\mathbf{X}$ in $\mathcal{F}_c$

$\mathbf{X}_o$: coordinates of $\mathbf{X}$ in $\mathcal{F}_o$

${}^c\mathbf{t}_o$: position of $\mathbf{O}$ in $\mathcal{F}_c$

${}^c\mathbf{R}_o$: rotation matrix between $\mathcal{F}_c$ and $\mathcal{F}_o$

$$\mathbf{R} = \cos\theta\,\mathbf{I}_3 + \sin\theta\,[\mathbf{u}]_\times + (1-\cos\theta)\,\mathbf{u}\mathbf{u}^\top$$

$\mathbf{u}$: rotation axis ($\|\mathbf{u}\| = 1$) θ : rotation angle around $\mathbf{u}$

$[\mathbf{u}]_\times$: skew symetric matrix related to $\mathbf{u}$: $[\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$

Kinematic screw (instantaneous velocity)

$\mathrm{v} = (\boldsymbol{v}, \boldsymbol{\omega})$: kinematic screw between the camera and the scene expressed at C in $\mathcal{F}_c$

$\boldsymbol{\omega}$: rotational velocity : $\quad [\boldsymbol{\omega}]_\times = {}^o\mathbf{R}_c^\top {}^o\dot{\mathbf{R}}_c = -{}^o\dot{\mathbf{R}}_c^\top {}^o\mathbf{R}_c$

$\boldsymbol{v}$: translational velocity at C : $\quad \boldsymbol{v}(\mathrm{O}) = -\boldsymbol{v}(\mathrm{C}) - \boldsymbol{\omega} \times \mathrm{CO}$

To express v at O in $\mathcal{F}_o$: ${}^o\mathrm{v} = {}^o\mathbf{V}_c\, \mathrm{v}$ with ${}^o\mathbf{V}_c = \begin{bmatrix} {}^o\mathbf{R}_c & [{}^o\mathbf{t}_c]_\times {}^o\mathbf{R}_c \\ \mathbf{0}_3 & {}^o\mathbf{R}_c \end{bmatrix}$

We can decompose v as $\mathbf{v} = \mathbf{v}_c - \mathbf{v}_o$

where $\mathbf{v}_c$: camera kinematic screw, expressed at C in $\mathcal{F}_c$

$\mathbf{v}_o$: object kinematic screw, expressed at C in $\mathcal{F}_c$

The interaction matrix

A set s of k visual features is given by a function from SE_3 to $\mathbb{R}^k$:

$$\mathrm{s} = \mathrm{s}(\mathbf{p}(t))$$

where $\mathbf{p}(t)$ is the pose between the camera and the scene.
We get

$$\dot{\mathrm{s}} = \frac{\partial \mathrm{s}}{\partial \mathbf{p}}\, \dot{\mathrm{p}} = \mathbf{L}_\mathrm{s}\, \mathrm{v}$$

where $\mathbf{L}_\mathrm{s}$ is the **interaction matrix** related to s
(Jacobian $\frac{\partial \mathrm{s}}{\partial \mathbf{p}} \approx \mathbf{L}_\mathrm{s}$ since $\dot{\mathrm{p}} = \mathbf{L}_\mathbf{p}\, \mathrm{v}$)

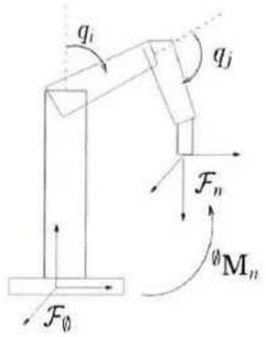
Using $\mathbf{v}_c$ and $\mathbf{v}_o$, we obtain :

$$\dot{\mathrm{s}} = \mathbf{L}_\mathrm{s}\, (\mathbf{v}_c - \mathbf{v}_o)$$

Robot Jacobian

Geometry of a robot arm defined by kinematics equations : $\mathbf{p}(t) = \mathbf{f}(\mathbf{q}(t))$

$\mathbf{q}$: joint positions ($\mathbf{q} \in \mathbb{R}^n$)
$\mathbf{p} \sim {}^{\emptyset}\mathbf{M}_n$: end-effector pose ($\mathbf{p} \in SE_3$)



End-effector kinematic screw given by :

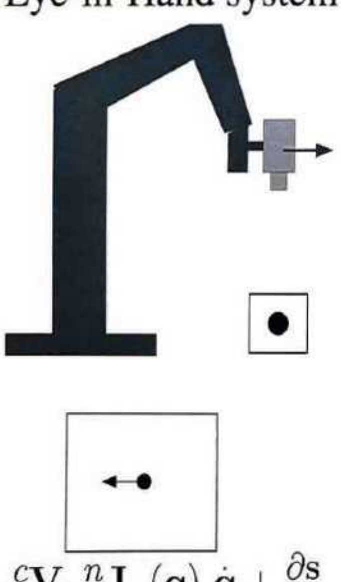
$\mathbf{v}_n = {}^n\mathbf{J}_n(\mathbf{q})\,\dot{\mathbf{q}}$ where ${}^n\mathbf{J}_n(\mathbf{q}) = \mathbf{M}_\mathbf{p}\,\frac{\partial \mathbf{p}}{\partial \mathbf{q}}$ is the robot jacobian

For velocity control, one computes $\dot{\mathbf{q}}^* = {}^n\mathbf{J}_n(\mathbf{q}^*)^{-1}\,\mathbf{v}_n^*$

Robot singularities = $\{\mathbf{q}_s, \det\left({}^n\mathbf{J}_n(\mathbf{q}_s)\right) = 0\}$

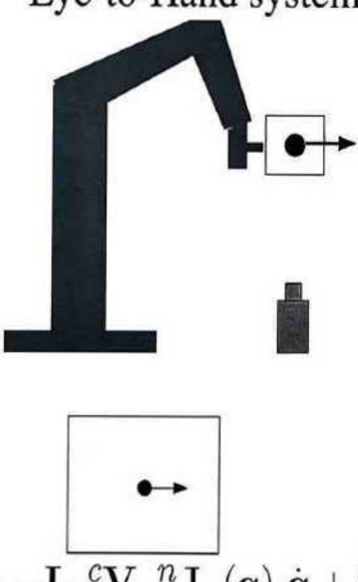
The feature Jacobian $\mathbf{J}_\mathbf{s}$

Eye-in-Hand system



$$\begin{aligned}\dot{\mathbf{s}} &= \mathbf{L}_\mathbf{s}{}^c\mathbf{V}_n{}^n\mathbf{J}_n(\mathbf{q})\,\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \\ &= \mathbf{J}_\mathbf{s}\,\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t}\end{aligned}$$

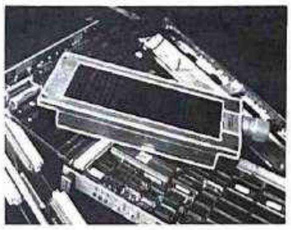
Eye-to-Hand system



$$\begin{aligned}\dot{\mathbf{s}} &= -\mathbf{L}_\mathbf{s}{}^c\mathbf{V}_n{}^n\mathbf{J}_n(\mathbf{q})\,\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \\ &= -\mathbf{L}_\mathbf{s}{}^c\mathbf{V}_\emptyset{}^\emptyset\mathbf{V}_n{}^n\mathbf{J}_n(\mathbf{q})\,\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t}\end{aligned}$$

- Modeling issues
  - ▷ Basics
  - ⇒ 3D visual features
  - ▷ 2D visual features

3D visual features with one camera



Based on pose estimation $\hat{\mathbf{p}}(t)$ from $\mathcal{F}_c$ to $\mathcal{F}_o$ using

- an image of the object: $\mathbf{x}(t)$
- the knowledge of the object 3D model: $\mathbf{X}$
- an estimation of the camera intrinsic parameters: x_c, y_c, f_x, f_y

$$\hat{\mathbf{p}}(t) = \hat{\mathbf{p}}(\mathbf{x}(t), \mathbf{X}, x_c, y_c, f_x, f_y)$$

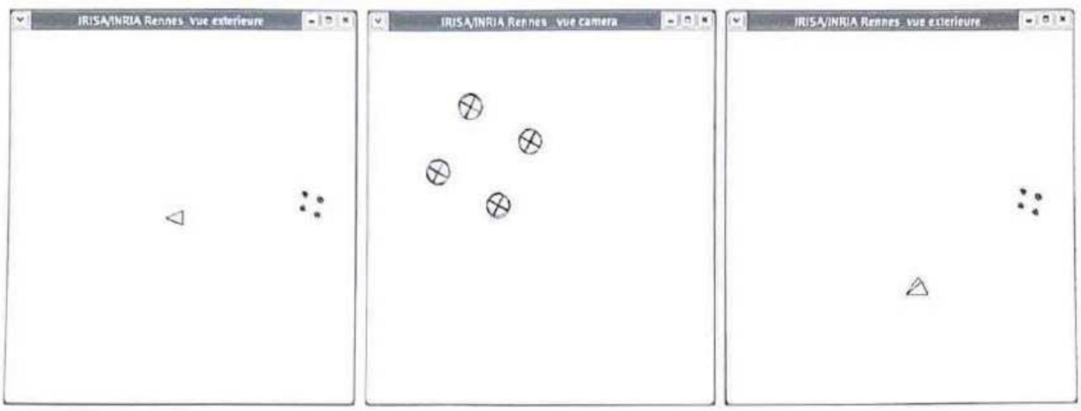
Pose estimation problem $\sim$ camera calibration problem
(intrinsic camera parameters already known)

3D visual features with one camera

Estimated pose $\hat{\mathbf{p}}(t) = \hat{\mathbf{p}}(\mathbf{x}(t), \mathbf{X}, x_c, y_c, f_x, f_y)$

$$\Rightarrow \quad \dot{\hat{\mathbf{p}}}(t) = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}\,\dot{\mathbf{x}} = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}\,\mathbf{L_x}\,\mathbf{v} \quad \Rightarrow \quad \mathbf{L}_{\hat{\mathbf{p}}} = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}\,\mathbf{L_x}$$

where $\mathbf{L_x}$ is known but $\frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}$ is unknown (and sometimes unstable)



3D visual features

Under the strong hypothesis that 3D estimation is perfect:

$$\frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{p}} = \mathbf{I}_6 \Rightarrow \dot{\hat{\mathbf{p}}} = \dot{\mathbf{p}} = \mathbf{L_p}\mathbf{v}$$

- representation $\theta\mathbf{u}$ of rotation ${}^{c^*}\mathbf{R}_c$

$$\mathbf{L}_{\theta \mathbf{u}} = \left[\, \mathbf{0}_3 \;\; \mathbf{L}_{\boldsymbol{\omega}} \,\right] \text{ where } \mathbf{L}_{\boldsymbol{\omega}} = \mathbf{I}_3 + \frac{\theta}{2}\,[\mathbf{u}]_\times + \left(1 - \frac{\text{sinc}\theta}{\text{sinc}^2\frac{\theta}{2}}\right)[\mathbf{u}]_\times^2$$

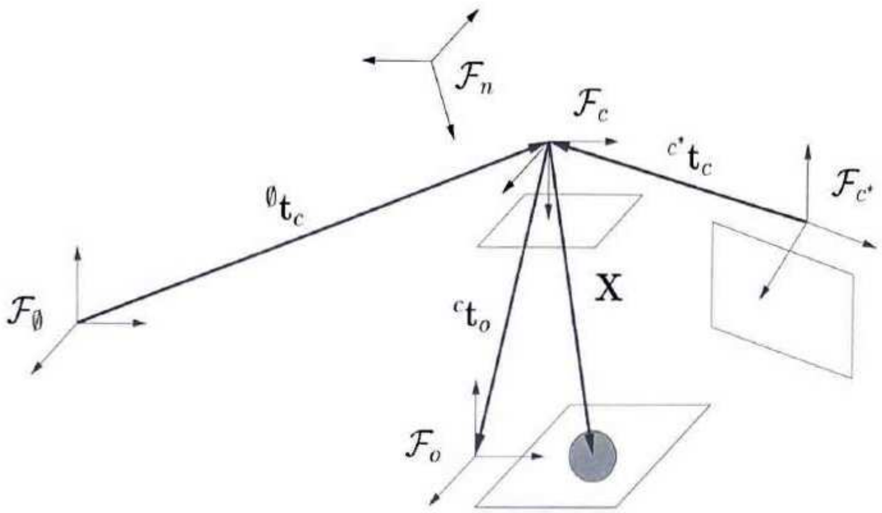
$\mathbf{L}_{\boldsymbol{\omega}}$ such that $\mathbf{L}_{\boldsymbol{\omega}}\theta\mathbf{u} = \mathbf{L}_{\boldsymbol{\omega}}^{-1}\,\theta\mathbf{u} = \theta\mathbf{u}$

- coordinates of a 3D point $\mathbf{X}$:

$\dot{\mathbf{X}} = \boldsymbol{v}(\mathbf{X}) = -\boldsymbol{v}(\mathbf{C}) - \boldsymbol{\omega} \times \mathbf{CX} = -\boldsymbol{v} + \mathbf{CX} \times \boldsymbol{\omega} = -\boldsymbol{v} + [\mathbf{X}]_\times \boldsymbol{\omega}$

$$\Rightarrow \mathbf{L_X} = \left[\, -\mathbf{I}_3 \;\; [\mathbf{X}]_\times \,\right] = \begin{bmatrix} -1 & 0 & 0 & 0 & -Z & Y \\ 0 & -1 & 0 & Z & 0 & -X \\ 0 & 0 & -1 & -Y & X & 0 \end{bmatrix}$$

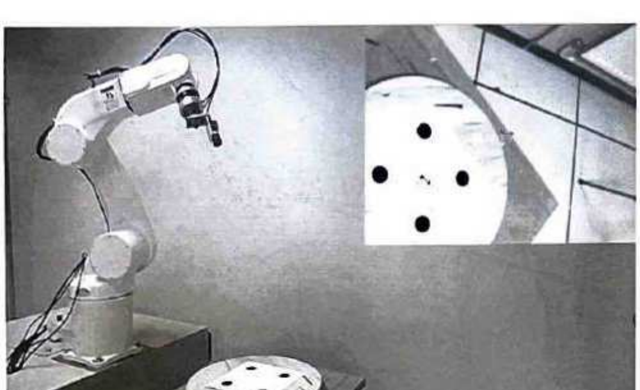
3D visual features for an eye-in-hand system

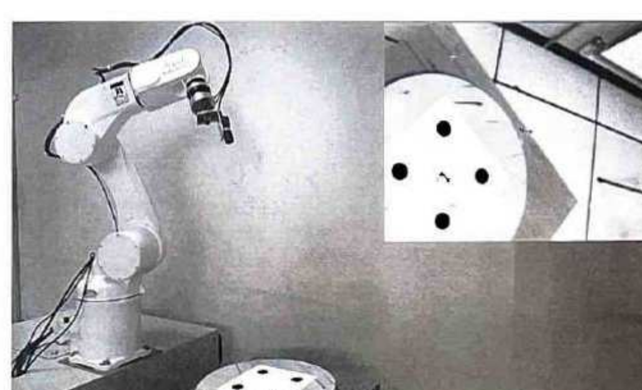


$$\mathbf{L}_{^c\mathbf{t}_o} = [\ -\mathbf{I}_3 \quad [^c\mathbf{t}_o]_\times\ ] \qquad \mathbf{L}_{^o\mathbf{t}_c} = [\ ^o\mathbf{R}_c \quad \mathbf{0}_3\ ]$$

$$\mathbf{L}_{^\emptyset\mathbf{t}_c} = [\ ^\emptyset\mathbf{R}_c \quad \mathbf{0}_3\ ] \qquad \mathbf{L}_{^{c^*}\mathbf{t}_c} = [\ ^{c^*}\mathbf{R}_c \quad \mathbf{0}_3\ ]$$

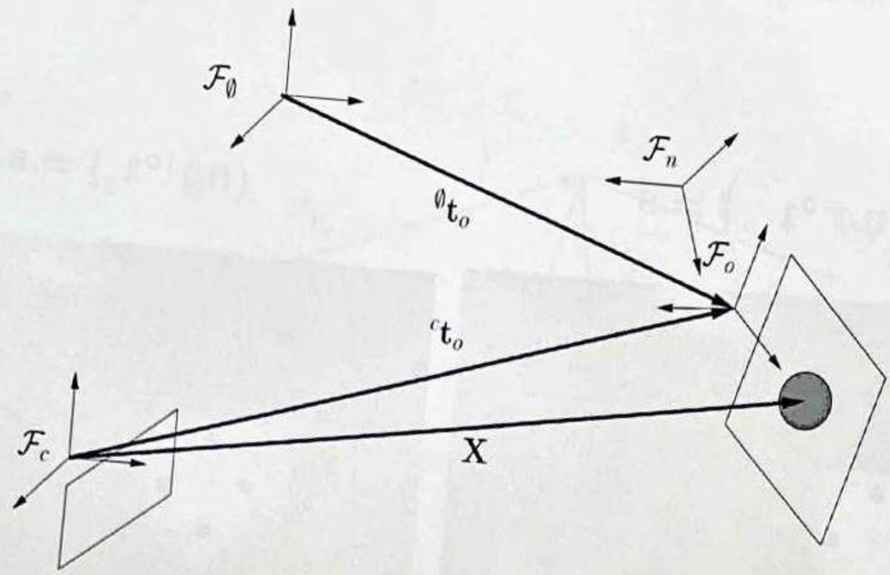
PBVS





$$\mathbf{s} = (^c\mathbf{t}_o, \theta\mathbf{u}) \qquad \mathbf{s} = \left(^{c^*}\mathbf{t}_c, \theta\mathbf{u}\right)$$

3D visual features for an eye-to-hand system

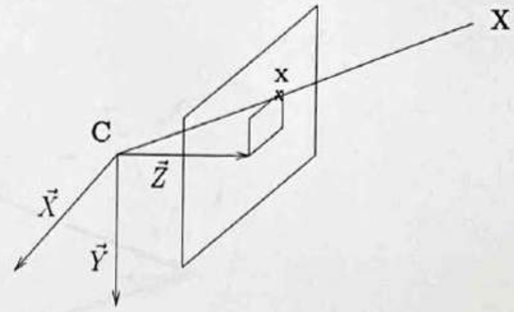


$$\mathbf{L}_{{}^c\mathbf{t}_o}\,{}^c\mathbf{V}_o = \left[\, -{}^c\mathbf{R}_o \;\; \mathbf{0}_3 \,\right] \qquad {}^{\emptyset}\dot{\mathbf{t}}_o = \left[\, \mathbf{I}_3 \;\; \mathbf{0}_3 \,\right]\, {}^{\emptyset}\mathbf{v}_o$$

- Modeling issues
  - ▷ Basics
  - ▷ 3D visual features
  - ⇒ 2D visual features

2D visual features: image point coordinates

Perspective projection : $\mathbf{x} = (x, y)$



$$x = X/Z \ , y = Y/Z$$

$$\Rightarrow \begin{cases} \dot{x} = \begin{bmatrix} 1/Z & 0 & -X/Z^2 \end{bmatrix} \dot{\mathbf{X}} \\ \dot{y} = \begin{bmatrix} 0 & 1/Z & -Y/Z^2 \end{bmatrix} \dot{\mathbf{X}} \end{cases}$$

Using a mobile camera and a fixed point:

$$\dot{\mathbf{X}} = \boldsymbol{v}(\mathbf{X}) = -\boldsymbol{v}(\mathbf{C}) - [\boldsymbol{\omega}]_\times \mathbf{CX} = \begin{bmatrix} -\mathbf{I}_3 & [\mathbf{X}]_\times \end{bmatrix} \mathbf{v}$$

We obtain:

$$\dot{\mathbf{x}} = \mathbf{L_x} \, \mathbf{v} \text{ where } \mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}$$

Image point expressed in pixels

$\mathbf{x} = (\, x \, , \, y \,) =$ image point coordinates expressed in meters
$\mathbf{x}_p = (\, x_p, y_p \,) =$ image point coordinates expressed in pixels

$$x_p = x_c + f_x \, x \ , \ y_p = y_c + f_y \, y$$

where $\mathbf{x}_c = (x_c, y_c)$ = principal point
and f_x, f_y = ratio between focal length f and pixel size.

$$\Rightarrow \mathbf{L}_{\mathbf{x}_p} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \mathbf{L_x}$$

$$= \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}$$

where $x = (x_p - x_c)/f_x \ , \ y = (y_p - y_c)/f_y$

Useful only for stability analysis wrt. camera calibration errors

2D visual features: image point coordinates

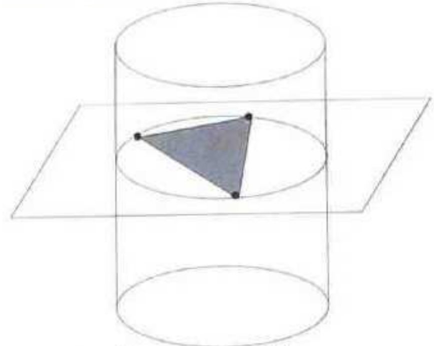
When $x = y = 0$ (principal point):

$$\mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & 0 & 0 & -1 & 0 \\ 0 & -1/Z & 0 & 1 & 0 & 0 \end{bmatrix}$$

A single point is adequate to control v_x or ω_y and v_y or ω_x

Using several points (at least 3) allows to control the 6 dof.

$$\mathbf{s} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \Rightarrow \mathbf{L_x} = \begin{bmatrix} \mathbf{L}_{\mathbf{x}_1} \\ \vdots \\ \mathbf{L}_{\mathbf{x}_n} \end{bmatrix}$$



Be careful to singularities in the interaction matrix ($\Rightarrow n \geq 4$)

What's about 3D information

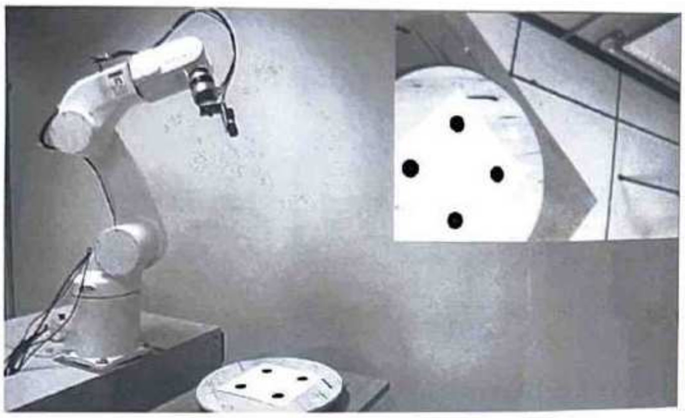
The depth Z_i of each point appears for the 3 translational dof (true $\forall \mathbf{s} \in$ 2D)

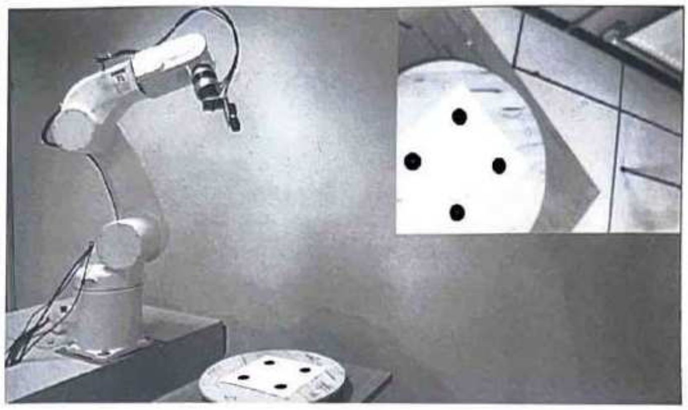
- Can be approximated
- Can be estimated:
  - by triangulation with stereovision
  - from pose if 3D object model available
  - up to a scale factor from epipolar geometry/homography with current & desired images
  - from structure from known motion

Note:

- For IBVS, the depth has an effect on the transient phase, not on the final accuracy (when the system is stable)
- For PBVS, 3D is involved for both the transient phase and the final accuracy, so problem in case of 3D noise

IBVS with points



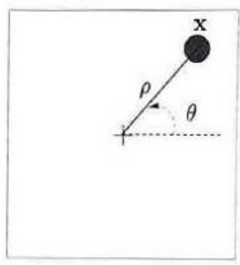


Using $\mathbf{v} = -\lambda \mathbf{L}^{+}_{\mathbf{s}(\mathbf{s},\mathbf{Z})}(\mathbf{s} - \mathbf{s}^*)$ Using $\mathbf{v} = -\lambda \mathbf{L}^{+}_{\mathbf{s}(\mathbf{s}^*,\mathbf{Z}^*)}(\mathbf{s} - \mathbf{s}^*)$

Image point in polar coordinates [Iwatsuki 02]

Use of (ρ, θ) for an image points instead of (x, y):

$$\rho = \sqrt{x^2 + y^2}\ ,\ \theta = \arctan \frac{y}{x}$$



Corresponding interaction matrix:

$$\mathbf{L}_\rho = [\ \frac{-\cos\theta}{Z}\ \ \frac{-\sin\theta}{Z}\ \ \frac{\rho}{Z}\ \ (1+\rho^2)\sin\theta\ \ -(1+\rho^2)\cos\theta\ \ \ 0\ ]$$

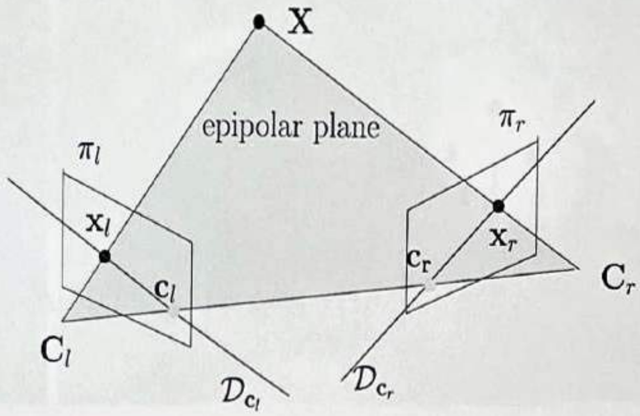
$$\mathbf{L}_\theta = [\ \frac{\sin\theta}{\rho Z}\ \ \frac{-\cos\theta}{\rho Z}\ \ 0\ \ \frac{\cos\theta}{\rho}\ \ \frac{\sin\theta}{\rho}\ \ -1\ ]$$

Better decoupling between v_z and ω_z

Be careful for the principal point ($x = y = \rho = 0,\ \theta$ undefined)

Image point for a stereovision system

$$\dot{\mathbf{x}}_l = \mathbf{L}_{\mathbf{x}_l}\,\mathbf{v}_l$$
$$\dot{\mathbf{x}}_r = \mathbf{L}_{\mathbf{x}_r}\,\mathbf{v}_r$$



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

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

$\mathbf{L}_{\mathbf{x}_l\mathbf{x}_r}$ is of rank 3 because of the epipolar constraint

- Generalization to multi-cameras systems immediate
- Probably better to use the coordinates of the 3D point

2D visual features: geometrical primitives

P_o : configuration of an *object feature* parameterized by $\mathbf{P_o}$
$p_i = \pi(P_o)$: configuration of an *image feature* parameterized by $\mathbf{p_i}$

Noting $\mathbf{P_o} = \varphi(P_o)$ and $\mathbf{p_i} = \psi(p_i)$, we get

$$\mathbf{p_i} = \nu(\mathbf{P_o}) = \psi \circ \pi \circ \varphi^{-1}(\mathbf{P_o})$$

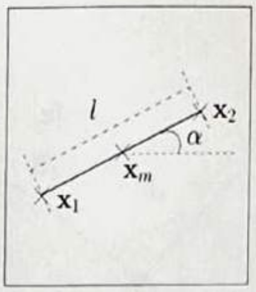
We also have $\mathbf{P_o} = \varphi \circ \delta(\mathbf{p}) \Rightarrow \mathbf{p_i} = \psi \circ \pi \circ \delta(\mathbf{p}) = \nu \circ \varphi \circ \delta(\mathbf{p})$

$$\begin{tikzcd}
\underset{(\mathbf{p})}{W \subseteq SE_3} \arrow[r, "\delta"'] & \underset{(P_o)}{U \subseteq \mathcal{P}_o} \arrow[r, "\pi"'] \arrow[d, "\varphi"] & \underset{(p_i)}{V \subseteq \mathcal{P}_i} \arrow[d, "\psi"] & \\
& \underset{(\mathbf{P_o})}{\mathbb{R}^n} \arrow[r, "\nu = \psi\circ\pi\circ\varphi^{-1}"'] & \underset{(\mathbf{p_i})}{\mathbb{R}^m} \arrow[r, "\sigma"'] & \underset{(\mathbf{s})}{\mathbb{R}^k}
\end{tikzcd}$$

Finally $\mathbf{s} = \sigma(\mathbf{p_i}) \Rightarrow \mathbf{L_s} = \dfrac{\partial \mathbf{s}}{\partial \mathbf{p_i}}\,\dfrac{\partial \mathbf{p_i}}{\partial \mathbf{P_o}}\,\mathbf{L_{P_o}}$

2D visual features: case of a segment

$$\mathbf{s} = \begin{bmatrix} x_m \\ y_m \\ l \\ \alpha \end{bmatrix} \text{ with } \begin{cases} x_m = (x_1 + x_2)/2 \\ y_m = (y_1 + y_2)/2 \\ l = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ \alpha = \arctan(y_1 - y_2)/(x_1 - x_2) \end{cases}$$



$$\begin{bmatrix} \mathbf{L}_{x_m} \\ \mathbf{L}_{y_m} \\ \mathbf{L}_l \\ \mathbf{L}_\alpha \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ \Delta x/l & \Delta y/l & -\Delta x/l & -\Delta y/l \\ -\Delta x/l^2 & \Delta x/l^2 & \Delta y/l^2 & -\Delta x/l^2 \end{bmatrix} \begin{bmatrix} \mathbf{L}_{x_1} \\ \mathbf{L}_{y_1} \\ \mathbf{L}_{x_2} \\ \mathbf{L}_{y_2} \end{bmatrix}$$

with $\Delta x = x_1 - x_2$ and $\Delta y = y_1 - y_2$.

Using $\begin{cases} x_1 = x_m + l\cos\alpha/2,\ y_1 = y_m + l\sin\alpha/2 \\ x_2 = x_m - l\cos\alpha/2,\ y_2 = y_m - l\sin\alpha/2 \end{cases}$, we get $\mathbf{L_s}(\mathbf{s}, Z_1, Z_2)$.

2D visual features: case of a segment

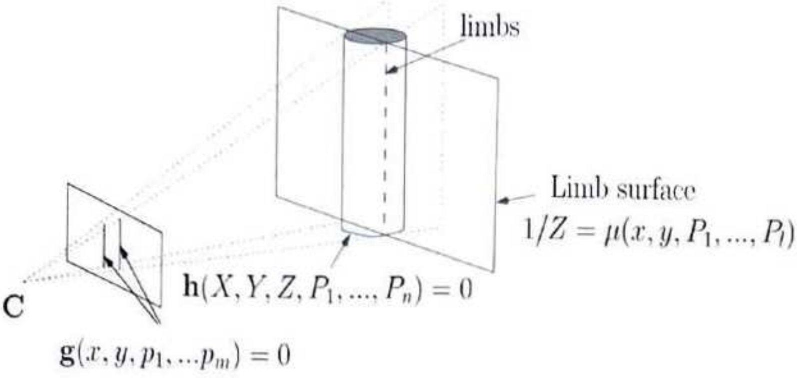
$$\begin{aligned} \mathbf{L}_{x_m} &= [\ -1/Z_m \quad 0 \quad x_m/Z_m + D\epsilon_x \quad x_{mwx} \quad -1 - x_{mwy} \quad y_m\ ] \\ \mathbf{L}_{y_m} &= [\ 0 \quad -1/Z_m \quad y_m/Z_m + D\epsilon_y \quad 1 + y_{mwx} \quad y_{mwy} \quad -x_m\ ] \\ \mathbf{L}_l &= [\ -Dc \quad -Ds \quad l/Z_m + D\epsilon_l \quad l_{wx} \quad l_{wy} \quad 0\ ] \\ \mathbf{L}_\alpha &= [\ Ds/l \quad -Dc/l \quad D\epsilon_\alpha \quad \alpha_{wx} \quad \alpha_{wy} \quad -1\ ] \end{aligned}$$

with $1/Z_m = (1/Z_1 + 1/Z_2)/2$ and $D = 1/Z_1 - 1/Z_2$

Nice triangular form for a segment parallel to the image plane ($D = 0$)

Exercise: Better parameterization: $\mathbf{s} = (x_m/l, y_m/l, 1/l, \alpha)$

Modeling geometrical primitives



3D primitive : $\mathbf{h}(\mathbf{X}, \mathbf{P_o}) = 0$

2D primitive : $\mathbf{g}(\mathbf{x}, \mathbf{p_i}) = 0$

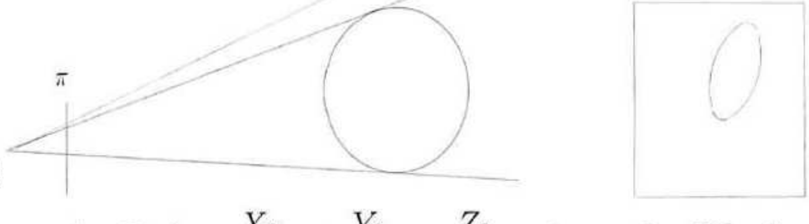
Limb surface : $\Rightarrow 1/Z = \mu(\mathbf{x}, \mathbf{P_o})$

- planar limb surface $\Leftrightarrow 1/Z = Ax + By + C$
- planar limb surface // image plane $\Leftrightarrow 1/Z = C \ (A = B = 0)$

2D visual features : case of the sphere

3D primitive : $\mathbf{h}(\mathbf{X}, \mathbf{P_o}) = (X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 - R^2 = 0$

$x = X/Z \ , \ y = Y/Z \Rightarrow K\frac{1}{Z^2} - 2(X_0x + Y_0y + Z_0)\frac{1}{Z} + x^2 + y^2 + 1 = 0$



$\Delta = 0 \Rightarrow \frac{1}{Z} = \mu(\mathbf{x}, \mathbf{P_o}) = \frac{X_0}{K}x + \frac{Y_0}{K}y + \frac{Z_0}{K}$ (eq. of a 3D plane)

$\Delta = 0 \Leftrightarrow (X_0x + Y_0y + Z_0)^2 - K(x^2 + y^2 + 1) = 0$

$\Leftrightarrow \mathbf{g}(\mathbf{x}, \mathbf{p_i}) = x^2 + a_1y^2 + 2a_2xy + 2a_3x + 2a_4y + a_5 = 0$

Image of a sphere = ellipse (circle if $X_0 = Y_0 = 0$)

Direct computation of the interaction matrix

$$\mathbf{L}_{\mathbf{p_i}} = \frac{\partial \mathbf{p_i}}{\partial \mathbf{P_o}} \mathbf{L}_{\mathbf{P_o}}$$

$$\begin{cases} a_1 = (R^2 - X_0^2 - Z_0^2)/(R^2 - Y_0^2 - Z_0^2) \\ a_2 = X_0 Y_0/(R^2 - Y_0^2 - Z_0^2) \\ \dots \end{cases} \Rightarrow \frac{\partial \mathbf{p_i}}{\partial \mathbf{P_o}} : \begin{cases} \dot{a}_1 = (-2X_0\dot{X}_0 - 2Z_0\dot{Z}_0)/(R^2 - Y_0^2 - Z_0^2) - \dots \\ \dot{a}_2 = \dots \\ \dots \end{cases}$$

$$\mathbf{L}_{\mathbf{P_o}} : \dot{\mathbf{X}}_0 = \begin{bmatrix} \dot{X}_0 \\ \dot{Y}_0 \\ \dot{Z}_0 \end{bmatrix} = \begin{bmatrix} -\mathbb{I}_3 & [\mathbf{X}_0]_\times \end{bmatrix} \mathbf{v} \Rightarrow \mathbf{L}_{\mathbf{P_o}} = \begin{bmatrix} -\mathbb{I}_3 & [\mathbf{X}_0]_\times \end{bmatrix}$$

We always have rank $\mathbf{L}_{\mathbf{p_i}}$ = rank $\mathbf{L}_{\mathbf{P_o}}$ = 3

Results in $\mathbf{L}_{\mathbf{p_i}}$ are function of 3D data $\mathbf{P_o} = (X_0, Y_0, Z_0, R)$

General method

$$\mathrm{g}(\mathbf{x}, \mathbf{p_i}) = 0 \Rightarrow \dot{\mathrm{g}}(\mathbf{x}, \mathbf{p_i}) = 0 \Leftrightarrow \frac{\partial \mathrm{g}}{\partial \mathbf{p_i}}(\mathbf{x}, \mathbf{p_i})\ \dot{\mathbf{p}}_\mathbf{i} = -\frac{\partial \mathrm{g}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{p_i})\ \dot{\mathbf{x}}\ ,\ \forall \mathbf{x} \in p_i$$

We have $\dot{\mathbf{x}} = \mathbf{L}_{xy}(\mathbf{x}, 1/Z)\ \mathbf{v} = \mathbf{L}_{xy}(\mathbf{x}, \mathbf{P_o})\ \mathbf{v}$

$$\Rightarrow \frac{\partial \mathrm{g}}{\partial \mathbf{p_i}}(\mathbf{x}, \mathbf{p_i})\ \dot{\mathbf{p}}_\mathbf{i} = -\frac{\partial \mathrm{g}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{p_i})\ \mathbf{L}_{xy}(\mathbf{x}, \mathbf{P_o})\ \mathbf{v}\ ,\ \forall \mathbf{x} \in p_i$$

If dim($\mathbf{p_i}$) = dim(p_i) = m, using m points of p_i,
we obtain a $m \times m$ linear system:

$$\mathbf{L}_{\mathbf{p_i}}(\mathbf{p_i}, \mathbf{P_o}) = \begin{bmatrix} \alpha_1(\mathbf{p_i}) \\ \vdots \\ \alpha_m(\mathbf{p_i}) \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\beta}_1(\mathbf{p_i}, \mathbf{P_o}) \\ \vdots \\ \boldsymbol{\beta}_m(\mathbf{p_i}, \mathbf{P_o}) \end{bmatrix} \text{ with } \begin{cases} \alpha_i(\mathbf{p_i}) = \frac{\partial \mathrm{g}}{\partial \mathbf{p_i}}(\mathbf{x}_i, \mathbf{p_i}), i = 1 \text{ to } m \\ \boldsymbol{\beta}_i(\mathbf{p_i}, \mathbf{P_o}) = -\frac{\partial \mathrm{g}}{\partial \mathbf{x}}(\mathbf{x}_i, \mathbf{p_i})\mathbf{L}_{xy}(\mathbf{x}_i, \mathbf{P_o}), \\ \quad i = 1 \text{ to } m \end{cases}$$

2D visual features : case of straight lines

$$\mathbf{h}(\mathbf{X},\mathbf{P_o}) = \begin{cases} h_1 = A_1X + B_1Y + C_1Z = 0 \\ h_2 = A_2X + B_2Y + C_2Z + D_2 = 0 \end{cases}$$

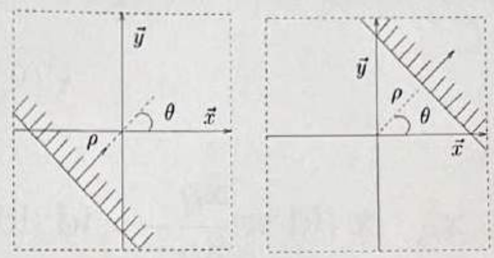
We obtain :

- function $\frac{1}{Z} = \mu(\mathbf{x},\mathbf{P_o})$ from $h_2 : 1/Z = Ax + By + C$
  with $A = -A_2/D_2, B = -B_2/D_2, C = -C_2/D_2$
- 2D straight line $\mathcal{D} : A_1x + B_1y + C_1 = 0$

Minimal parameterization $\mathbf{p_i} = (\rho, \theta)$

$\mathbf{g}(\mathbf{x},\mathbf{p_i}) = x\cos\theta + y\sin\theta - \rho = 0$

with $\theta = \arctan(B_1/A_1)$
and $\rho = -C_1/\sqrt{A_1^2 + B_1^2}$.



$\rho < 0$ $\rho > 0$

Computation of the interaction matrix

$\mathbf{g}(\mathbf{x},\mathbf{p_i}) = x\cos\theta + y\sin\theta - \rho = 0$

$$\dot{\mathbf{g}}(\mathbf{x},\mathbf{p_i}) = 0 \Rightarrow \dot{\rho} + (x\sin\theta - y\cos\theta)\,\dot{\theta} = \dot{x}\cos\theta + \dot{y}\sin\theta\ ,\ \forall \mathbf{x} \in \mathcal{D}$$

Using (for instance) points of $\mathcal{D}$ with coordinates

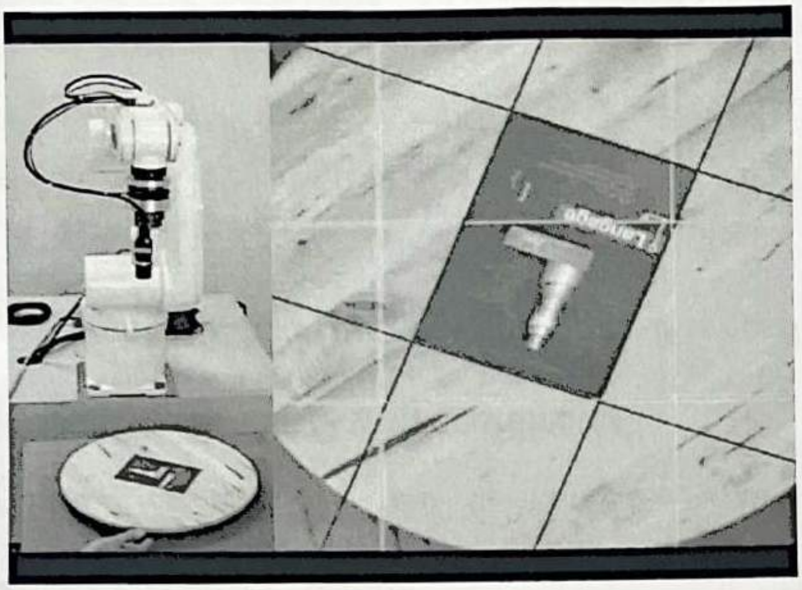
$$(\rho\cos\theta, \rho\sin\theta) \text{ and } (\rho\cos\theta + \sin\theta, \rho\sin\theta - \cos\theta)$$

we obtain

$$\begin{aligned} \mathbf{L}_\rho &= [\ \lambda_\rho c\theta \ \ \lambda_\rho s\theta \ \ -\lambda_\rho\rho \ \ (1+\rho^2)s\theta \ \ -(1+\rho^2)c\theta \ \ 0\ ] \\ \mathbf{L}_\theta &= [\ \lambda_\theta c\theta \ \ \lambda_\theta s\theta \ \ -\lambda_\theta\rho \ \ -\rho c\theta \ \ -\rho s\theta \ \ -1\ ] \end{aligned}$$

with $\lambda_\rho = -(A\rho c\theta + B\rho s\theta + C)$ and $\lambda_\theta = Bc\theta - As\theta$

IBVS with straight lines



2D visual features : case of a circle

$$\mathbf{h}(\mathbf{X}, \mathbf{P_o}) = \begin{cases} h_1 = (X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 - R^2 = 0 \\ h_2 = \alpha(X - X_0) + \beta(Y - Y_0) + \gamma(Z - Z_0) = 0 \end{cases}$$

$$\text{From } h_2 : \frac{1}{Z} = Ax + By + C \;\text{ with } \begin{cases} A = \alpha/(\alpha X_0 + \beta Y_0 + \gamma Z_0) \\ B = \beta/(\alpha X_0 + \beta Y_0 + \gamma Z_0) \\ C = \gamma/(\alpha X_0 + \beta Y_0 + \gamma Z_0) \end{cases}$$

$$\text{Using } h_1 : \mathbf{g}(\mathbf{x}, \mathbf{p_i}) = x^2 + a_1 y^2 + 2a_2 xy + 2a_3 x + 2a_4 y + a_5 = 0$$

Image of a circle = **ellipse** and a circle if $a_1 = 1$ et $a_2 = 0$, that is

$$A = B = 0 \quad \text{or} \quad \begin{cases} A = 2X_0/(X_0^2 + Y_0^2 + Z_0^2 - R^2)\,, \\ B = 2Y_0/(X_0^2 + Y_0^2 + Z_0^2 - R^2) \end{cases}$$

2D visual features : case of a circle

Better parameterization for ellipses : $\mathbf{p_i} = (x_g, y_g, \mu_{20}, \mu_{11}, \mu_{02})$

Centered **moments** : $\mu_{ij} = \int\int_{\mathcal{D}(t)} (x - x_g)^i (y - y_g)^j dxdy$

$\mathbf{L_{p_i}}$ is always of full rank 5, but for the centered circle $(x_g = y_g = \mu_{11} = A = B = 0, \mu_{20} = \mu_{02} = r^2)$ where:

$$\mathbf{L_{p_i}} = \begin{bmatrix} -1/Z_0 & 0 & 0 & 0 & -1-r^2 & 0 \\ 0 & -1/Z_0 & 0 & 1+r^2 & 0 & 0 \\ 0 & 0 & 2r^2/Z_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2r^2/Z_0 & 0 & 0 & 0 \end{bmatrix}$$

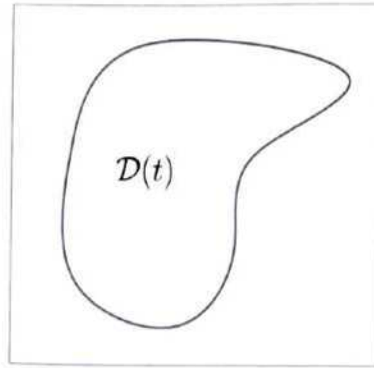
Summary

3D primitives	2D primitives	Parameterization
point	point	(x, y) or (ρ, θ)
segment	segment	(x_1, y_1, x_2, y_2) $(x_m/l, y_m/l, 1/l, \alpha)$
straight line	straight line	(ρ, θ)
circle	ellipse	$(x_g, y_g, \mu_{20}, \mu_{11}, \mu_{02})$
sphere	ellipse	$(x_g, y_g, a = \pi r^2)$
cylinder	2 straight lines	$(\rho_1, \theta_1, \rho_2, \theta_2)$
planar object	moments	$(a, x_g, y_g, \theta, ...)$

$\mathbf{L_s}$ also available for distance from a point to a straight line, angle between two straight lines, etc.

Moments definition

moments: $m_{ij} = \iint_{\mathcal{D}(t)} x^i y^j \, dx dy$



widely used in pattern recognition [Hu 1962]

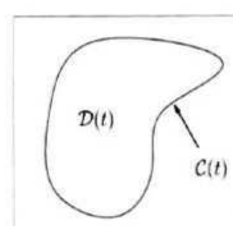
related to intuitive features:

- area a : m_{00}
- center of gravity $\mathbf{x_g}$: from m_{10} and m_{01}
- object orientation α and inertial axes : from m_{20}, m_{11}, and m_{02}
- skewness : from m_{30} and m_{03}

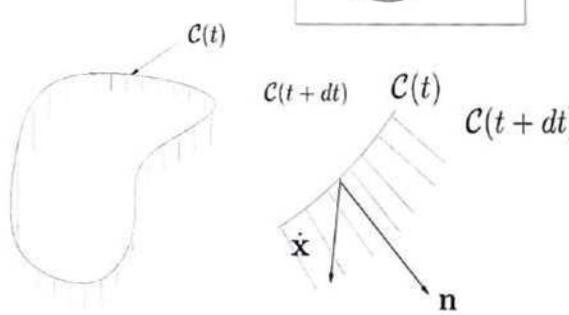
Interest in visual servoing: objects of complex or simple shape

Computation of the interaction matrix $\mathbf{L}_{m_{ij}}$

$$m_{ij}(t) = \iint_{\mathcal{D}(t)} f(x,y) dx dy \quad (f(x,y) = x^i y^j)$$



$$\Rightarrow \dot{m}_{ij} = \oint_{\mathcal{C}(t)} f(x,y) \, \dot{\mathbf{x}}^\top \mathbf{n} \, dl$$



Using Green's theorem :

$$\dot{m}_{ij} = \iint_{\mathcal{D}(t)} \text{div}[f(x,y)\dot{\mathbf{x}}] \, dx dy$$

$$\begin{aligned} \Rightarrow \dot{m}_{ij} &= \iint_{\mathcal{D}} [\frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial y}\dot{y} + f(x,y)(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y})] dx dy \\ &= \iint_{\mathcal{D}} [i x^{i-1} y^j \dot{x} + j x^i y^{j-1} \dot{y} + x^i y^j (\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y})] dx dy \end{aligned}$$

Computation of the interaction matrix $\mathbf{L}_{m_{ij}}$

$$\dot{m}_{ij} = \iint_{\mathcal{D}} [i x^{i-1} y^j \dot{x} + j x^i y^{j-1} \dot{y} + x^i y^j (\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y})] dx dy$$

$$\begin{cases} \dot{x} = [\,-1/Z \quad 0 \quad x/Z \quad xy \quad -1-x^2 \quad y\,]\,\mathbf{v} \\ \dot{y} = [\quad 0 \quad -1/Z \; y/Z \; 1+y^2 \quad -xy \quad -x\,]\,\mathbf{v} \end{cases}$$

For planar object: $1/Z = Ax + By + C$ from which we deduce:

$$\begin{cases} \frac{\partial \dot{x}}{\partial x} = [\,-A \quad 0 \quad (2Ax+By+C) \quad y \quad -2x \quad 0\,]\,\mathbf{v} \\ \\ \frac{\partial \dot{y}}{\partial y} = [\quad 0 \quad -B \; (Ax+2By+C) \; 2y \quad -x \quad 0\,]\,\mathbf{v} \end{cases}$$

($A = B = 0$ when the object is parallel to the image plane)

Interaction matrix for moments

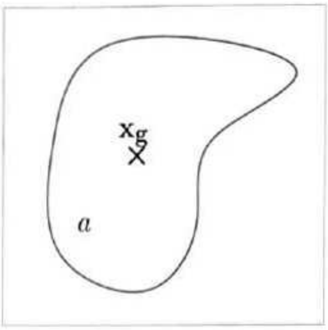
$$\mathbf{L}_{m_{ij}} = \left[\, m_{vx} \; m_{vy} \; m_{vz} \; m_{wx} \; m_{wy} \; m_{wz} \,\right]$$

$$\text{with} \begin{cases} m_{vx} = -i\,(Am_{ij} + Bm_{i-1,j+1} + Cm_{i-1,j}) - Am_{ij} \\ m_{vy} = -j\,(Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) - Bm_{ij} \\ m_{vz} = (i+j+3)(Am_{i+1,j} + Bm_{i,j+1} + Cm_{ij}) - Cm_{ij} \\ m_{wx} = \;\; (i+j+3)\, m_{i,j+1} + j\, m_{i,j-1} \\ m_{wy} = -(i+j+3)\, m_{i+1,j} - i\, m_{i-1,j} \\ m_{wz} = i\, m_{i-1,j+1} - j\, m_{i+1,j-1} \end{cases}$$

$\mathbf{L}_{m_{ij}}$ can be computed from moments of order less than $i + j + 2$ and from plane parameters A, B and C for translational components.

Area, Center of gravity

Area $a = m_{00}$



$$\mathbf{L}_a = [\; -aA \;\; -aB \;\; a(3/Z_g - C) \;\; 3ay_g \;\; -3ax_g \;\; 0 \;]$$

Object cog: $x_g = m_{10}/m_{00}, y_g = m_{01}/m_{00}$

$$\begin{aligned}
\mathbf{L}_{x_g} &= [\; -1/Z_g \quad 0 \quad x_g/Z_g + \epsilon_1 \quad x_g y_g + 4n_{11} \quad -(1 + x_g^2 + 4n_{20}) \quad y_g \;] \\
\mathbf{L}_{y_g} &= [\; 0 \quad -1/Z_g \quad y_g/Z_g + \epsilon_2 \quad 1 + y_g^2 + 4n_{02} \quad -x_g y_g - 4n_{11} \quad -x_g \;]
\end{aligned}$$

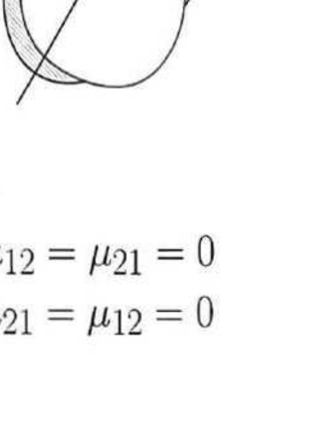
(generalization of the pure point case)

$$\begin{cases} 1/Z_g = Ax_g + By_g + C \\ \epsilon_1 = 4(An_{20} + Bn_{11}) \\ \epsilon_2 = 4(An_{11} + Bn_{02}) \end{cases} \quad n_{ij} = \mu_{ij}/a \text{ with } \begin{cases} \mu_{20} = m_{20} - ax_g^2 \\ \mu_{02} = m_{02} - ay_g^2 \\ \mu_{11} = m_{11} - ax_g y_g \end{cases}$$

Centered moments $\mu_{ij} = \iint_{\mathcal{D}} (x - x_g)^i (y - y_g)^j \, dx dy$

$$\mathbf{L}_{\mu_{ij}} = \left[\, \mu_{vx} \; \mu_{vy} \; \mu_{vz} \; \mu_{wx} \; \mu_{wy} \; \mu_{wz} \,\right]$$

$\mu_{vx} = \mu_{vy} = 0$ when $A = B = 0$



Object orientation $\alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$

$$\mathbf{L}_\alpha = \left[\, \alpha_{vx} \; \alpha_{vy} \; \alpha_{vz} \; \alpha_{wx} \; \alpha_{wy} \; -1 \,\right]$$

$$\begin{cases} \alpha_{vx} = \alpha_{vy} = \alpha_{vz} = 0 \text{ when } A = B = 0 \\ \alpha_{wx} = 0 \text{ when } x_g = y_g = 0 \text{ and } \mu_{03} = \mu_{12} = \mu_{21} = 0 \\ \alpha_{wy} = 0 \text{ when } x_g = y_g = 0 \text{ and } \mu_{30} = \mu_{21} = \mu_{12} = 0 \end{cases}$$

Cooking moments

- Normalization of $\mathbf{s} = (x_g, y_g, a)$:

$$\mathbf{s}_n = (x_n, y_n, a_n) \text{ with } a_n = 1/\sqrt{a}, x_n = x_g/\sqrt{a}, y_n = y_g/\sqrt{a}$$

$$\Rightarrow \mathbf{L}^{\|}_{\mathbf{x}_n} = \begin{bmatrix} -\kappa & 0 & 0 & a_n\epsilon_{11} & -a_n(1+\epsilon_{12}) & y_n \\ 0 & -\kappa & 0 & a_n(1+\epsilon_{21}) & -a_n\epsilon_{11} & -x_n \\ 0 & 0 & -\kappa & -3y_n/2 & 3x_n/2 & 0 \end{bmatrix} \quad (A{=}B{=}0)$$

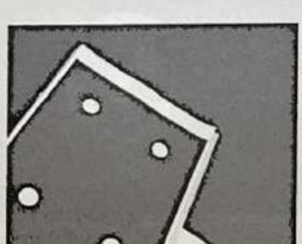
Pure image-based, but so near from position-based...

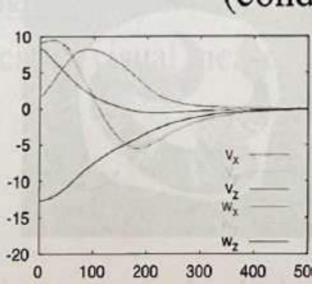
- Moment invariants: some combinations of moments are invariant to 2D translations, scale, and 2D rotation, so that by selecting adequately two of such combinations r_i and r_j:

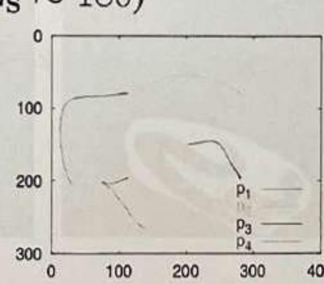
$$\mathbf{L}^{\|}_{r_i r_j} = \begin{bmatrix} 0 & 0 & 0 & r_{i_{wx}} & r_{i_{wy}} & 0 \\ 0 & 0 & 0 & r_{j_{wx}} & r_{j_{wy}} & 0 \end{bmatrix} \quad (A{=}B{=}0)$$

Interest of cooking visual features

Using the coordinates of 4 points for s (cond $\mathbf{L_s} \approx 180$)

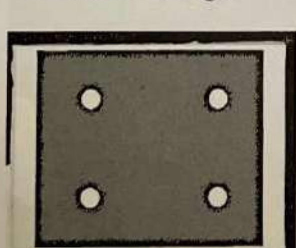


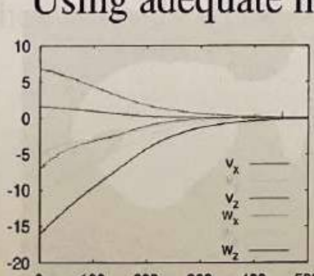


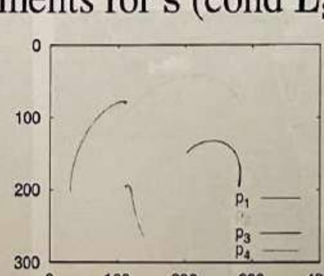


Initial image

Using adequate moments for s (cond $\mathbf{L_s} \approx 2$)





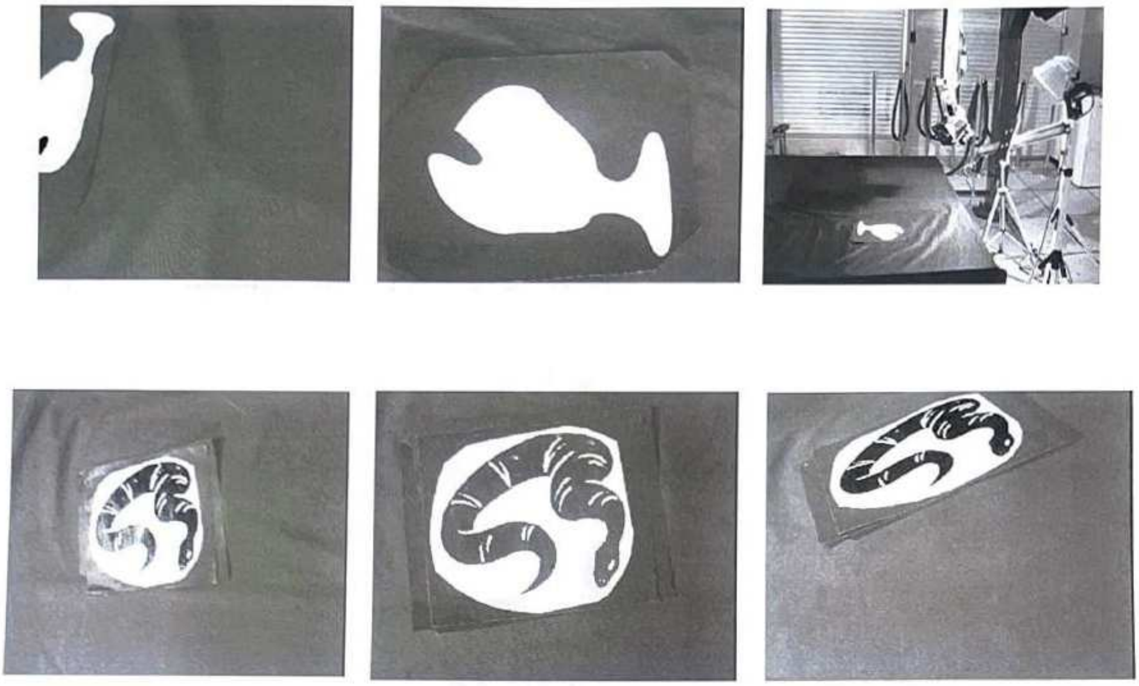


Desired image

Camera velocity

Image trajectories

Visual servoing with 2D moments



A new family of visual servoing: photometric VS

Remove the image processing part:

- No more extraction nor tracking visual measurements near video rate

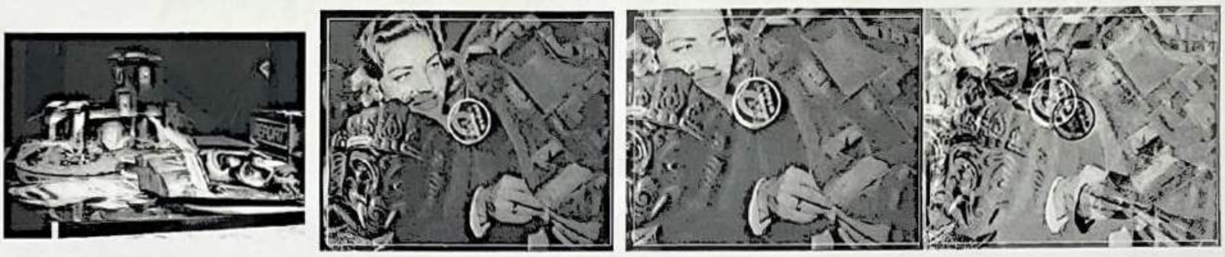
Advantages:

- Robustness to image processing errors and noise!
- End-to-end control (here without deep learning)

Photometric/direct/dense visual servoing

Visual features: intensity of each pixel $\mathbf{s} = \mathbf{I}(\mathbf{x}(t))$

$\mathbf{I}^*$ $\mathbf{I}$ $\mathbf{I} - \mathbf{I}^*$



Modeling: $\mathbf{L_I} = -\nabla \mathbf{I_x} \, \mathbf{L_x}$ (function of the image content)

$\mathcal{L} = \frac{1}{2} \|\mathbf{I} - \mathbf{I}^*\|$ highly non linear

Drawbacks: small convergence domain, strange robot trajectory

But no feature extraction, tracking nor matching

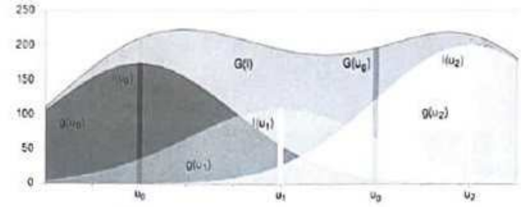
+ excellent positioning accuracy

Photometric visual servoing

Robustness to global illumination changes by using $\mathbf{s} = (\mathbf{I} - \bar{\mathbf{I}})/\sigma_{\mathbf{I}}^2$

Robustness to outliers (occlusion) by using $\mathbf{s} = \rho_{\mathbf{I}} \, \mathbf{I}$

Mixture of Gaussians



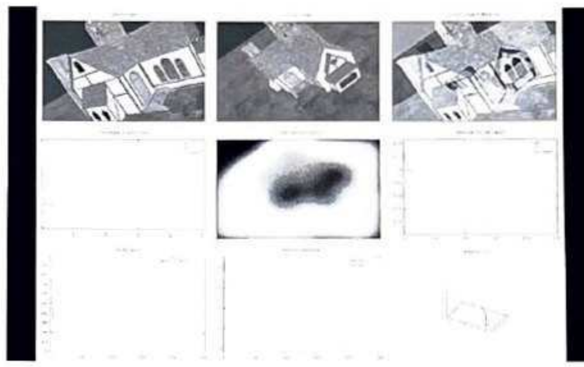
Enlarge the convergence domain

$$G(\mathbf{u_g}, \lambda_g) = \sum_{\mathbf{u_i} \in \mathbf{I}} I(\mathbf{u_i}) \ \exp\ \left(-\frac{(u_g - u_i)^2 + (v_g - v_i)^2}{2\lambda_g^2} \right)$$

$\lambda_g = 0.1$ $\lambda_g = 5$ $\lambda_g = 10$ $\lambda_g = 20$

Control simultaneously

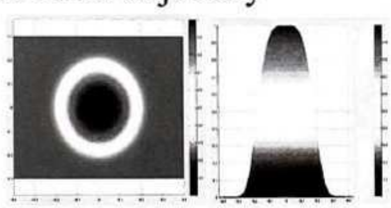
- the camera motion
- the expansion parameter
  (large to small)



Photometric moments

Going back to geometric features for enlarging the convergence domain
and improving the robot trajectory

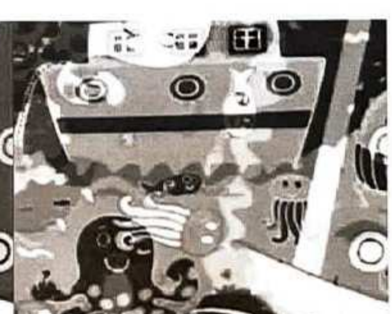
$$m_{pq} = \iint_{\pi} x^p y^q \, w(\mathbf{x}) \, I(\mathbf{x}, t) \, \mathrm{d}x \, \mathrm{d}y$$



Then select adequate moments (area, cog, main orientation, …)

$\mathbf{I}^*$

$\mathbf{I}$



$\mathbf{I} - \mathbf{I}^*$

- Modeling issues
  - ▷ Basics
  - ▷ 2D visual features
  - ▷ 3D visual features

$\Rightarrow$ Omni-directional vision sensor, vision + structured light

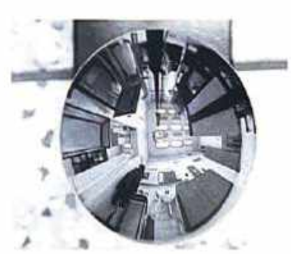
Modeling for omnidirectionnal vision [Nayar 01]

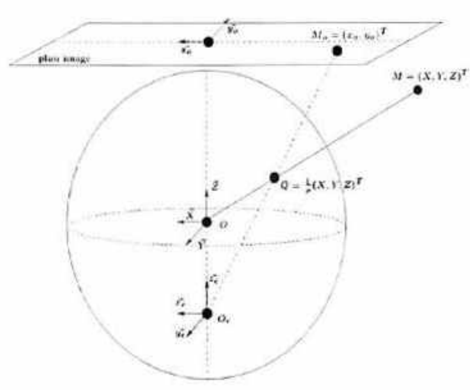
Single viewpoint systems

$$x_o = \frac{X}{\eta Z + \xi\sqrt{X^2+Y^2+Z^2}} \qquad (x = \frac{X}{Z})$$

$$y_o = \frac{Y}{\eta Z + \xi\sqrt{X^2+Y^2+Z^2}} \qquad (y = \frac{Y}{Z})$$

- $\eta = 1, \xi = 1$: parabolic mirror
- $\eta = 1, \xi = \xi_1$: planar mirror
- $\eta = 1, \xi = \xi_2$: hyperbolic mirror
- $\eta = 1, \xi = 0$: perspective projection
- $\eta = 0, \xi = 1$: spherical projection





Interaction matrix for a point

Using perspective projection:

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

Using omnidirectionnal vision [Barreto 2002]:

$$\mathbf{L}_{x_oy_o} = \begin{bmatrix} -\frac{1}{\rho}\left(\frac{\eta\gamma-\xi}{\nu} - \xi x_o^2\right) & \frac{\xi x_o y_o}{\rho} & \frac{x_o\gamma}{\rho} & \eta x_o y_o & -\frac{\eta-\xi\gamma}{\nu} - \eta x_o^2 & y_o \\ \frac{\xi x_o y_o}{\rho} & -\frac{1}{\rho}\left(\frac{\eta\gamma-\xi}{\nu} - \xi y_o^2\right) & \frac{y_o\gamma}{\rho} & \frac{\eta-\xi\gamma}{\nu} + \eta y_o^2 & -\eta x_o y_o & -x_o \end{bmatrix}$$

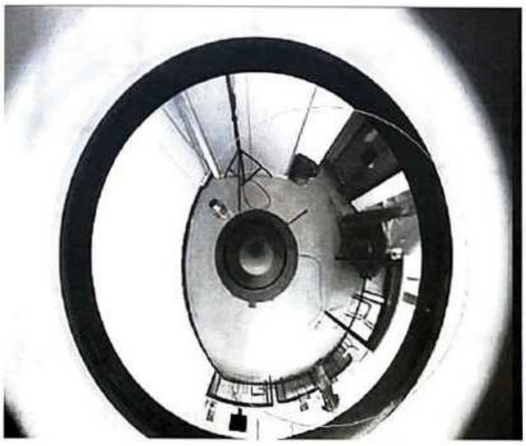
with $\nu = \eta^2 - \xi^2$, $\rho = \sqrt{X^2+Y^2+Z^2}$ and $\gamma = \sqrt{1+\nu(x_o^2+y_o^2)}$.

- For a parabolic mirror:
  $\eta = 1, \xi = 1, \nu = 0, \gamma = 1, \frac{\eta\gamma-\xi}{\nu} = \frac{1+x_o^2+y_o^2}{2}, \frac{\eta-\xi\gamma}{\nu} = \frac{1-x_o^2-y_o^2}{2}$

For the image of a straight line, i.e. an ellipse, see [Mezouar, IROS 04]

Considering other visual sensors: omnidirectional cameras

Use of straight lines in omnidirectional images

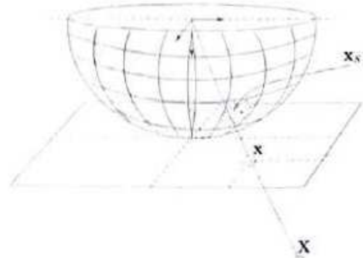


LASMEA

Modeling for spherical projection $(\eta = 0, \xi = 1)$

- can be used from a perspective sensor or an omnidirectional sensor

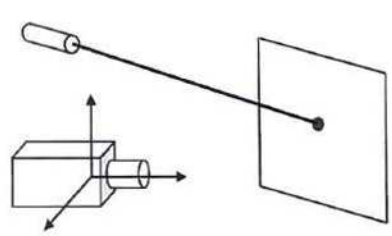
$\mathbf{x}_s = \mathbf{X}/\rho$ with $\rho = \sqrt{X^2 + Y^2 + Z^2}$



$$\mathbf{L}_{\mathbf{x}_s} = \begin{bmatrix} -\frac{1}{\rho}\left(1 - x_s^2\right) & \frac{x_s y_s}{\rho} & \frac{x_s z_s}{\rho} & 0 & -z_s & y_s \\ \frac{x_s y_s}{\rho} & -\frac{1}{\rho}\left(1 - y_s^2\right) & \frac{y_s z_s}{\rho} & z_s & 0 & -x_s \\ \frac{x_s z_s}{\rho} & \frac{y_s z_s}{\rho} & -\frac{1}{\rho}\left(1 - z_s^2\right) & -y_s & x_s & 0 \end{bmatrix}$$

Passivity property ($\|\dot{\mathbf{x}}_s\|$ independent of $\boldsymbol{\omega}$) [Hamel-Mahony 02]
Invariance property: for instance $\mathbf{L}_{a_s} = \left[\, a_x \;\; a_y \;\; a_z \;\; 0 \;\; 0 \;\; 0 \,\right]$

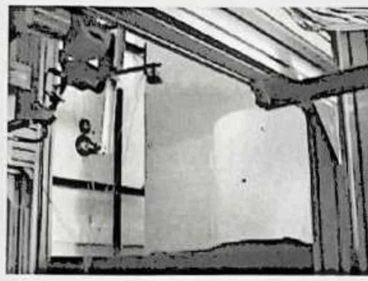
Modeling for coupling vision and structured light

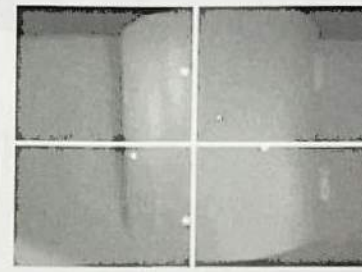


- Structured ligth rigidly linked to the object:
  - no change at all in the modeling
- Structured ligth rigidly linked to the camera:
  - Points, straight lines, ellipses, see [Motyl 92]
  - Points revisited, see [Pagès, IROS 04]

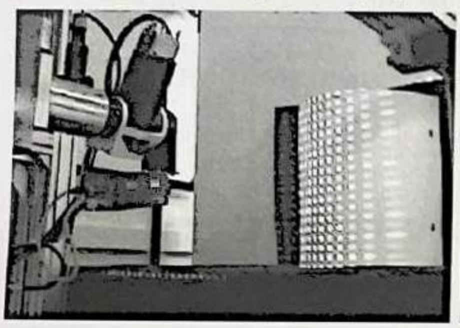
Considering other visual sensors: camera + structured light

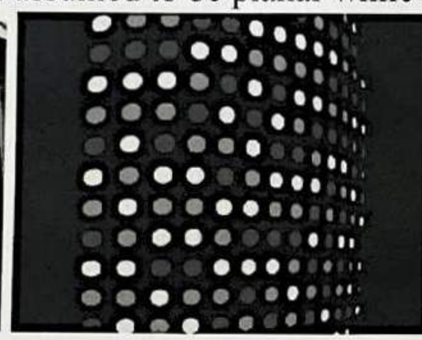
Eye-in-hand system





Eye-to-hand system (object assumed to be planar while it is not)





Desired image

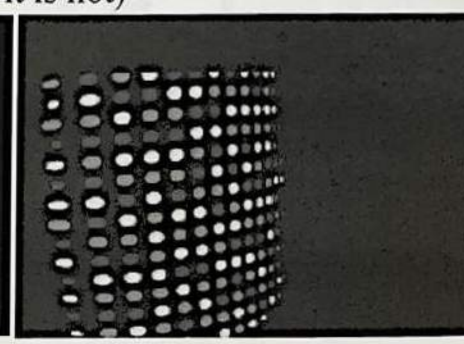


Image sequence

Other approach: direct numerical estimation

Using N measurements of $\mathbf{v_c}$ and corresponding $\dot{\mathbf{s}}$ around $\mathbf{s}^*$

- Off-line learning of $\mathbf{L_s}$:

  With 1 measurement, $\mathbf{L_s}\,\mathbf{v_c} = \dot{\mathbf{s}}$: k equations and $k \times 6$ unknowns

  With $N(\geq 6)$, $\mathbf{L_s}\,\mathbf{A} = \mathbf{B}$ where $\mathbf{A} \in \mathbb{R}^{6\times N}$ and $\mathbf{B} \in \mathbb{R}^{k\times N}$

  $$\Rightarrow \widehat{\mathbf{L_s}} = \mathbf{B}\mathbf{A}^+$$

- Off-line learning of $\mathbf{L_s^+}$ (better method):

  With 1 measurement, $\mathbf{L_s^+}\,\dot{\mathbf{s}} = \mathbf{v_c}$: 6 equations and $6 \times k$ unknowns

  With $N(\geq k)$, $\mathbf{L_s^+}\,\mathbf{B} = \mathbf{A} \Rightarrow \widehat{\mathbf{L_s^+}} = \mathbf{A}\mathbf{B}^+$

- Other methods: neural networks,...

Methods valid locally around $\mathbf{s}^*$ only since $\mathbf{L_s}$ is not constant.

Stability impossible to demonstrate

Other approach: direct numerical estimation

On-line iterative estimation (based on Broyden update):

$$\widehat{\mathbf{L_s}}(t+1) = \widehat{\mathbf{L_s}}(t) + \frac{\alpha}{\mathbf{v_c}^\top \mathbf{v_c}} \left(\dot{\mathbf{s}} - \widehat{\mathbf{L_s}}(t)\mathbf{v_c} \right) \mathbf{v_c}^\top$$

Be careful to initial value $\widehat{\mathbf{L_s}}(t0)$

Stability impossible to demonstrate

May be useful for unknown complex objects or unmodeled systems

1) Modeling issues

⇒ 2) Control issues

▷ Control of visual tasks ($m = n$)

▷ Target tracking

▷ Classification of the visual tasks

▷ Hybrid tasks ($m < n$)

Task specification

- Just specify **s*** or **s***(t)
  (such as an object has to appear at the center of the image)
- Specify a desired pose and deduce the value of **s***
  (but 3D model of the object + camera calibration needed)
- Teach by showing:
  1. go to the desired pose;
  2. acquire the corresponding image;
  3. determine **s*** in the same way as **s**(t).

Control in visual servoing

Regulation to 0 of an error (task function):

$$\mathbf{e}(\mathbf{p}(t)) = \mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^*$$

Numerous solutions:

- P, PI, PID controller [Weiss 87]
- Non linear control law [Hashimoto 93, Reyes 98]
- Optimal control (LQ, LQG) [Papanikilopoulos 93, Hashimoto 96]
- Predictive controller [Gangloff 98]
- Robust controller H_∞ [Khadraoui 96]
- etc.

Visual task function

With k visual features s, one constraints m robot dof ($m = \text{rank } \mathbf{L_s}$)

- If $m < n$, it is possible to consider a supplementary task (trajectory following, joint limits avoidance, etc.)
  $\Rightarrow$ **Hybrid tasks**

- If $m = n$, all the robot dof are controlled using the **visual task function** $\mathbf{e}(\mathbf{p}(t)) = \mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^*$

Control law

Since $\mathbf{e}(\mathbf{p}(t)) = \mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^*$, we have

$$\dot{\mathbf{e}} = \mathbf{L_s}\, \mathbf{v}_q + \frac{\partial \mathbf{e}}{\partial t} \quad \text{where} \begin{cases} \mathbf{v}_q = \mathbf{v}_c \text{ for eye-in-hand system} \\ \mathbf{v}_q = -\mathbf{v}_o \text{ for eye-to-hand system} \end{cases}$$

We obtain ideally for an exponential decrease of $\mathbf{e}$ $\quad (\dot{\mathbf{e}} = -\lambda\, \mathbf{e})$

$$\mathbf{v}_q = \mathbf{L_s}^+ \, (-\lambda\, \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t})$$

Since $\mathbf{L_s}$ and $\frac{\partial \mathbf{e}}{\partial t}$ are not perfectly known, one uses

$$\mathbf{v}_q = \widehat{\mathbf{L_s}}^+ \, (-\lambda\, \mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}})$$

Stability analysis

Behavior of the closed-loop system :

$$\dot{\mathbf{e}} = \mathbf{L_s}\,\mathbf{v}_q + \frac{\partial \mathbf{e}}{\partial t} = -\lambda\, \mathbf{L_s}\widehat{\mathbf{L_s}}^{+}\mathbf{e} \;-\; \mathbf{L_s}\widehat{\mathbf{L_s}}^{+}\widehat{\frac{\partial \mathbf{e}}{\partial t}} + \frac{\partial \mathbf{e}}{\partial t}$$

In case of tracking a mobile object, tracking errors as soon as:

$$\frac{\partial \mathbf{e}}{\partial t} \neq \mathbf{L_s}\widehat{\mathbf{L_s}}^{+}\widehat{\frac{\partial \mathbf{e}}{\partial t}}$$

If $k = m = n$ ($\widehat{\mathbf{L_s}}^{+} = \widehat{\mathbf{L_s}}^{-1}$), global asymptotic stability (GAS) if

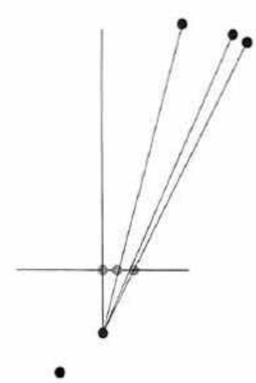
$$\mathbf{L_s}\widehat{\mathbf{L_s}}^{-1} > 0$$

If $k > m = n$, local asymptotic stability (LAS) if

$$\widehat{\mathbf{L_s}}^{+}\mathbf{L_s} > 0$$

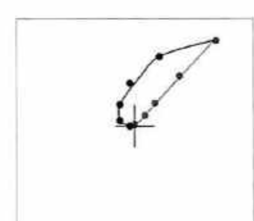
A simple case $k = m = n = 2$

Case of a pan-tilt camera observing a point :



$$\mathbf{s} = (x, y)\ ,\ \mathbf{s}^* = (0, 0)$$

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} xy & -(1+x^2) \\ 1+y^2 & -xy \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}$$



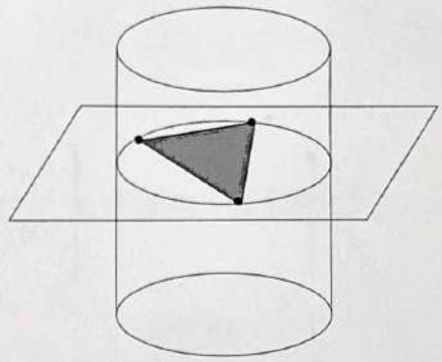
$$\mathbf{v}_C = -\lambda\, \widehat{\mathbf{L_s}}^{-1}\,(\mathbf{s} - \mathbf{s}^*)$$

$$\Leftrightarrow \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = -\frac{\lambda}{1 + x^2 + y^2} \begin{bmatrix} y \\ -x \end{bmatrix}$$

If no error occurs, $\dot{\mathbf{s}} = -\lambda\, \mathbf{s}$: trajectory = straight line in the image

Control when $k = m = n = 6$

- **Impossible with basic 2D visual features**



Using 3 points ($\mathbf{L_s}$ is 6×6)

- pose ambiguity (4 solutions)
- possible singularity of $\mathbf{L_s}$

Control when $k = m = n = 6$

- **Possible with 3D visual features** (assumed to be perfectly estimated)

$$\text{If } \mathbf{s} = \begin{bmatrix} {}^{c^*}\mathbf{t}_c \\ \theta\mathbf{u} \end{bmatrix}, \mathbf{v}_c = \begin{bmatrix} v \\ \omega \end{bmatrix} = -\lambda \begin{bmatrix} {}^{c*}\mathbf{R}_c & 0_3 \\ 0_3 & \mathbf{L}_\omega \end{bmatrix}^{-1} \begin{bmatrix} {}^{c^*}\mathbf{t}_c \\ \theta\mathbf{u} \end{bmatrix} = -\lambda \begin{bmatrix} {}^{c}\mathbf{R}_{c^*}\,{}^{c^*}\mathbf{t}_c \\ \theta\mathbf{u} \end{bmatrix}$$

Advantages

- $\mathbf{L_s}$ block-diagonal and never singular
- GAS in perfect conditions
- Translational and rotational motions decoupled
- Camera trajectory : straight line in 3D space

Drawbacks

- No control in the image (the target may get out of the image)

Control when $k = m = n = 6$

- **Possible with 3D visual features** (assumed to be perfectly estimated)

$$\text{If } \mathbf{s} = \begin{bmatrix} {}^{c}\mathbf{t}_o \\ \theta\mathbf{u} \end{bmatrix}, \mathbf{v}_c = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = -\lambda \begin{bmatrix} -\mathbf{I}_3 & [{}^{c}\mathbf{t}_o]_\times \\ \mathbf{0}_3 & \mathbf{L}_{\boldsymbol{\omega}} \end{bmatrix}^{-1} \begin{bmatrix} {}^{c}\mathbf{t}_o - {}^{c^*}\mathbf{t}_o \\ \theta\mathbf{u} \end{bmatrix}$$

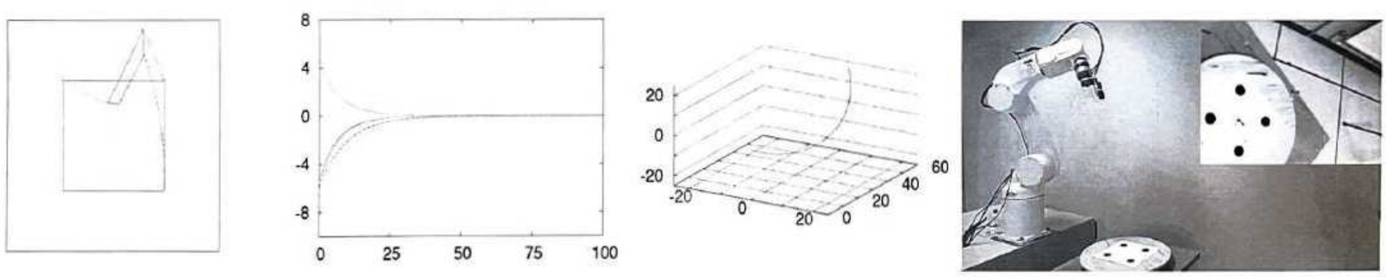
Advantages

- $\mathbf{L_s}$ block-triangular and never singular
- GAS in perfect conditions
- Rotational motions decoupled
- Straight line trajectory of O in the image

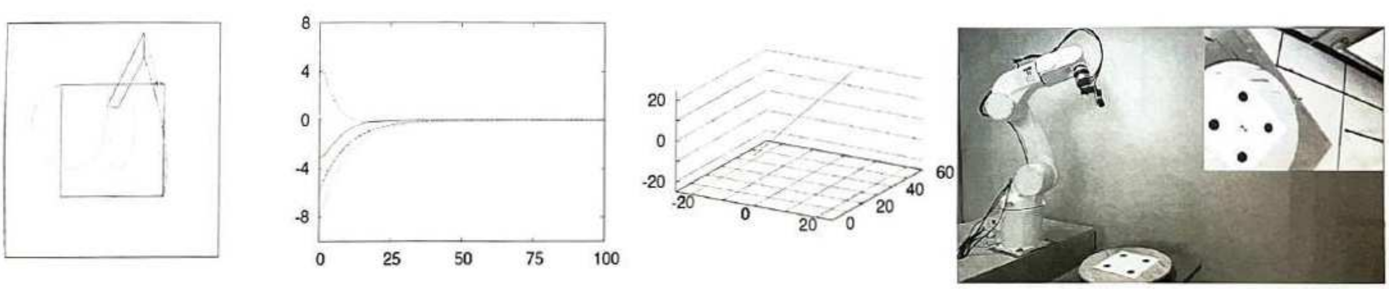
Drawbacks

- Camera trajectory : no more a straight line in 3D space

Results using ${}^{c}\mathbf{t}_o$



Results using ${}^{c^*}\mathbf{t}_c$



Control when $k = m = n = 6$: 2 1/2 D visual servoing

Idea : Combine 2D image data and 3D data

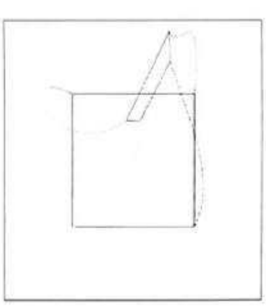
$$\mathbf{s} = \begin{bmatrix} {}^{c^*}\mathbf{t}_{cx} \\ {}^{c^*}\mathbf{t}_{cy} \\ {}^{c^*}\mathbf{t}_{cz} \\ x \\ y \\ \theta \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \text{translation to realize} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{image point coordinates} \\ \rightarrow \text{orientation} \end{array} \Rightarrow \mathbf{L_s} = \begin{bmatrix} {}^{c^*}\mathbf{R}_c & \mathbf{0}_3 \\ \frac{1}{Z}\mathbf{L}_{v\omega} & \mathbf{L}_\omega \end{bmatrix}$$

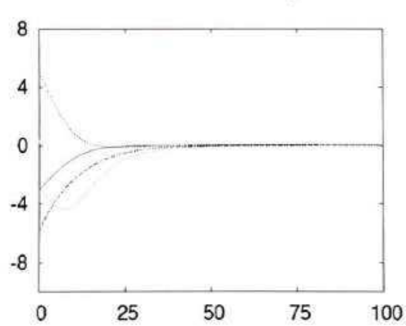
Advantages:

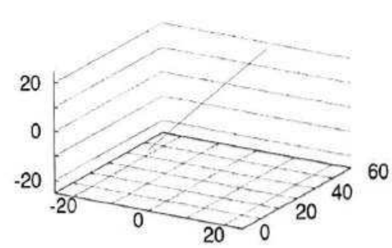
- Camera trajectory : straight line in 3D space
- Trajectory in the image of the selected point : straight line
- GAS in perfect conditions

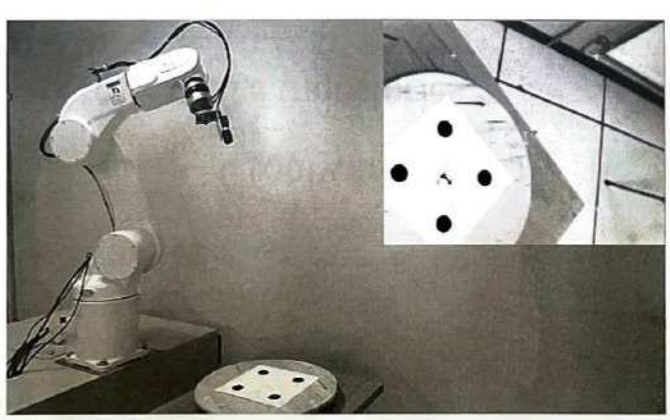
Drawback: $\mathbf{L_s}$ only block-triangular

Results using $\mathrm{s} = ({}^{c^*}\mathbf{t}_c, \mathbf{x}_g, \theta u_z)$









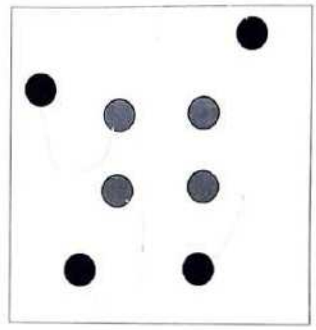
Control when $k > m, m = n = 6$ using 2D visual features

Usual choices for $\widehat{\mathbf{L_s}}$: $\widehat{\mathbf{L_s}}_{|\mathbf{s}=\mathbf{s}^*}$ or $\widehat{\mathbf{L_s}}_{|\mathbf{s}=\mathbf{s}(t)}$

- Control law $\mathbf{v}_q = -\lambda \, \widehat{\mathbf{L_s}}^{+}_{|\mathbf{s}=\mathbf{s}^*} \, (\mathbf{s} - \mathbf{s}^*)$

  stable if $\widehat{\mathbf{L_s}}^{+}_{|\mathbf{s}=\mathbf{s}^*} \, \mathbf{L_s} > 0$ (only around $\mathbf{s}^*$)

  No real control of the image trajectories



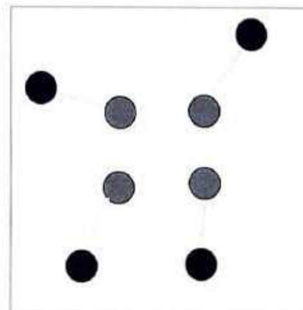
- Control law $\mathbf{v}_q = -\lambda \, \widehat{\mathbf{L_s}}^{+} \, (\mathbf{s} - \mathbf{s}^*)$

  tries to ensure $\dot{\mathbf{s}} = -\lambda \, (\mathbf{s} - \mathbf{s}^*)$

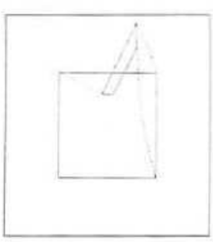
  Possible local minima (LAS)

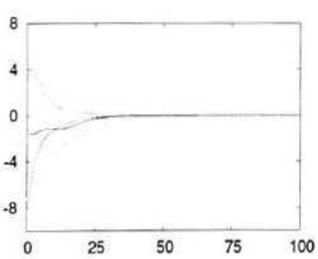
  due to too much constraints

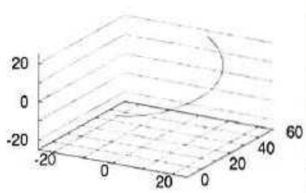
  Needs to estimate the depths

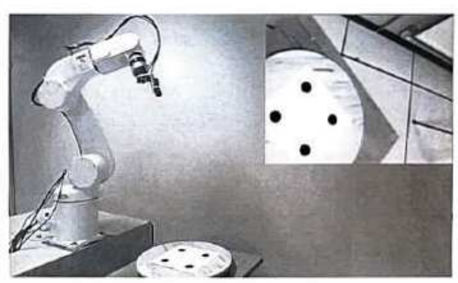


Results using $\mathbf{s} = (x_1, y_1, ..., x_4, y_4)$ and $\mathbf{L_s}$

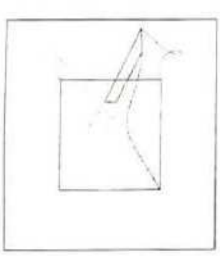


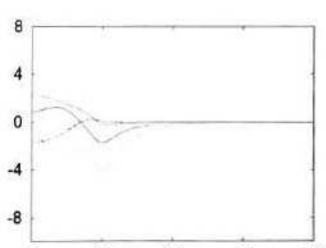


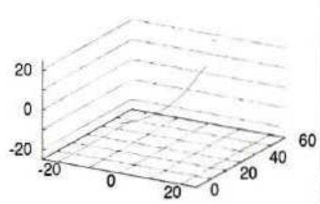


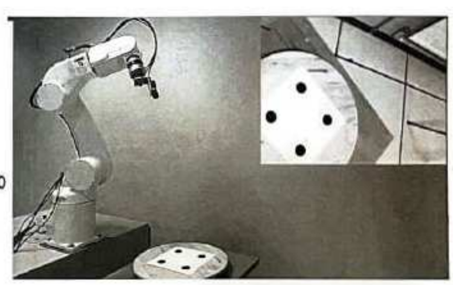


Results using $\mathbf{s} = (x_1, y_1, ..., x_4, y_4)$ and $\mathbf{L_s}_{|\mathbf{s}=\mathbf{s}^*}$









⇒ Control issues

▷ Control of visual tasks ($m = n$)

⇒ Target tracking

▷ Classification of the visual tasks

▷ Hybrid tasks ($m < n$)

Target tracking

PI Controller: Integral term classical in Automatic Control to reduce tracking errors

$\mathbf{I}_k$: estimation of $\frac{\partial \mathbf{e}}{\partial t}$ at iteration k

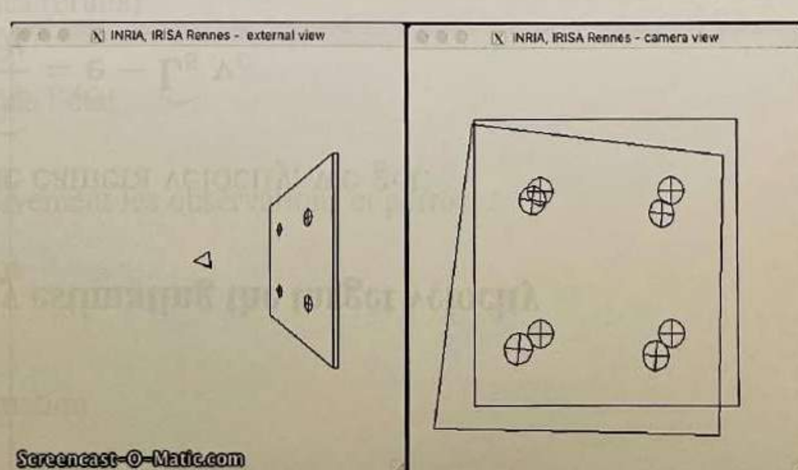
$$\mathbf{v}_q = \widehat{\mathbf{L}_\mathbf{s}}^+(-\lambda \mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}}) = \widehat{\mathbf{L}_\mathbf{s}}^+(-\lambda \mathbf{e}_k - \mu \sum_{j=0}^{k} \mathbf{e}_j)$$

$$\mathbf{I}_0 = 0$$

$$\mathbf{I}_{k+1} = I_k + \mu\, \mathbf{e}_k = \mu \sum_{j=0}^{k} \mathbf{e}_j$$

Efficient to track a target at constant velocity:

$$\mathbf{I}_{k+1} = \mathbf{I}_k \text{ if } \mathbf{e}_k = 0$$



Target tracking by estimating the target velocity

If it is possible to measure the camera velocity, we get:

$$\widehat{\frac{\partial \mathbf{e}}{\partial t}} = \widehat{\dot{\mathbf{e}}} - \widehat{\mathbf{L_s}}\ \mathbf{v}_c$$

with $\widehat{\dot{\mathbf{e}}}_k = \frac{\mathbf{e}_k - \mathbf{e}_{k-1}}{\Delta t}$ for instance.

A Kalman filter may then be used.

Le filtrage de Kalman

Filtrage de Kalman = ***estimation linéaire récursive optimale*** à partir :

- d'un modèle d'état (dynamique interne d'évolution du système)
- d'un modèle de mesure (observations).

intégrant des incertitudes potentielles (bruits)

Objectif : Estimer/suivre la valeur de l'état

Le filtre de Kalman intègre successivement les observations et permet :

- d'améliorer l'estimation de l'état
- d'assurer son suivi
- de calculer la variance de l'estimation

Filtre de Kalman

Equation des modèles :

$$\begin{aligned} S(k+1) &= A(k)S(k) + B(k)W(k) && \text{Évolution de l'état} \\ X(k) &= C(k)S(k) + D(k)N(k) && \text{Mesures (observations)} \end{aligned}$$

où :

- $S(k)$ est la valeur inconnue ***de l'état*** à l'itération k ;
- X représente les ***observations*** effectuées ;
- W est un bruit blanc de moyenne nulle et de variance V_W ;
- N est un bruit blanc de moyenne nulle et de variance V_N (décorrelé de W) ;
- les matrices A, B, C et D décrivent les formes des modèles retenus.

Le filtre de Kalman procède en deux étapes :

- phase de **prédiction** ;
- phase de **lissage** ou **filtrage**.

Exemples de modèles

- ***Modèle d'état à position constante :*** dans ce cas très simple, on a :

$$\begin{aligned} s(k+1) &= s(k) + w(k) \\ x(k) &= s(k) + n(k) \end{aligned}$$

- ***Modèle d'état à vitesse constante :*** l'état est alors de dimension 2 $S(k) = (s(k), \dot{s}(k))$ et l'équation du modèle est donnée par :

$$S(k+1) = AS(k) + BW(k)$$

$$\text{avec } A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \text{ et } B = I_2$$

Si l'on ne mesure que la position x de l'état, l'equation de mesure a la forme :

$$x(k) = CS(k) + n(k) \text{ avec } C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Phase de prédiction

$\widehat{S}(k-1/k-1)$: résultat de l'estimation à l'itération $k-1$

$\widehat{S}(k/k-1)$: valeur prédite de $S(k)$.

$$\boxed{\widehat{S}(k/k-1) = A(k-1)\widehat{S}(k-1/k-1)}$$

L'erreur de prédiction s'écrit

$$\widetilde{S}(k/k-1) = S(k) - \widehat{S}(k/k-1)$$

La matrice de covariance $\Gamma_{\widetilde{S}(k/k-1)}$ de l'erreur de prédiction est donnée par :

$$\boxed{\Gamma_{\widetilde{S}(k/k-1)} = A(k-1)\Gamma_{\widetilde{S}(k-1/k-1)}A^T(k-1) + B(k-1)V_W B^T(k-1)}$$

où $\Gamma_{\widetilde{S}(k-1/k-1)}$ est la matrice de covariance de l'erreur d'estimation.

Phase de filtrage

Prise en compte de la ***mesure*** $X(k)$ et
filtrage de la ***valeur prédite*** pour obtenir la valeur estimée $\widehat{S}(k/k)$:

$$\boxed{\widehat{S}(k/k) = \widehat{S}(k/k-1) + G(k)\left[X(k) - C(k)\widehat{S}(k/k-1)\right]}$$

où :

- $\left[X(k) - C(k)\widehat{S}(k/k-1)\right] = \left[X(k) - \widehat{X}(k/k-1)\right]$ = ***innovation***
- $G(k)$ est le ***gain du filtre***, calculé afin que la mesure soit indépendante de l'erreur de prédiction (tel que $\mathbf{E}\left[X(k)\widetilde{S}^T(k+1/k)\right] = 0$) :

$$\boxed{G(k) = \Gamma_{\widetilde{S}(k/k-1)}C^T\left[C\Gamma_{\widetilde{S}(k/k-1)}C^T + DV_N D^T\right]^{-1}}$$

Erreur d'estimation : $\widetilde{S}(k/k) = S(k) - \widehat{S}(k/k)$
Matrice de covariance de l'erreur d'estimation $\Gamma_{\widetilde{S}(k/k)}$ donnée par :

$$\boxed{\Gamma_{\widetilde{S}(k/k)} = (I - G(k)C(k))\,\Gamma_{\widetilde{S}(k/k-1)}}$$

En résumé

Équation d'état

$$S(k+1) = A(k)S(k) + B(k)W(k) \tag{1}$$

Équation de mesure

$$X(k) = C(k)S(k) + D(k)N(k) \tag{2}$$

Équations de prédiction

$$\widehat{S}(k/k-1) = A(k-1)\widehat{S}(k-1/k-1) \tag{3}$$

$$\Gamma_{\widetilde{S}(k/k-1)} = A(k-1)\Gamma_{\widetilde{S}(k-1/k-1)}A^T(k-1) + B(k-1)V_W B^T(k-1) \tag{4}$$

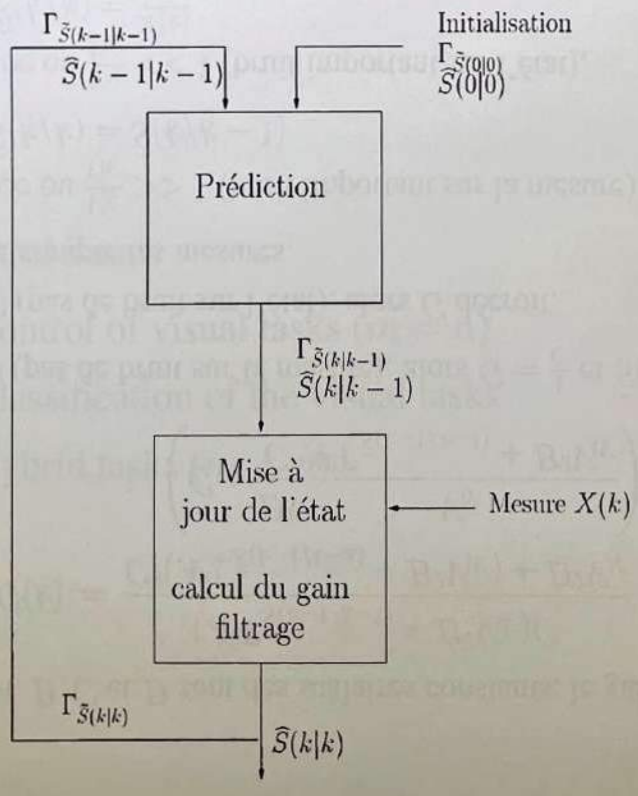
Équations de filtrage

$$\widehat{S}(k/k) = \widehat{S}(k/k-1) + G(k)\left[X(k) - C(k)\widehat{S}(k/k-1)\right] \tag{5}$$

$$G(k) = \Gamma_{\widetilde{S}(k/k-1)}C^T\left[C\Gamma_{\widetilde{S}(k/k-1)}C^T + DV_N D^T\right]^{-1} \tag{6}$$

$$\Gamma_{\widetilde{S}(k/k)} = (I - G(k)C(k))\,\Gamma_{\widetilde{S}(k/k-1)} \tag{7}$$

Shéma de principe



Interprétation

Si on suppose que A, B, C et D sont des scalaires constants, le gain G s'écrit :

$$\begin{aligned} G(k) &= \frac{(A^2\Gamma_{\widetilde{S}(k-1/k-1)} + B^2V_W)C}{C^2(A^2\Gamma_{\widetilde{S}(k-1/k-1)} + B^2V_W) + D^2V_N} \\ &= 1/\left(C + \frac{D^2}{C}\frac{V_N}{A^2\Gamma_{\widetilde{S}(k-1/k-1)} + B^2V_W}\right) \end{aligned}$$

1. Lorsque $V_N = 0$ (pas de bruit sur la mesure), alors $G = \frac{1}{C}$ et $\widehat{S}(k/k) = \frac{X(k)}{C}$
2. Lorsque $V_W = 0$ (pas de bruit sur l'état), alors G décroît.
   On prend peu en compte les mesures.
3. Lorsque $V_N \to \infty$ ou $\frac{V_N}{V_W} >> 1$ (bruit important sur la mesure),
   alors $G = 0$ et $\widehat{S}(k/k) = \widehat{S}(k/k-1)$
4. Lorsque $V_W \to \infty$ ou $\frac{V_N}{V_W} << 1$ (bruit important sur l'état),
   alors $G = \frac{1}{C}$ et $\widehat{S}(k/k) = \frac{X(k)}{C}$

$\Rightarrow$ Control issues

$\triangleright$ Control of visual tasks ($m = n$)

$\Rightarrow$ Classification of the visual tasks

$\triangleright$ Hybrid tasks ($m < n$)

Task classification: virtual link

- The task $\mathbf{s}(t) = \mathbf{s}^*$ defines a virtual link between the sensor and its environment.
- This link is characterized by the set $\mathcal{S}^*$ of 3D motions such that $\dot{\mathbf{s}} = 0$

$$\mathcal{S}^* = \text{Ker } \mathbf{L_s}$$

- dim $\mathcal{S}^*$ = class of the link

Name	Class	T	R	Geometric symbol
Rigid	0	0	0	
Prismatic	1	1	0	
Rotary	1	0	1	
Sliding pivot	2	1	1	
Plane-to-plane	3	2	1	
Bearing	3	0	3	
Linear rectlinear	4	2	2	
Linear annular	4	1	3	
Point	5	2	3	

Case of a point

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

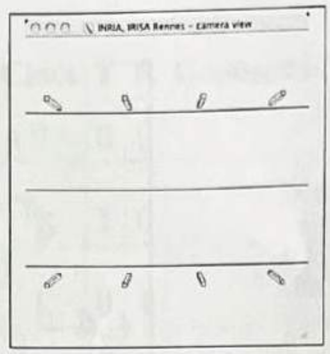
$$\Rightarrow \mathbf{L}_{xy} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}$$

$$\Rightarrow \mathcal{S}^* = \begin{bmatrix} x & 0 & Z(1+x^2+y^2) & 0 \\ y & 0 & 0 & Z(1+x^2+y^2) \\ 1 & 0 & 0 & 0 \\ 0 & x & -xy & 1+x^2 \\ 0 & y & -(1+y^2) & xy \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$\Rightarrow$ Link of class 4

Prismatic link

$$\mathcal{S}^* = (1, 0, 0, 0, 0, 0)$$



Using 3 (horizontal) straight lines

3D straight lines :

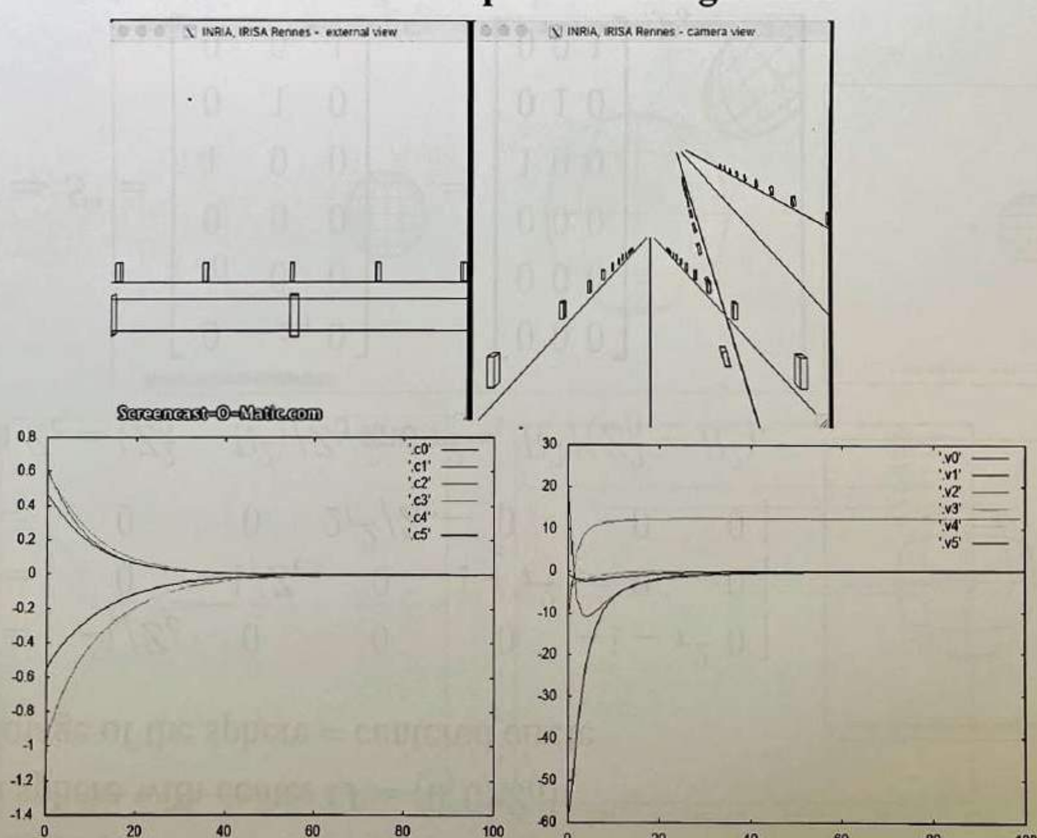
$$\mathbf{h}_i(\mathbf{X}, \mathbf{P}) = \begin{cases} Y - \frac{Y_i^*}{Z_i^*} Z = 0 \\ Z - Z_i^* = 0 \end{cases}, \; i = 1, 2, 3$$

2D straight lines : $\rho_i = Y_i^* / Z_i^*$, $\theta_i = \pi/2$

$$\Rightarrow \mathbf{L}_{\rho_i \theta_i} = \begin{bmatrix} 0 & -1/Z_i^* & \rho_i/Z_i^* & (1+\rho_i^2) & 0 & 0 \\ 0 & 0 & 0 & 0 & -\rho_i & -1 \end{bmatrix}$$

With a 3 dof mobile robot (v_x, v_z, ω_y), 1 straight line is sufficient.

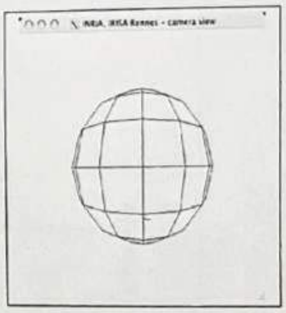
Prismatic link from 3 parallel straight lines



Bearing

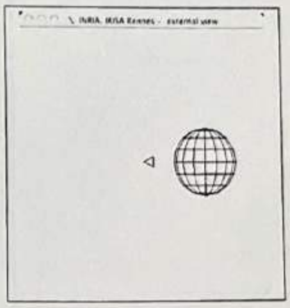
Using a sphere with center $\mathbf{O} = (0, 0, Z_0)$

$\Rightarrow$ Image of the sphere = centered circle



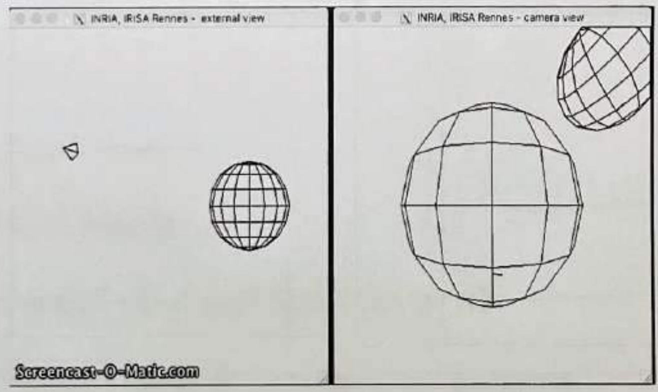
$$\begin{array}{llllllll}
\mathbf{L}_{x_c} & = [& -1/Z_c & 0 & 0 & 0 & -1-r^2 & 0\,] \\
\mathbf{L}_{y_c} & = [& 0 & -1/Z_c & 0 & 1+r^2 & 0 & 0\,] \\
\mathbf{L}_{\mu} & = [& 0 & 0 & 2r^2/Z_c & 0 & 0 & 0\,]
\end{array}$$

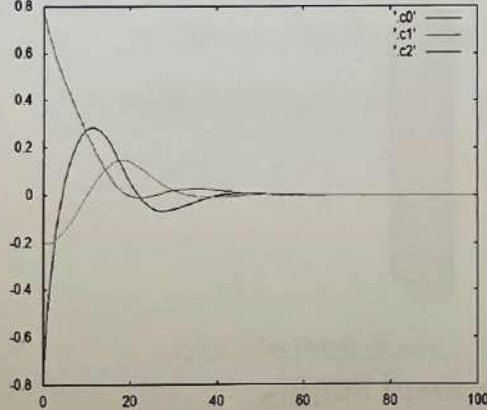
with $Z_c = (Z_0^2 - R^2)/Z_0$ and $r^2 = R^2/(Z_0^2 - R^2)$.

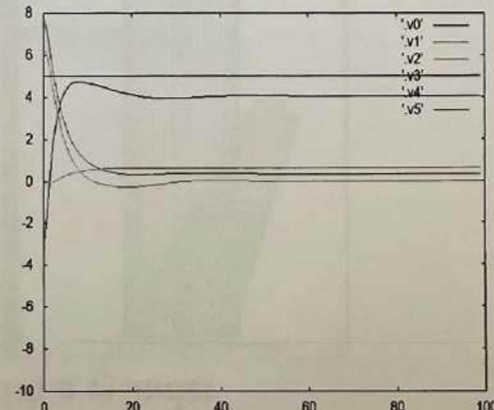


$$\Rightarrow S^* = \begin{bmatrix} 0 & -z_0 & 0 \\ z_0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{|\mathcal{F}_c} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{|\mathcal{F}_o}$$

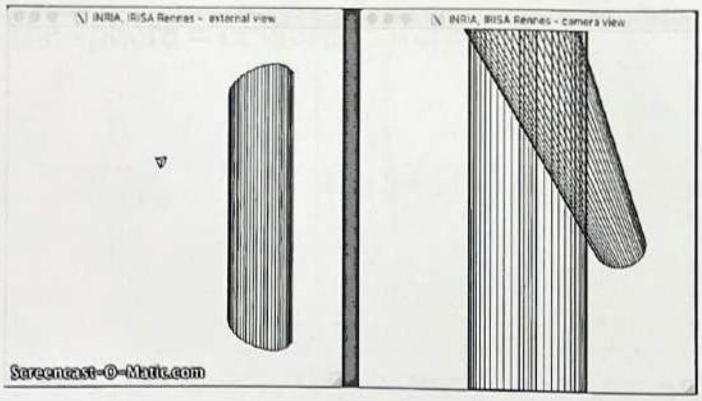
Bearing with a sphere

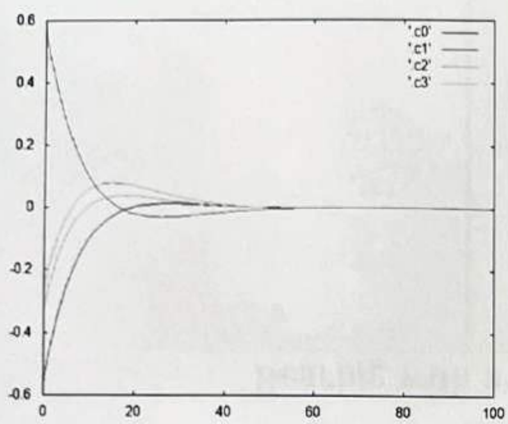


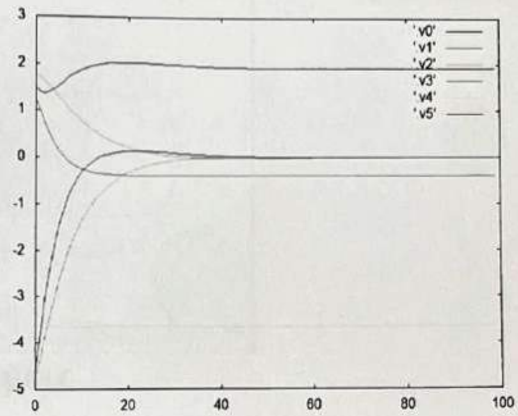




Sliding pivot with a cylinder







- ⇒ Control issues
  - ▷ Control of visual tasks ($m = n$)
  - ▷ Target tracking
  - ▷ Classification of the visual tasks
- ⇒ Hybrid tasks ($m < n$)

Visual task function

With k visual features s, one constraints m $(= n - N \leq k)$ robot dof

using the **visual task function** $\mathbf{e}_1(\mathbf{p}(t)) = \mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^*$

⇒ If $m < n$, it is possible to consider a supplementary task (trajectory following, joint limits avoidance, etc.).

Problem : How to combine both tasks ?

- $\mathbf{e}_1$: primary task
- $\mathbf{e}_2$: secondary task, considered at best under the constraint that $\mathbf{e}_1$ is satisfied.

Global task function

A task function e regulating $\mathbf{e}_2$ to 0 under the constraint $\mathbf{e}_1 = 0$ is given by :

$$\mathbf{e} = \widehat{\mathbf{L_s}}^+ \mathbf{e}_1 + (\mathbf{I}_n - \widehat{\mathbf{L_s}}^+ \widehat{\mathbf{L_s}})\, \mathbf{e}_2$$

where:

- $\mathbf{e}_1 = \mathbf{s} - \mathbf{s}^*$
- $(\mathbf{I}_n - \widehat{\mathbf{L_s}}^+ \widehat{\mathbf{L_s}})$: projection operator on the null space of $\widehat{\mathbf{L_s}}$

$$\Rightarrow \widehat{\mathbf{L_s}}(\mathbf{I}_n - \widehat{\mathbf{L_s}}^+ \widehat{\mathbf{L_s}})\, \mathbf{e}_2 = \mathbf{0}\ ,\ \forall \mathbf{e}_2$$

- if $m = n$, $(\mathbf{I}_n - \widehat{\mathbf{L_s}}^+ \widehat{\mathbf{L_s}}) = \mathbf{0}$

Control law

Since we have

$$\dot{\mathbf{e}} = \mathbf{L_e}\,\mathbf{v}_q + \frac{\partial \mathbf{e}}{\partial t} \quad \text{where} \begin{cases} \mathbf{v}_q = \mathbf{v}_c \text{ for eye-in-hand system} \\ \mathbf{v}_q = -\mathbf{v}_o \text{ for eye-to-hand system} \end{cases}$$

we obtain ideally for an exponential decrease of $\mathbf{e}$ $\quad (\dot{\mathbf{e}} = -\lambda\,\mathbf{e})$

$$\mathbf{v}_q = \mathbf{L_e}^{-1}\,(-\lambda\,\mathbf{e} - \frac{\partial \mathbf{e}}{\partial t})$$

with $\mathbf{L_e} = \widehat{\mathbf{L_s}}^{+}\mathbf{L_s} + (\mathbf{I}_n - \widehat{\mathbf{L_s}}^{+}\widehat{\mathbf{L_s}})\,\mathbf{L}_{\mathbf{e}_2}$ $\quad$ if $\widehat{\mathbf{L_s}}$ is constant

Since $\mathbf{L_e} \approx \mathbf{I}_n$ and $\frac{\partial \mathbf{e}_1}{\partial t}$ is not perfectly known, one uses

$$\mathbf{v}_q = -\lambda\,\mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}}$$

Stability analysis (same as before)

Example of secondary tasks

- **Trajectory following:**

  for instance, constant velocity v_z along z camera axis

$$\Rightarrow \mathbf{e}_2 = \begin{bmatrix} 0 \\ 0 \\ z(t) - z(0) - v_z t \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \frac{\partial \mathbf{e}_2}{\partial t} = \begin{bmatrix} 0 \\ 0 \\ -v_z \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Control law: $\mathbf{v}_q = -\lambda\,\mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}} = -\lambda\,\mathbf{e} - (\mathbf{I}_n - \widehat{\mathbf{L_s}}^{+}\widehat{\mathbf{L_s}})\frac{\partial \mathbf{e}_2}{\partial t}$

1) Modeling

2) Control

⇒ 3) Applications and other issues

Other issues

- Considering the epipolar geometry (depth up to a scalar factor)
- Coupling visual servoing and path planning ($\mathbf{s}^*(t)$)
- Considering other visual sensors (RGB-D, ultrasound probes)
- Considering dynamic constraints (non holonomic, under actuated systems)
- Task sequencing
- Combining vision + force

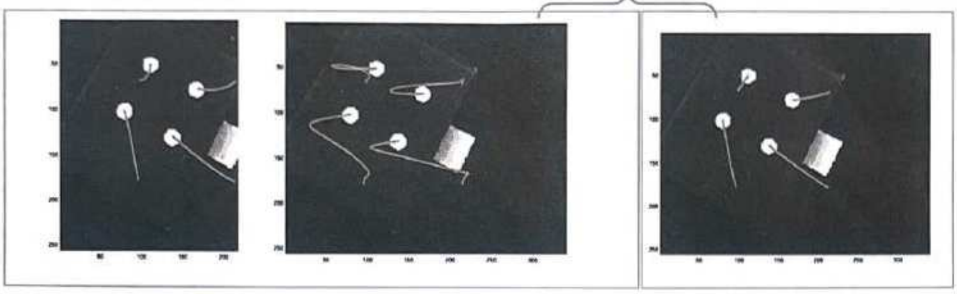
Coupling path planning and visual servoing (1)

- Dealing with large displacements
- Improving robustness wrt. calibration errors

s* no more constant: s*(t)

without error

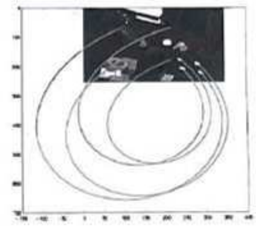
with errors



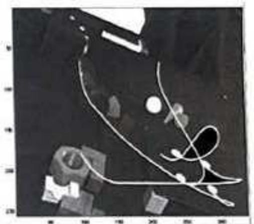
without planning

with planning

- Introduction of constrainsts
  - in the workspace
  - In the image
  - in the articular space
    (potential functions)

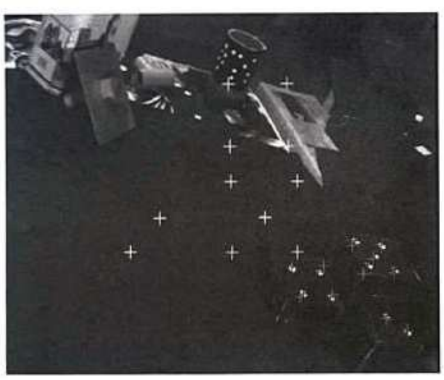


without constrainsts



with constrainsts

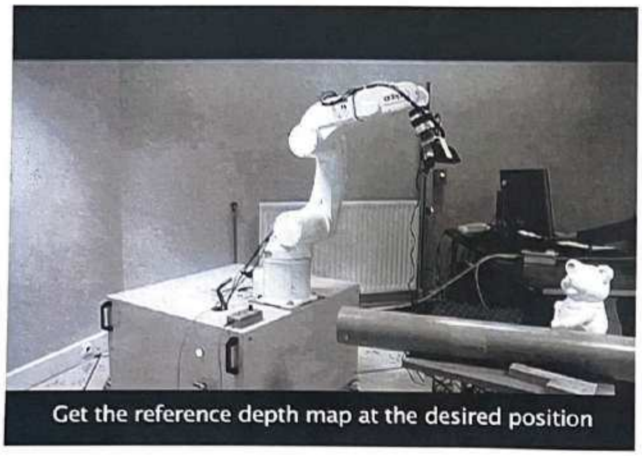
Coupling path planning and visual servoing (2)

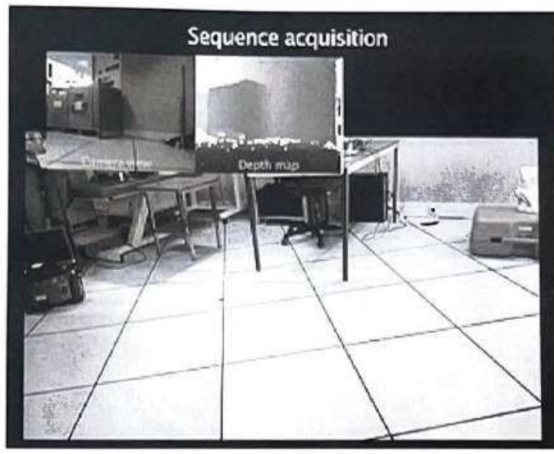


Considering other visual sensors: RGB-D

$\mathbf{s} = \rho_{\mathbf{Z}} \mathbf{Z}(\mathbf{x}(t)$

$\rho_{\mathbf{Z}}$: weights for dealing with outliers



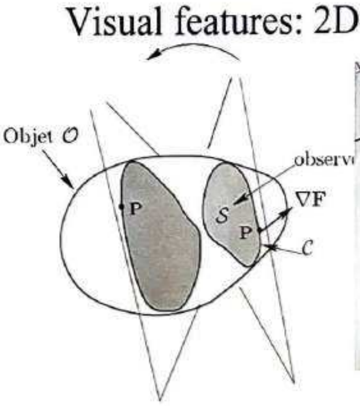


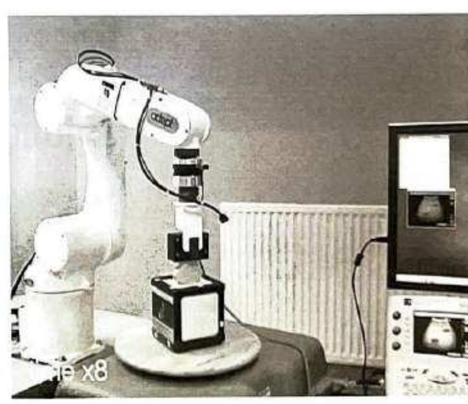
Considering other visual sensors: ultra-sound probes

Modeling revisited:

- Complete observation available in the probe plane
- No observation at all outside the probe plane

Visual features: 2D moments





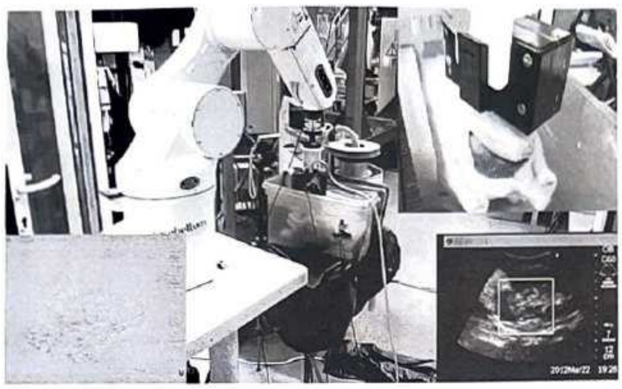
Potential clinical applications in medical robotics:

- image stabilization for diagnosis
- biopsies and therapy procedures: needle insertion

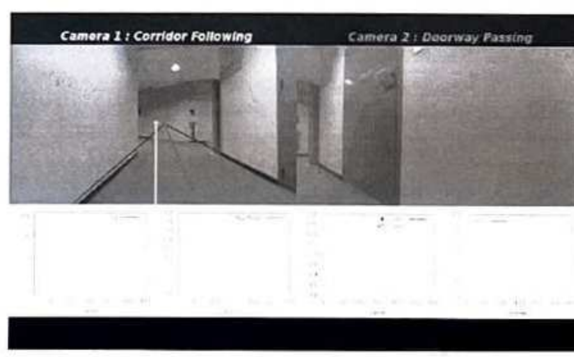
Considering other visual sensors: ultra-sound probes

Visual features: $\mathbf{s} = \mathbf{I}(\mathbf{x}(t))$

Motion compensation: R-GPC (repetitive predictive controller)



Corridor following by a wheelchair



Car following

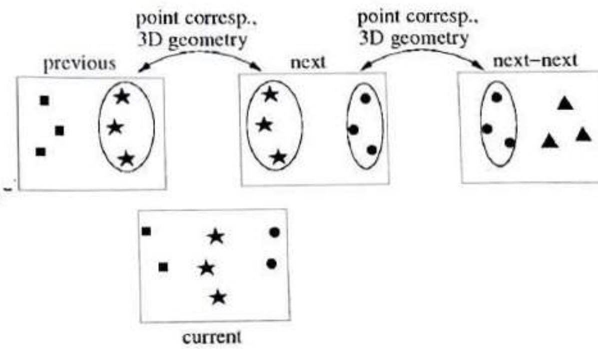
Autonomous navigation

Classical approach:

- teaching: global 3D reconstruction and accurate 3D localization (SLAM)
- following a specified 3D trajectory through accurate 3D localization

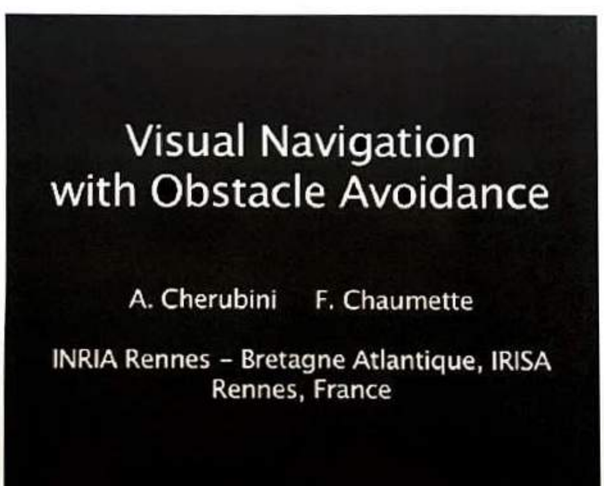
Approach developed: Accurate localization and mapping not mandatory

- teaching: topological description of the environment with key frames
- only local 3D reconstruction (points tracking and points transfer)
- navigation expressed as visual features to be seen (and not successive poses to be reached)
- simple IBVS for navigation

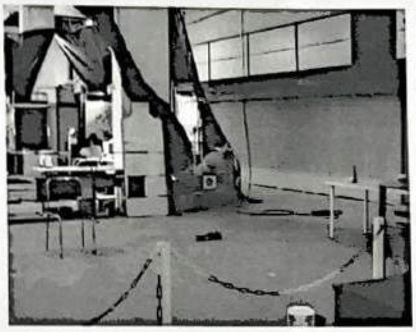


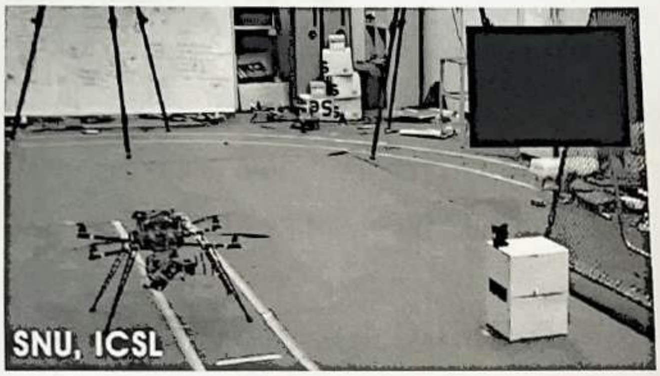
Autonomous navigation with obstacles avoidance

- Using a laser-range finder for obstacles localization
- Using a pan-tilt camera to observe the visual path while avoiding obstacles
- Redundancy framework to combine visual navigation and obstacles avoidance

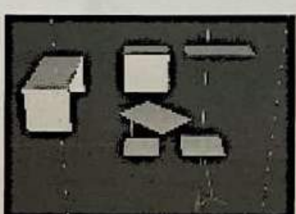


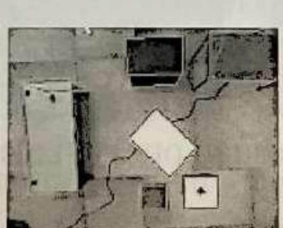
IBVS for X4-flyer

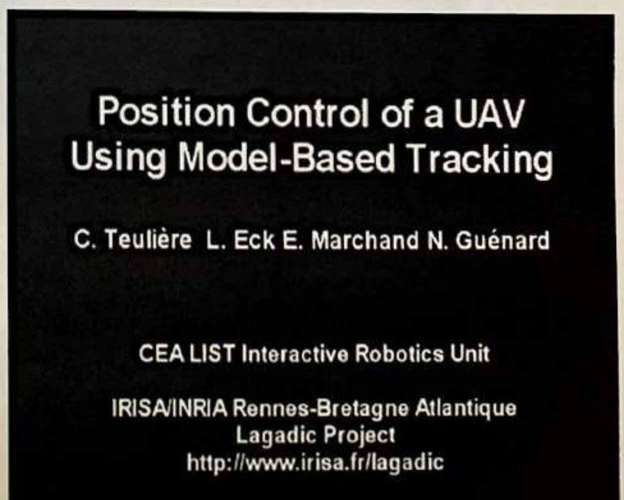




PBVS for X4-flyer

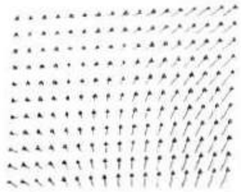
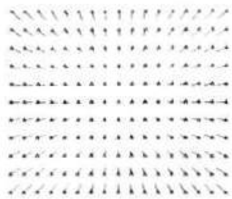






VTOL aircraft landing and terrain following

Using optical flow in visual servoing:

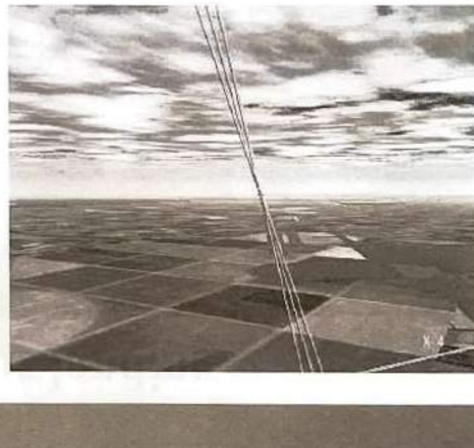
From  to 

Optical flow used to estimate the ratio translational velocity/depth

Visual servoing for autonomous landing

Automatic landing from the image of a runway (border and middle lines)

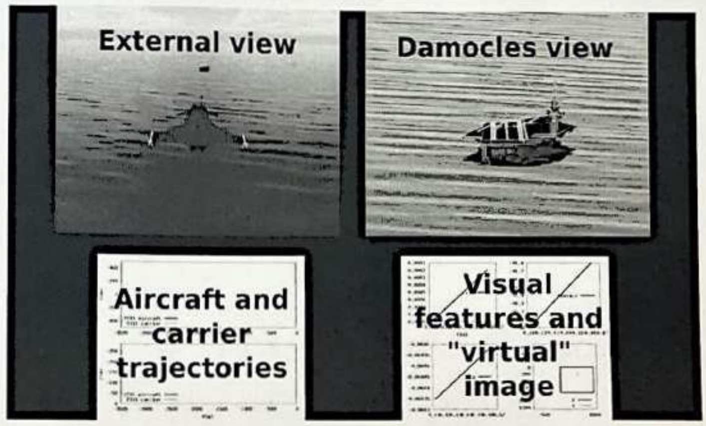
- planning image trajectories to be followed taking the aircraft dynamics into account (under actuated system)
- decoupled lateral and longitudinal control law
- adequate visual features for control
  - vanishing point and lines orientation for lateral control
  - vanishing point, slope, aircraft velocity for longitudinal control (LQR)



Visual servoing for autonomous landing

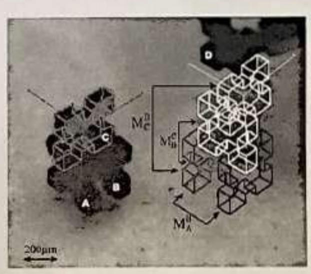
Automatic landing on an aircraft carrier

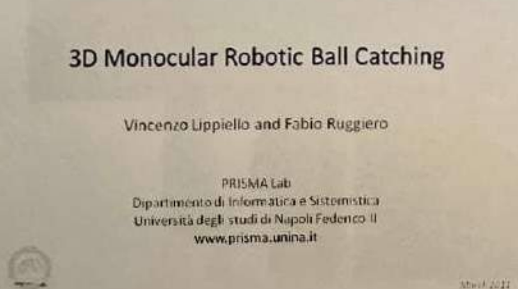
- 3D localization using 3D model-based tracking
- Landing using adequate visual features
- Compensating for the carrier motion

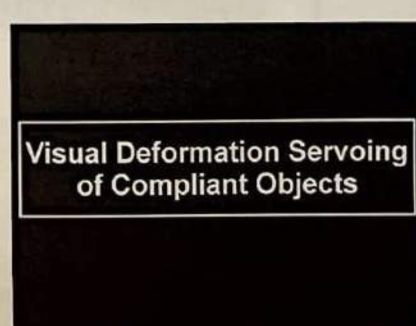


Applications

- Vision-based manipulation
- Vision-based navigation on any kind of robots
- Parallel robots
- Humanoid robots
- Etc.

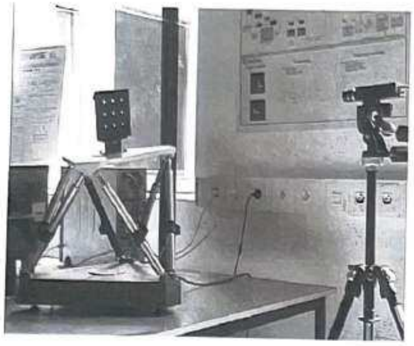


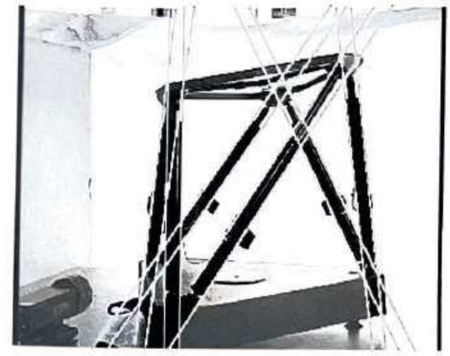


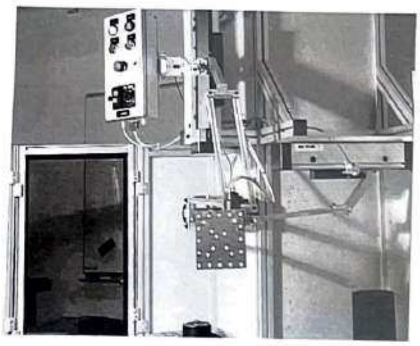


Parallel robots

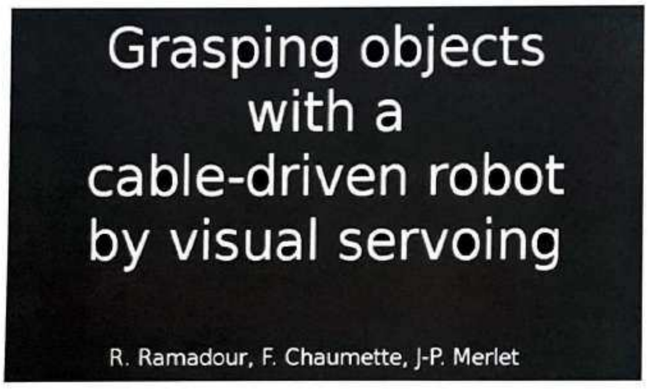
LASMEA







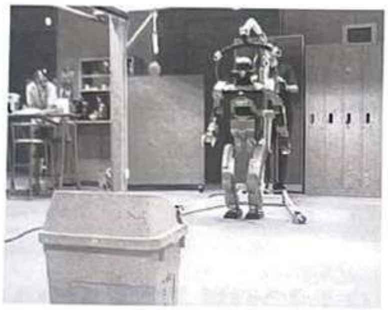
Cable-driven parallel robots



Visual servoing for a humanoid robot

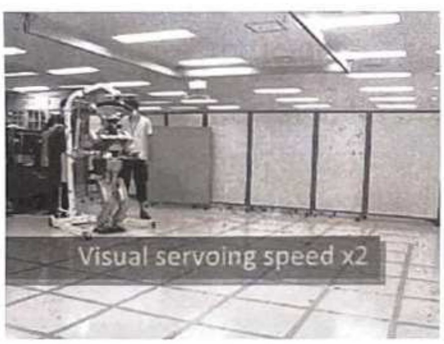
Catching a ball while walking by task sequencing and redundancy
- walking planned for security
- gaze control
- equilibrium control
- joint limits avoidance

All tasks managed by a stack
- Remove the good task for ensuring the constraints
- Put the task back as soon as possible



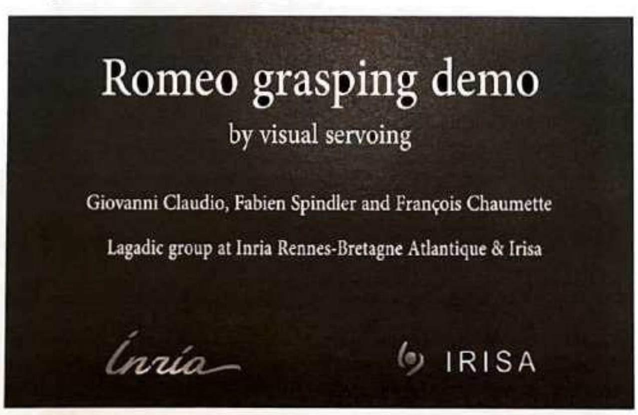

AIST
JRL-JAPAN

Visual locomotion of a humanoid robot



AIST
JRL-JAPAN

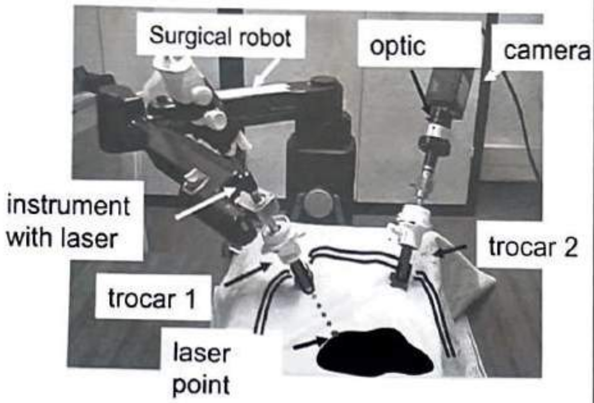
Vision-based manipulation / gaze control

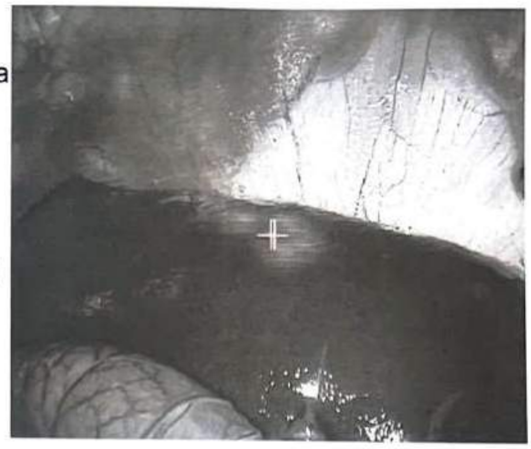


Medical robotics

Visual servoing for laparoscopic surgery

Goals: Development of semi-autonomous control modes using visual servoing to help the surgical gesture.





Autonomous tasks:

- Automatic retrieving of surgical instruments which are out of the field of view.
- Automatic 3-D positioning of surgical instruments.

Medical robotics

Compensation of complex physiological motions

- Motions of organs induced by heart beating or respiration.

Compensation of respiratory motions :

- Organs tracking (liver)
- Use of a repetitive predictive control scheme which learns the periodic cycle of the perturbation motions.

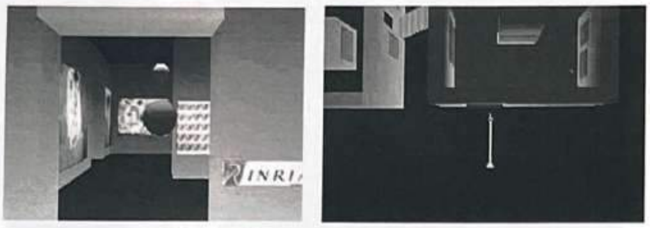
Beating Heart Tracking :

- Combination of cardiac and respiratory motions.
- Use of an adaptive filtering to estimate separate contributions of breathing and heartbeats.
- Predictive control scheme which anticipates the perturbation due to an adaptive disturbance predictor.

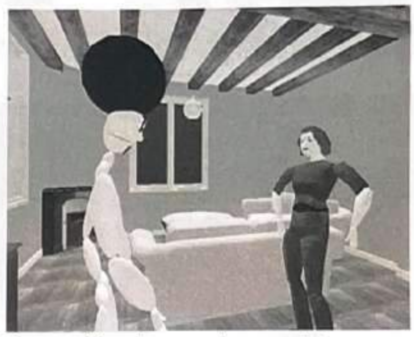
Visual servoing in computer animation

Control of realistic camera motion in virtual environments

- Task specification in the image
- Obstacles avoidance from redundancy



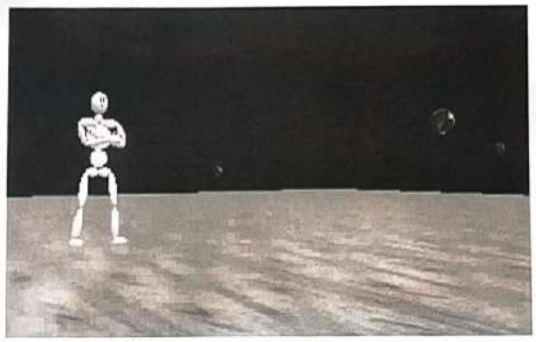
Follow the guide



Dialog with cinematographic constraints

Control of synthetic humanoids:

- Gaze and locomotion control



Applications: video games, films,...