

Asservissement visuel / Visual servoing

**PROGRAMME
UNIT-GDR ROBOTIQUE**



François Chaumette, INRIA Rennes-Bretagne Atlantique
Novembre 2010

References

- [1] F. Chaumette: Visual servoing. In *Robot Manipulators: Modeling, Performance Analysis and Control*, E. Dombre, W. Khalil (eds.), Chap. 6, pp. 279-336, ISTE, 2007
(french version in *Traité IC2*, Hermès, 2002).
- [2] F. Chaumette, S. Hutchinson: Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82-90, December 2006.
http://www.irisa.fr/lagadic/pdf/2006_ieee_ram_chaumette.pdf
- [3] F. Chaumette, S. Hutchinson: Visual servo control, Part II: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109-118, March 2007.
http://www.irisa.fr/lagadic/pdf/2007_ieee_ram_chaumette.pdf
- [4] S. Hutchinson, G. Hager, P. Corke: A tutorial on visual servo control, *IEEE Trans. on Robotics and Automation*, 12(5):651-670, October 1996.



Requirements

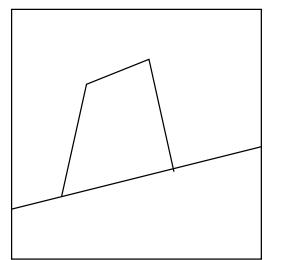
- Basics in linear algebra (Moore-Penrose pseudo-inverse)
- Basics in automatic control (PID controller)
- Basics in computer vision (pose estimation, epipolar geometry)

Overview

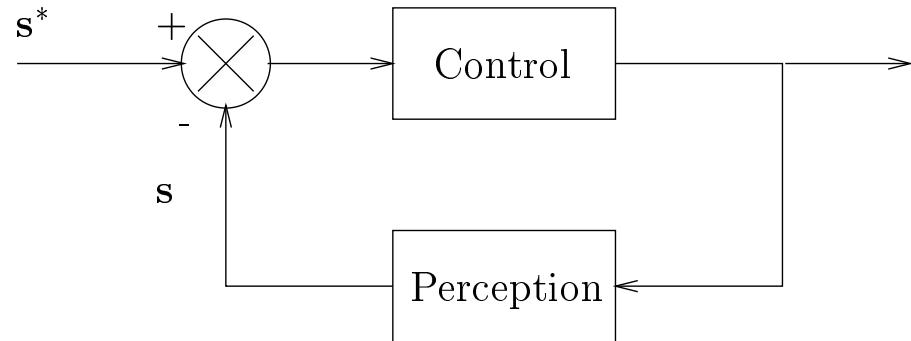
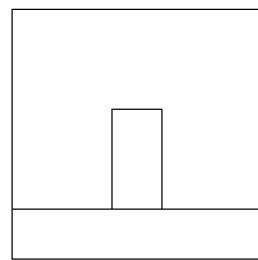
- Introduction
- Modeling visual features (p. 13)
- Control (p. 59)
- Applications (p. 95)



Aim of visual servoing: to realize robotics tasks (i.e. to control robot motion) using visual data embedded in a closed-loop system.



?



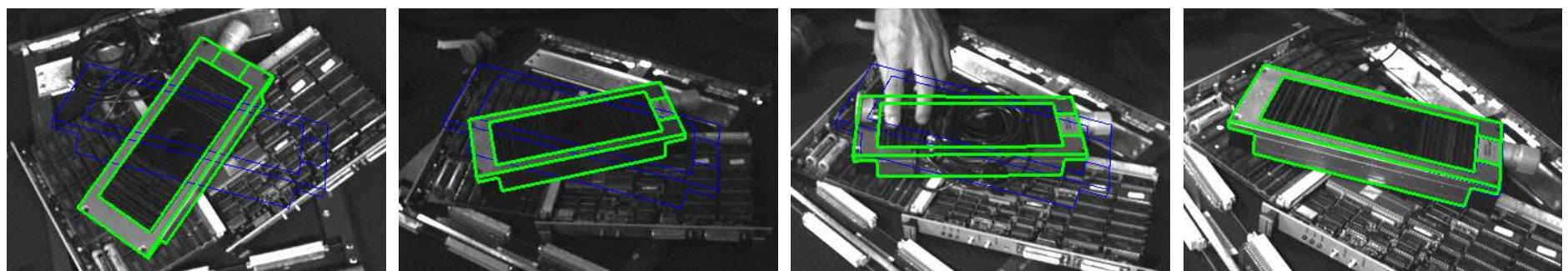
Necessary steps:

- Select k adequate visual features s to control m dof ($k \geq m, m \leq 6$)
- Determine the goal s^*
- Regulate the error $(s - s^*)$ to 0
- Image processing (matching and tracking near video rate)

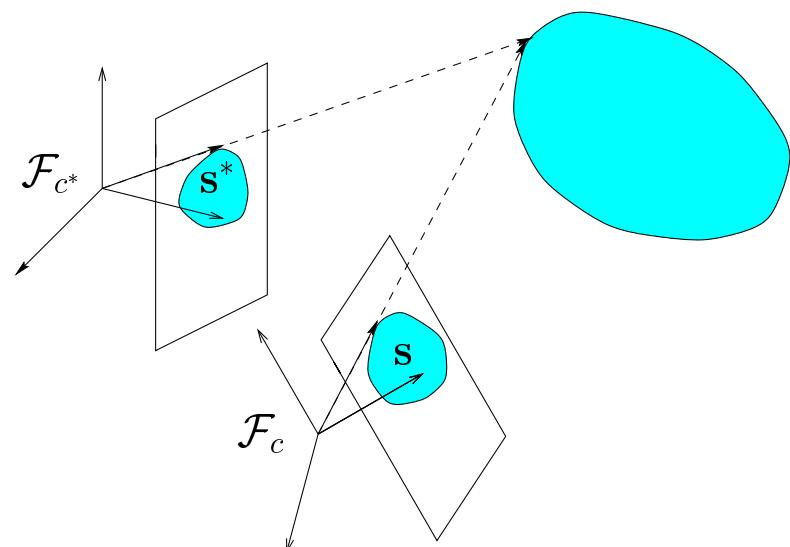
Pedestrian tracking using a pan/tilt camera



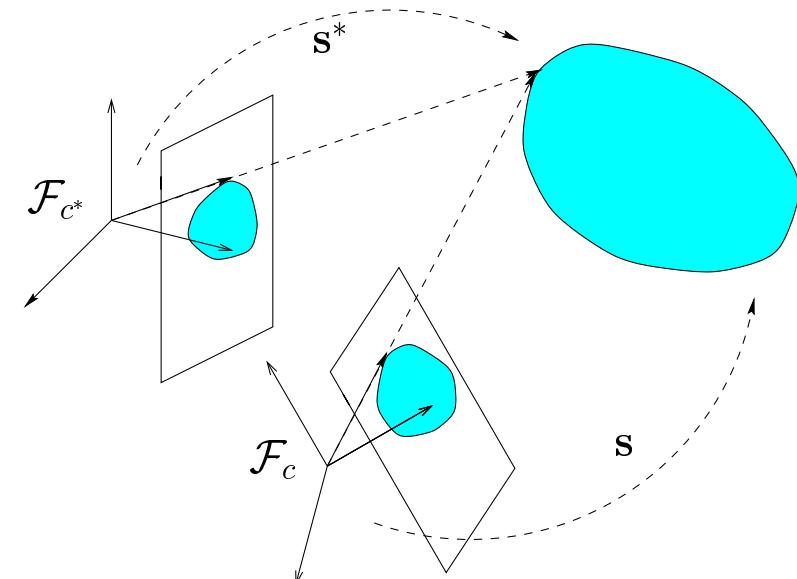
Positioning/grasping task using a 6 dof robot arm



Positioning task



2D visual features



3D visual features

Visual features: $\mathbf{s} = \mathbf{s}(\mathbf{p}(t)) \Rightarrow \dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}$ where:

- \mathbf{L}_s = interaction matrix (similar to a jacobian matrix)
- $\mathbf{v} = (\mathbf{v}, \boldsymbol{\omega})$ = instantaneous velocity (or kinematic screw)
with 3 translational and 3 rotational components

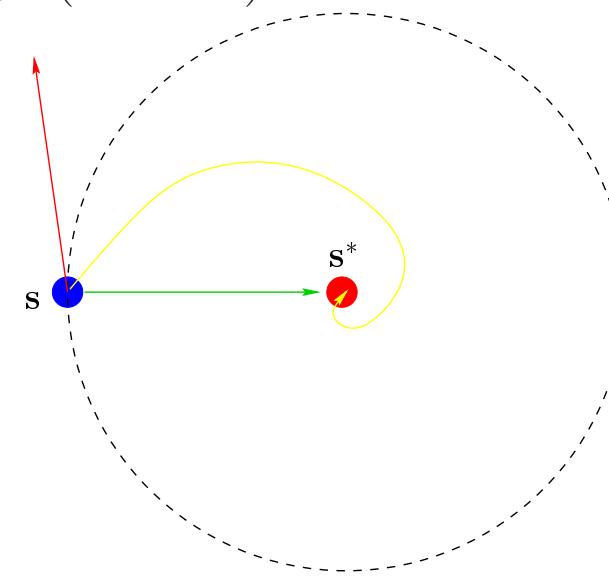
Principle of the control law

If we want $\dot{s} = -\lambda(s - s^*)$ (exponential decoupled decrease):

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

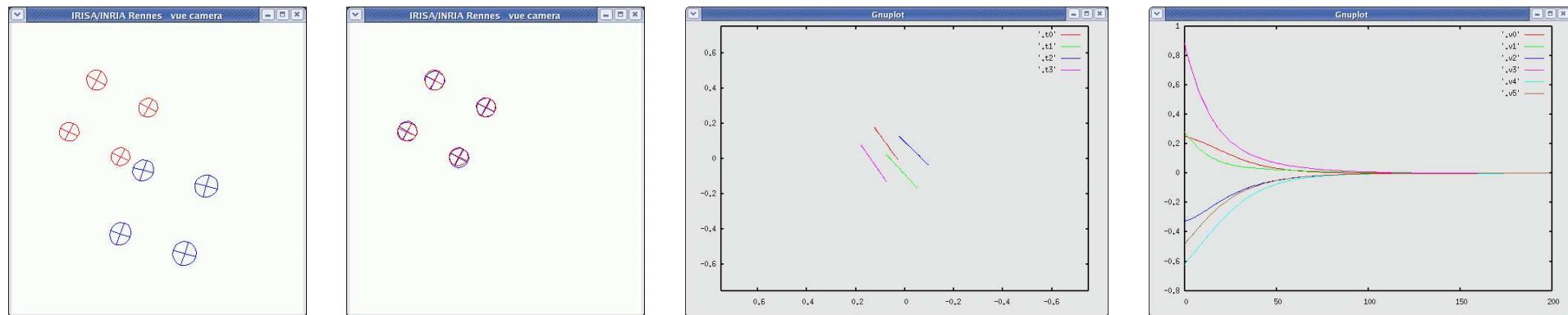
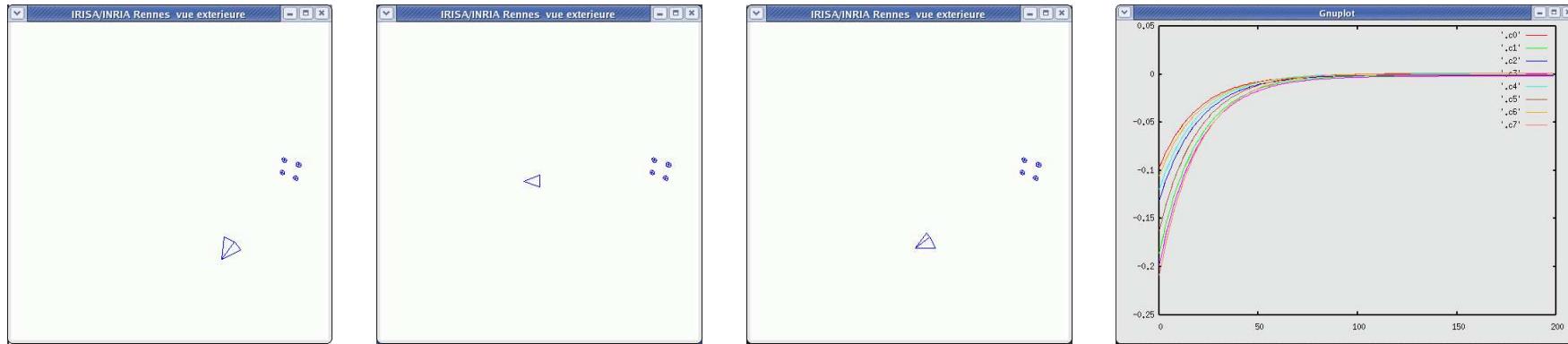
Closed-loop system: $\dot{s} = L_s v = -\lambda L_s \widehat{L}_s^+(s - s^*)$

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

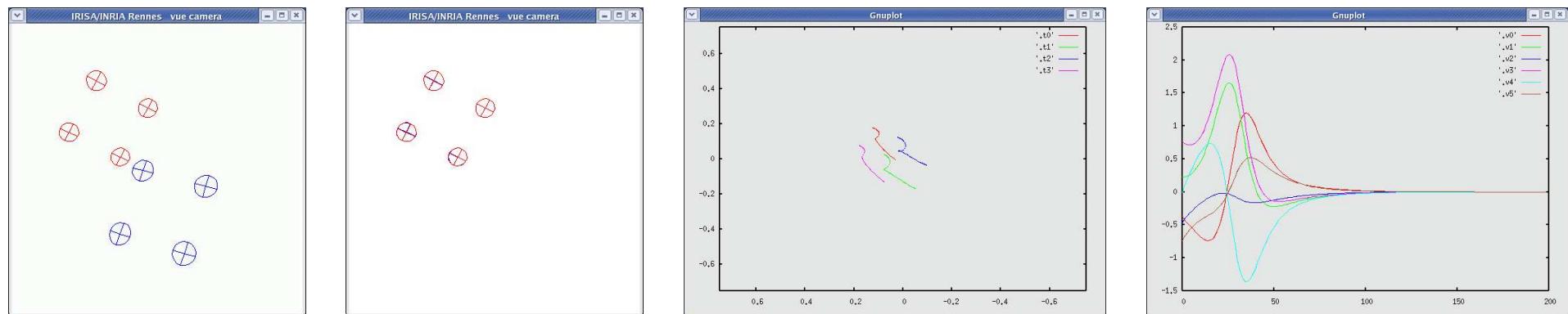
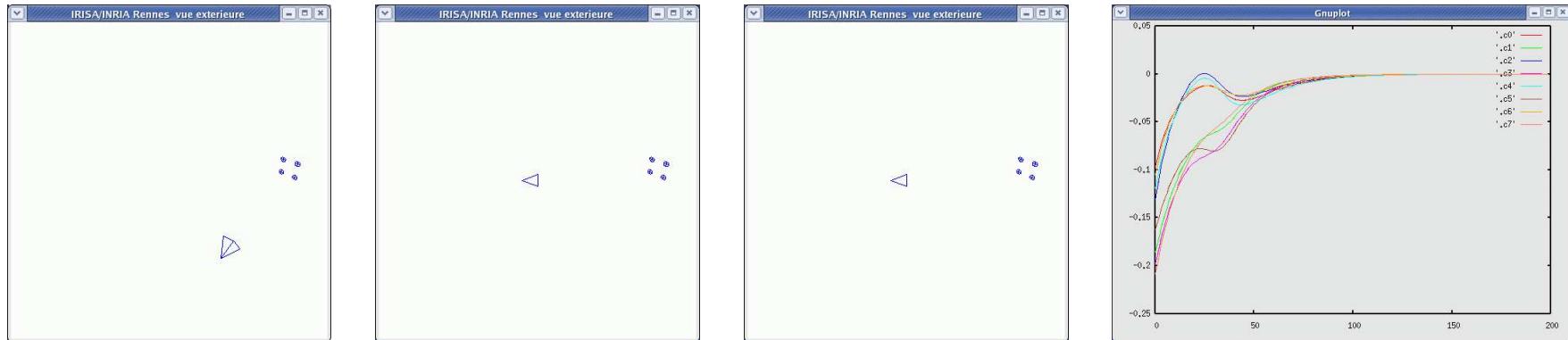


2D and 3D behavior directly linked by the choice of s
(through L_s and \widehat{L}_s)

Example 1: reaching a local minimum using \widehat{L}_s^+

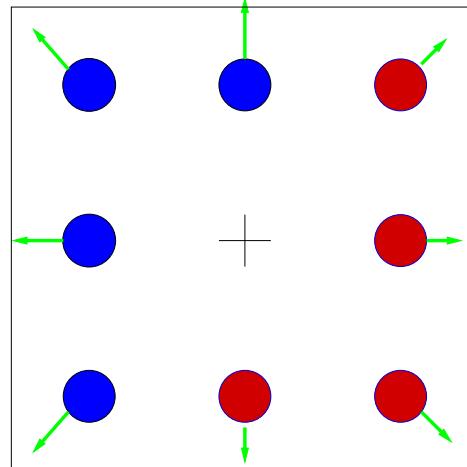
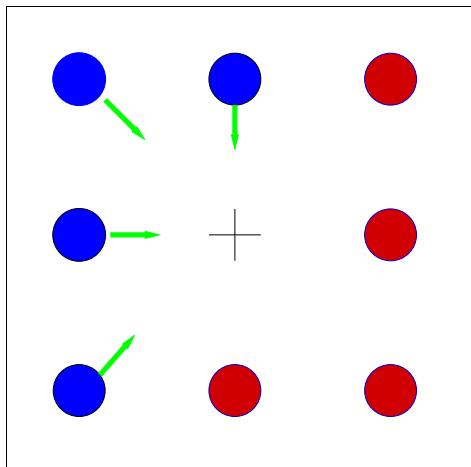


Example 2: reaching the global minimum using $\widehat{L}_s^+|_{s=s^*}$

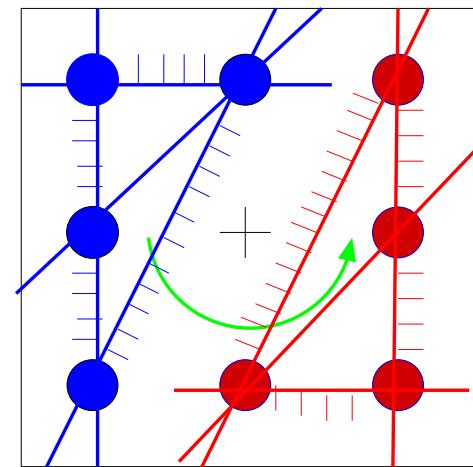


Example 3: reaching a singularity of L_s

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



Other choice



At singularity, rank $L_s = 2$.

Perfect behavior if s is composed of 2D straight lines parameters
(or composed of exact pose parameters)

Main goal: select adequate s for a given task

At least : select s such that $\text{rank } L_s = m$ around s^*
and such that $\text{Ker } L_s = \mathcal{S}^*$ (virtual linkage:
plane-to-plane, bearing, ball joint, etc.)

At most (yet a dream for 6 dof!) : select s such that $L_s = I_6$

- one feature for each robot dof
- perfect decoupling, same behavior of s and v
- none singularity nor local minima (global stability)
- ideal condition number
- control of a linear system

⇒ 1) Modeling issues

- ▷ Basics
- ▷ 2D visual features (p. 19)
- ▷ 3D visual features (p. 48)
- ▷ Omni-directional vision sensor, vision + structured light (p. 54)

2) Control issues (p. 59)

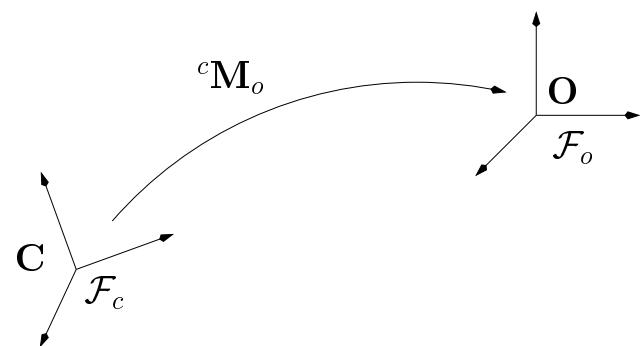
- ▷ Control of visual tasks (p. 59)
- ▷ Classification of the visual tasks (p. 79)
- ▷ Hybrid tasks (p. 86)

3) Applications (p. 95)



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 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 symmetric 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)

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

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

$$\mathbf{v} : \text{translational velocity at } \mathbf{C} : \quad \mathbf{v}(\mathbf{O}) = \mathbf{v}(\mathbf{C}) + [\boldsymbol{\omega}]_{\times} \mathbf{CO}$$

To express \mathbf{v} at \mathbf{O} in \mathcal{F}_o : ${}^o\mathbf{v} = {}^o\mathbf{V}_c \mathbf{v}$ 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 \mathbf{v} as $\mathbf{v} = \mathbf{v}_c - \mathbf{v}_o$

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

\mathbf{v}_o : object kinematic screw, expressed at \mathbf{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 :

$$s = s(p(t))$$

where $p(t)$ is the pose between the camera and the scene.

We get

$$\dot{s} = \frac{\partial s}{\partial p} \dot{p} = L_s v$$

where L_s is the **interaction matrix** related to s

(Jacobian $\frac{\partial s}{\partial p} \approx L_s$ since $v = M_p \dot{p}$ with $M_p \approx I_6$)

Using v_c and v_o , we obtain :

$$\dot{s} = L_s (v_c - v_o)$$

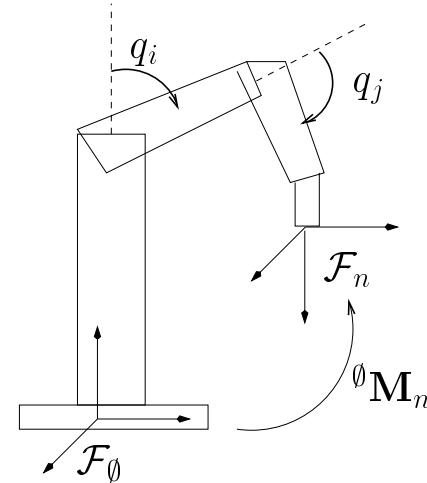


Robot Jacobian

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 ($\mathbf{p} \in SE_3$)



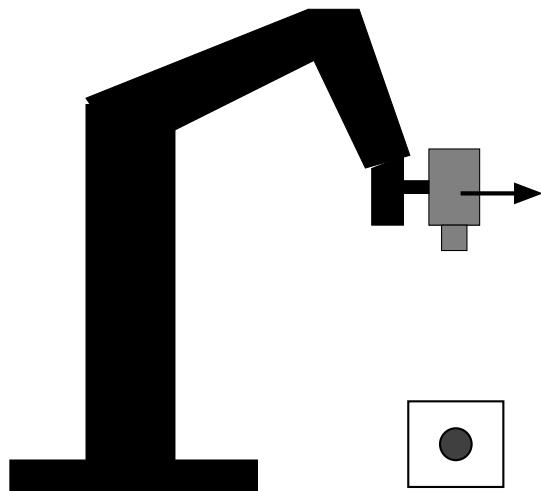
End-effector kinematic screw given by :

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

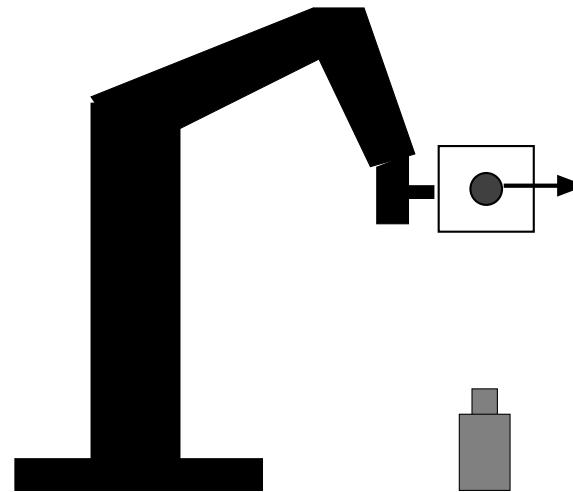
For velocity control, one computes $\dot{\mathbf{q}}^* = {}^n \mathbf{J}_n(\mathbf{q}^*)^{-1} \mathbf{v}_n^*$

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

Eye-in-Hand system



Eye-to-Hand system



$$\dot{s} = L_s^c V_n^n J_n(q) \dot{q} + \frac{\partial s}{\partial t}$$

$$\begin{aligned}\dot{s} &= -L_s^c V_n^n J_n(q) \dot{q} + \frac{\partial s}{\partial t} \\ &= -L_s^c V_\emptyset^\emptyset V_n^n J_n(q) \dot{q} + \frac{\partial s}{\partial t}\end{aligned}$$

- Modeling issues
 - ▷ Basics
 - ⇒ 2D visual features
 - ▷ 3D visual features
 - ▷ Omni-directional vision sensor, vision + structured light

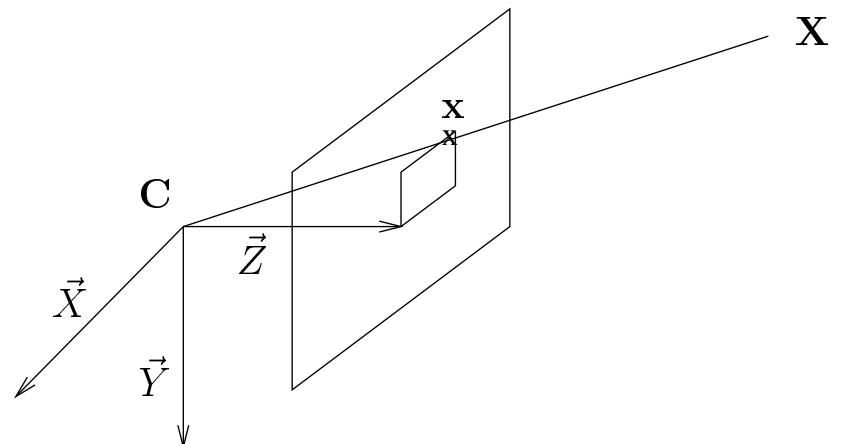


2D visual features: image point coordinates

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

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

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



Using a mobile camera and a fixed point:

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

We obtain:

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

2D visual features: image point coordinates

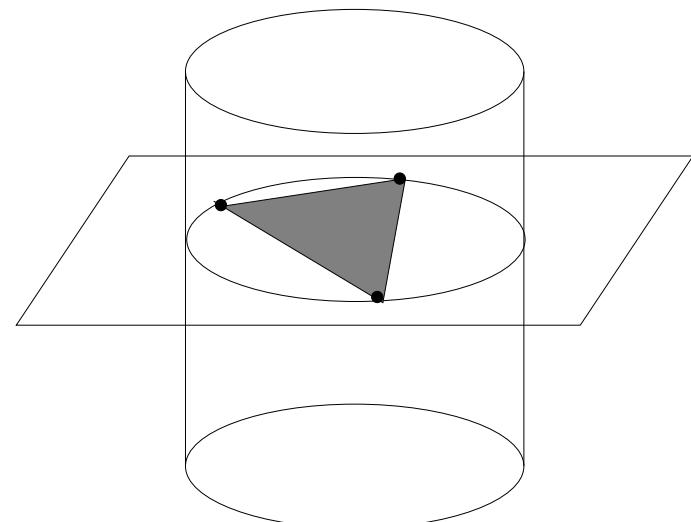
When $x = y = 0$ (principal point):

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

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 ($\Rightarrow n \geq 4$)

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

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

where $\mathbf{x}_c = (x_c, y_c)$ = principal point

and f_x, f_y = ratio between focal length f and pixel size.

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

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

Useful for stability analysis wrt. calibration errors

Image point expressed in pixels

when focal length is an available supplementary dof

$$x_p = x_c + \frac{f}{l_x} x \quad , \quad y_p = y_c + \frac{f}{l_y} y$$

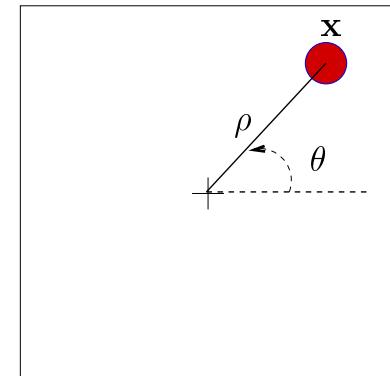
$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = \mathbf{L}_{\mathbf{x}_p} \mathbf{v} + \begin{bmatrix} x/l_x \\ y/l_y \end{bmatrix} \quad \dot{f} = \mathbf{L}_{\mathbf{x}_p} \mathbf{v} + \begin{bmatrix} (x_p - x_c)/f \\ (y_p - y_c)/f \end{bmatrix} \quad \dot{f}$$
$$= \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y & x/f \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x & y/f \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \dot{f} \end{bmatrix}$$

Useful redundancy wrt. motion along the optical axis v_z

Image point in cylindrical coordinates [Iwatsuki 02]

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

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



Corresponding interaction matrix:

$$\mathbf{L}_\rho = \begin{bmatrix} -\frac{\cos \theta}{Z} & -\frac{\sin \theta}{Z} & \frac{\rho}{Z} & (1 + \rho^2) \sin \theta & -(1 + \rho^2) \cos \theta & 0 \end{bmatrix}$$

$$\mathbf{L}_\theta = \begin{bmatrix} \frac{\sin \theta}{\rho Z} & \frac{-\cos \theta}{\rho Z} & 0 & \frac{\cos \theta}{\rho} & \frac{\sin \theta}{\rho} & -1 \end{bmatrix}$$

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

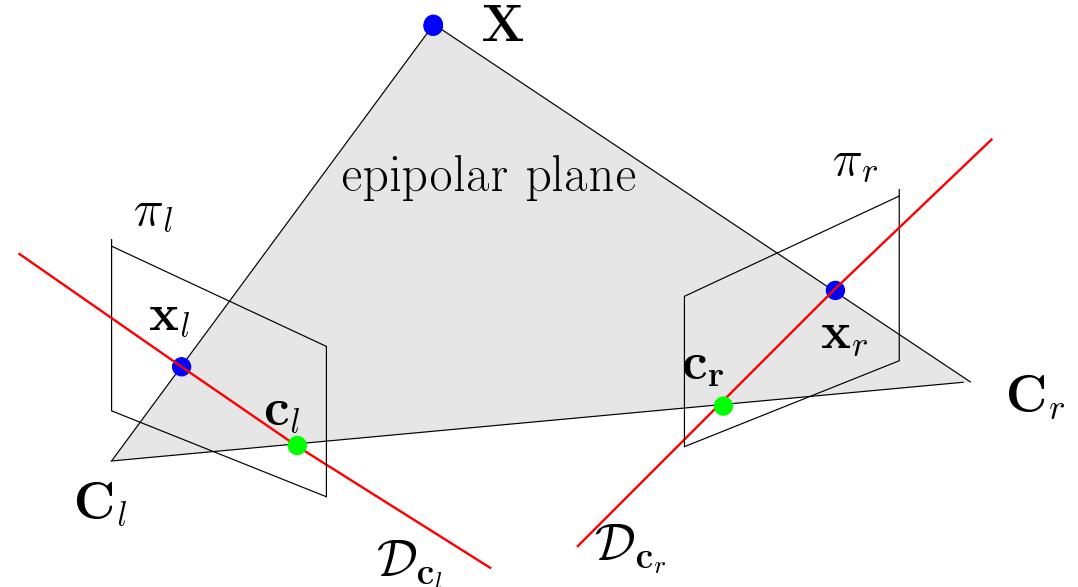
$$\dot{\mathbf{x}}_l = \mathbf{L}_{\mathbf{x}_l} \mathbf{v}_l$$

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

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

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

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



- Generalization to multi-cameras systems immediate

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

$$\mathbf{p}_i = \nu(\mathbf{P}_o) = \psi \circ \pi \circ \varphi^{-1}(\mathbf{P}_o)$$

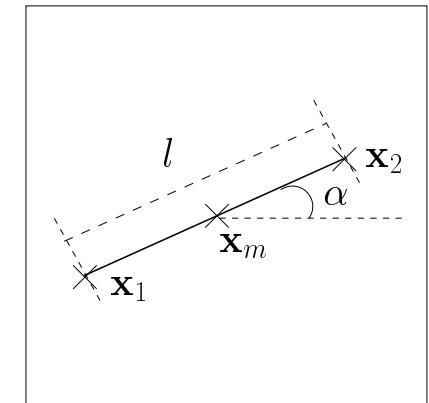
We also have $\mathbf{P}_o = \varphi \circ \delta(\mathbf{p}) \Rightarrow \mathbf{p}_i = \psi \circ \pi \circ \delta(\mathbf{p}) = \nu \circ \varphi \circ \delta(\mathbf{p})$

$$\begin{array}{ccccccc} W \subseteq SE_3 & \xrightarrow{\quad} & U \subseteq \mathcal{P}_o & \xrightarrow{\quad} & V \subseteq \mathcal{P}_i & & \\ (\mathbf{p}) & \xrightarrow{\delta} & (P_o) & \xrightarrow{\pi} & (p_i) & & \\ & & \downarrow \varphi & & \downarrow \psi & & \\ & & \mathbb{R}^n & \longrightarrow & \mathbb{R}^m & \longrightarrow & \mathbb{R}^k \\ & & (\mathbf{P}_o) & \xrightarrow{\quad} & (\mathbf{p}_i) & \xrightarrow{\sigma} & (\mathbf{s}) \\ & & \nu = \psi \circ \pi \circ \varphi^{-1} & & & & \end{array}$$

Finally $\mathbf{s} = \sigma(\mathbf{p}_i) \Rightarrow \mathbf{L}_s = \frac{\partial \mathbf{s}}{\partial \mathbf{p}_i} \frac{\partial \mathbf{p}_i}{\partial \mathbf{P}_o} \mathbf{L}_{\mathbf{P}_o}$

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

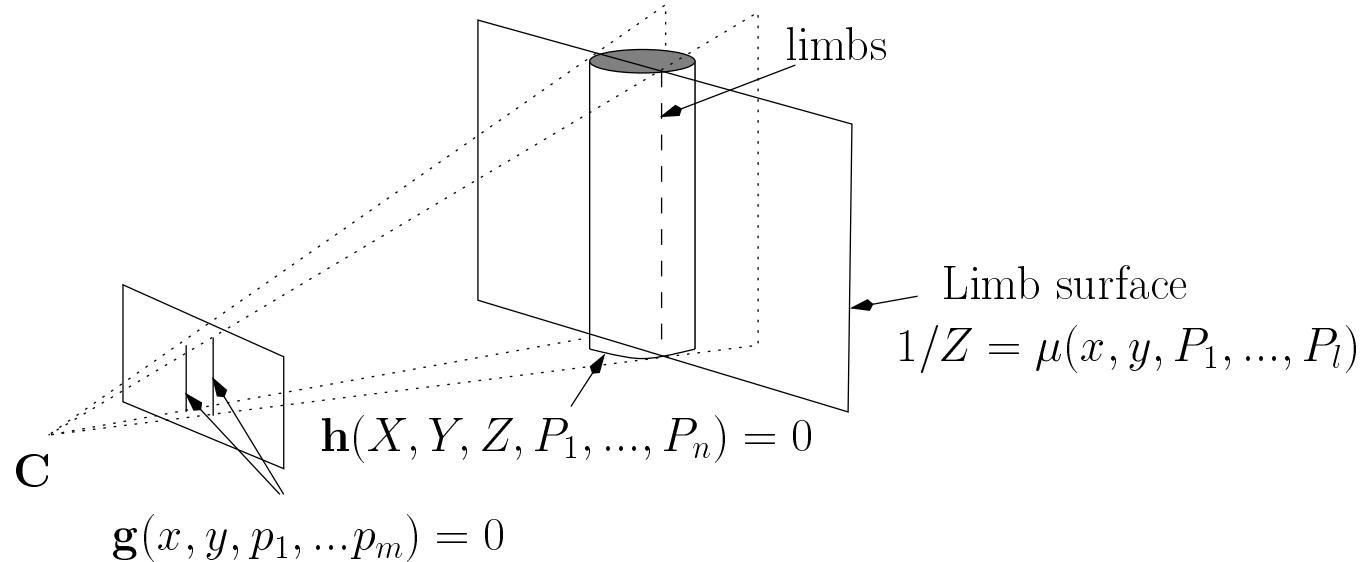
$$\begin{aligned}\mathbf{L}_{x_m} &= \begin{bmatrix} -1/Z_m & 0 & x_m/Z_m + D\epsilon_x & x_{mw} & -1 - x_{mwy} & y_m \end{bmatrix} \\ \mathbf{L}_{y_m} &= \begin{bmatrix} 0 & -1/Z_m & y_m/Z_m + D\epsilon_y & 1 + y_{mw} & y_{mwy} & -x_m \end{bmatrix} \\ \mathbf{L}_l &= \begin{bmatrix} -Dc & -Ds & l/Z_m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

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 a geometrical primitive



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

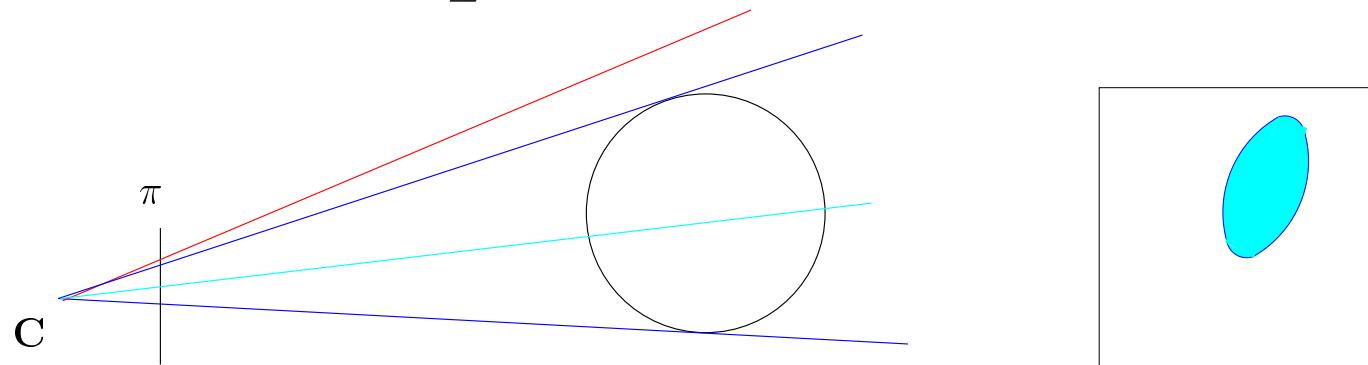
2D primitive : $\mathbf{g}(\mathbf{x}, \mathbf{p_i}) = 0$

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

2D visual features : case of the sphere

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

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



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

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

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

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

Direct computation of the interaction matrix

$$L_{p_i} = \frac{\partial p_i}{\partial P_o} L_{P_o}$$

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

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

We always have $\text{rank } L_{p_i} = \text{rank } L_{P_o} = 3$

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



Other (better) method

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

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

$$\Rightarrow \frac{\partial g}{\partial p_i}(x, p_i) \dot{p}_i = -\frac{\partial g}{\partial x}(x, p_i) L_{xy}(x, P_o) v, \forall x \in p_i$$

If $\dim(p_i) = \dim(p_i) = m$, using m points of p_i ,

we obtain a $m \times m$ linear system:

$$L_{p_i}(p_i, P_o) = \begin{bmatrix} \alpha_1(p_i) \\ \vdots \\ \alpha_m(p_i) \end{bmatrix}^{-1} \begin{bmatrix} \beta_1(p_i, P_o) \\ \vdots \\ \beta_m(p_i, P_o) \end{bmatrix} \text{ with } \begin{cases} \alpha_i(p_i) = \frac{\partial g}{\partial p_i}(x_i, p_i), i = 1 \text{ to } m \\ \beta_i(p_i, P_o) = -\frac{\partial g}{\partial x}(x_i, p_i) L_{xy}(x_i, P_o), \\ \quad i = 1 \text{ to } m \end{cases}$$

2D visual features : case of straight lines

$$h(\mathbf{X}, \mathbf{P}_o) = \begin{cases} h_1 = A_1 X + B_1 Y + C_1 Z = 0 \\ h_2 = A_2 X + B_2 Y + C_2 Z + D_2 = 0 \end{cases}$$

We obtain :

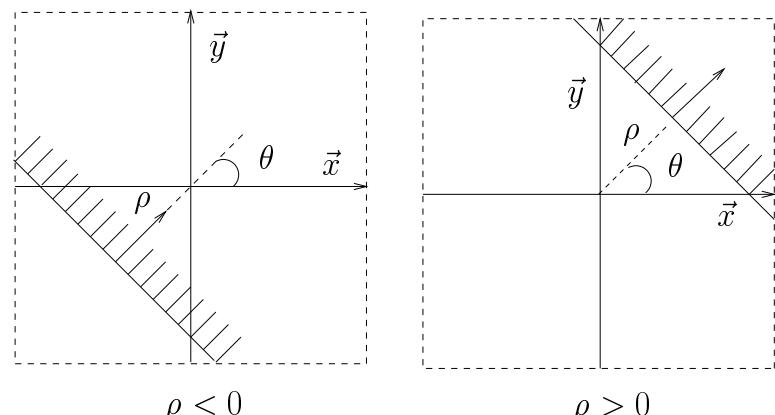
- function $\frac{1}{Z} = \mu(\mathbf{x}, \mathbf{P}_o)$ from $h_2 : 1/Z = Ax + By + C$
with $A = -A_2/D_2, B = -B_2/D_2, C = -C_2/D_2$
- 2D straight line $\mathcal{D} : A_1x + B_1y + C_1 = 0$

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

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

$$\text{with } \theta = \arctan(B_1/A_1)$$

$$\text{and } \rho = -C_1 / \sqrt{A_1^2 + B_1^2}.$$



Computation of the interaction matrix

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

From $\mathbf{g}(\mathbf{x}, \mathbf{p}_i) = 0$, we write $x = f(y, \rho, \theta)$ to get :

$$(-\dot{\theta}/\cos \theta) y + (\dot{\rho} + \rho \tan \theta \dot{\theta}) = y K_1(\mathbf{p}_i, \mathbf{P}_o) \mathbf{v} + K_2(\mathbf{p}_i, \mathbf{P}_o) \mathbf{v}, \forall y \in \mathbb{R}$$

We obtain
$$\begin{cases} \dot{\theta} = -K_1(\mathbf{p}_i, \mathbf{P}_o) \cos \theta \mathbf{v} \\ \dot{\rho} = (K_2(\mathbf{p}_i, \mathbf{P}_o) + K_1(\mathbf{p}_i, \mathbf{P}_o)\rho \sin \theta) \mathbf{v} \end{cases}$$

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

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

Exercise : obtain the same result using 2 points of \mathcal{D} ,
for example $(\rho \cos \theta, \rho \sin \theta)$ and $(\rho \cos \theta + \sin \theta, \rho \sin \theta - \cos \theta)$



2D visual features : case of a circle

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

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

Using $h_1 : \mathbf{g}(\mathbf{x}, \mathbf{p_i}) = x^2 + a_1y^2 + 2a_2xy + 2a_3x + 2a_4y + a_5 = 0$

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** : $\mu_{ij} = \iint_{\mathcal{D}(t)} (x - x_g)^i (y - y_g)^j dx dy$

$\mathbf{L}_{\mathbf{p_i}}$ 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:

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

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	$(\rho_1, \theta_1, \rho_2, \theta_2)$

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]

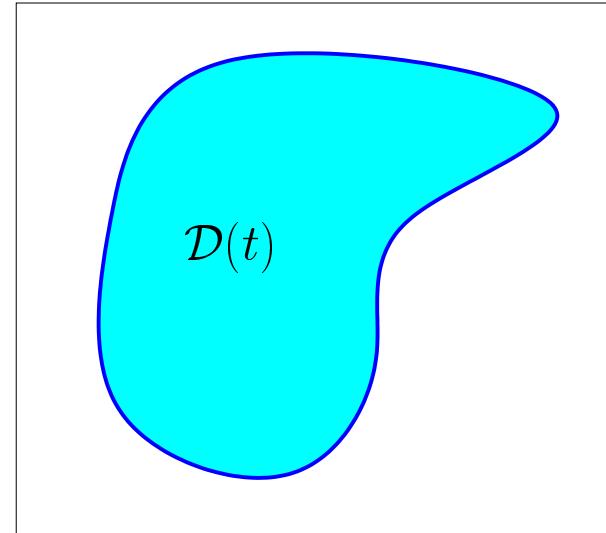
related to intuitive features:

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}

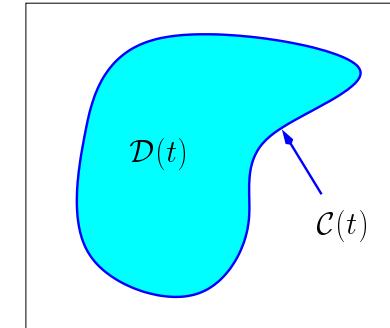
skewness : from m_{30} and m_{03}



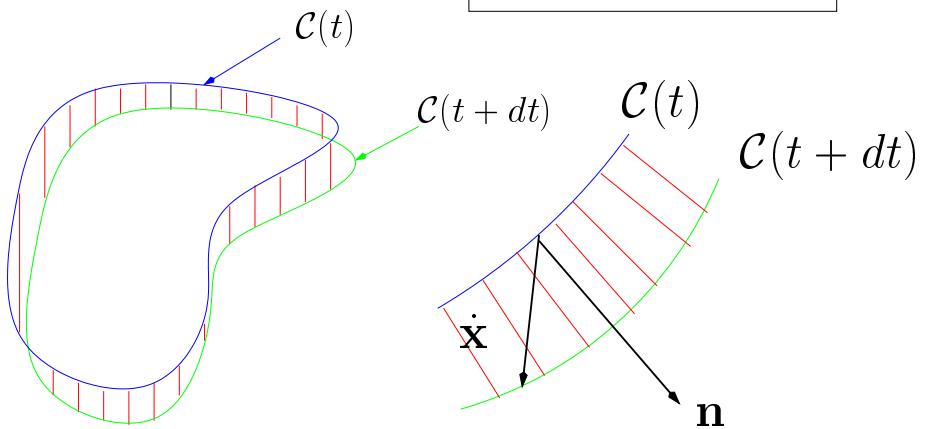
Interest in visual servoing: objects of complex or simple shape

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 :

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

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

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

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

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

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

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

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

Interaction matrix for moments

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

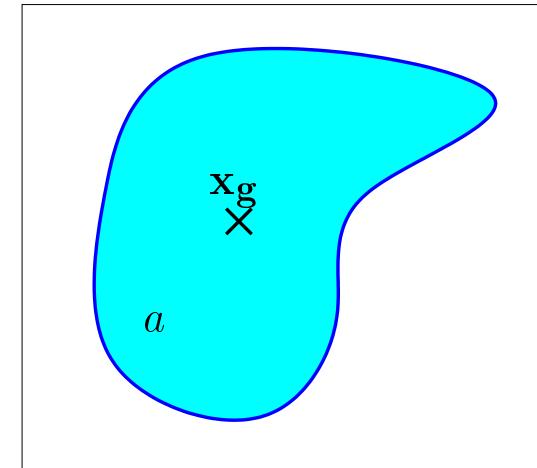
with $\left\{ \begin{array}{l} m_{vx} = -i (A m_{ij} + B m_{i-1,j+1} + C m_{i-1,j}) - A m_{ij} \\ m_{vy} = -j (A m_{i+1,j-1} + B m_{ij} + C m_{i,j-1}) - B m_{ij} \\ m_{vz} = (i+j+3)(A m_{i+1,j} + B m_{i,j+1} + C m_{ij}) - C m_{ij} \\ m_{wx} = (i+j+3) m_{i,j+1} + j m_{i,j-1} \\ m_{wy} = -(i+j+3) m_{i+1,j} - i m_{i-1,j} \\ m_{wz} = i m_{i-1,j+1} - j m_{i+1,j-1} \end{array} \right.$

$\mathbf{L}_{m_{ij}}$ 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: $x_g = m_{10}/m_{00}, y_g = m_{01}/m_{00}$

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

(generalization of the pure point case)

$$\left\{ \begin{array}{l} 1/Z_g = Ax_g + By_g + C \\ \epsilon_1 = 4(A n_{20} + B n_{11}) \\ \epsilon_2 = 4(A n_{11} + B n_{02}) \end{array} \right.$$

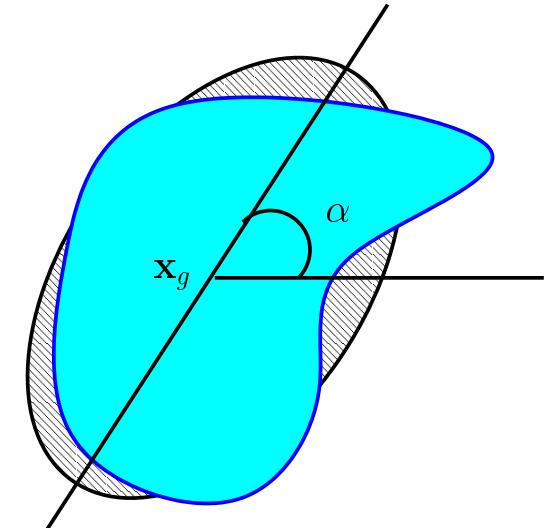
$$n_{ij} = \mu_{ij}/a \text{ with}$$

$$\left\{ \begin{array}{l} \mu_{20} = m_{20} - ax_g^2 \\ \mu_{02} = m_{02} - ay_g^2 \\ \mu_{11} = m_{11} - ax_g y_g \end{array} \right.$$

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

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

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



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

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

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

Cooking moments

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

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

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

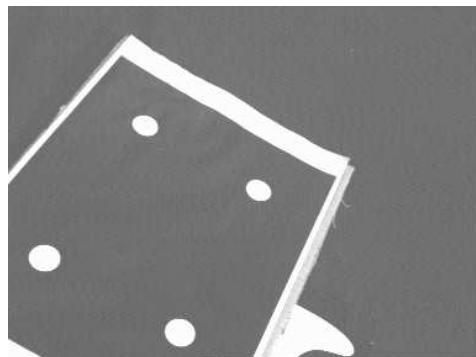
Pure image-based, but so near from position-based...

- 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 :

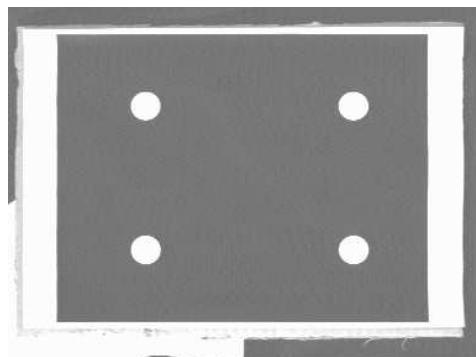
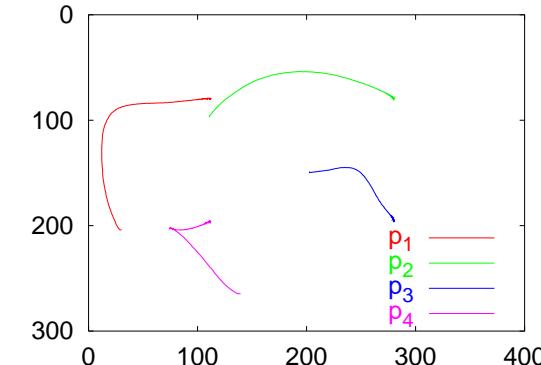
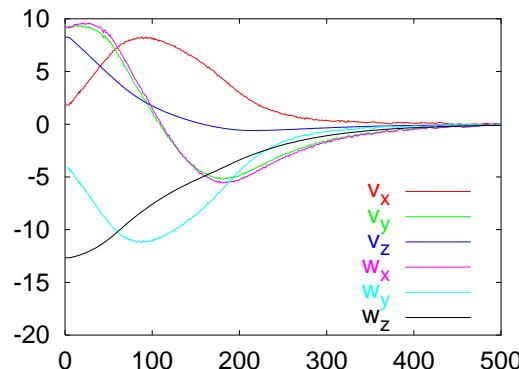
$$\mathbf{L}_{r_i r_j}^{\parallel} = \begin{bmatrix} 0 & 0 & 0 & r_{iwx} & r_{iwy} & 0 \\ 0 & 0 & 0 & r_{jwx} & r_{jwy} & 0 \end{bmatrix} \quad (A=B=0)$$

Interest of cooking visual features

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

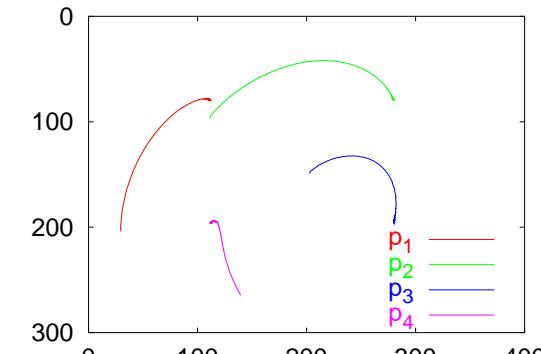
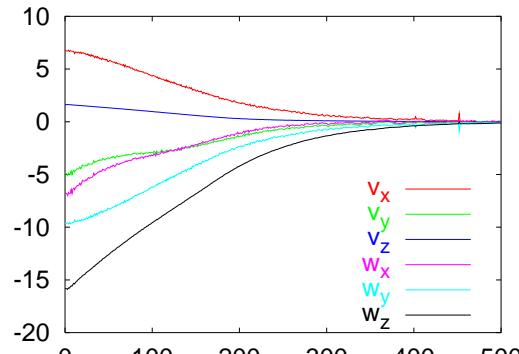


Initial image



Desired image

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



Camera velocity

Image trajectories

Other approach: direct numerical estimation

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

- Off-line learning of L_s :

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

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

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

- Off-line learning of L_s^+ (better method):

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

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

- Other methods: neural networks,...

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

Stability impossible to demonstrate



Other approach: direct numerical estimation

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

$$\widehat{\mathbf{L}}_{\mathbf{s}}(t+1) = \widehat{\mathbf{L}}_{\mathbf{s}}(t) + \frac{\alpha}{\mathbf{v}_c^\top \mathbf{v}_c} \left(\dot{\mathbf{s}} - \widehat{\mathbf{L}}_{\mathbf{s}}(t) \mathbf{v}_c \right) \mathbf{v}_c^\top$$

Be careful to initial value $\widehat{\mathbf{L}}_{\mathbf{s}}(t_0)$

Stability impossible to demonstrate

May be useful for unknown complex objects or unmodeled systems

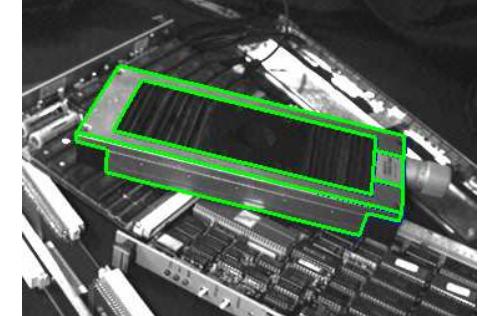
- Modeling issues
 - ▷ Basics
 - ▷ 2D visual features
 - ⇒ 3D visual features
 - ▷ Omni-directional vision sensor, vision + structured light



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 CAD model: \mathbf{X}
- an estimation of the camera intrinsic parameters: x_c, y_c, f_x, f_y



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

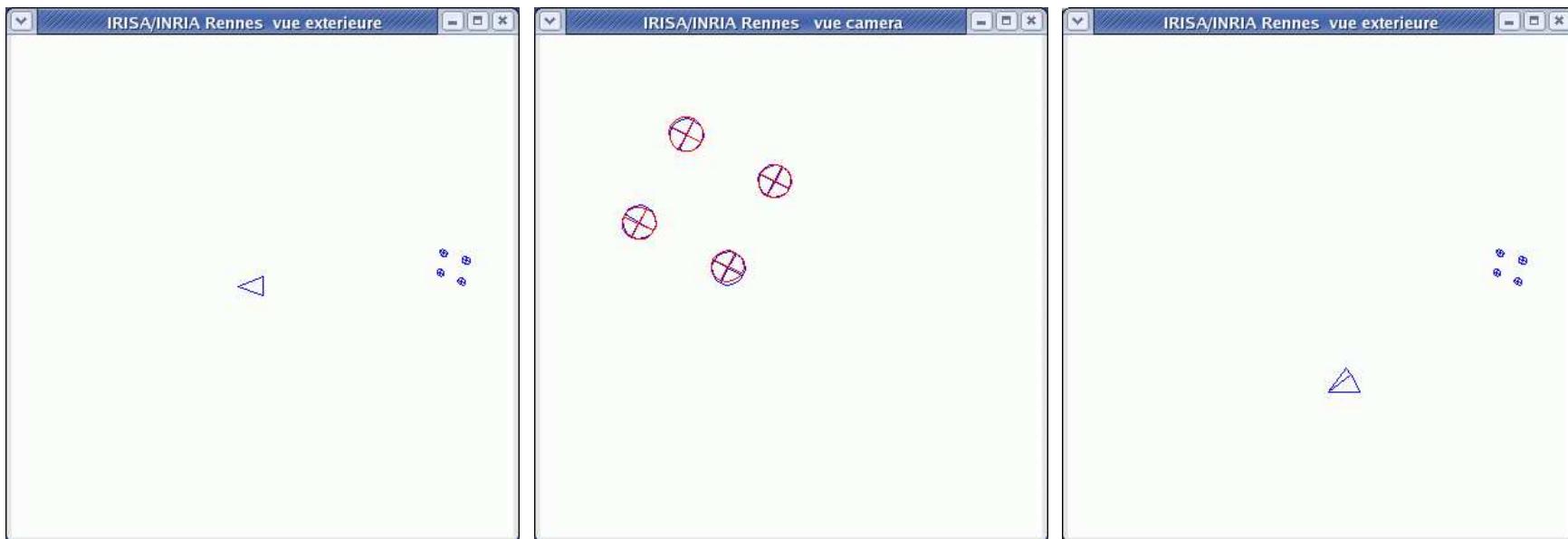
Pose estimation problem \sim camera calibration problem
(intrinsic camera parameters already known)

3D visual features with one camera

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

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

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:

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

- parameters $\theta \mathbf{u}$ that represent rotation ${}^c \mathbf{R}_c$

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

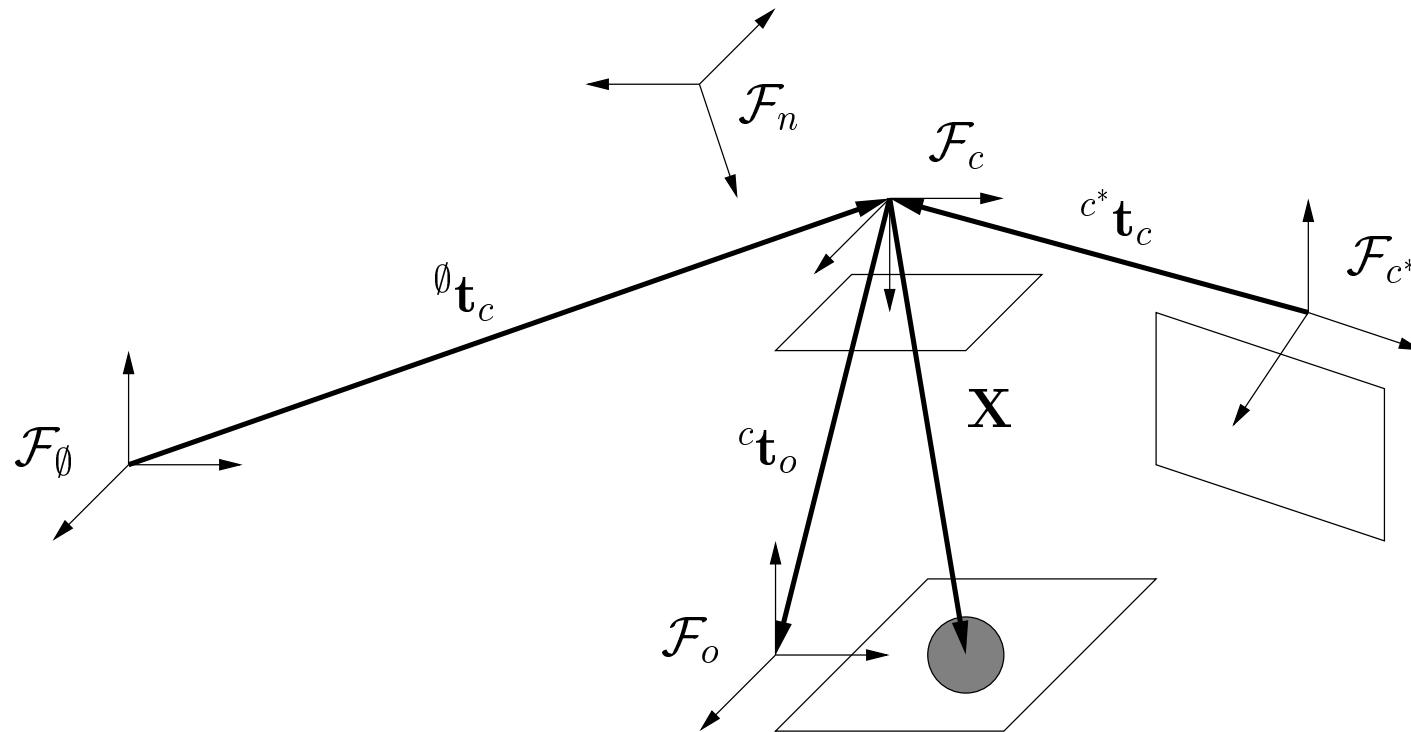
- parameters $\theta \mathbf{u}$ that represent rotation ${}^c \mathbf{R}_{c^*}$

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

In both cases, \mathbf{L}_{ω} such that $\mathbf{L}_{\omega} \theta \mathbf{u} = \mathbf{L}_{\omega}^{-1} \theta \mathbf{u} = \pm \theta \mathbf{u}$

- coordinates of a 3D point \mathbf{X} : $\dot{\mathbf{X}} = -\mathbf{v} - [\omega]_{\times} \mathbf{X} \Rightarrow \mathbf{L}_{\mathbf{X}} = [-\mathbf{I}_3 \ [\mathbf{X}]_{\times}]$

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



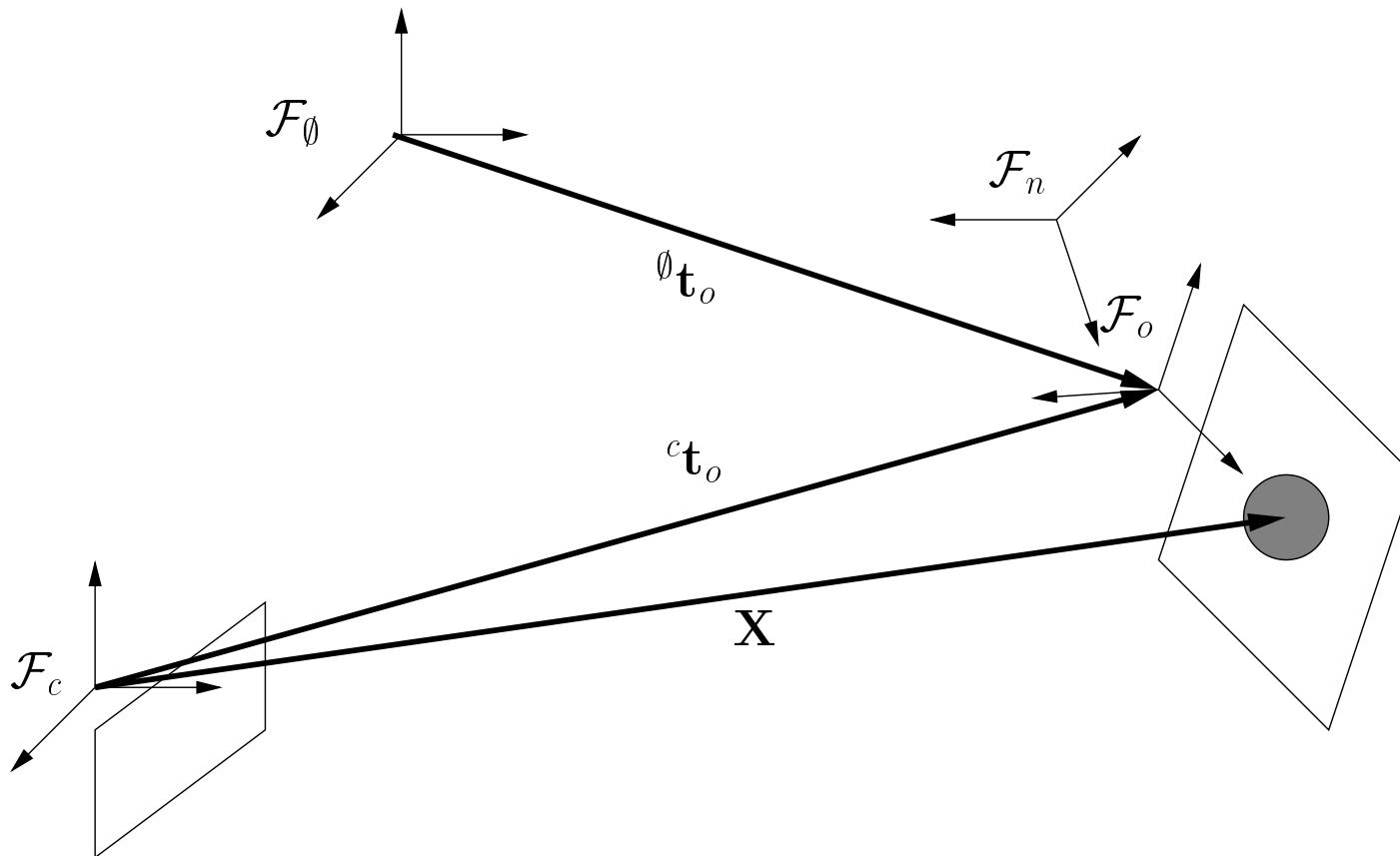
$$\mathbf{L}_{c\mathbf{t}_o} = [-\mathbf{I}_3 \quad [c\mathbf{t}_o]_{\times}]$$

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

$$\mathbf{L}_{o\mathbf{t}_c} = [{}^o\mathbf{R}_c \quad \mathbf{0}_3]$$

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

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



$$\mathbf{L}_{c\mathbf{t}_o} {}^c\mathbf{V}_o = \begin{bmatrix} -{}^c\mathbf{R}_o & \mathbf{0}_3 \end{bmatrix}$$

$$\emptyset\dot{\mathbf{t}}_o = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} \emptyset\mathbf{v}_o$$

- Modeling issues
 - ▷ Basics
 - ▷ 2D visual features
 - ▷ 3D visual features
- ⇒ Omni-directional vision sensor, vision + structured light



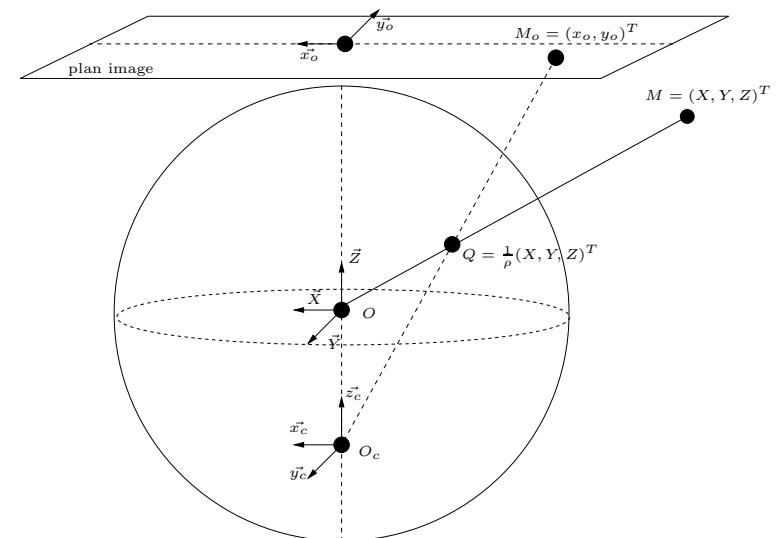
Modeling for omnidirectional vision [Nayar 01]

Single viewpoint systems

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

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

- $\eta=1, \xi=1$: 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]:

$$\mathbf{L}_{x_0y_0} = \begin{bmatrix} -\frac{1}{\rho} \left(\frac{\eta\gamma-\xi}{\nu} - \xi x_o^2 \right) & \frac{\xi x_0 y_0}{\rho} & \frac{x_0 \gamma}{\rho} & \eta x_0 y_0 & -\frac{\eta-\xi\gamma}{\nu} - \eta x_o^2 & y_0 \\ \frac{\xi x_0 y_0}{\rho} & -\frac{1}{\rho} \left(\frac{\eta\gamma-\xi}{\nu} - \xi y_o^2 \right) & \frac{y_0 \gamma}{\rho} & \frac{\eta-\xi\gamma}{\nu} + \eta y_o^2 & -\eta x_0 y_0 & -x_o \end{bmatrix}$$

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:

$$\eta = 1, \xi = 1, \nu = 0, \gamma = 1, \frac{\eta\gamma-\xi}{\nu} = \frac{1+x_o^2+y_o^2}{2}, \frac{\eta-\xi\gamma}{\nu} = \frac{1-x_o^2-y_o^2}{2}$$

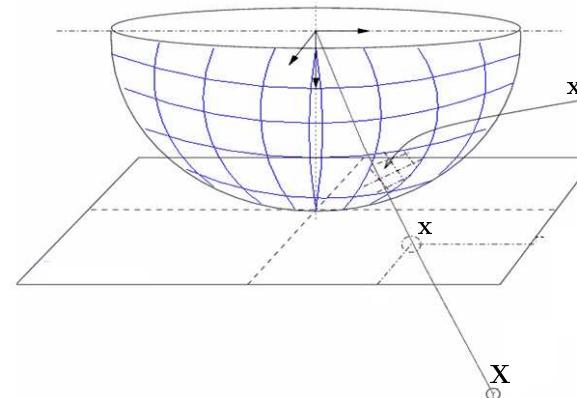
For the image of a straight line, i.e. an ellipse, see [Mezouar, IROS 04]



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

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

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



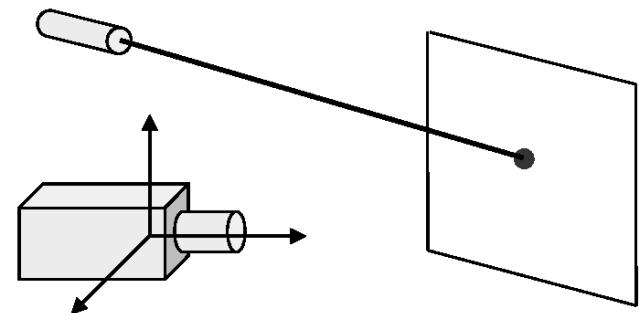
$$\mathbf{L}_{\mathbf{x}_s} = \begin{bmatrix} -\frac{1}{\rho} (1 - x_s^2) & \frac{x_s y_s}{\rho} & \frac{x_s z_s}{\rho} & 0 & -z_s & y_s \\ \frac{x_s y_s}{\rho} & -\frac{1}{\rho} (1 - y_s^2) & \frac{y_s z_s}{\rho} & z_s & 0 & -x_s \\ \frac{x_s z_s}{\rho} & \frac{y_s z_s}{\rho} & -\frac{1}{\rho} (1 -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

Passivity property ($\|\dot{\mathbf{x}}_s\|$ independent of ω) [Hamel-Mahony 02]

Invariance property: for instance $\mathbf{L}_{a_s} = [a_x \ a_y \ a_z \ 0 \ 0 \ 0]$

Modeling for coupling vision and structured light

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



- 1) Modeling issues
- ⇒ 2) Control issues
 - ▷ Control of visual tasks ($m = n$)
 - ▷ Classification of the visual tasks
 - ▷ Hybrid tasks ($m < n$)

Control in visual servoing

Regulation of a task function: $\mathbf{e}(\mathbf{p}(t)) = \mathbf{C} (\mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^*)$

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]



Visual task function

With k visual features \mathbf{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.)
⇒ **Hybrid tasks**

- If $m = n$, all the robot dof are controlled
using the **visual task function** :

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

where \mathbf{C} is a $m \times k$ combination matrix of full rank m .

Control law

Since $e(p(t)) = C(s(p(t)) - s^*)$, we have

$$\dot{e} = L_e v_q + \frac{\partial e}{\partial t} \quad \text{where} \quad \begin{cases} v_q = v_c \text{ for eye-in-hand system} \\ v_q = -v_o \text{ for eye-to-hand system} \end{cases}$$

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

$$v_q = L_e^{-1} \left(-\lambda e - \frac{\partial e}{\partial t} \right) \quad \text{with } L_e = CL_s \text{ if } C \text{ is constant}$$

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

$$v_q = \widehat{L}_e^{-1} \left(-\lambda e - \frac{\widehat{\partial e}}{\partial t} \right)$$

Stability analysis

Behavior of the closed-loop system :

$$\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{v}_q + \frac{\partial \mathbf{e}}{\partial t} = -\lambda \widehat{\mathbf{L}_e \mathbf{L}_e^{-1}} \mathbf{e} - \widehat{\mathbf{L}_e \mathbf{L}_e^{-1}} \frac{\partial \mathbf{e}}{\partial t} + \frac{\partial \mathbf{e}}{\partial t}$$

If $\frac{\partial \mathbf{e}}{\partial t} = \widehat{\frac{\partial \mathbf{e}}{\partial t}} = 0$, $\|\mathbf{e}\|$ always decreases (global stability) if

$$\widehat{\mathbf{L}_e \mathbf{L}_e^{-1}} > 0$$

To suppress tracking errors and obtain the desired behavior $\dot{\mathbf{e}} = -\lambda \mathbf{e}$:

$$\widehat{\mathbf{L}_e} = \mathbf{L}_e \quad \text{and} \quad \widehat{\frac{\partial \mathbf{e}}{\partial t}} = \frac{\partial \mathbf{e}}{\partial t}$$

In practice ($m = n$)

- If $k = m$, $\mathbf{C} = \mathbb{I}_m$, $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ $\Rightarrow \mathbf{L}_e = \mathbf{L}_s$, $\widehat{\mathbf{L}}_e = \widehat{\mathbf{L}}_s$

$$\mathbf{v}_q = -\lambda \widehat{\mathbf{L}}_s^{-1} (\mathbf{s} - \mathbf{s}^*) - \widehat{\mathbf{L}}_s^{-1} \frac{\partial \mathbf{s}}{\partial t} \quad \text{stable if } \mathbf{L}_s \widehat{\mathbf{L}}_s^{-1} > 0$$

- If $k > m$, $\mathbf{C} = \widehat{\mathbf{L}}_s^+|_{\mathbf{s}=\mathbf{s}^*}$ (1) or $\mathbf{C} = \widehat{\mathbf{L}}_s^+$ (2) , $\widehat{\mathbf{L}}_e = \mathbb{I}_n$,

$$\mathbf{v}_q = -\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t} \text{ stable if :}$$

(1) $\widehat{\mathbf{L}}_s^+|_{\mathbf{s}=\mathbf{s}^*} \mathbf{L}_s > 0$ (only around \mathbf{s}^*)

(2) $\widehat{\mathbf{L}}_s^+ \mathbf{L}_s > 0$ (only locally since \mathbf{C} not constant)

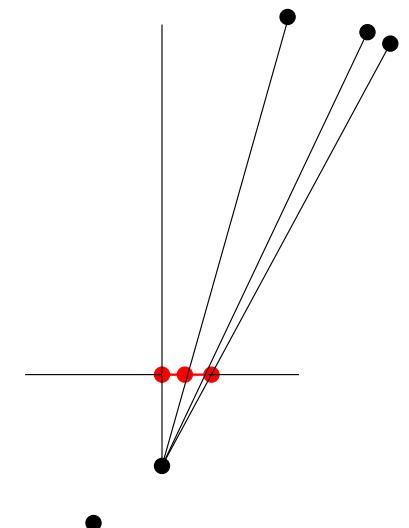
If translational dof are controlled and 2D visual features are used,
an estimation $\widehat{\mathbf{P}}$ or $\widehat{\mathbf{P}}^*$ is necessary to compute $\widehat{\mathbf{L}}_s$

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

Case of a pan-tilt camera observing a point :

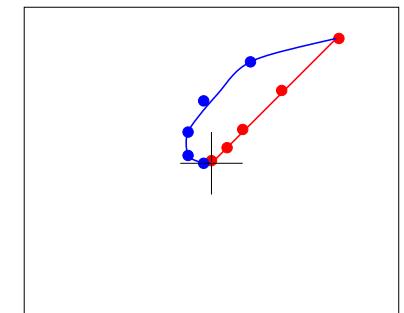
$$\mathbf{s} = (x, y), \mathbf{s}^* = (0, 0), \mathbf{C} = \mathbb{I}_2$$

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



$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_{\mathbf{s}}}^{-1} (\mathbf{s} - \mathbf{s}^*)$$

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



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

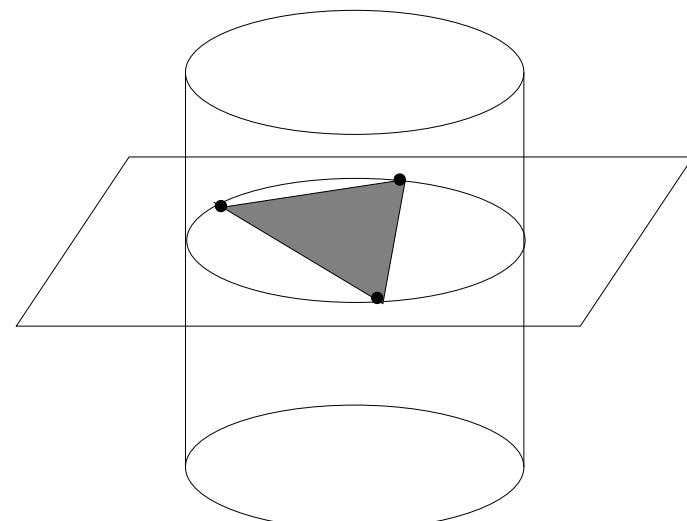
Control when $k = m = n = 6$

$$\mathbf{C} = \mathbb{I}_6 \quad \Rightarrow \quad \mathbf{v}_q = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}}^{-1}(\mathbf{s} - \mathbf{s}^*) \text{ stable if } \mathbf{L}_{\mathbf{s}} \widehat{\mathbf{L}}_{\mathbf{s}}^{-1} > 0$$

- Impossible with only 2D visual features

Using 3 points ($\mathbf{L}_{\mathbf{s}}$ is 6×6)

- pose ambiguity (4 solutions)
- possible singularity of $\mathbf{L}_{\mathbf{s}}$



Control when $k = m = n = 6$

- Possible with 3D visual features

For instance, if $\mathbf{s} = \begin{bmatrix} {}^c\mathbf{t}_c \\ \theta\mathbf{u} \end{bmatrix}$, $\mathbf{v}_c = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = -\lambda \begin{bmatrix} {}^c\mathbf{R}_c {}^c\mathbf{t}_c \\ \theta\mathbf{u} \end{bmatrix}$

Advantages

- \mathbf{L}_s block-diagonal and never singular
- 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 $k = m = n = 6$: 2 1/2 D visual servoing

Idea : Combine 2D image data and 3D data

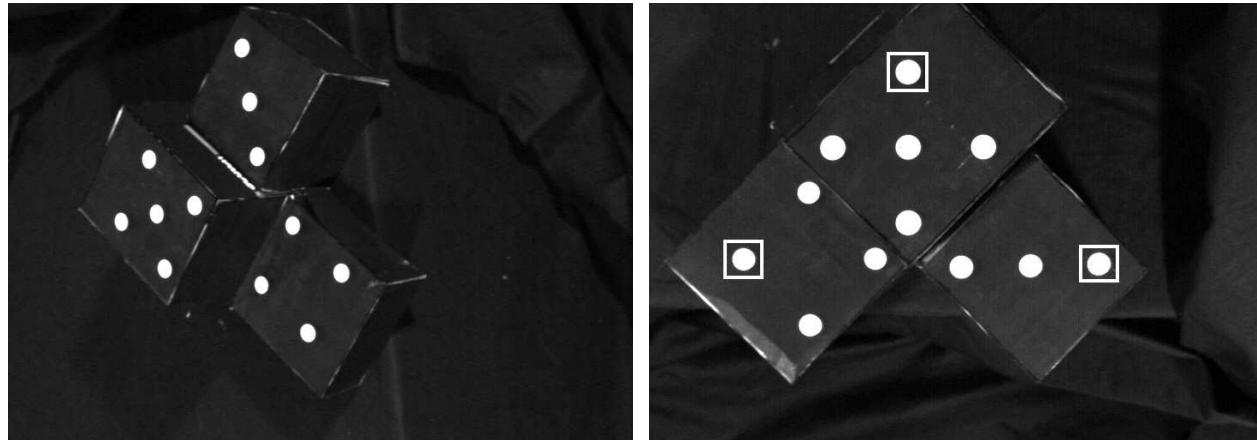
$$\mathbf{s} = \begin{bmatrix} x \\ y \\ \log Z \\ \theta u_x \\ \theta u_y \\ \theta u_z \end{bmatrix} \quad \left. \begin{array}{l} \text{image point} \\ \text{coordinates} \\ \rightarrow \text{rel. depth} \\ \text{rotation} \\ \text{to} \\ \text{realize} \end{array} \right\}$$

$\Rightarrow \mathbf{L}_s$ triangular
and never singular

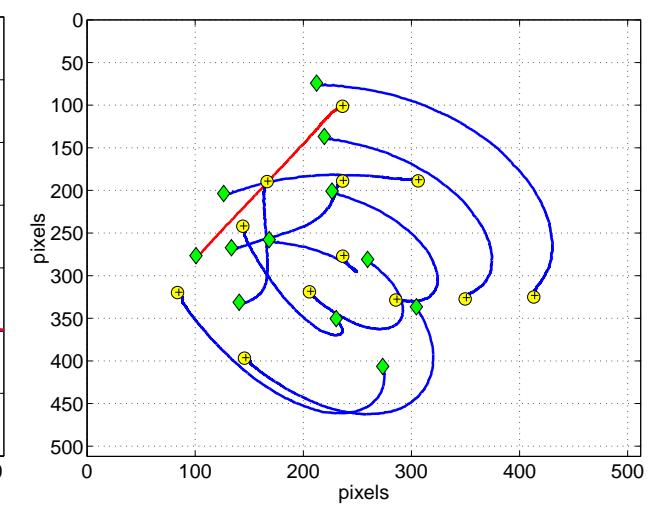
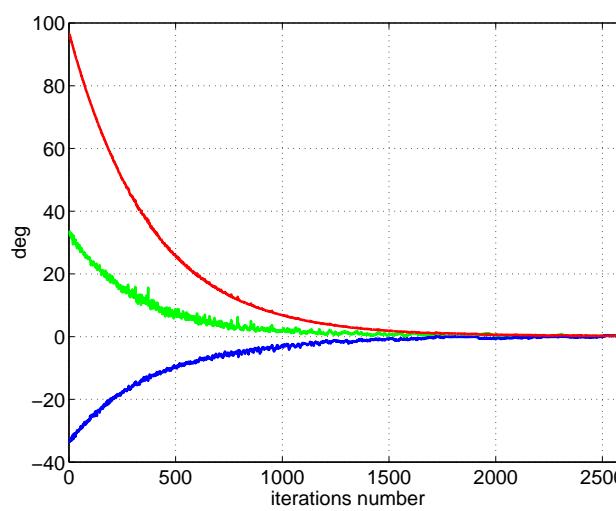
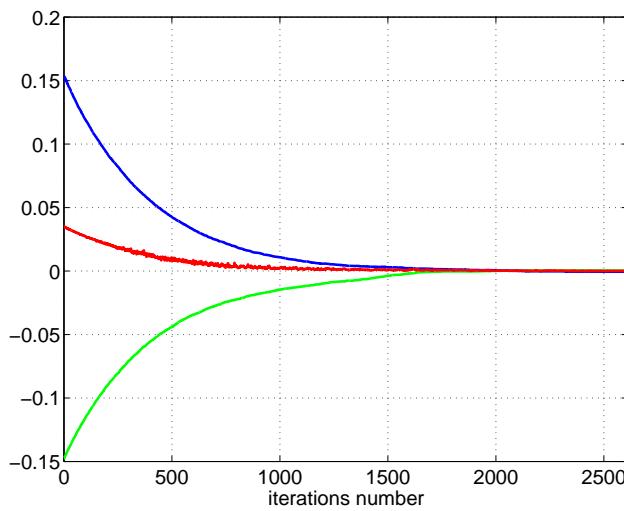
Advantages :

- Decoupled control scheme (image point trajectory : straight line)
- Analytical conditions for global stability possible
in presence of calibration errors
- No 3D CAD model needed, only one scalar unknown

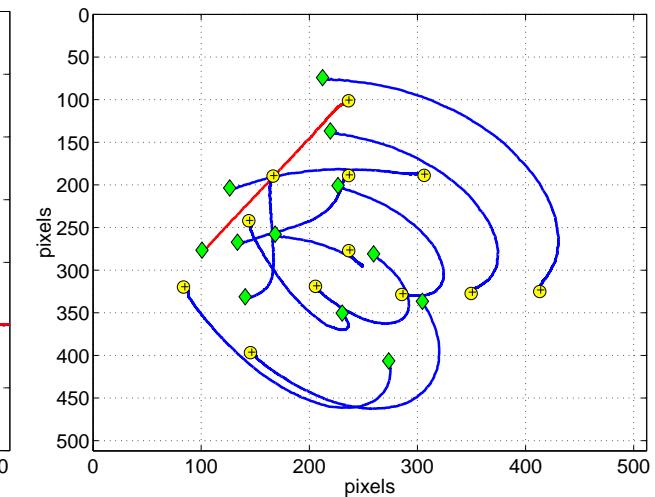
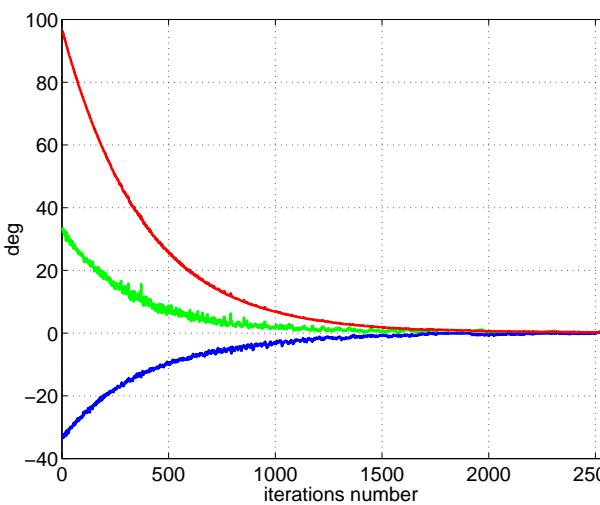
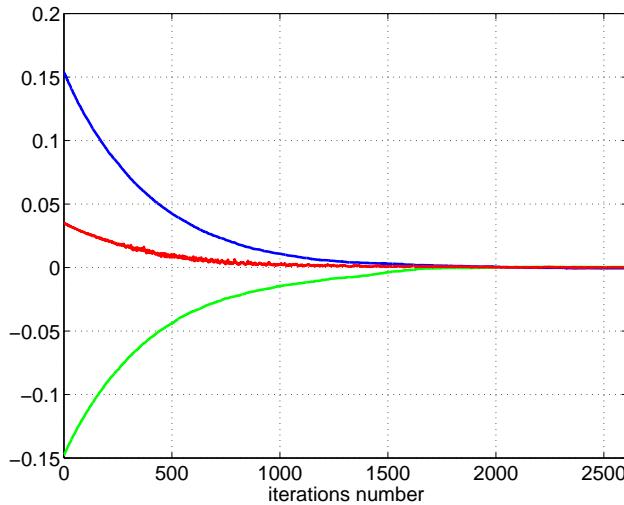
Results



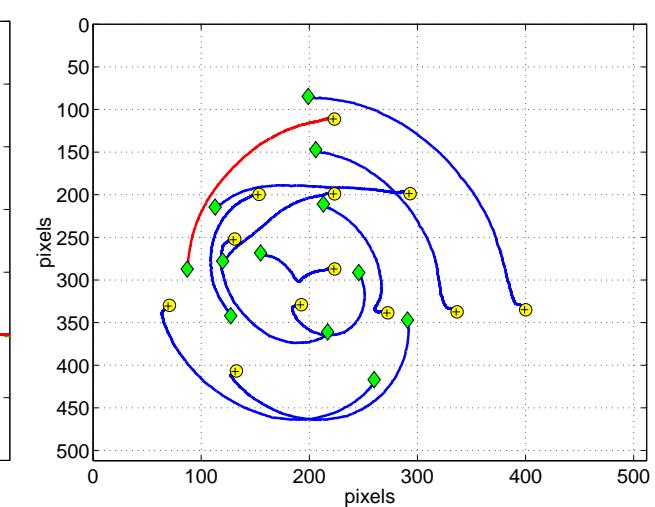
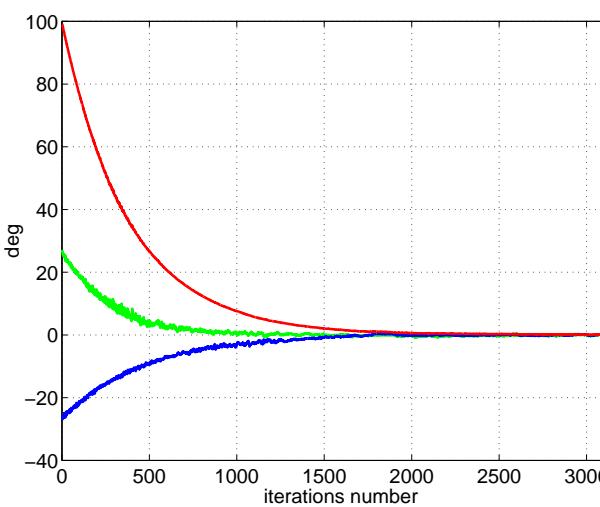
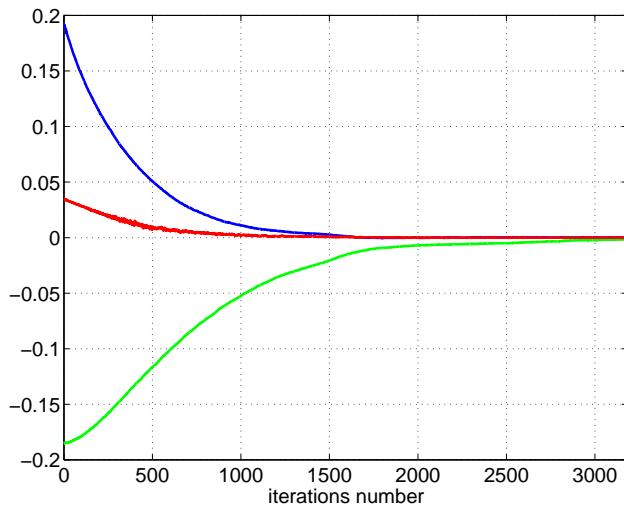
Coarse calibration



Coarse calibration



Bad calibration



Other 2 1/2 D VS scheme

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

Advantages:

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

Drawback:

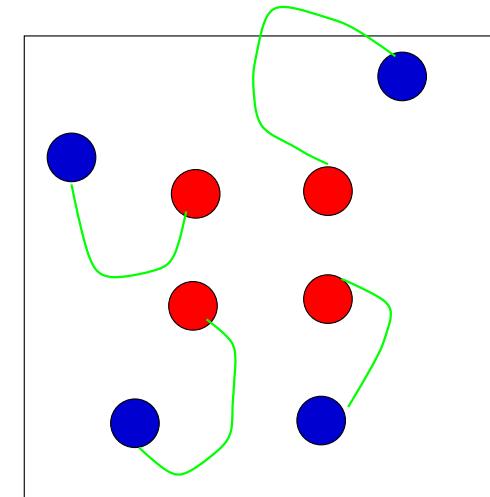
\mathbf{L}_s only block-triangular

\Rightarrow analytical conditions for global stability difficult to obtain

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

- Control law $\mathbf{v}_q = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ (\mathbf{s} - \mathbf{s}^*)$
stable if $\widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ \mathbf{L}_{\mathbf{s}} > 0$ (only around \mathbf{s}^*)

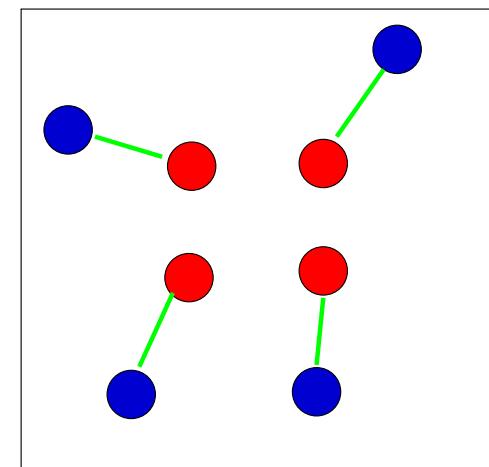
No real control of the image trajectories



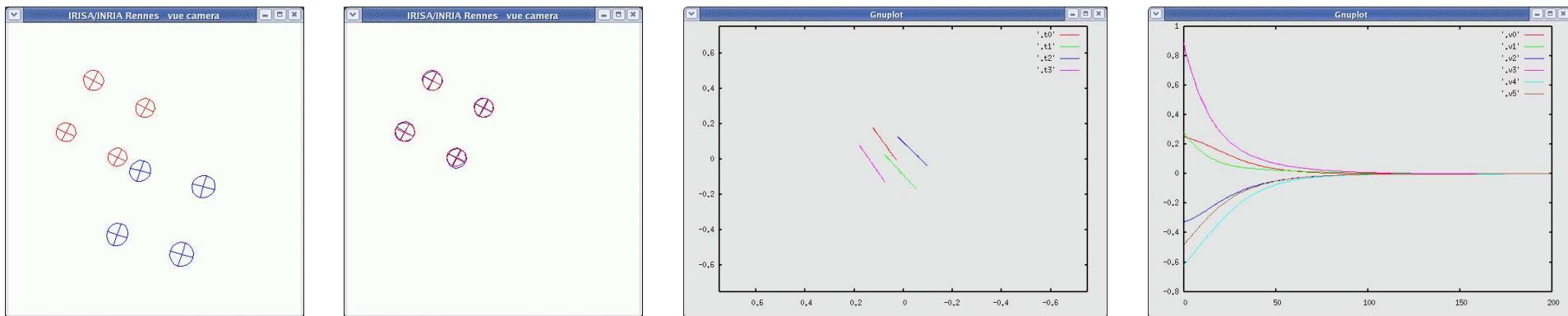
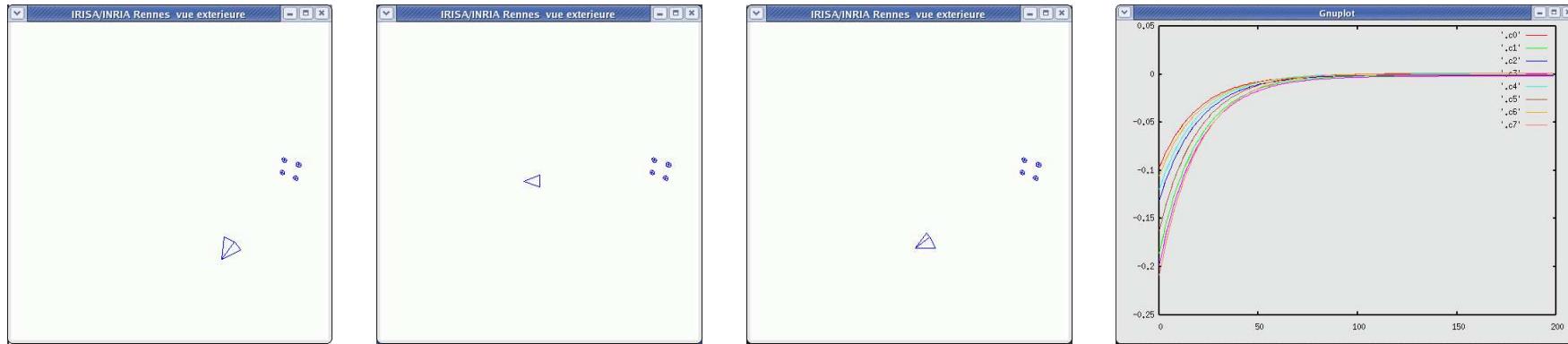
- Control law $\mathbf{v}_q = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}}^+ (\mathbf{s} - \mathbf{s}^*)$
tries to ensure $\dot{\mathbf{s}} = -\lambda (\mathbf{s} - \mathbf{s}^*)$

Possible local minima

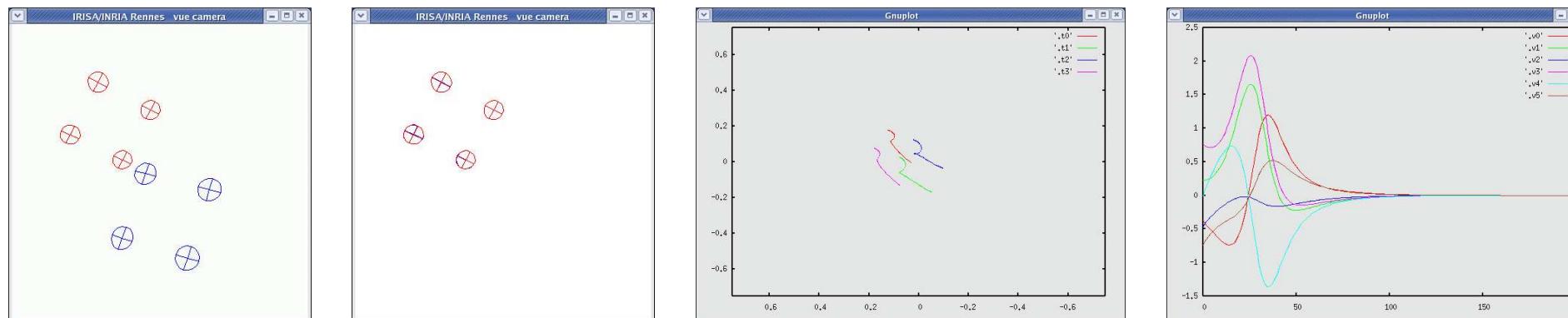
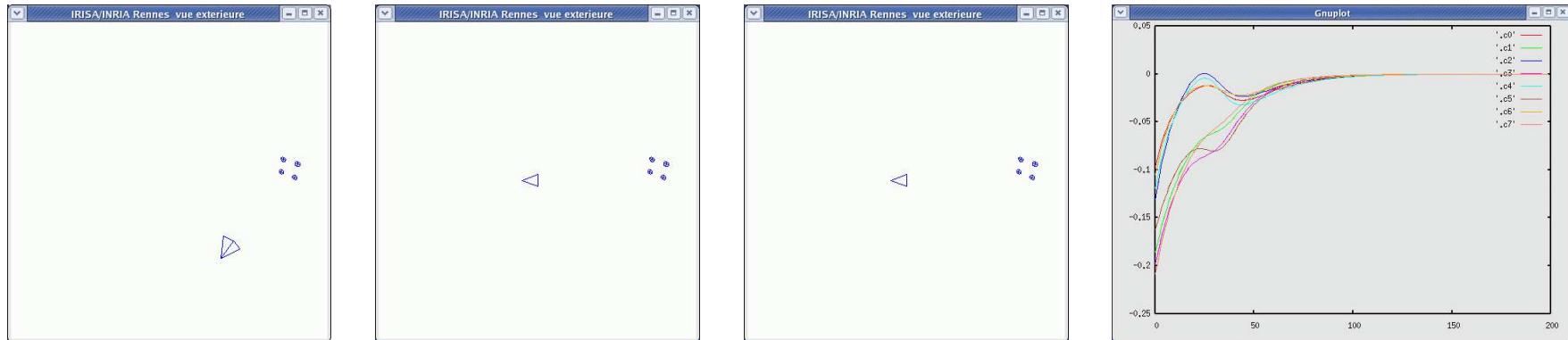
because \mathbf{C} ($= \widehat{\mathbf{L}}_{\mathbf{s}}^+$) is not constant



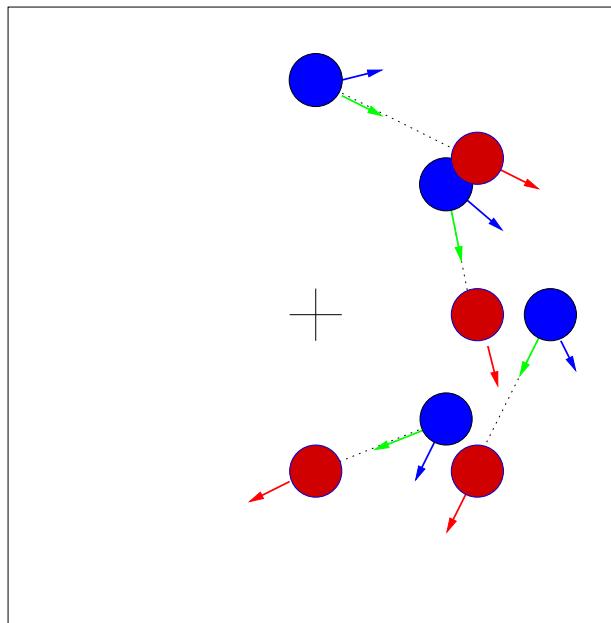
Reaching a local minimum using \widehat{L}_s^+



Reaching the global minimum using $\widehat{L}_s|_{s=s^*}$



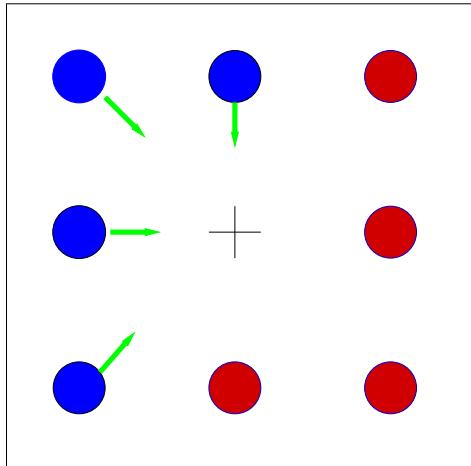
Geometrical interpretation of the behavior obtained



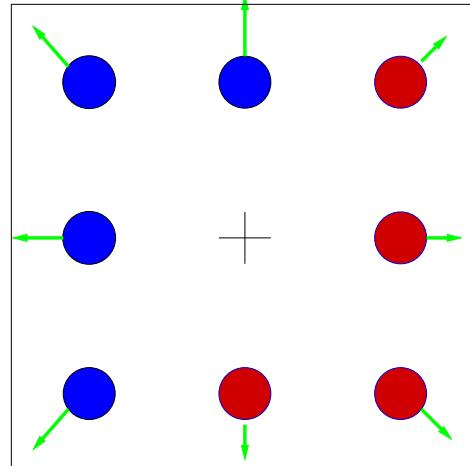
Rotation of 45^o

Reaching a singularity of L_s

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

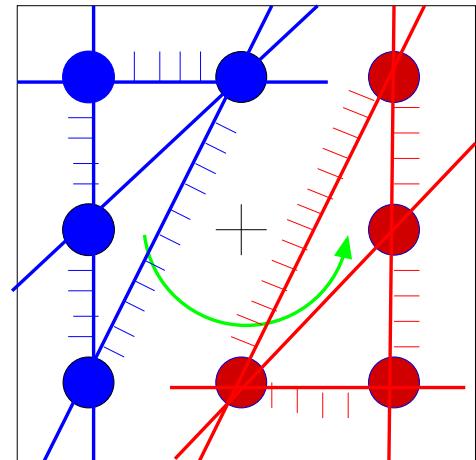


Using \widehat{L}_s^+



Using $\widehat{L}_{s|s=s^*}^+$

Other choice



Using \widehat{L}_s^+ or $\widehat{L}_{s|s=s^*}^+$

At singularity, rank $L_s = 2$.

Perfect behavior if s is composed of 2D straight lines parameters
(or 2D moments, or 2.5D VS, or 3D VS)

Target tracking

PI Controller

Integral term classical in Automatic Control to reduce tracking errors

If \mathbf{I}_k is the estimation of $\frac{\partial \mathbf{e}}{\partial t}$ at iteration k , we have :

$$\begin{aligned}\mathbf{I}_{k+1} &= I_k + \mu \mathbf{e}_k \text{ with } \mathbf{I}_0 = 0 \\ &= \mu \sum_{j=0}^k \mathbf{e}_j\end{aligned}$$

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:

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

with

$$\left\{ \begin{array}{l} \widehat{\mathbf{L}_e} = \mathbf{C} \widehat{\mathbf{L}_s} \\ \widehat{\dot{\mathbf{e}}}_k = \frac{\mathbf{e}_k - \mathbf{e}_{k-1}}{\Delta t} = \mathbf{C} \frac{\mathbf{s}_k - \mathbf{s}_{k-1}}{\Delta t} \end{array} \right.$$

A Kalman filter may then be used.

⇒ Control issues

▷ Control of visual tasks ($m = n$)

⇒ Classification of the visual tasks

▷ Hybrid tasks ($m < n$)

Classification of the vision-based tasks

\mathcal{S}^* = set of motions such that $\dot{\mathbf{s}} = 0 : \mathcal{S}^* = \text{Ker } \mathbf{L}_{\mathbf{s}}$

A **virtual link** between the camera and the scene is defined by a set of compatible constraints : $\mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^* = 0$

A virtual link is characterized by \mathcal{S}^* since $\mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^* = 0 \Rightarrow \dot{\mathbf{s}} = 0$

class of the virtual link = dimension N of \mathcal{S}^* .

The k constraints involved by the visual features are independent if $k = 6 - N$.

If $k > 6 - N$, the visual features are redundant.



Case of a point

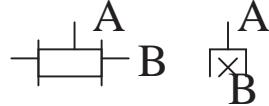
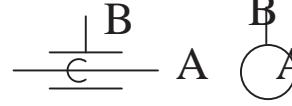
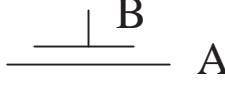
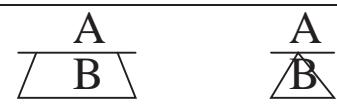
$$\mathbf{s} = (x, y)$$

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

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

⇒ Link of class 4



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 rectilinear	4	2	2	
Linear annular	4	1	3	
Point	5	2	3	

Rigid link

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

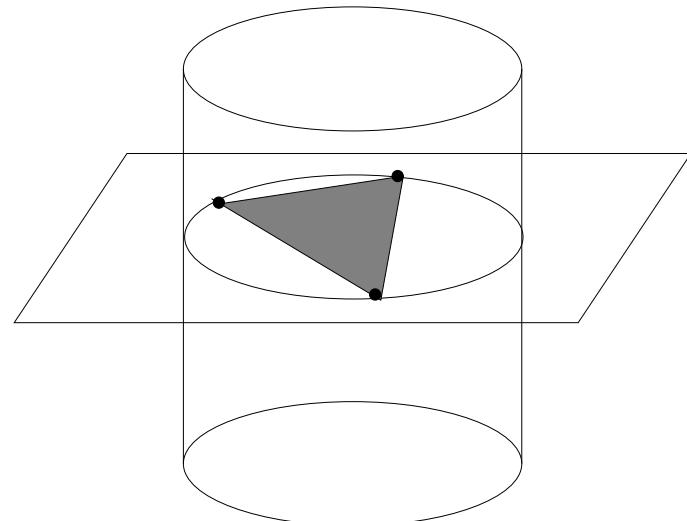
Using 3 points :

$$\mathbf{L}_s = \begin{bmatrix} -1/Z_1 & 0 & x_1/Z_1 & x_1y_1 & -(1+x_1^2) & y_1 \\ 0 & -1/Z_1 & y_1/Z_1 & 1+y_1^2 & -x_1y_1 & -x_1 \\ -1/Z_2 & 0 & x_2/Z_2 & x_2y_2 & -(1+x_2^2) & y_2 \\ 0 & -1/Z_2 & y_2/Z_2 & 1+y_2^2 & -x_2y_2 & -x_2 \\ -1/Z_3 & 0 & x_3/Z_3 & x_3y_3 & -(1+x_3^2) & y_3 \\ 0 & -1/Z_3 & y_3/Z_3 & 1+y_3^2 & -x_3y_3 & -x_3 \end{bmatrix}$$

Isolated singularities exist

4 poses are solution of the P3P problem

Solution : Using at least 4 points



Prismatic link

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

Using 3 (horizontal) straight lines

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

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

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

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



Bearing

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

\Rightarrow Image of the sphere = centered circle

$$\begin{aligned}\mathbf{L}_{x_c} &= \begin{bmatrix} -1/Z_c & 0 & 0 & 0 & -1 - r^2 & 0 \end{bmatrix} \\ \mathbf{L}_{y_c} &= \begin{bmatrix} 0 & -1/Z_c & 0 & 1 + r^2 & 0 & 0 \end{bmatrix} \\ \mathbf{L}_\mu &= \begin{bmatrix} 0 & 0 & 2r^2/Z_c & 0 & 0 & 0 \end{bmatrix}\end{aligned}$$

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

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

⇒ Control issues

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

Visual task function

With k visual features \mathbf{s} , one constraints m ($= n - N \leq k$) robot dof using the **visual task function** $\mathbf{e}_1(\mathbf{p}(t)) = \mathbf{C} (\mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^*)$ where \mathbf{C} is a $m \times k$ combination matrix of full rank m .

⇒ 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, expressed as a cost function to be minimized under the constraint that \mathbf{e}_1 is satisfied.

Global task function

A task function \mathbf{e} minimizing the objective function h_s under the constraint $\mathbf{e}_1 = 0$ is given by :

$$\begin{aligned}\mathbf{e} &= \mathbf{W}^+ \mathbf{e}_1 + (\mathbb{I}_n - \mathbf{W}^+ \mathbf{W}) \mathbf{g}_s^\top \\ &= \mathbf{W}^+ \mathbf{C} (\mathbf{s} - \mathbf{s}^*) + (\mathbb{I}_n - \mathbf{W}^+ \mathbf{W}) \mathbf{g}_s^\top\end{aligned}$$

where :

- \mathbf{g}_s = gradient of h_s ($\mathbf{g}_s = \frac{\partial h_s}{\partial \mathbf{p}}$)
- \mathbf{W} is a $m \times n$ matrix of full rank m such that

$$\text{Ker } \mathbf{W} = \text{Ker } \mathbf{L}_{\mathbf{s}}$$

$$\Rightarrow (\mathbb{I}_n - \mathbf{W}^+ \mathbf{W}) \mathbf{g}_s^\top \in \text{Ker } \mathbf{L}_{\mathbf{s}}, \forall \mathbf{g}_s^\top$$

- if $m = n$, $\mathbf{W} = \mathbb{I}_n$, $\mathbf{e} = \mathbf{e}_1 = \mathbf{C} (\mathbf{s} - \mathbf{s}^*)$

Control law (similar as before)

Since we have

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

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

$$\mathbf{v}_q = \mathbf{L}_\mathbf{e}^{-1} \left(-\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t} \right)$$

with $\boxed{\mathbf{L}_\mathbf{e} = \mathbf{W}^+ \mathbf{C} \mathbf{L}_\mathbf{s} + (\mathbb{I}_n - \mathbf{W}^+ \mathbf{W}) \frac{\partial \mathbf{g}_s^T}{\partial \mathbf{p}}}$ if \mathbf{C} and \mathbf{W} are constant

Since $\mathbf{L}_\mathbf{e}$ and $\boxed{\frac{\partial \mathbf{e}_1}{\partial t}}$ are not perfectly known, one uses

$$\mathbf{v}_q = \widehat{\mathbf{L}}_\mathbf{e}^{-1} \left(-\lambda \mathbf{e} - \frac{\widehat{\partial \mathbf{e}}}{\partial t} \right)$$

Stability analysis (same as before)

Behavior of the closed-loop system :

$$\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{v}_q + \frac{\partial \mathbf{e}}{\partial t} = -\lambda \widehat{\mathbf{L}_e \mathbf{L}_e^{-1}} \mathbf{e} - \widehat{\mathbf{L}_e \mathbf{L}_e^{-1}} \frac{\partial \mathbf{e}}{\partial t} + \frac{\partial \mathbf{e}}{\partial t}$$

If $\frac{\partial \mathbf{e}}{\partial t} = \widehat{\frac{\partial \mathbf{e}}{\partial t}} = 0$, $\|\mathbf{e}\|$ always decreases (global stability) if

$$\widehat{\mathbf{L}_e \mathbf{L}_e^{-1}} > 0$$

To suppress tracking errors and obtain the desired behavior $\dot{\mathbf{e}} = -\lambda \mathbf{e}$:

$$\widehat{\mathbf{L}_e^{-1}} = \mathbf{L}_e \quad \text{and} \quad \widehat{\frac{\partial \mathbf{e}}{\partial t}} = \frac{\partial \mathbf{e}}{\partial t}$$

In practice (similar as before)

- If $k = m = n$, $\boxed{\mathbf{W} = \mathbf{C} = \mathbb{I}_m}$, $\mathbf{e} = \mathbf{e}_1 = \mathbf{s} - \mathbf{s}^* \Rightarrow \mathbf{L}_e = \mathbf{L}_s, \widehat{\mathbf{L}}_e = \widehat{\mathbf{L}}_s$

$$\mathbf{v}_q = -\lambda \widehat{\mathbf{L}}_s^{-1} (\mathbf{s} - \mathbf{s}^*) - \widehat{\mathbf{L}}_s^{-1} \frac{\partial \mathbf{s}}{\partial t} \quad \text{stable if } \mathbf{L}_s \widehat{\mathbf{L}}_s^{-1} > 0$$

- If $k > m$, $\boxed{\mathbf{C} = \mathbf{W} \widehat{\mathbf{L}}_s^+|_{\mathbf{s}=\mathbf{s}^*}}$, $\widehat{\mathbf{L}}_e = \mathbb{I}_n$,

$$\mathbf{v}_q = -\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t} \quad \text{stable if } \boxed{\mathbf{W} \widehat{\mathbf{L}}_s^+|_{\mathbf{s}=\mathbf{s}^*} \mathbf{L}_s \mathbf{W}^+ > 0} \quad (\text{only around } \mathbf{s}^*)$$

If translational dof are controlled and 2D visual features are used,
an estimation $\widehat{\mathbf{P}}$ or $\widehat{\mathbf{P}}^*$ is necessary to compute $\widehat{\mathbf{L}}_s$

Example of secondary tasks

- **Trajectory following:**

for instance, constant velocity along z camera axis

$$h_s = \frac{1}{2} (z(t) - z(0) - v_z t)^2$$

$$\Rightarrow g_s^\top = \begin{bmatrix} 0 \\ 0 \\ z(t) - z(0) - v_z t \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \frac{\partial g_s^\top}{\partial t} = \begin{bmatrix} 0 \\ 0 \\ -v_z \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\text{Control law: } \mathbf{v}_q = -\lambda \mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}} = -\lambda \mathbf{e} + (\mathbb{I}_n - \mathbf{W}^+ \mathbf{W}) v_z$$

Target tracking (same as before)

PI Controller

Integral term classical in Automatic Control to reduce tracking errors

If \mathbf{I}_k is the estimation of $\boxed{\frac{\partial \mathbf{e}_1}{\partial t}}$ at iteration k , we have :

$$\begin{aligned}\mathbf{I}_{k+1} &= I_k + \mu \mathbf{e}_{1k} \text{ with } \mathbf{I}_0 = 0 \\ &= \mu \sum_{j=0}^k \mathbf{e}_{1j}\end{aligned}$$

Efficient to track a target at constant velocity :

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

Target tracking by estimating the target velocity

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

$$\widehat{\frac{\partial \mathbf{e}_1}{\partial t}} = \widehat{\dot{\mathbf{e}}_1} - \widehat{\mathbf{L}_{\mathbf{e}_1}} \mathbf{v}_c$$

with

$$\begin{cases} \widehat{\mathbf{L}_{\mathbf{e}_1}} = \mathbf{C} \widehat{\mathbf{L}_{\mathbf{s}}} \\ \widehat{\dot{\mathbf{e}}_{1k}} = \frac{\mathbf{e}_{1k} - \mathbf{e}_{1k-1}}{\Delta t} = \mathbf{C} \frac{\mathbf{s}_k - \mathbf{s}_{k-1}}{\Delta t} \end{cases}$$

A Kalman filter may then be used.

- 1) Modeling issues
 - 2) Control issues
- ⇒ 3) Applications

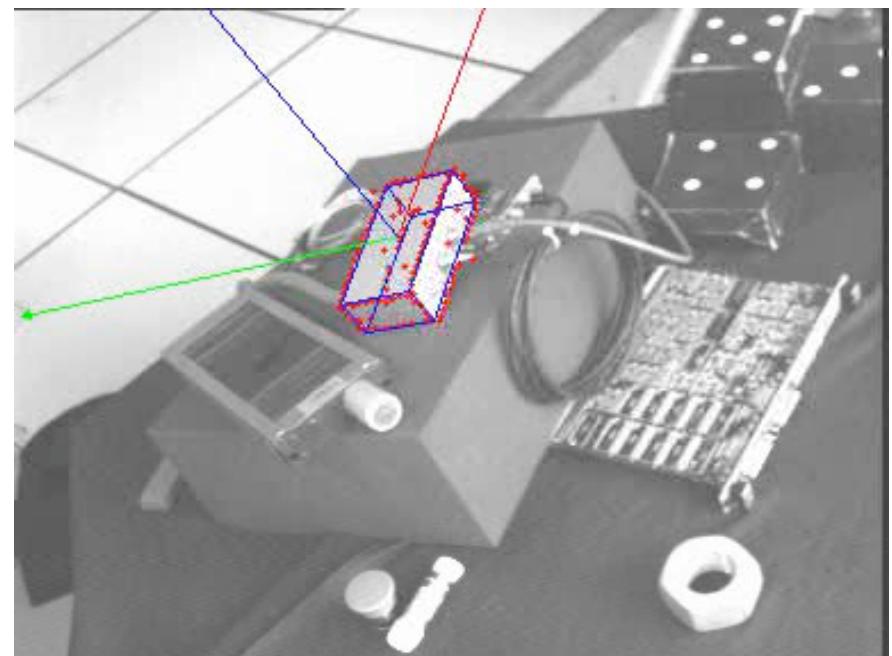
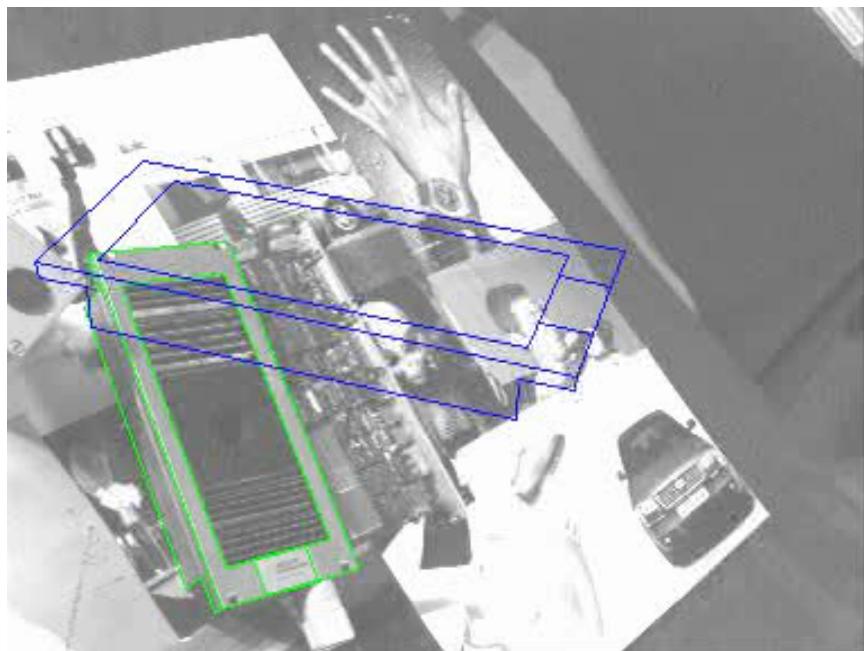
Target tracking by gaze (pan-tilt) control

- Direct link between visual features (cog) and pan-tilt
- No 3D data at all required in the control scheme
- Trying to reduce the tracking errors due to the unknown target motions
- Image processing: image motion-based estimation



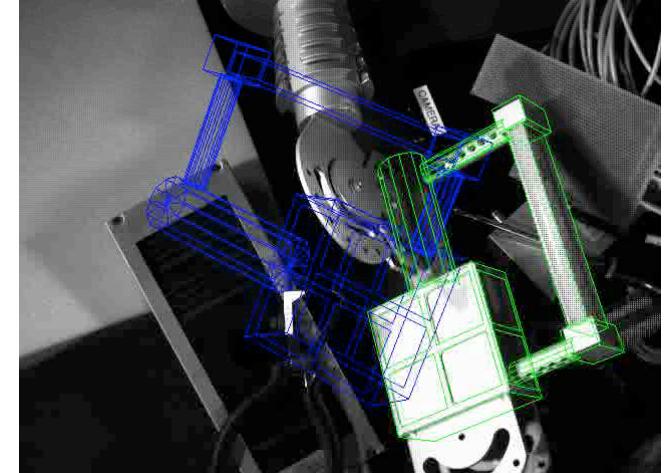
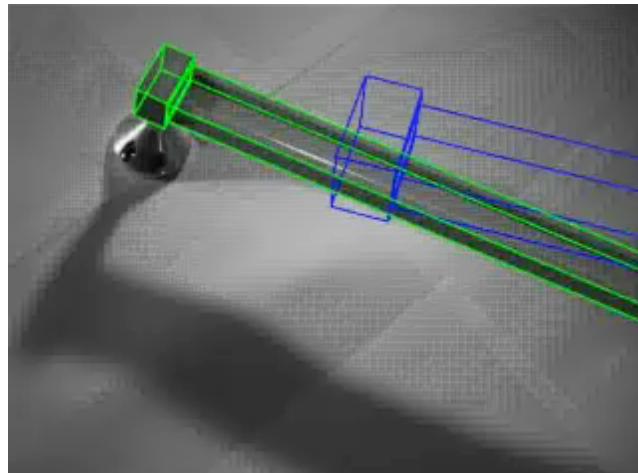
6 dof positioning and target tracking task

Using a robust 3D model-based visual tracking

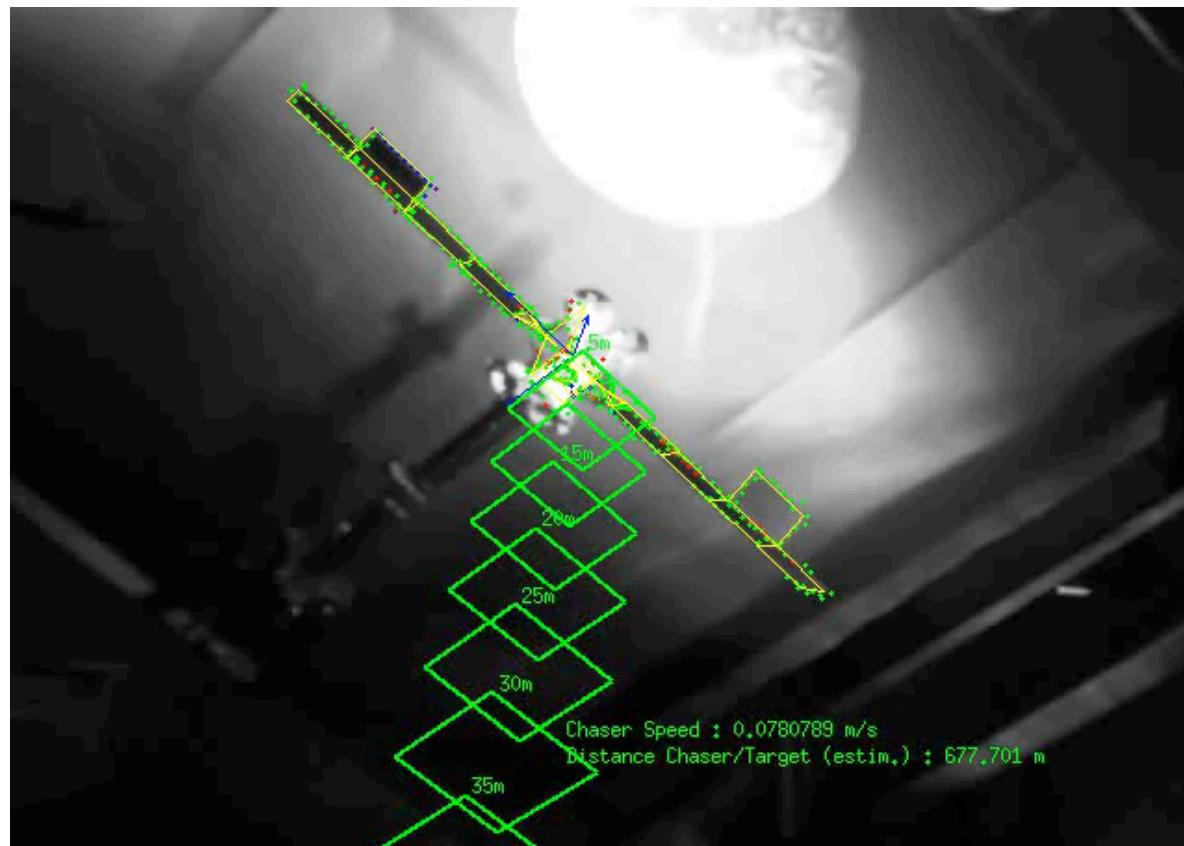


Manipulation in space environment

ESA Vimanco project: using Eurobot on the ISS



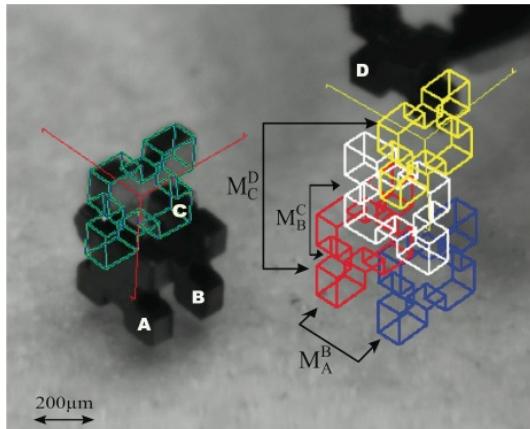
Satellite rendezvous



Micromanipulation



Assembly of MEMS compounds



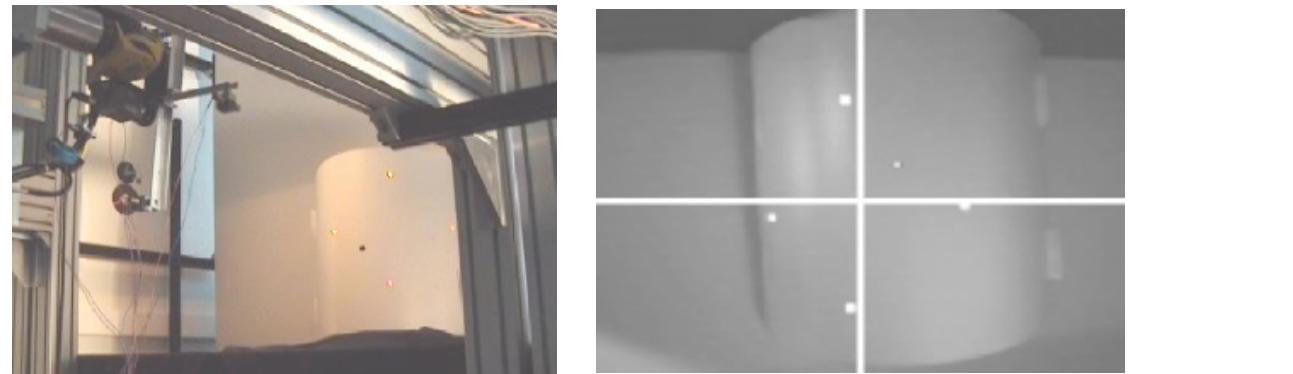
Size 400 $\mu\text{m} \times 400\mu\text{m}$

IBVS with structured lights

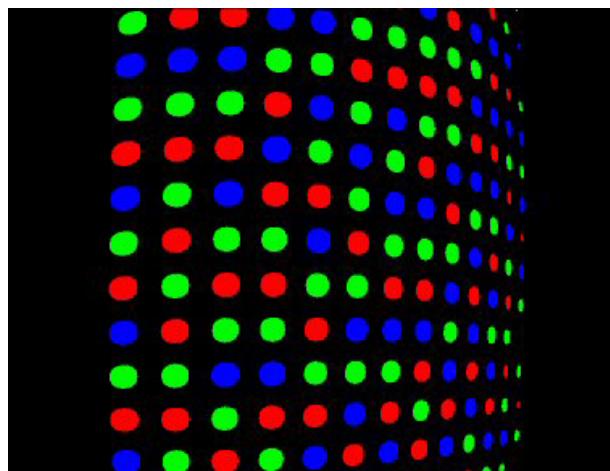
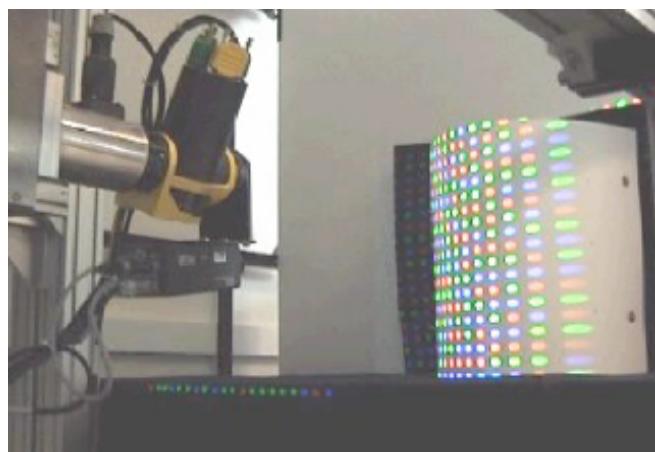


In collaboration with Girona University, Spain, and Cemagref, Rennes

- Eye-in-hand system



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



Desired image

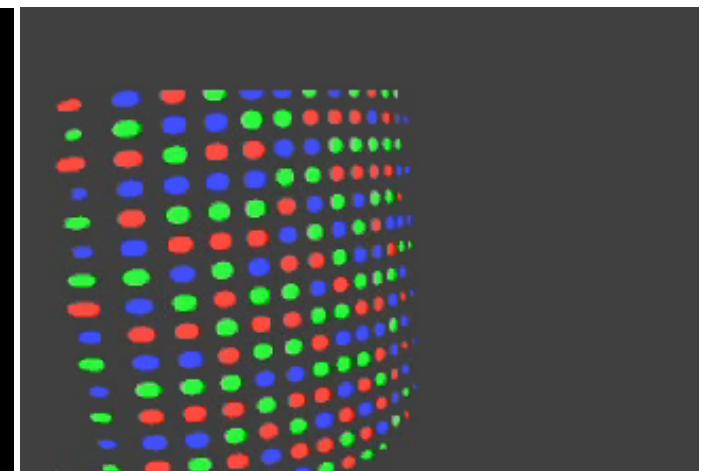
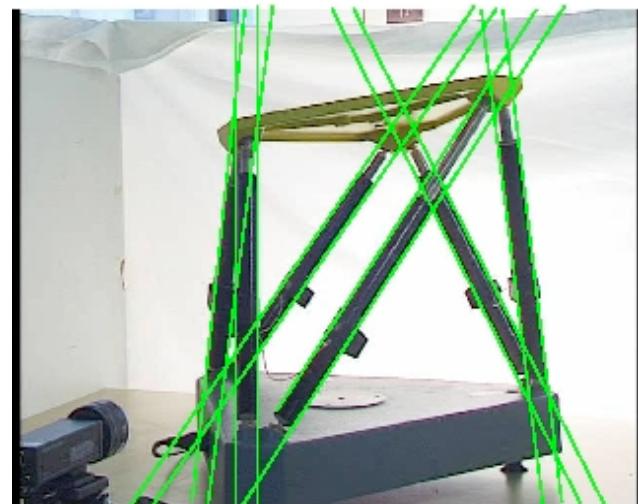
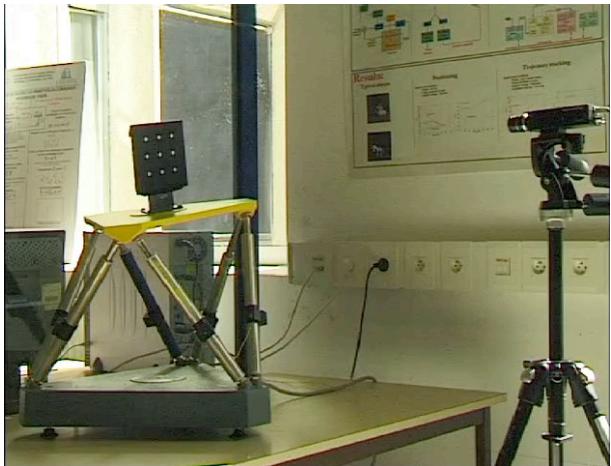


Image sequence

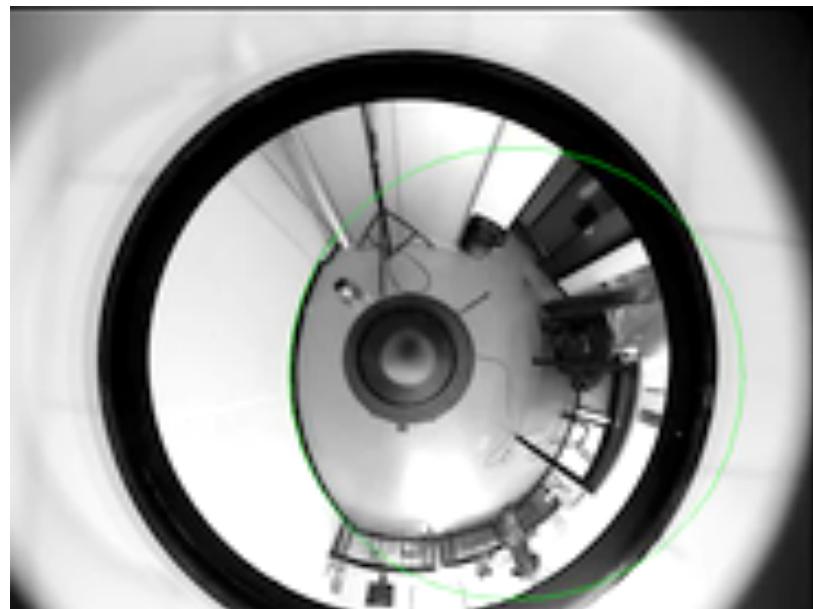
Parallel robots



Navigation of mobile robot in indoor environment



- Use of straight line in omnidirectional images



Car following

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE
centre de recherche **SOPHIA ANTIPOLIS - MÉDITERRANÉE**

RINRIA



INRIA Sophia-Antipolis, France - Copyright (C) 2005.
INRIA Sophia-Antipolis, France - Copyright (C) 2005.

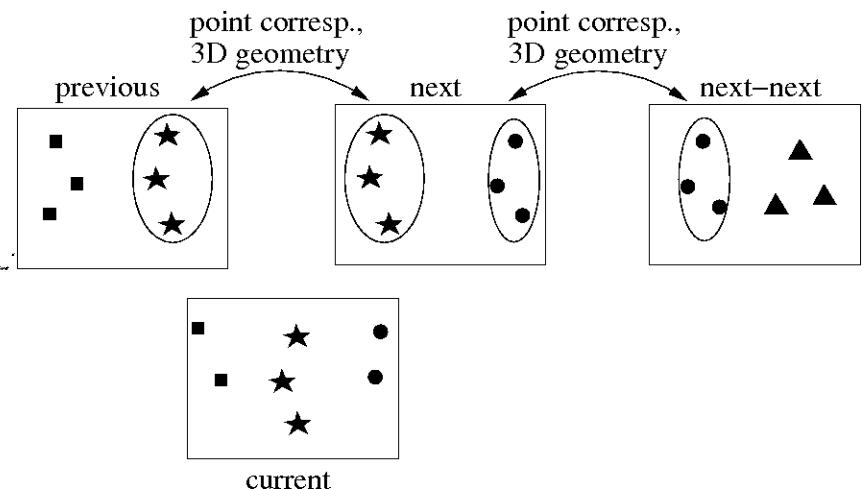
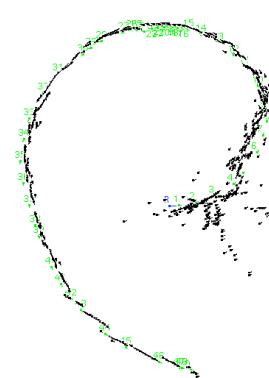
Navigation in a urban environment

Classical approach:

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

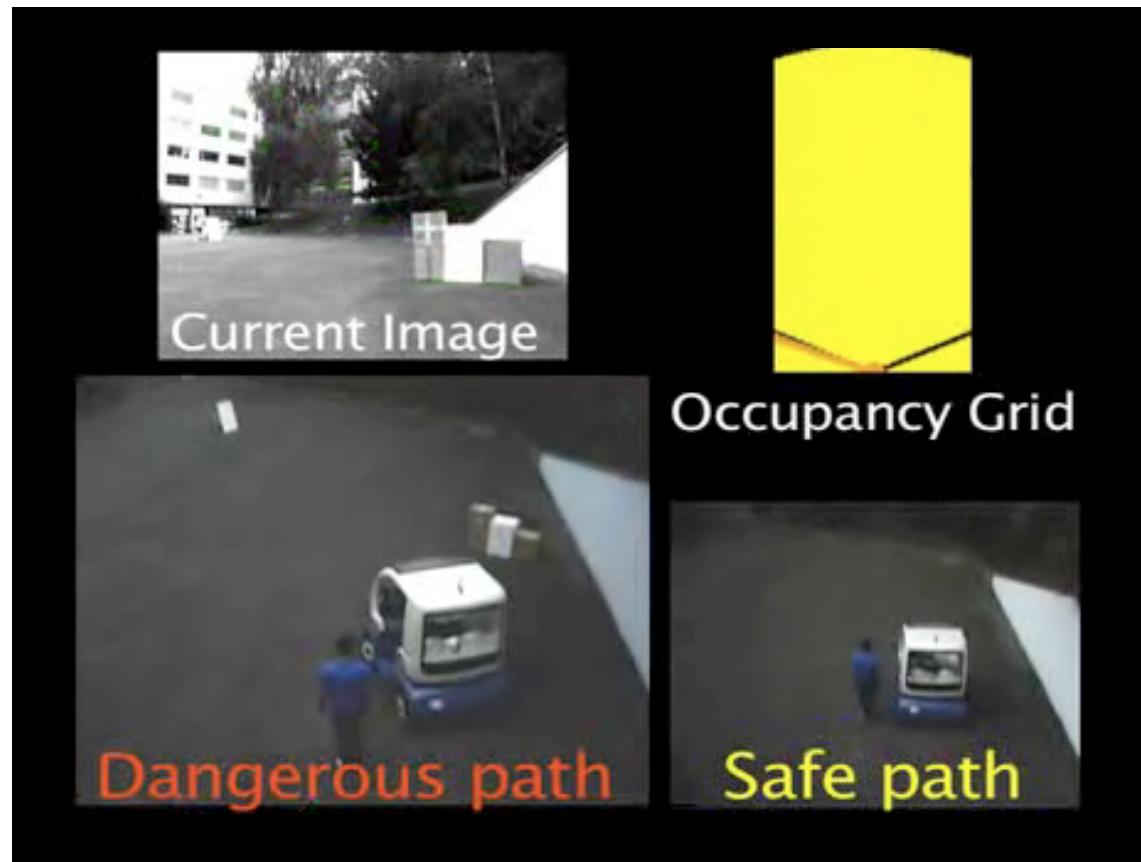
Approach developed: Accurate localization and mapping not mandatory

- 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 and obstacles avoidance

- Using a laser-range finder for obstacles localization
- Using a pan-tilt camera to observe the visual path while avoiding obstacles



IBVS for X4-flyer positioning

In collaboration with CEA List, I3S and ANU





VTOL aircraft landing and terrain following

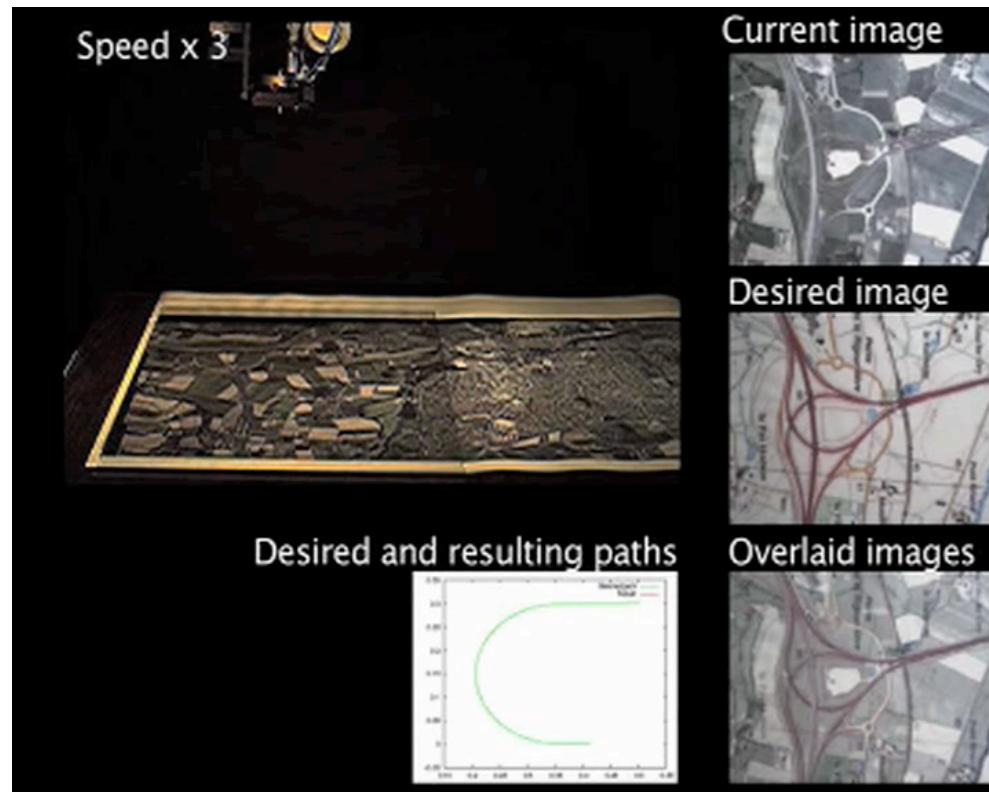


Optical flow used to estimate the ratio translational velocity/depth



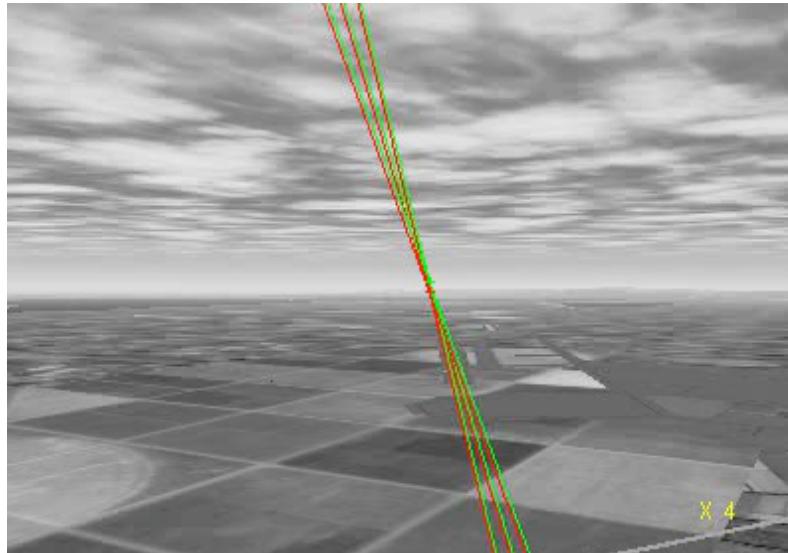
Multi-modal visual servoing

- ❑ Task specified on one image modality (a map for instance)
- ❑ Task realized with another image modality
using the mutual information between the two modalities



IBVS for fixed wing aircraft landing

In collaboration with Dassault Aviation (Paris)

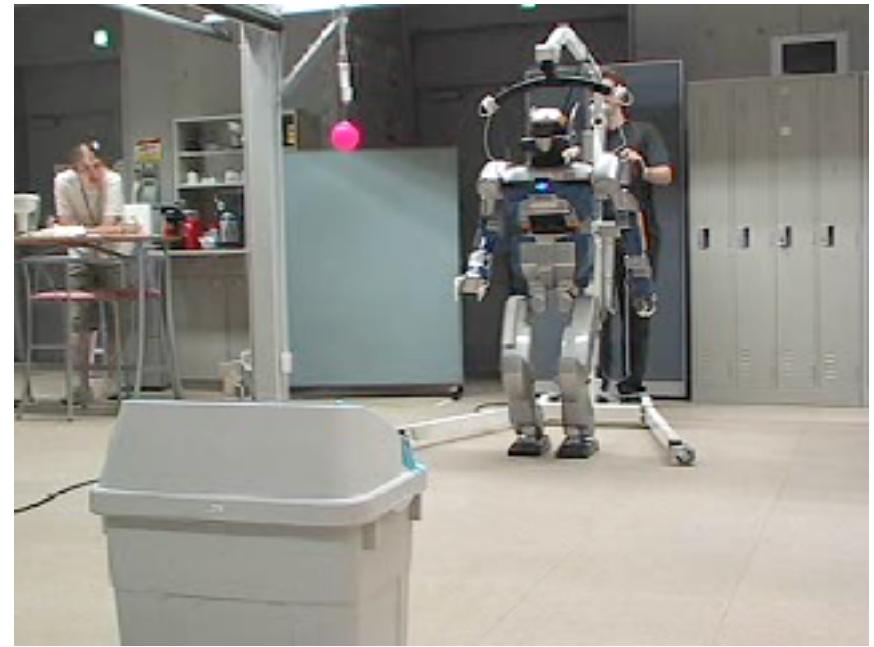


Task sequencing (catching a ball)

- ❑ Idea: to give as much freedom as possible to take constraints (joint limits, occlusions, obstacles) into account
 - Scheme more reactive than reactive path planning
 - Scheme more versatile than classical visual servoing
- ❑ Visual elementary task managed by a stack
 - Remove the good task for ensuring the constraints
 - Put the task back when possible

Application to a humanoid robot

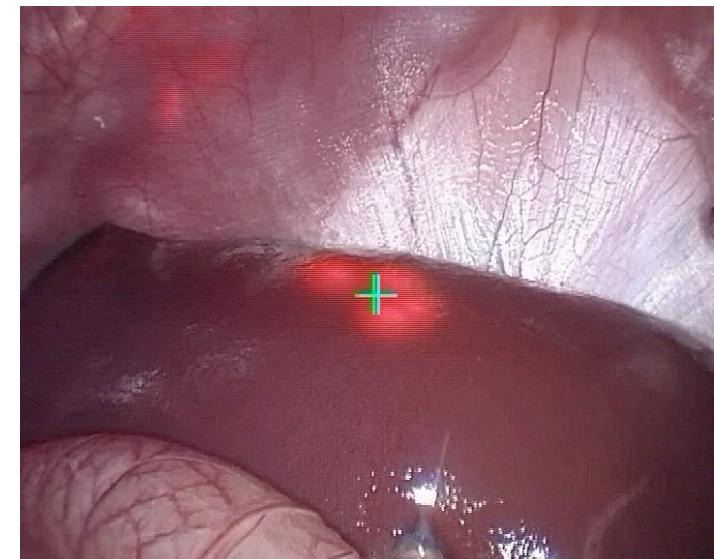
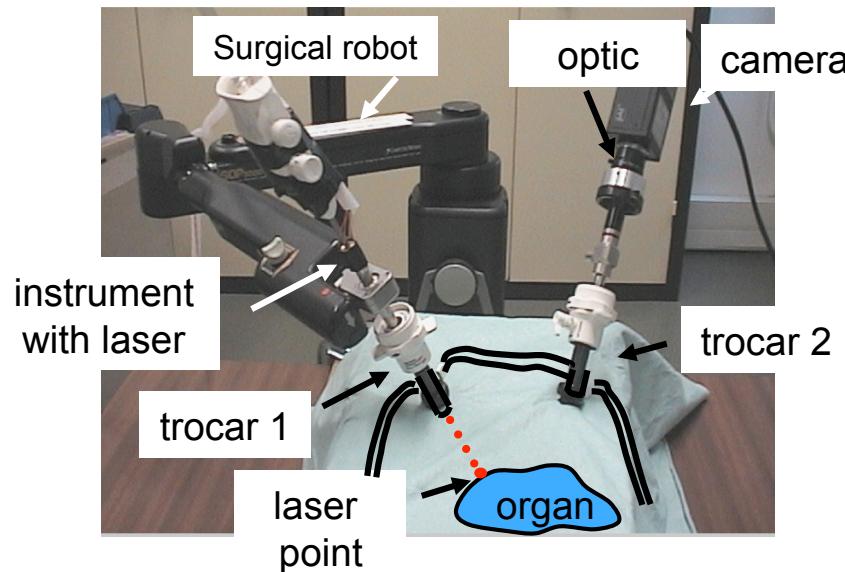
(in collaboration with AIST/CNRS, Tsukuba)



Medical robotics

Visual servoing for laparoscopic surgery

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

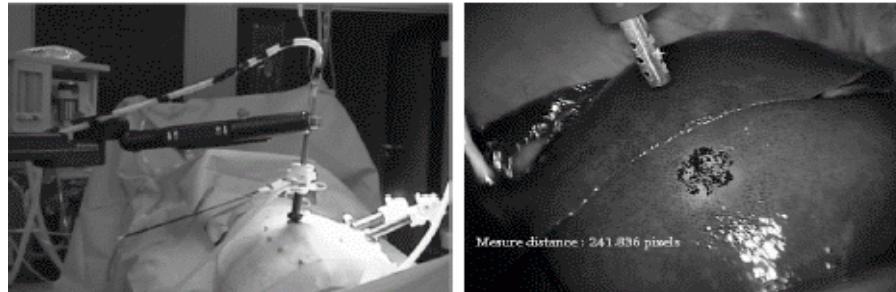


Autonomous tasks:

- 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.



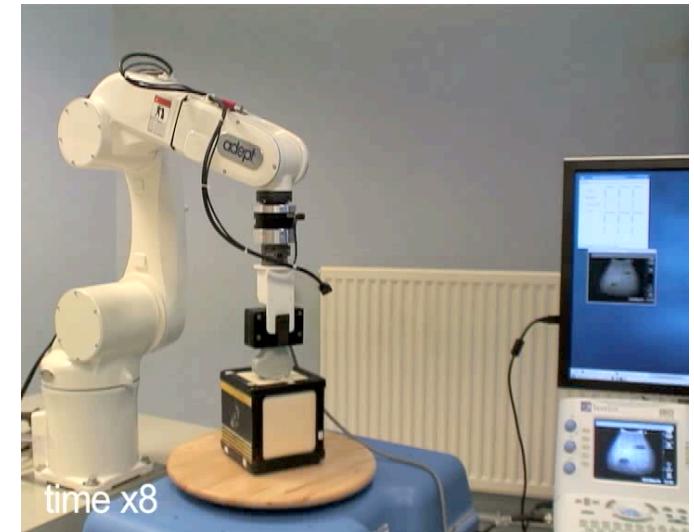
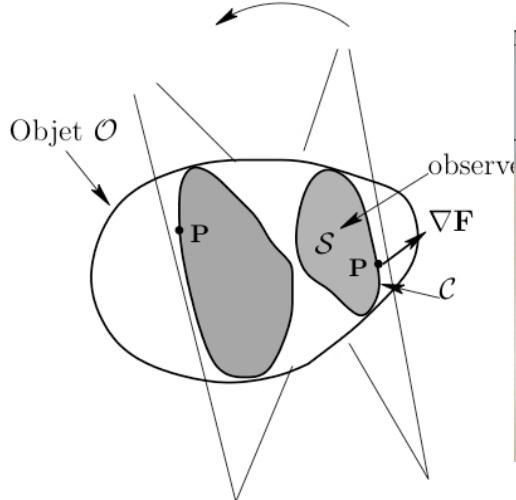
- ❑ Beating Heart Tracking :
 - Combination of cardiac and respiratory motions.
 - Use of an adaptive filtering to estimate separate contributions of breathing and heartbeats.
 - Predictive control scheme which anticipates the perturbation due to an adaptive disturbance predictor.

Visual servoing with 2D ultrasound images

Considering soft tissues and 2D moments as visual features

Modeling revisited:

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



Potential clinical applications in medical robotics:

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

Computer animation

- ❑ Control of camera motion in virtual environments
 - Task specification in the image
 - Obstacles avoidance thanks to redundancy framework



Museum visit

Cinematographic
constraints (dialog)



- ❑ Control of synthetic humanoids
 - Gaze control of a humanoid
 - Locomotion control
- ❑ Applications: video games, films,...

