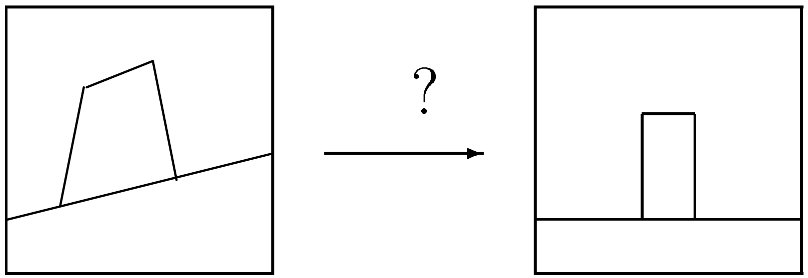
Visual servoing

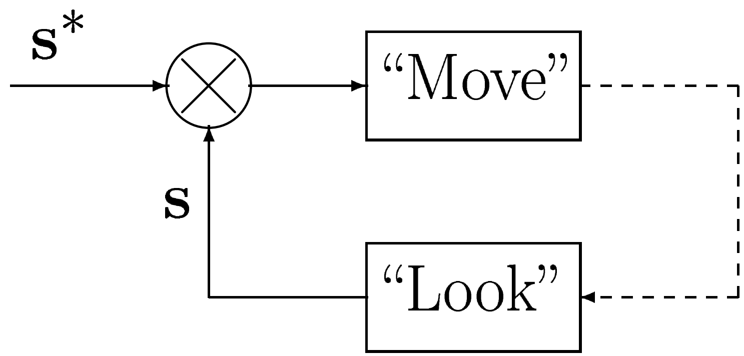
François Chaumette

IRISA / INRIA Rennes, France

http://www.irisa.fr/vista

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 $\mathbf{s}$ to control n dof $(k \geq n, n \leq 6)$

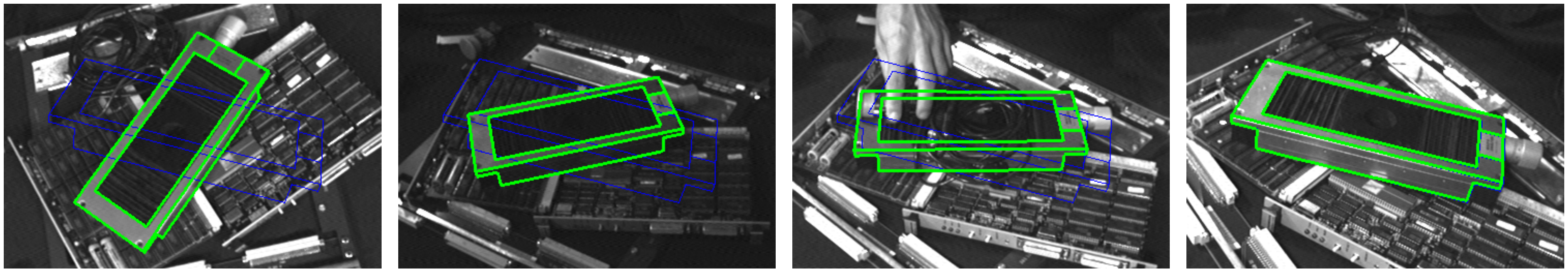
Determine the goal $\mathbf{s}^*$

Regulate the error $(\mathbf{s} - \mathbf{s}^*)$

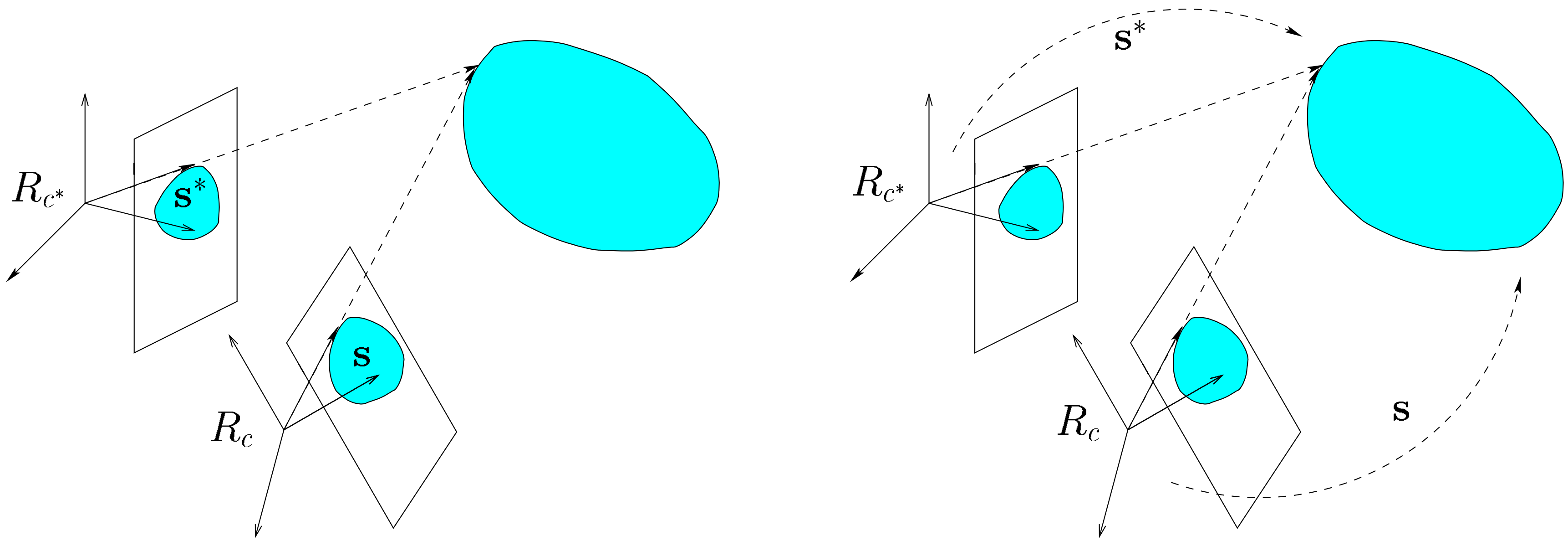
Image processing (matching and tracking near video rate)

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: $\mathbf{s} = \mathbf{s}(\mathbf{r}(t)) \quad \Rightarrow \quad \dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}^T \, \mathcal{T}$ where:

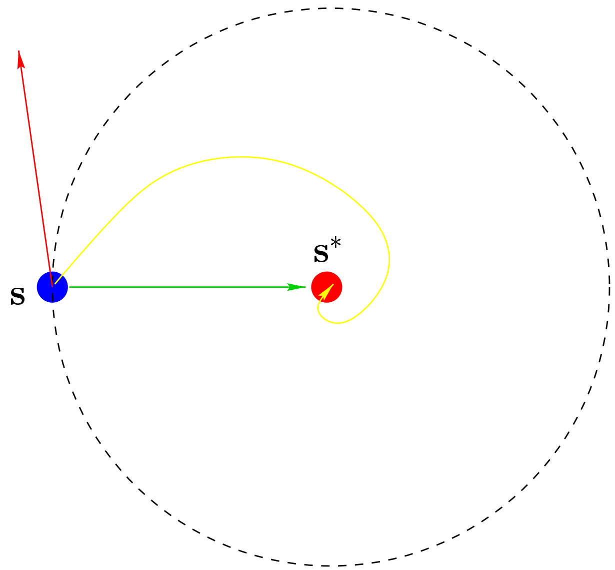
- $\mathbf{L}_{\mathbf{s}}^T$ = interaction matrix (jacobian)
- $\mathcal{T} = (V, \omega)$ = kinematic screw (translational and rotational velocity)

Principle of the control law

$$\mathcal{T}_c = -\lambda \, \widehat{\mathbf{L}_{\mathbf{s}}^T}^+ (\mathbf{s} - \mathbf{s}^*)$$

Closed-loop system: $\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}^T \, \mathcal{T}_c = -\lambda \, \mathbf{L}_{\mathbf{s}}^T \widehat{\mathbf{L}_{\mathbf{s}}^T}^+ (\mathbf{s} - \mathbf{s}^*)$

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



$\Rightarrow$ 1) Modeling issues

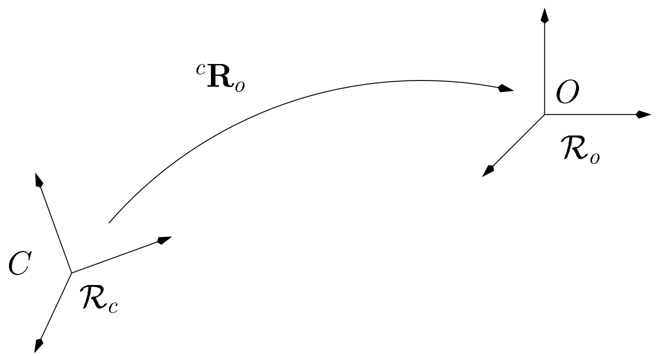
2) Classification of the vision-based tasks

3) Control issues

4) Use of other exteroceptive sensors

Change of frames

pose $\mathbf{r} \in SE_3, \mathbf{r} \sim \mathbf{R}$



$$\mathbf{M}_c = {}^c\mathbf{A}_o\,\mathbf{M}_o + {}^c\mathbf{T}_o$$

$\mathbf{M}_c$: coordinates of M in R_c

$\mathbf{M}_o$: coordinates of M in R_o

${}^c\mathbf{T}_o$: position of O in R_c

${}^c\mathbf{A}_o$: rotation matrix between R_c and R_o

$$\mathbf{A} = \cos\theta\;\mathbb{I}_3 + \sin\theta\;\tilde{\mathbf{U}} + (1-\cos\theta)\;\mathbf{U}\mathbf{U}^T$$

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

$\tilde{\mathbf{U}}$: skew symetric matrix related to $\mathbf{U}$: $\tilde{\mathbf{U}} = \begin{pmatrix} 0 & -U_z & U_y \\ U_z & 0 & -U_x \\ -U_y & U_x & 0 \end{pmatrix}$

Kinematic screw

$\mathcal{T} = (\mathbf{V}^T, \omega^T)^T$: kinematic screw between the camera and the scene expressed at C in R_c

ω : rotational velocity : $\tilde{\omega} = \dot{\mathbf{A}}\mathbf{A}^T = \mathbf{A}^T\dot{\mathbf{A}}$

$\mathbf{V}$: translational velocity at C : $\mathbf{V}(N) = \mathbf{V}(M) + \tilde{\omega}\,\mathbf{MN}$

To express $\mathcal{T}$ at O in R_o : ${}^o\mathcal{T} = {}^o\mathbb{T}_c\mathcal{T}$ with ${}^o\mathbb{T}_c = \begin{pmatrix} {}^o\mathbf{A}_c & {}^o\tilde{\mathbf{P}}_c{}^o\mathbf{A}_c \\ \mathbf{0}_3 & {}^o\mathbf{A}_c \end{pmatrix}$

We can decompose $\mathcal{T}$ as $\mathcal{T} = \mathcal{T}_c - \mathcal{T}_o$

where $\mathcal{T}_c$: camera kinematic screw, expressed at C in R_c

$\mathcal{T}_o$: object kinematic screw, expressed at C in R_c

The interaction matrix

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

$$\mathbf{s} = \mathbf{s}(\mathbf{r}(t))$$

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

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \, \dot{\mathbf{r}} = \mathbf{L}_{\mathbf{s}}^T \, \mathcal{T}$$

where $\mathbf{L}_{\mathbf{s}}^T$ is the **interaction matrix** related to $\mathbf{s}$ (jacobian)

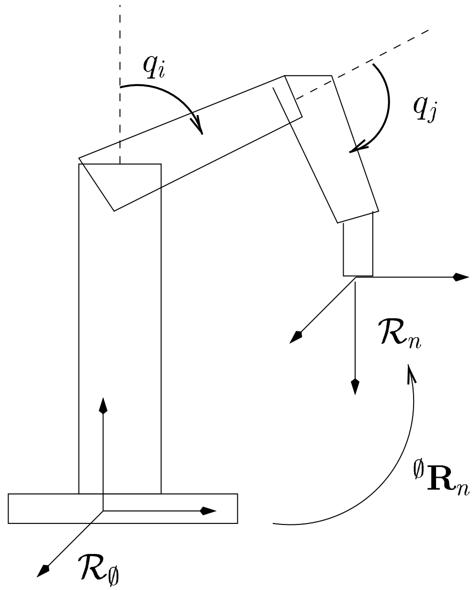
Using $\mathcal{T}_c$ and $\mathcal{T}_o$, we obtain :

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}^T \, (\mathcal{T}_c - \mathcal{T}_o)$$

Robot Jacobian

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

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



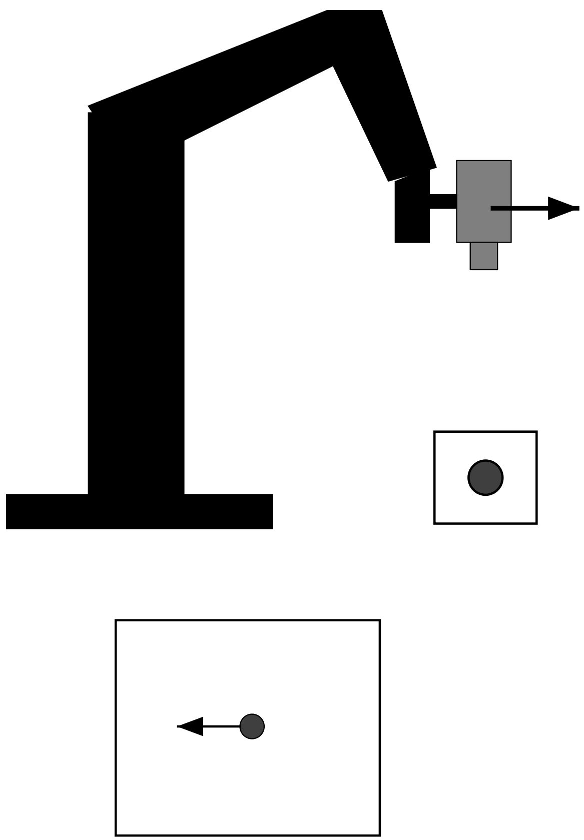
End-effector kinematic screw given by :

$$\mathcal{T}_n = {}^n\mathbf{J}_n(\mathbf{q})\,\dot{\mathbf{q}} \text{ where } {}^n\mathbf{J}_n(\mathbf{q}) = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \text{ is the robot jacobian}$$

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

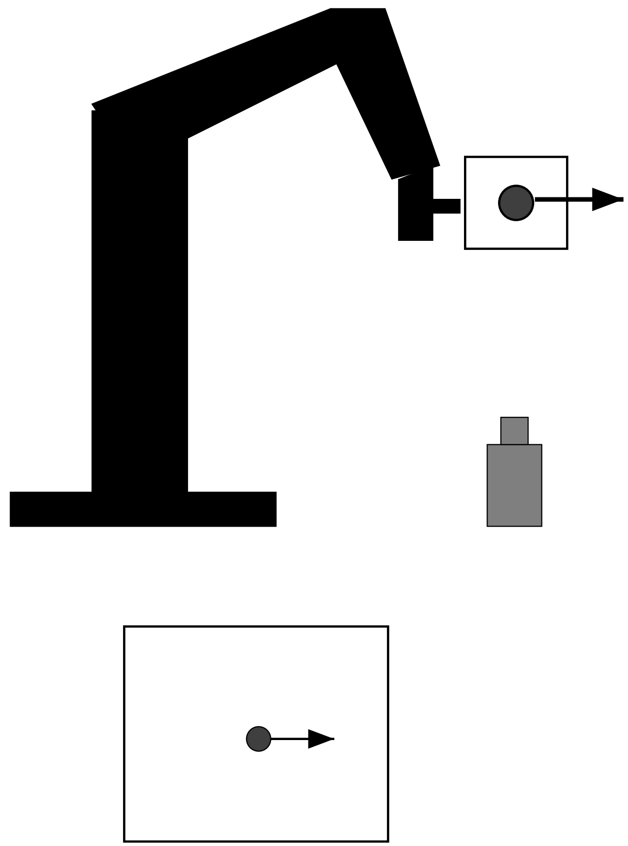
Robot singularities $= \{\mathbf{q}_s, \det\left({}^n\mathbf{J}_n(\mathbf{q}_s)\right) = 0\}$

Eye-in-Hand system



$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}^{T}\, {}^{c}\mathbb{T}_{n}\, {}^{n}\mathbf{J}_{n}(\mathbf{q})\ \dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t}$$

Eye-to-Hand system

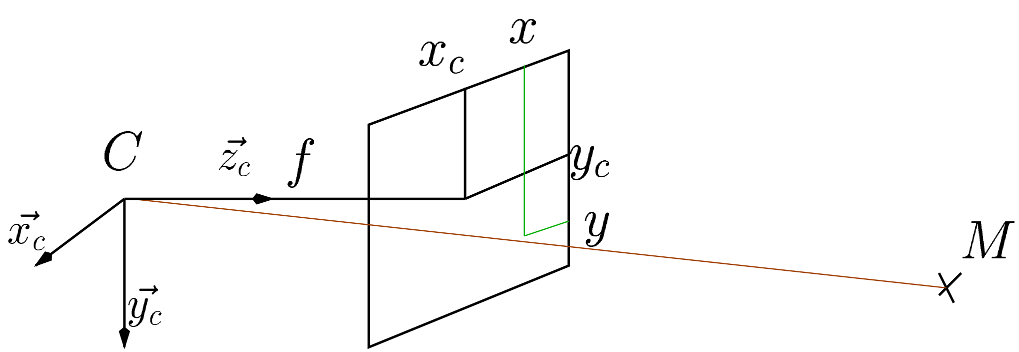


$$\begin{aligned}\dot{\mathbf{s}} &= -\mathbf{L}_{\mathbf{s}}^{T}\, {}^{c}\mathbb{T}_{n}\, {}^{n}\mathbf{J}_{n}(\mathbf{q})\ \dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \\ &= -\mathbf{L}_{\mathbf{s}}^{T}\, {}^{c}\mathbb{T}_{\emptyset}\, {}^{\emptyset}\mathbb{T}_{n}\, {}^{n}\mathbf{J}_{n}(\mathbf{q})\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

Camera modeling

Definitions

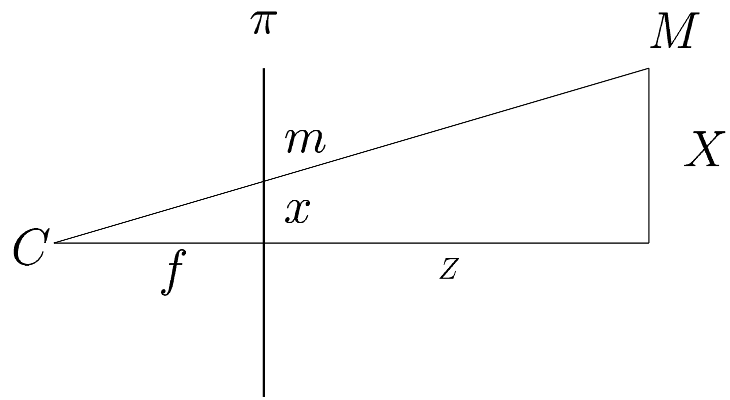
- Origin of camera frame $\mathcal{R}_c$ = projection center C.
- $(C, \vec{x_c}) \parallel$ image rows, $(C, \vec{y_c}) \parallel$ image columns
- Principal point (X_c, Y_c) = intersection between axis $(C, \vec{z_c})$ and image plane π.
- Focal length $f = d(C, \pi)$.



Perspective projection

Let M with coordinates (X, Y, Z) in R_c and m with coordinates $(x, y, 1)$ the perspective projection of M. We have:

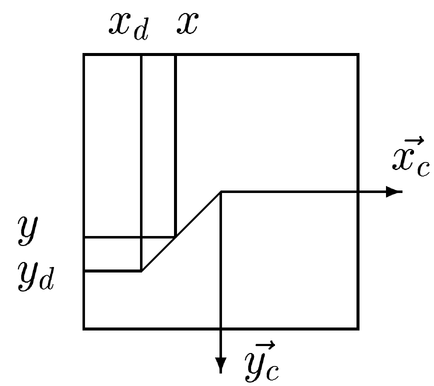
$$x = f\,X\,/\,Z\;,\;\; y = f\,Y\,/\,Z$$



Radial distorsion

Let K be the coefficient of radial distorsion.
The observed position in the image of point M is in fact $m_d\,(x_d, x_d)$ with :

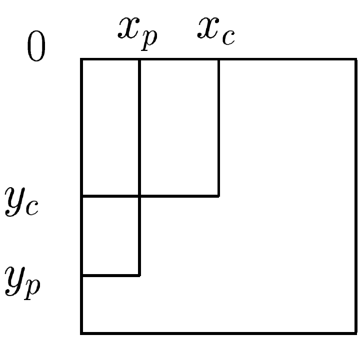
$$x_d = x + Kx(x^2 + y^2)\ ,\ \ y_d = y + Ky(x^2 + y^2)$$



Sampling : from meters to pixels

Let $m_p\,(x_p, y_p)$ the position in pixels of m_d.
We have

$$x_p = x_c + x_d \,/\, l_x \ , \ \ y_p = y_c + y_d \,/\, l_y$$



where l_x and l_y are the pixel length and width.

Complete camera model

$$\begin{cases} x_p = x_c + f_x \frac{X}{X} + K_d f_x \frac{X}{Z}(\frac{X^2}{Z^2} + \frac{Y^2}{Z^2}) \\ \\ y_p = y_c + f_y \frac{Y}{Z} + K_d f_y \frac{Y}{Z}(\frac{X^2}{Z^2} + \frac{Y^2}{Z^2}) \end{cases} \text{where} \begin{cases} f_x = f/l_x \\ f_y = f/l_y \\ K_d = K f^2 \end{cases}$$

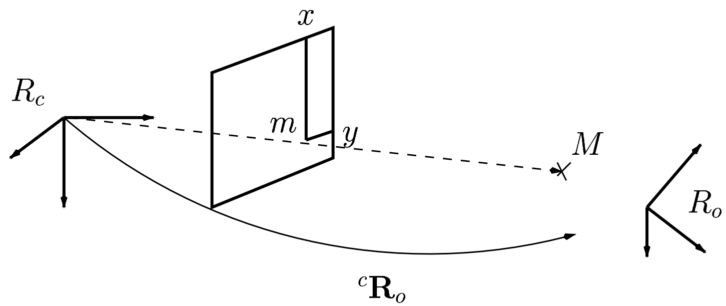
Intrinsic camera parameters : x_c, y_c, f_x, f_y and K_d.

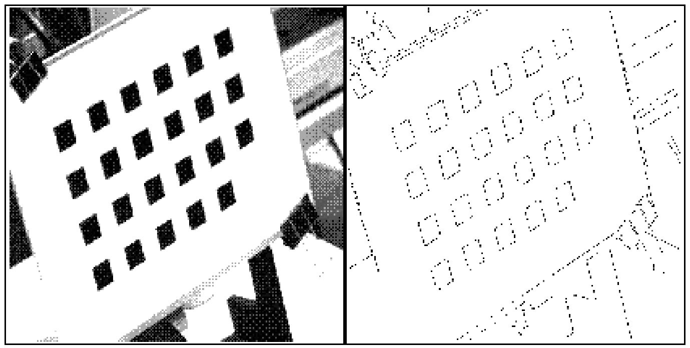
After **calibration** step (when $K_d \sim 0$) :

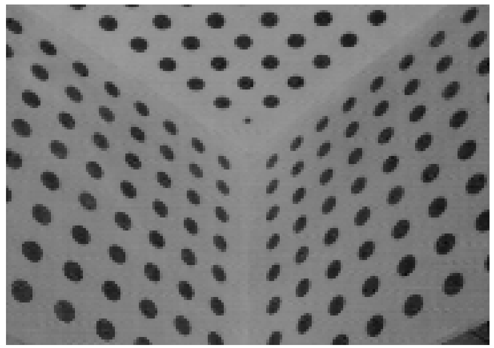
$$\begin{cases} x = (x_p - x_c)/f_x \\ y = (y_p - y_c)/f_y \end{cases} \Rightarrow \begin{cases} x = X/Z \\ y = Y/Z \end{cases}$$

Extrinsic parameters

In the previous equations, coordinates $\mathbf{X}$ of M are unknown since expressed in R_c. For calibration, use of a **grid** where $\mathbf{X}_o$ is known in R_o. Then $\mathbf{X} = {}^c\mathbf{A}_o\mathbf{X}_o + {}^c\mathbf{T}_o$







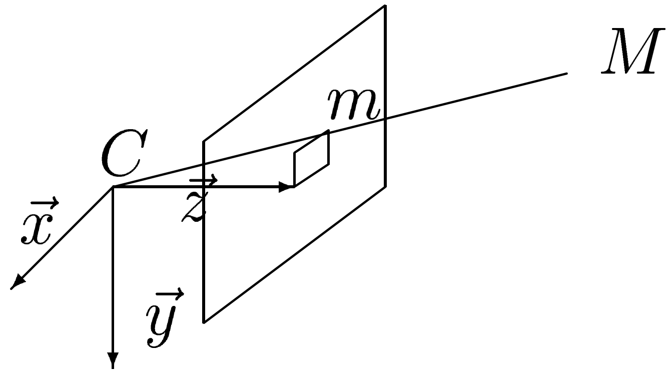
Extrinsic camera parameters : pose $\mathbf{r} \sim {}^c\mathbf{R}_o : {}^c\mathbf{A}_o, {}^c\mathbf{T}_o$

Each grid point provides 2 equations with measurements $\mathbf{x}_{p_i}, \mathbf{X}_{o_i}$.

In visual servoing, **coarse** calibration generally sufficient.

2D visual features: image point coordinates

Perspective projection :



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

$$\Rightarrow \begin{cases} \dot{x} = \begin{pmatrix} 1/Z & 0 & -X/Z^2 \end{pmatrix} . V(M) \\ \dot{y} = \begin{pmatrix} 0 & 1/Z & -Y/Z^2 \end{pmatrix} . V(M) \end{cases}$$

Using a mobile camera and a fixed point : $V(M) = -V(C) - \omega \times M$

We obtain: $\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \mathbf{L}_{xy}^T \, \mathcal{T}$

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

2D visual features: geometrical primitives

P_c : configuration of an *object feature* parametrized by $\mathbf{P}$

$p_c = \pi(P_c)$: configuration of an *image feature* parametrized by $\mathbf{p}$

Noting $\mathbf{P} = \varphi(P_c)$ and $\mathbf{p} = \psi(p_c)$, we get $\mathbf{p} = \nu(\mathbf{P}) = \psi \circ \pi \circ \varphi^{-1}(\mathbf{P})$

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

$$
\begin{array}{ccccccccc}
W \subseteq SE_3 & \longrightarrow & U \subseteq \mathcal{P}_s & & \longrightarrow & & V \subseteq \mathcal{P}_i & & \\
(\mathbf{r}) & \delta & (P_c) & & \pi & & (p_c) & & \\
 & & \Big\downarrow \varphi & & & & \Big\downarrow \psi & & \\
 & & \mathbb{R}^n & & \longrightarrow & & \mathbb{R}^m & \longrightarrow & \mathbb{R}^k \\
 & & (\mathbf{P}) & & \nu = \psi \circ \pi \circ \varphi^{-1} & & (\mathbf{p}) & \sigma & (\mathbf{s})
\end{array}
$$

Finally $\mathbf{s} = \sigma(\mathbf{p}) \Rightarrow \mathbf{L}_{\mathbf{s}}^T = \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial \mathbf{P}} \frac{\partial \mathbf{P}}{\partial \mathbf{r}}$

2D visual features: case of a segment

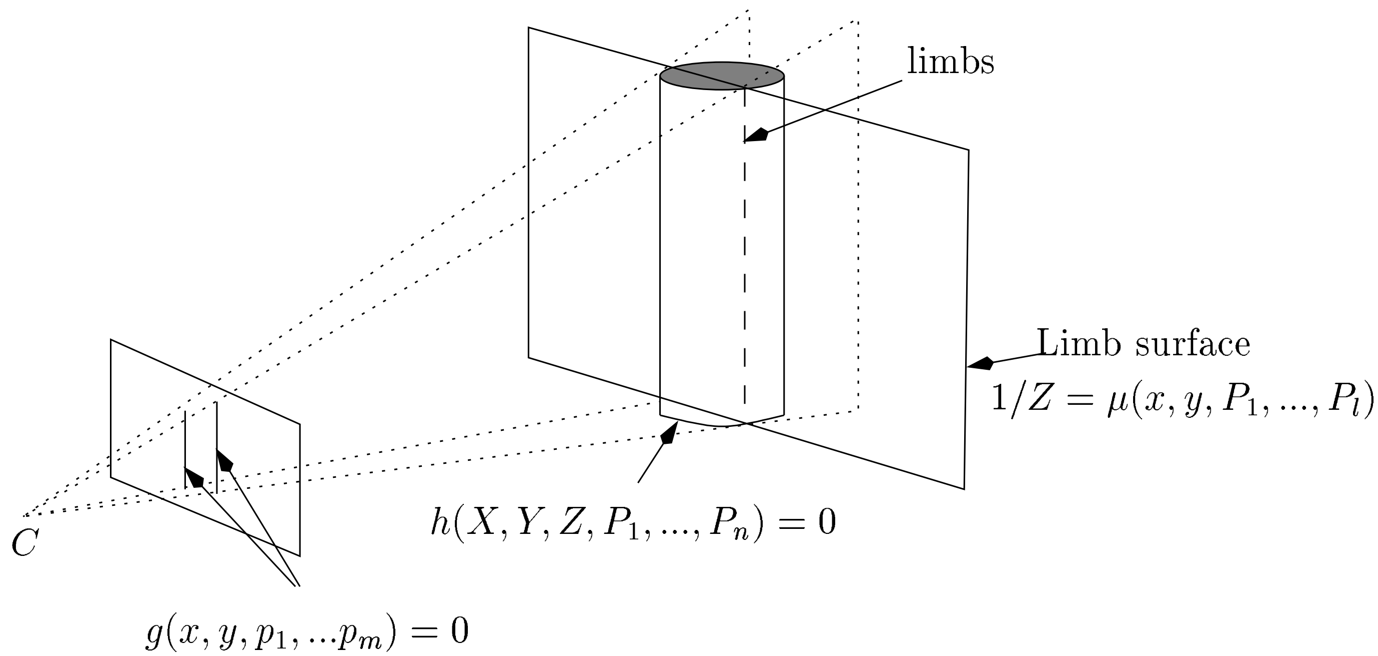
$$\mathbf{s} = \begin{pmatrix} x_c \\ y_c \\ l \\ \alpha \end{pmatrix} = \sigma(x_1, y_1, x_2, y_2) \text{ with } \begin{cases} x_c = (x_1 + x_2)/2 \\ y_c = (y_1 + y_2)/2 \\ l = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ \alpha = \arctan(y_1 - y_2)/(x_1 - x_2) \end{cases}$$

We obtain
$$\begin{pmatrix} \mathbf{L}_{x_c}^T \\ \mathbf{L}_{y_c}^T \\ \mathbf{L}_{l}^T \\ \mathbf{L}_{\alpha}^T \end{pmatrix} = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ \Delta x/l & \Delta y/l & -\Delta x/l & -\Delta y/l \\ -\Delta x/l^2 & \Delta x/l^2 & \Delta y/l^2 & -\Delta x/l^2 \end{pmatrix} \begin{pmatrix} \mathbf{L}_{x_1}^T \\ \mathbf{L}_{y_1}^T \\ \mathbf{L}_{x_2}^T \\ \mathbf{L}_{y_2}^T \end{pmatrix}$$

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

Using $\begin{cases} x_1 = x_c + l\cos\alpha/2, \; y_1 = y_c + l\sin\alpha/2 \\ x_2 = x_c - l\cos\alpha/2, \; y_2 = y_c - l\sin\alpha/2 \end{cases}$, we get $\mathbf{L}_{\mathbf{s}}^T(\mathbf{s}, Z_1, Z_2)$.

Modeling a geometrical primitive



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

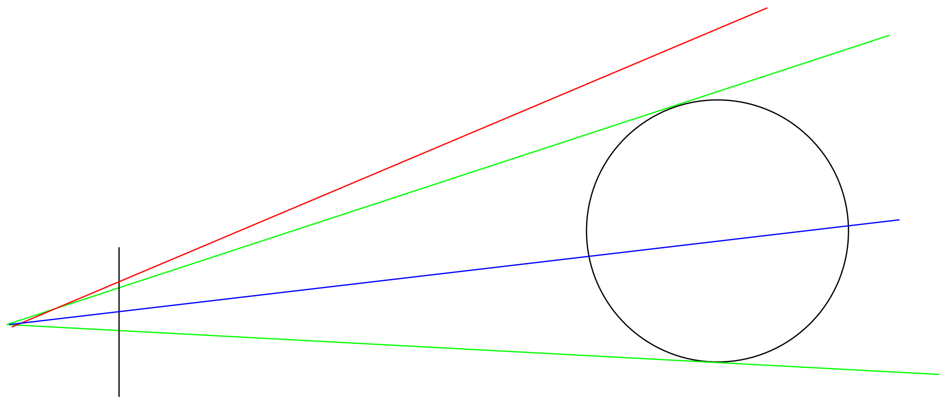
2D primitive : $g(\mathbf{x}, \mathbf{p}) = 0$

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

2D visual features : case of the sphere

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

$x = X/Z \ , \ y = Y/Z \Rightarrow K\frac{1}{Z^2} - 2(X_0 x + Y_0 y + Z_0)\frac{1}{Z} + x^2 + y^2 + 1 = 0$



$\Delta = 0 \Rightarrow \frac{1}{Z} = \mu(\mathbf{x}, \mathbf{P}) = \frac{X_0}{K}x + \frac{Y_0}{K}y + \frac{Z_0}{K}$

$\Delta = 0 \Leftrightarrow (X_0 x + Y_0 y + Z_0)^2 - K(x^2 + y^2 + 1) = 0$

$$\Leftrightarrow g(\mathbf{x}, \mathbf{p}) = x^2 + a_1 y^2 + 2a_2 xy + 2a_3 x + 2a_4 y + a_5 = 0$$

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

Direct computation of the interaction matrix

$$\mathbf{L}_{\mathbf{p}}^T = \frac{\partial \mathbf{p}}{\partial \mathbf{P}} \frac{\partial \mathbf{P}}{\partial \mathbf{r}}$$

$$\left\{ \begin{array}{l} a_1 = (R^2 - X_0^2 - Z_0^2)/(R^2 - Y_0^2 - Z_0^2) \\ \\ a_2 = X_0 Y_0/(R^2 - Y_0^2 - Z_0^2) \\ \ldots \end{array} \right. \Rightarrow \frac{\partial \mathbf{p}}{\partial \mathbf{P}} : \left\{ \begin{array}{l} \dot{a}_1 = (-2X_0\dot{X}_0 - 2Z_0\dot{Z}_0)/(R^2 - Y_0^2 - Y_0^2) - \ldots \\ \dot{a}_2 = \ldots \end{array} \right.$$

$$\frac{\partial \mathbf{P}}{\partial \mathbf{r}} : \left\{ \begin{array}{l} \dot{x}_0 = -V_x - z_0\Omega_y + y_0\Omega_z = \begin{pmatrix} -1 & 0 & 0 & 0 & -Z_0 & Y_0 \end{pmatrix} \mathcal{T} \\ \dot{y}_0 = -V_y - x_0\Omega_z + z_0\Omega_x = \begin{pmatrix} 0 & -1 & 0 & Z_0 & 0 & -X_0 \end{pmatrix} \mathcal{T} \\ \dot{z}_0 = -V_z - y_0\Omega_x + x_0\Omega_y = \begin{pmatrix} 0 & 0 & -1 & -Y_0 & X_0 & 0 \end{pmatrix} \mathcal{T} \end{array} \right.$$

We always have rank $\mathbf{L}_{\mathbf{p}}^T = 3$

Results in $\mathbf{L}_{\mathbf{p}}^T$ are function of 3D data $\mathbf{P} = (X_0, Y_0, Z_0, R)$

Other method

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

We have $\dot{\mathbf{x}} = \mathbf{L}_{xy}^T(\mathbf{x}, 1/Z)\ \mathcal{T} = \mathbf{L}_{xy}^T(\mathbf{x},\mathbf{P})\ \mathcal{T}$

$$\Rightarrow \frac{\partial g}{\partial \mathbf{p}}(\mathbf{x},\mathbf{p})\ \dot{\mathbf{p}} = -\frac{\partial g}{\partial \mathbf{x}}(\mathbf{x},\mathbf{p})\ \mathbf{L}_{xy}^T(\mathbf{x},\mathbf{P})\ \mathcal{T}\ ,\ \forall x \in p_c$$

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

$$\mathbf{L}_{\mathbf{p}}^T(\mathbf{p},\mathbf{P}) = \begin{pmatrix} \alpha_1(\mathbf{p}) \\ \vdots \\ \alpha_m(\mathbf{p}) \end{pmatrix}^{-1} \begin{pmatrix} \beta_1(\mathbf{p},\mathbf{P}) \\ \vdots \\ \beta_m(\mathbf{p},\mathbf{P}) \end{pmatrix} \text{ with } \begin{cases} \alpha_i(\mathbf{p}) = \frac{\partial g}{\partial \mathbf{p}}(\mathbf{x}_i,\mathbf{p}),\ i = 1 \text{ to } m \\ \beta_i(\mathbf{p},\mathbf{P}) = -\frac{\partial g}{\partial \mathbf{x}}(\mathbf{x}_i,\mathbf{p})\mathbf{L}_{xy}^T(\mathbf{x}_i,\mathbf{P}), \\ \qquad i = 1 \text{ to } m \end{cases}$$

2D visual features : case of straight lines

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

We obtain :

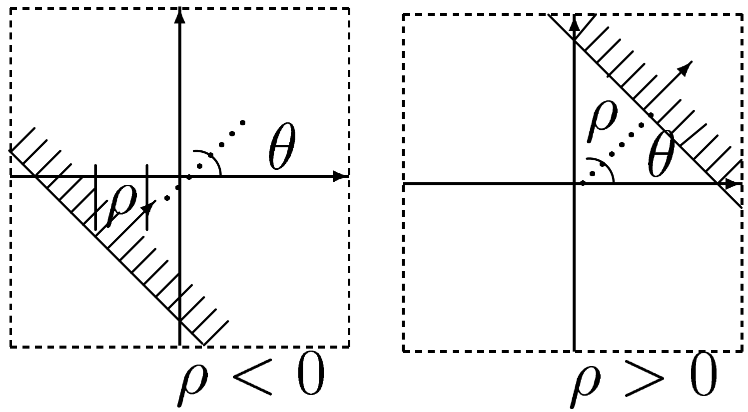
- function $\frac{1}{Z} = \mu(\mathbf{x},\mathbf{P})$ from h_2 : $1/z = -(a_2X + b_2Y + c_2)/d_2$
- 2D straight line $\mathcal{D}$: $A_1x + B_1y + C_1 = 0$

Minimal parameterization (ρ, θ) :

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

with $\theta = \arctan(B_1/A_1)$

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



Computation of the interaction matrix

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

From g, we write $x = f(y,\rho,\theta)$ to get :

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

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

$$\Rightarrow \begin{array}{l} L_\rho^T = [\ \lambda_\rho c\theta \ \ \lambda_\rho s\theta \ \ -\lambda_\rho\rho \ \ (1+\rho^2)s\theta \ \ -(1+\rho^2)c\theta \ \ 0\ ] \\ L_\theta^T = [\ \lambda_\theta c\theta \ \ \lambda_\theta s\theta \ \ -\lambda_\theta\rho \ \ -\rho c\theta \ \ -\rho s\theta \ \ -1\ ] \end{array}$$

with $\lambda_\rho = (a_2\rho c\theta + b_2\rho c\theta + c_2)/d_2$ and $\lambda_\theta = (a_2 s\theta - b_2 c\theta)/d_2$

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

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

$$\text{From } h_2 : \frac{1}{Z} = Ax + By + C \quad \text{with} \quad \begin{cases} A = \alpha/(\alpha x_0 + \beta y_0 + \gamma z_0) \\ B = \beta/(\alpha x_0 + \beta y_0 + \gamma z_0) \\ C = \gamma/(\alpha x_0 + \beta y_0 + \gamma z_0) \end{cases}$$

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

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 :

$$\textbf{moments } m_{ij} = \int\int_{\mathcal{D}(t)} x^i y^j \, dx dy, \;\; i+j = 1, 2$$

$\mathbf{L_p}^T$ is always of full rank 5, but for the centered circle $(x_c = y_c = \mu_{11} = a = b = 0, \mu_{20} = \mu_{02} = r^2)$ where :

$$\mathbf{L_p}^T = \begin{pmatrix} -1/Z_0 & 0 & 0 & 0 & -1-r^2 & 0 \\ 0 & -1/Z_0 & 0 & 1+r^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2r^2/Z_0 & 0 & 0 & 0 \\ 0 & 0 & 2r^2/Z_0 & 0 & 0 & 0 \end{pmatrix}$$

Summary

3D primitives	2D primitives	Parameterization
point	point	(x, y)
segment	segment	(x_1, y_1, x_2, y_2) (x_c, y_c, l, α)
straight line	straight line	(ρ, θ)
circle	ellipse	$(x_c, y_c, \mu_{20}, \mu_{11}, \mu_{02})$
sphere	ellipse	$(x_c, y_c, (\mu_{20} + \mu_{02})/2)$
cylinder	2 straight lines	$(\rho_1, \theta_1, \rho_2, \theta_2)$

$\mathbf{L}_\mathbf{s}^T$ also available for distance from a point to a straight line, angle between two straight lines, etc.

2D visual features : planar object with complex shapes

- **polar signature** $\rho(\theta)$ of edges as Fourier series

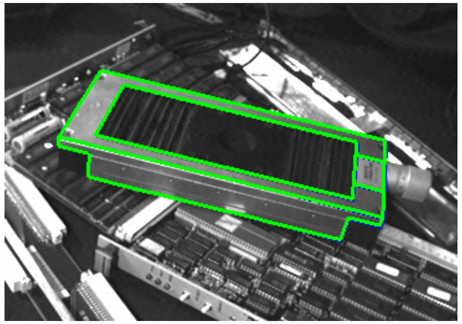
- **moments**: $m_{ij}(t) = \int\int_{\mathcal{D}(t)} x^i y^j \, dx dy$

2D visual features : unknown complex objects

Analytical form of $\mathbf{L}_\mathbf{s}^T$ not available

Off-line learning: **eigenspace** for **appearance** [Nayar 96, Deguichi 97]
neural networks [Suh 93, Wells 96]

3D visual features



Based on pose estimation $\hat{\mathbf{r}}(t)$ from R_c to R_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

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

Pose estimation problem $\sim$ camera calibration problem

(intrinsic camera parameters already known)

Dementhon's method (1995)

$$(x_i, y_i) = \pi(X_i, Y_i, Z_i)_{|\mathcal{R}_o} = \pi(X'_i, Y'_i, Z'_i)_{|\mathcal{R}_c} \Leftrightarrow \left\{ \begin{array}{l} x_i = \dfrac{X'_i}{Z'_i} = \dfrac{a_{11}X_i + a_{12}Y_i + a_{13}Z_i + X_o}{a_{31}X_i + a_{32}Y_i + a_{33}Z_i + Z_o} \\ \\ y_i = \dfrac{Y'_i}{Z'_i} = \dfrac{a_{21}X_i + a_{22}Y_i + a_{23}Z_i + Y_o}{a_{31}X_i + a_{32}Y_i + a_{33}Z_i + Z_o} \end{array} \right.$$

One sets $\left\{ \begin{array}{lcl} \epsilon_i & = & (a_{31}X_i + a_{32}Y_i + a_{33}Z_i)/Z_o \\ I^T & = & (a_{11}/Z_o, a_{12}/Z_o, a_{13}/Z_o, X_o/Z_o) \\ J^T & = & (a_{21}/Z_o, a_{22}/Z_o, a_{23}/Z_o, Y_o/Z_o) \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \begin{pmatrix} X_i & Y_i & Z_i & 1 \end{pmatrix} I = x_i(1+\epsilon_i) \\ \begin{pmatrix} X_i & Y_i & Z_i & 1 \end{pmatrix} J = y_i(1+\epsilon_i) \end{array} \right.$

One would obtain 2 **linear systems** with N equations and 4 unknowns if ϵ_i was not unknown.

4 non planar points at least have to be used

But ϵ_i is unknown...

Linear iterative method

0. Initialization : $\epsilon_i = 0, i = 1 \cdots N$

1. Solve the linear systems $\mathbf{AX} = \mathbf{B}$
   ($\mathbf{A}^+$ has to be computed only once).

2. $Z_0 = 2/(\sqrt{I_1^2 + I_2^2 + I_3^2} + \sqrt{J_1^2 + J_2^2 + J_3^2}) \Rightarrow a_{ij}, X_o$ and Y_o.

3. $\epsilon_i = (a_{31}X_i + a_{32}Y_i + a_{33}Z_i)/Z_o$.

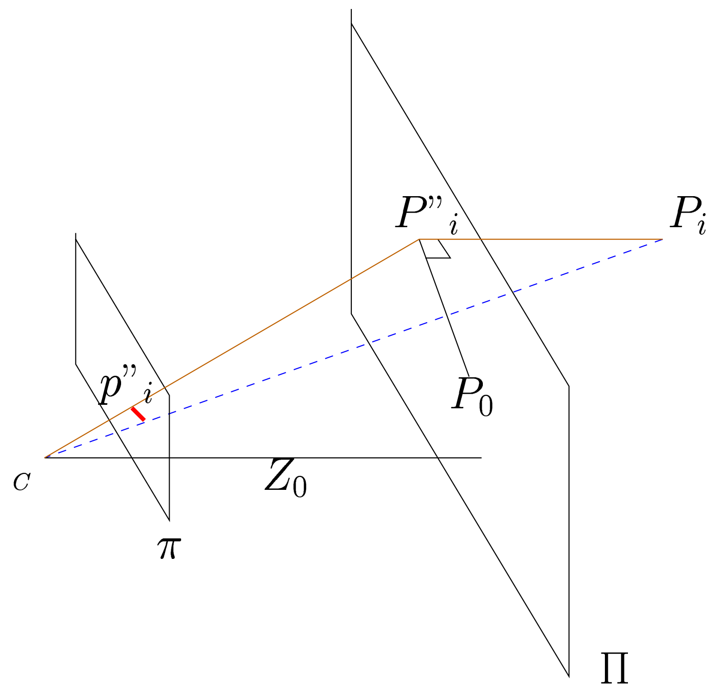
4. Go to 1 if needed.

Generally, only few iterations (4 or 5) are necessary if ϵ_i is not too large.

Geometric interpretation

Projection model:

para-perspective.



$P"_i$ has coordinates (X_i', Y_i', Z_o) in $\mathcal{R}_c$

$$\Rightarrow x"_i = X_i'/Z_o \ , \ y"_i = Y_i'/Z_o$$

$$\Rightarrow \epsilon_i = \frac{x"_i - x_i}{x_i} = \frac{y"_i - y_i}{y_i}$$

Efficient algorithm if $\epsilon_i \ll 1$, *i.e.* $d(\pi, \Pi) \gg d(p_i, \Pi)$.

Non linear method (Lowe's method)

Objective function to be minimized

$$C = \sum_{i=1}^{N} [(x_{m_i} - x_i)^2 + (y_{m_i} - y_i)^2]$$

- (x_i, y_i) : measure in the image
- $(x_{m_i}, y_{m_i}) = \pi({}^c\mathbf{A}_o \mathbf{X}_{o_i} + {}^c\mathbf{T}_o)$

Rotation ${}^c\mathbf{A}_o$ parameterized by $\theta\mathbf{U} \Rightarrow$ 6 independent unknowns

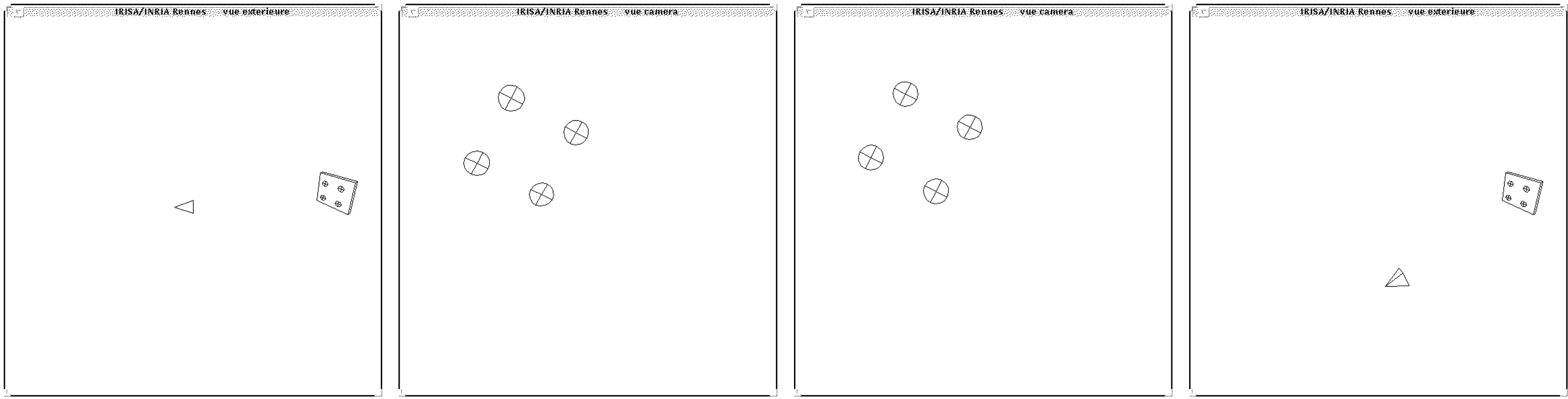
Minimization using Gauss-Newton method (*Levenberg-Marquartd*)

3D visual features

$$\text{estimated pose } \hat{\mathbf{r}}(t) = \hat{\mathbf{r}}(\mathbf{x}(t), \mathbf{X}, x_c, y_c, f_x, f_y)$$

$$\Rightarrow \quad \dot{\hat{\mathbf{r}}}(t) = \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}}\,\dot{\mathbf{x}} = \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}}\,\frac{\partial \mathbf{x}}{\partial \mathbf{r}}\,\mathcal{T} \quad \Rightarrow \quad \mathbf{L}_{\hat{\mathbf{r}}}^T = \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}}\,\frac{\partial \mathbf{x}}{\partial \mathbf{r}}$$

where $\frac{\partial \mathbf{x}}{\partial \mathbf{r}} = \mathbf{L}_{\mathbf{x}}^T$ is known but $\frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}}$ is unknown (and sometimes unstable)



3D visual features

Under the strong hypothesis that pose estimation is perfect

- rotation $\theta\mathbf{u}$ to realize

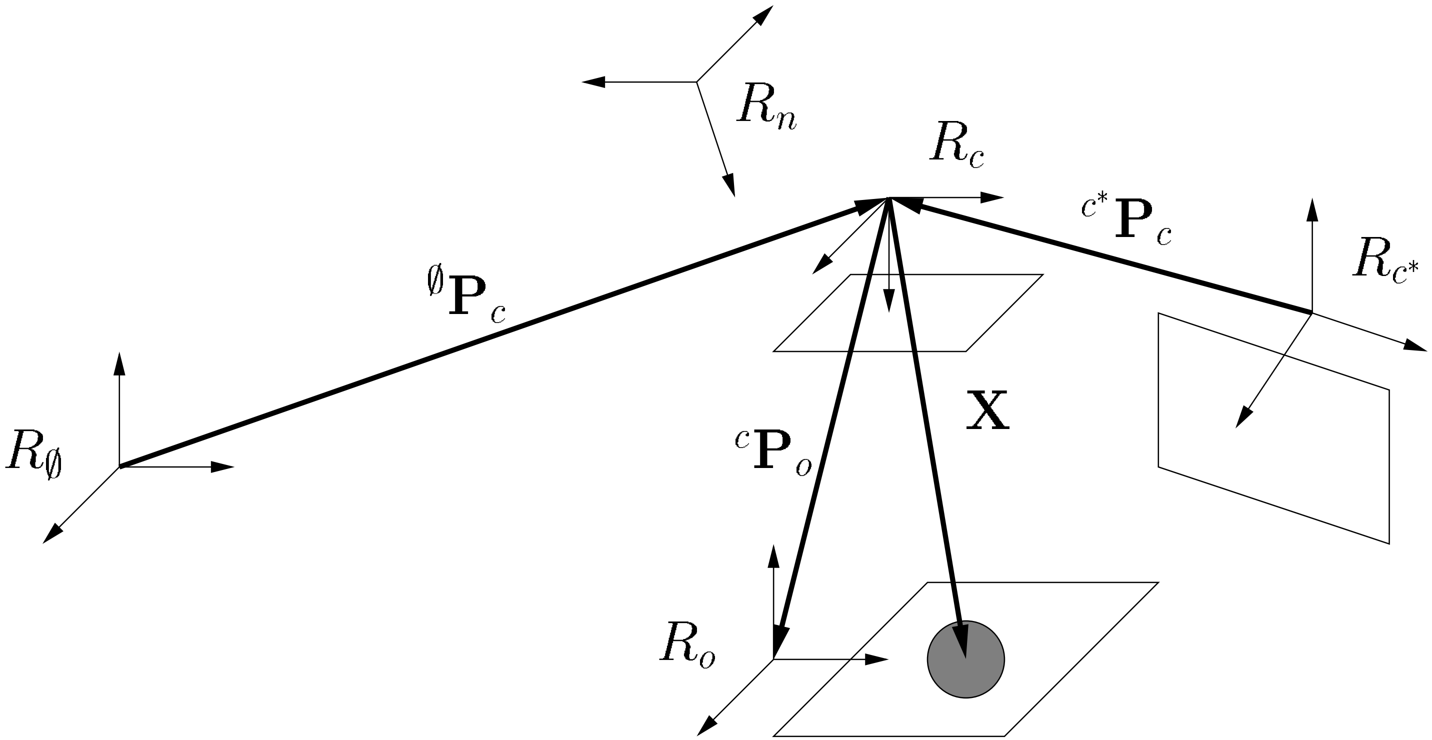
$$\mathbf{L}_{\theta\mathbf{u}}^{T} = \begin{pmatrix} \mathbf{0}_3 & \mathbf{L}_\omega \end{pmatrix} \text{ where } \mathbf{L}_\omega \text{ such that } \mathbf{L}_\omega \theta\mathbf{u} = \mathbf{L}_\omega^{-1}\, \theta\mathbf{u} = \theta\mathbf{u}$$

$$\mathbf{L}_\omega = \mathbb{I}_3 - \frac{\theta}{2}\,\tilde{\mathbf{u}} + \left(1 - \frac{\mathrm{sinc}\theta}{\mathrm{sinc}^2\frac{\theta}{2}}\right)\tilde{\mathbf{u}}^2$$

- coordinates of a 3D point X : $\dot{\mathbf{X}} = -\mathbf{V} - \tilde{\omega}\mathbf{X}$

$$\Rightarrow \mathbf{L}_{\mathbf{X}}^{T} = \begin{pmatrix} -\mathbb{I}_3 & \tilde{\mathbf{X}} \end{pmatrix}$$

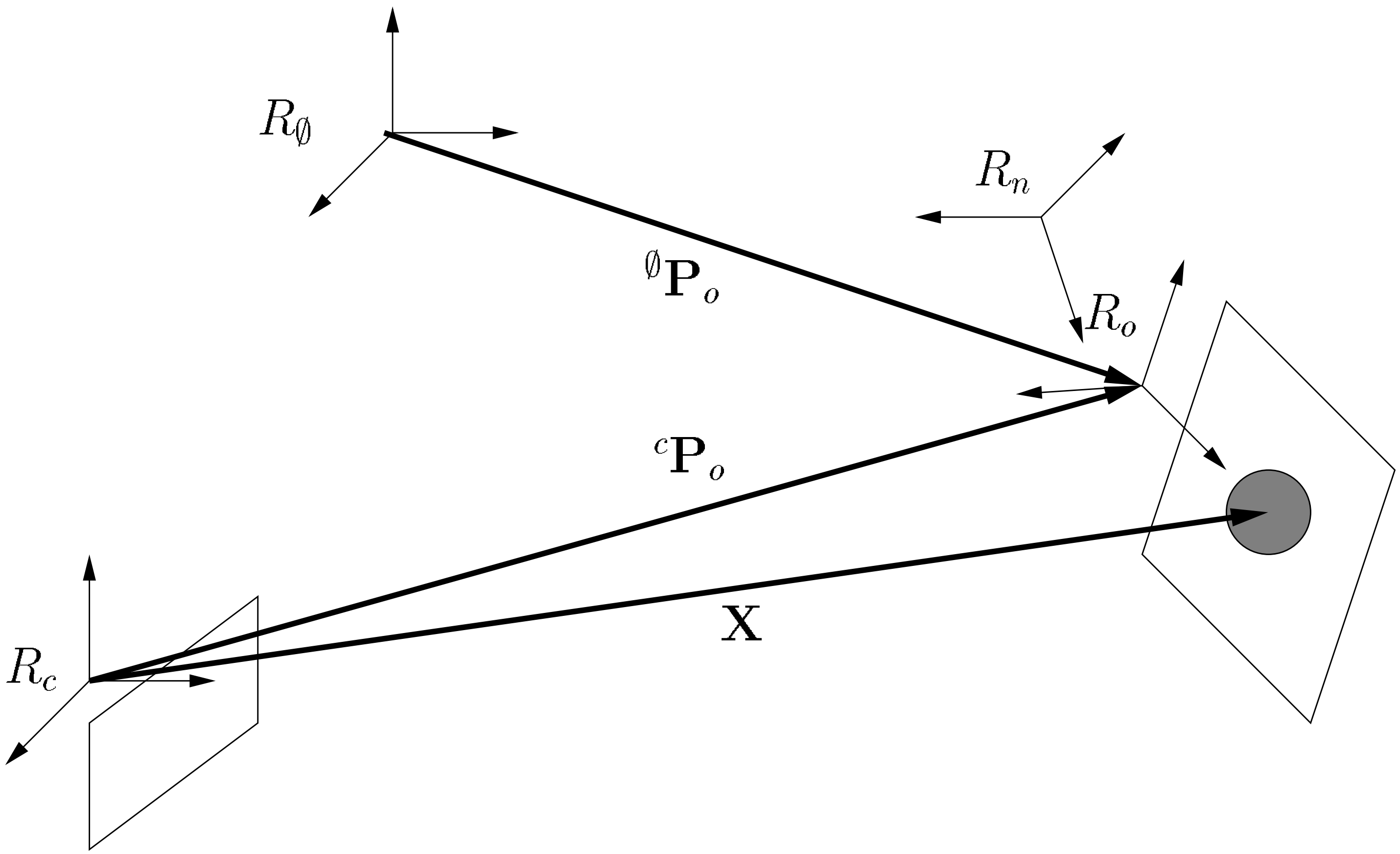
3D visual features for an eye-in-hand system



$$\mathbf{L}^T_{{}^c\mathbf{P}_o} = \begin{pmatrix} -\mathbb{I}_3 & {}^{\mathbf{c}}\tilde{\mathbf{P}}_{\mathbf{o}} \end{pmatrix} \qquad \mathbf{L}^T_{{}^o\mathbf{P}_c} = \begin{pmatrix} {}^o\mathbf{A}_c & \mathbf{0}_3 \end{pmatrix}$$

$$\mathbf{L}^T_{{}^\emptyset\mathbf{P}_c} = \begin{pmatrix} {}^\emptyset\mathbf{A}_c & \mathbf{0}_3 \end{pmatrix} \qquad \mathbf{L}^T_{{}^{c^*}\mathbf{P}_c} = \begin{pmatrix} {}^{c^*}\mathbf{A}_c & \mathbf{0}_3 \end{pmatrix}$$

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



$$\mathbf{L}^T_{{}^c\mathbf{P}_o}\,{}^c\mathbb{T}_o = \begin{pmatrix} -{}^c\mathbf{A}_o & \mathbf{0}_3 \end{pmatrix} \qquad {}^\emptyset\dot{\mathbf{P}}_o = \begin{pmatrix} \mathbb{I}_3 & \mathbf{0}_3 \end{pmatrix}\ {}^\emptyset\mathcal{T}_o$$

1) Modeling issues

$\Rightarrow$ 2) Classification of the vision-based tasks

3) Control issues

4) Use of other exteroceptive sensors

Classification of the vision-based tasks

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

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

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

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.

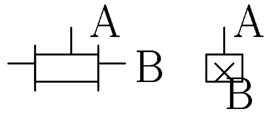
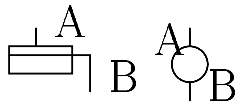
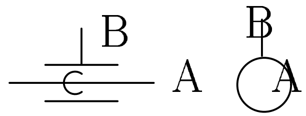
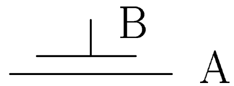
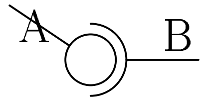
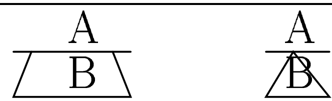
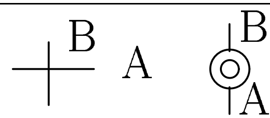
Case of a point

$$\mathbf{s} = (x\ y)^T$$

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

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

$\Rightarrow$ 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 rectlinear	4	2	2	
Linear annular	4	1	3	
Point	5	2	3	

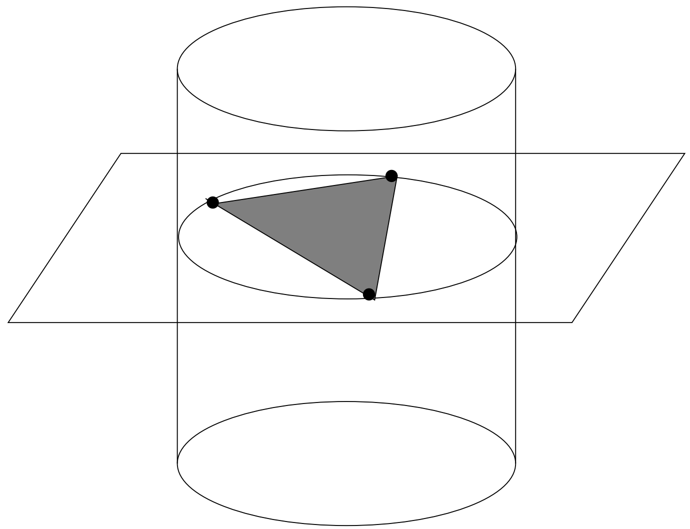
Rigid link

$$\mathcal{S}^* = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

Using 3 points :

$$\mathbf{L}_{\mathbf{s}}^T = \begin{pmatrix} -1/Z_1 & 0 & x_1/Z_1 & x_1y_1 & -(1+x_1^2) & y_1 \\ 0 & -1/Z_1 & y_1/Z_1 & 1+y_1^2 & -x_1y_1 & -x_1 \\ -1/Z_2 & 0 & x_2/Z_2 & x_2y_2 & -(1+x_2^2) & y_2 \\ 0 & -1/Z_2 & y_2/Z_2 & 1+y_2^2 & -x_2y_2 & -x_2 \\ -1/Z_3 & 0 & x_3/Z_3 & x_3y_3 & -(1+x_3^2) & y_3 \\ 0 & -1/Z_3 & y_3/z_3 & 1+y_3^2 & -x_3y_3 & -x_3 \end{pmatrix}$$

Isolated singularities exist



4 poses are solution of the **P3P** problem

Solution : Using at least 4 points

Prismatic link

$$\mathcal{S}^* = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

Using 3 (horizontal) straight lines

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

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

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

With a 3 dof mobile robot (V_x, V_z, Ω_y), 1 straight line is sufficient.