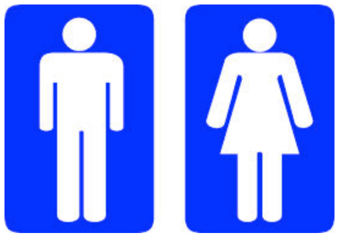
- Robots
- Sensors
- **Image processing**
- Computer vision
- Robot control
- Example applications

Image processing

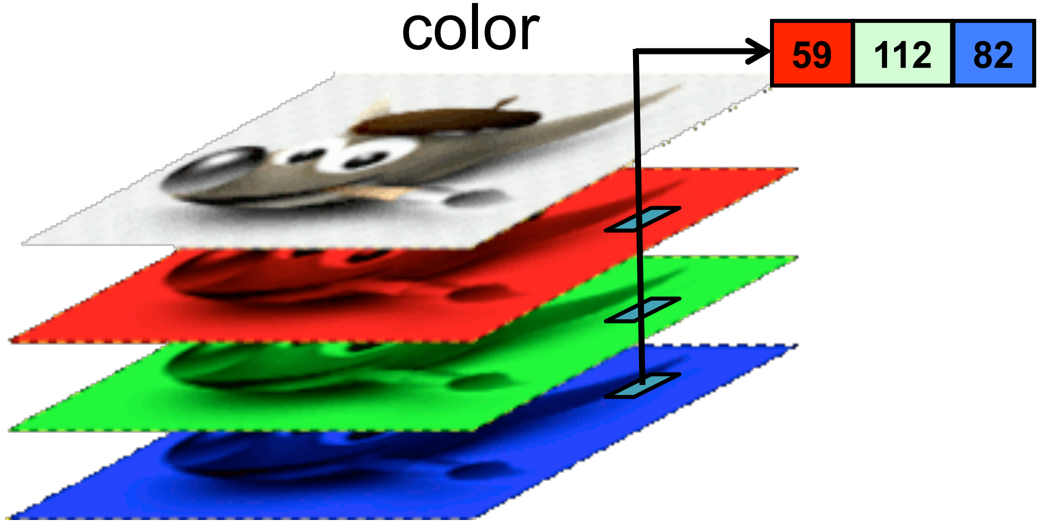
- signal processing for which the input is an image, the output either an image or a set of characteristics related to the image
- most image-processing techniques involve treating the image as a 2D signal and applying standard signal-processing techniques to it



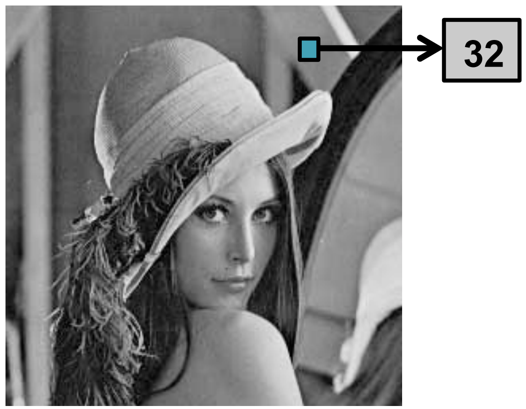
- image signal
- convolution
- morphological operations
- template matching

Image Signal

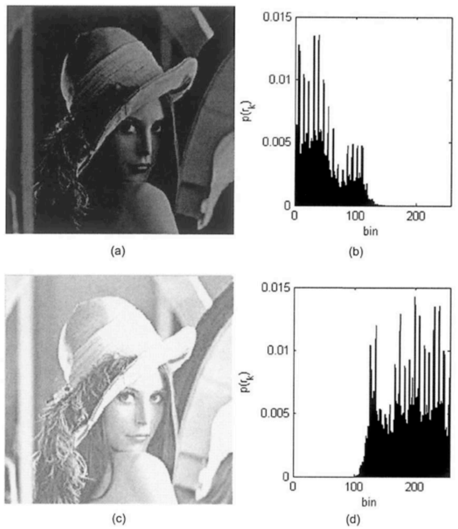
color



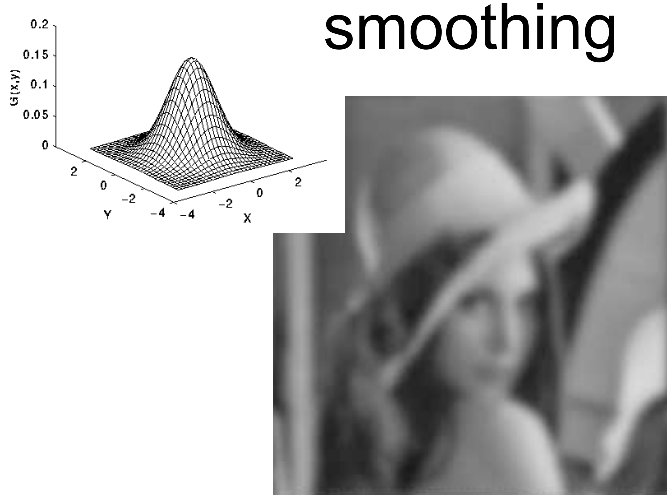
grey



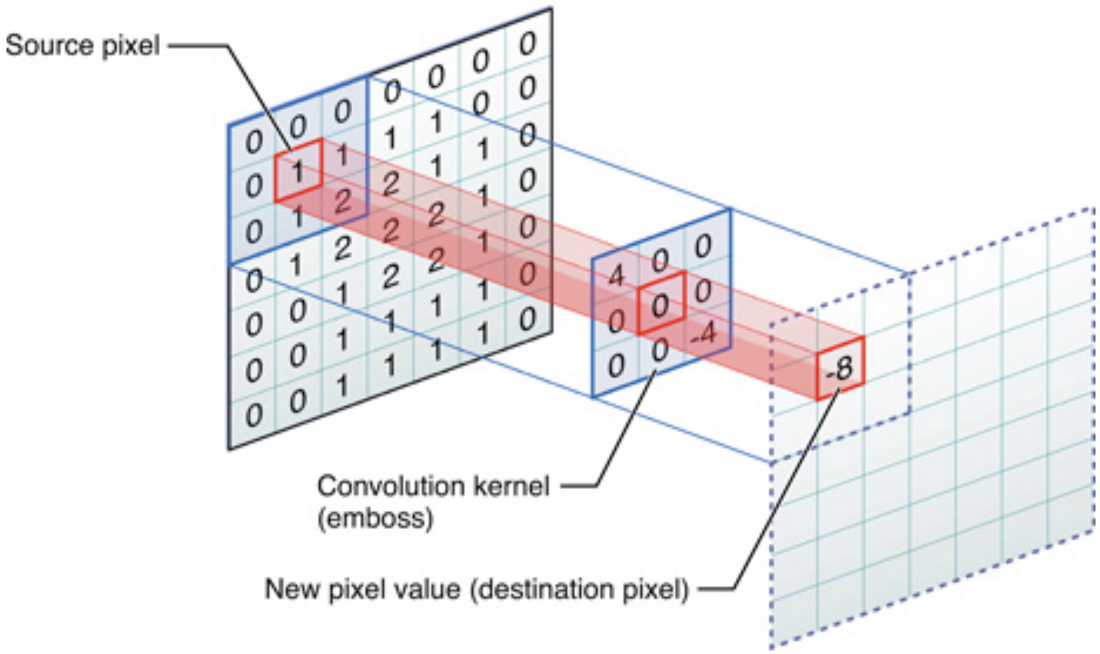
Luminosity changes



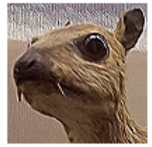
smoothing



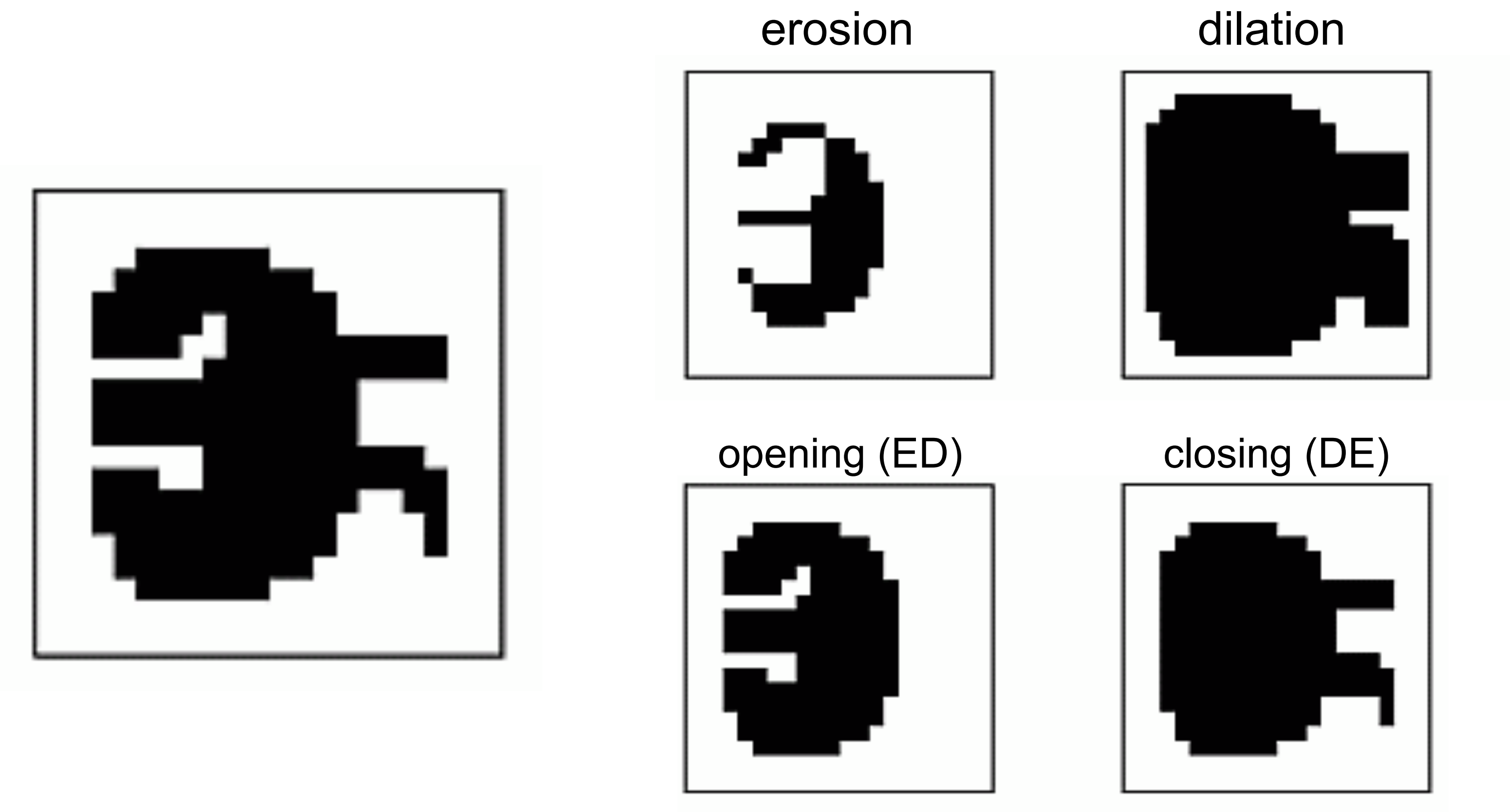
Convolution

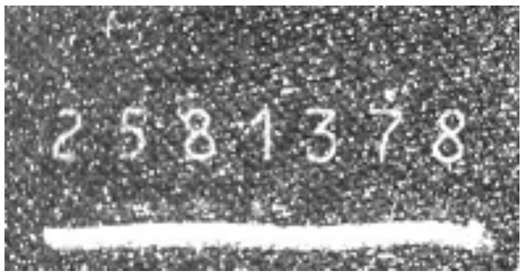


$$f(x,y) = \frac{1}{N} \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+j) w(x+i, y+j)$$

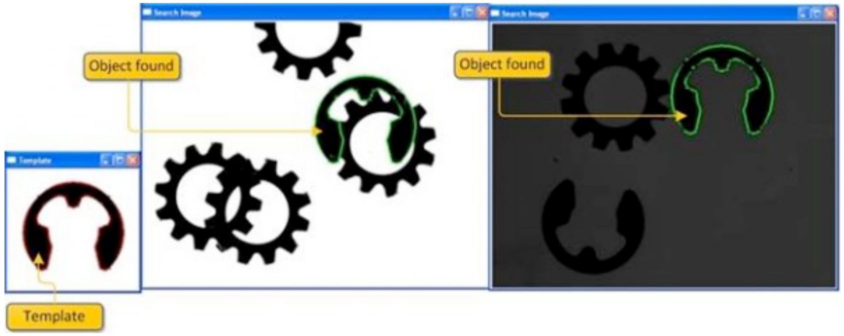
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

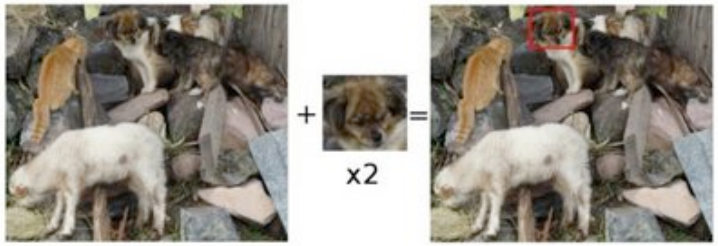
Morphological operators



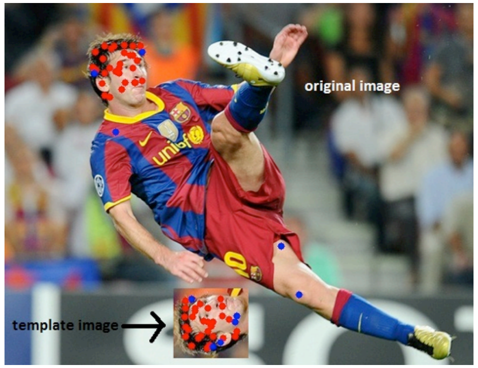


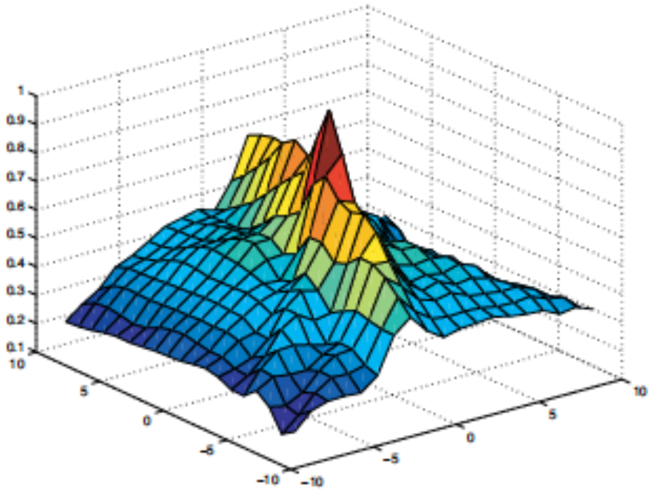
Template matching

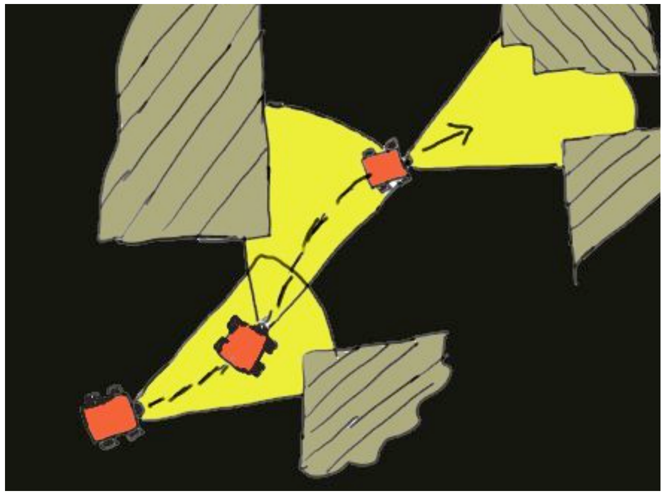




$$\frac{\sum_{i,j}(I_1(i,j)-\bar{I}_1)(I_2(i+u,j+v)-\bar{I}_2)}{\sqrt{\sum_{i,j}(I_1(i,j)-\bar{I}_1)^2\sum_{i,j}(I_2(i,j)-\bar{I}_2)^2}}$$







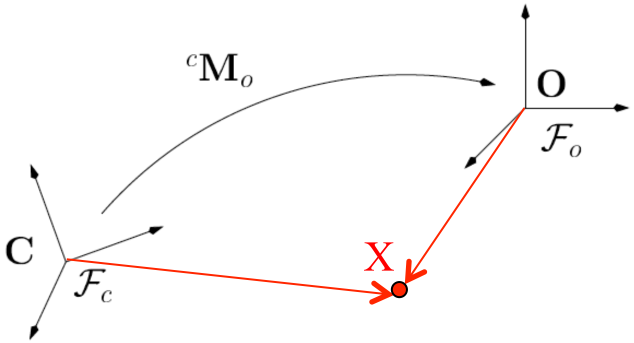
- Robots
- Sensors
- Image processing
- **Computer vision**
- Robot control
- Example applications

Computer vision

- computer vision includes methods for acquiring, processing, analyzing, and understanding images from the real world to produce numerical or symbolic information
- the goal is to duplicate the abilities of human vision by electronically perceiving and understanding an image
- for this, it relies on geometry, physics, statistics, and learning theory

- homogeneous transformations
- epipolar geometry
- homography
- 3D reconstruction
- rigid object recognition and tracking
- non-rigid object recognition and tracking

Homogeneous transformations



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

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

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

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

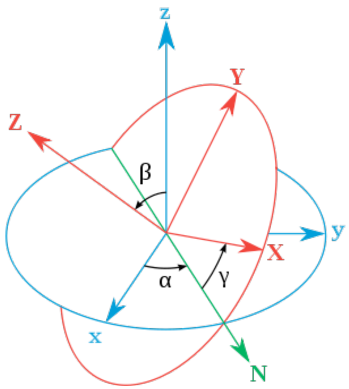
${}^c\mathbf{R}_o$: rotation matrix between $\mathcal{F}_c$ and $\mathcal{F}_o$

$$\begin{bmatrix} X_c \\ 1 \end{bmatrix} = {}^cM_o \begin{bmatrix} X_o \\ 1 \end{bmatrix} = \begin{bmatrix} {}^cR_o & {}^ct_o \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_o \\ 1 \end{bmatrix}$$

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

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

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

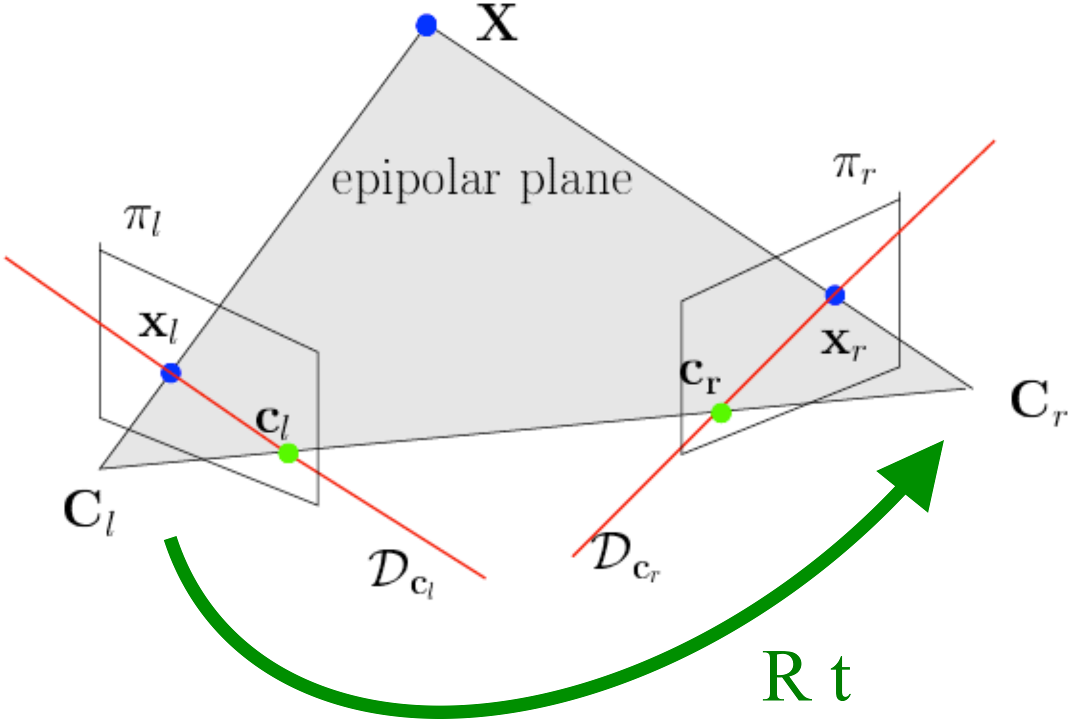


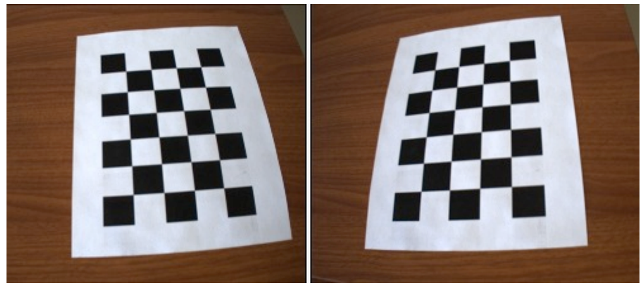
Epipolar geometry

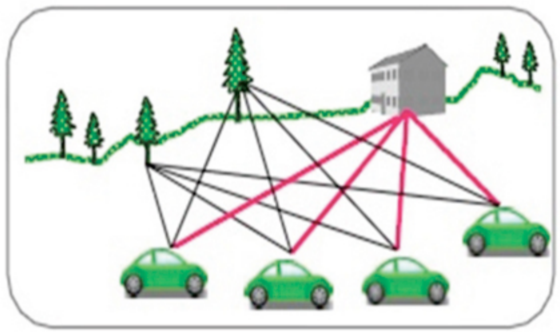
$$\mathbf{x}_l^\top \mathbf{F} \mathbf{x}_r = 0$$

$$\mathbf{E} = \mathbf{K}_l^\top \, \mathbf{F} \, \mathbf{K}_r$$

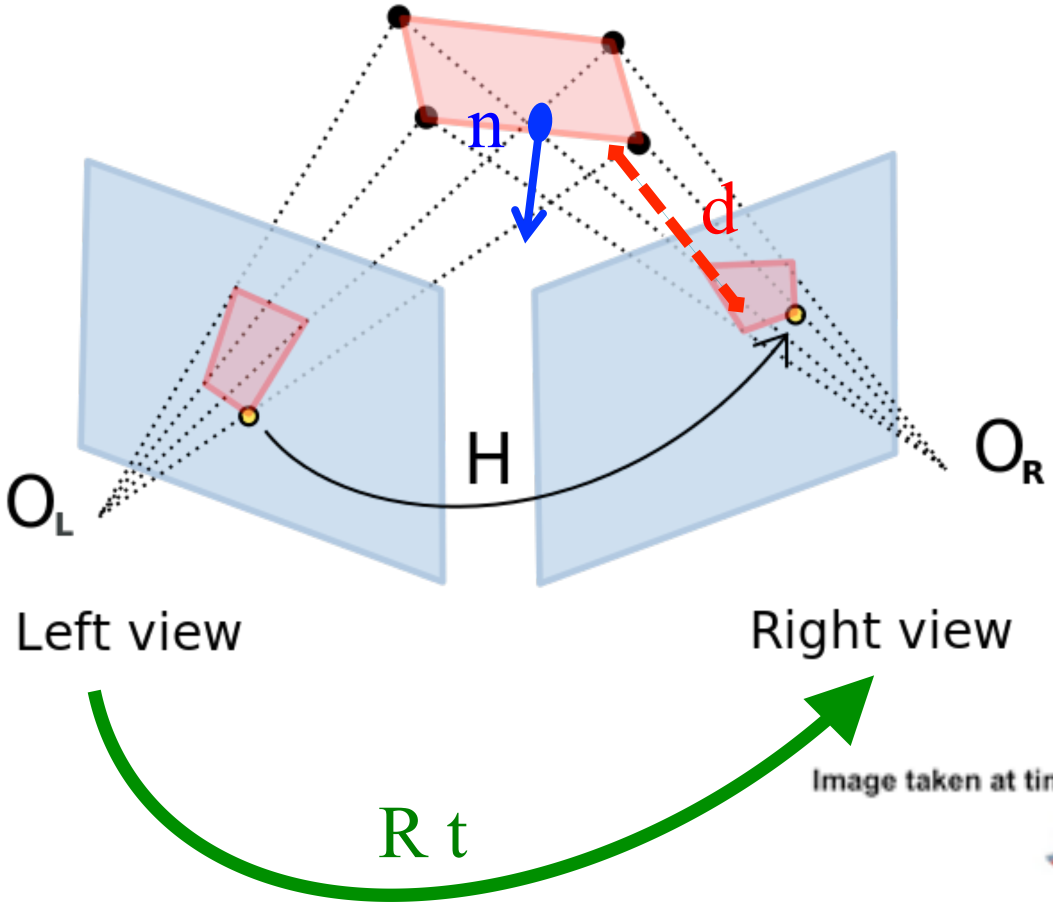
$$\mathbf{E} = \mathbf{R} \, [\mathbf{t}]_\times$$





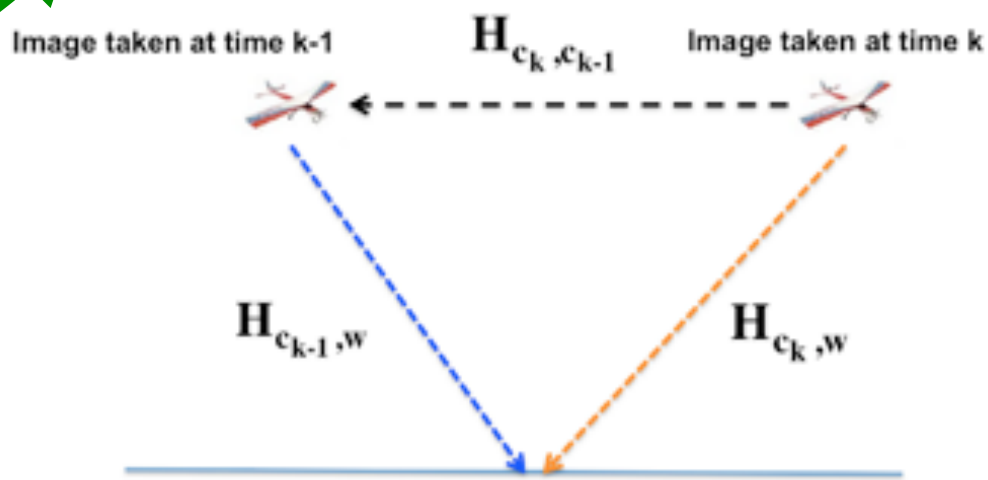


Homography

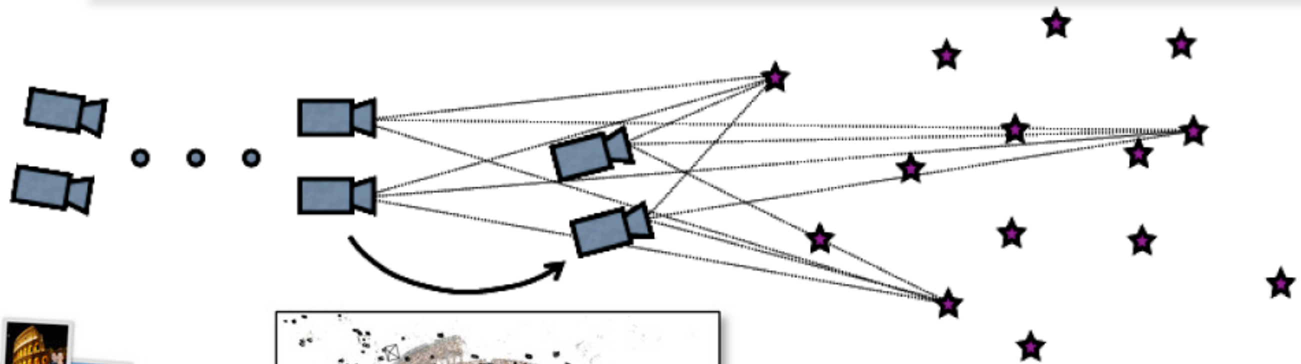


$$x_l = K_l H K_r^{-1} x_r$$

$$H = R - \frac{tn^T}{d}$$



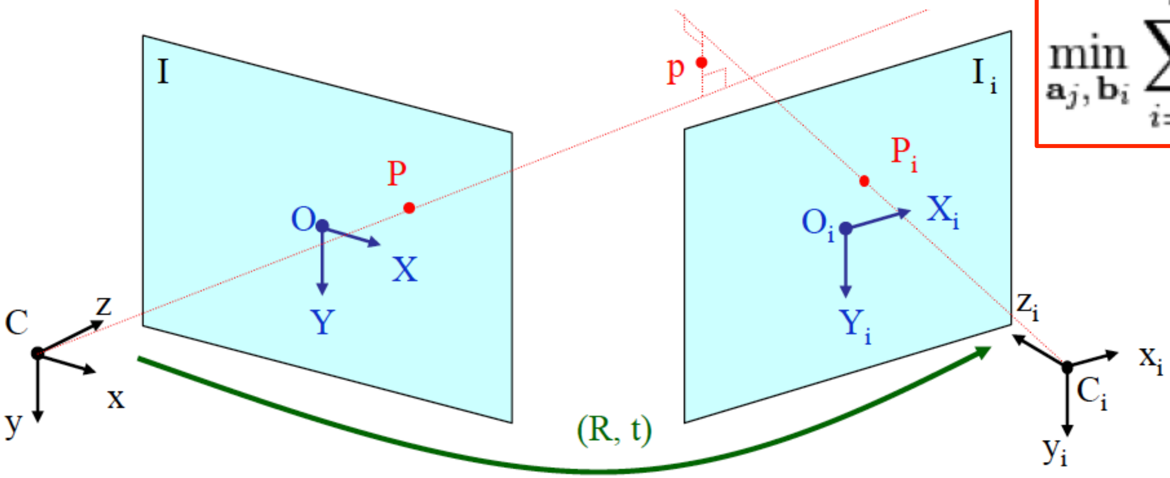
3D Reconstruction

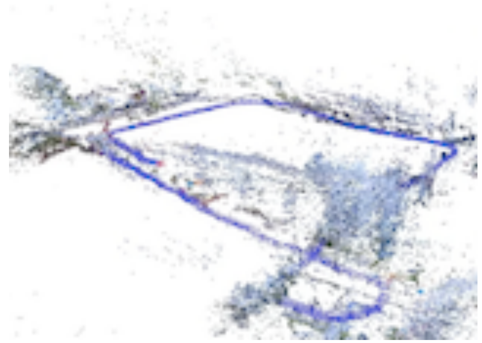


Bundle adjustment

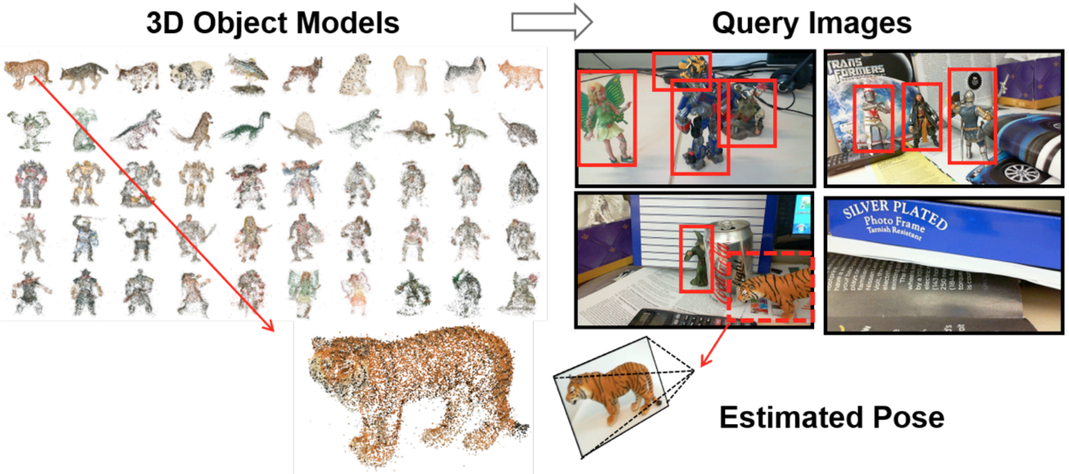
points, views, reprojection

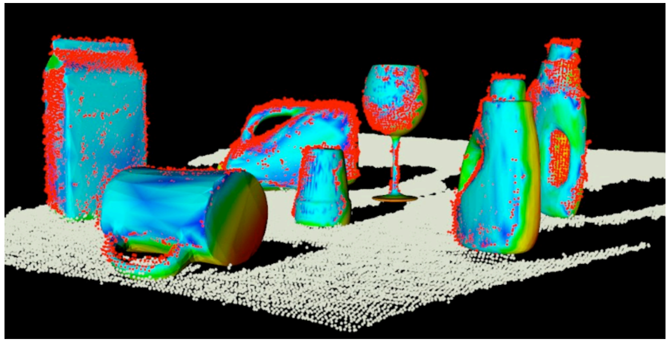
$$\min_{\mathbf{a}_j,\,\mathbf{b}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij}\, d(\mathbf{Q}(\mathbf{a}_j,\, \mathbf{b}_i),\, \mathbf{x}_{ij})^2$$



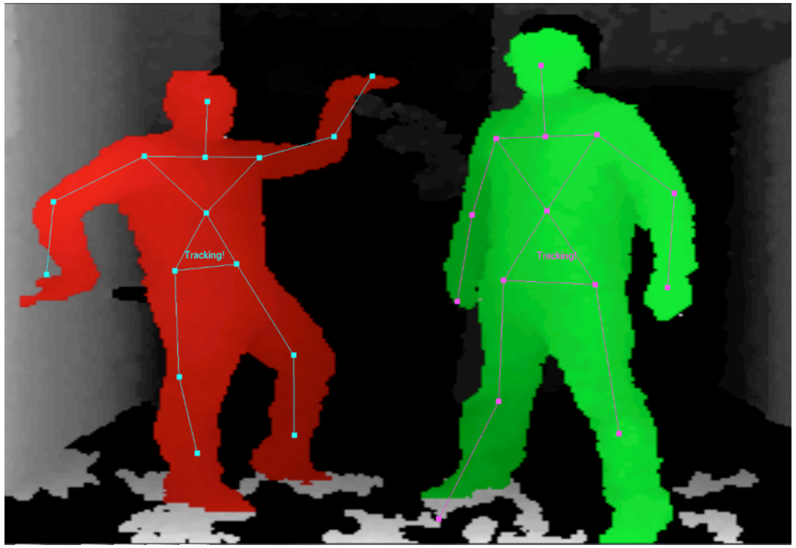


Rigid object recognition and tracking





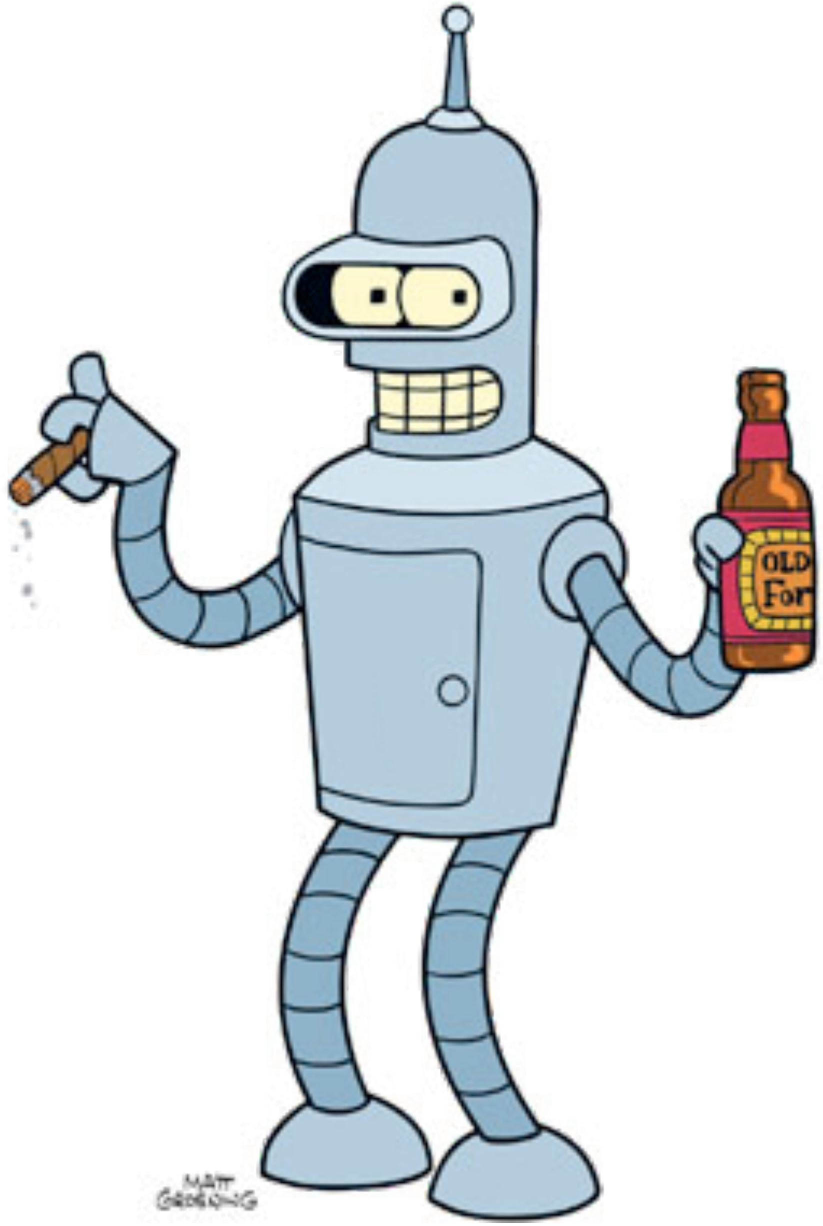
Non-rigid object recognition and tracking



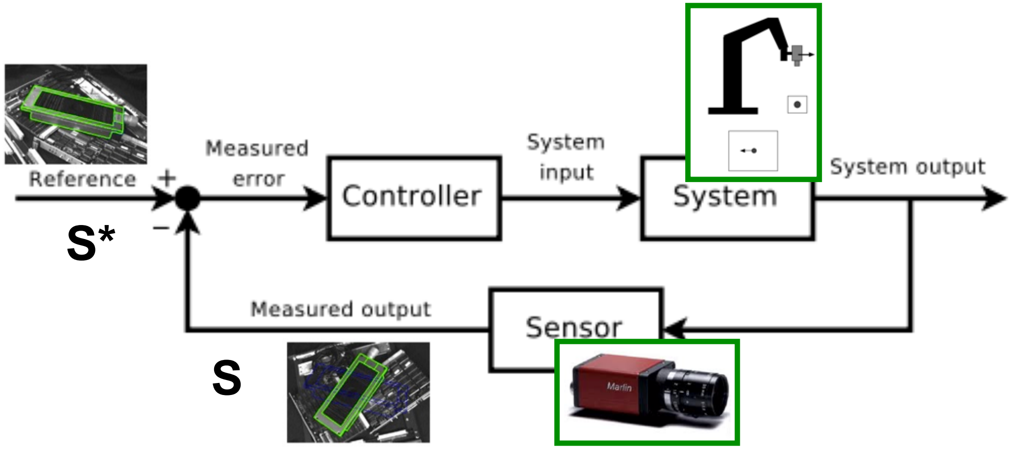
- Robots
- Sensors
- Image processing
- Computer vision
- **Robot control**
- Example applications

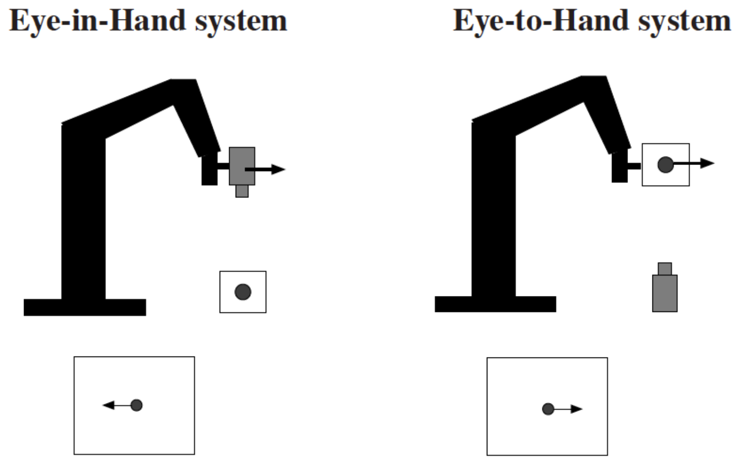
Robot control (visual servoing)

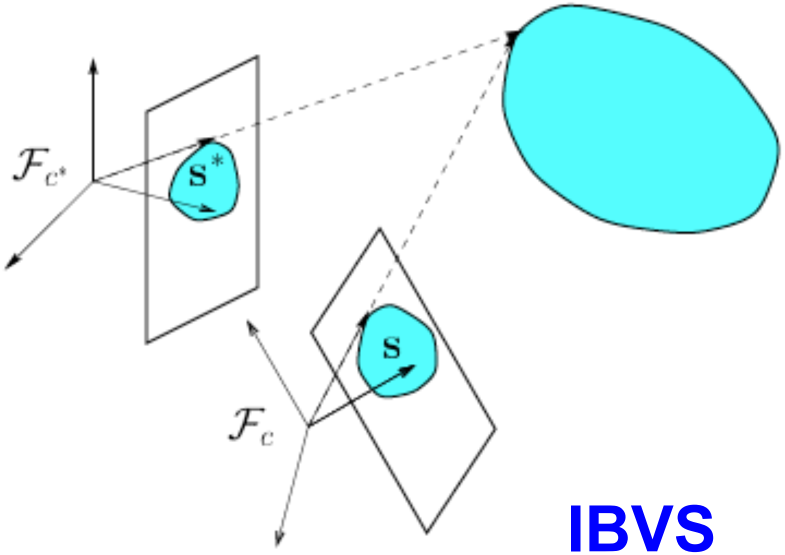
- basics
- interaction matrix
- geometrical primitives
- image moments
- other control strategies



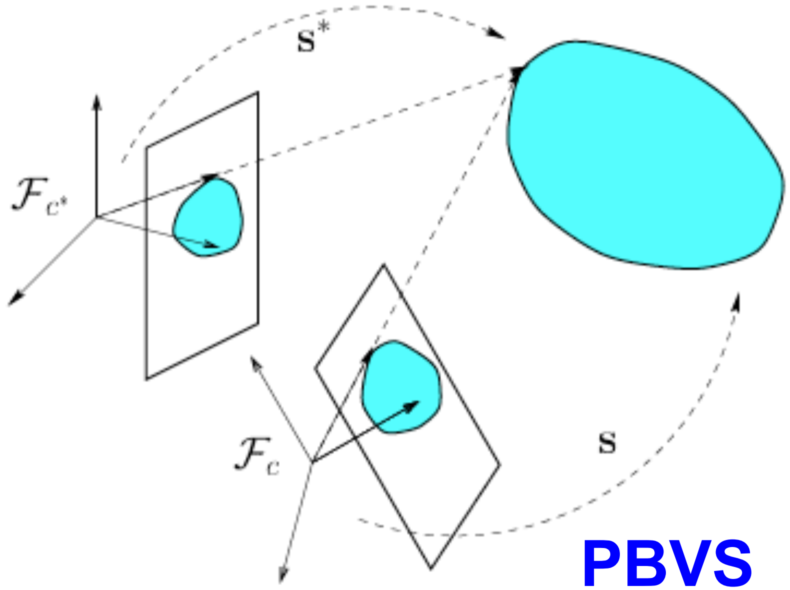
Basics







2D visual features

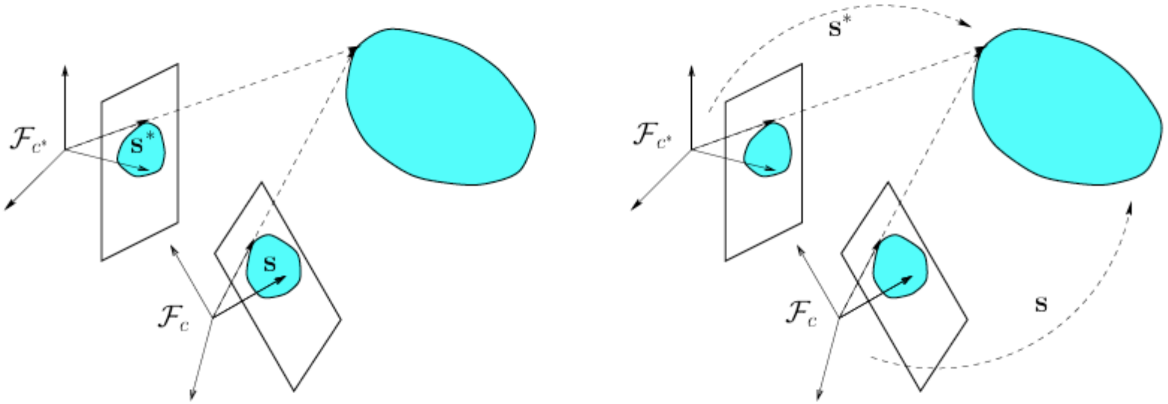


3D visual features

Basics

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

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

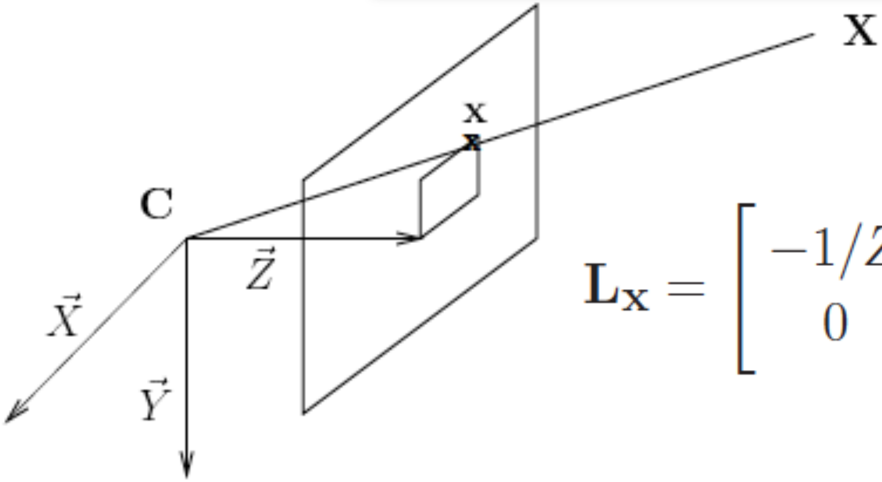


If we want $\dot{\mathbf{s}} = -\lambda(\mathbf{s} - \mathbf{s}^*)$ (exponential decoupled decrease):

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

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

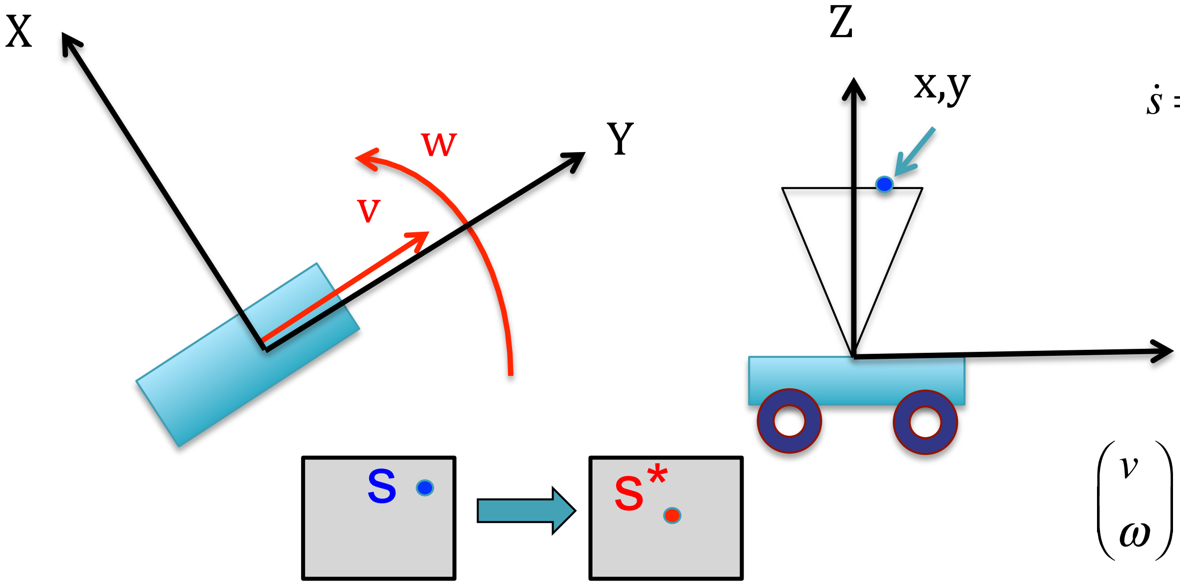
Interaction matrix



$$\dot{\mathbf{s}} = \mathbf{L_s}\,\mathbf{v}$$

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

For example…



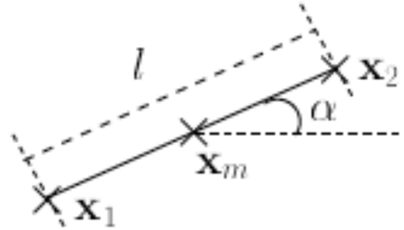
$$\dot{s} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & y \\ -1/Z & -x \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

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

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = -\lambda \begin{pmatrix} -Zx/y & -Z \\ 1/y & 0 \end{pmatrix} \begin{pmatrix} x - x^* \\ y - y^* \end{pmatrix}$$

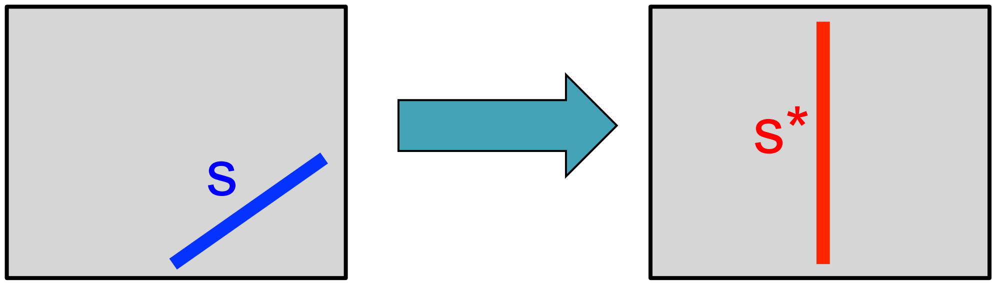
Geometric primitives

Example: segment



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

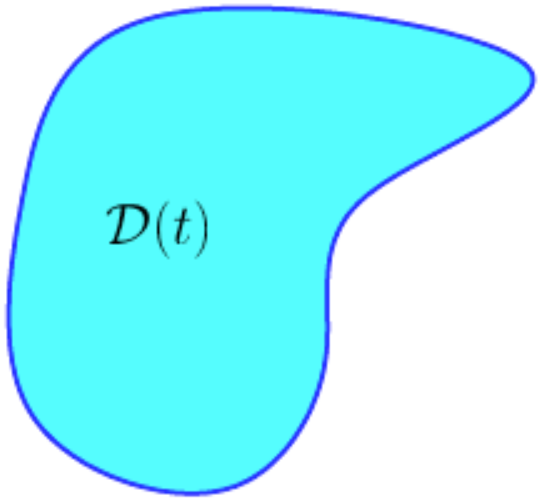
$$\mathbf{v} = -\lambda\, \widehat{\mathbf{L}_{\mathbf{s}}}^{+} (\mathbf{s} - \mathbf{s}^*)$$



Also possible with: spheres, cylinders, straight lines…

Image moments

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}

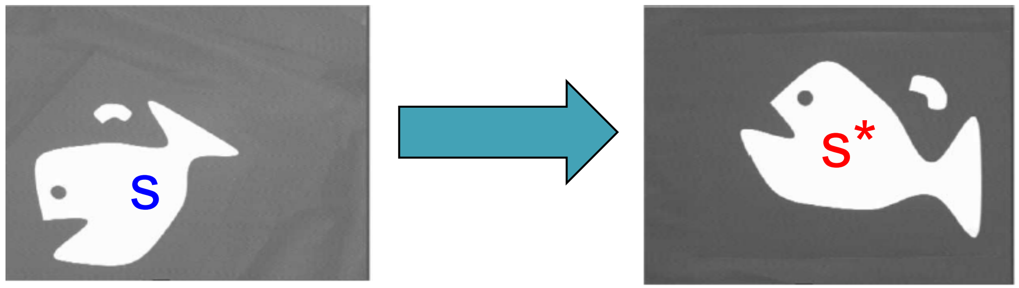
center of gravity $\mathbf{x_g}$: from m_{10} and m_{01}

For planar object: $1/Z = Ax + By + C$

Area $a = m_{00}$

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

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



Other control strategies

- 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]

Moving target tracking $\mathbf{v}_q = \widehat{\mathbf{L}_\mathbf{e}}^{-1} (-\lambda\, \mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}})$

redundancy

2 and ½ visual servoing

Idea : Combine 2D image data and 3D data

$$\mathbf{s} = \begin{bmatrix} x \\ y \\ \log Z \\ \theta u_x \\ \theta u_y \\ \theta u_z \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{image point coordinates} \\ \rightarrow \text{rel. depth} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{rotation to realize} \end{matrix} \Rightarrow \mathbf{L_s}\ \textbf{triangular} \text{ and never singular}$$

- Robots
- Sensors
- Image processing
- Computer vision
- Robot control
- **Example applications**