ICRA'2016 Tutorial on Vision for Robotics

IEEE Robotics & Automation Society

Visual Servoing

François Chaumette

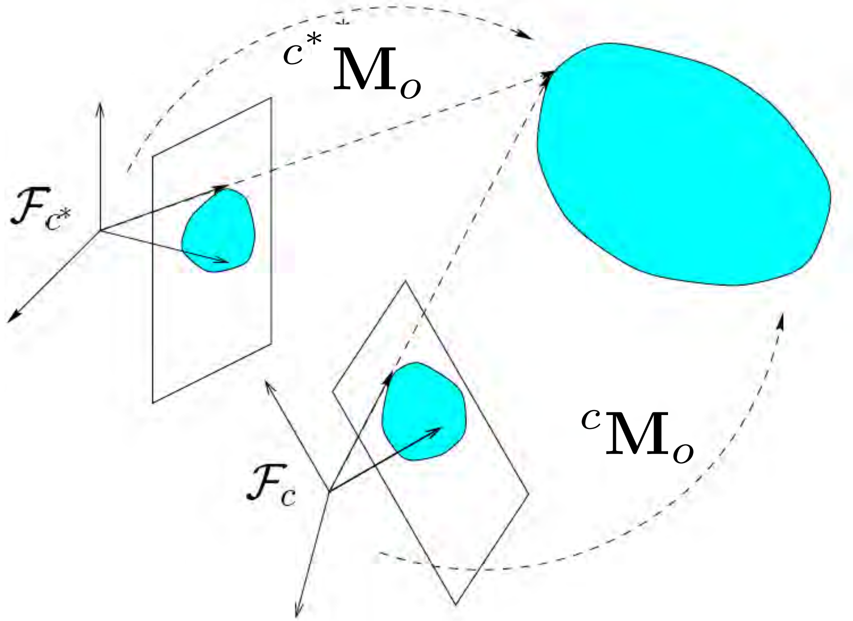
Lagadic group
Inria at Irisa
Rennes, France

http://team.inria.fr/lagadic *http://visp.inria.fr*

How to control robot motion from vision?

1st basic idea: determine only once the displacement to be done (open loop/saccade)

$$^{c^*}\mathbf{M}_c = {}^{c^*}\mathbf{M}_o \, {}^{c}\mathbf{M}_o^{-1}$$



Advantages:

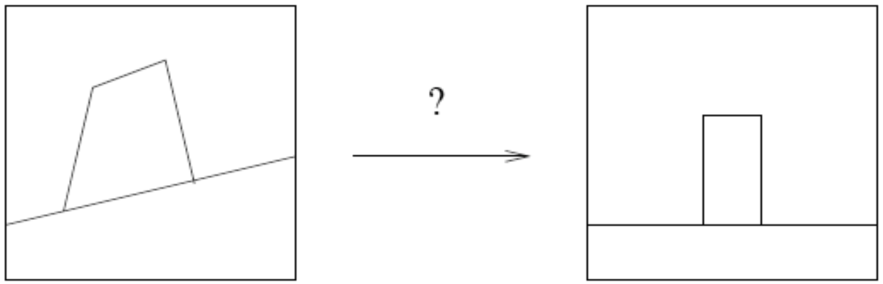
- Only one image to be processed and one very fast displacement to be achieved if the full system is perfectly calibrated

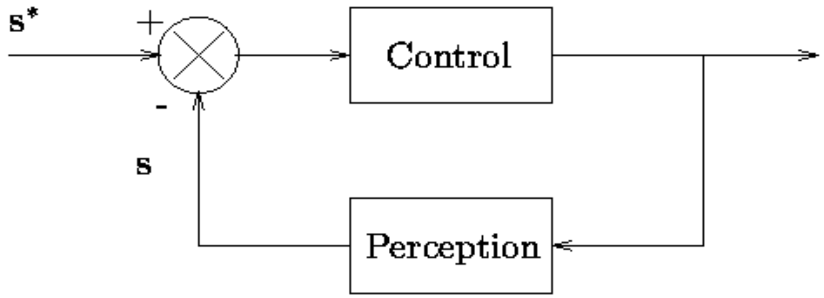
Drawbacks:

- Not robust to modeling and calibration errors
- Iterating may help, or not… Object detection for each new image

What is visual servoing?

Vision-based closed loop control of a dynamic system
by iterative minimization of a visual error (Lyapunov function)





Advantages:

- Positioning accuracy
- Robustness with respect to modeling and calibration errors
- Reactive to changes (target tracking)

Drawbacks:

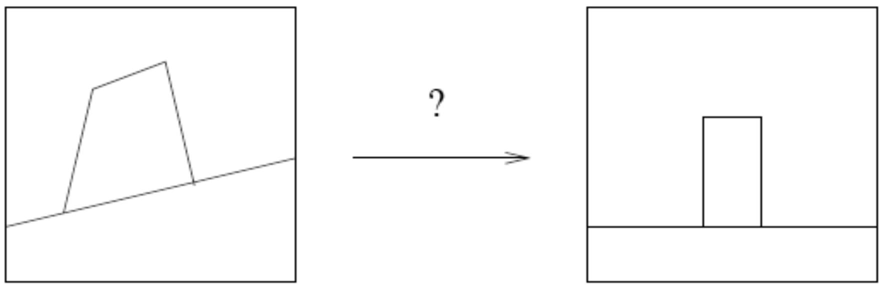
- Need many images to be processed

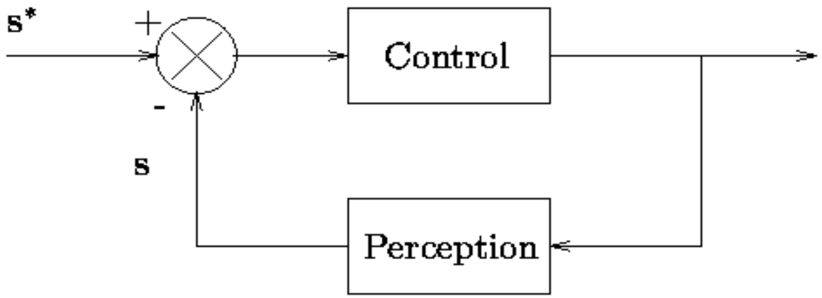
What is visual servoing?

Usual steps:

- extract and track visual measurements near video rate
- design visual features and control schemes from the available measurements
- taking into account the system and environment constraints

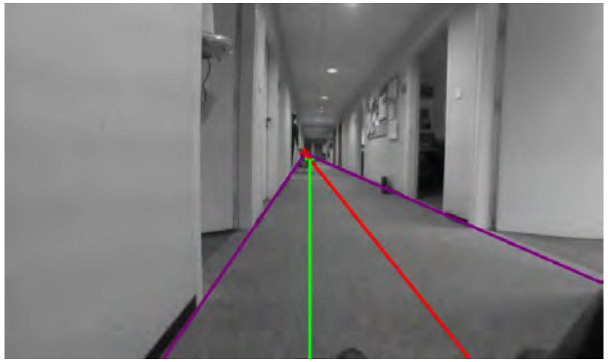
for an adequate system behavior (stability, robustness, …)

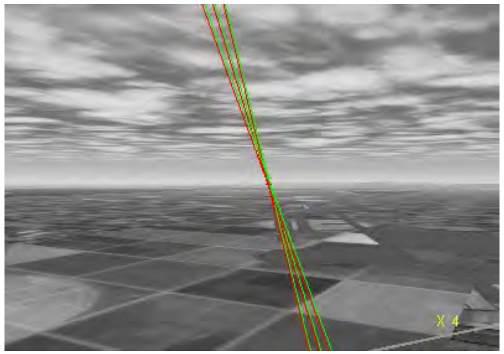


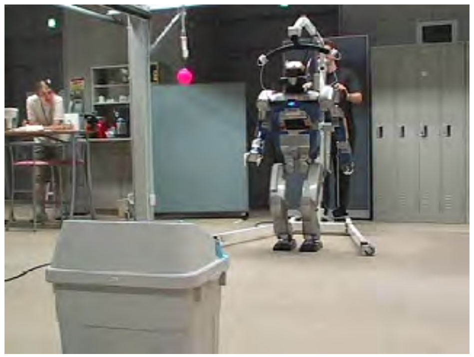


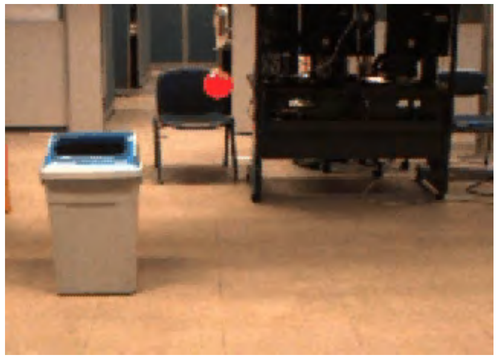
A wide spectrum of applications

Just need a camera and a robot



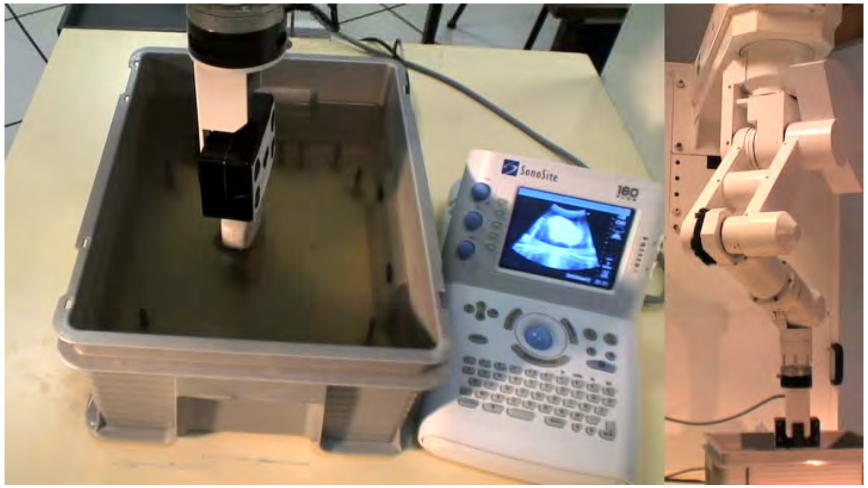




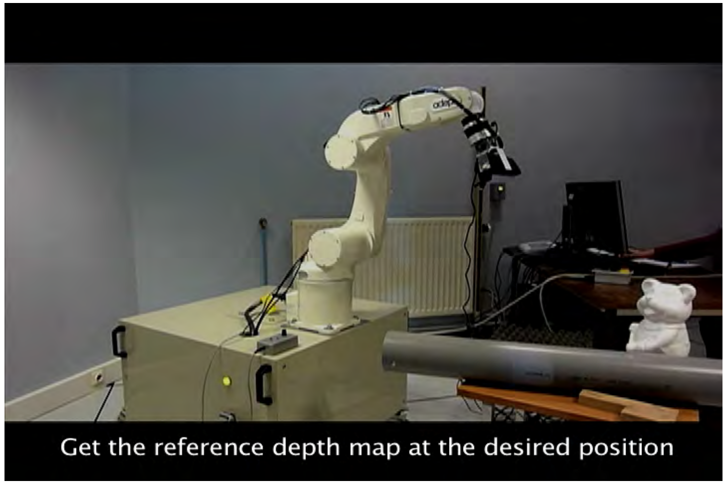


Whatever sort of vision sensor

Omnidirectional camera

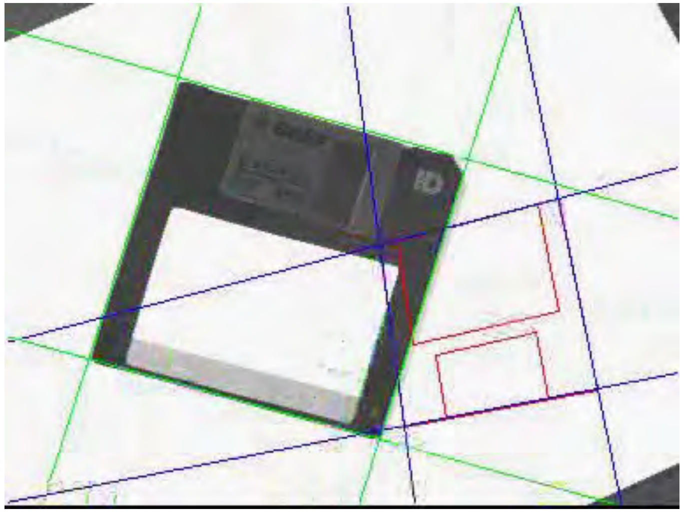


2D US probe



RGB-D sensor

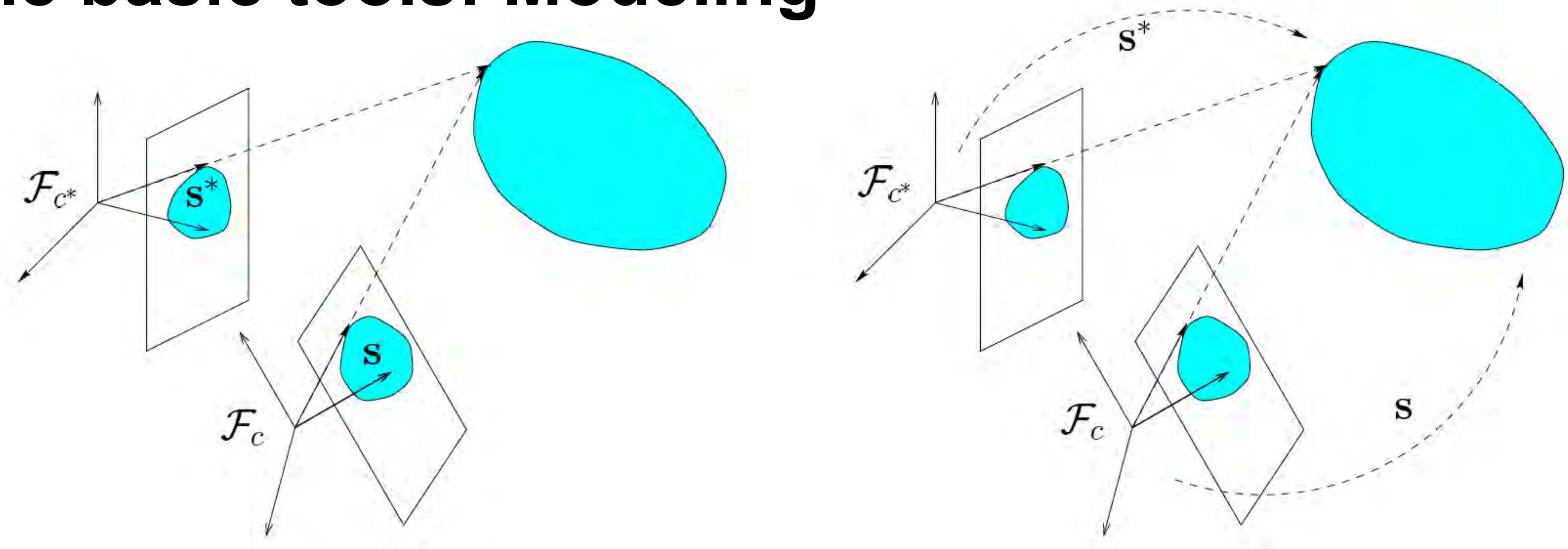
Just need a camera



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

Just need a computer

The basic tools: Modeling



2D visual features (IBVS) / 3D visual features (PBVS)

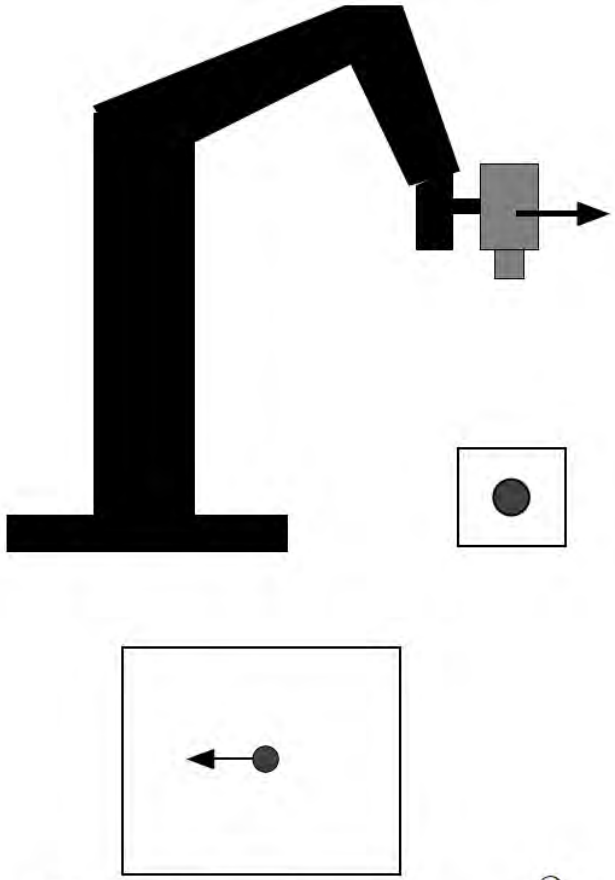
Same principle in both cases (but not same properties)

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

- $\mathbf{L_s}$: interaction matrix, $\mathbf{J_s}$: feature Jacobian
- $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega}) \in se_3$: instantaneous camera velocity in camera frame

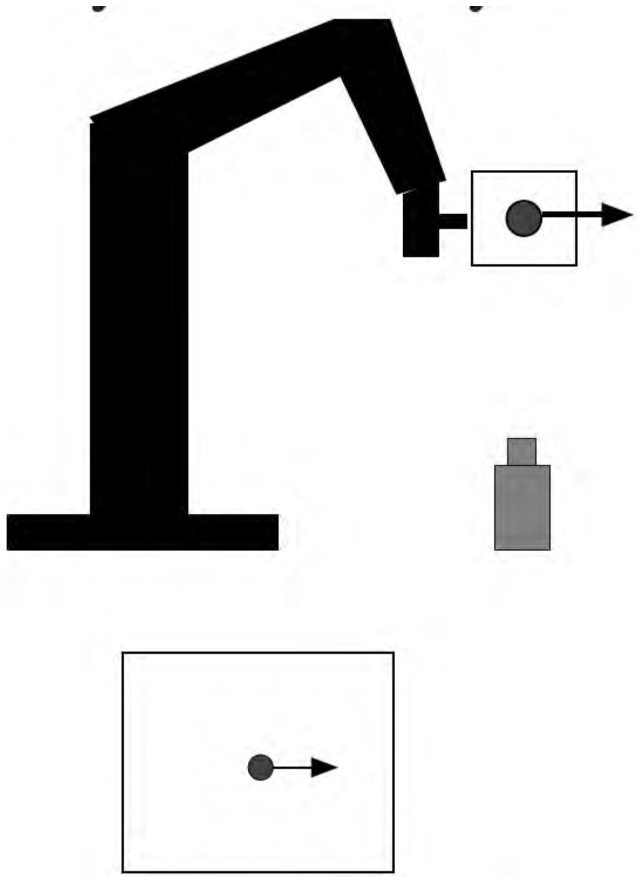
The basic tools: the feature Jacobian

Eye-in-hand system



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

Eye-to-hand system



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

The basic tools

Modeling: $\dot{\mathbf{s}} = \mathbf{L_s v}$

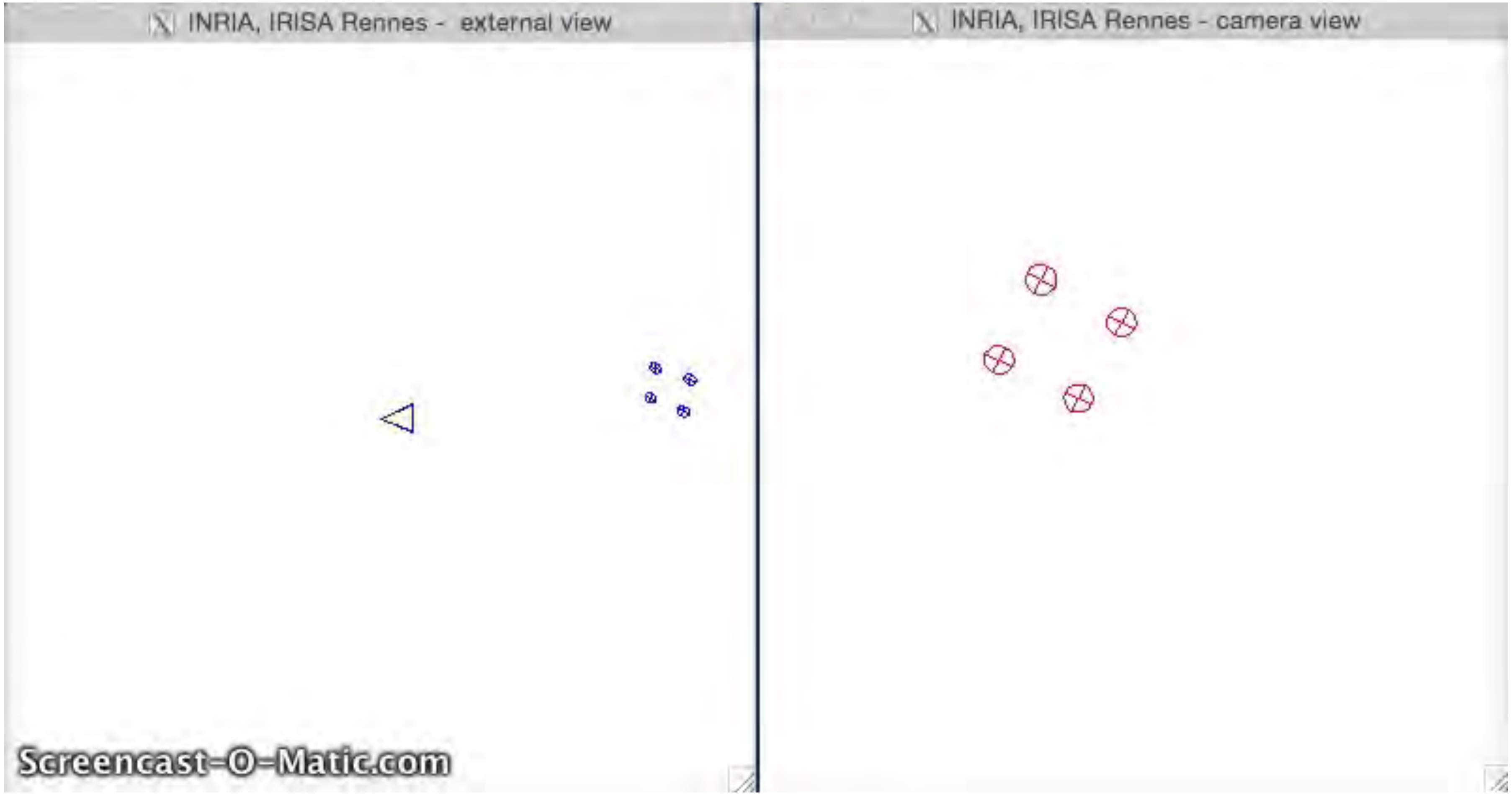
Control: $\mathbf{v} = -\lambda \, \widehat{\mathbf{L_s}}^+ (\mathbf{s} - \mathbf{s}^*)$ to try to ensure $\dot{\mathbf{s}} = -\lambda(\mathbf{s} - \mathbf{s}^*)$

(exponential decoupled decrease)

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

$\dot{\mathcal{L}} = -\lambda \, (\mathbf{s} - \mathbf{s}^*)^T \, \mathbf{L_s}\widehat{\mathbf{L_s}}^+ (\mathbf{s} - \mathbf{s}^*)$ Usually, LAS only

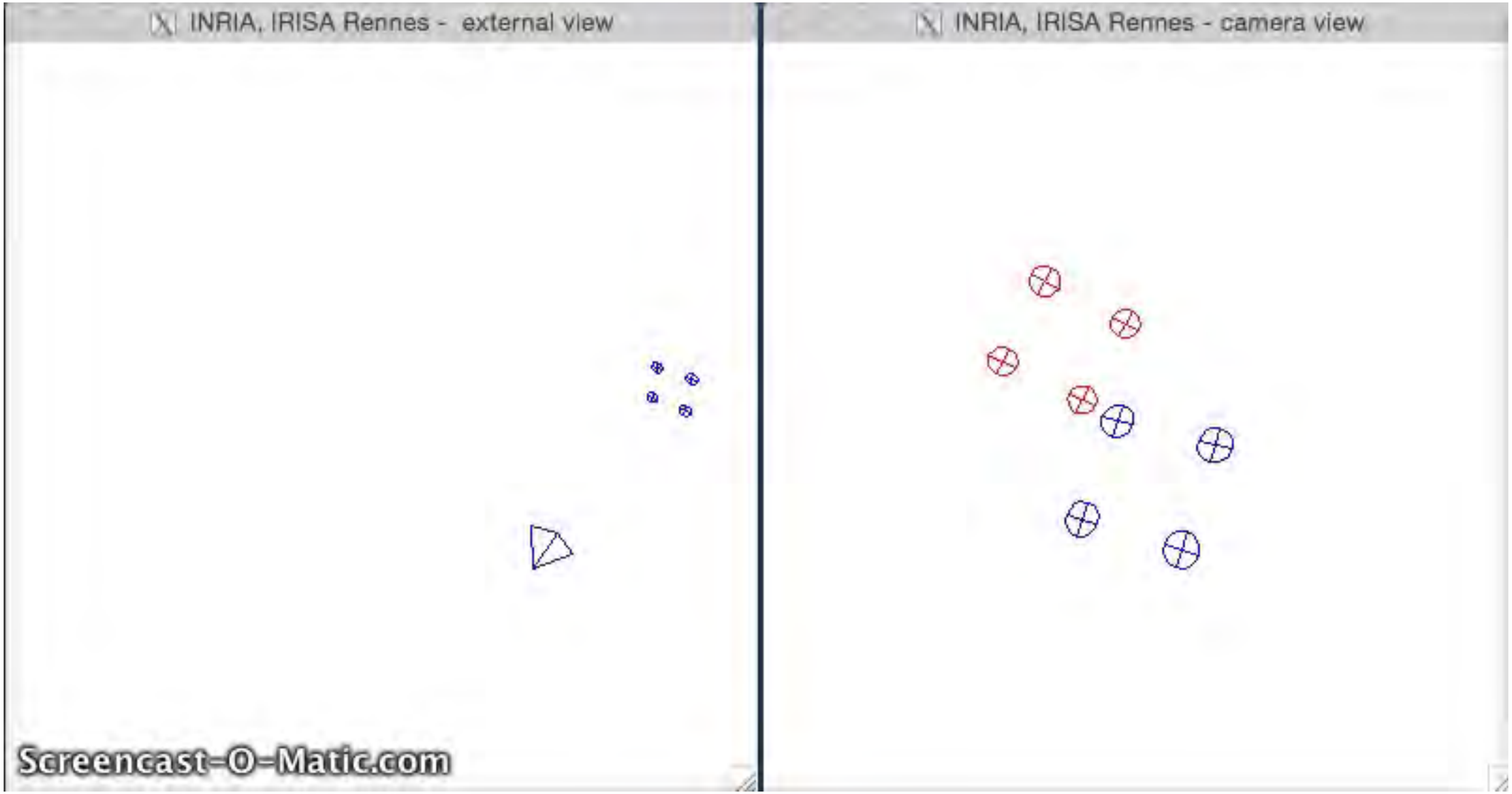
Usually LAS only: potential local minimum (for 6 dof)



$$\mathbf{s} = (x_1, y_1, ..., x_4, y_4) \qquad \mathbf{v} = -\lambda \mathbf{L}_{\mathbf{s}}^{+}(\mathbf{s} - \mathbf{s}^*)$$

This local minimum can be avoided with another choice of $\mathbf{s}$ or $\widehat{\mathbf{L}_{\mathbf{s}}}^{+}$

Usually LAS only: potential local minimum (for 6 dof)



$$\mathbf{s} = (x_1, y_1, ..., x_4, y_4) \qquad \mathbf{v} = -\lambda \mathbf{L}_{\mathbf{s}^*}^{+} (\mathbf{s} - \mathbf{s}^*)$$

Local minimum avoided by using $\widehat{\mathbf{L}_\mathbf{s}}^{+} = \mathbf{L}_{\mathbf{s}^*}^{+}$ (very coarse approximation)

The basic tools

Modeling: $\dot{\mathbf{s}} = \mathbf{L_s} \mathbf{v}$

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

For an image point:

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

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

- Can be approximated: $Z_i(t) = Z_i^*$
- Can be estimated: $Z_i(t) = \widehat{Z_i}(t)$
  - by triangulation with stereovision
  - from pose if 3D object model available
  - up to a scale factor from epipolar geometry/homography with current & desired images
  - from structure from known motion

The basic tools

Modeling: $\dot{\mathbf{s}} = \mathbf{L_s v}$

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

For an image point:

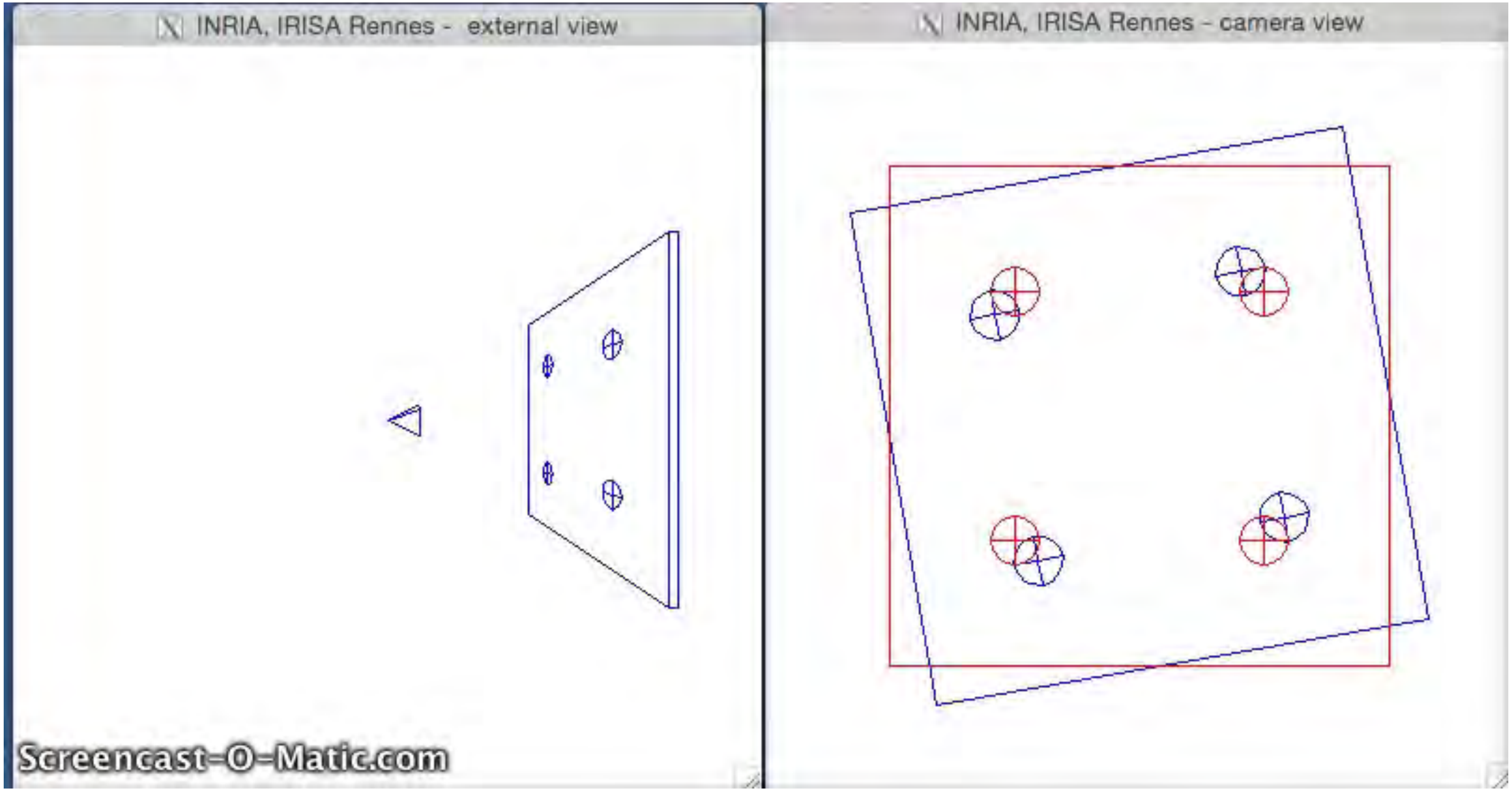
$$\mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}$$

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

Different choices of $\mathbf{s}$ will induce different image & robot behaviors

One open problem for 6 dof (solved for 4 dofs)

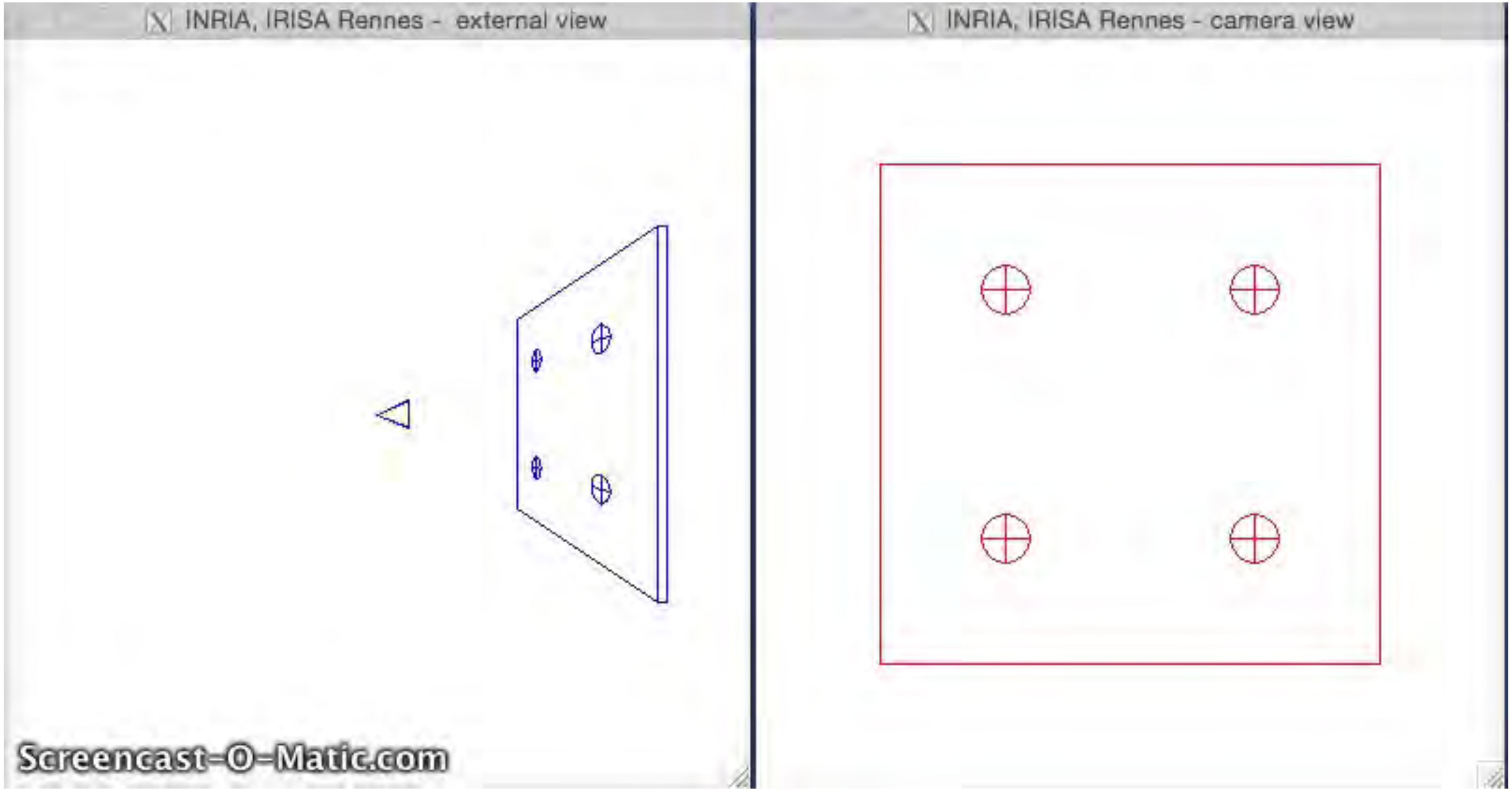
What are the visual features for an optimal behavior?



$$\mathbf{s} = (x_1, y_1, ..., x_4, y_4)$$

What are the good visual features?

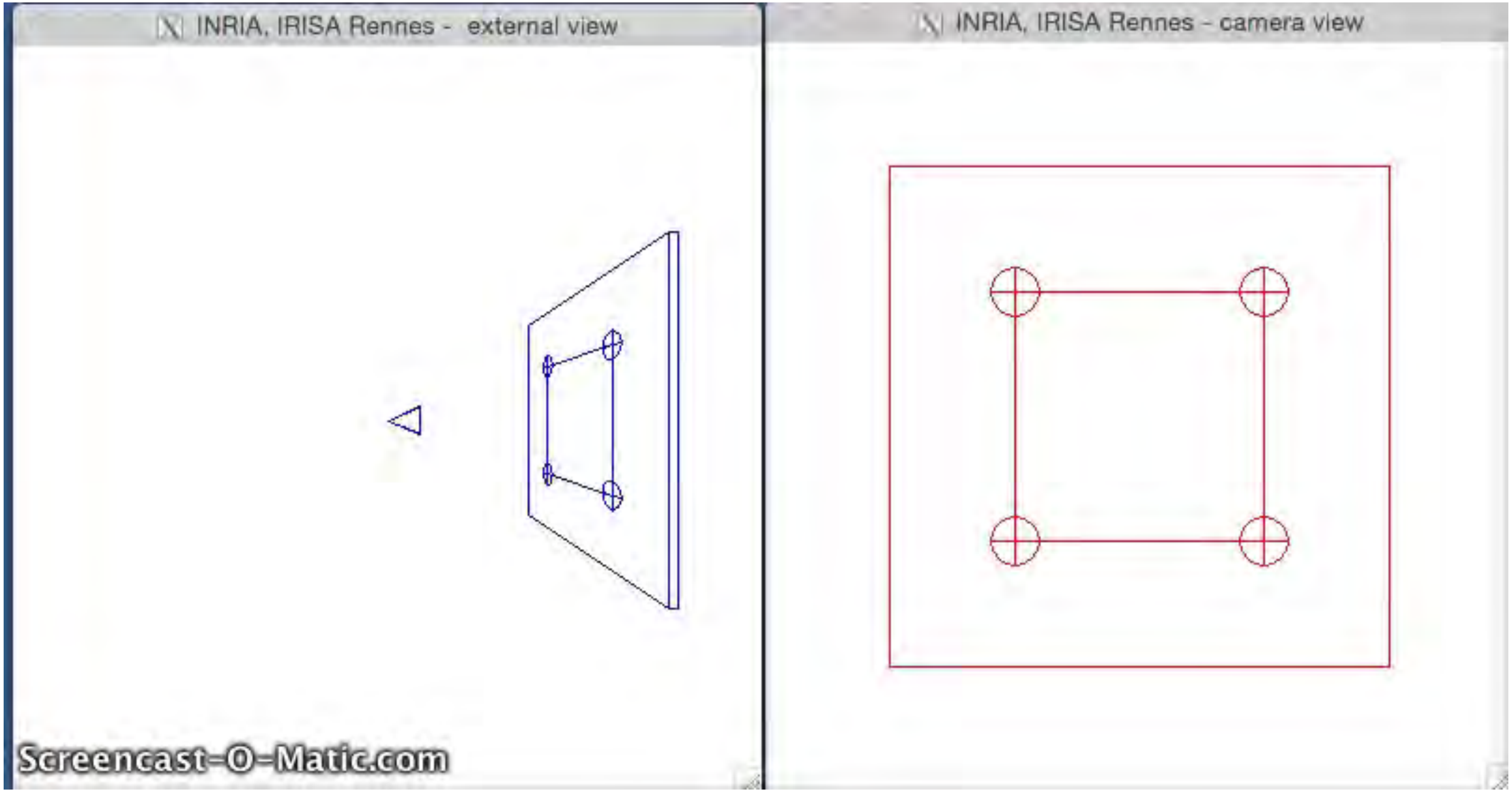
A very bad choice



$$\mathbf{s} = (x_1, y_1, ..., x_4, y_4)$$

What are the good visual features?

A perfect choice for this particular configuration



$$\mathbf{s} = (\rho_1, \theta_1, ..., \rho_4, \theta_4)$$

The basic tools

Modeling: $\dot{\mathbf{s}} = \mathbf{L_s} \mathbf{v}$

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

$\mathbf{L_s}$ known for many visual features:

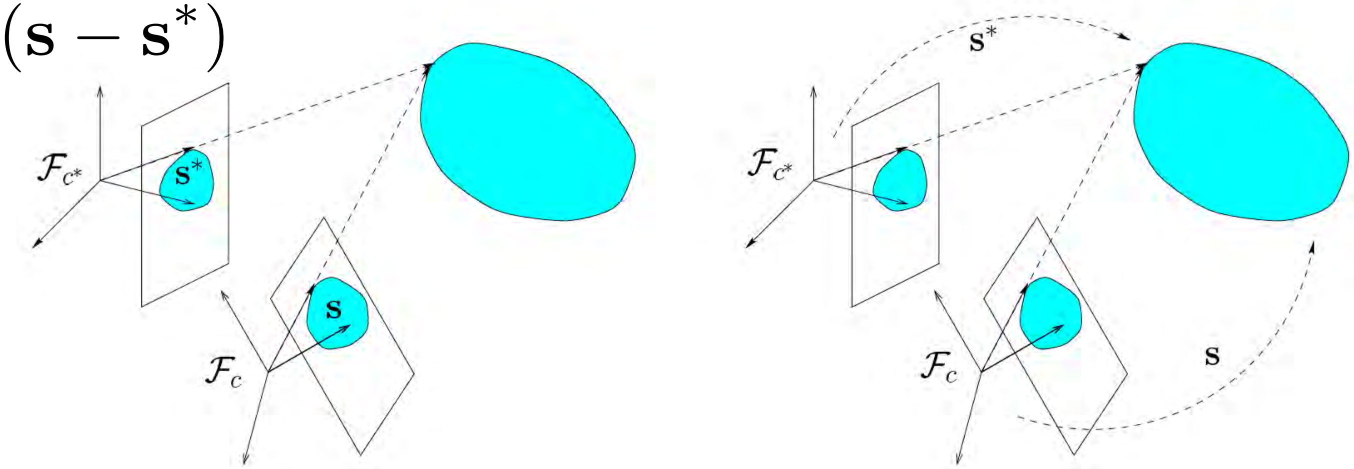
- In 3D, directly from kinematics: ${}^{c}\mathbf{t}_o,\ {}^{c^*}\mathbf{t}_c,\ \theta\mathbf{u}$: GAS if 3D is perfect
- In 2D:
  - Point, segment, straight line, circle, cylinder, sphere, …
  - Moments for planar or almost planar shapes

If $\mathbf{L_s}$ unknown, it can be estimated (off-line, on-line, by learning) but be careful to non-linearity and stability

From your application (robot dof, object, task), search for the best choice

My 2 cents on the endless debate: IBVS vs PBVS

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



2D visual features (IBVS) / 3D visual features (PBVS)

For IBVS, 3D appears in $\widehat{\mathbf{L}_{\mathbf{s}}}$ but not in $\mathbf{s}$

So 3D noise will affect the transient, but not the accuracy at the goal

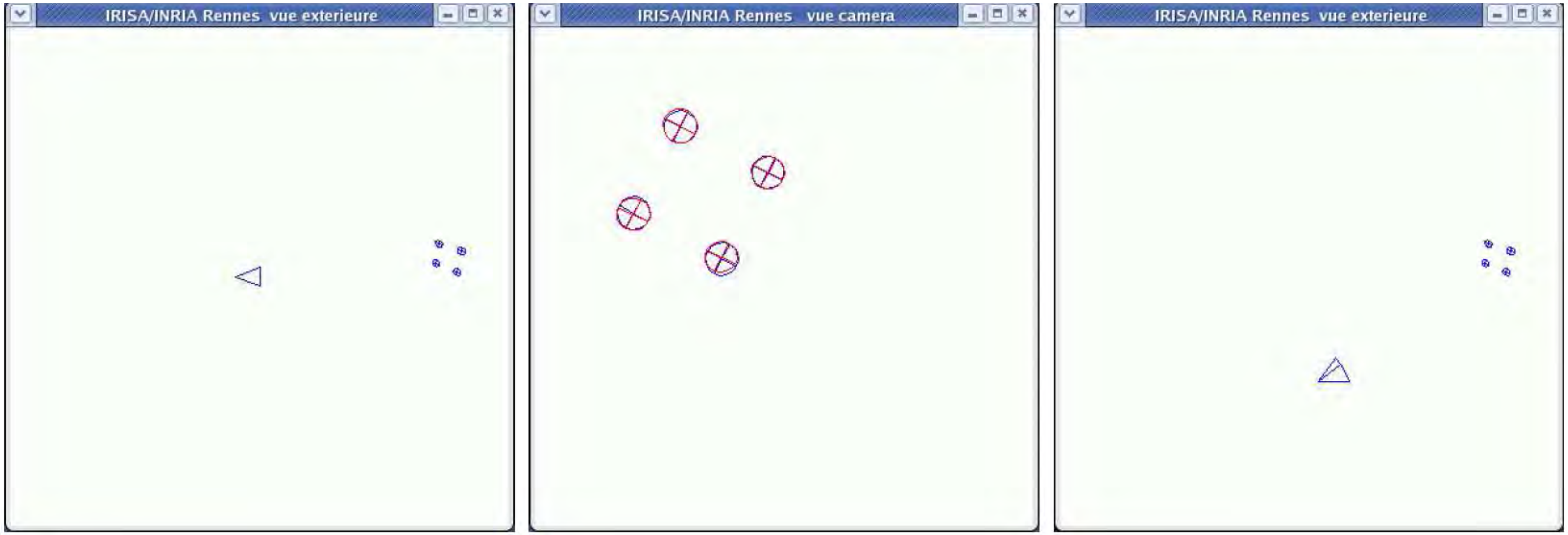
This is not the case for PBVS

Pose estimation may be unstable

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

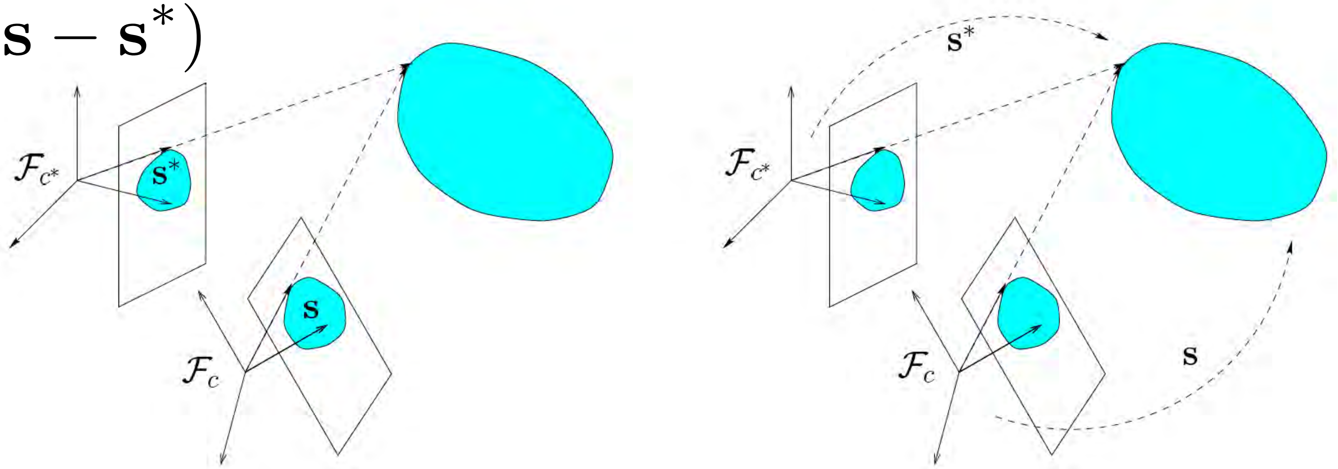
$$\Rightarrow \quad \dot{\hat{\mathbf{p}}}(t) = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}} \mathbf{L_x} \mathbf{v} \quad \Rightarrow \quad \mathbf{L}_{\hat{\mathbf{p}}} = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}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

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



My 2 cents on the endless debate: IBVS vs PBVS

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



2D visual features (IBVS) / 3D visual features (PBVS)

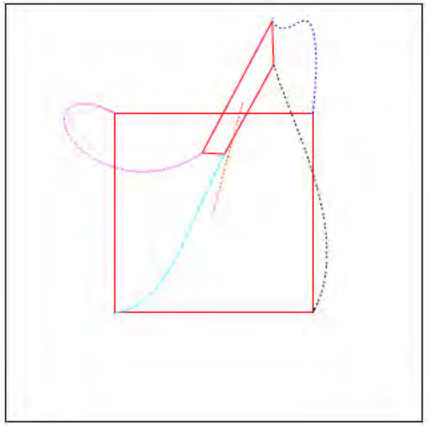
For IBVS, 3D appears in $\widehat{\mathbf{L}_{\mathbf{s}}}$ but not in $\mathbf{s}$

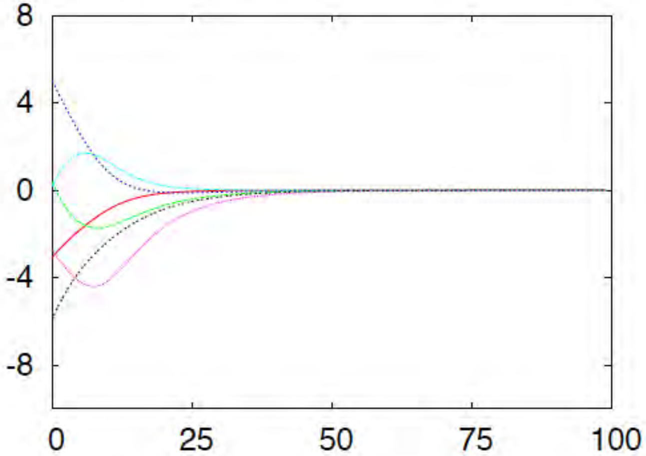
So 3D noise will affect the transient, but not the accuracy at the goal

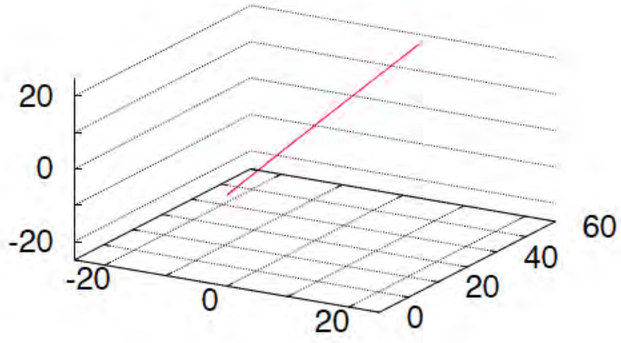
This is not the case for PBVS

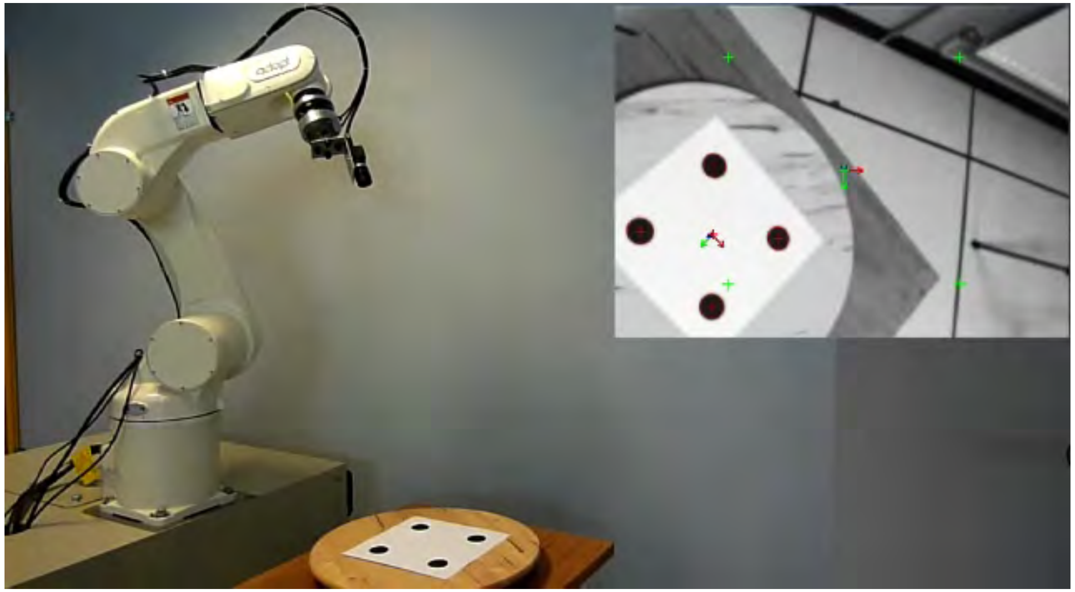
And the winner was to combine 2D and 3D visual features (2 ½ D VS)…

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



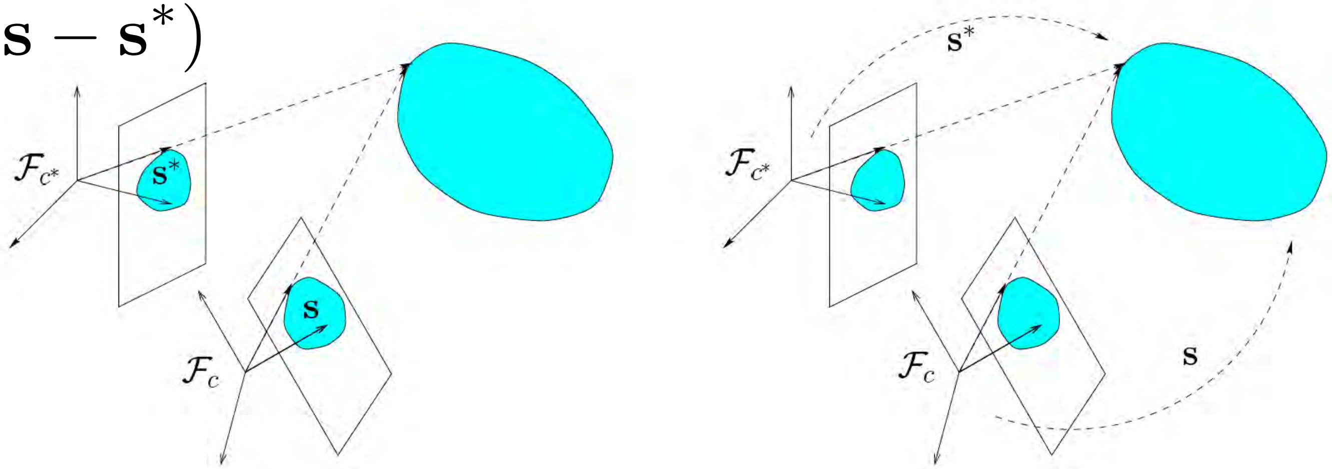






My 2 cents on the endless debate: IBVS vs PBVS

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



2D visual features (IBVS) / 3D visual features (PBVS)

For IBVS, 3D appears in $\widehat{\mathbf{L}_{\mathbf{s}}}$ but not in $\mathbf{s}$

So 3D noise will affect the transient, but not the accuracy at the goal

This is not the case for PBVS

And the winner was to combine 2D and 3D visual features (2 ½ D VS)…

Now, try to design IBVS with PBVS behavior (search for $\mathbf{s}$ such that $\mathbf{L}_{\mathbf{s}} \approx \mathbf{I}$)

A new family of visual servoing: photometric VS

Remove the image processing part in the usual steps:

- extract and track visual measurements near video rate
- design visual features and control schemes from the available measurements

Advantages:

- Robustness to image processing errors and noise!

Photometric visual servoing

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

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

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

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

Drawbacks: small convergence domain, strange robot trajectory

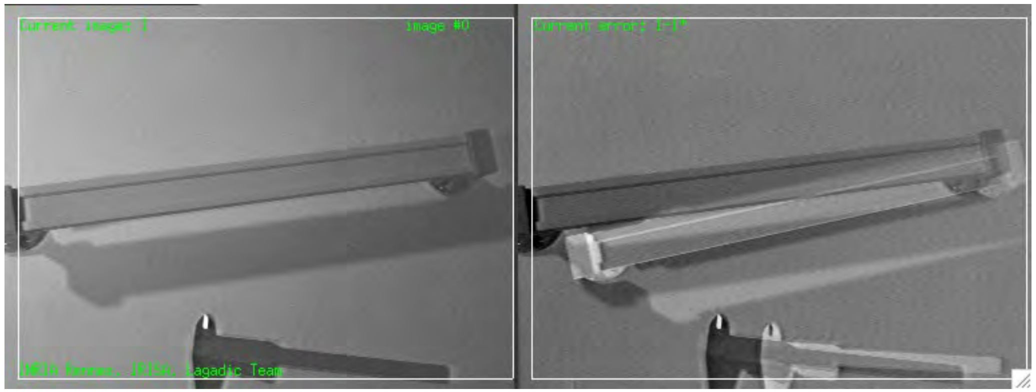
But no feature extraction, tracking nor matching

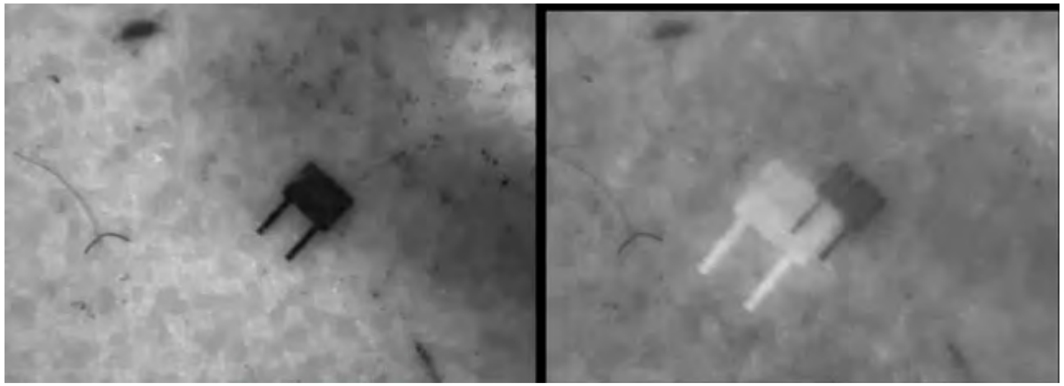
+ excellent positioning accuracy

Photometric visual servoing

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

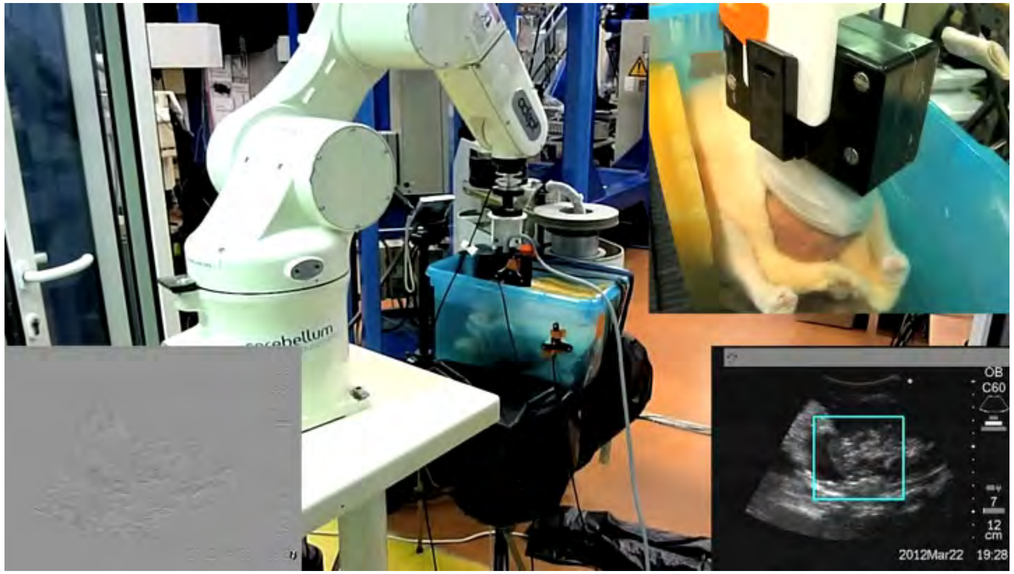
Robustness to outliers (occlusion) by using $\mathbf{s} = \rho_{\mathbf{I}}\ \mathbf{I}$



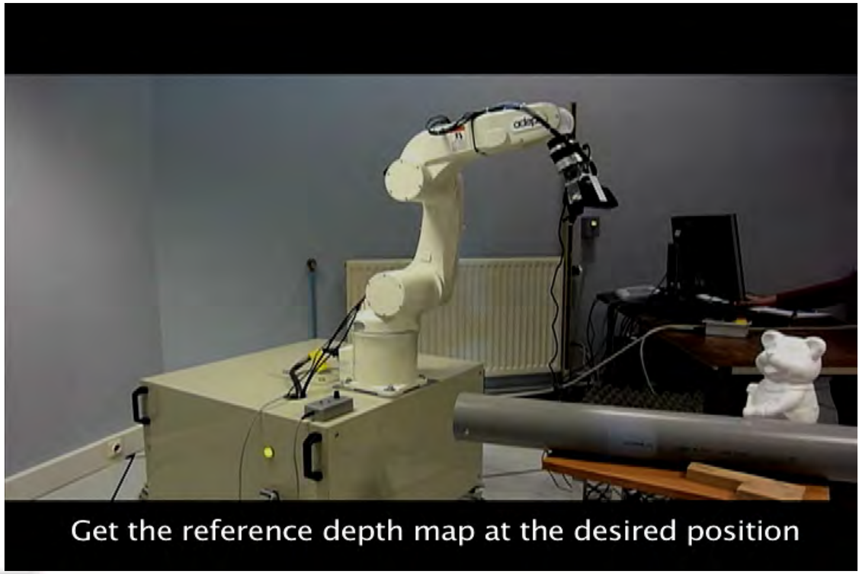


Accuracy < 0.1 µm

Similar on ultrasound images



Similar on depth map from RGB-D sensor: $\mathbf{s} = \rho_{\mathbf{Z}}\, \mathbf{Z}$



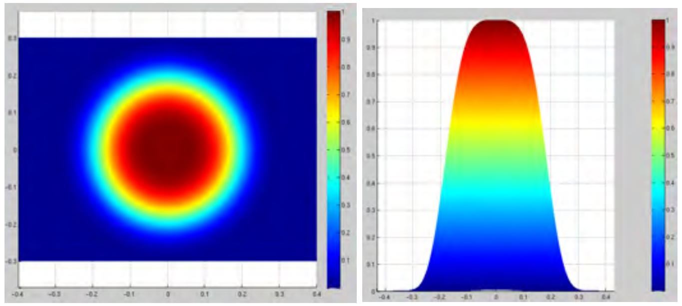
In the same spirit:

- RGB components, spatial gradient image
- Sum of conditional variance: $\mathcal{L} = \|\mathbf{I}(\mathbf{x}) - \widehat{\mathbf{I}}(\mathbf{x})\|$ with $\widehat{\mathbf{I}}(\mathbf{x}) = \epsilon(\mathbf{I}(\mathbf{x}), \mathbf{I}^*(\mathbf{x}))$
- Maximize mutual information between current and desired image
- Histogram-based visual servoing
- Mixture of Gaussian
- Wavelet
- …

Photometric moments

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

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



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

$\mathbf{I}^*$

$\mathbf{I}$

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

Other/open issues

- Target tracking
  - PI controller
  - Estimate, predict and compensate the target motion (feed forward)

- Consider constraints:
  - visibility, occlusion, obstacles
  - joint limits, singularities
  - dynamics: non holonomy, under-actuation
    - Path planning in the image, optimal control, MPC
    - Redundancy, task sequencing, stack of tasks

- Multi sensor-based control
  - Modeling, fusion

To go further

- F. Chaumette, S. Hutchinson, P. Corke: Visual servoing, in Chapter 34 of Handbook of Robotics, 2nd edition, expected for IROS'2016.
- Many papers in the field
- Do not hesitate to use ViSP for visual tracking and visual servoing:

visp.inria.fr

Thanks for your attention

Acknowledgments: Lagadic colleagues and the VS worlwide community