Visual Servoing

Charlie Kemp

4632B/8803
Mobile Manipulation
Lecture 8

From: http://www.hsi.gatech.edu/visitors/maps/

From: http://www.hsi.gatech.edu/visitors/maps/

4^{th} floor
4100Q
M Building
167

First office
on HSI side

Find the KUKA cup.

Manipulate the KUKA cup.

Overview

- Manipulation
- Global approach
- Local approach
- Visual Servoing

Manipulating a rigid body

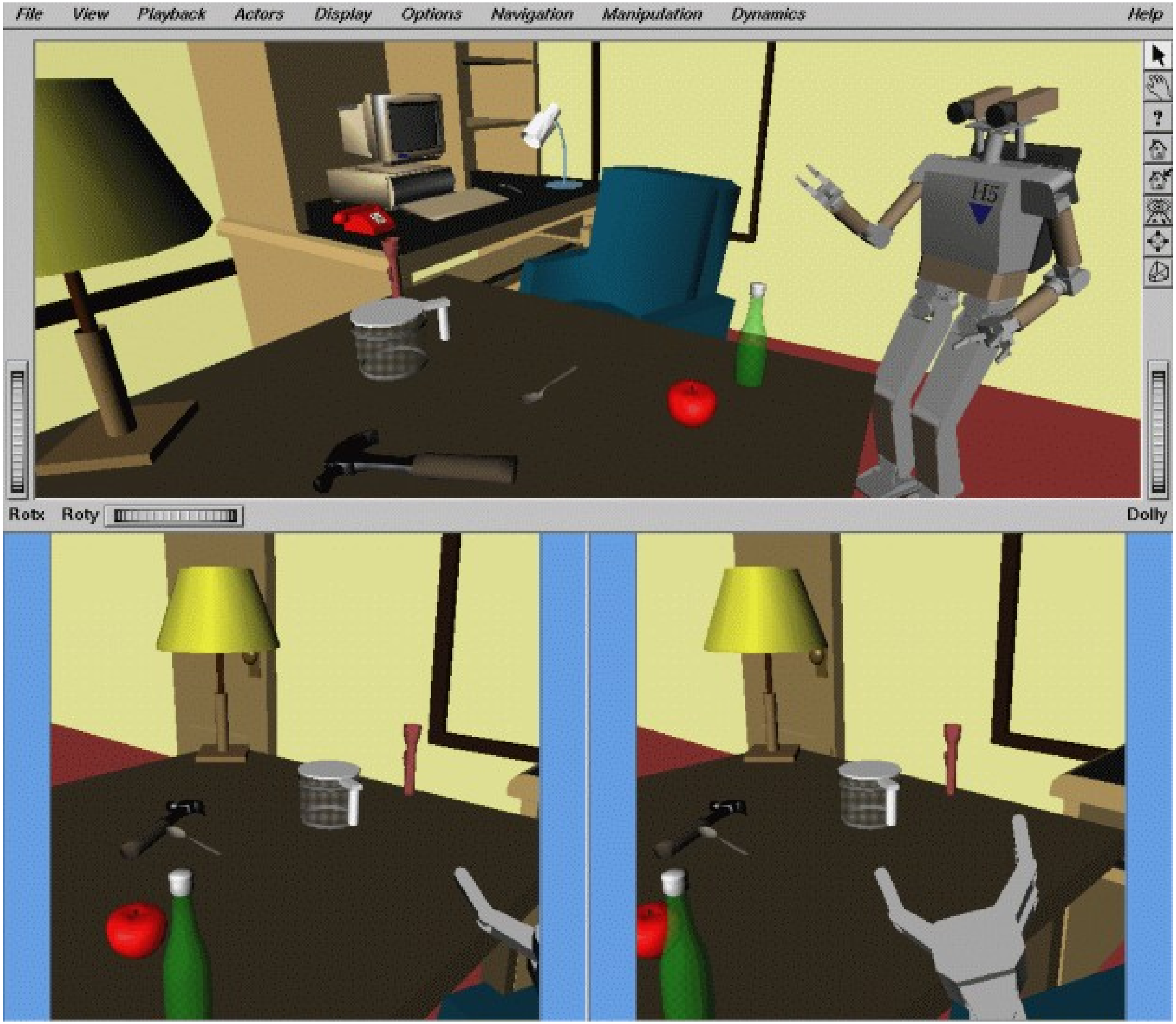
- Grasping it
- Transporting it
- Placing it

Rigid body transformations

Initial pose $\Rightarrow$ Final pose

We have a generative model!

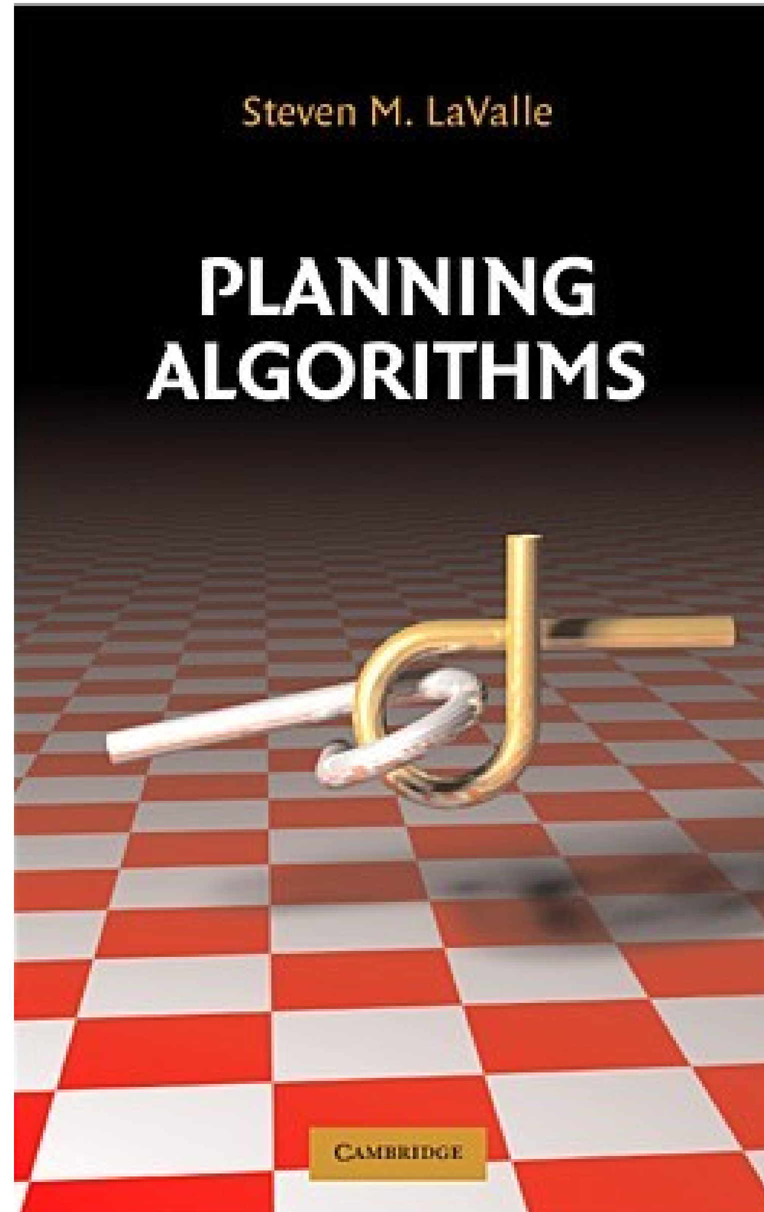
Motion planning
(search for an entire trajectory)



From: James Kuffner's home page (CMU)
http://www.kuffner.org/james/humanoid/pics/mainWindowSnap.gif

Motion planning

- Global solutions
- Drawbacks
  - Difficult to move from simulation to the real world
  - Sensing rarely included
  - Usually assume known state
  - Usually assume well-modeled transitions between states
  - May not meet real-time constraints



From: Steve LaValle's home page at UIUC (http://msl.cs.uiuc.edu/~lavalle/)

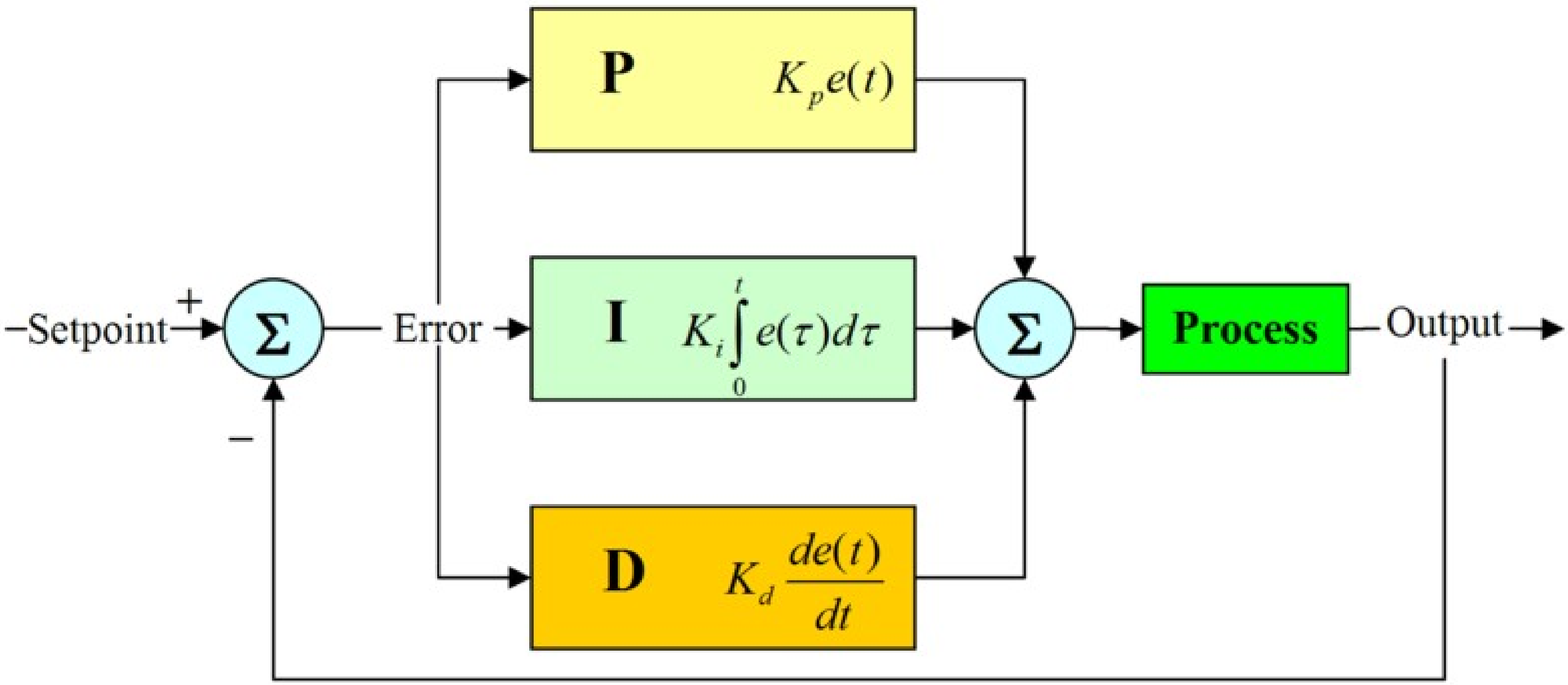
Can we find an efficient and robust local solution?

Rigid body transformations

Initial pose → Final pose

- Define an error function
- Locally minimize this error function
- Simple feedback control (PID)
- Not sensor specific
  - e.g., Rod Grupen's group at UMass Amherst

proportional-integral-derivative controller (PID controller)

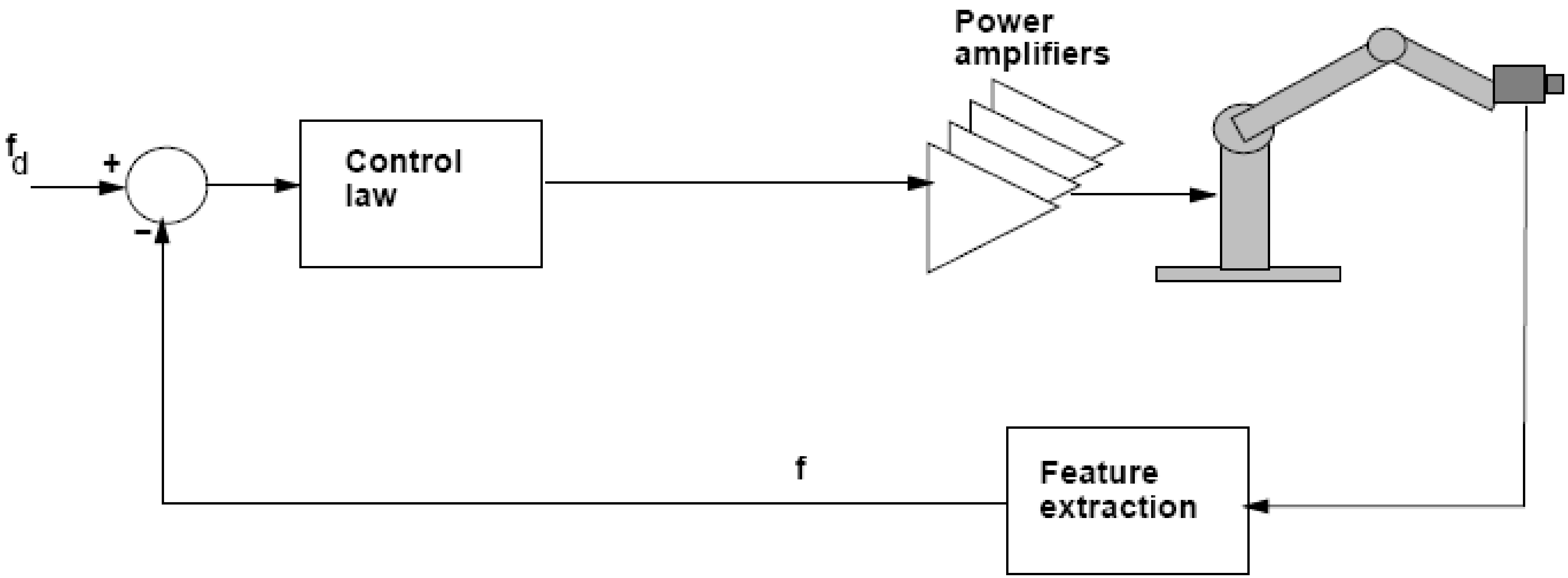


From: wikipedia article on PID control
http://en.wikipedia.org/wiki/Image:Pid-feedback-nct-int-correct.png

What poses?
Relative to what?

- Grasping it
- Transporting it
- Releasing it

Visual Servoing



From: Hutchinson et. al., *A Tutorial on Visual Servo Control,*
IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.

Visual servoing

S. Hutchinson, G. Hager, and P. Corke,
A tutorial on visual servo control,
IEEE Transactions on Robotics and Automation,
vol. 12, pp. 651–670, October 1996.

Examples

http://people.csail.mit.edu/cckemp/publications.shtml

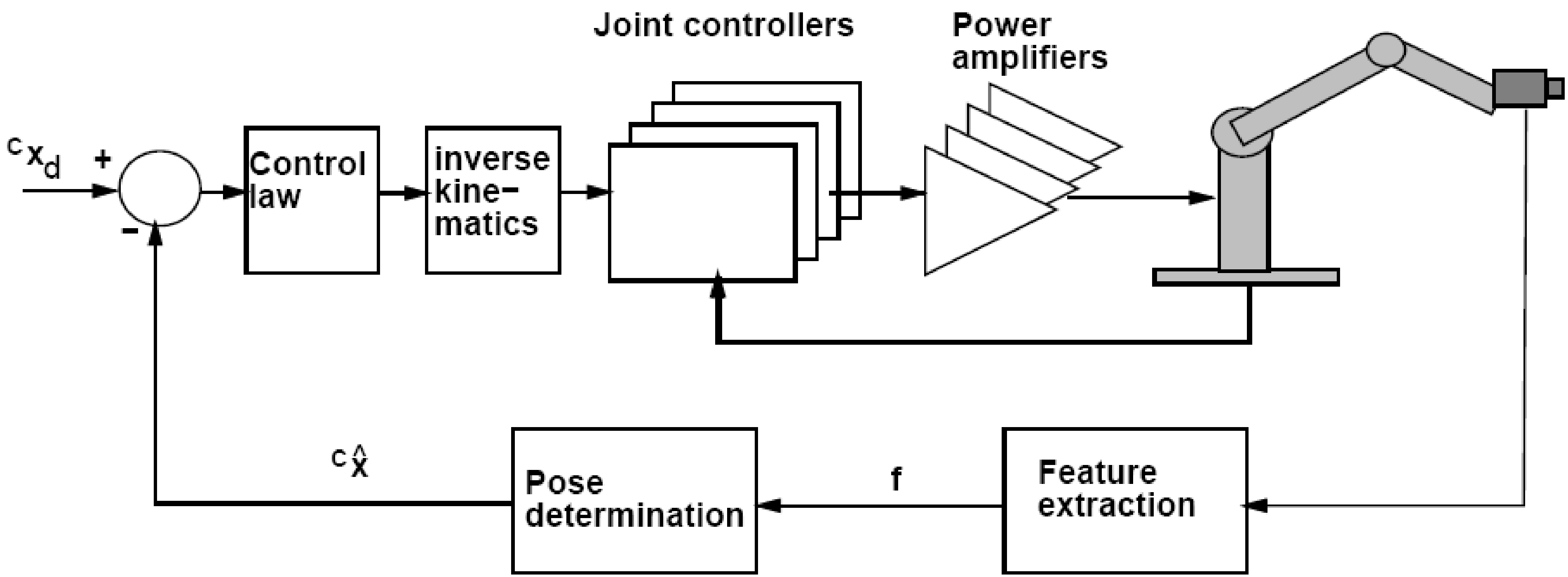
Examples

http://people.csail.mit.edu/cckemp/publications.shtml

Major options

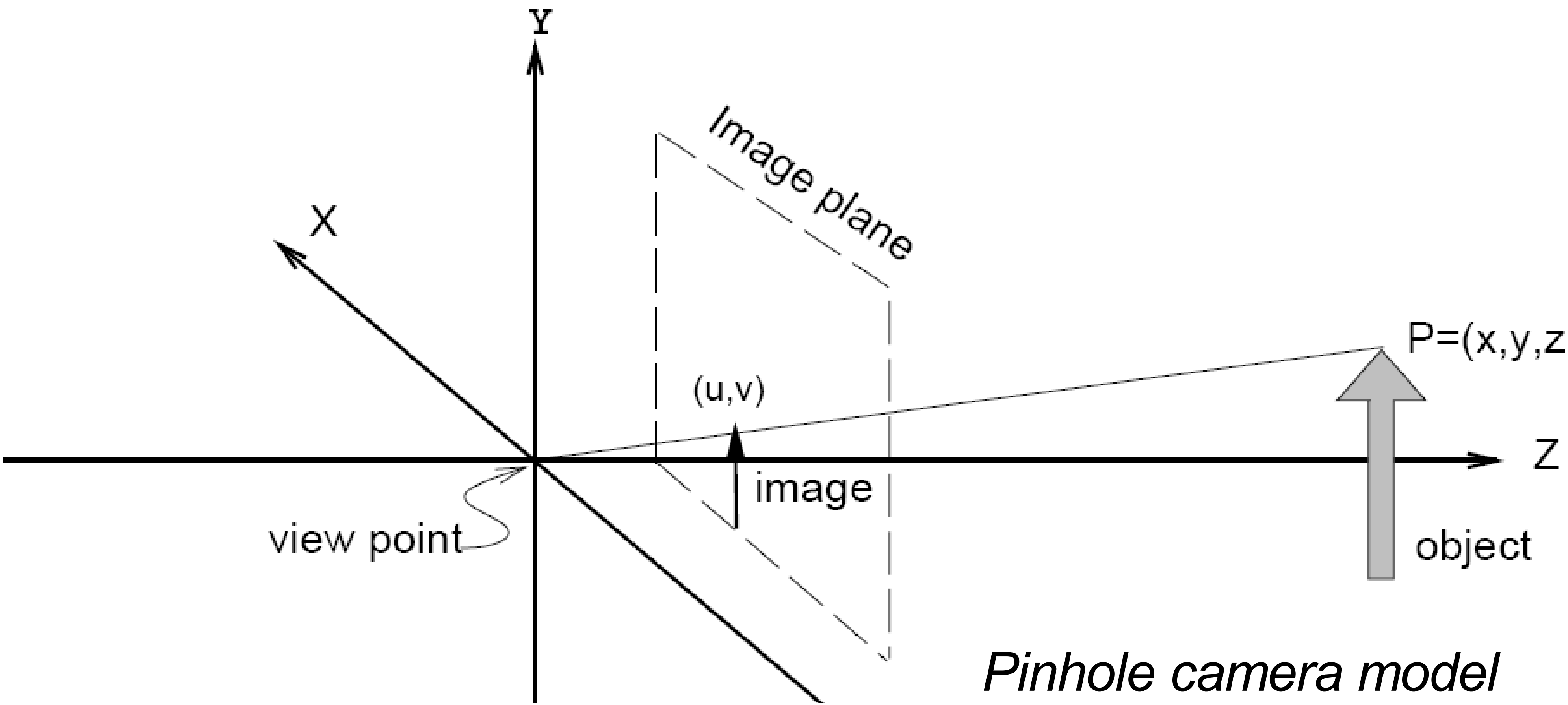
- Camera placement
  - End-effector mounted
    - Eye-in-hand
    - EOL (endpoint open-loop)
  - Fixed in the workspace
    - Might be PTZ
    - ECL (endpoint closed-loop)
- Error function
  - 3D pose
  - Image features
- Control hierarchy

Error function in 3D



From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*

Generative Model (Basic Image Formation)



Pinhole camera model

$$\pi(x,y,z) = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{\lambda}{z} \begin{bmatrix} x \\ y \end{bmatrix}$$

Perspective projection

From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*

Camera Calibration

- http://www.vision.caltech.edu/bouguetj/calib_doc/
- Estimates parameters
  - focal length, principal point, skew coefficient, distortions (radial and tangential)
- Rectify the images

Example
fixed camera, point to point

$$\mathbf{E}_{pp}(\mathbf{x}_e; \mathbf{S}, {}^e\mathbf{P}) = \mathbf{x}_e \circ {}^e\mathbf{P} - \mathbf{S}$$

$$\mathbf{u}_3 = -k E_{pp}(\hat{\mathbf{x}}_e; \hat{\mathbf{x}}_c \circ {}^c\widehat{\mathbf{S}}, {}^e\mathbf{P}) = -k \left(\mathbf{x}_e \circ {}^e\mathbf{P} - \hat{\mathbf{x}}_c \circ {}^c\widehat{\mathbf{S}} \right)$$

$\hat{\mathbf{x}}_e$, $\hat{\mathbf{x}}_c$ or ${}^c\widehat{\mathbf{S}}$

robot kinematics, camera calibration and visual reconstruction

From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*

Example
eye-in-hand, EOL

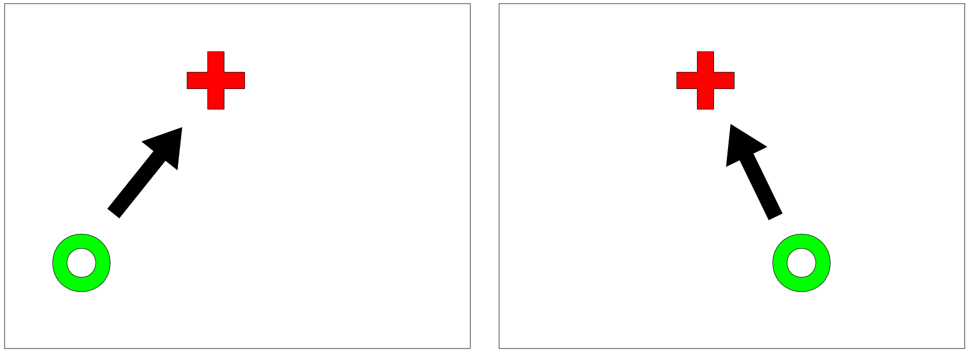
$$ {}^{e}\mathbf{E}_{pp}(\mathbf{x}_e; \mathbf{S}, {}^{e}\mathbf{P}) = {}^{e}\mathbf{P} - {}^{e}\mathbf{x}_0 \circ \mathbf{S} $$

$$ \widehat{\mathbf{S}} = \hat{\mathbf{x}}_e \circ {}^{e}\hat{\mathbf{x}}_c \circ {}^{c}\widehat{\mathbf{S}} $$

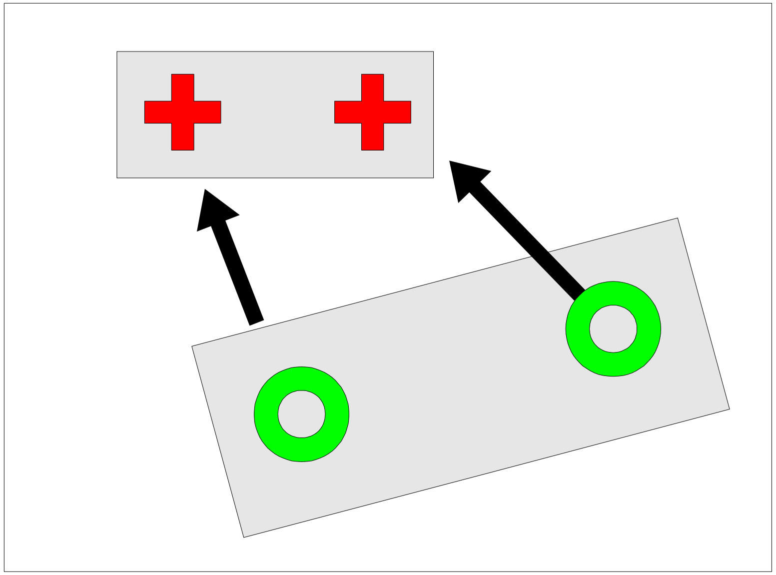
$$ \begin{aligned} {}^{e}\mathbf{u}_3 &= -k\ {}^{e}\mathbf{E}_{pp}(\hat{\mathbf{x}}_e; \hat{\mathbf{x}}_e \circ {}^{e}\hat{\mathbf{x}}_c \circ {}^{c}\widehat{\mathbf{S}}, {}^{e}\mathbf{P}) \\ &= -k({}^{e}\mathbf{P} - {}^{e}\mathbf{x}_0 \circ {}^{0}\hat{\mathbf{x}}_e \circ {}^{e}\hat{\mathbf{x}}_c \circ {}^{c}\widehat{\mathbf{S}}) = -k({}^{e}\mathbf{P} - {}^{e}\hat{\mathbf{x}}_c \circ {}^{c}\widehat{\mathbf{S}}) \end{aligned} $$

From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*

servo without reconstruction
(error as a function of image features)



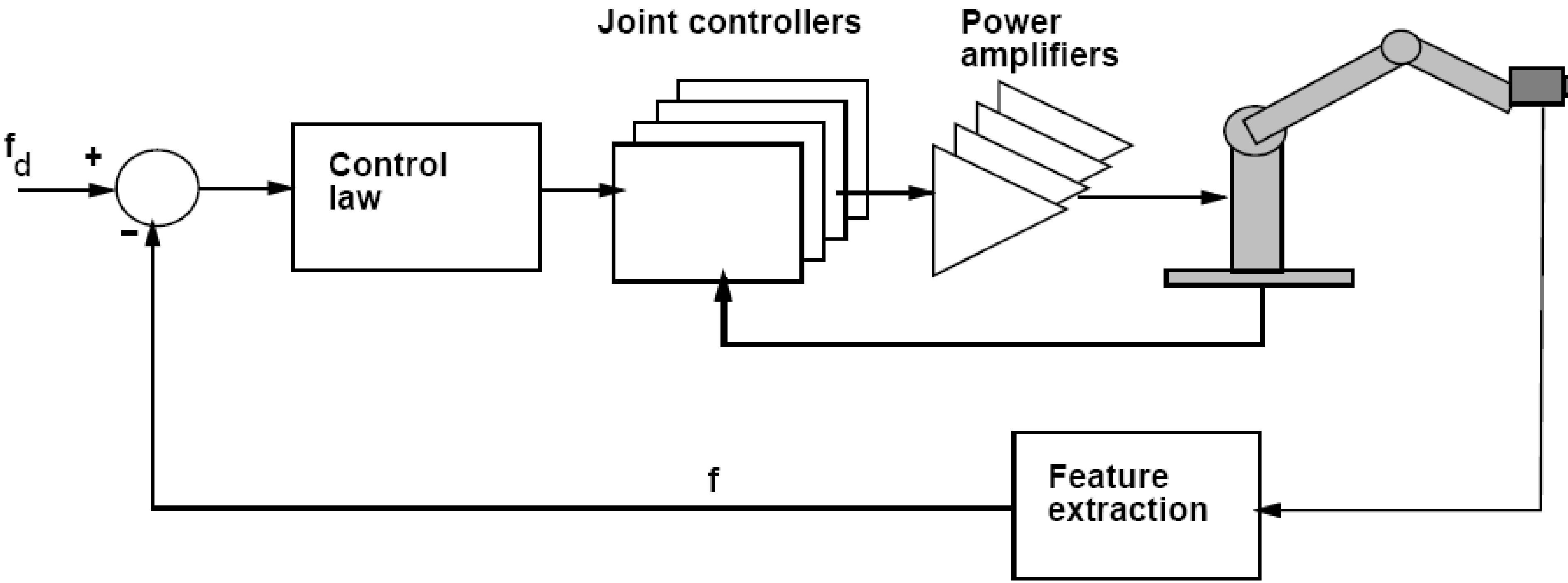
Multi-camera



Single camera

Error measurements imply 3D

feature based servoing



From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*

Image Jacobian

$$\dot{\mathbf{f}} = \mathbf{J}_v \dot{\mathbf{r}}$$

$$J_v(\mathbf{r}) = \left[\frac{\partial}{\partial \mathbf{r}}\right] = \begin{bmatrix} \frac{\partial v_1(\mathbf{r})}{\partial r_1} & \cdots & \frac{\partial v_1(\mathbf{r})}{\partial r_m} \\ \vdots & & \vdots \\ \frac{\partial v_k(\mathbf{r})}{\partial r_1} & \cdots & \frac{\partial v_k(\mathbf{r})}{\partial r_m} \end{bmatrix}$$

From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*

Example Image Jacobian

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{z} & 0 & \frac{-u}{z} & \frac{-uv}{\lambda} & \frac{\lambda^2 + u^2}{\lambda} & -v \\ 0 & \frac{\lambda}{z} & \frac{-v}{z} & \frac{-\lambda^2 - v^2}{\lambda} & \frac{uv}{\lambda} & u \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*

Inverting the Image Jacobian

$$\dot{\mathbf{r}} = \mathbf{J}_v^{-1}\dot{\mathbf{f}}$$

$$\dot{\mathbf{r}} = \mathbf{J}_v^{+}\dot{\mathbf{f}} + (\mathbf{I} - \mathbf{J}_v^{+}\mathbf{J}_v)\mathbf{b}$$

Overconstrained

$$\mathbf{J}_v^{+} = (\mathbf{J}_v^{T}\mathbf{J}_v)^{-1}\mathbf{J}_v^{T}$$

$$\dot{\mathbf{r}} = \mathbf{J}_v^{+}\dot{\mathbf{f}}$$

Underconstrained

$$\mathbf{J}_v^{+} = \mathbf{J}_v^{T}(\mathbf{J}_v\mathbf{J}_v^{T})^{-1}$$

$$(\mathbf{I} - \mathbf{J}_v^{+}\mathbf{J}_v)\mathbf{b}$$

lie in the null space of $\mathbf{J}_v$

From: Hutchinson et. al., *A Tutorial on Visual Servo Control,*
IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.

Resolved rate motion control

$$\mathbf{e}(\mathbf{f}) = \mathbf{f}_d - \mathbf{f}$$

$$\mathbf{u} = \mathbf{J}_v^{-1}(\mathbf{r})\dot{\mathbf{f}}$$

$$\mathbf{u} = \mathbf{K}\mathbf{J}_v^{-1}(\mathbf{r})\mathbf{e}(\mathbf{f})$$

From: Hutchinson et. al., *A Tutorial on Visual Servo Control,*
IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.

servo without calibration

- Estimate the jacobian online

Estimation of the full Jacobian was solved mathematically by Broyden in the 60's [20], and later rediscovered in robotics by [11, 10, 15]. A first order updating formula which converges to the Jacobian after n linearly independent moves is:

$$\hat{J}_{k+1} = \hat{J}_k + \frac{(\Delta \mathbf{y}_{measured} - \hat{J}_k \mathbf{x})\mathbf{x}^T}{\mathbf{x}^T \mathbf{x}}$$

Note that this estimation accepts movements along arbitrary directions $\mathbf{x}$ and thus needs no additional data other than what is available as a part of the manipulation task we want to solve.

From: Jägersand M. Nelson R. Visual Space Task Specification, Planning and Control In *Proc. of IEEE Int. Symp. on Computer Vision 95*, p 521-526, 1995.

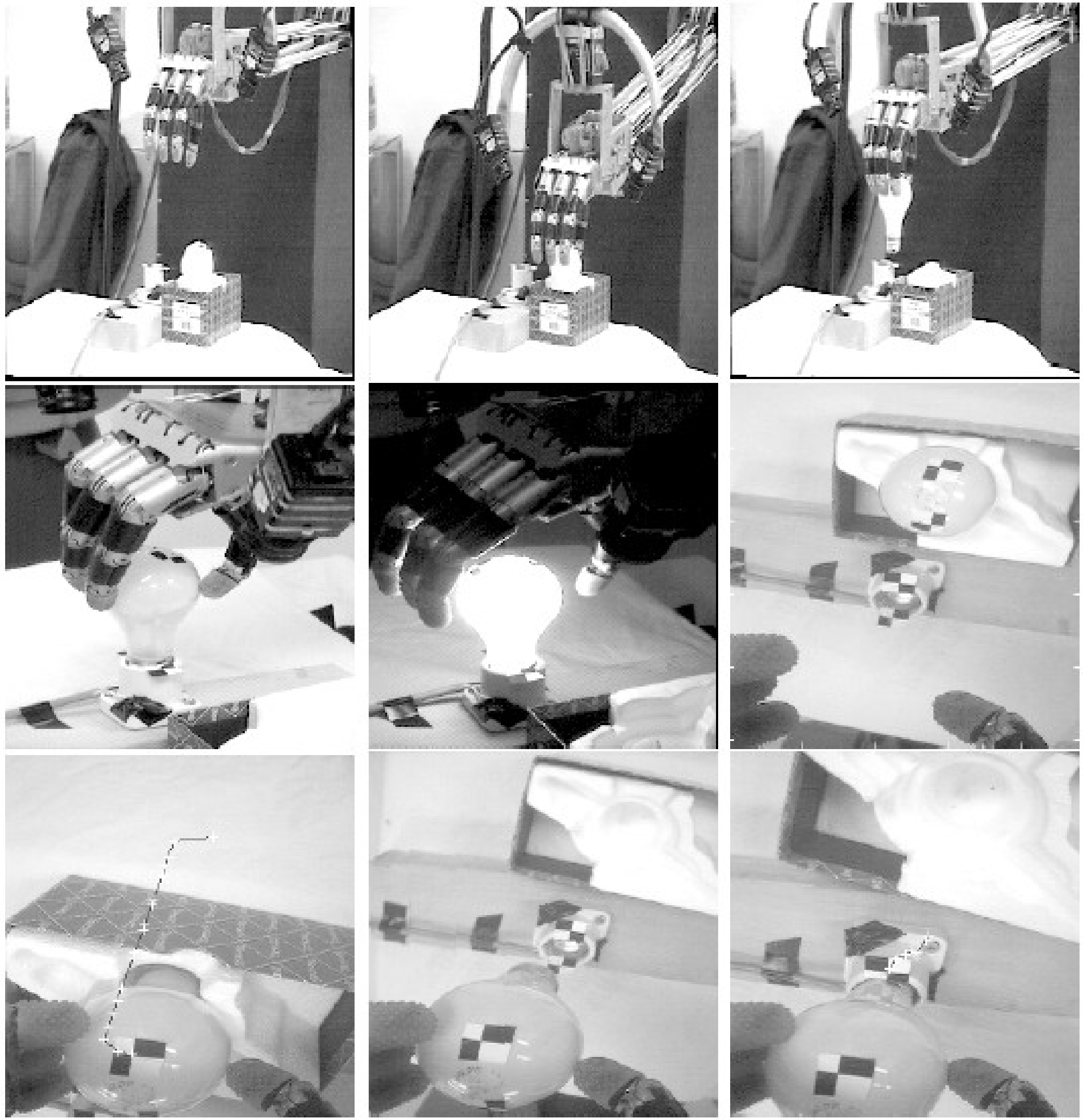
Martin Jägersand

University of Rochester -> Yale -> University of Alberta

Jägersand M. and Nelson R., Visual Space **Task Specification, Planning and Control**. *In Proc. of IEEE Int. Symp. on Computer Vision 95*, p 521-526, 1995.

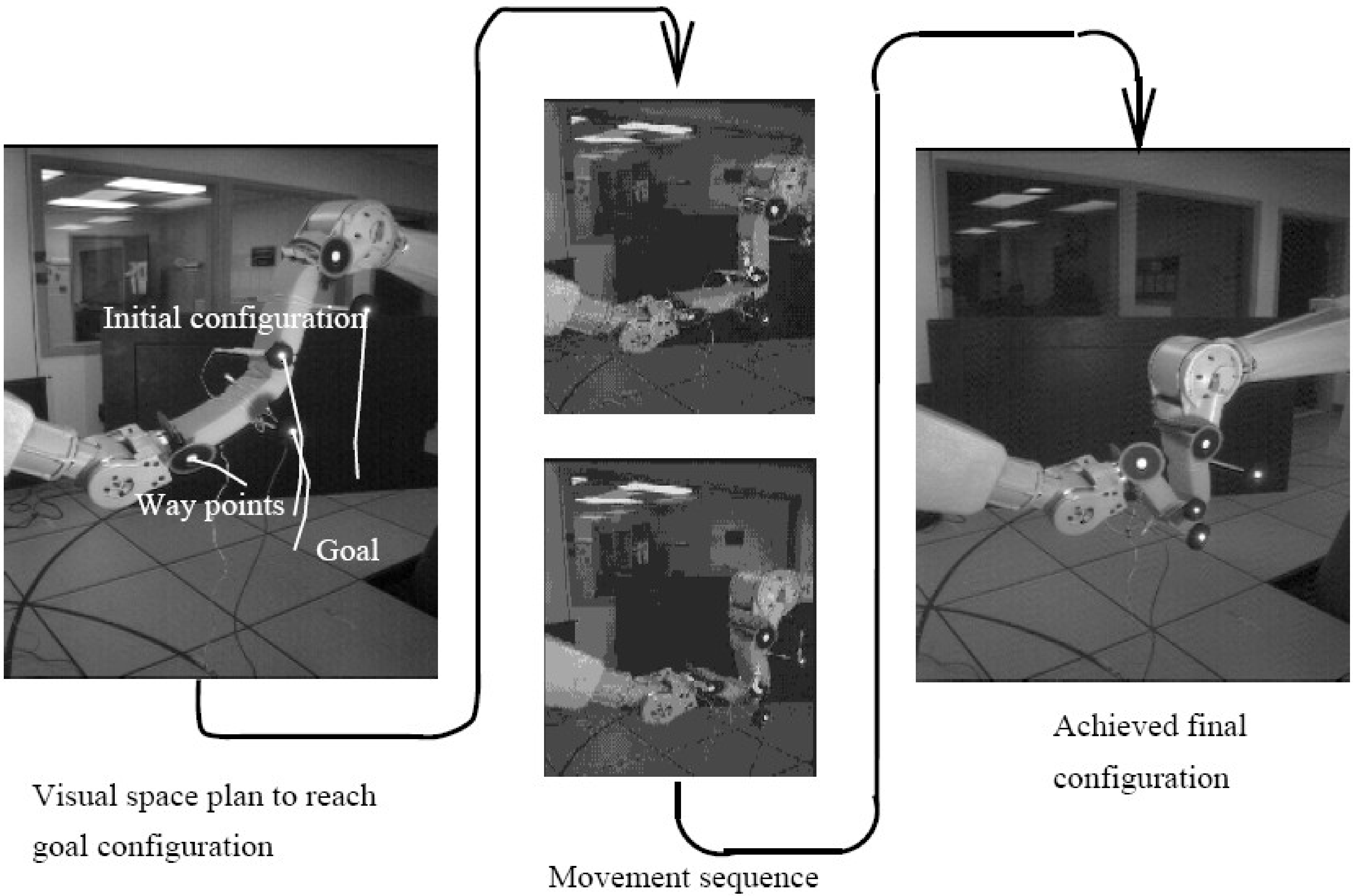
“A short description of higher level aspects of uncalibrated visual control. Many experiments solving complex manipulation tasks in unstructured environments.” - Martin Jägersand

From: http://www.cs.ualberta.ca/~jag/

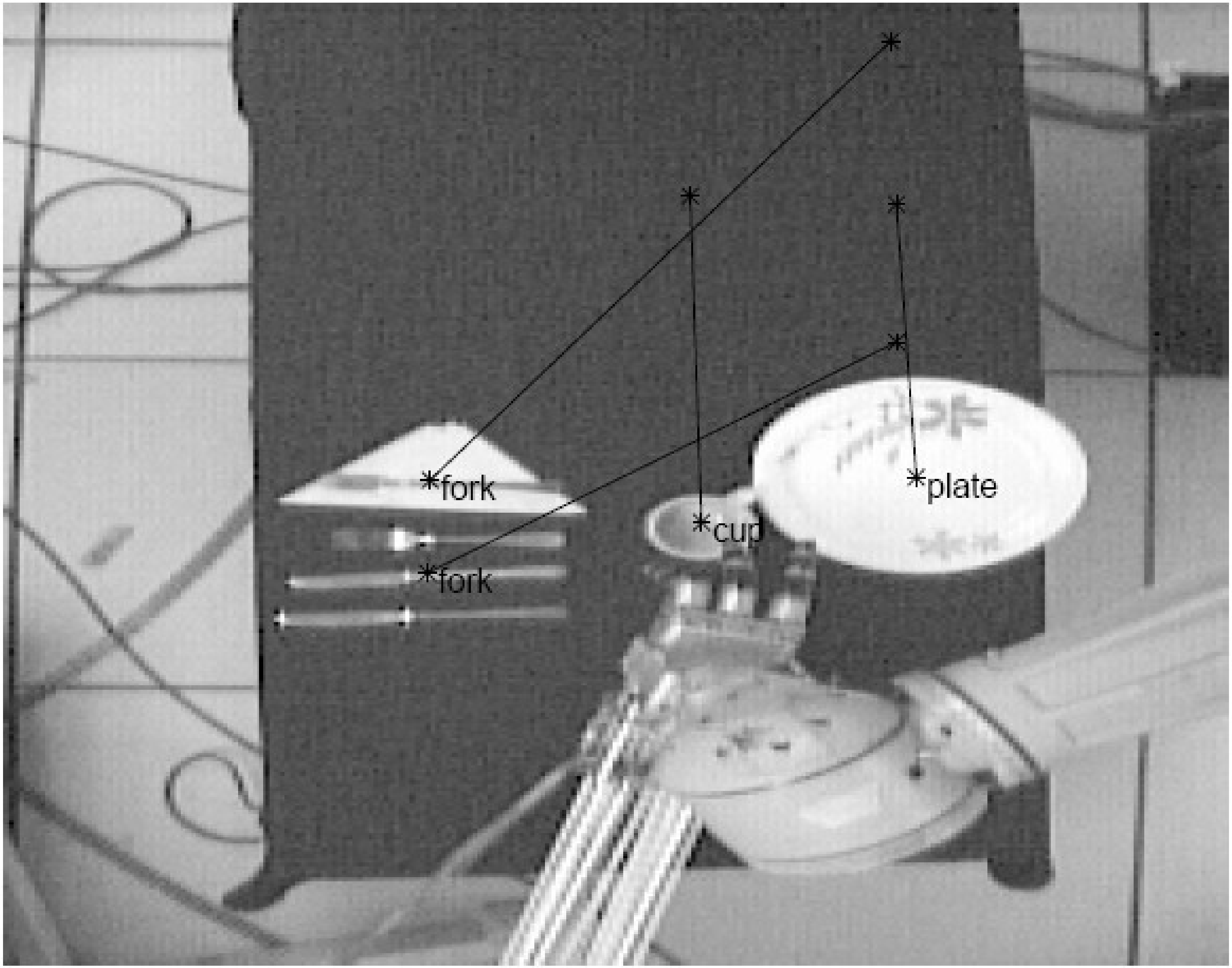


From: Jägersand M. Nelson R. Visual Space Task Specification, Planning and Control
In *Proc. of IEEE Int. Symp. on Computer Vision 95*, p 521-526, 1995.

Non-rigid Manipulation



From: Jägersand M. Nelson R. Visual Space Task Specification, Planning and Control In *Proc. of IEEE Int. Symp. on Computer Vision 95*, p 521-526, 1995.



From: Jägersand M. Nelson R. Visual Space Task Specification, Planning and Control
In *Proc. of IEEE Int. Symp. on Computer Vision 95*, p 521-526, 1995.

Discussion questions?

- When is visual servoing a good idea?
- When is visual servoing a bad idea?

What will you do?

Next week

- You're the presenters!

Extra

Velocity of a Rigid Object

$$\mathbf{T}(t) = [T_x(t), T_y(t), T_z(t)]^T$$

$$\Omega(t) = [\omega_x(t), \omega_y(t), \omega_z(t)]^T$$

$$\dot{\mathbf{P}} = \Omega \times \mathbf{P} + \mathbf{T}$$

$$\mathrm{sk}(\mathbf{P}) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

$$\dot{\mathbf{P}} = -\mathrm{sk}(P)\Omega + \mathbf{T}$$

$$\dot{\mathbf{r}} = \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Velocity screw

$$\dot{\mathbf{P}} = A(\mathbf{P})\dot{\mathbf{r}}$$

$$A(\mathbf{P}) = [I_3 \mid -\mathrm{sk}(\mathbf{P})]$$

From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*

Velocity of a Rigid Object (continued)

$$ {}^e\dot{\mathbf{r}} = [{}^e\mathbf{T}; {}^e\Omega] $$

$$ \dot{\mathbf{r}} = \begin{bmatrix} \Omega \\ \mathbf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_e {}^e\Omega \\ \mathbf{R}_e {}^e\mathbf{T} - {}^e\Omega \times \ \mathbf{t}_e \end{bmatrix} $$

$$ \dot{\mathbf{P}} = A(\mathbf{x}_e \circ {}^e\mathbf{P})\dot{\mathbf{r}} $$

From: Hutchinson et. al., *A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, October 1996.*