Visual Servoing

Charlie Kemp

4632B/8803
Mobile Manipulation
Lecture 8
4th floor
4100Q
M Building
167

First office on HSI side

From: http://www.hsi.gatech.edu/visitors/maps/
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 → 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 $\rightarrow$ 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)
What poses?
Relative to what?

• Grasping it
• Transporting it
• Releasing it
Visual Servoing

Visual servoing

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

http://people.csail.mit.edu/cckemp/publications.shtm
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,
Generative Model (Basic Image Formation)

Pinhole camera model

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

Perspective projection

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

\[ E_{pp}(x_e; S, {eP}) = x_e \circ {eP} - S \]

\[ u_3 = -k E_{pp}(\hat{x}_e; \hat{x}_c \circ \hat{cS}, {eP}) = -k \left( x_e \circ {eP} - \hat{x}_c \circ \hat{cS} \right) \]

\[ \hat{x}_e, \hat{x}_c \text{ or } \hat{cS} \]

robot kinematics, camera calibration and visual reconstruction

Example
eye-in-hand, EOL

\[ eE_{pp}(x_e; S, ^eP) = ^eP - ^e{x}_0 \circ S \]

\[ \hat{S} = \hat{x}_e \circ ^e\hat{x}_c \circ ^c\hat{S} \]

\[ ^e{u}_3 = -k \cdot eE_{pp}(\hat{x}_e; \hat{x}_e \circ ^e\hat{x}_c \circ ^c\hat{S}, ^eP) \]

\[ = -k \cdot (\hat{x}_0 \circ ^0\hat{x}_e \circ ^e\hat{x}_c \circ ^c\hat{S}) = -k \cdot (^eP \circ ^e\hat{x}_c \circ ^c\hat{S}) \]

From: Hutchinson et. al., A Tutorial on Visual Servo Control, 
servo without reconstruction
(error as a function of image features)

Multi-camera

Single camera

Error measurements imply 3D
feature based servoing

Image Jacobian

\[ \dot{\mathbf{f}} = \mathbf{J}_v \dot{\mathbf{r}} \]

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

Example Image Jacobian

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

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 \( \mathbf{J}_v \)

Resolved rate motion control

\[ e(f) = f_d - f \]

\[ u = J^{-1}_v(r) \dot{f} \]

\[ u = KJ^{-1}_v(r)e(f) \]

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:

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

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.


“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/
Non-rigid Manipulation

Initial configuration
Way points
Goal

Achieved final configuration

Visual space plan to reach goal configuration

Movement sequence

From: Jägersand M. Nelson R. Visual Space Task Specification, Planning and Control
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

\[ T(t) = [T_x(t), T_y(t), T_z(t)]^T \]
\[ \Omega(t) = [\omega_x(t), \omega_y(t), \omega_z(t)]^T \]

\[ \dot{\mathbf{P}} = \mathbf{\Omega} \times \mathbf{P} + \mathbf{T} \]

\[ \text{sk}(\mathbf{P}) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \]

\[ \dot{\mathbf{P}} = -\text{sk}(\mathbf{P})\mathbf{\Omega} + \mathbf{T} \]

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

Velocity screw

\[ \dot{\mathbf{P}} = A(\mathbf{P})\dot{\mathbf{r}} \]

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

Velocity of a Rigid Object (continued)

\[
\begin{align*}
e\dot{r} &= [eT; e \Omega] \\
\dot{r} &= \begin{bmatrix} \Omega \\ T \end{bmatrix} = \begin{bmatrix} R_e e\Omega \\ R_e eT - e\Omega \times t_e \end{bmatrix} \\
\dot{P} &= A(x_e \circ eP)\dot{r}
\end{align*}
\]