Visual Servoing

Henrik I. Christensen

Robotics and Intelligent Machines @ GT
College of Computing
Georgia Institute of Technology
Atlanta, GA
hic@cc.gatech.edu

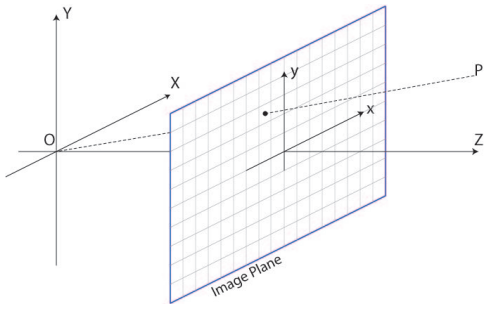
February 14, 2008

Outline

Introduction

- Recap of camera model
- Overview of strategies to visual servoing
- The basic theory behind models for visual servoing
- A few examples to illustrate use of the system.

Pin-Hole Model



Pin-Hole Model

- The relations are then:

$$\begin{aligned} \frac{x}{\lambda} &= \frac{X}{Z} \\ \frac{y}{\lambda} &= \frac{Y}{Z} \\ &\Rightarrow \\ x &= \frac{\lambda X}{Z} \\ y &= \frac{\lambda Y}{Z} \end{aligned}$$

Pin-Hole Model

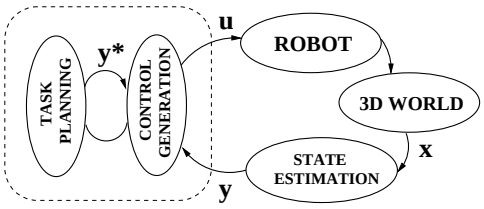
- Remember Homogeneous Coordinates?

$$\vec{P} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- Define the Perspective transform as

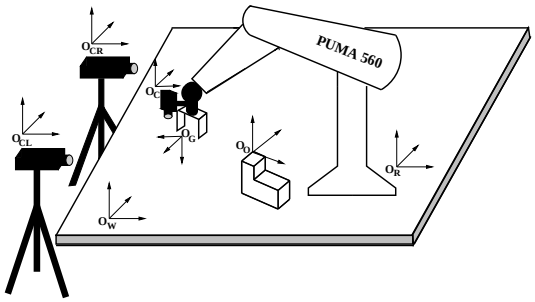
$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 0 \end{bmatrix}$$

The general problem of servoing



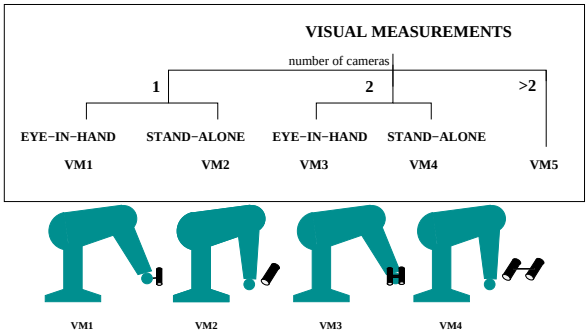
Outline

Coordinate Systems



Basic Organization

Camera - Robot Configurations

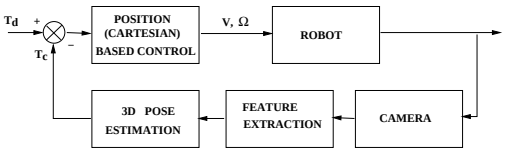


Outline

Approaches to servoing

- Position Based Servoing
- Image Based Servoing
- $2\frac{1}{2}$D Servoing

Position Based Servoing



Position Servoing

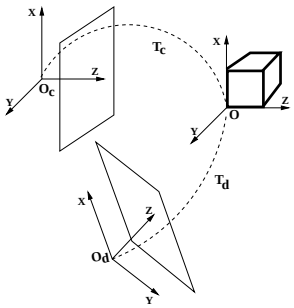


Image Based Servoing

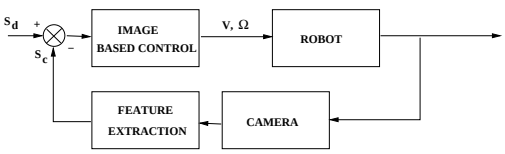
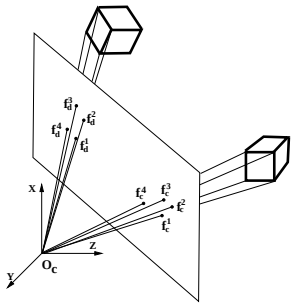
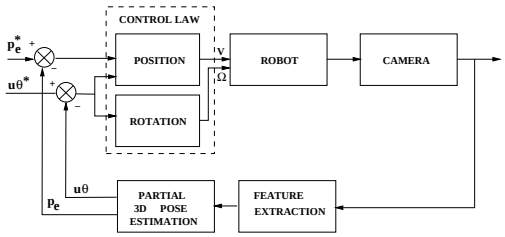


Image Servoing



2.5D Servoing



Outline

Image Based Servoing

- Specify the task as an image based task
- Associate an error function (e) with the task
- Goal is achieved when $e = 0$
- Derivation of an Image Jacobian to relate image changes to control of the robot

Image Jacobian

$$J(\mathbf{P}) = \left[\frac{\delta \mathbf{f}}{\delta \mathbf{P}} \right] = \begin{bmatrix} \frac{\delta f_1(\mathbf{P})}{\delta P_1} & \cdots & \frac{\delta f_1(\mathbf{P})}{\delta P_m} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_k(\mathbf{P})}{\delta P_1} & \cdots & \frac{\delta f_k(\mathbf{P})}{\delta P_m} \end{bmatrix}$$

Point Motion

- The point motion can be written as

$$^C\dot{\mathbf{P}} = {}^C\Omega \times {}^C\mathbf{P} + {}^C\mathbf{V}$$

- Or more detailed

$$\begin{aligned} \dot{x} &= z\omega_y - y\omega_z + T_x \\ \dot{y} &= x\omega_z - z\omega_x + T_y \\ \dot{z} &= y\omega_x - x\omega_y + T_z \end{aligned}$$

Point flow

- Given the perspective projection

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{\lambda}{z} \begin{bmatrix} x \\ y \end{bmatrix}$$

- The image flow is then

$$\begin{aligned} \dot{u} &= \lambda \frac{z\dot{x} - x\dot{z}}{z^2} \\ \dot{v} &= \lambda \frac{z\dot{y} - y\dot{z}}{z^2} \end{aligned}$$

Image Jacobian

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} k_u & 0 \\ 0 & k_v \end{bmatrix} \begin{bmatrix} \frac{1}{Z} & 0 & -\frac{x}{Z} & -xy & 1+x^2 & -y \\ 0 & \frac{1}{Z} & -\frac{y}{Z} & -1-y^2 & xy & x \end{bmatrix} \begin{bmatrix} V_X \\ V_Y \\ V_Z \\ \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}$$

For k image points?

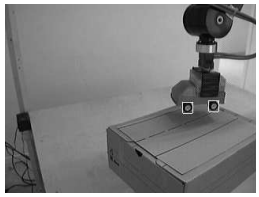
$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \vdots \\ \dot{u}_k \\ \dot{v}_k \end{bmatrix} = \mathbf{A} \begin{bmatrix} \frac{1}{Z_1} & 0 & -\frac{x_1}{Z_1} & -x_1 y_1 & 1+x_1^2 & -y_1 \\ 0 & \frac{1}{Z_1} & -\frac{y_1}{Z_1} & -1-y_1^2 & x_1 y_1 & x_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{Z_k} & 0 & -\frac{x_k}{Z_k} & -x_k y_k & 1+x_k^2 & -y_k \\ 0 & \frac{1}{Z_k} & -\frac{y_k}{Z_k} & -1-y_k^2 & x_k y_k & x_k \end{bmatrix} \begin{bmatrix} V_X \\ V_Y \\ V_Z \\ \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}$$

Considerations

- Naive use of image based servoing might generate surprising motion of the camera
- Comparison:

Approach	Pros	Cons
PBVS	Intuitive	Calibration?
IBVS	Accuracy	Singularities?

Illustrative Examples

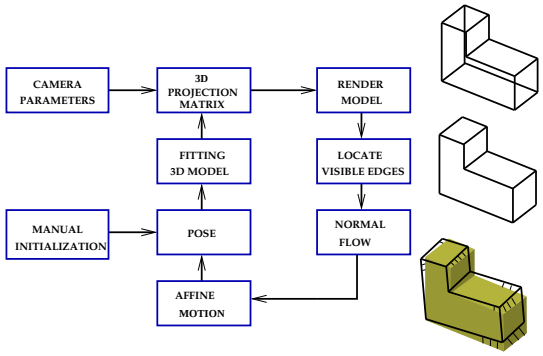


Outline

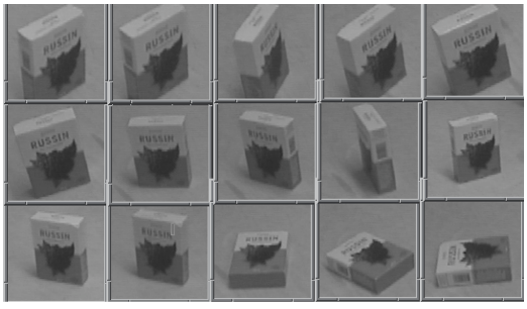
Smart Strategies

- Doing large scale servoing is really hard
- Doing limited recognition is much easier
- How about recognition based servoing?

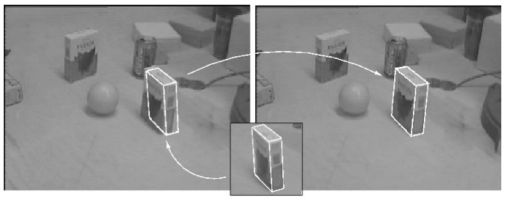
Model Based Servoing



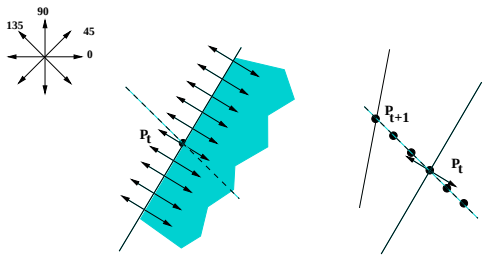
Initial Pose Estimation



Initialization



Moving Edge Model



Fitting the model

- How does one fit the model to the object?
- A local pertubation model might be adequate
- Or optimization of differential motion (Drummond & Cipola)

An polygonal approximation

$$\begin{bmatrix} x_i^{t+1} \\ y_i^{t+1} \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} + \begin{bmatrix} a_2 & a_3 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x_i^t \\ y_i^t \end{bmatrix} + \begin{bmatrix} a_6 & a_7 & 0 \\ 0 & a_6 & a_7 \end{bmatrix} \begin{bmatrix} {x_i^t}^2 \\ x_i^t y_i^t \\ {y_i^t}^2 \end{bmatrix}$$
$$= \mathbf{W}(X_i^t)\Theta$$

Model Based Servoing

Outline

Summary

- Overview of methods for visual servoing
- Major camera - robot configurations
- The basic motion equations for design of control
- Strategies to arrive at a good solution