ROBOTICS & VISUAL SERVOING INTRODUCTION

PROBLEM: Guiding a robot to a target

Assembly robotics: component insertion
Mobile robotics: docking

VISUAL SERVOING THEORY

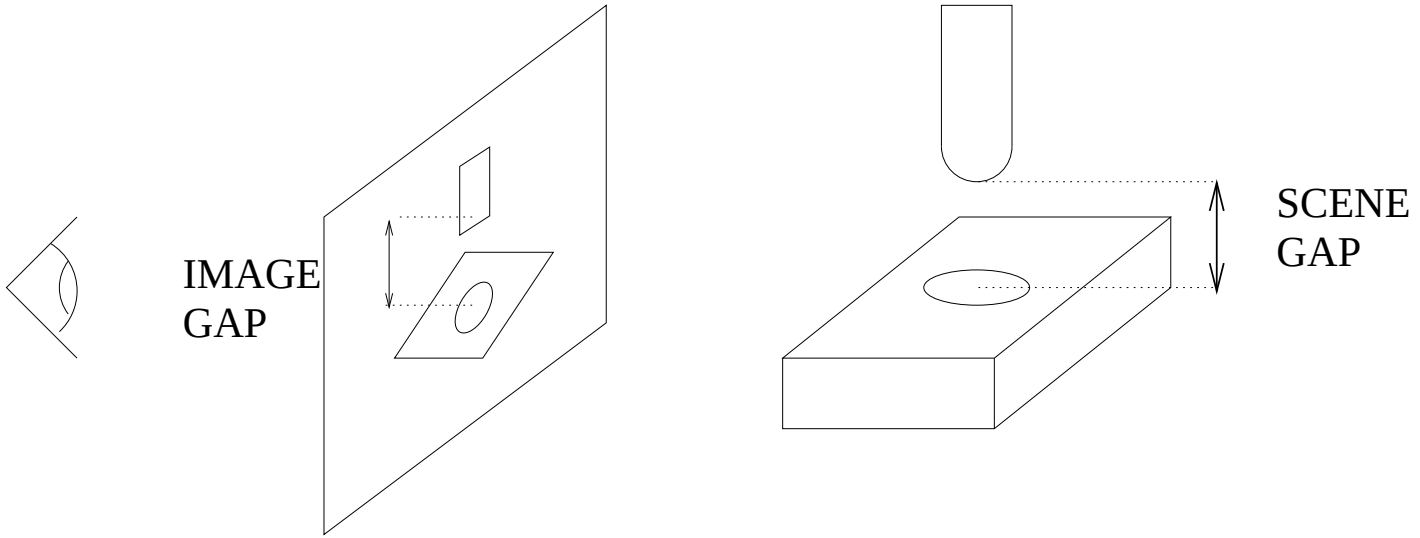
APPROACH 1: MODEL DRIVEN

KNOWN:
Robot manipulation model
Part & robot position known
Camera calibrated
Target model

BUT: errors exist and combine - noise, unknowns, gear backlash
SO: need very accurate calibration & mechanisms

TRY 2: VISUAL SERVOING

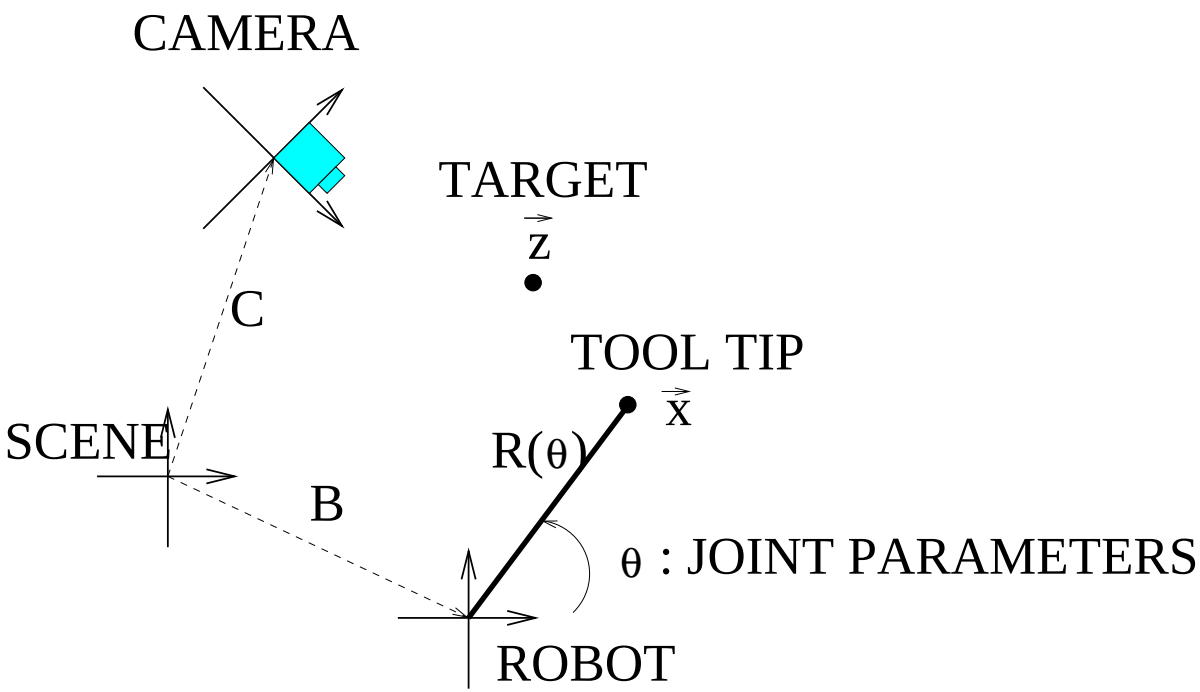
Move so that observed positions directly link to task distances



- Incrementally move to reduce image gap
- Don't know exact positions

MORE FORMAL MODEL

CONSIDER:



VARIOUS QUANTITIES

B - ROBOT POSITION WRT SCENE

C - CAMERA POSITION WRT SCENE

P - CAMERA PROJECTION MODEL

$\vec{\theta}$ - ROBOT JOINT PARAMETERS

R($\vec{\theta}$) - TOOL POSITION WRT ROBOT

LINKING EQUATION

IF:

$\vec{x}$ - TOOL TIP WRT TOOL

$\vec{z}$ - TARGET WRT SCENE

$\vec{a}$ - IMAGE OF TOOL TIP

$\vec{b}$ - IMAGE OF TARGET

THEN:

$$\vec{a} = \mathrm{PC}^{-1}\mathrm{BR}(\vec{\theta})\vec{x}$$
$$\vec{b} = \mathrm{PC}^{-1}\vec{z}$$

APPROACH 1: MODEL DRIVEN

Calibrate **P**, **C**, **B**, $\mathrm{R}(\vec{\theta})$ accurately

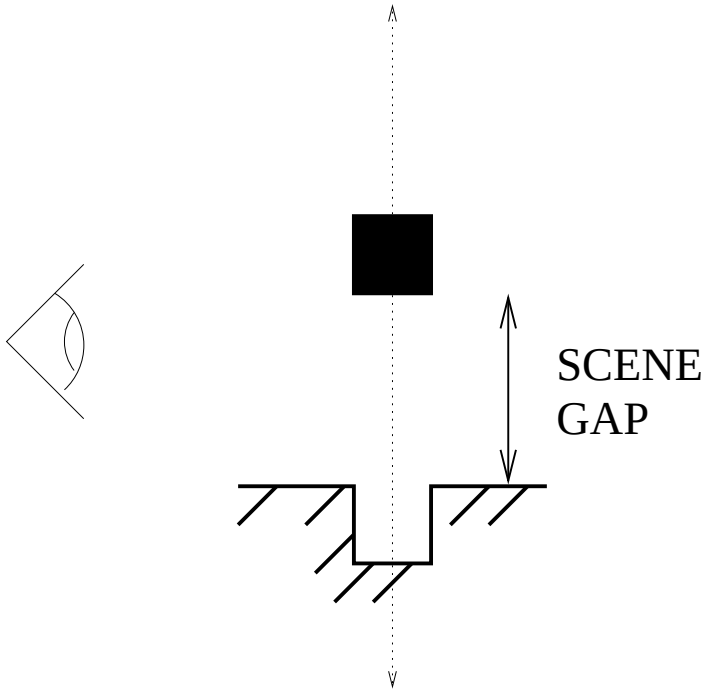
SOLVE:

$$\vec{\theta} = \vec{f}(\vec{z}, \vec{x}, \mathrm{P}, \mathrm{C}, \mathrm{B})$$

Hard analytics
Not always solveable

MID-LECTURE QUESTION

If only vertical robot motion possible to reduce the gap:



How to get peg exactly into hole?

APPROACH 2: VISUAL SERVOING

How much to move joints ($\Delta\vec{\theta}$) to reduce target error $\vec{\Delta} = \vec{a} - \vec{b}$?

Visually estimate $\vec{f}()$ such that

$$\Delta\vec{\theta} \doteq \vec{f}(\vec{\Delta})$$

Use $\Delta\vec{\theta}$ to partially approach target
and recompute $\vec{\Delta}$
Iterate

ESTIMATING $\vec{f}()$

Move robot joint i slightly from θ_i to $\theta_i + \epsilon$

Observe tool tip moves slightly from $\vec{a}$ to $\vec{a} + \vec{\delta}_i$

Compute:

$$\frac{\partial \vec{a}}{\partial \theta_i} \doteq \frac{(\vec{a} + \vec{\delta}_i) - \vec{a}}{(\theta_i + \epsilon) - \theta_i} = \frac{\vec{\delta}_i}{\epsilon}$$

Repeat for all i to estimate **JACOBEAN** matrix $\mathbf{J} = \frac{1}{\epsilon}[\vec{\delta}_1, \ldots, \vec{\delta}_N]$ for N joints

USING THE JACOBEAN

Math theory:

$$\vec{\Delta} \doteq \mathrm{J}\Delta\vec{\theta}$$

J is $2 \times N$, so use pseudo-inverse

$$\boxed{\Delta\vec{\theta} = (\mathrm{J}^{\mathrm{T}}\mathrm{J})^{-1}\mathrm{J}^{\mathrm{T}}\vec{\Delta}}$$

But **J** is only approximate and not true inverse?

So, move $\alpha\Delta\vec{\theta}$ and iterate ($\alpha < 1$)

SERVOING ALGORITHM

DO

Compute $\vec{\Delta}$
Estimate $\mathbf{J}$
Compute $\Delta\vec{\theta}$
Move joints $\alpha\Delta\vec{\theta}$ where $0 < \alpha < 1$

WHILE $|| \vec{\Delta} ||> \tau$ pixels

$\tau > 1$? Maybe robot is 'shakey'

CONVERGENCE?

If α small enough, should always be reducing $\vec{\Delta}$

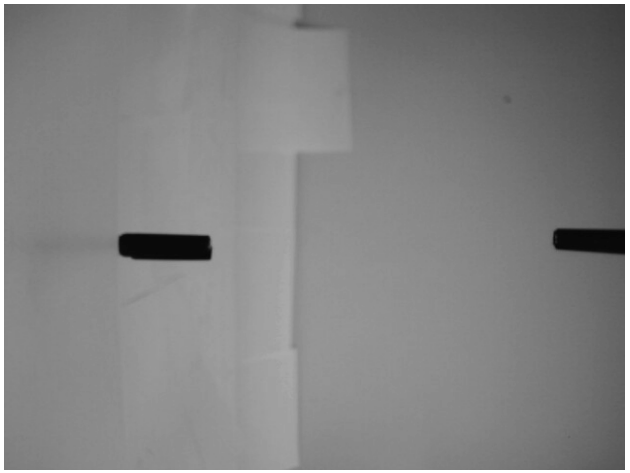
As $\mathbf{J}$ is linear, moving $\alpha\Delta\vec{\theta}$ should reduce position error by approximately $\alpha\vec{\Delta}$

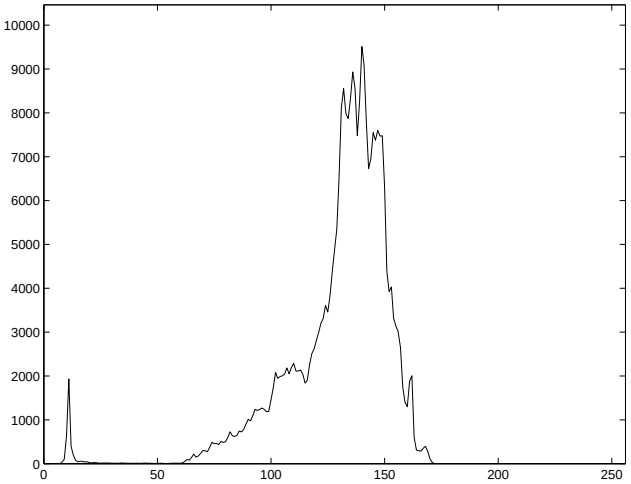
STEP	MOVEMENT	REMAINDER
1	$\alpha \mid\mid \vec{\Delta} \mid\mid$	$(1-\alpha) \mid\mid \vec{\Delta} \mid\mid$
2	$(1-\alpha)^1\alpha \mid\mid \vec{\Delta} \mid\mid$	$(1-\alpha)^2 \mid\mid \vec{\Delta} \mid\mid$
	...	
n	$(1-\alpha)^{n-1}\alpha \mid\mid \vec{\Delta} \mid\mid$	$(1-\alpha)^n \mid\mid \vec{\Delta} \mid\mid$

Total movement $= \Sigma_{n=1}^{\infty}(1-\alpha)^{n-1}\alpha \mid\mid \vec{\Delta} \mid\mid$
$= \left(\frac{1}{1-(1-\alpha)}\right) \alpha \mid\mid \vec{\Delta} \mid\mid = \mid\mid \vec{\Delta} \mid\mid$

VISUAL SERVOING RESULTS 1

Initial position and histogram

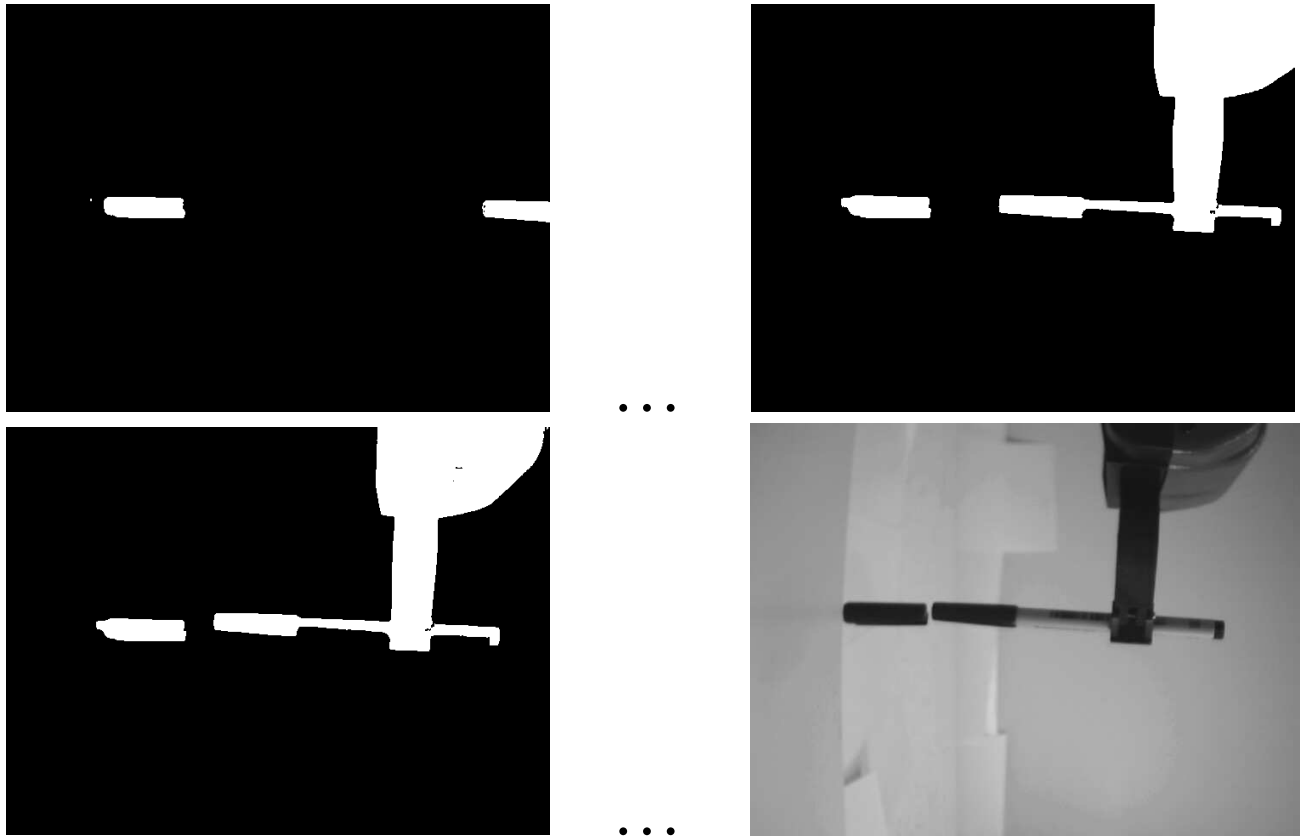




Camera on side

VISUAL SERVOING RESULTS 2

Note: changing binary images



5 iterations

MID-LECTURE QUESTION

What could go wrong with the thresholding and visual servoing approach?

What We Have Learned

- Servoing versus model based control
- Basics of visual servoing