

# Gesture Interface

## A Naive Framework

---

Pan An (Andy)

December 1, 2016

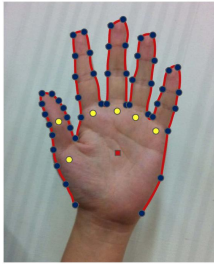
National University of Singapore

# Table of Contents

1. Introduction
2. Inverse Kinematics
3. Learning Methods

# Introduction

---



**Figure 1:** Example of Hand Gesture.

## Color Based Method

The above picture is an example of color based gesture recognition. Color based algorithms are normally heavily relying on background color.

### Spatial Motions

Swipe, tracking, clapping...

### Finger Motions

Grabbing, rub, pointing and etc.

### Muscles and Other Biosignals

All Involves hardwares.

### Note

Naive camera based gestures are used.

# Marker Based Methods

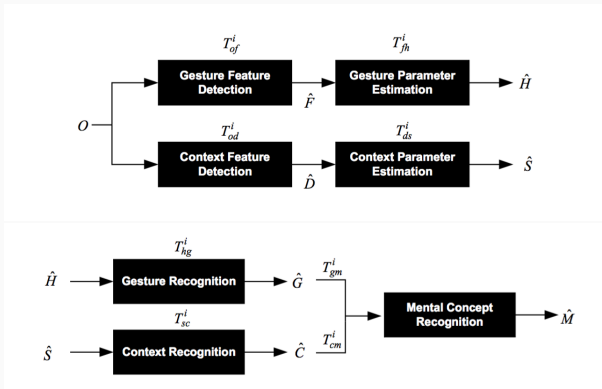


Figure 2: Overview of Gesture Model

# Marker Based Methods

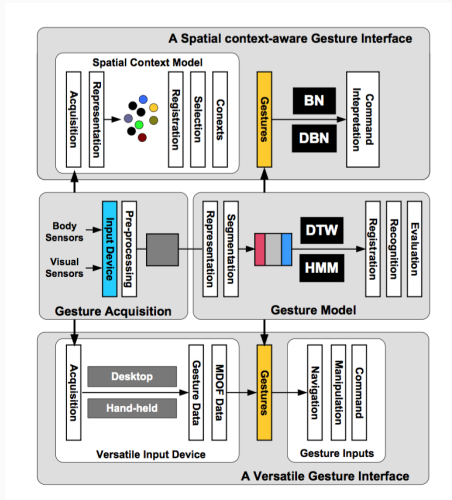


Figure 2: Gesture Model

# Marker Based Methods

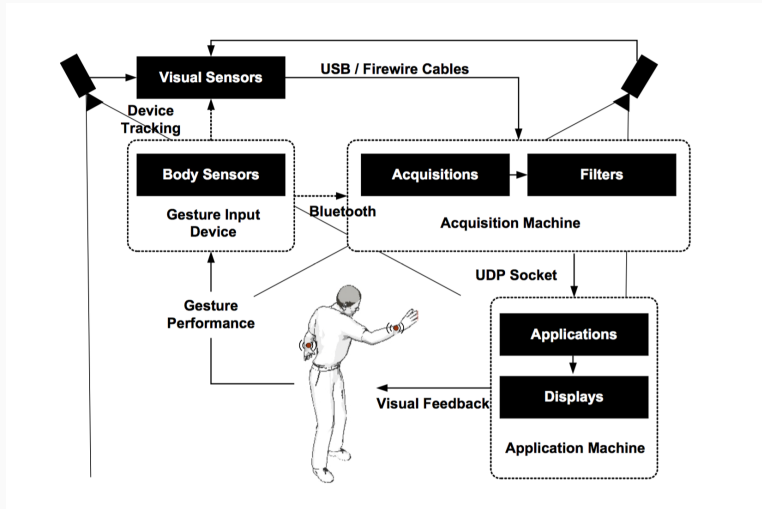


Figure 2: An Example System

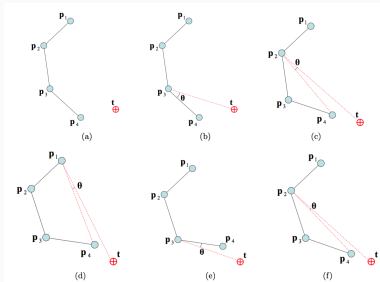
# Inverse Kinematics

---



- Cyclic Coordinate Descent
- Jacobian Transpose
- FABRIK
- Neural Network

# Cyclic Coordinate Descent



## Basic Idea

- Greedy
- Iterative
- Does not care whether the target is within range

## Why

WangXin made the CCD work.

# Jacobian Transpose

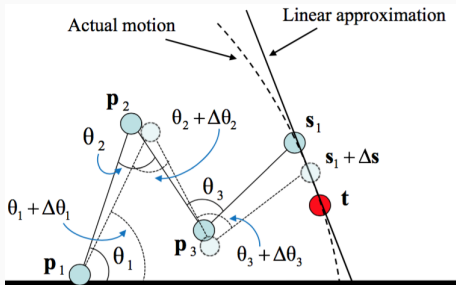


Figure 3: Jacobian Transpose

## Definitions

$J$  Partial Derivation of the entire chain system.

$\theta$  Vector of  $\theta$  values.

$s$  Vector of end effectors.

$p_j$  Position of the joints.

## Jacobian Matrix

$$J(\theta_{ij}) = \left( \frac{\partial s_i}{\partial \theta_j} \right)_{ij}$$

Where  $i = 1, \dots, k$  and  $j = 1, \dots, n$ .

In this case  $k = 1$  and  $n = 3$ .

# Jacobian Transpose

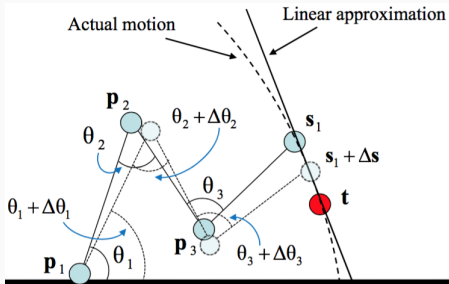


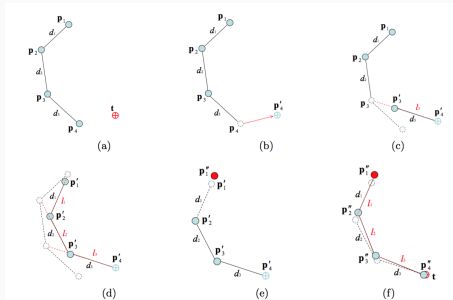
Figure 3: Jacobian Transpose

## Jacobian Transpose

Jacobian Transpose is to move the angles with a step of

$$\Delta\theta = \alpha J^T \vec{e}$$

Where  $\vec{e}$  is the vector of direction of the step and  $\alpha$  is the selected step size.



**Figure 4: FABRIK**

## FABRIK

Stands for Forward And Backward Reaching Inverse Kinematics.

## Definitions

$\mathbf{t}$  Vector of targets.

$d_i$  Distance between each joint  $d_i = \|\mathbf{p}_{i+1} - \mathbf{p}_i\|$

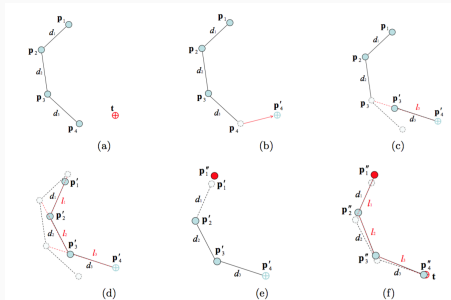


Figure 4: FABRIK

**Data:**  $d, t, p$

**Result:** The new joint positions  $p$   
initialization;

**if** *Target is not within range* **then**

    point directly to the target and return;

**else**

$b = p_1$  and  $dif_A = |p_n - t|$ ;

**while**  $dif_A > tolerance$  **do**

$p_n = t$ ;

**for**  $i = n - 1, \dots, 1$  **do**

$r_i = |p_{i+1} - p_i|$ ;

$\lambda_i = \frac{d_i}{r_i}$ ;

$p_i = (i - \lambda_i)p_{i+1} - \lambda_i p_i$ ;

**end**

$p_1 = b$ ;

**for**  $i = 1, \dots, n - 1$  **do**

$r_i = |p_{i+1} - p_i|$ ;

$\lambda_i = \frac{d_i}{r_i}$ ;

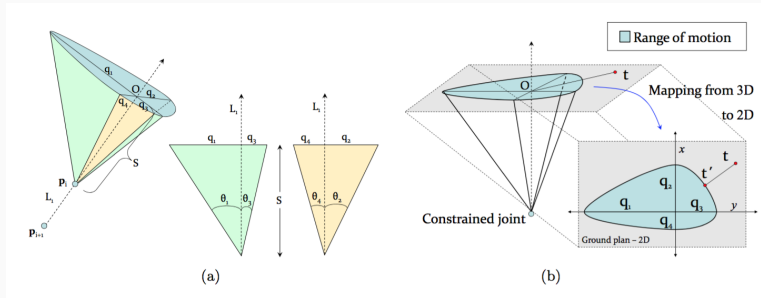
$p_i = (i - \lambda_i)p_{i+1} - \lambda_i p_i$ ;

**end**

**end**

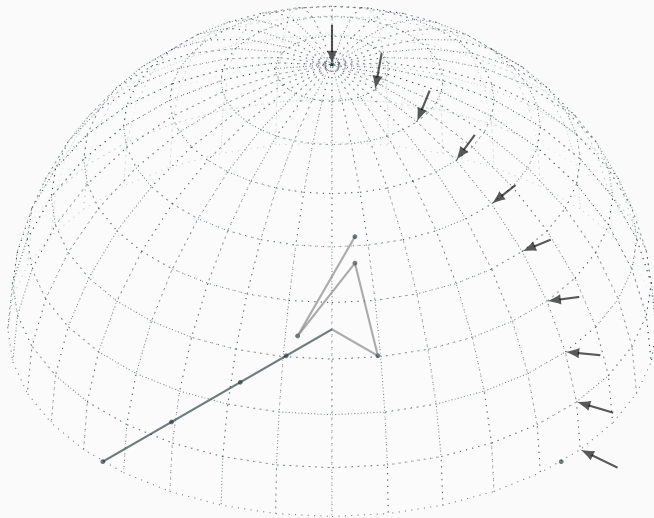
**end**

# FABRIK Constraints



**Figure 5:** Example of Constraints in a Joint System

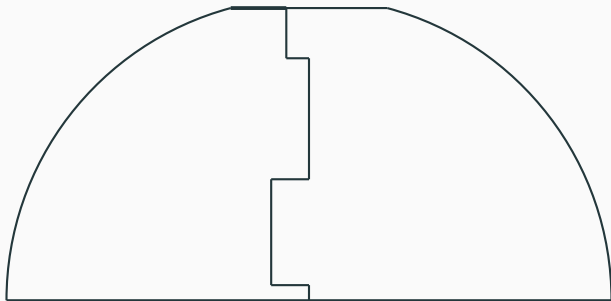
# Reachable Space



**Figure 6:** Reachable Space of a Fully Free Robot Arm



## Reachable Space for UR10

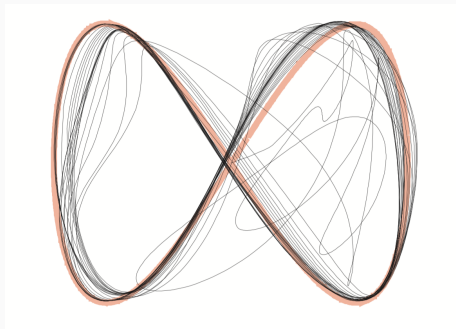


**Figure 7:** Reachable Space for UR10

# Learning Methods

---

# Actions(Scenarios)



## Natural Gesture of a Robot

Unpredictability

Robustness

Gesture when reaching the end effector

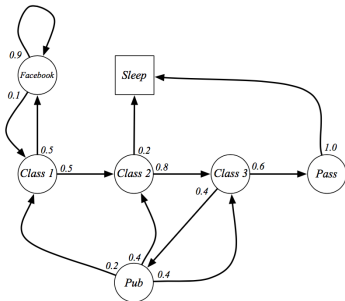
## Learning Method

Monte Carlo(policy search)

Dynamic Programming(value function)

Neural Network

# Model



$$\mathcal{P} = \begin{matrix} & \begin{matrix} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{matrix} \\ \begin{matrix} C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{matrix} & \begin{bmatrix} & & & & & 0.5 & \\ & 0.5 & & & & & 0.2 \\ & & 0.8 & & & & \\ & & & 0.6 & 0.4 & & 1.0 \\ 0.2 & 0.4 & 0.4 & & & & \\ 0.1 & & & & & 0.9 & \\ & & & & & & 1 \end{bmatrix} \end{matrix}$$

## Definitions

$\pi$  Policy: mapping from states to actions

$S$  A set of States

$A$  A set of Actions

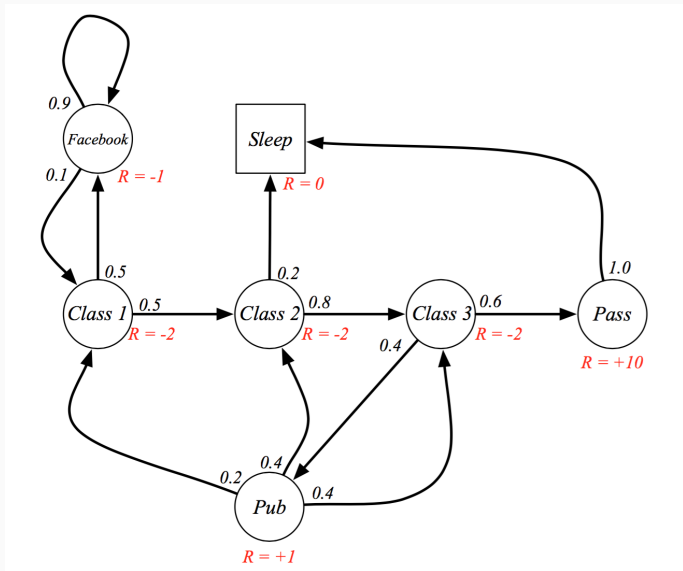
$R$  Reward Function

$P$  State transition function

$v(s)$  State Value Function of a MRP

$\gamma$  Discount Function,  $\gamma \in [0, 1]$

# Model



## Goal

Discover an optimal policy that maximizes:

$$J = E\{\sum_{h=0}^H R_h\}$$

Where  $H$  is the steps the algorithm takes

## Expanding

Expanding the above with reward settings:

$$\begin{aligned} \max_{\pi} J(\pi) &= \sum_{s,a} \mu^{\pi}(s) \pi(s,a) R(s,a) \\ \text{s.t. } \mu^{\pi} &= \sum_{s,a} \mu^{\pi}(s) \pi(s,a) T(s,a,s'), \forall s' \in S, \\ 1 &= \sum_{s,a} \mu^{\pi}(s) \pi(s,a) \\ \pi(s,a) &\leq 0, \forall s \in S, a \in A \end{aligned}$$

Where  $\mu$  is the distribution of states.

## Methods

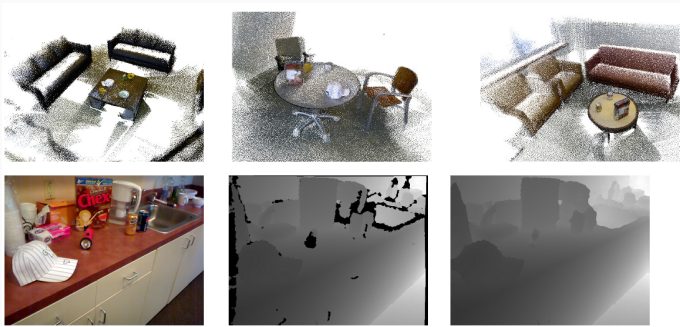
- Video Stream
- Depth Camera Stream
- 3D Reconstruction(Event Camera)

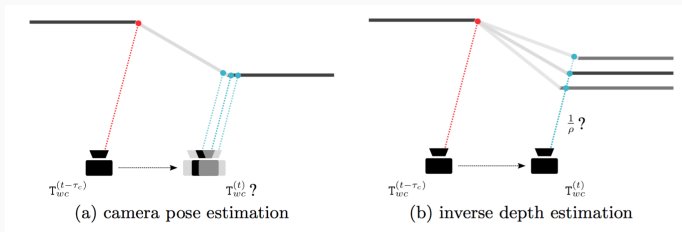


# Methods

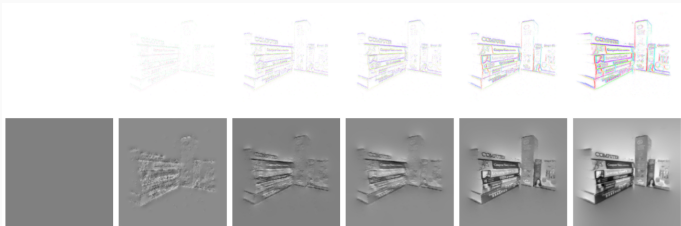


# Methods





# Methods



# Methods



**Questions?**

# Backup slides

Sometimes, it is useful to add slides at the end of your presentation to refer to during audience questions.

The best way to do this is to include the `appendixnumberbeamer` package in your preamble and call `\appendix` before your backup slides.

**metropolis** will automatically turn off slide numbering and progress bars for slides in the appendix.



P. Erdős.

**A selection of problems and results in combinatorics.**

In *Recent trends in combinatorics (Matrahaza, 1995)*, pages 1–6.  
Cambridge Univ. Press, Cambridge, 1995.



R. Graham, D. Knuth, and O. Patashnik.

**Concrete mathematics.**

Addison-Wesley, Reading, MA, 1989.



G. D. Greenwade.

**The Comprehensive Tex Archive Network (CTAN).**

*TUGBoat*, 14(3):342–351, 1993.



D. Knuth.

**Two notes on notation.**

*Amer. Math. Monthly*, 99:403–422, 1992.





H. Simpson.

**Proof of the Riemann Hypothesis.**

preprint (2003), available at

<http://www.math.drofnats.edu/riemann.ps>, 2003.