



# Learning and Predicting Human Intentions Through Interactions

Fumin Zhang  
Georgia Institute of Technology

The research work is supported by ONR grants N00014-14-1-0635 and N00014-16-1-2667; NSF grants CMMI-1436284 and OCE-1559475; NRL N0017317-1-G001; and NOAA NA16NOS0120028.



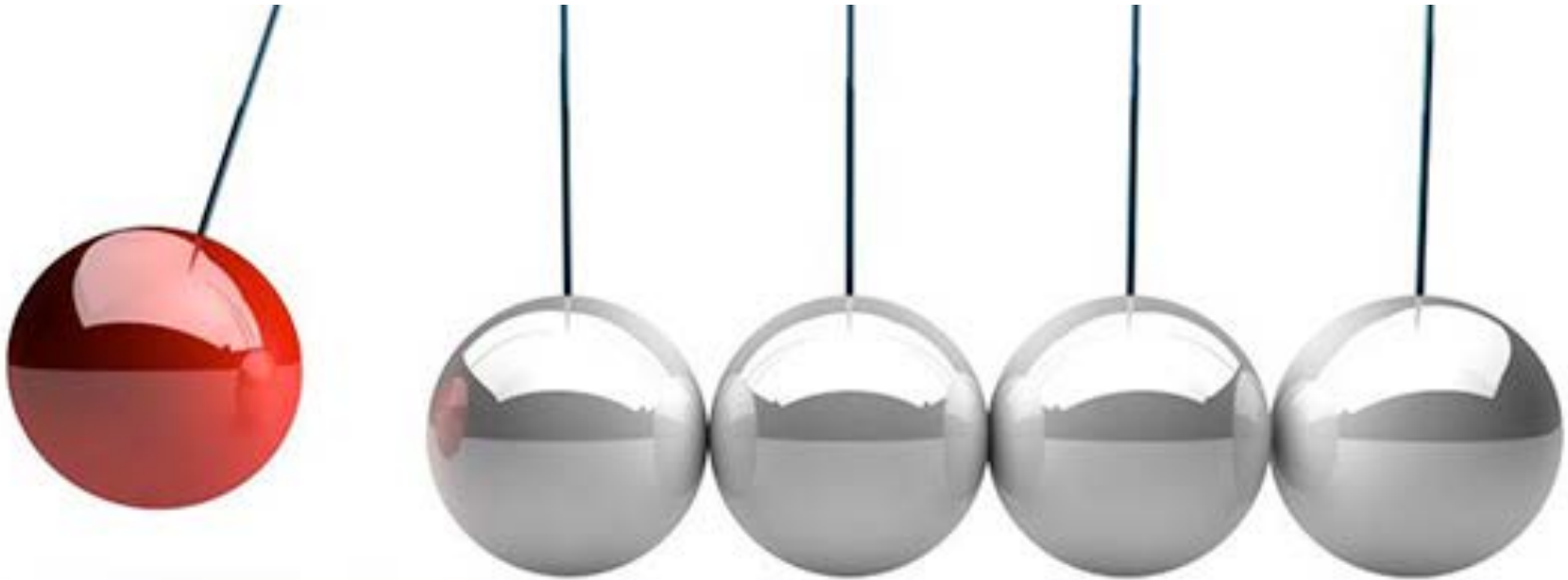


# Natural and Fun to Work With



# Physical Action and Reaction

Georgia Tech  
Systems Research



Newton's Third Law

Action =  $-$  Reaction



# H-R Action and Reaction



## Hidden Intentions

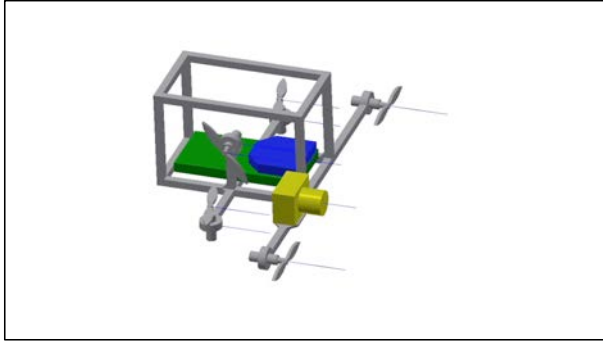
1. Difficult to *Recognize* Human Intentions
2. Difficult to *Predict* Human Intentions



# Georgia Tech Miniature Autonomous Blimp

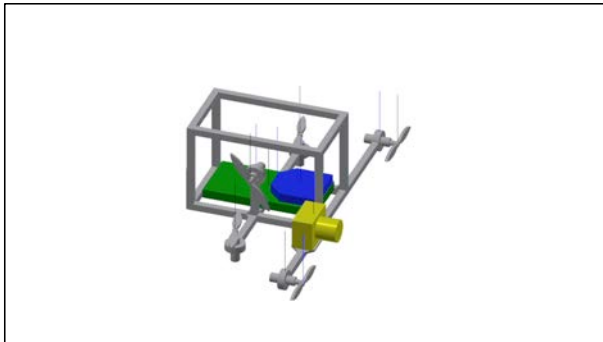
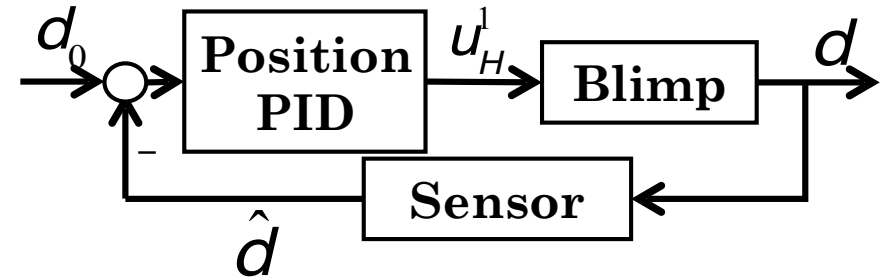


# Feedback Control



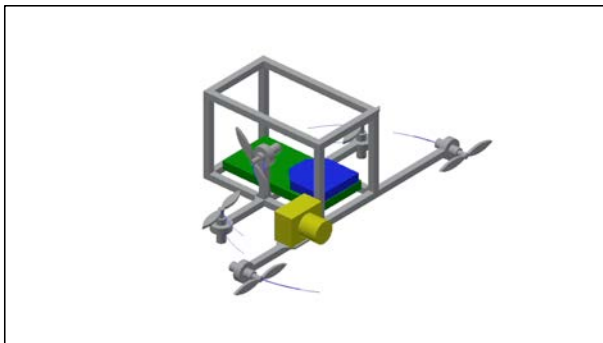
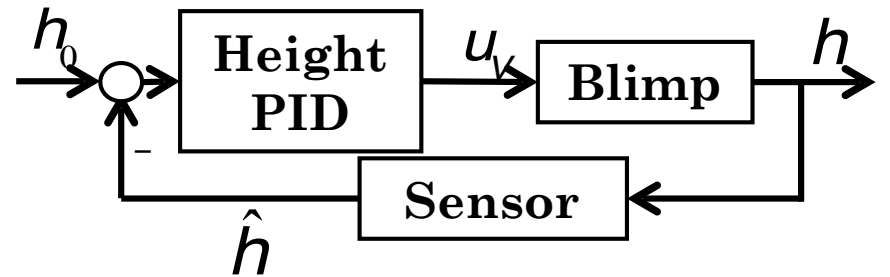
Forward and backward motion

$$m\ddot{d} = F_z + f_z$$



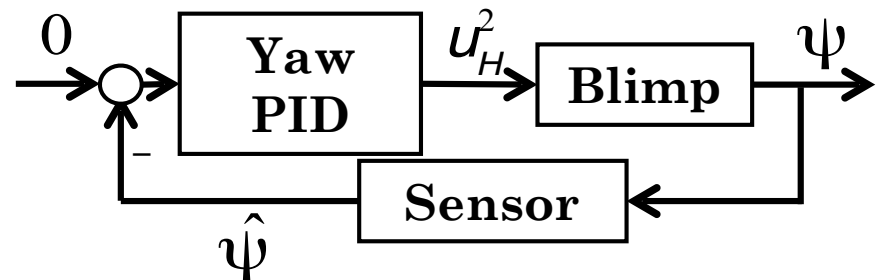
Up and down motion

$$m\ddot{h} = F_y + f_y$$



Spinning motion

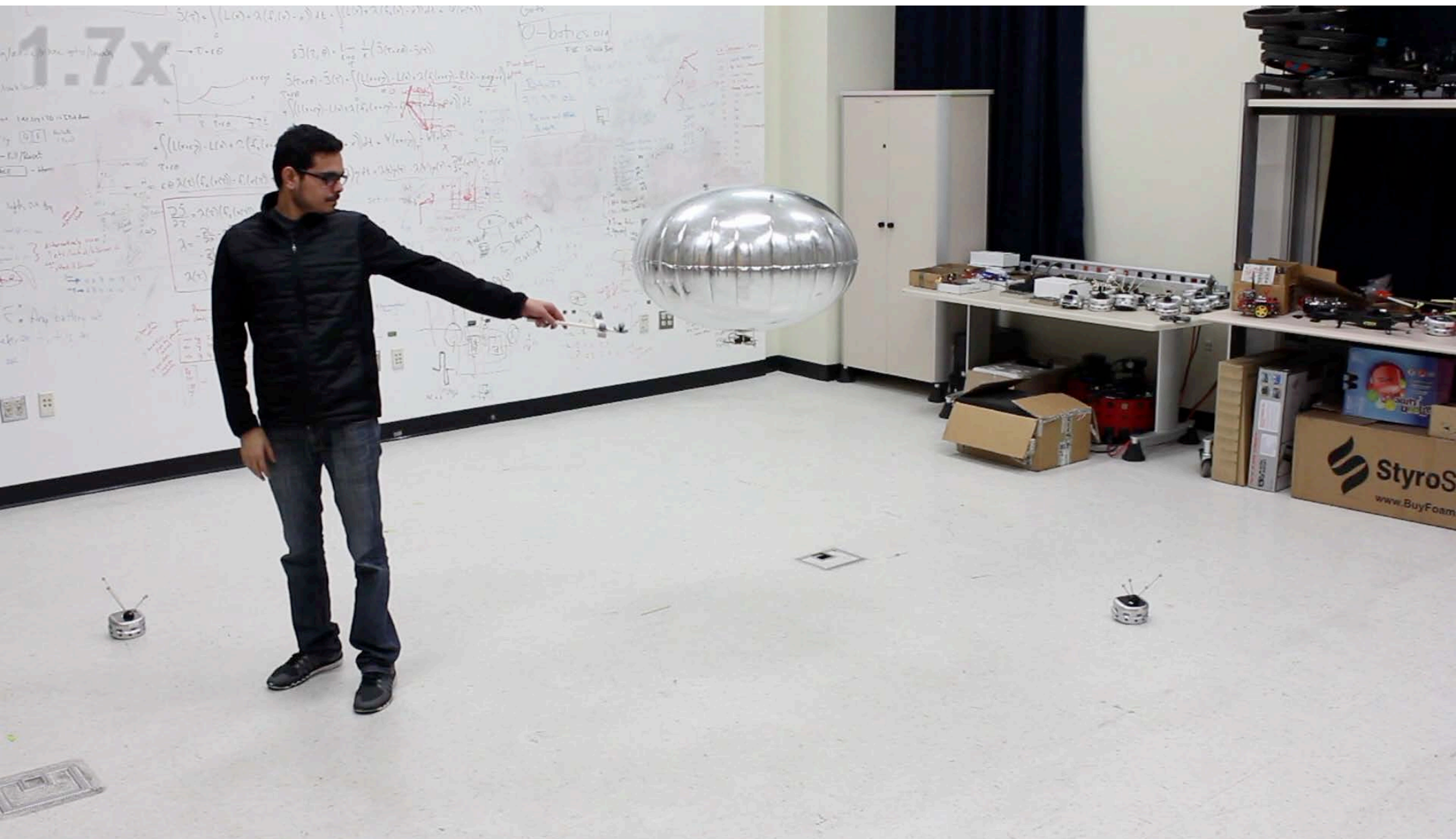
$$I\ddot{\psi} = M + \tau$$





# Point and Fly

Georgia Tech  
Systems Research



# Intentions are Clear



**Localization markers** are placed on the blimp, the destinations, and the wand.

**Human Intention:**

Move blimp to one of the two destinations.

Blimp control performs well.

Interaction is **structured and staged**.



# Pointing Motion on Computer

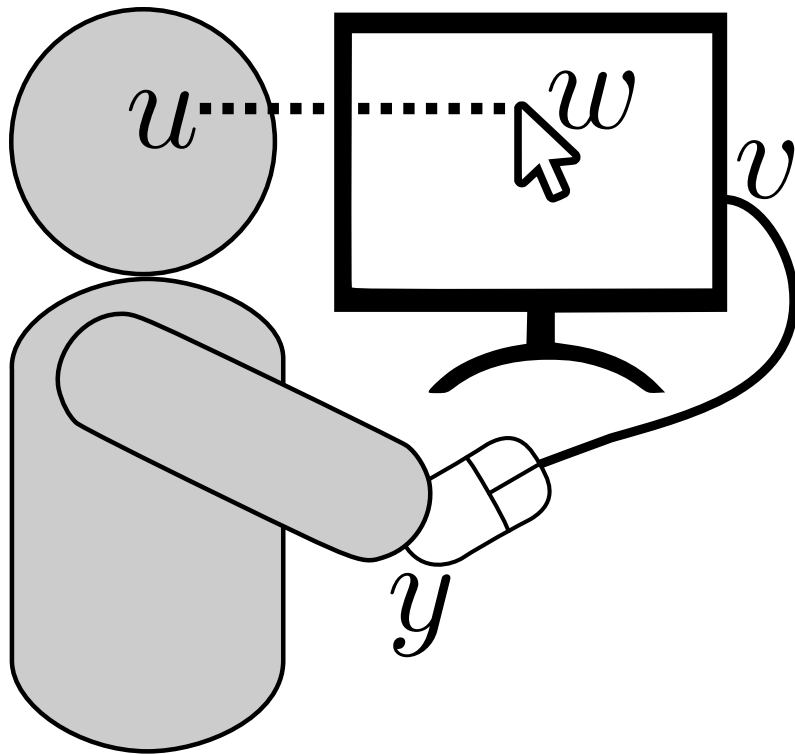
Georgia Tech  
Systems Research



To better design blimp motion, we need to understand human pointing motion better.

Avoiding blimp dynamics (temporarily)

# Mouse Pointing



$y$ : pointing device position  
 $v$ : measured device position  
 $w$ : displayed pointer position  
 $u$ : perceived pointer position

*How to describe the human intention?*

# The VITE Model



## Vector Integration to Endpoint Model:

$$\dot{\nu} = \gamma(-\nu + \rho - u)$$

$$\dot{y} = g(t)[\nu]_d^+$$

$\rho$  Desired position of pointer on screen.

$\nu$  Difference vector.

Assume  $\rho = 0$ , and only consider the 1-D case.

The switching function

$$[\nu]_d^+ = 0 \text{ if } \nu u < 0$$

$$[\nu]_d^+ = \nu \text{ if } \nu u > 0$$

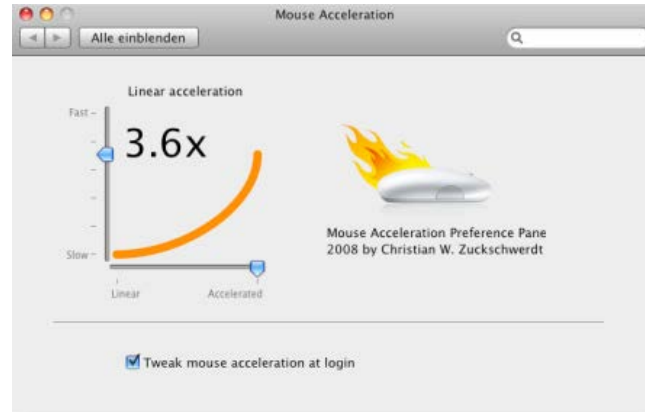
Bullock D, and Grossberg S. Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review* 1988; 95(1):49–90.



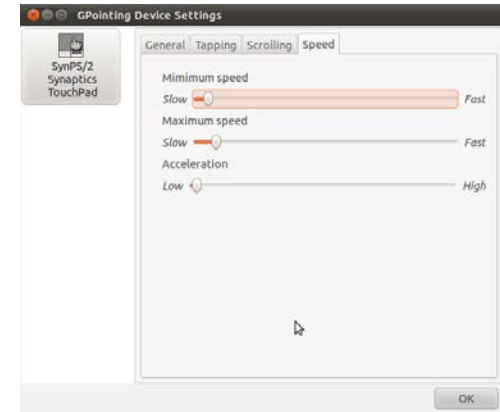
# Pointer Acceleration



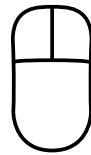
Windows



Mac



Linux



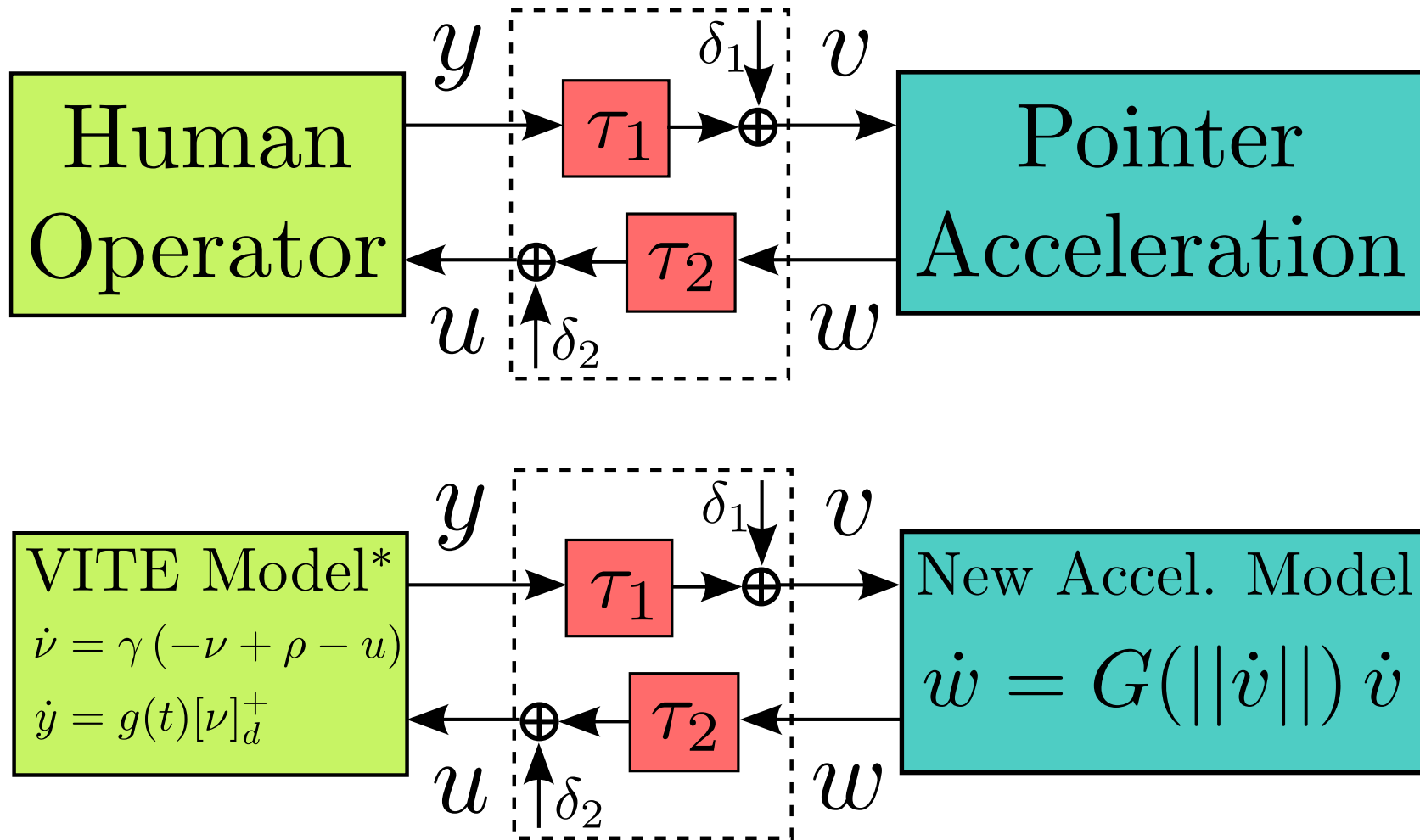
Slow  
Movements



Fast  
Movements



# Modeling Pointer Acceleration



.P. Varnell, M. Malisoff and F. Zhang, “**Stability and Robustness Analysis for Human Pointing Motions with Acceleration under Feedback Delays,**” *International Journal of Robust and Nonlinear Control*, 27(5)703-721, 2017.

# Acceleration Strategies



Name	Scaling Function	I/O Velocity Plot
No Acceleration	$G(  \dot{v}  ) = k_1$	
Threshold	$G(  \dot{v}  ) = \begin{cases} k_1, & \text{if } 0 \leq   \dot{v}   < c \\ k_2, & \text{if }   \dot{v}   \geq c \end{cases}$	
Linear	$G(  \dot{v}  ) = k_1 + k_2  \dot{v}  $	
Implementation	Scaling Function	
Microsoft Windows	Threshold (3 levels)	
Mac OSX	Linear (based on visual inspection only)	
Linux (xinput)	8 types (includes threshold, linear, polynomial, etc.)	



# Closed Loop Dynamics



Assume no delay, 1-D, target position is at zero

## Model

*Given the previous assumptions, and letting  $x = [w \quad \nu]^\top$ , the closed loop pointing dynamics are*

$$\dot{x} = \begin{bmatrix} \tilde{G}(x_2^+) x_2^+ \\ -\gamma (x_1 + x_2) \end{bmatrix}$$

*where  $\tilde{G}(\cdot) = g G(g \cdot)$  and  $x_1^+ = \begin{cases} x_1, & \text{if } x_1 \geq 0 \\ 0, & \text{else} \end{cases}$*



## Theorem (Stability of Human Pointing with Acceleration)

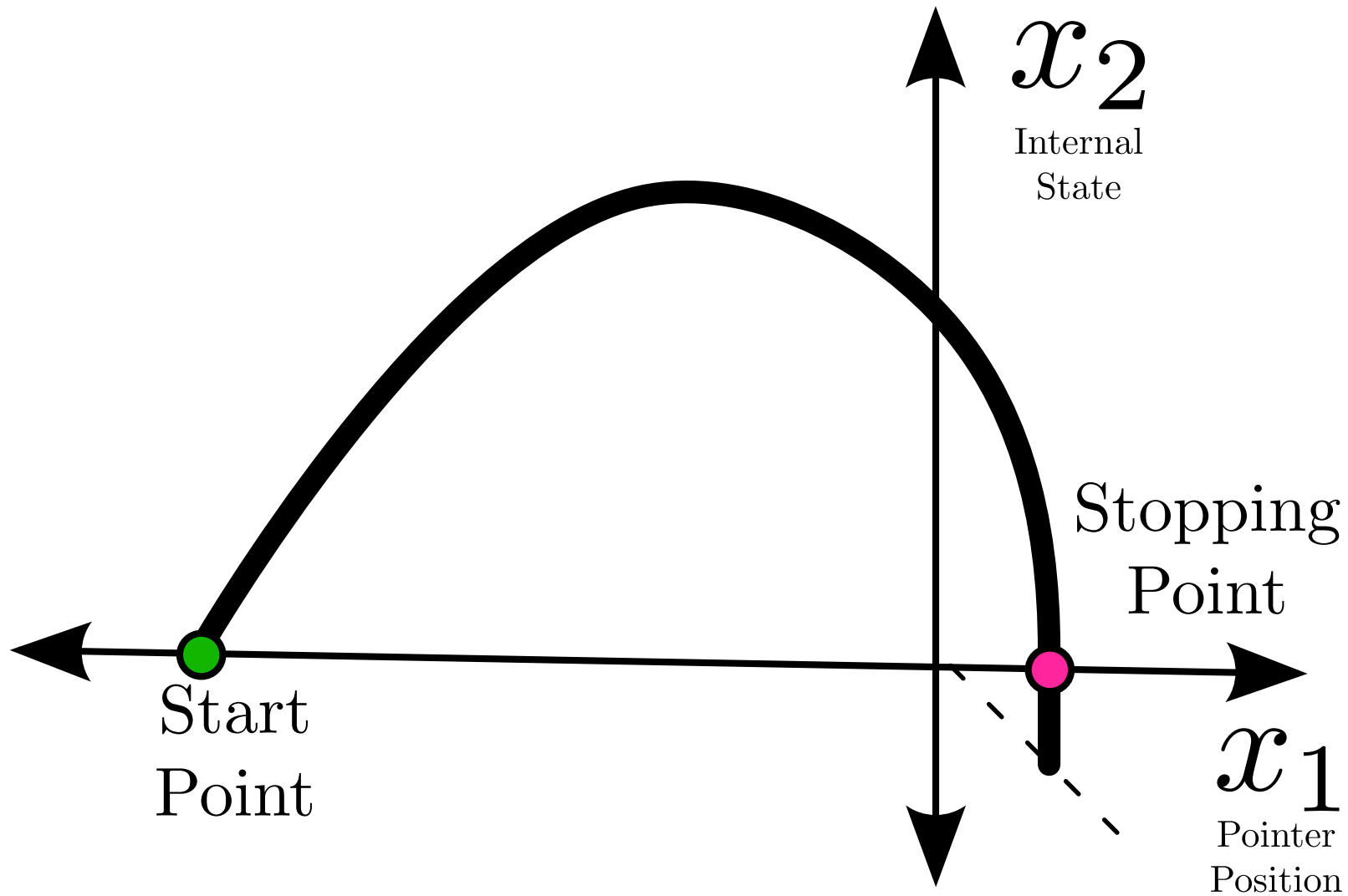
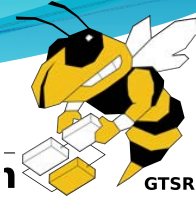
*For all  $\gamma \in \mathbb{R}_{>0}$  , and all  $\tilde{G} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  non-decreasing, the system defined by:*

$$\dot{x} = \begin{bmatrix} \tilde{G}(x_2^+) x_2^+ \\ -\gamma (x_1 + x_2) \end{bmatrix}$$

*has an equilibrium set  $E = \{x : x_2 = -x_1 \text{ and } x_1 \geq 0\}$  to which the state is globally asymptotically stable.*

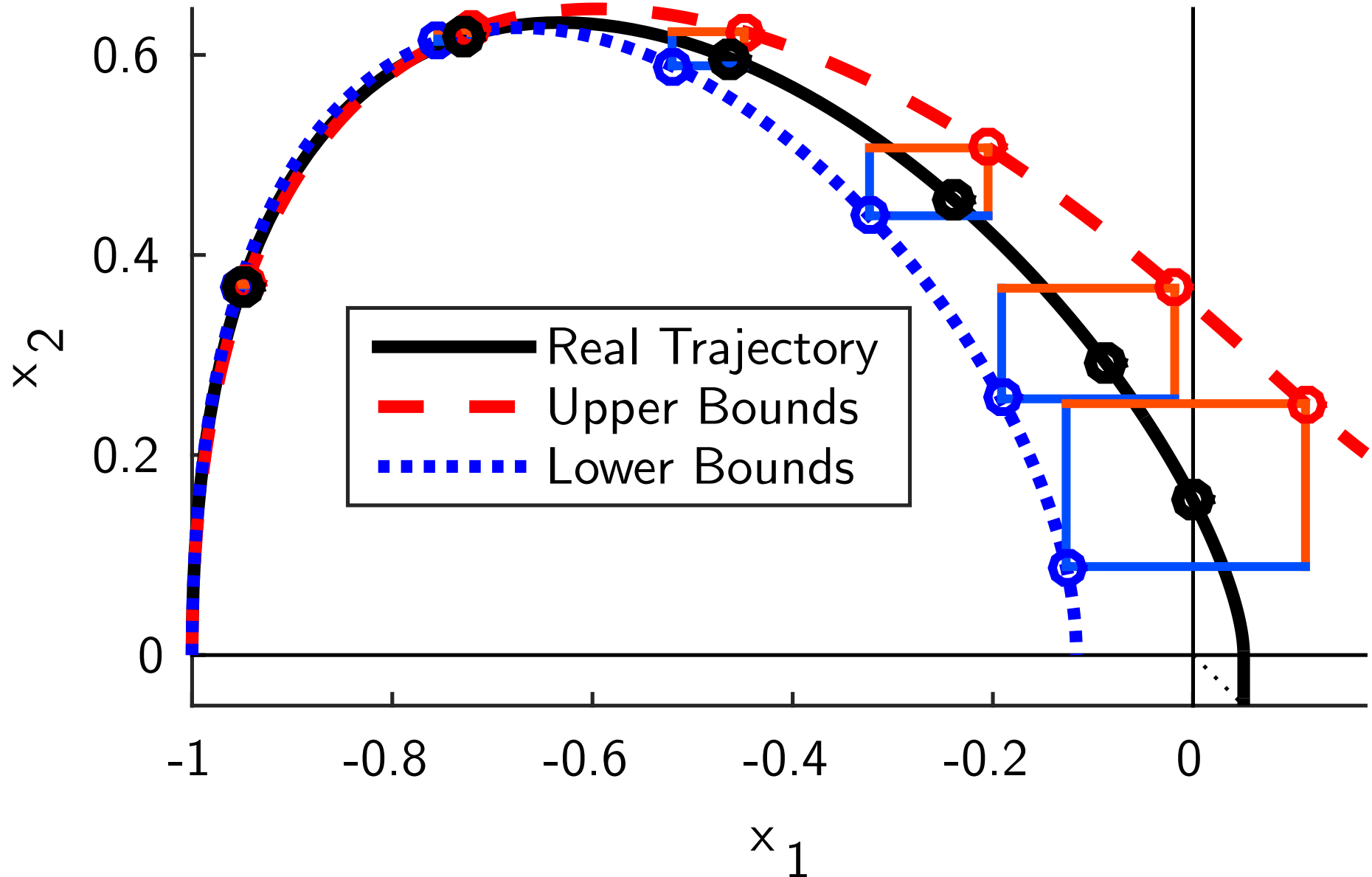
Note the equilibrium states are NOT desired by human.

# Finite Time Performance



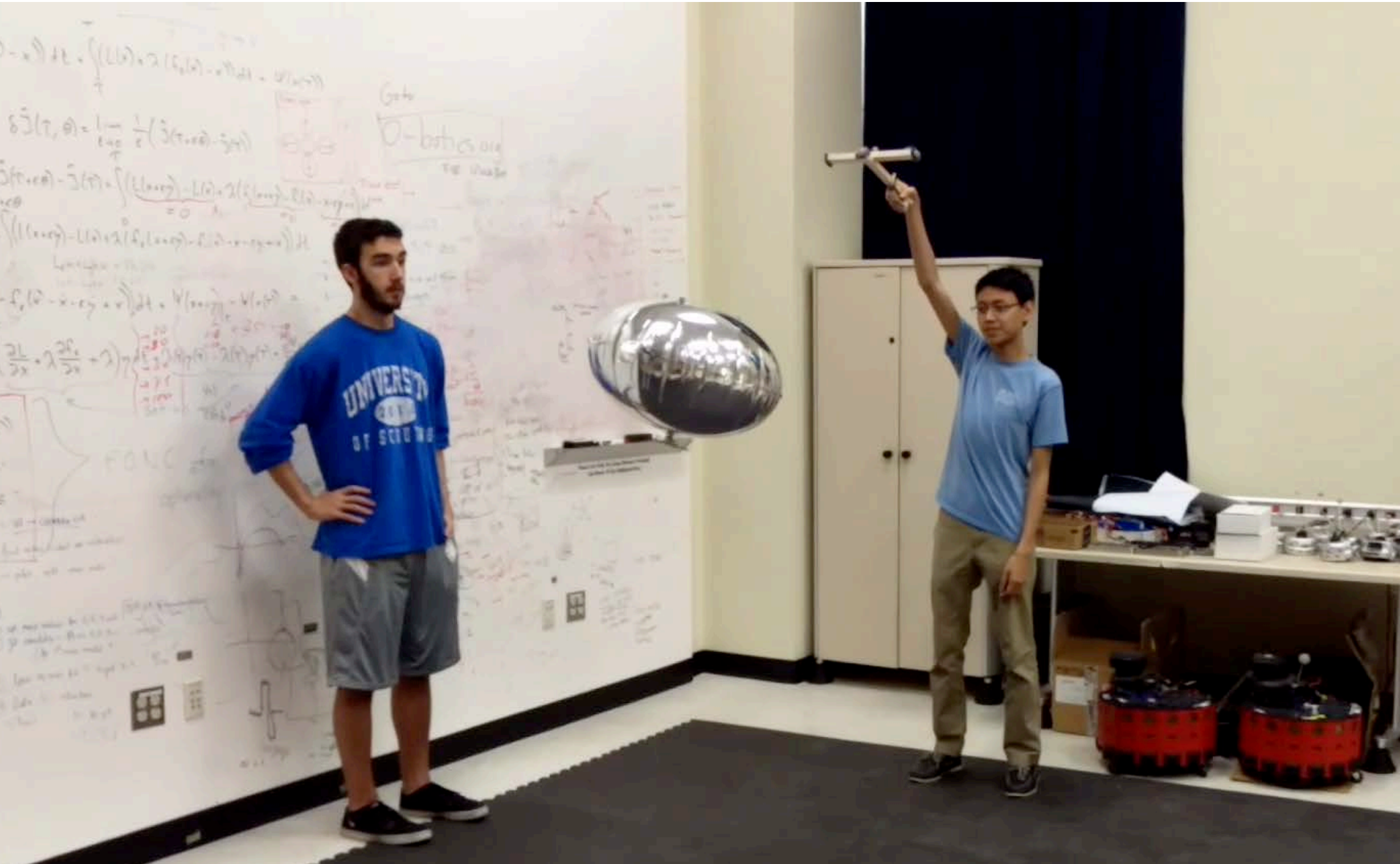


# Effect of Time Delay



# Replacing Pointer by Blimp

Georgia Tech  
Systems Research

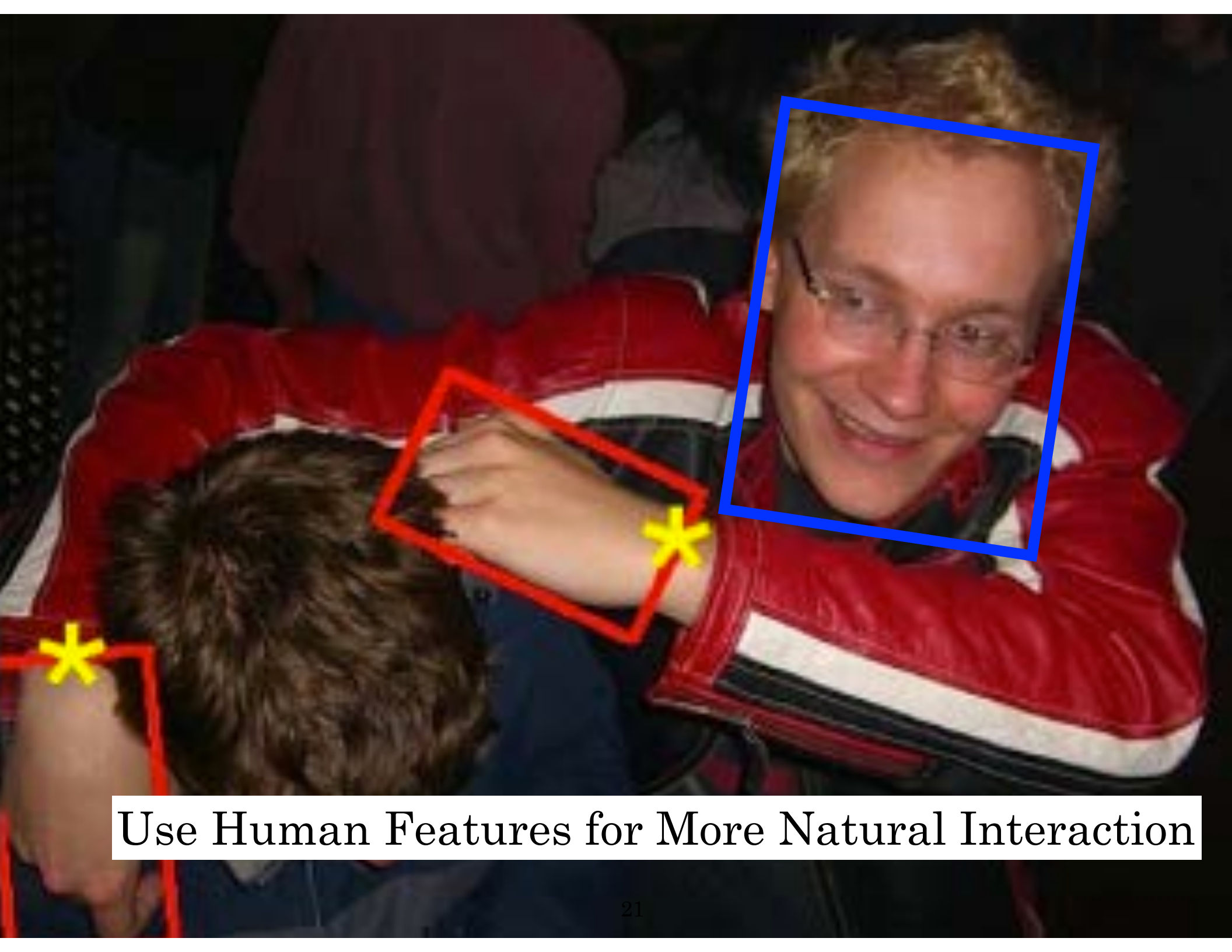


# Swarm Following Human

Georgia Tech  
Systems Research







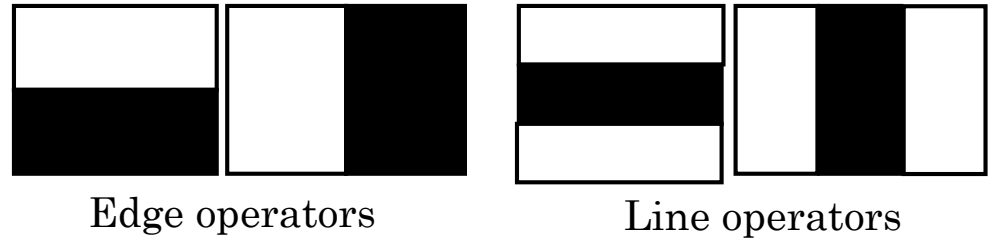
Use Human Features for More Natural Interaction

# Haar Face Detector



Frontal face image set

- Haar operators



- Haar features



feature 1

feature 2

feature 3

$$f_1(I_n)$$

$$f_2(I_n)$$

$$f_3(I_n)$$

# Adaboosting



- Define a classifier for each Haar feature:

$$h_m(I_n) = \begin{cases} 0, & f_m(I_n) < \theta \\ 1, & f_m(I_n) \geq \theta \end{cases} \longrightarrow \text{Learned based on training data}$$

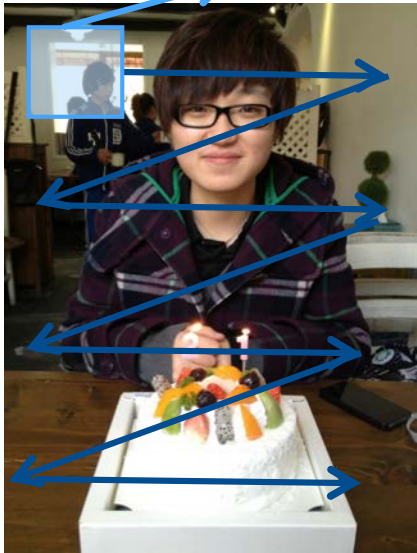
- Strong classifier by Adaboosting:

$$H(x) = \begin{cases} 0, & \sum_{m=1}^M \alpha_m h_m(x) < \frac{1}{2} \sum_{m=1}^M \alpha_m \\ 1, & \text{otherwise} \end{cases} \longrightarrow \text{Learned by Adaboosting}$$

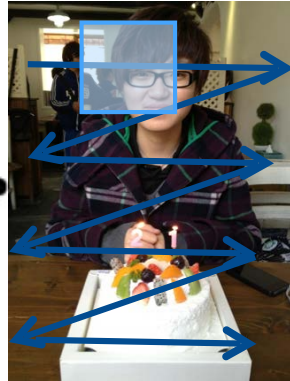
# Face Detection



Search window



...



...



Sub-image  $I$

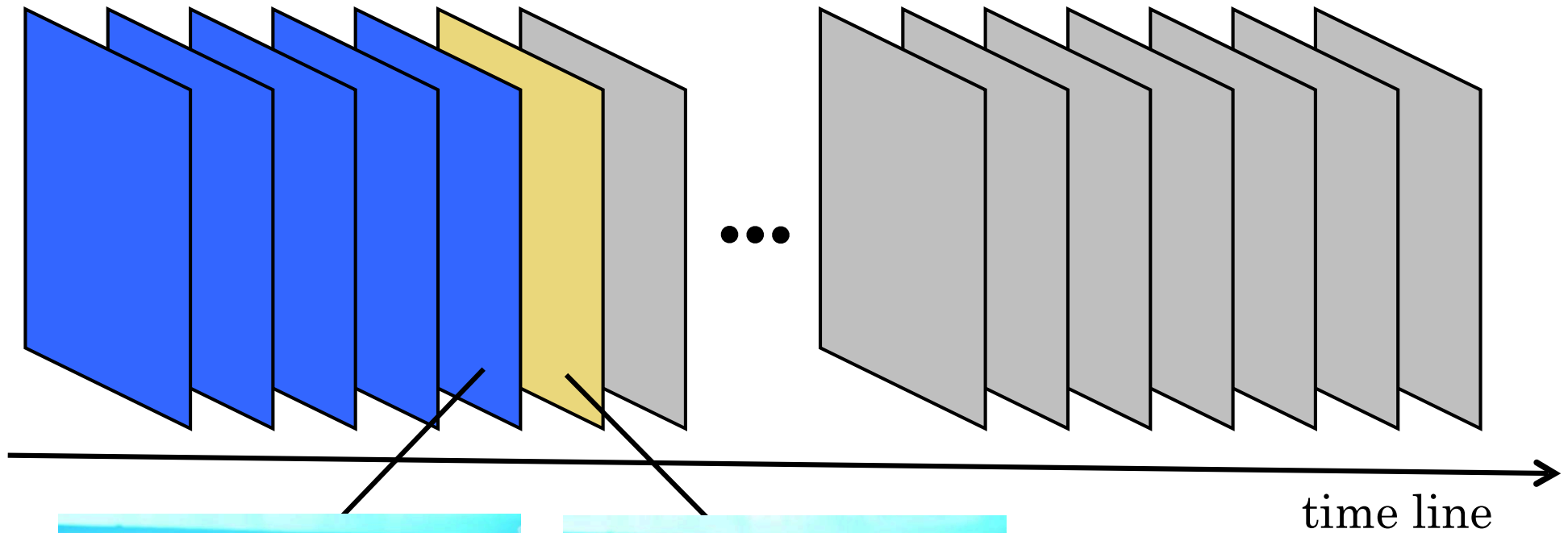


Face detected!

$$H(x) = \begin{cases} 0, & \sum_{m=1}^M \alpha_m h_m(x) < \frac{1}{2} \sum_{m=1}^M \alpha_m \\ 1, & \text{otherwise} \end{cases}$$



# Face Tracking in Real-time Video

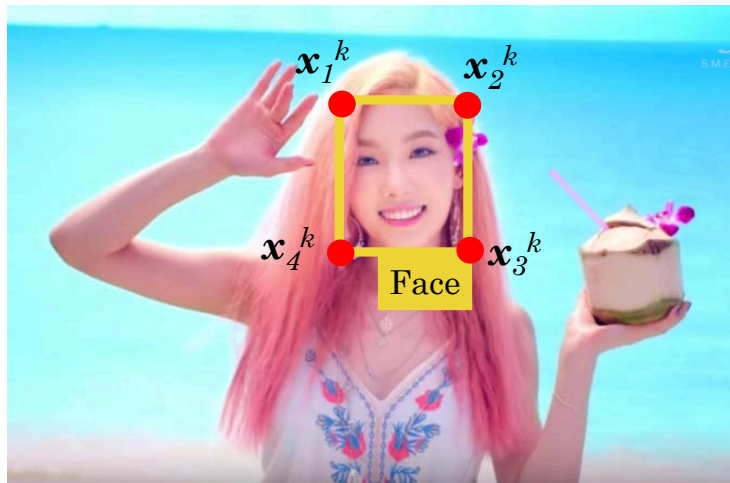


Frame  $I_k$



Frame  $I_{k+1}$

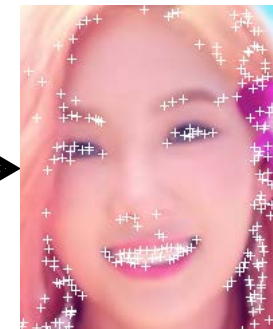
# Extract Feature Point



Frame  $k$



Corner  
points



Coordinates of the corner points of the bounding box:

$$\mathbf{x}_b^k = (i_b, j_b), b = 1, \dots, 4$$

Coordinates of feature point  $i$  in image frame  $k$ :

$$\mathbf{x}_c^F = (i_c, j_c), c = 1, \dots, N_c$$

# Kanade-Lucas-Tomasi (KLT)



**Assumption:** the face in frame  $k+1$  does not move too much compared to frame  $k$ .

Define the displacement vector  $\mathbf{d} = (x_d, y_d)$ , the image model:

$$I_k(\mathbf{x}_i) = I_{k+1}(\mathbf{x}_i - \mathbf{d}) + n(\mathbf{x}_i)$$

Solve the displacement:

$$\mathbf{d}^* = \underset{\mathbf{d}}{\operatorname{argmin}} \sum_{c=1}^{N_F} \sum_{l=1}^9 w_l \left( I_{k+1}(\mathbf{x}_{c,l}^F) - I_k(\mathbf{x}_{c,l}^F) - \mathbf{g} \cdot \mathbf{d} \right)^2$$

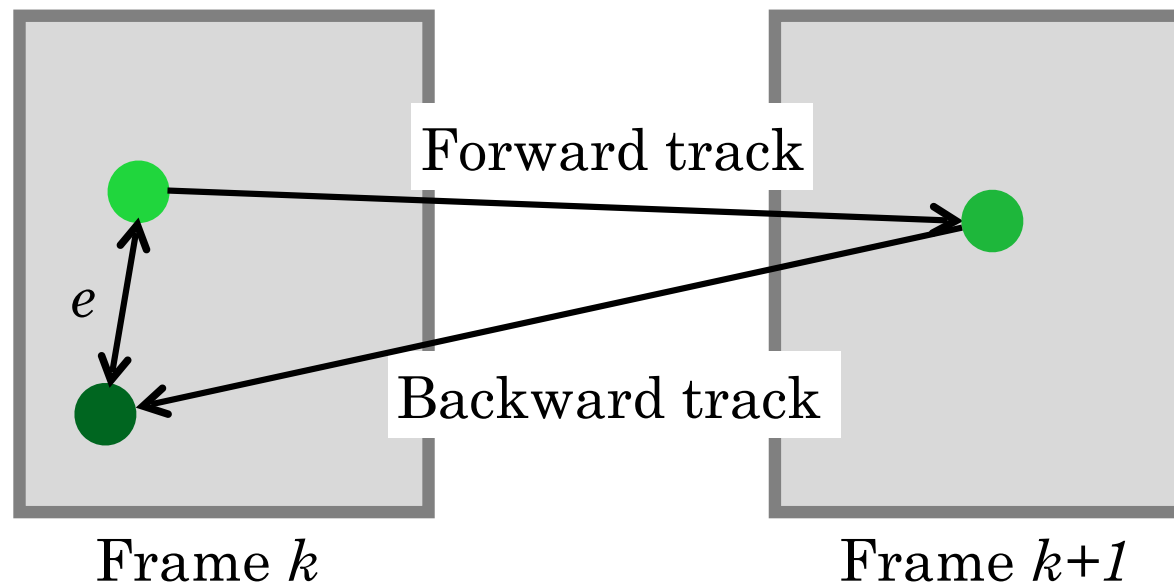
$\mathbf{x}_{c,1}^F$	$\mathbf{x}_{c,2}^F$	$\mathbf{x}_{c,3}^F$
$\mathbf{x}_{c,4}^F$	$\mathbf{x}_{c,5}^F$	$\mathbf{x}_{c,6}^F$
$\mathbf{x}_{c,7}^F$	$\mathbf{x}_{c,8}^F$	$\mathbf{x}_{c,9}^F$

Neighborhood of  
feature point  $\mathbf{x}_c^F$

# Bidirectional Error



What if a feature point disappears in the next frame?



If  $e > \text{threshold}$ , then this point is regarded as lost and has no corresponding feature point in frame  $k+1$ .



# Track Face

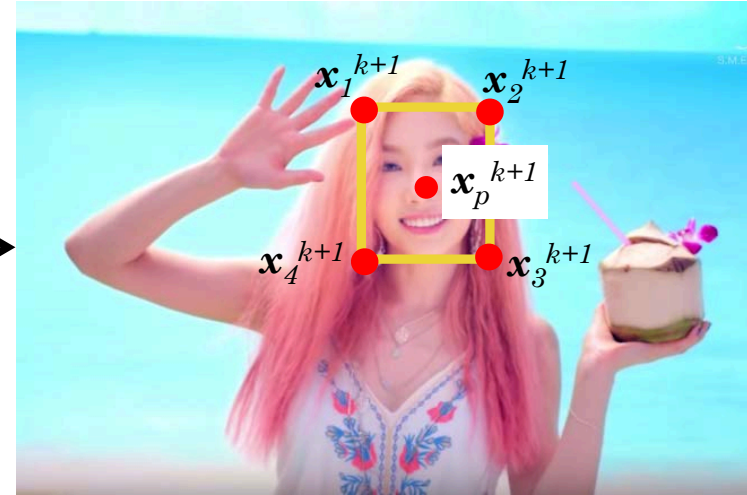


Bounding box of face in frame  $k+1$ :

$$\mathbf{x}_b^{k+1} = \mathbf{x}_b^k - \mathbf{d}^*, b = 1, \dots, 4$$



Frame  $k$



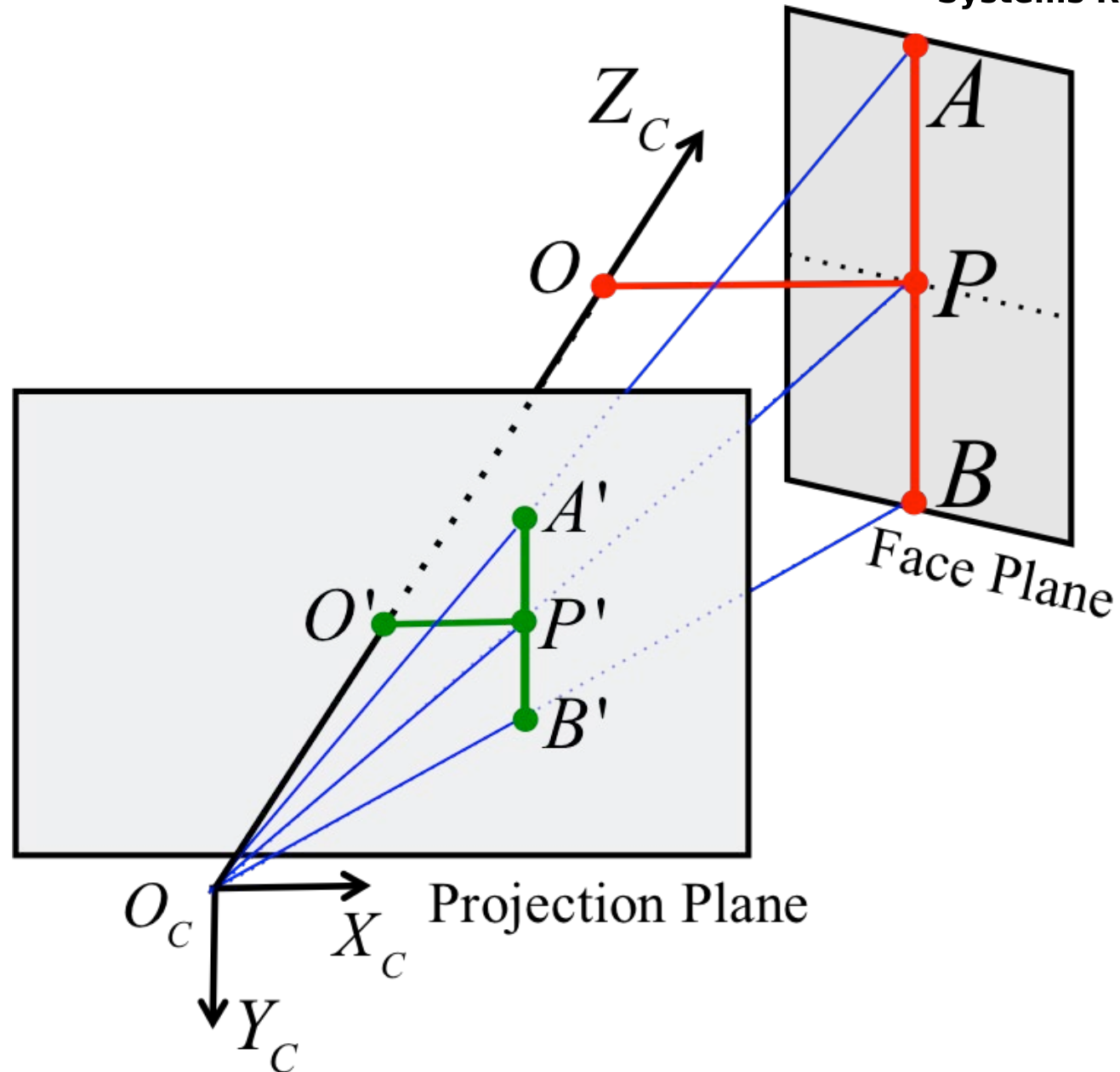
Frame  $k+1$

Face Center:

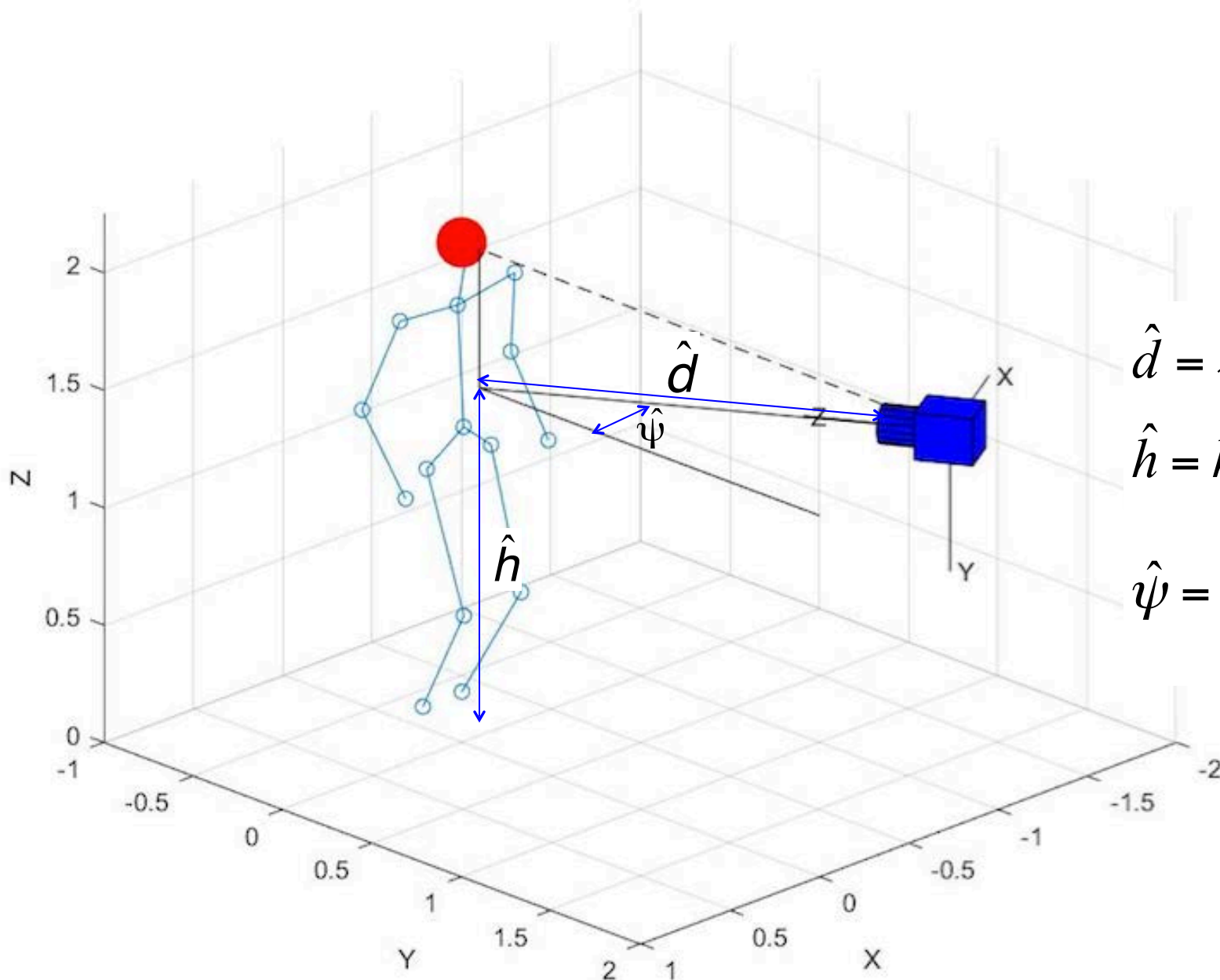
$$\mathbf{x}_p^{k+1} = (i_p, j_p) = \frac{1}{4} \sum_{b=1}^4 \mathbf{x}_b^{k+1}$$

# Human Position Estimation

Georgia Tech  
Systems Research



# Compute Relative Displacement

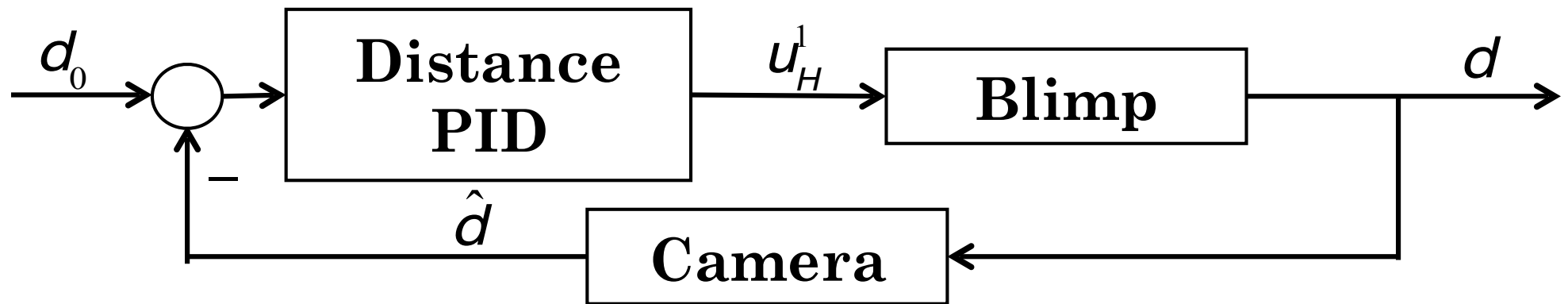


$$\hat{d} = \sqrt{\hat{x}_P^2 + \hat{z}_P^2}$$

$$\hat{h} = h_0 - \hat{y}_P$$

$$\hat{\psi} = \arcsin\left(\frac{\hat{x}_P}{\hat{d}}\right)$$

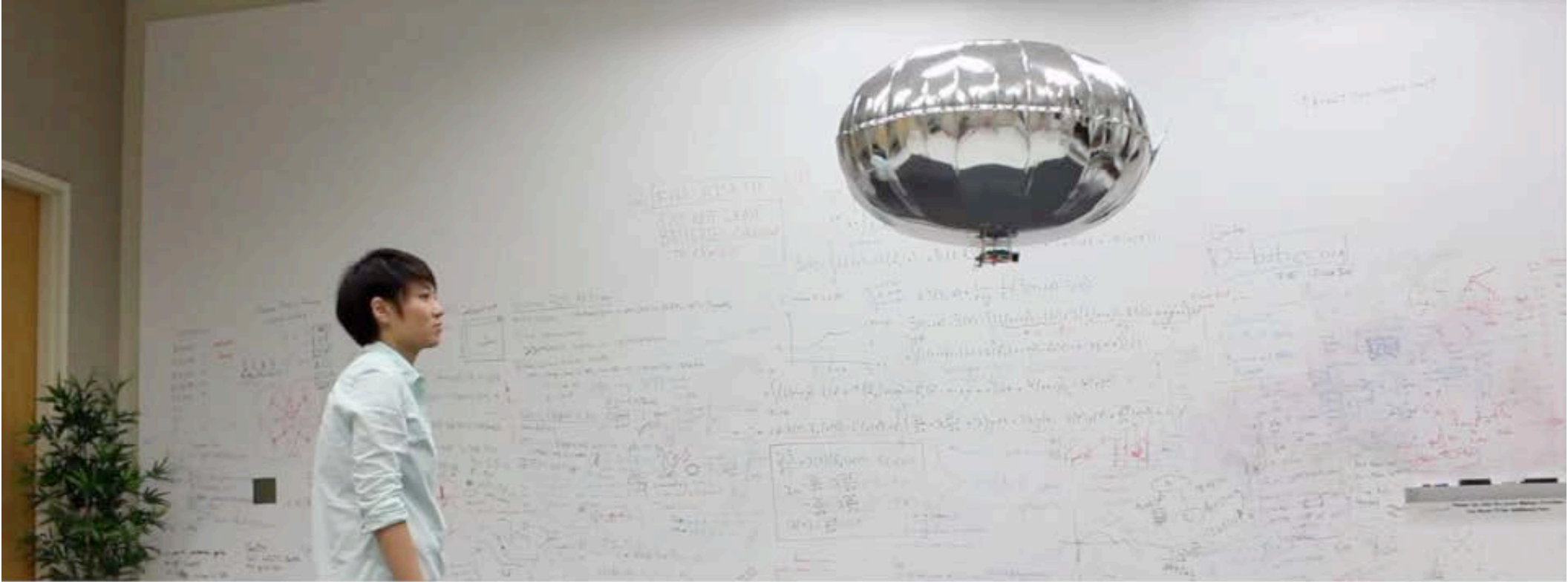
# Reaction





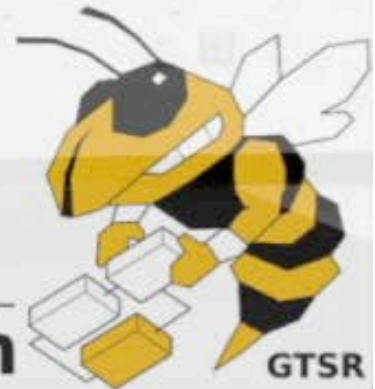
# Blimp Tracking Human

Georgia Tech  
Systems Research

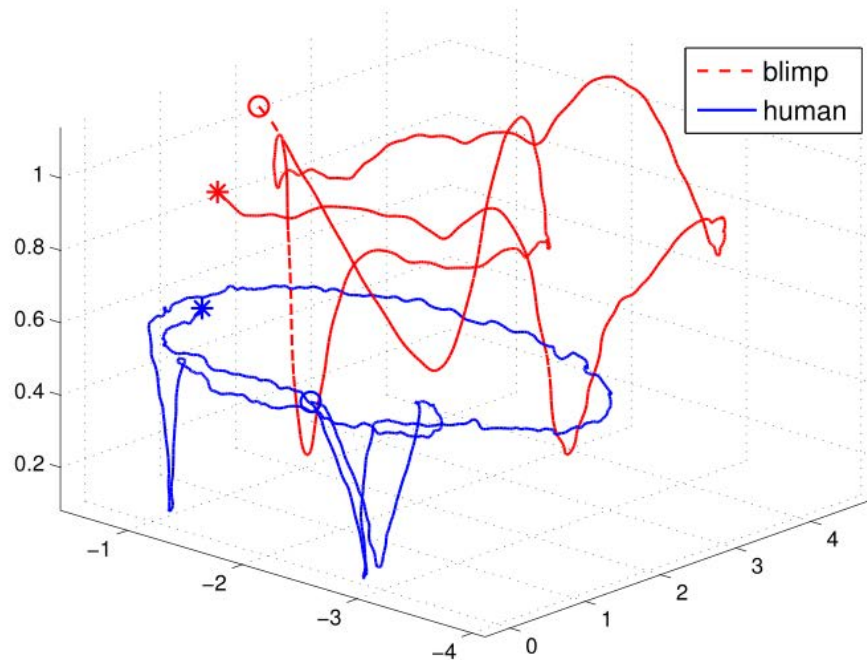


*Presented by*

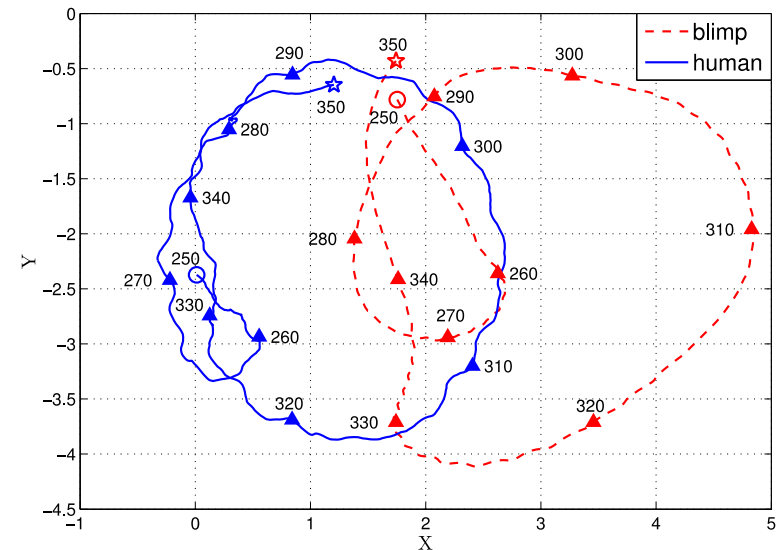
Georgia Tech  
Systems Research



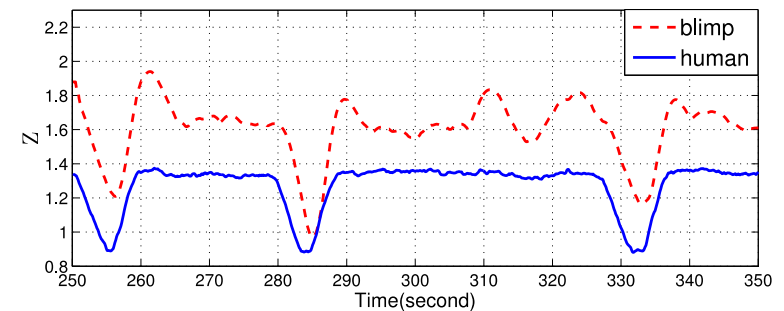
# Experiment Results



3D view of the blimp and human trajectories



Top view of the blimp and human trajectories



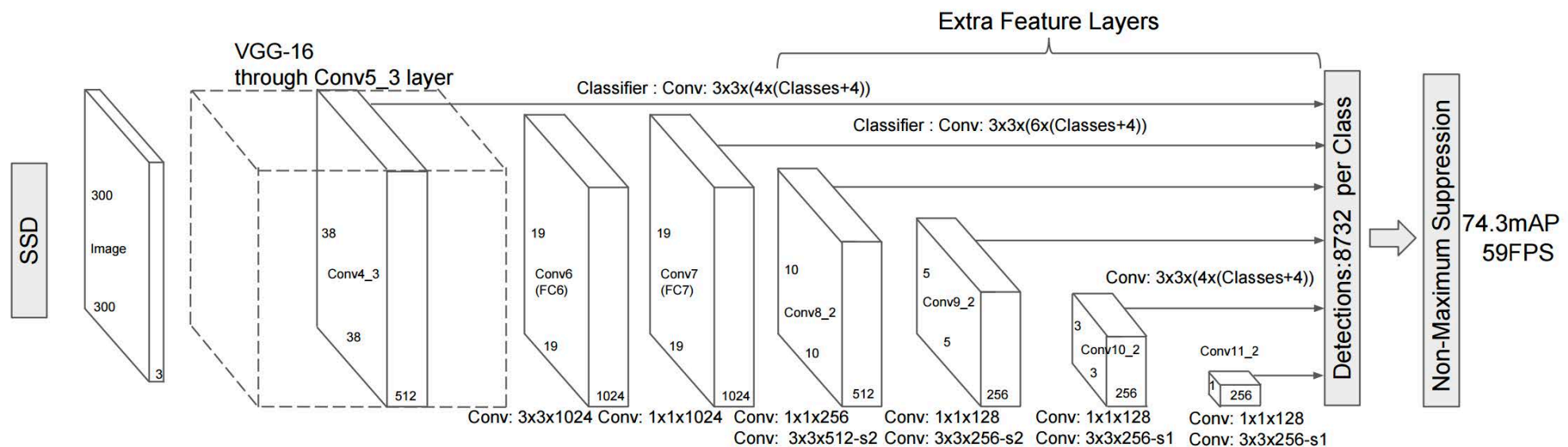
Height of the blimp and human

# Hand Detection



Detect human face and hands at the same time to localize human and recognize human's intention.

Use Single-Shot multibox Detector (SSD), which is able to detect different kinds of objects at the same time.

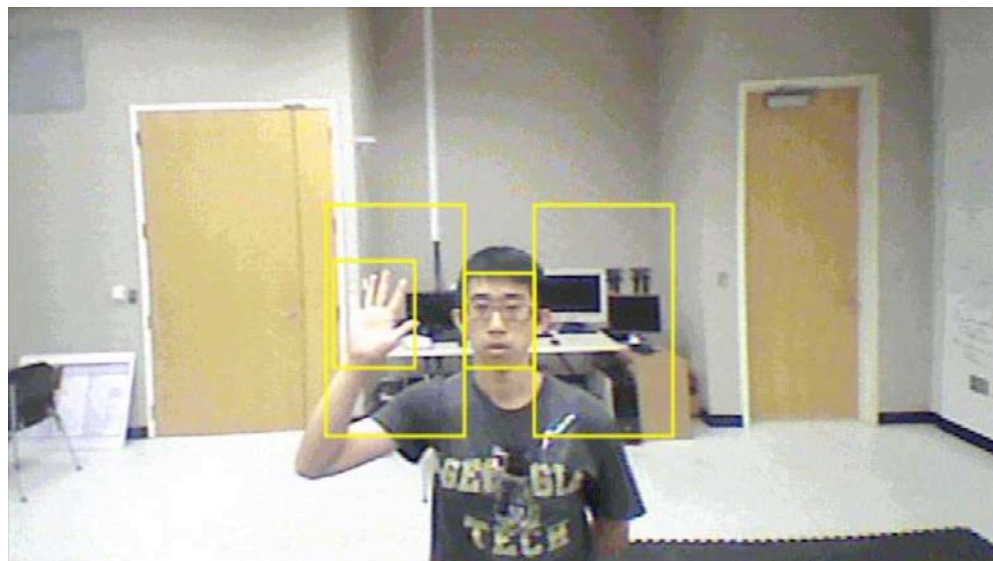




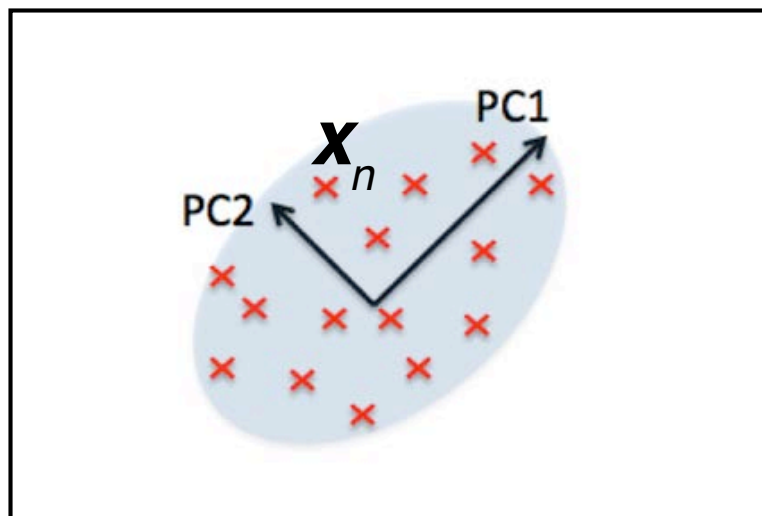
# Two Gestures



Horizontal hand movement

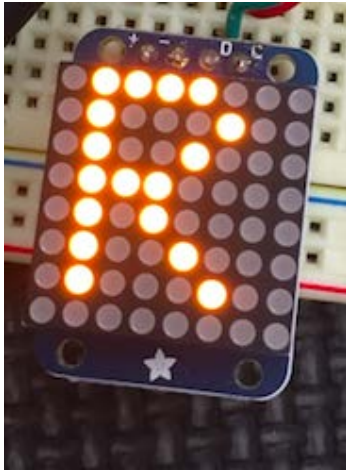


Vertical hand movement



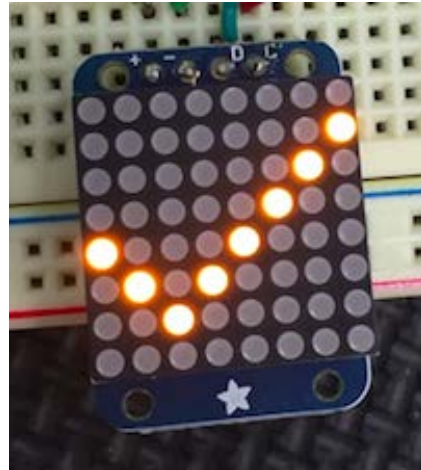


# Blimp's Intentions

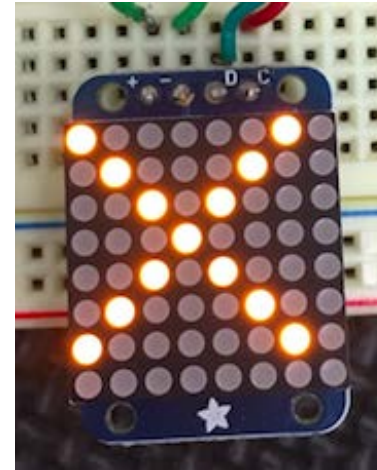


Face  
detected.

Put hand  
near face.



Hand  
detected.



No hand  
detected.



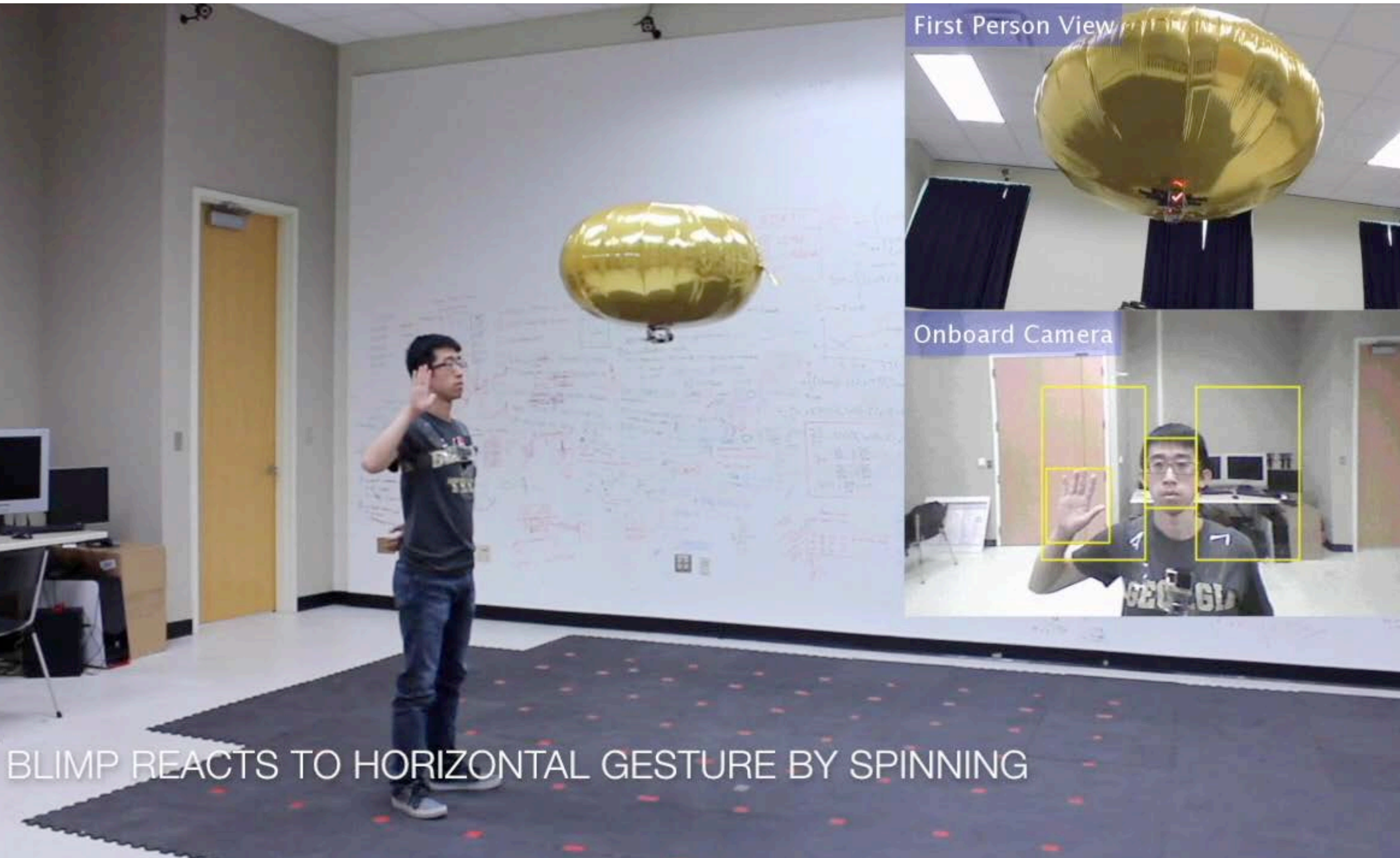
Gesture is  
recognized.

Blimp will  
react.

LED Display: Low Power, Light Weight.

# Spinning Movement

Georgia Tech  
Systems Research



First Person View



Onboard Camera



BLIMP REACTS TO HORIZONTAL GESTURE BY SPINNING

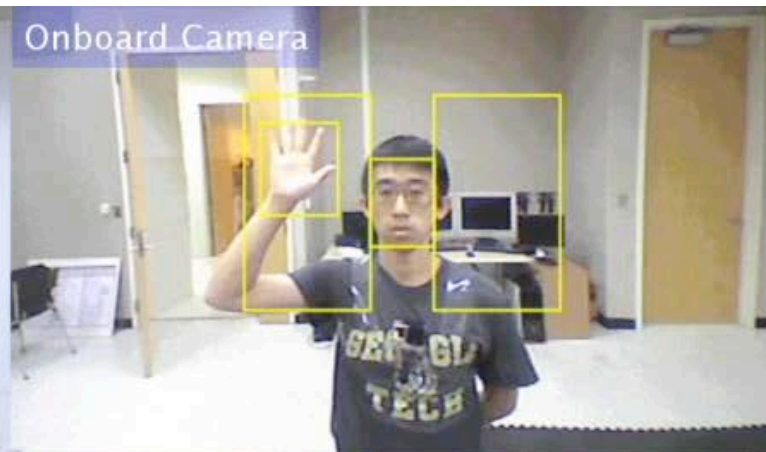




First Person View



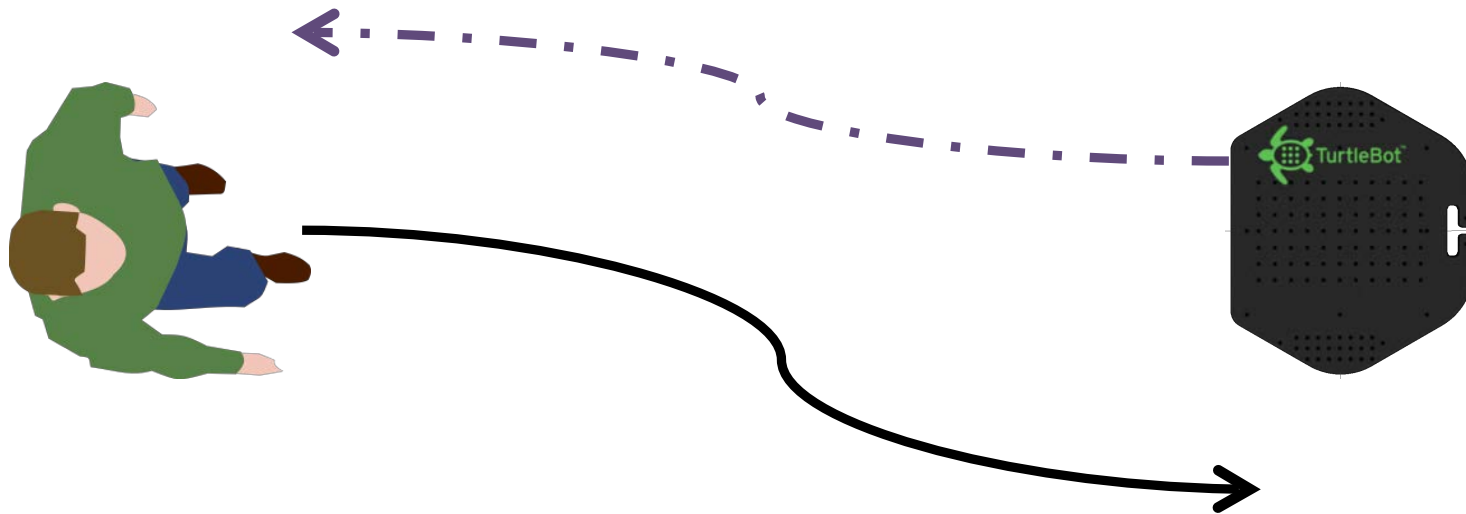
Onboard Camera



BLIMP REACTS TO VERTICAL GESTURE BY MOVING AWAY AND BACK

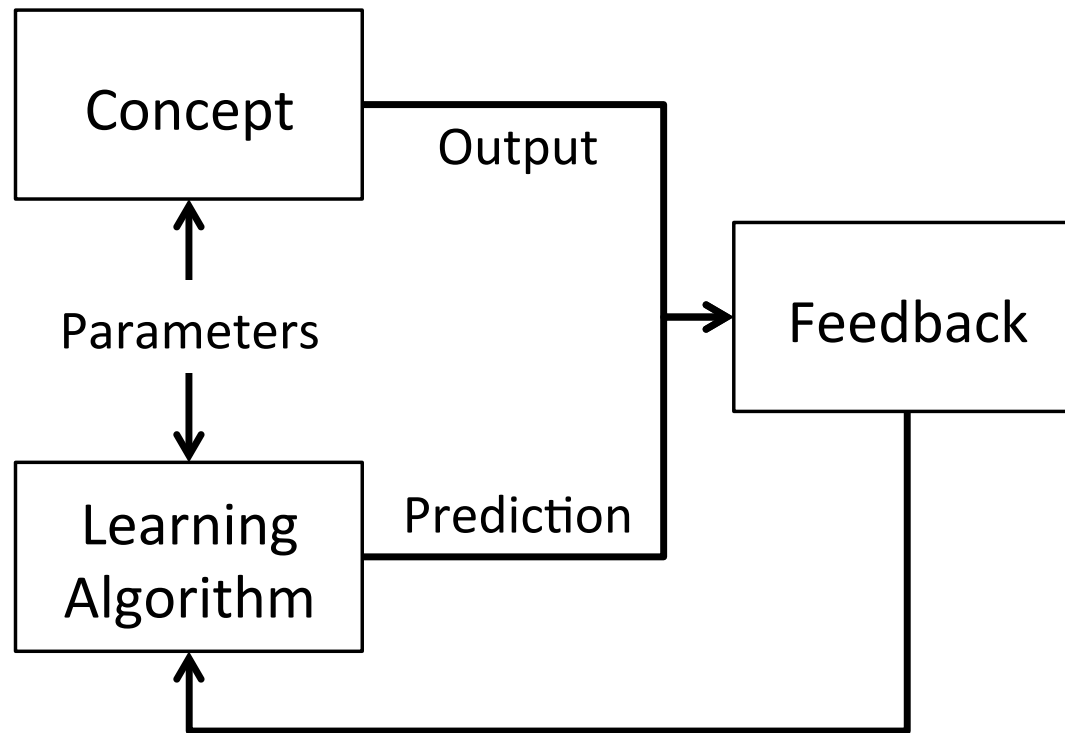
# Predicting Human Intention

Georgia Tech  
Systems Research



A simple problem: Left or Right?

# Online Iterative Learning



**Adaptiveness:** Expected number of iterations before updating prediction after *permanent* concept change (drift)

**Consistency:** Expected number of iterations before updating predictions after *temporary* concept change (noise)



# Consistency



**Inconsistent (More Switching)**



**Consistent (Less Switching)**



# Two Expert Learning



Weight  $W_1$



Weight  $W_2$

Expert with higher weight wins.  
If an expert is wrong, its weight is cut by half.

# Weighted Majority vs. Winnow



*Weighted Majority Algorithm:*

If an expert is correct, its weight does not change.

*Winnow Algorithm:*

If an expert is correct, its weight doubles.

*Dual Expert Algorithm:*

If an expert is correct, its weight doubles but the maximum weight is 0.5.

**Which algorithm is more adaptive, which one is more consistent?**

# Comparing Algorithms



$$\begin{array}{lcl}
 \text{WMA} & \frac{w_1}{w_2} & \frac{0.5}{0.5} \uparrow \\
 \text{Winnow} & \frac{w_1}{w_2} & \frac{0.5}{0.5} \uparrow \\
 \text{DEA} & \frac{w_1}{w_2} & \frac{0.5}{0.5} \uparrow
 \end{array}$$

- Expert 1 ( $w_1$ ) predicts up
- Expert 2 ( $w_2$ ) predicts down
- Results recorded below as  $R = w_1 / w_2$

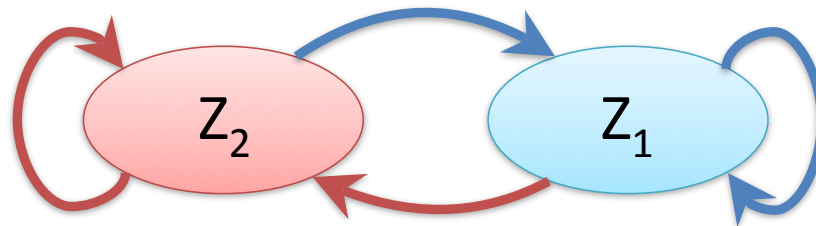
WMA
Winnow
DEA

Error	Switch
0	0
0	0
0	0



# Markov Chain for WMA



*Random Switch Model:*



$$\mathbb{P}_M = \begin{bmatrix} p_1 & p_2 \\ p_1 & p_2 \end{bmatrix}$$

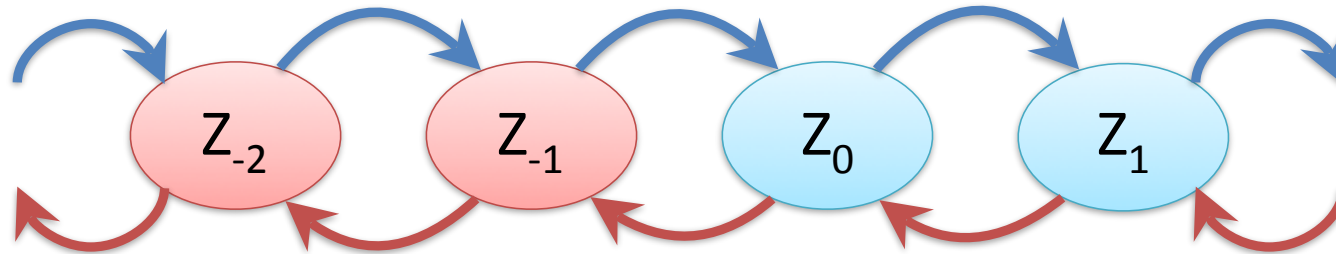
- $Z_1$  :  $R=1$ , algorithm predicts 1
- $Z_2$  :  $R = \frac{1}{2}$ , algorithm predicts 2
- $p_1$  :  Probability that prediction 1 is correct
- $p_2$  :  Probability that prediction 2 is correct



# Markov Chain for Winnov



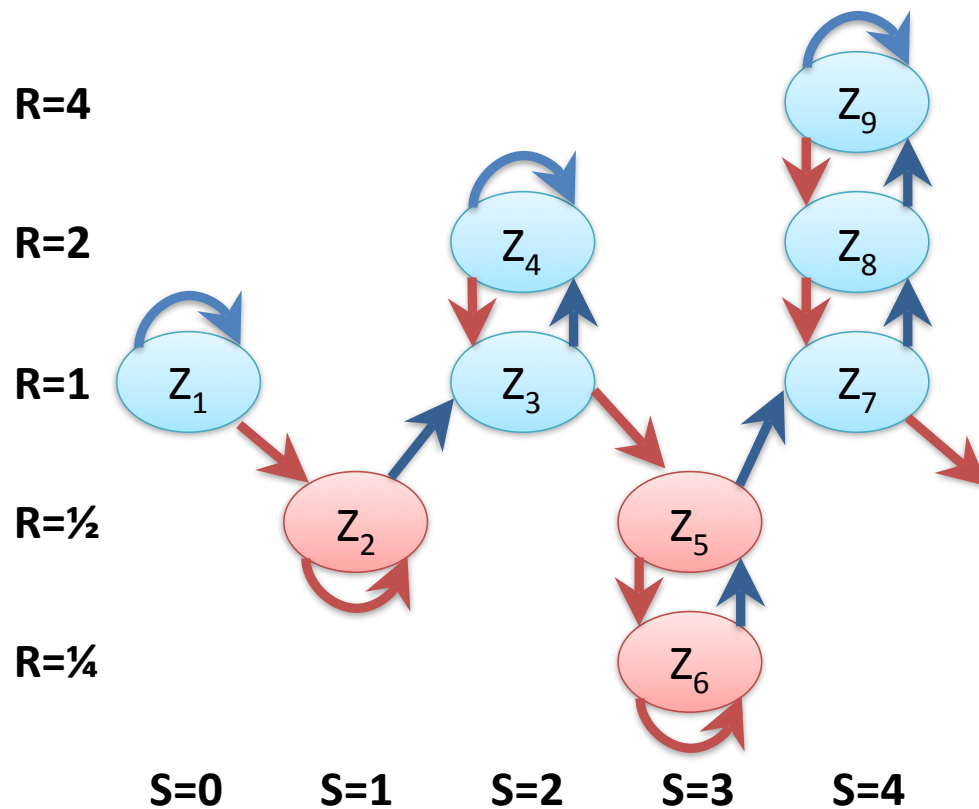
*Random Walk Model:*



- $Z_i : R=2^i$   
 $-i \geq 0$  predicts 1  
 $-i < 0$  predicts 2

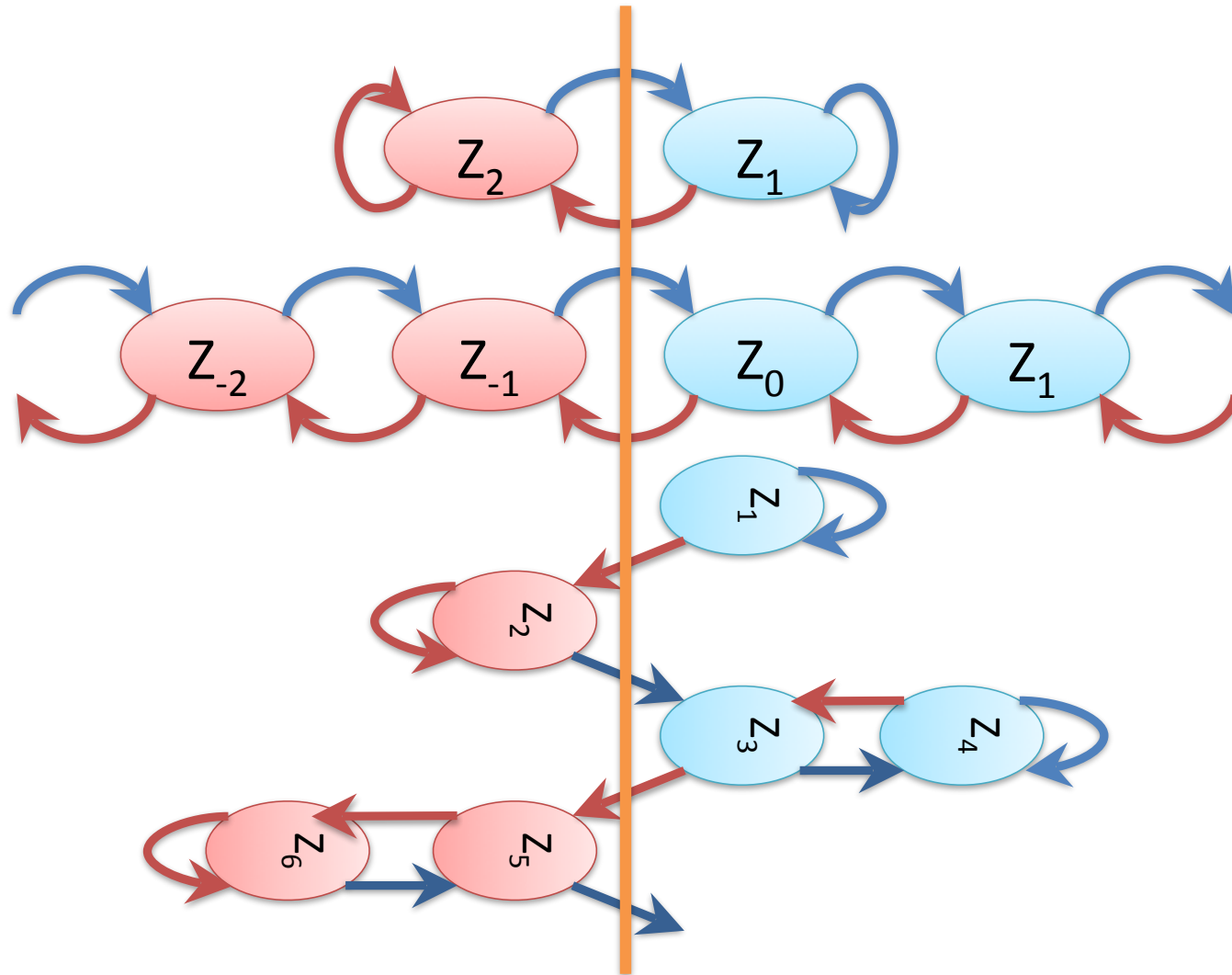
$$\mathbb{P}_W = \begin{bmatrix} \cdot & \cdot & \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot \\ \dots & 0 & p_1 & 0 & 0 & 0 & \dots \\ \dots & p_2 & 0 & p_1 & 0 & 0 & \dots \\ \dots & 0 & p_2 & 0 & p_1 & 0 & \dots \\ \dots & 0 & 0 & p_2 & 0 & p_1 & \dots \\ \dots & 0 & 0 & 0 & p_2 & 0 & \dots \\ \cdot & \cdot & \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot \end{bmatrix}$$

# MC for DEA



$$\mathbb{P}_D = \begin{bmatrix} p_1 & p_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & p_2 & p_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & p_1 & p_2 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & p_2 & p_1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & p_2 & p_1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & p_1 & p_2 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_1 & 0 & p_2 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_2 & 0 & p_1 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_2 & p_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

# Switching Manifold



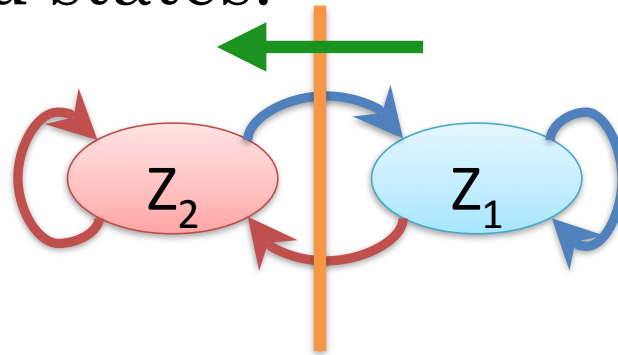
# Adaptiveness and Consistency



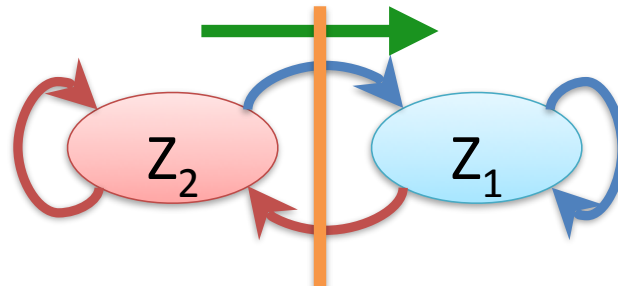
Assume “1” (blue state) is the intention e.g.

$$p_1 > p_2$$

**Adaptiveness:** Reciprocal of *mean hitting time* from blue states to red states.



**Consistency:** *Mean hitting time* from red states to blue states.



# Comparison



*Weighted Majority Algorithm:*

Adaptiveness:  $p_1$

Consistency:  $\frac{1}{p_2}$

*Winnow Algorithm:*

Adaptiveness:  $p_1 - p_2$

Consistency:  $\infty$

*Duel Expert Algorithm:*

Adaptiveness:

$$\frac{p_1 - p_2}{1 - \left(\frac{p_2}{p_1}\right)^{n+1}}$$

Consistency:

$$\frac{1}{p_2} \frac{1 - \left(\frac{p_1}{p_2}\right)^{n+1}}{1 - \frac{p_1}{p_2}}$$



# Consistant Adaptive Online-Learning for Robots Making Binary Choices

Carol Young and Fumin Zhang

Front facing view  
from turtlebot  
mounted webcam



EDEA weights  
Blue ~ right  
Red ~ left

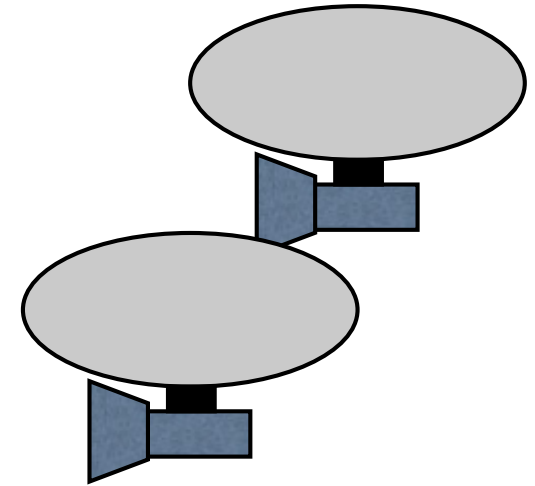
# Result Analysis



Test	Error Rate	Switch Rate
<b><i>Experiment Average</i></b>	<b>35%</b>	<b>12.5%</b>
Sim 5%	21.8%	9.8%
<b><i>Sim 10%</i></b>	<b>27.2%</b>	<b>10.7%</b>
Sim 30%	42.2%	12.1%



# Multiple Blimps



# Stereo Vision Setting



Measurements:

Camera position and pose

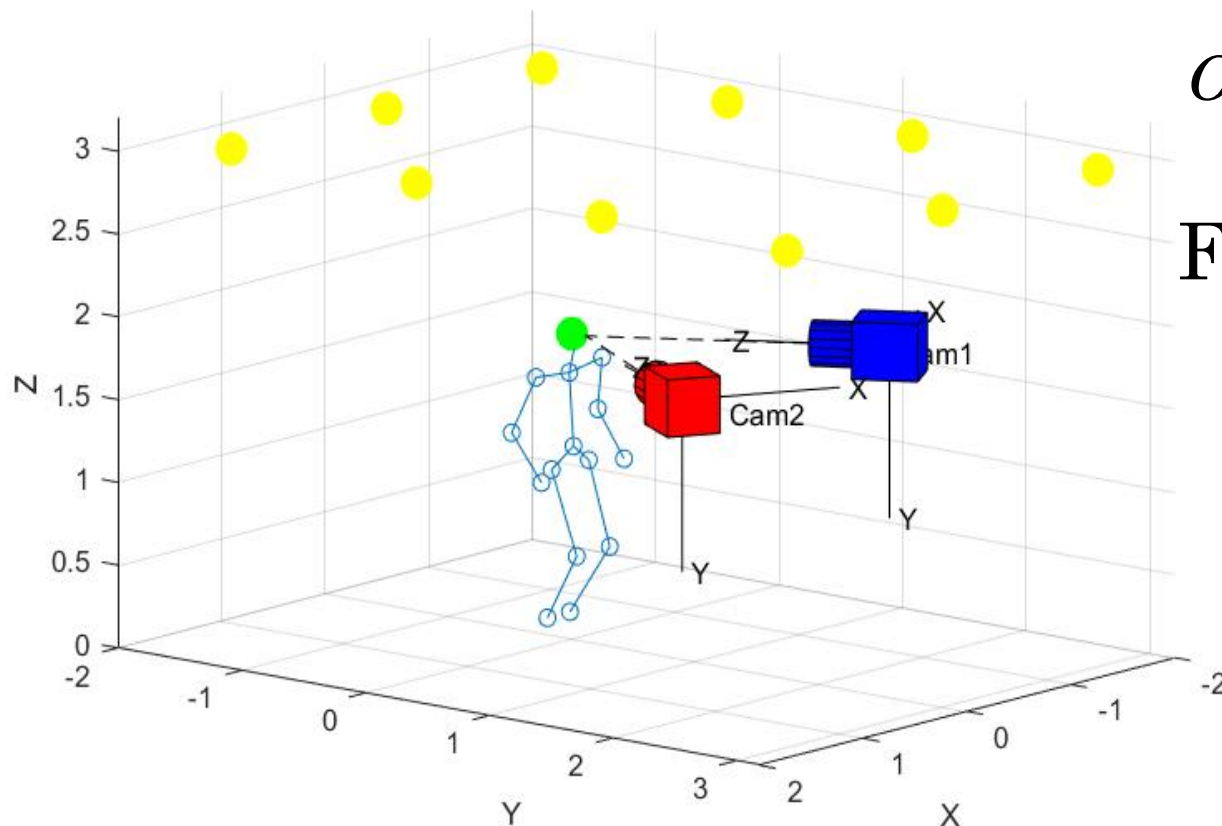
$$O_L = [x_{O_L}, y_{O_L}, z_{O_L}, \phi_{O_L}, \theta_{O_L}, \psi_{O_L}]^T$$

$$O_R = [x_{O_R}, y_{O_R}, z_{O_R}, \phi_{O_R}, \theta_{O_R}, \psi_{O_R}]^T$$

Face location in images

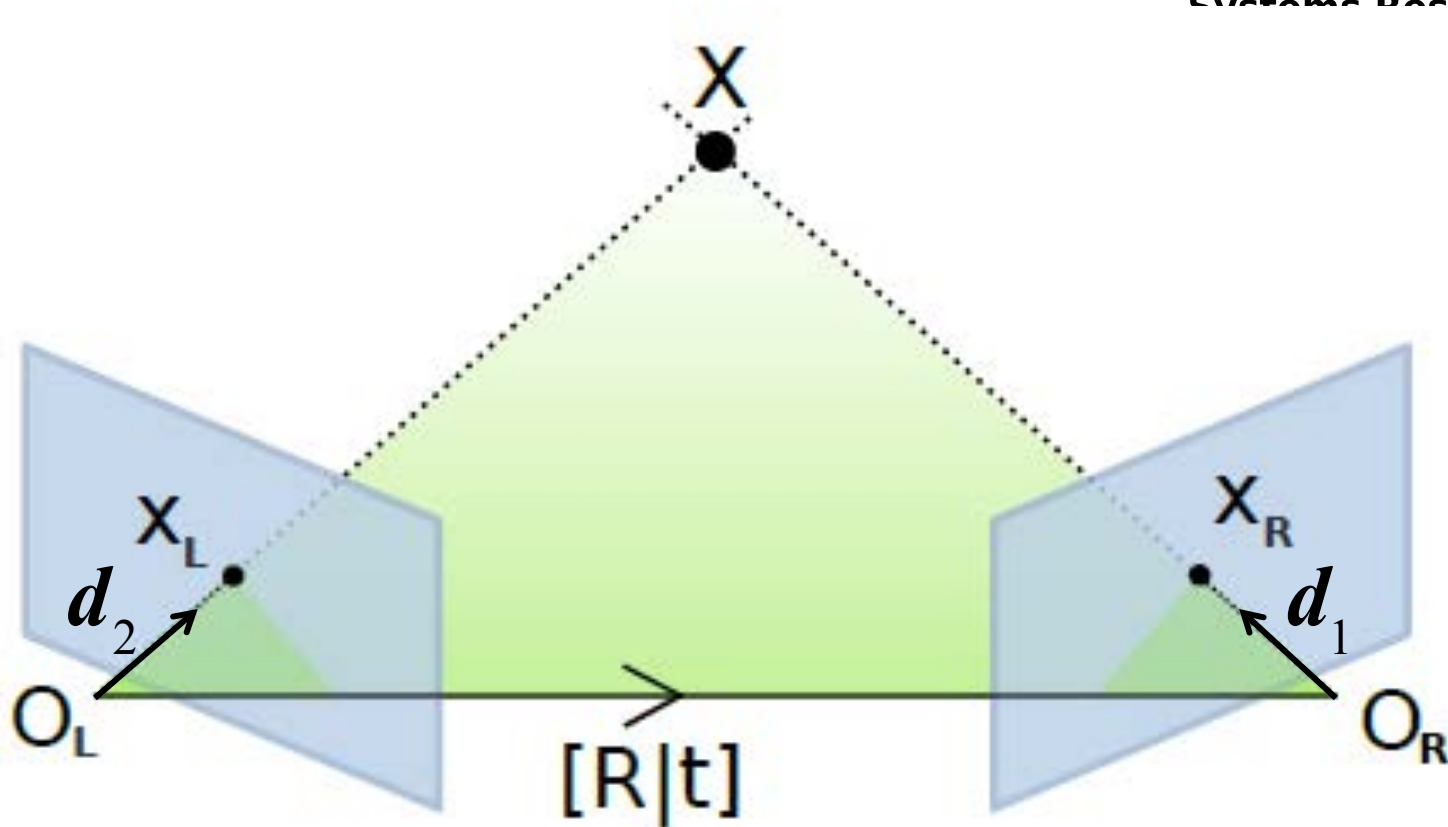
$$X_L = [u_L, v_L, 1]^T$$

$$X_R = [u_R, v_R, 1]^T$$





# Localize the Human



Cam 2

Cam 1

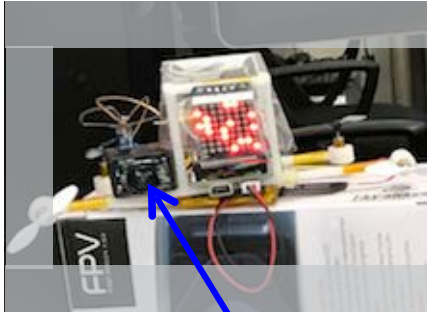
$$X_1 = \alpha_1 d_1 + O_R$$

$$X_2 = \alpha_2 d_2 + O_L$$

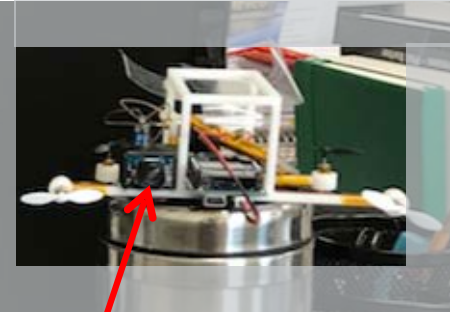
Find  $(\alpha_1^*, \alpha_2^*) = \underset{\alpha_1, \alpha_2}{\operatorname{argmin}} \|X_1 - X_2\|_2$

# Preliminary Results

Georgia Tech  
Systems Research



Camera 1



Camera 2



# Conclusion



Recognizing and predicting human intention is very important for unmanned system to generate proper reaction.

Lack of reliable model is the major challenge.

Integration of data-driven methods and analytical methods is necessary.