Human Gaze Commands Classification: A Shape Based Approach to Interfacing with Robots

PRESENTER/AUTHOR: TREVOR LYNN CRAIG

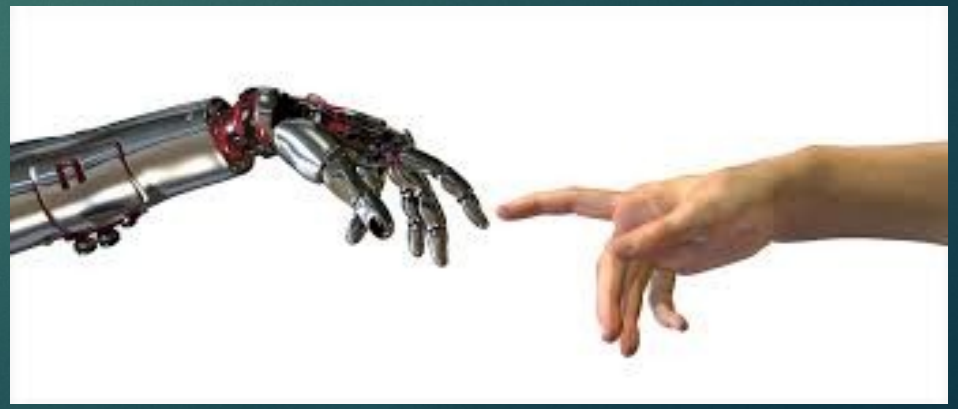
AUTHORS: CARL A. NELSON, SONGPO LI, XIAOLI ZHANG

Abstract

- Severely mobility-limited individuals or lack of motor control
- Eye tracking methods allow for robotic commands
  - Gazepoint GP3 Tracker
- Tracking method chosen to be based on shapes
- Data processed using custom algorithms in MATLAB
- Data passed to NAO

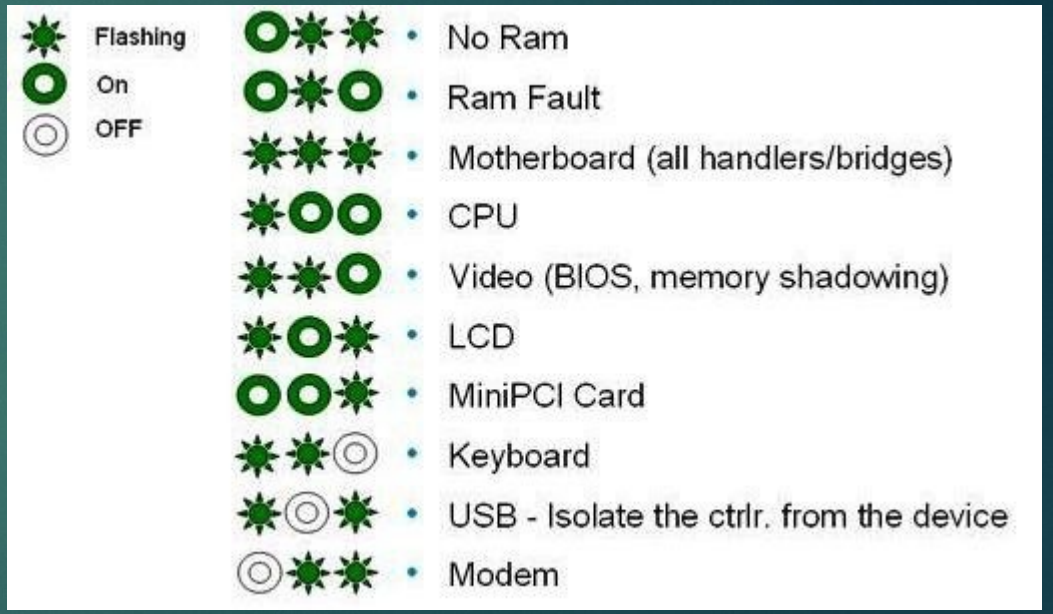
Background

- Language barrier between human commands and robotic comprehension
  - Human Robot Communication and Interaction (HRI)
  - Command interfaces: keyboard, mouse, and voice.
- Challenge: users have limited command inputs
- Gaze-command robots could be used to assist severely mobility-limited individuals or lack of motor control
  - Injury and/or degenerative disease to nervous system
  - Mute, paralyzed, confined patients



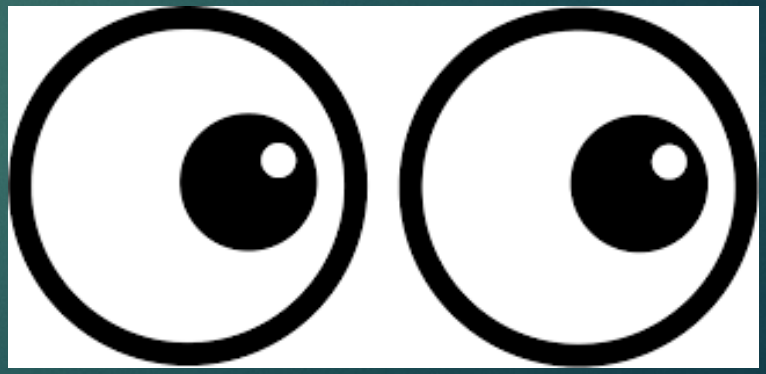
Possible Solutions: Blink Based

- Timing and number of blinks
  - Blink duration varies
  - Involuntary blinks
    - Re-centering eyes after blinking
  - Number of blinks per minute vary on situation
- Blink code complexity increases with commands
  - Computers communicate this way as a last resort
  - Eyes allow 2 bit input
    - Eyes full open or closed not optimal
    - Winking



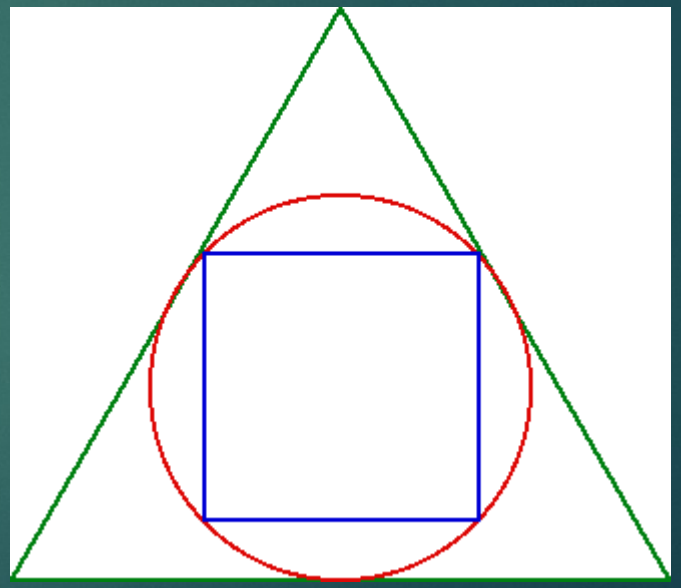
Possible Solutions: Strictly Gazed

- Eyes focus on one point to issue interest or command
  - Stare off into space
  - Focusing on object for other purpose
    - Reading, watching birds, examining wall texture.
- Rapid eye movements between objects (saccades)
  - Varies in amount based on situation
- Blinking
- When are commands being inputted?
  - When do commands stop?



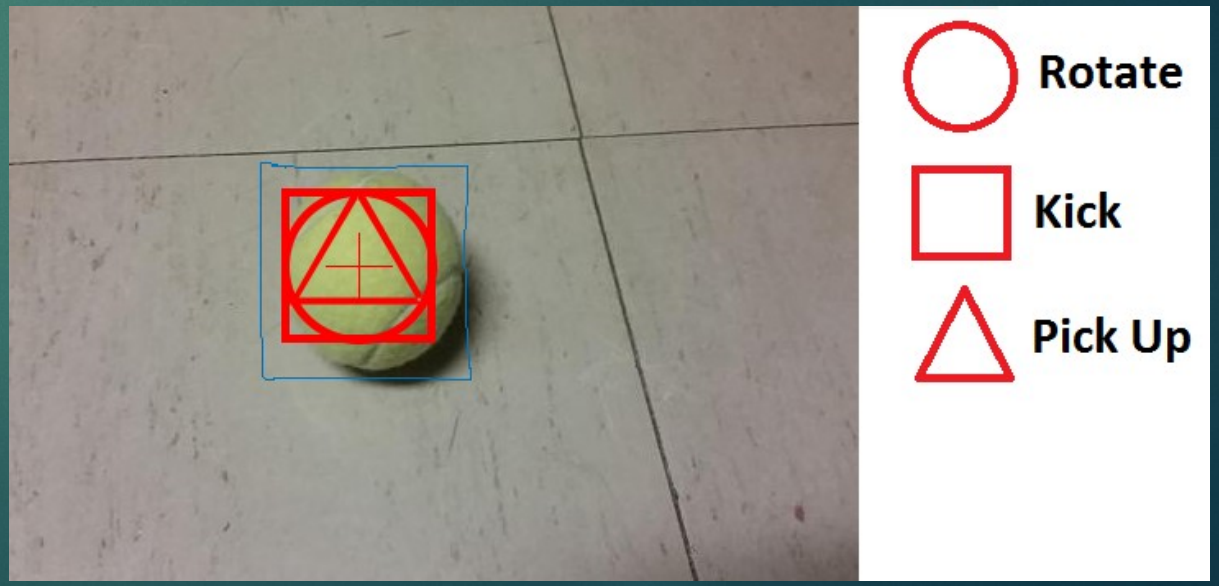
Solution: Shape Based Method

- Commands are easily recognizable
  - Circle, Triangle, and Square
- More easily remembered than blinks
- Can still use blinks
- Can vary in size and shape for options
- Also location independent
- Clear start and finish



General Process

- Gazepoint GP3 Eye Tracker (60 Hz)
- Software converts to x and y pixel coordinates
- Program detects shape
- Does programmed action dependent on state.



State Machine

- Context specific commands
  - Location and object recognition (NAO)
- Kitchen scenario
  - Recognizes empty glass, issues set of commands
    - Fill the glass (circle)
    - Put the glass in the cupboard (triangle)
    - Bring me the glass (square)

Program Methods

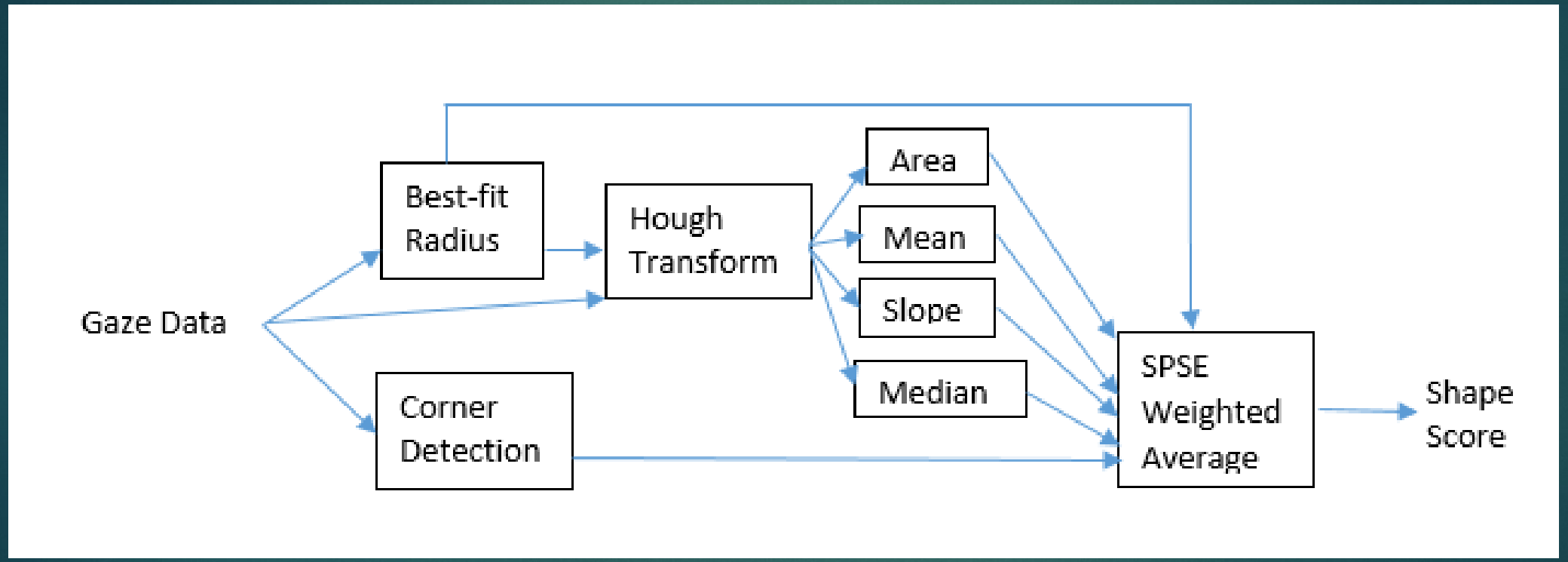
- There is a settings file for each user
- Individual testing contribute to a finalized score
  - Shape Points Score Estimation (SPSE)
- SPSE is made up of 6 sub-scores
  - Best fit radius
  - Corner Detection
  - Area of Hough Transform
  - Mean of Hough Transform
  - Slope of Hough Transform
  - Median of Hough Transform

$$SPSE = \sum_{i=1}^{n} w_i \left(1 - \frac{M_{i,actual} - M_{i,ideal}}{M_{i,ideal}}\right)$$

M_i is shape matching normalized metric
w_i is the individual weights

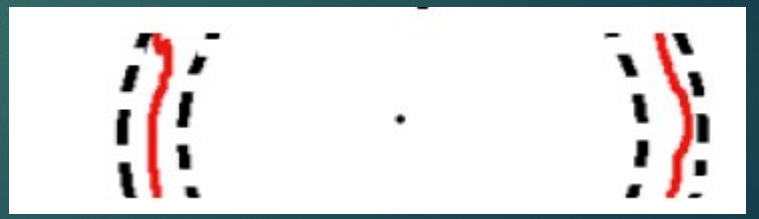
WEIGHTS USED FROM CALIBRATION						
Metric	Radius	Corner	Area	Mean	Slope	Median
Weights	10	3	4	4.5	3	3

Program Flow Chart



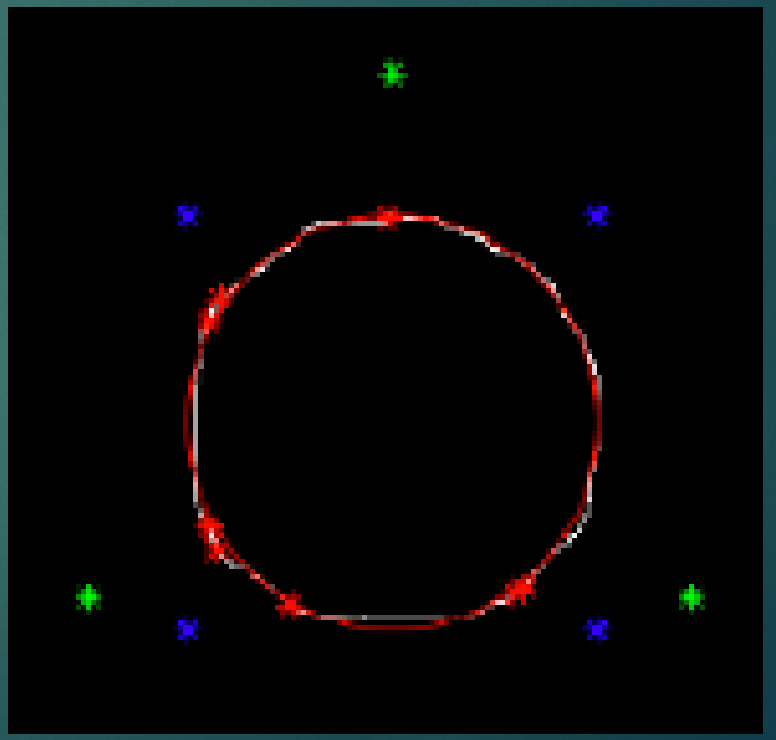
Best Fit Radius

- X and Y data is ran across a radius filter that tries shape outlines
  - Upper and lower bounds
  - Counts points within the shape profile
    - Bounds can be set dependent on user
- SPSE score is updated
- Best shape radius detected is passed to the main program



Corner Finding

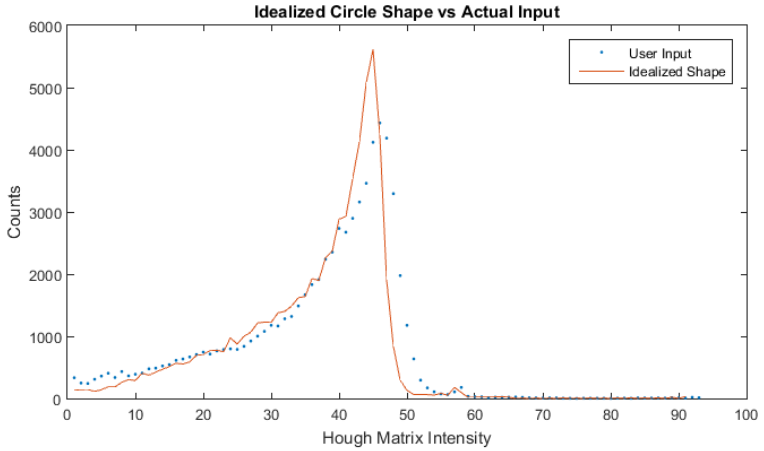
- Harris corner detection build into MATLAB
- Checks corner location to Radius location
- Outputs a sub-score for the SPSE

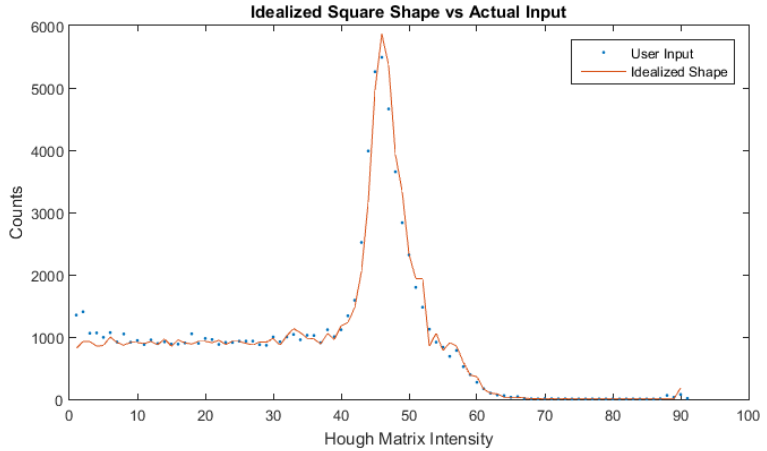


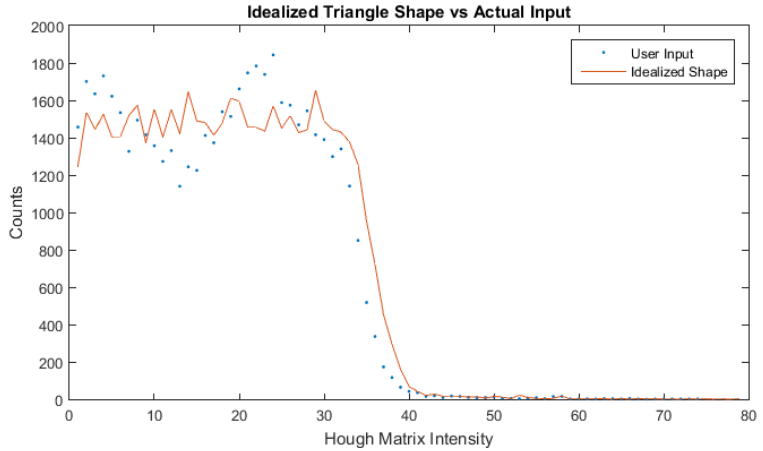
Hough Transform

- Designed to detect lines in a binary image
  - $\rho = x*\cos(\theta) + y*\sin(\theta)$
  - Histograms to show the matrix intensity
- All radii for different shape basis preprocessed
  - Faster computation
  - Can compare user data with “perfect” shape data

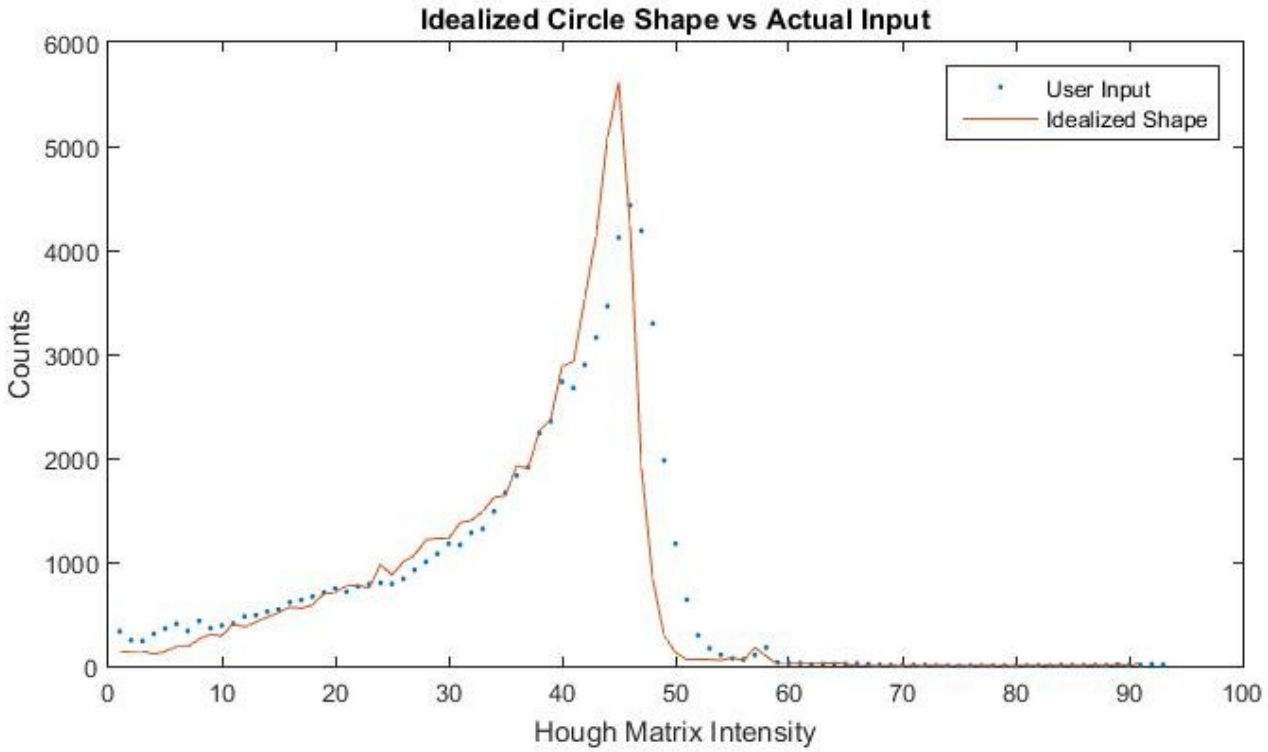
Graph Results



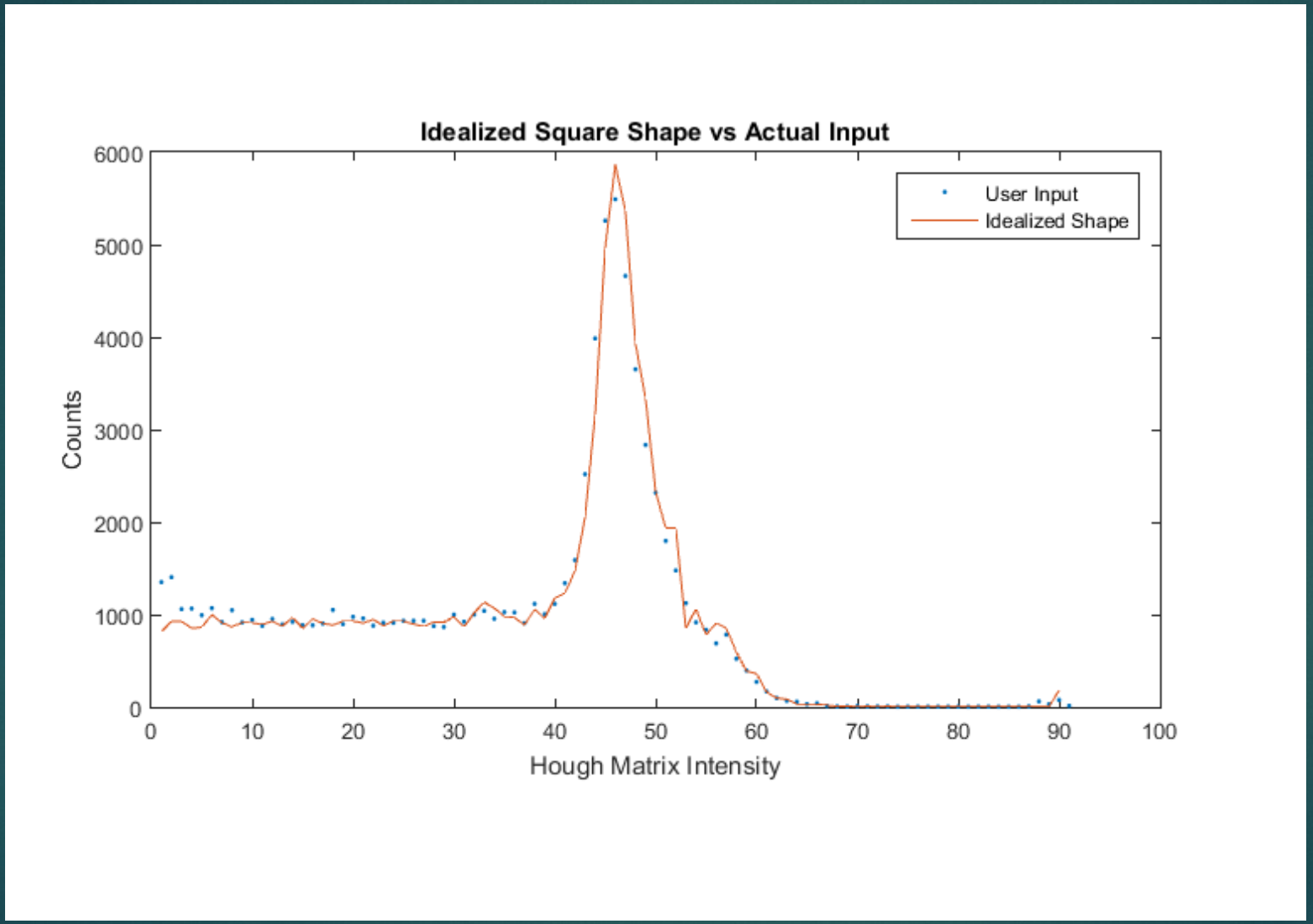




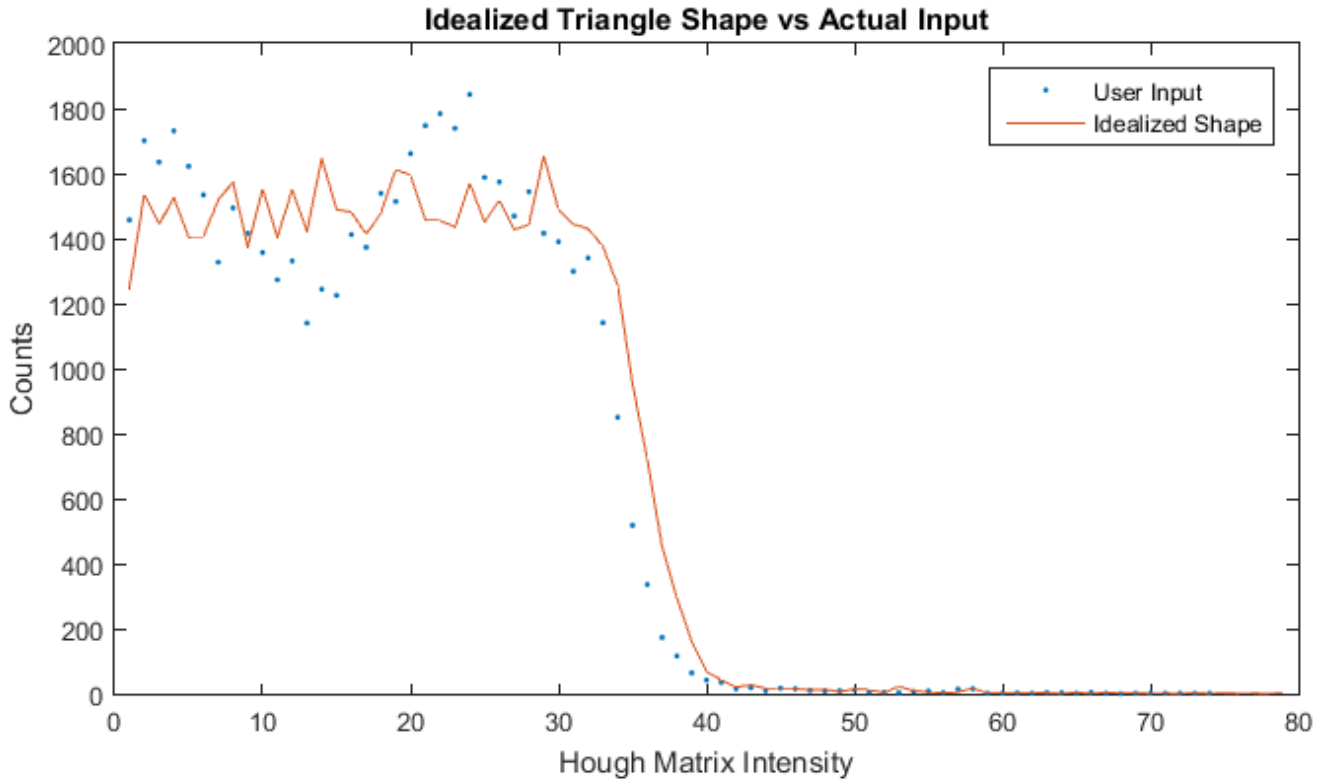
Circle Graphs



Square Graphs

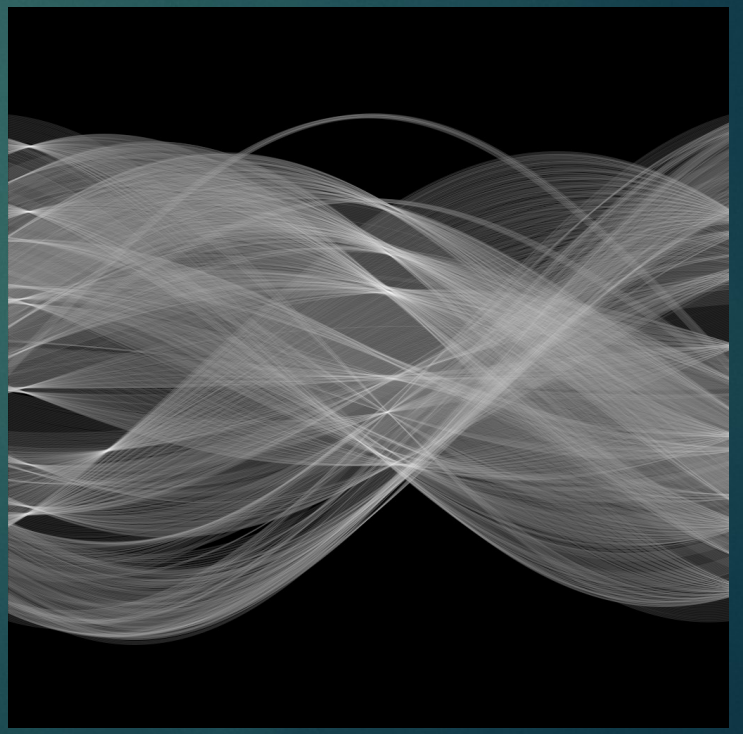


Triangle Graphs



Hough Method Metrics

- Area of Hough Transform
  - Extra area from roughness leads to more detection of circles
- Mean of Hough Transform
  - Squares and Circles have similar means but triangles stand out
- Slope of Hough Transform
  - Easy to distinguish squares from circles
- Median
  - Always goes Triangle->Circle->Square



Mouse vs Eyes

- Results are comparable
- Mouse drawn results are higher than eyes but similar
  - Familiarity of mouse, unfamiliarity of using eyes
- Differences
  - Mouse results more smooth than eyes
    - Affects Corner Finding techniques
    - Affects Bounding Area Techniques

Best Fit Radius

PERCENTAGE MATCH USING BOUNDING SHAPES WITH GAZE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**7.04515**	5.275	5.04847
	Square	5.15393	**5.19377**	3.58795
	Triangle	4.67693	3.45145	**5.82683**

PERCENTAGE MATCH USING BOUNDING SHAPES WITH MOUSE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**7.87036**	5.88343	2.19373
	Square	0.573813	**7.10807**	1.72813
	Triangle	0.35031	2.99144	**6.71473**

Extra area from roughness leads to more detection of circles.

Mouse drawn shapes are more smooth and fit within tighter bounds.

Circle and squares have many shared bounding points.

Corner Detection

CORNER DETECTION BY SHAPE WITH GAZE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**1.95**	0.7	0.75
	Square	1.7	**1.9**	0.75
	Triangle	1.65	0.6	**1.9**

CORNER DETECTION BY SHAPE WITH MOUSE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**1.0**	0.1	0
	Square	.7	**1.7**	0.3
	Triangle	0	0.5	**1.1**

Roughness from eyes detects more corners.

Non-compliant corner locations

Area of Hough Transform

AREA OF HOUGH TRANSFORM – GAZE CASE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**2.547442**	1.903898	1.419948
	Square	1.536639	**2.485485**	1.695392
	Triangle	3.183223	0.440902	**2.513845**

AREA OF HOUGH TRANSFORM – MOUSE CASE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**2.958929**	0.947028	2.456028
	Square	1.999308	**2.422011**	1.421321
	Triangle	0.84204	2.646811	**2.777951**

Area for a circle shape and a square shape are nearly identical.

Extra roughness adds area that affects results for example the triangle case.

Mean of Hough Transform

MEAN OF FIRST ONE-THIRD OF HOUGH TRANSFORM – GAZE CASE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**2.626377**	3.770403	0.647659
	Square	1.571241	**3.828741**	1.244637
	Triangle	0.835423	2.005884	**3.846952**

MEAN OF FIRST ONE-THIRD OF HOUGH TRANSFORM – MOUSE CASE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**4.185824**	3.09798	0.237311
	Square	2.279997	**4.253781**	0.619156
	Triangle	0.684837	2.134503	**4.240643**

Average value for square and circle are very close

Triangle stands out.

Slope of Hough Transform

SLOPE OF HOUGH TRANSFORM – GAZE CASE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**1.437732**	2.24371	0.7518
	Square	0.965202	**2.138216**	1.319481
	Triangle	0.387915	1.862779	**2.239229**

SLOPE OF HOUGH TRANSFORM – MOUSE CASE

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**2.778436**	2.617368	0.045264
	Square	2.368715	**2.578591**	0.158205
	Triangle	0.699052	1.2077	**2.497919**

Averaging window with 5 data points the highest matrix intensity value.

Variance is an issue for max data point.

Median of Hough Transform

MEDIAN OF HOUGH TRANSFORM – GAZE CASE

Circle Results	478
Square Results	816.9
Triangle Results	261.5

MEDIAN OF HOUGH TRANSFORM – MOUSE CASE

Circle Results	251.1
Square Results	836.9
Triangle Results	58.3

Always goes Triangle->Circle->Square

Based on radius size and shape

Final Results

SHAPE DETECTION FINAL RESULTS USING EYE GAZE.

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**18.6067**	14.89301	9.617871
	Square	11.92702	**18.54621**	9.597459
	Triangle	12.43349	9.56102	**18.42686**

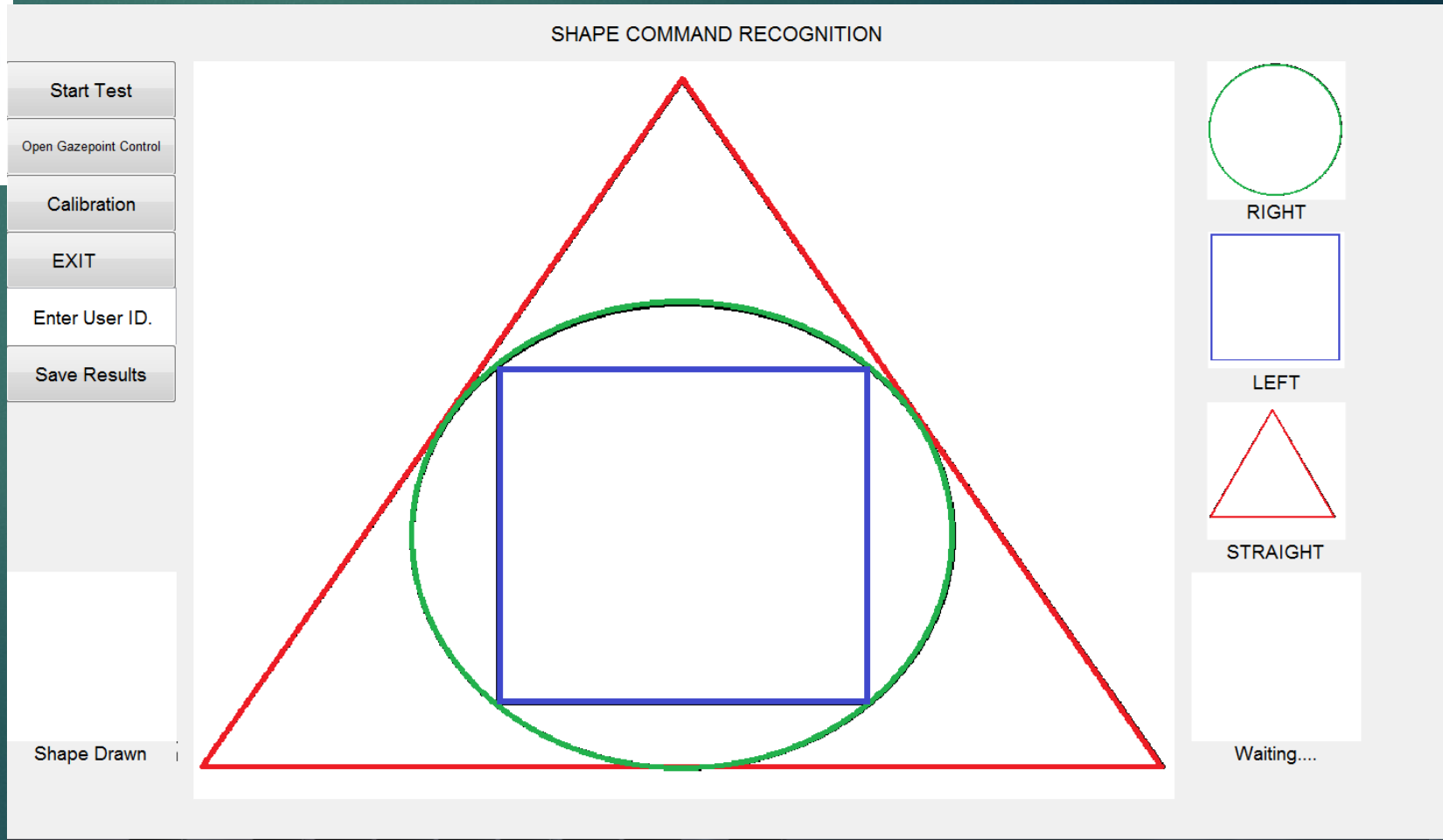
SHAPE DETECTION FINAL RESULTS USING MOUSE INPUT

		Shape Detected		
		Circle	Square	Triangle
Shape Drawn	Circle	**21.79355**	13.6458	5.932328
	Square	14.08615	**21.06246**	5.226813
	Triangle	6.929033	10.48045	**20.13124**

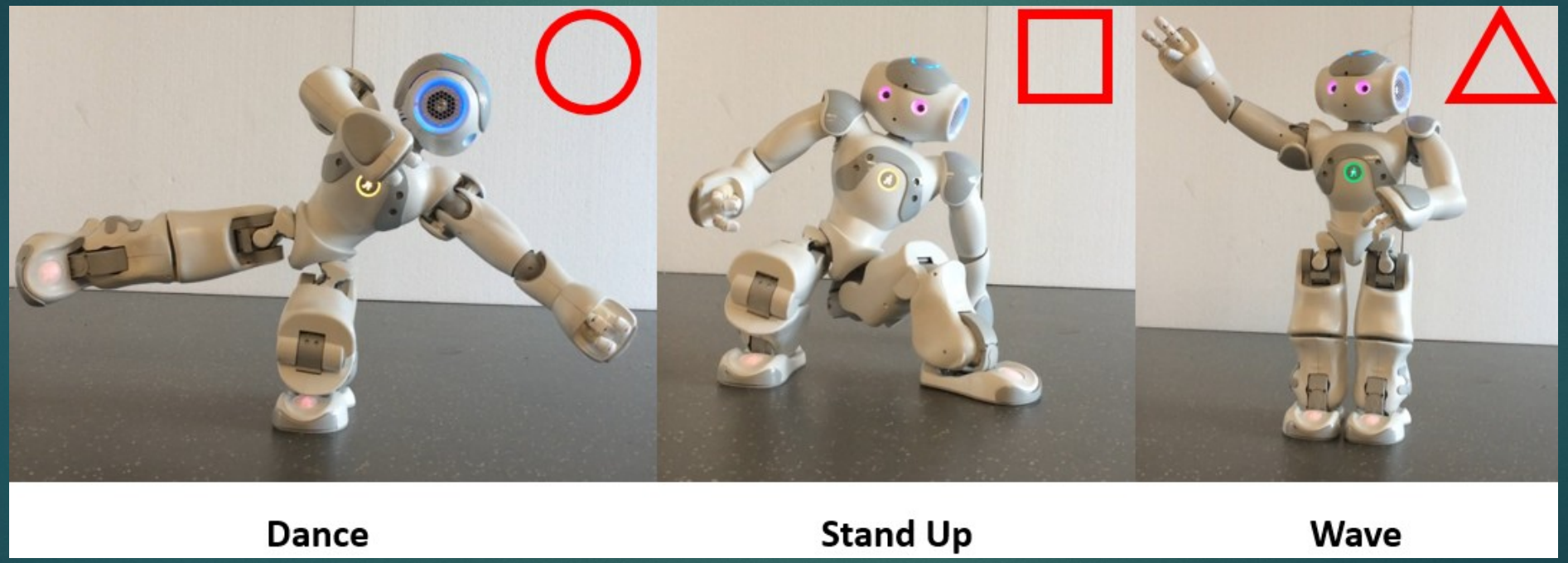
The right shape was detected.

A cut off value of 15 could be set for no false detections.

Interface Example

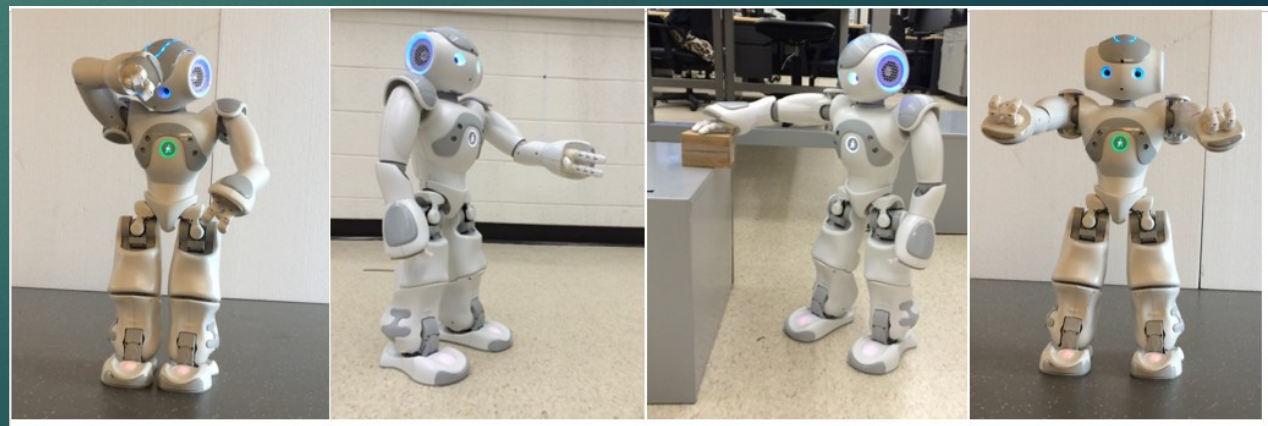


Testing with Robot



Review and Conclusion

- The detection of mouse-drawn shapes and eye gaze-drawn shapes were comparable.
- Shape based recognition is a viable option for communication.
- Blinks can be difficult to count and easily confused, whereas shapes are more natural and can vary in size for intensity of the command
- People who may lack the motor skills necessary to control technology can still use this method



Questions and Contact

- Contact should be made to: trevor.craig@huskers.unl