Utilizing Object Detection of Humanoid Robots for Implementation in Color Recognition and Motion Planning

TCNJ THE COLLEGE OF NEW JERSEY SCHOOL OF ENGINEERING

Nicole Lim; Advisor: Dr. Seung-yun Kim

Department of Electrical and Computer Engineering, The College of New Jersey, Ewing, NJ

New Jersey Space Grant Consortium

Abstract

Algorithms and operating systems are continually being evolved in need to develop more equipped and reliable robotic systems capable of performing in many real-world dynamics. The NAO robot is used to observe the ability of robots to perform specific tasks and the accuracy and reliability in the robot's performance. The robot is tasked with color detection, accurately identifying object distances and motion planning. Algorithms modeled in Python with use of OpenCV allowed for successful object registration.

Petri nets

Definition: A graphical tool used for modeling and simulating various complex systems. Petri nets are a 5-tuple: PN = (P, T, F, W, Mo) where P = Places, T = Transitions, F = Arcs, W = Arc Weight, and Mo = Initial Marking Tokens. These variables represent the states, operations, and connections within systems, where a tokens movement throughout the net represents the current state of the system.

Firing Rules: A token may only fire once a transition is enabled, meaning the correct amount of tokens must be present in the input place and the arc weight conditions must be satisfied. The firing of a token moves the token from input place through the transition, then finally to the output place.

Object Detection and Motion Planning

Using OpenCV and Choregraphe applications, the NAO Robot detects the color of each puzzle piece for use in the grasping and path motion behavior process.

An algorithm allows for the creation of trackbars for detection of hue, saturation, and value, HSV, of each corresponding color for further implementation with the NAO. In other words, trackbars allow for the masking of unwanted colors, thus allowing for the accurately measured HSV values seen in Table 1.

Table 1. Minimum and Maximum HSV Values

	Hue Minimum	Hue Maximum	Saturation Minimum	Saturation Maximum	Value Minimum	Value Maximum
Navy Blue Diamond	105	165	185	255	100	255
Red Oval	0	7	195	253	138	255
Purple Rectangle	113	150	101	255	134	255
Forest Green Circle	48	98	156	210	80	164
Orange Square	8	28	154	209	195	255
Periwinkle Hexagon	67	110	89	255	190	255
Pink Pentagon	130	173	64	255	207	255
Lime Green Triangle	35	60	115	235	170	255
Yellow Star	23	45	90	245	164	255

Masks are used simultaneously with the bitwise_and operator to create the resulting image. This operation adds two images together to detect the color desired by filtering out all the unwanted pixels, as shown below in Figure 1.

Figure 1. OpenCV Masked and Resulting Image

For motion planning, the Choregraphe timeline application stores the arm movements. NAO will grasp the puzzle pieces positioned at the marked location on the table by the user. The main goal is to store many small keyframes to return the puzzle piece to the correct location in one smooth movement, seen in Figure 2.

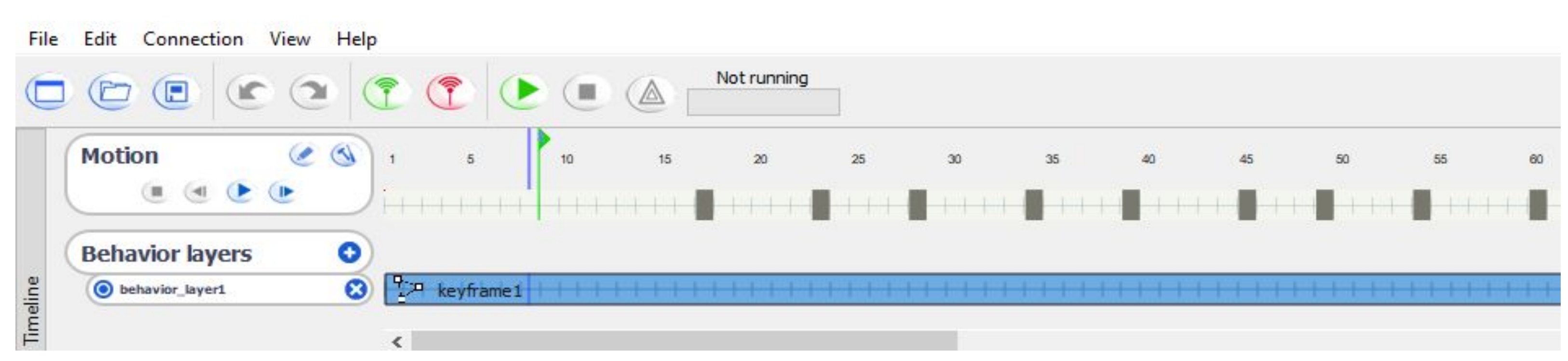


Figure 2. Choregraphe Timeline

Results

The Color Detection algorithm created detects all nine colors represented in the puzzle board with the assistance of trackbars. The grasping and motion planning procedure using the NAO's left hand was achieved for the circle, square, pentagon, and triangle for a total of 220 completed trials. Storing various keyframes allowed NAO to correctly identify the predetermined location of the puzzle piece to be grasped to then fulfill the final motions of returning the puzzle piece to its respective location.

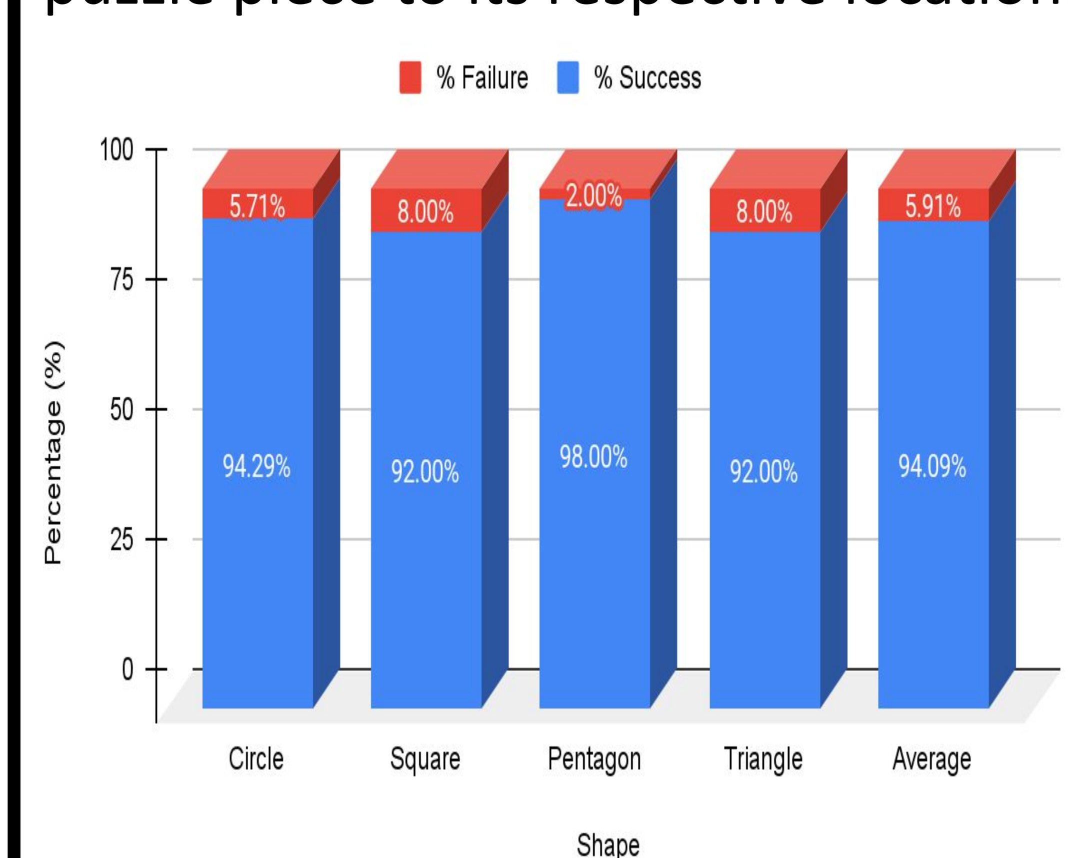


Figure 3: Success and Failure Rate

Figure 4. NAO Motion Planning

Conclusion & Future Work

To discover an appropriate way of sending a ping from Python to Choregraphe for a connection between the color recognition and motion planning aspects. Also, to further improve the NAO's ability to locate objects.

Acknowledgement

This work was supported by the New Jersey Space Grant Consortium and by MUSE 2021, conducted at The College of New Jersey.

Utilizing Object Detection of Humanoid Robots for Implementation in Motion Planning and Stereo Camera Calibrations

SCHOOL OF ENGINEERING

Nicole Lim and Elizabeth Lopez; Advisor: Dr. Seung-yun Kim

Department of Electrical and Computer Engineering, The College of New Jersey, Ewing, NJ

New Jersey Space Grant Consortium

Abstract

Algorithms and operating systems are continually being evolved in need to develop more equipped and reliable robotic systems capable of performing in many real-world dynamics. The NAO robot is used to observe the ability of robots to perform specific tasks and the accuracy and reliability in the robot's performance. The robot is tasked with color detection, accurately identifying object distances and motion planning. Algorithms modeled in Python with use of OpenCV allowed for successful object registration.

Petri nets

Definition: A graphical tool used for modeling and simulating various complex systems. Petri nets are a 5-tuple: PN = (P, T, F, W, Mo) where P = Places, T = Transitions, F = Arcs, W = Arc Weight, and Mo = Initial Marking Tokens. These variables represent the states, operations, and connections within systems, where a tokens movement throughout the net represents the current state of the system.

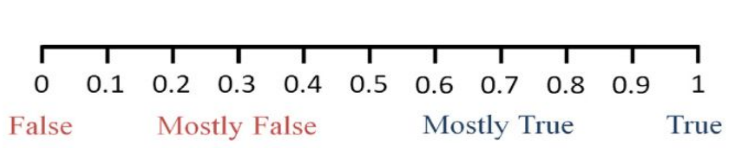
Firing Rules: A token may only fire once a transition is enabled, meaning the correct amount of tokens must be present in the input place and the arc weight conditions must be satisfied. The firing of a token moves the token from input place through the transition, then finally to the output place.

Fuzzy Logic

Definition: Uses fuzzy set theory to describe, analyze, and process qualitative or imprecise data.

Fuzzy Set: A set that defines the truthfulness of a particular set. With the truth value ranging from 0 to 1, where 0 and 1 are completely false and true, respectively. Decimal values between 0 and 1 allow for partial membership indicating degree of truth.

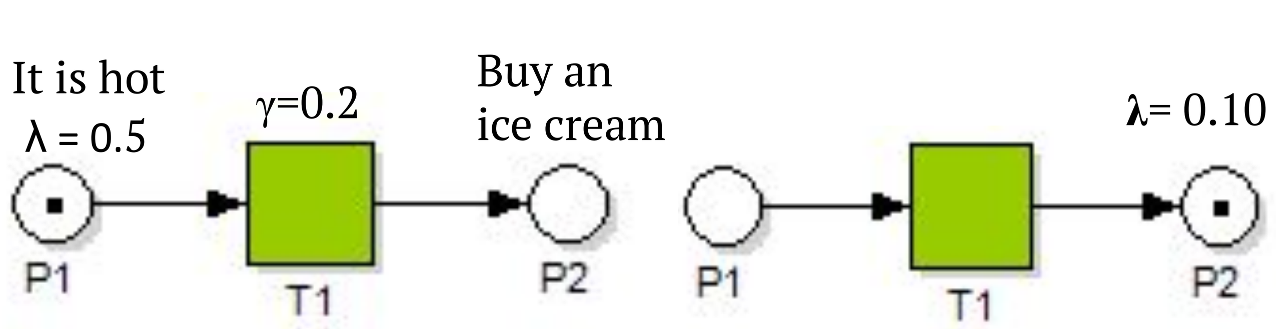
$$\mu_x : X \rightarrow [0,1]$$



Fuzzy Petri nets

Definition: A subset of Petri nets that incorporates fuzzy logic. Decisions are made based on the truthfulness value and fuzzy logic allows for conclusions on qualitative situations lacking precise reasoning.

Firing Rules: Tokens are assigned a certainty factor (λ) and transitions a threshold value (γ). The firing of a token is enabled when λ > γ. The resulting truth value is σ = λ × γ.



Object Detection and Motion Planning

Using OpenCV and Choregraphe applications, the NAO Robot detects the color of each puzzle piece for use in the grasping and path motion behavior process.

An algorithm allows for the creation of trackbars for detection of hue, saturation, and value, HSV, of each corresponding color for further implementation with the NAO. In other words, trackbars allow for the masking of unwanted colors, thus allowing for the accurately measured HSV values seen in Table 1.

Table 1. Minimum and Maximum HSV Values

	Hue Minimum	Hue Maximum	Saturation Minimum	Saturation Maximum	Value Minimum	Value Maximum
Navy Blue Diamond	105	165	185	255	100	255
Red Oval	0	7	195	253	138	255
Purple Rectangle	113	150	101	255	134	255
Forest Green Circle	48	98	156	210	80	164
Orange Square	8	28	154	209	195	255
Periwinkle Hexagon	67	110	89	255	190	255
Pink Pentagon	130	173	64	255	207	255
Lime Green Triangle	35	60	115	235	170	255
Yellow Star	23	45	90	245	164	255

Masks are used simultaneously with the bitwise_and operator to create the resulting image. This operation adds two images together to detect the color desired by filtering out all the unwanted pixels, as shown below in Figure 1.

Figure 1. OpenCV Masked and Resulting Image

For motion planning, the Choregraphe timeline application stores the arm movements. NAO will grasp the puzzle pieces positioned at the marked location on the table by the user. The main goal is to store many small keyframes to return the puzzle piece to the correct location in one smooth movement, seen in Figure 2.

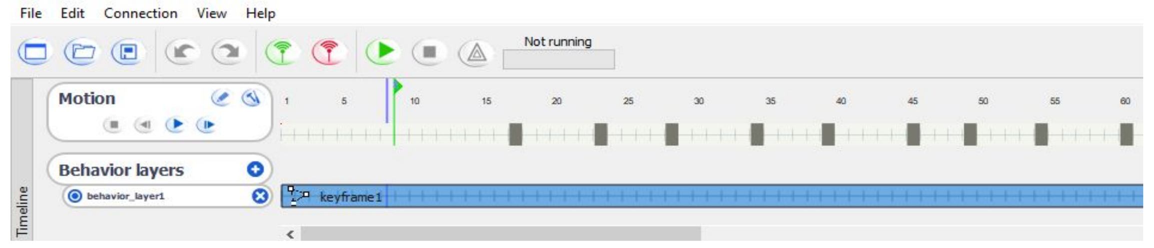


Figure 2. Choregraphe Timeline

Object Detection and Stereo Camera Calibrations

Two web cameras with a baseline (b) as shown in Figure 1 are used to create a stereo camera which takes multiple photos from varying angles of a checkerboard print-out. After running these images through a stereo camera calibration app, a camera calibration matrix (K) as shown in (1) is produced, providing the camera's focal length (f) in pixels.

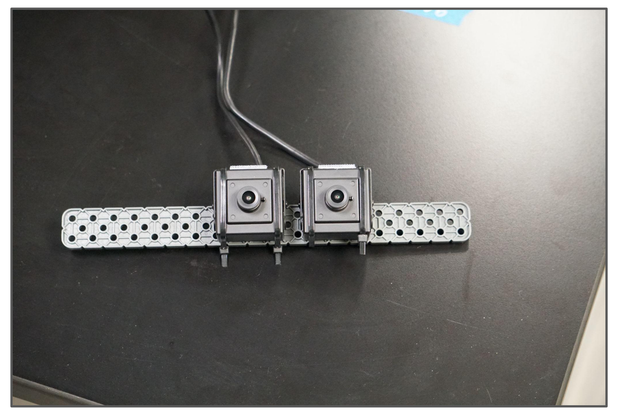


Figure 1. 55 mm baseline distanced cameras

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The cameras then take photos of a red ball of diameter 100 mm at calculated distances (d). The images are passed through the created Reduction Filter method, which involves converting the RGB image to HSV to eliminate colors out of the red range from the image.

With only the red ball remaining in the image, a Hough Circle Transform circle is created, outlining the ball and locating it's center point. Finding the difference between the ball's center point location on corresponding frames from the two cameras as shown in Figure 2 obtains the measurement of disparity in pixels (uL - uR).

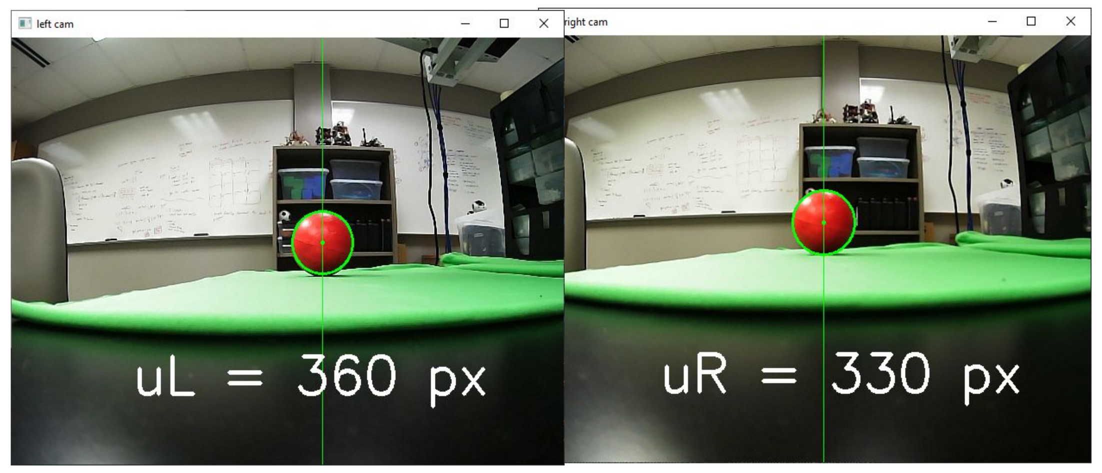


Figure 2. center points of red ball in both frames

Finally using the Stereo Camera Distance Ranging Method, the distance from the cameras to the red ball is calculated in (2).

$$d = \text{distance from camera to object} = \frac{\text{baseline (mm)} * \text{focal length (pixels)}}{\text{disparity (pixels)}} = \frac{b * f}{uL - uR} \quad (2)$$

Results

The Color Detection algorithm created detects all nine colors represented in the puzzle board with the assistance of trackbars. The grasping and motion planning procedure using the NAO's left hand was achieved for the circle, square, pentagon, and triangle for a total of 220 completed trials. Storing various keyframes allowed NAO to correctly identify the predetermined location of the puzzle piece to be grasped to then fulfill the final motions of returning the puzzle piece to its respective location.

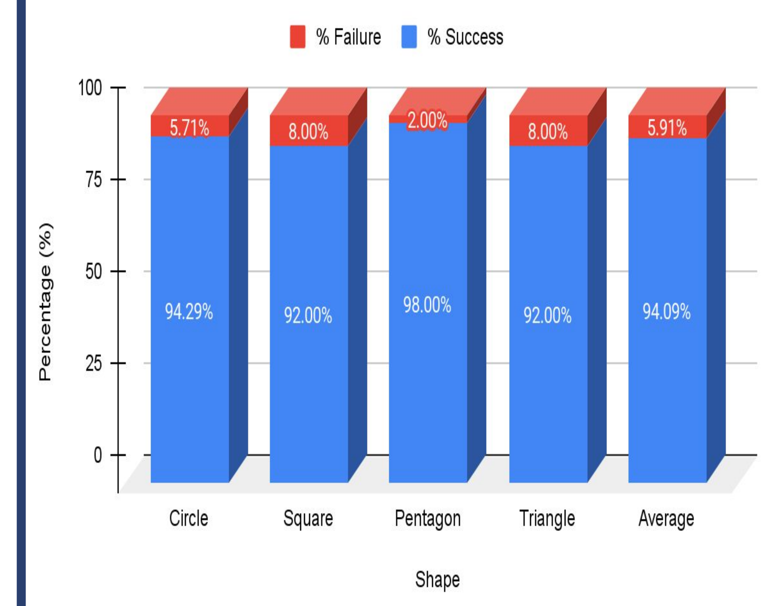


Figure 3: Success and Failure Rate

Figure 4. NAO Motion Planning

The Optimized Stereo Vision Distance Detection algorithm was finalized after multiple trials resulted in precision and accuracy. Creating the Reduction Filter method in collaboration with accurate camera calibration matrix data allowed for reliable results as shown in Figure 3. This algorithm can be implemented for the future of humanoid robots, by utilizing two cameras in the same location of their eyes to become more human-like.

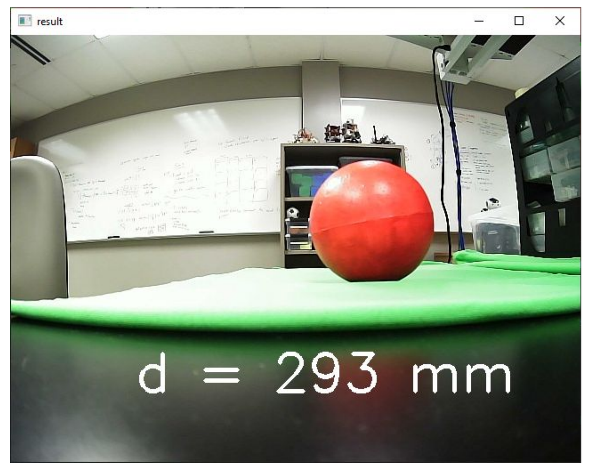


Figure 3. 300 mm distance from cameras

Conclusion & Future Work

To discover an appropriate way of sending a ping from Python to Choregraphe for a connection between the color recognition and motion planning aspects. Also, to further improve the NAO's ability to locate objects. To improve the accuracy of longer ranged distances with the created stereo camera distance ranging algorithm and apply these methods to the NAO robot.

Acknowledgement

This work was supported by the New Jersey Space Grant Consortium and by MUSE 2021, conducted at The College of New Jersey.