UPennalizers

Team Report Fall 2021

Penn Engineering | GRASP Laboratory

General Robotics, Automation, Sensing & Perception Lab

Acknowledgements
We would like to thank the GRASP Lab and our faculty advisors CJ Taylor for their support and guidance throughout the last year. We would also like to thank our generous sponsors Blue River Tech, Cruise Automation, and CCTV Camera World.

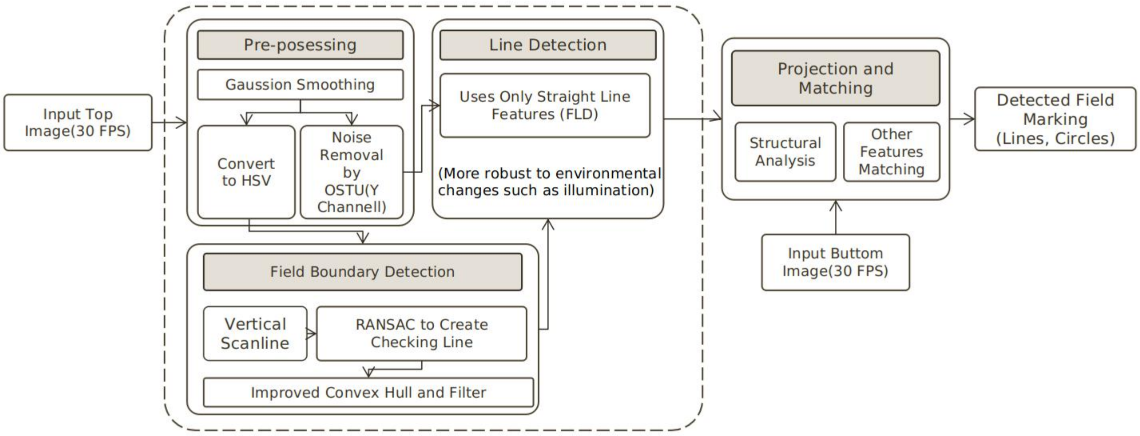
Below are the abstracts from the 2021 R&D Symposium where members presented on their semester long projects in the fields of computer vision, locomotion, and behavior.

Illumination and Shadow Invariant Feature Detection

Yuwei Wu

MS in Systems Engineering ‘21

Robust feature detection under various lighting conditions is crucial for the Nao robot's localization under the dynamic environment. Previous work of Upennalizers develops the adaptive field detection, which restricts the valid region of the robot field of view as the green reference for classification. We now present a light invariant feature detection pipeline for RoboCup SPL, which improve boundary detection and develop line detection directly by straight-line features, rather than previous scanline segment checking and grouping.



The results are as follows, which demonstrated the distinguish performance, especially when the illumination varies during the competition.

Strong artificial lighting

Different natural lighting

VISION: GOALPOST DETECTION USING OPENCV IN PYTHON

M. Abdullah Khalid
Systems Science & Engineering ‘23

Detecting the opponent goalpost is arguably the most important yet one of the trickiest challenges that our robot must accomplish. In past years, it has been observed that different factors such as natural and artificial light, shade, background distractions as well as field lines are often confused as goalposts which leads to a significant time-waste. In this project, a Probabilistic Hough Transform Algorithm is investigated as a potential improvement in our team codebase.

The Hough Transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. It is implemented through the OpenCV Library in Python. Although a classical approach, it boasts a faster runtime over modern implementations like Neural Networks which gives it a clear advantage in time-constrained settings such as a RoboCup Soccer game itself.

The algorithm was evaluated using a dataset of 540 images, 50 which actually contained a goalpost, and the rest did not. The precision and recall were calculated 58.7% and 74.0% respectively. Although not overly impressive, it is worth noting that the runtime looms within an astonishing 36 to 64 milliseconds per image! Looking forward, we plan to implement, analyze and compare the performance of different algorithms for object detection and also gather more diverse data to achieve the same.

Results:

# Images	Predicted Positive	Predicted Negative	**Total:**
Actual Positive	37	13	50
Actual Negative	26	464	490
Total:	63	477	**540**

Improving Robotic Goal Post Detection

Vision

Tasos Panagopoulos
Computer Science '24

Caleb Gupta
Computer Science '24

In previous years, it had been difficult for the UPennalizers Robots to be able to detect the outlines of a goalpost. That task is critical to localization, as knowing where the goalpost is located helps robots aim accurately and can thus lead to more goals for our team. We worked on the first task in improving detection, being able to answer the polar question of whether or not a given frame has the goalposts identified.

We chose to implement a neural network for goalpost detection, as classical vision algorithms are less accurate and harder to develop. Thus, we needed a large number of images to train our model. We chose to use the dataset from the BitBots team, a group from the humanoid league who had an incredibly useful dataset as their images were already labeled. Due to their help, we could focus on creating an efficient convolutional neural network. After gathering around 4000 images from their dataset and splitting them into a training set and a test set, we developed a neural network using Lua and the Torch library. Our final network consisted of 2 convolutional layers, 2 max pooling layers, and used the NLL (Negative Log-Likelihood) loss function. We were ultimately able to achieve 96% accuracy in detecting whether or not an image has such a post, as is displayed by the two images to the left (the first image being false and the second image true).

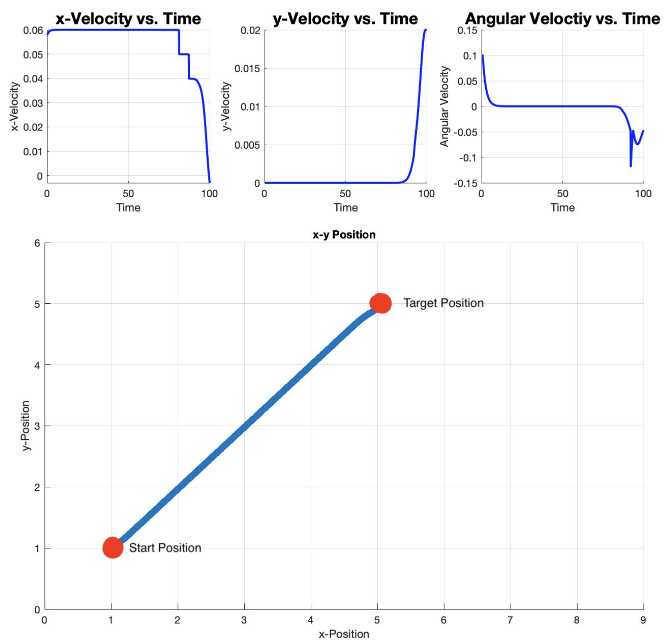
Future directions include being able to pinpoint the pixels within a frame that outline the goalpost to allow the robot to localize based on its distance from the goal posts, as well as improving the speed of our neural net. We also hope to reduce the number of false positives by changing the parameters or structure of our convolutional neural network. The purpose for targeting false positives above false negatives is that when a goalpost is in sight the robot will be told to shoot, a procedure that could ultimately cost? our team a lot of time. However, a false negative is much less detrimental as many images will be processed, and milliseconds will be lost opposed to the seconds and potentially minutes that a false positive would lead to.

Analyzing Robot Velocity During Position Change

Jessica Sui
Mechanical Engineering and Applied Mechanics '23

Georgia Georgostathi
Bioengineering '24

At any point in the game, the robot may be given a target position, or a position on the field it wants to get to and it will take a series of actions to go from its current position to the aforementioned target position. We are especially curious about the points on this journey where the robot may be susceptible to falling over, since this would not only take time away from gameplay but also has the potential of harming the robot. We theorized that at points where the robot would be most susceptible to falling over, there would be sudden changes or jumps in its velocity.

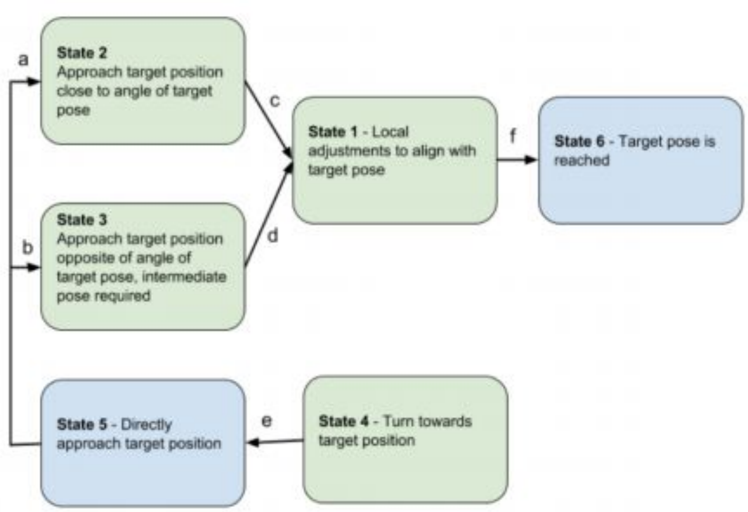


Thus we set out to graph and analyze the velocity of the robot when it is moving from one position to the next. The three graphs on top represent the x, y, and angular velocity of the robot for 100 seconds after the robot has been given a target position. The bottom graph can be viewed as a visual representation of the 9x6 field and each point is the robot's position after another 0.1 seconds. We can gather from the velocity graphs where the robot's movements are smooth, confirming our current techniques, and also at what points the robot might be in trouble of falling over. From here, we can continue this project by coming up with a tolerance for the maximum change in velocity (acceleration) the robot can undergo without being in danger of falling over. This may allow us to increase the speed of the robot, improving its performance during gameplay.

Velocity Generation Strategy

Vraj Patel
Electrical Engineering '22

During the soccer match, the localization of the ball requires measurements to be taken by the Aldebaran Nao robot's sensors. These sensors are inherently noisy and cause instability and path divergence in the robots. These unstable motions are further amplified by sets of incompatible translational and rotational velocities in the current strategy. Two major areas of improvement for the current strategy involve developing a method that stabilizes the robots while quickly converging on the ball location. Therefore, the new velocity generation strategy is based on a finite state machine that has six states that direct the robot to particular motions when approaching a target pose.



The initial state is chosen based on the relative distance and angle when a new target pose is selected. The process of a finite state machine allows velocities that are chosen to be in order and converge to the desired target pose. The sequential process increases stability by reasonably making sure the robot is facing the ball when localizing. Along with stabilizing the system, a FSM is fairly simple in its implementation which reduces complexity and ultimately leads to faster localization of the ball.

Improving the Target Position of the Supporter

Katelyn DePaula & Zhiyan Lu

Computer Engineering '24 & Computer Science '22

By definition, a supporting player assists the player with the ball with defense and attacks and is not a goalie or defender. Our project mainly seeks to enhance the supporter's attacking capacity.

Given the flexibility and potential of this role, we strengthened its presence in the opponent's field by designating aggressive coordinate positions for it to undertake during the staging of an attack. Our new target positions reduce unnecessary vertical movements and thus unnecessary motor usage, endorse a forward playing style that improves passing, shooting, and rebounding opportunities, and enable timely reaction in the case when the ball is lost.

Our specific implementation maneuvers the x-position of the supporter (along the length of the field) for it to move further upfield and ahead of the ball, before the opponent's goal. If players on our team have been taken out due to penalties or injury, the supporter stands closer to the ball as the attacking behaviors become riskier due to lack of defense. Figure 1 below illustrates the more defensive position when less team members are on the field and Figure 3 illustrates the more upfield position when all five team members are present.

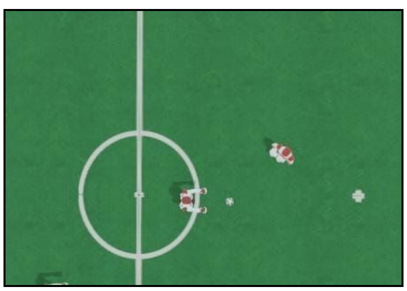


Figure 1. Three Players

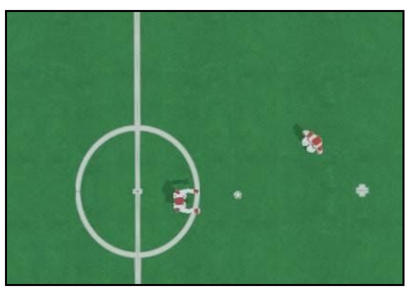


Figure 2. Four Players

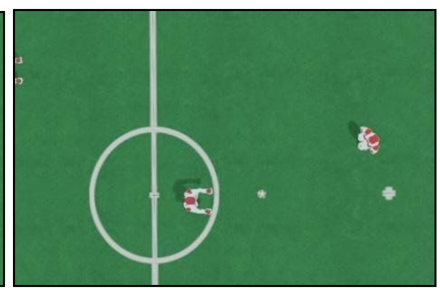
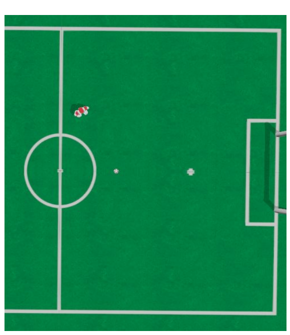
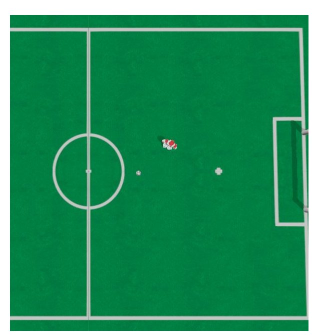


Figure 3. Five Players

We have found that the new supporter algorithm relocates to a desired position that is a shorter traveling distance vertically than in the previous method. The supporter also stands 50.7% closer to the goal to prepare for attacks. In the future, we plan to further improve the supporter position by both avoiding opposing players in our target position when receiving passes and defensively blocking opposing players from reaching the ball.



Old Position



New Position