C0-CueTips

ECE Capstone Proposal

Andrew Gao (agao2), Debrina Angelica (dangelic), Tjun Jet Ong (tjunjeto),

Use Case

Problem

Learning to play eight-ball pool from scratch requires a steep learning curve. Without proper guidance from professionals or friends, it can often lead to frustration or discouragement.

Solution

CueTips is an Computer Vision based pool table that enhances the pool learning experience by projecting predicted shot trajectory in real time.

Areas

- Software systems (Computer Vision)
- Hardware Systems (Embedded Devices, Hardware IMU)
- Signals and Systems (Wireless communication between multiple devices)

Use Case Requirements

Ball Detection Accuracy	0.2 in	Minimize the ball detection error to reduce the angle of misalignment in its trajectory.
Trajectory Prediction Accuracy	2.0 degrees	The angular width of a pocket is around 4-5 degrees [1]
Overall latency	100 ms	Users have the illusion that they receive instantaneous feedback [2]

[1] https://forums.azbilliards.com/threads/fractional-aiming-and-required-accuracy.522183/
[2] Miller, R. B. (1968). Response time in man-computer conversational transactions. Proc. AFIPS Fall Joint Computer Conference Vol. 33, 267-277.

Technical Challenges (1)

Rigidity of Camera	A large, rigid frame must be built to ensure camera stability for accurate detections. It must also be able to hold our projector for accurate projections.
Lighting Conditions	Different lighting conditions may cause discrepancies in edge detection. Thus, we must ensure our model caters to the different conditions.
Cue Stick Vertical Position	Height of the cue stick position may vary and not be accurately detected by our camera, which would cause inaccuracies in the ball's trajectory.

Technical Challenges (2)

Prediction Latency	Our goal for real-time object detection and feedback will be computationally intensive, leading to high latency from sensor detection to trajectory projection.
Uncertainty in Physics Calculations	The accuracy of physics calculations are limited to the accuracy of our object detection algorithms. Furthermore, other conditions in real life could affect whether the trajectory can be followed.
Image Distortion	The images we obtain from the camera may not exactly represent the real conditions. It is likely we will run into issues with offset, slight image distortion, scratches on the lens, etc.

Solution Approach: Trajectory Prediction

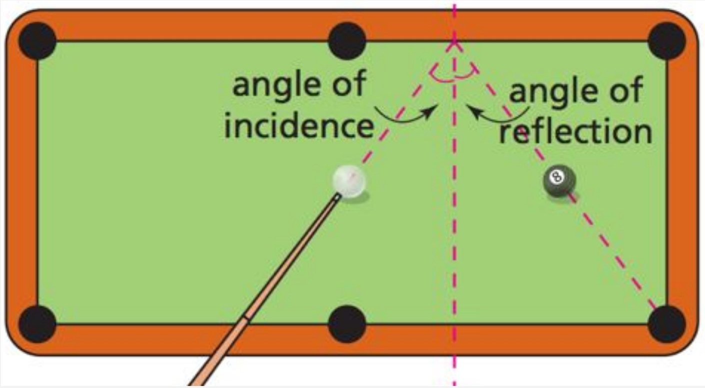
Backend Computation	Most backend computation will be physics simulation. Some libraries we were thinking of using: Pymunk, Scipy, etc.
8-Ball Physics Research	Implement 8ball simulations based on research papers and pool simulation libraries (i.e. pooltool) 

Image source

Solution Approach: Object Detection

Computer Vision Model	Use edge detection to segment cue stick, ball, and table. Responsible for correctly identifies entities on the pool table in order for us to accurately do computation for predicted trajectories.
Stick Orientation	The stick orientation and ensuring the user holds it correctly will be tackled on two fronts: 1. Computer Vision model ensures cue stick is positioned correctly left-to-right (2D) 2. IMU accounts for up-and-down adjustments (angle at which ball is hit)
AprilTags	Use AprilTags to determine the camera's position and orientation in a given environment. Minimizes image distortion and provides better accuracy for stick orientation.

Solution Approach: Instantaneous Feedback

Projector	Projector will be used for displaying the system's trajectory predictions in real-time. We aim to have system provide < 100 ms latency feedback.
Camera	HD (1080p) Camera. Using 4K/8K resolution will slow down the entire system, we need it to be real-time. There are a number of 1080p resolution cameras for such projects (e.g. Arducam, RaspberryPi).
Web Application	Our web application will be one of primary ways the user interacts with the project. Displays: 1. Whether user is holding cue stick correctly, 2. Acceleration/force with which they strike with the cue stick.

Testing, Verification, and Metrics

Requirement	Testing Strategy	Error Metrics
Trajectory Accuracy	• Conduct user testing on our system with an experienced player. • Record a video and trace the ball's trajectory. • Measure the angle between predicted trajectory and actual trajectory	Angle: < **2 deg**
Ball Prediction Accuracy	• Scatter pool balls on table and project our model's perception on their position. • Measure the distance between the centers of our projection and the ball's center.	Distance: < **0.2 in**

Testing, Verification, and Metrics

Requirement	Testing Strategy	Error Metrics
Overall Latency	• Measure the time between the user's repositioning of the cue stick (state of table) and the update of the projection on the table • Time the python code and print out the timings after a particular change (calculates the latency of our calculations) • Record state changes on video to measure the action-response time.	Latency: < **100 ms**

Tasks and Division of Labor

Andrew	Computer Vision models for cue stick, ball, and edge detection.
Debrina	Web Application, IMU accelerometer testing with cue stick
Tjun Jet	Physics calculations for trajectory prediction, Hardware sensor integration (Cameras, Projector) with computer
All	Building of frame to mount camera and projector, Testing and integration

Schedule

Category	Person	Task	W4	W5	W6	W7	BRK	W8	W9	W10	W11	W12	W13	W14
Design	All	Acquire components	■											
	Andrew	Research camera libraries	■											
	Tjun Jet	Research CV libraries and application methods	■											
Computer Vision	Andrew	Detection of Cue Stick	■											
	Andrew	Detection of Pool Balls		■										
	Andrew	Detection of Walls of the Table	■											
	Andrew	First round testing of detection accuracy			■									
	Andrew	Second round testing of detection accuracy						■						
Trajectory Projection	Tjun Jet	Physics calculations on a ball's trajectory	■											
	Tjun Jet	Compute trajectory based on online images		■										
	Tjun Jet	Successfully compute a trajectory based on actual inputs			■									
	Tjun Jet	Output trajectory onto projector						■						
	Tjun Jet	First round testing of trajectory accuracy							■					
Camera and Projector Mount	All	Design frame to mount camera and projector	■											
	All	Build frame for camera mount		■										
	All	Mount camera and projector onto frame			■									
	All	Calibrate camera to testing environment (April Tags etc.)				■								
Hardware	Debrina	Access the camera from computer through wireless		■										
	Debrina	Mount IMU and stands onto the CueStick			■									
	Debrina	Ensure IMU Data is correct with cue stick movements				■								
	Debrina	IMU Data well integrated with web application						■						
Web Application	Debrina	Create local application				■								
	Debrina	Integrate camera feed to web application						■						
	Debrina	Integrate accelerometer feed to web application						■						
	Debrina	Integrate recommendation system to the web application							■					
	All	Error checking and refining								■				
Integration	All	Communication between devices								■				
	All	Integration with Web Application									■			
	All	Full integration										■		
Slack	All	Slack Time											■	■