C0-CueTips

Design Review

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

Use Case

Problem

There is a steep learning curve when it comes to learning how to play billiards. Without proper guidance from professionals or friends, it can often lead to frustration or discouragement.

Solution

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

ECE Areas

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

Use Case Requirements

Use Case Requirement	Technical Requirement
Detect pool balls within **0.2in** from actual ball location	CV model accurately detect ball's contours and calculates center no further than **0.2in** from ball's actual location.
Predicted trajectory must be within **2°** of actual trajectory	Object detection must be calibrated to frame generated by AprilTags, and calculations from physics models must be accurate.
Achieve latency of less than **100ms**	One cycle of sending video data, executing object detection, running physics calculations, and outputting trajectory prediction executes in at most **100ms**.

Solution Approach: Object Detection

Pool Ball Detection	Use Canny edge detection and cv2.findContours to detect cue balls and identify their centerpoints.
Wall Detection	Use Hough Line transform to detect lines in video feed.
Cue Stick Detection	Use findContours to detect cue stick. Use a 9-axis IMU attachment on pool cue to gather and data. Process using Ceva's sensor hub software (to detect vertical position of cue stick).
Localization & Calibration with AprilTags	Use AprilTags to determine the camera's position and orientation in a given environment. Minimize image distortion and provides better accuracy for stick orientation.

Solution Approach: Trajectory Prediction

8-Ball Physics Libraries	Implement 8ball simulations based on online pool simulation libraries. The ones we are focusing on are PoolTool and Pool Projection.

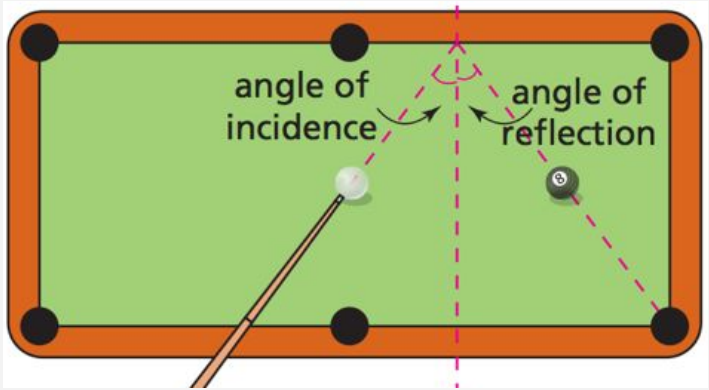
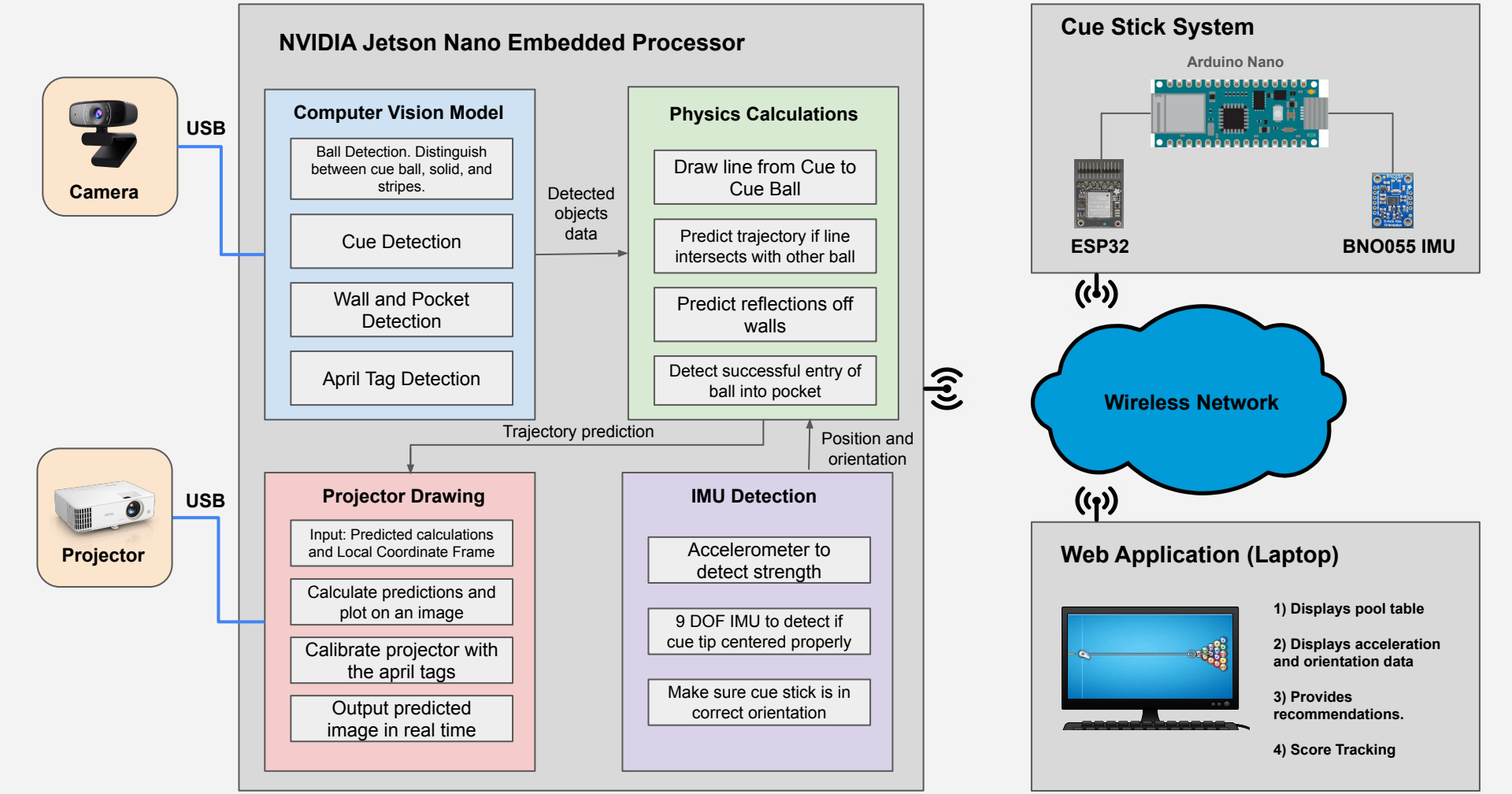


Image source

Solution Approach: Instantaneous Feedback

Projector	Projector mounted overhead will display our predicted trajectories on the pool table.
Camera	720p webcam at 60fps and a diagonal field of vision of 78°. Mounted overhead to provide data on game state that is used by our object detection and physics models.
Web Application	Our web application will be one of primary ways the user interacts with the project. Displays: 1. Full view of table and predictions. 2. Whether user is holding cue stick correctly. 3. Acceleration/force with which users strike the ball with the cue stick.


NVIDIA Jetson Nano Embedded Processor
Camera
USB
Computer Vision Model
Ball Detection. Distinguish between cue ball, solid, and stripes.
Cue Detection
Wall and Pocket Detection
April Tag Detection
Detected objects data
Physics Calculations
Draw line from Cue to Cue Ball
Predict trajectory if line intersects with other ball
Predict reflections off walls
Detect successful entry of ball into pocket
Trajectory prediction
Position and orientation
Projector
USB
Projector Drawing
Input: Predicted calculations and Local Coordinate Frame
Calculate predictions and plot on an image
Calibrate projector with the april tags
Output predicted image in real time
IMU Detection
Accelerometer to detect strength
9 DOF IMU to detect if cue tip centered properly
Make sure cue stick is in correct orientation
Cue Stick System
Arduino Nano
ESP32
BNO055 IMU
Wireless Network
Web Application (Laptop)
1) Displays pool table
2) Displays acceleration and orientation data
3) Provides recommendations.
4) Score Tracking

Implementation Plan

What we are referencing	Past pool table project implementations. Open-source libraries for building physics engine, AprilTag detection, object detection.
What programs we use	Python, OpenCV for edge detection. React/Flask for web server display. Arduino for cue stick system.
What we are buying	Camera, projector, and the mount for both. NVIDIA Jetson Nano , Arduino Nano, ESP32 WiFi module, BNO055 IMU.
What we are developing on our own	1) Environment detection model: gathering frame data from camera, motion from IMU. 2) Video/frame processing system - includes physics computation, edge/object detection, etc. 3) Sensor fusion algorithms

Testing

Latency	**Target:** < 100ms end-to-end. **Procedure:** We will time the code's execution from the point that it receives a particular frame to the point that it outputs a predicted trajectory based on that frame.
Trajectory Projection accuracy	**Target:** < 2^{o} between real and predicted trajectory. **Procedure:** Project the predicted trajectory on the table. An experienced pool player will take 10 shots, and we will video record the shots. Measure average angle deviation from the video.
Object detection accuracy	**Target:** < 0.2in between real and predicted ball detections. **Procedure:** Project detections of balls and measure distance between real and projected balls by taking a picture and measuring the difference through scaling the image.

Validation and Verification

Latency	**Verification:** Time code execution from start to end. **Risk Factors:** Timing of code may not accurately reflect actual latency as there might be additional latency due to wireless transmission.
Trajectory Projection accuracy	**Verification:** Video record the player's shot and measure the angle between the ball's actual trajectory and our predicted trajectory. **Risk Factors:** Variation in pool shots from player testing as they may not hit completely in the center. Thus, we will take 10 shots and average.
Object detection accuracy	**Verification:** Measure the difference in position between our projection and the actual position of the ball and cue. **Risk Factors:** Parallax error may lead to potential inaccuracy in error measurement. It may also be difficult to measure differences in ball location.

Work Distribution

Andrew	• AprilTag Detection and Integration. • Take model output and test it on projector.
Debrina	• Ensure accurate detections for cue stick, table walls, solid and striped balls. • Web Application
Tjun Jet	• Physics calculations for wall reflections and ball collisions. • Hardware sensor fusion between cue stick and camera.
All	• Building of frame to mount camera and projector, Testing and integration

Schedule

Category	Person	Status	Task	W4	W5	W6	W7	BRK	W8	W9	W10	W11	W12	W13	W14
Design	All	Doing	Acquire components												
	Debrina	Doing	Research camera libraries												
	Tjun Jet	Done	Research CV libraries and application methods												
Computer Vision	Debrina	Done	Detection of Cue Stick												
	Debrina	Done	Detection of Pool Balls												
	Debrina	Done	Detection of Walls of the Table												
	Debrina	Doing	Detection of Pockets												
	Debrina	Doing	Distinguish between cue ball, solids and stripes												
	Debrina	Doing	First round testing of detection accuracy												
	Debrina	Not Done	Second round testing of detection accuracy												
	Andrew	Done	April Tag Detection												
Trajectory Projection	Tjun Jet	Done	Physics calculations of cue ball's reflection on stick												
	Tjun Jet	Doing	Handle the case of cue ball not aligned with cue stick												
	Tjun Jet	Doing	Physics calculations on a ball's trajectory												
	Tjun Jet	Not Done	Compute trajectory based on online images												
	Tjun Jet	Not Done	Successfully compute a trajectory based on actual inputs												
	Andrew	Not Done	Output trajectory onto projector												
	Tjun Jet	Not Done	First round testing of trajectory accuracy												
Camera and Projector Mount	All	Not Done	Design frame to mount camera and projector												
	All	Not Done	Build frame for camera mount												
	All	Not Done	Mount camera and projector onto frame												
	All	Not Done	Calibrate camera to testing environment (April Tags etc.)												
Hardware	Andrew	Done	Access the camera from computer through wireless												
	Andrew	Not Done	Mount IMU and stands onto the CueStick												
	Andrew	Not Done	Ensure IMU Data is correct with cue stick movements												
	Andrew	Not Done	IMU Data well integrated with web application												
Web Application	Andrew	Not Done	Create local application												
	Andrew	Not Done	Integrate camera feed to web application												
	Andrew	Not Done	Integrate accelerometer feed to web application												
	Andrew	Not Done	Integrate recommendation system to the web application												
	All	Not Done	Error checking and refining												
Integration	All	Not Done	Communication between devices												
	All	Not Done	Integration with Web Application												
	All	Not Done	Full integration												
Slack	All	Not Done	Slack Time												