ECE 4950 - Project 2
The Big Lebowski: Autonomous Bowling Robot

Team 19

Corey Golec, John Phillips, Jacob Criswell, Avi Patel, Josue Morales-Sanchez

Date: 10/10/25

Executive Summary

The goal of this lab was to design and validate a camera-based sensor to identify the colors, shapes, counts, and positions of various stickers placed on an 8.5″×11″ sheet of paper. This camera was directly interfaced with MATLAB data processing software written by the team to implement a robust and error-tuned background-subtraction pipeline with color and shape classification. This served as the foundation for our overarching project for this semester, which is the camera system of the autonomous bowling robot. In addition to this video processing demonstration, the team developed measurable engineering requirements derived from a given customer's requirements. Three sensor concepts were drafted and judged with a weighted decision matrix to find the best variation. Ultimately, a fixed overhead camera design was chosen with a physical mockup, as well as a website scaffold being prepared to serve as the baseline from which further projects would branch.

Engineering Requirements for Camera Subsystem

We begin designing our camera system by translating the provided customer requirements into engineering requirements. We knew what the camera requirements were on the customer side, so the group tried to translate that into something that could easily be reproduced with the technology that we had. For each given customer requirement, we attempted to convert it into a metric that could be quantifiably measured with an isolated test. The following is a table going through the translation process:

Table 1: Camera Requirements

Customer Req.	Engineering Requirements
The system must detect all pins and or colored stickers placed in the camera view	The camera and image processing in MATLAB must correctly identify 100% of the pins and or shapes in the field of view under standard lighting conditions
The system must identify the position of each pin and sticker	The centroid of each object must be detected within +-5 pixels of its true location in the camera frame
The system must be able to differentiate between multiple objects of varying shapes and colors	The vision algorithm must correctly classify at least 95% of all test cases involving multiple different shapes at different angles and rotations
The system must accurately determine the color of each sticker	Color classification accuracy must be at least 95% for all defined colors: red, green, yellow, and blue
The sensing process must be stable and repeatable	Repeated detection of a stationary configuration must produce consistent position and color results with less than 5% variance between trials.
The system must provide real-time feedback to the GUI	The full image capture and processing cycle must complete in less than a second per frame
The sensing system must be low-cost and easy to integrate	The selected camera must be below $30 and interface directly with the Image Acquisition toolbox in MATLAB via USB connection

Hardware Documentation

To implement our game state sensor, we used a Logitech Brio 100 camera, a PVC mounting rig, and a light source. We used this hardware in conjunction with a laptop running MATLAB 2025b to perform the game state image processing. We mounted the camera to the PVC rig in a "vertical" configuration, allowing us to get a long image of the "lane" where the game will be played. We also implemented a light holder into the PVC rig, which allows us to create consistent lighting conditions for the camera. These consistent lighting conditions enable us to obtain a consistent image for our image processing. Figure 1 below shows the combined system.

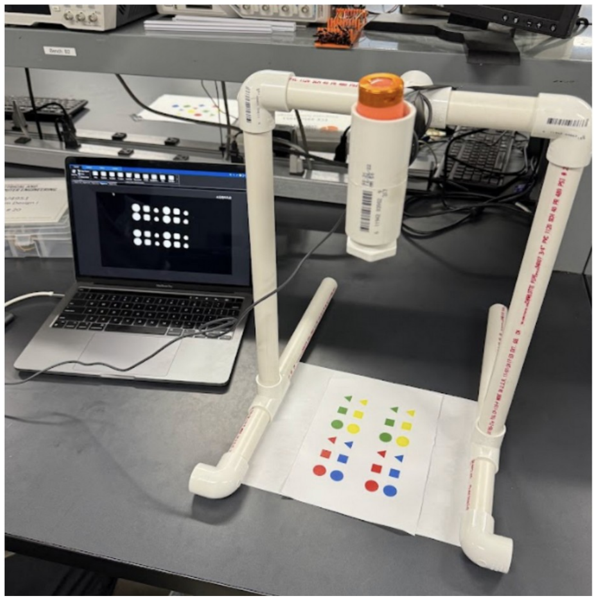

Figure 1: Camera System with PVC Rig

The resolution for the Logitech Brio 100 is 1920 pixels by 1080 pixels. Based on our image capture, the Brio 100 captures approximately 11 ⅛ inches by 7 ⅜ inches of the game board, indicating a resolution of about 25,273 pixels per square inch. At our rig height, usable FOV $\approx 11\frac{1}{8}'' \times 7\frac{3}{8}''$ — pixel density $\approx 1920 \times 1080 \div (11.125 \times 7.375) \approx 25{,}273$ px/in^2 — ≈ 44 px per $\frac{3}{4}''$ sticker diameter (sufficient for robust segmentation and centroiding). Overall, the camera can be used as an effective sensor for determining the state of the game, but it is also heavily dependent on the lighting and the distance at which the camera sits above the game plane. To combat these dependencies, we utilize a flashlight for more consistent lighting and mount the camera at a fixed location relative to the game plane. This solution ensures that the number of pixels measured for each shape remains constant, resulting in more accurate results.

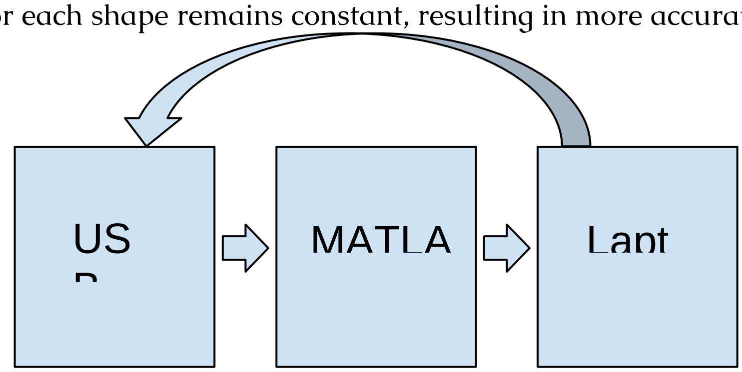

Figure 2: Connection Diagram

Software Documentation

The software developed for this project uses MATLAB's image processing toolbox along with a Logitech Brio 100 Webcam to detect and classify markers in the foreground based on shape and color on a captured background image or "game board" per customer requirements. The software developed has five functional stages:

1. Background and Foreground Image Acquisition
2. Background subtraction
3. Noise Filtering
4. Object Segmentation and Measurement
5. Shape and Color classification.

Below is a flow diagram organizing each segment of the program as it occurs.

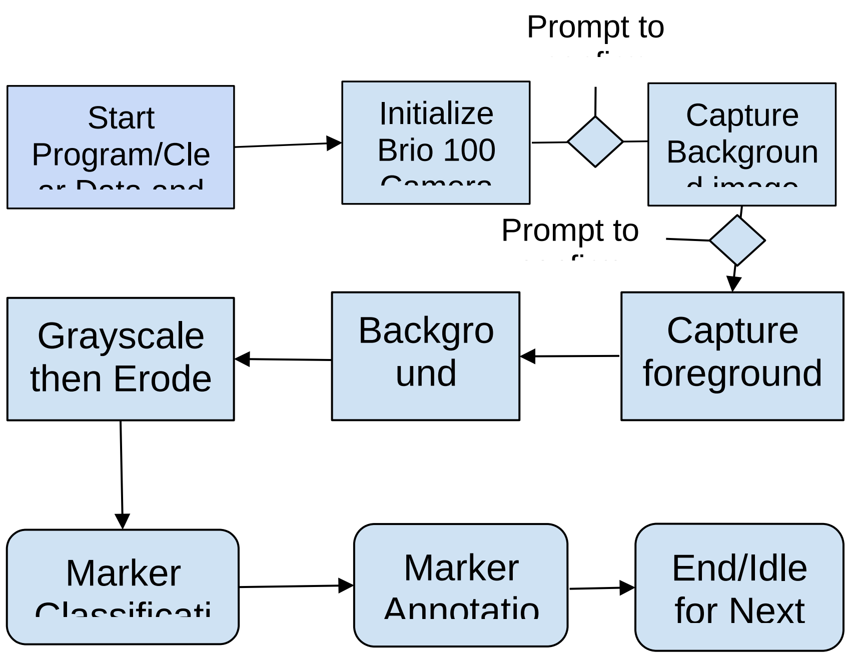

Figure 3: Flow Diagram of Software Implementation

As Figure 3 shows, the program begins by first initializing the connected Brio 100 camera. After initializing the camera, the program holds while the user gets the background image ready. Once the image is ready to be taken, the user can advance the program, which takes the photo. The program then pauses again while the user prepares the game board with the shapes. Once the user is ready to take a picture of the game board, the user advances the program again, which takes a picture of the current game board. After this, the program will continue until it finishes without any further user input. After getting both images, the program subtracts the background image from the game board image to get the shapes. With this background-subtracted image, we fix the pixel intensities and then convert the image to grayscale for use with a binarization function. After turning the image into a binary image, we erode it to get rid of artifacts and then dilate it to bring some of the pixel mass back. Next, we take the region properties of the binary image to find the shapes. If a detected object has significant mass, we label the marker directly with its respective shape and a color tag based on the radius of the shape and the pixel colors. We do this for all elements detected with the region properties function. Finally, we display the labeled game state image.

As a whole, there are a few data structures that the program uses to store data about the images and games. Table 2 below shows the data structures that are used in our program and the purpose the structure serves.

Table 2. Data structures used in the image processing program

Data Structure	Purpose
“gamestate”	Primary data structure holding data for the current state of the game. Holds original images and binaries. Also holds the location and color of all detected objects.
“STATS”	Region property structures saving centroid, area bounded box and axis measurements for a detected object
“centroid” and “color”	Arrays storing the positional and classification results of each marker. Found within “gamestate” structures

Overall the program is designed to utilize MATLABs image processing toolbox to utilize simple image differencing and efficiently and reliably produce marker information per the customers specifications [6]. This program is designed with efficiency in mind and does not overutilize computer resources to achieve its results. Furthermore, with the development of a user confirmation dialog box we prevent any potential misreading as a result of poor background and foreground imaging as we allow the user to set up the gameboard to acceptable standards before proceeding with the program's functions.

Engineering Requirements for the Entire System

Once the team had a good understanding of what was being asked, the overall project engineering requirements had to be determined. We do this in order to understand the problem statement and apply what was used in this project to the overall bowling project. By converting the given customer requirements into engineering requirements, we can develop a plan to fulfill said customer requirements. The verification tests can be used as checkpoints to incrementally verify every small part that makes up the system. The customer requirements translation is presented in Table 3 below.

Table 3: Overall Project Customer → Engineering → Verification Requirements

Customer Req.	Engineering Req.	Verification Test
"The user will place the pins in their desired location…"	Capture the board before and after pins have been placed. Process the images to locate the placed pins.	Use MATLAB to place and store the locations of random pin placements. Test various configurations.
"The user… will input (via the GUI) what pin or pins need to be hit."	Develop a graphical interface that allows the user to select a pin to be knocked down, based on the initial location of the pins from program start.	Using the graphical interface, select the target pin and check to see if the location of the target pin is stored in MATLAB.
"...the system must determine the most efficient way to achieve the user's goal…	Calculate the possible ways to knock down the desired pin based on the other pins present.	Based on pin locations, use MATLAB to calculate and return the position at which the robot needs to be aligned in order to hit the pin
"...take action accordingly."	Design a mechanism to launch the ball at sufficiently high speed to knock down the target pin (and other obstacles)	Manually try various power levels and angles to verify that the ball can hit all pins
"...the designed system should be quiet…"	Avoid harsh movements. Stray away from motors when possible.	Test the amount of movement that can be used to reduce sound without decreasing efficiency.
"... efficient, fast…"	Avoid unnecessary movements. Quick response time to user inputs.	Compare the system's response time from initial user input with a threshold response time.
"...safe, and user-friendly…"	Avoid exposed mechanics or electronics. The graphical interface should be clean and easy to understand.	Poll the non-technical group on perceived safety and cleanliness of the system as a whole.

Design Solutions

With an established grasp on the project, the group came together to brainstorm several ideas and then used this collection of ideas to try and come up with an optimal solution. Our plan for brainstorming was for everyone to individually draw their idea of what they had in mind for this project and compare. Everyone had very different ideas, which we then found the pros and cons of each. We used a weighted matrix to decide on a final idea. After our final design was in mind, we made a mockup using cardboard and paper just to portray our idea. Figure 4 below shows a few of the design solutions we created.

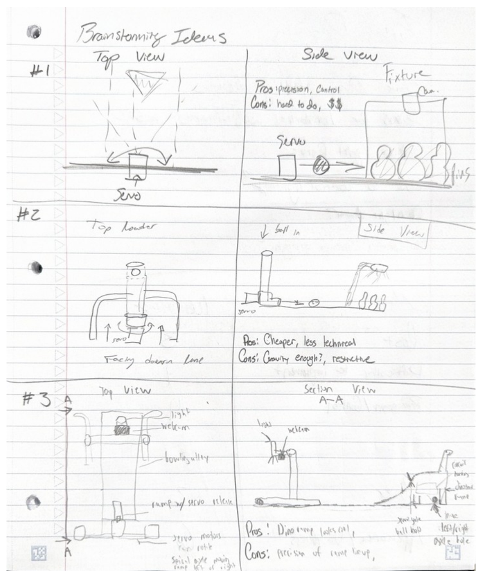

Figure 4: 3 Top Brainstormed Ideas

To briefly describe these, we brainstormed three feasible concepts, sketched and critiqued them, and then retained the best.

1. Concept A — XY gantry with moving camera: Camera on 2-axis carriage scans the field; high coverage but slower and mechanically complex.
2. Concept B — Fixed overhead camera (selected): Rigid PVC frame, fixed standoff, constant LED; simplest, fastest path to robust vision; minimal mechanics.
3. Concept C — Smartphone camera dock: Phone on stand streaming frames; easy setup, but variable optics/auto-exposure complicate repeatability.

Using the weighted matrix(mentioned on the next page), we determined that the general basis for design concept two would be the most optimal way to implement the project. Although design concept two was determined to be the best, we decided to incorporate elements from the other design solutions into the final design. Figure 5 shows our mockup implementation for this design concept.

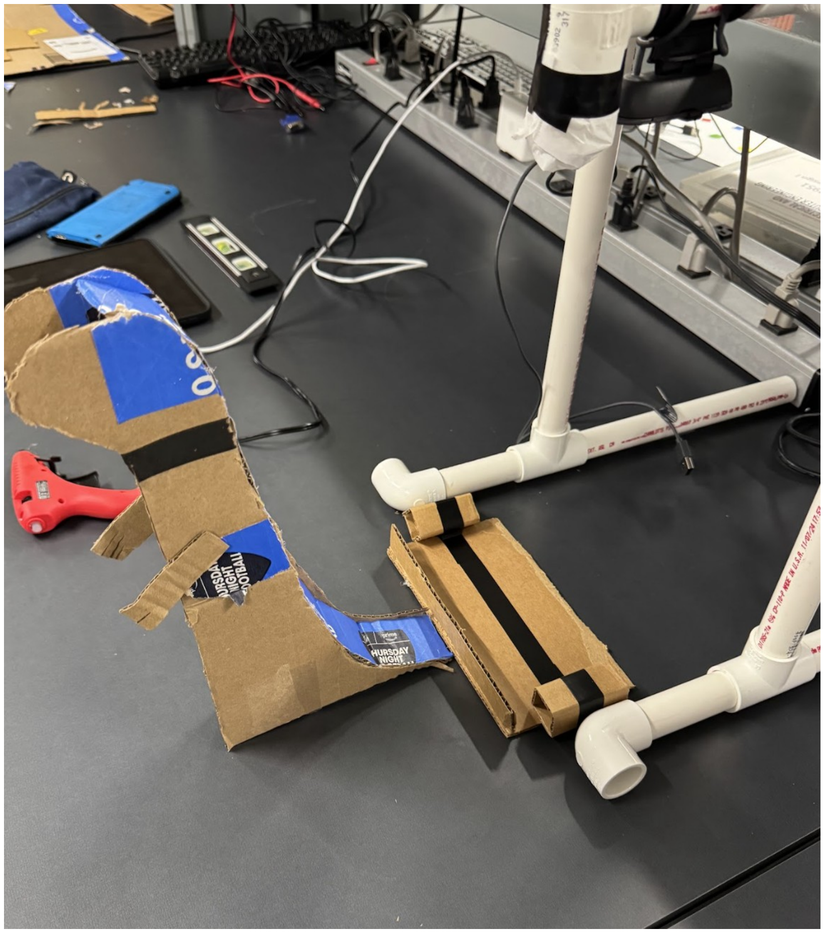

Figure 5: Final design concept mockup

Concept Evaluation

After brainstorming several concepts for the project, the team created a weighted decision matrix to evaluate the pros and cons of each design concept. We started by establishing seven criteria upon which each model will be ranked. The criteria are as follows:

1. Accuracy (25%): Accuracy is the estimated measure of how well the given design concept would target and hit a selected pin.

2. Launch Power (20%): Launch power is the estimated measure of how much power (and speed) a given concept could transfer to the ball when launched.

3. Ease of Implementation (10%): Ease of implementation represents how complex a given design concept would be with regards to the physics and construction of the design.

4. Cost (15%): Cost is the estimated total price for the components and implementation of a given concept.

5. Speed/Efficiency (25%): Speed and efficiency relate to how quickly and effciently the design will be able to react to user input and complete the desired task.

6. User Friendliness (5%): User friendliness relates to how easy the design is to interact with at runtime.

7. Appearance (5%): Appearance relates to the overall physical appearance and cohesiveness of the design concept.

Using these criteria, we then ranked the design concepts we came up with based on each criteria. Table 4 shows the weighted decision matrix the team used.

Table 4: Weighted Concept Evaluation Matrix

Criteria	Weight	Design Concept 1	Design Concept 2	Design Concept 3
Accuracy	0.25	5	3	4
Launch Power	0.2	3	4	3
Ease of Implementation	0.1	1	5	3
Cost	0.15	3	5	4
Speed/Efficiency	0.25	2	3	3
User Friendliness	0.05	4	5	5
Appearance	0.05	1	3	5
Score		2.375	2.8625	2.6

Overall, we selected Concept B (fixed overhead camera) because it maximizes detection accuracy, color stability, and repeatability while meeting the ≤ 1s latency and low-cost requirements. The rigid mount removes scan-time and alignment errors, the controlled LED improves hue separability, and the simple USB interface accelerates software integration. The cardboard mockup demonstrates accurate background field coverage, sensor mounting, ball path clearance and built-in tolerance for noise reduction. The final construction using this concept looks to be robust.

References

[1] Team 19, “Camera Requirements,” internal course document, Clemson Univ., ECE 4950, Fall 2025.

[2] Team 19, “Engineering Requirements—Customer to Engineering Mapping and Verification Plan,” internal course document, Clemson Univ., ECE 4950, Fall 2025.

[3] Team 19, “frankenstein.m — MATLAB Prototype for Game-State Sensing,” internal code/documentation, Clemson Univ., ECE 4950, Fall 2025.

[4] MathWorks, Image Processing Toolbox and Image Acquisition Toolbox Documentation, MATLAB R2025b, Natick, MA, USA, 2025. [Online]. Available: https://www.mathworks.com/help/

[5] Logitech, “Brio 100 Full HD Webcam — Technical Specifications,” 2025. [Online]. Available: https://www.logitech.com/

[6] A. Abdulrahman and S. Varol, “A Review of Image Segmentation Using MATLAB Environment,” 2020 8th International Symposium on Digital Forensics and Security (ISDFS), Beirut, Lebanon, 2020, pp. 1–5, doi: 10.1109/ISDFS49300.2020.9116191.

Score	Pts		
	5	**General Format - Professional Looking Document/Preparation (whole document)** a) Fonts, margins (11pt, times new roman, single spaced. 1" margins on all sides). b) Spelling and grammar are correct c) Layout of pictures – all figures need numbers and captions and must be referenced in the text d) Follows the page limitations below. e) References. Use IEEE reference format. f) This grading sheet is included as the final page.	**O3-SA1:1**
	5	**Page 1: Title, Group Name, Group Members, and Date** **Executive Summary** (1 concise, well-written paragraph) Provide an overview of this project. Briefly describe what you did and what you learned.	**O3-SA1:1**
	20	**Subsystem Design** **Page 2: Engineering Requirements for the Camera subsystem** (~1 page) In the context of just the Camera-as-a-Sensor, make a two column table that contains a column for the Customer Requirements (what are the functions of the sensing system?) and the resulting Engineering Requirements. Each row should contain a specific customer requirement and the resulting engineering requirement. One customer requirement may generate multiple engineering requirements. For example, the customer will want an "accurate" system, the Engineering Requirement could be 99.5% detection success.	**O1-SA5:2**
	20	**Pages 2-3: Document Hardware** (1 page) Describe and show images of the equipment used, connection diagrams, calculation of resolution – pixels per square inch/cm on game board etc. Is the camera an appropriate sensor?	**O2-SA2:1**
	20	**Pages 3-4: Document Software** (2 pages) Using Flowcharts, state diagrams, data structures etc. describe how the software is implemented. There is no need to include the source code.	**O2-SA2:1**
	30	**Overall System Design** **Page 5: Engineering Requirements for the Entire System** (~1 page) Make a three-column table that lists the Customer Requirement in the first column and the resulting Engineering Requirement(s) in the second column. Note that a customer requirement might branch to multiple engineering requirements. The third column should list the test that will be done on the prototype to verify that the design chosen meets each requirement.	**O1-SA5:2**
		Pages 6-7: Design Solutions (~2 pages) Use a brain storming session to generate concepts. Document your top three most feasible ideas with sketches and brief descriptions while explaining the main features of each concept.	**O2-SA1:4**
		Page 8: Concept Evaluation (~1 page) Use a complex (weighted) Concept Evaluation Matrix to show how the final design was chosen from the three best ideas described previously. Include a description of the weighting factors. Make sure to use at least six of your most important criteria in the matrix and be sure to provide and include the weighting factor (i.e. importance) for each criterion.	**O2-SA1:4**

		Describe your final design choice (~1 paragraph) The robot mock up will be graded based on the following criterial: a) Does it demonstrate the proposed concept in sufficient detail? b) Is it well thought out? c) Does it look like the final construction using this concept will be robust?	
		Page 9: Grading Sheet	
	20	**General Website Format** – ***provide link here:*** Is the website: a) Aesthetically clean b) Complete. Does it include: 1. The Team Description? 2. Reports 1 and 2? 3. Outline of future sections? 4. Proper use of graphics? c) Follow accessibility compliance at: https://siteimprove.com/en-us/accessibility/ada-compliance-website/	**O3-SA1:1**
	10	**Demo Slideshow Presentation** (~3 slides) a) What did you do to complete this project? b) How did all the team members get involved? c) How do you believe your team dynamic is?	

Website Link: https://sites.google.com/g.clemson.edu/team19-thebiglebowski/home