Autonomous Baseball Rover Report

Will Bricca, Amanda Katz, Alec Perkins, Dylan Kauffmann

***Abstract* – Retrieving baseballs after practice is a repetitive and time-consuming task for players, parents, and coaches. This project introduces an autonomous rover capable of detecting, approaching, and collecting baseballs using computer vision and a rotating broom mechanism. With primary users being youth baseball coaches, this project aims to streamline post-practice cleanup, reduce manual labor, increase efficiency, and provide a scalable solution for field operations.**

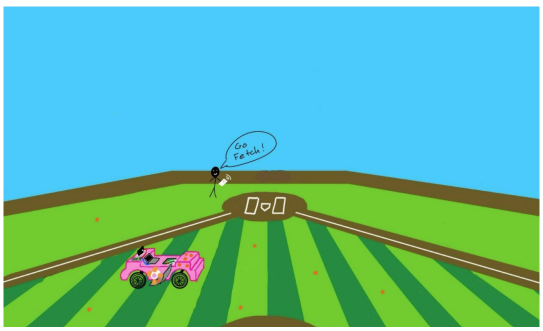

Fig. 1. Sketch of autonomous baseball rover in action

I. Prototype 1

Plan: Our first prototype explores the scope of our project as we start the innovation process with the goal of successfully creating an autonomous rover.

- Requirements: Develop a Python-based vision system using OpenCV to reliably detect neon orange baseballs, integrate a third-party application to stream high-resolution video from an iPhone camera to enhance detection accuracy, and design a wheel well with a rotating mechanism that actively guides baseballs upward into the collection system, ensuring smooth and consistent ball transport.
- Design: Color masking and contour detection techniques were applied to reliably identify neon orange baseballs in the video feed. Additionally, a custom wheel well was 3D printed with precise dimensions to fit a standard baseball, enabling controlled guidance of the ball into the collection system.
- Calculation: Required ramp angle θ estimated using $\tan\theta = \frac{h}{L}$, where $h = 0.05$ m and $L = 0.15$ m $\Rightarrow \theta \approx 18.4^\circ$.

Test: The video feed successfully detected orange contours, confirming the viability of our vision pipeline. However, it occasionally misidentified similarly colored objects—such as hair or other orange-toned items—as baseballs. Further tuning of the color masking and contour filtering will be required to improve accuracy in our specific environment. Mechanical testing showed that the current wheel well design was unable to complete full ball transport due to the baseball's weight. This highlighted the need for a more robust system with active guidance, prompting design revisions.

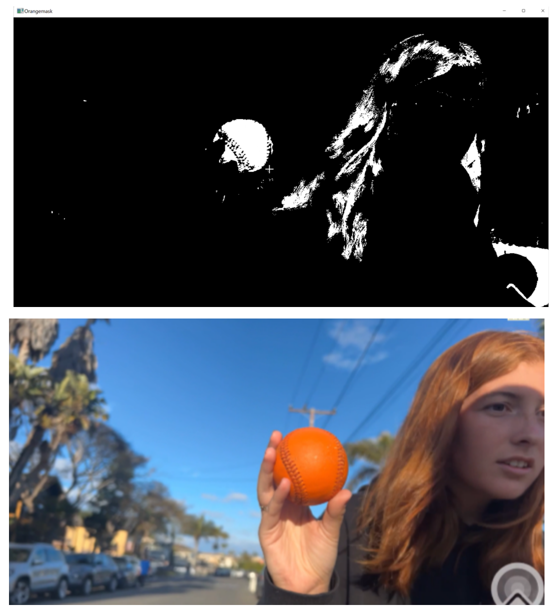

Fig. 2. Open CV ball detection

Learn: It is a challenge to build adaptable systems. Each color mask required careful tuning to specific lighting conditions, shapes, and hue ranges, making consistent detection difficult across different environments. This highlighted the importance—and difficulty—of creating a general, one-size-fits-all solution. Achieving a more robust and flexible detection system remains a short-term goal.

On the mechanical side, testing revealed that the ball pickup mechanism needs to be wider and redesigned to engage more effectively with the ground. A more clever approach will be required—potentially one that allows the collector to rotate with the ground to generate additional momentum and ensure reliable ball intake.

II. Prototype 2

Plan: Prototype two will focus on implementing accurate distance estimation to the baseball using the iPhone camera's specifications and optical principles. This will enable precision detection. This distance estimation allows sets the stage for the system to be capable of autonomous movement. Additionally, we continue refining the ball pickup mechanism.

Manuscript received October 16, 2025. Project conducted in ME 153, UC Santa Barbara.

Fig. 3. First prototype of a 3-D printed wheel well

- Requirements: For Prototype 2, the system must use computer vision to detect and accurately calculate the distance to an orange baseball. The RC car should be able to drive either remotely or autonomously. Additionally, the ball pickup mechanism must be redesigned to reliably collect baseballs from the ground and function effectively during movement.
- Design: The software was designed to include a real-time interface that calculates and displays the distance between the camera and the detected baseball, using live calibration based on optical parameters. On the mechanical side, we designed a radial intake system featuring five broom arms mounted at 72° intervals within a PVC pipe. Each broom is sized to be short enough to graze the grass, yet long and sturdy enough to lift a standard baseball into the intake path. To ensure consistent and powerful collection, the pickup mechanism is powered by its own dedicated motor.

Fig. 4. Second prototype of broom mechanism to pick up baseballs in a wheel well

- Calculation: To estimate the distance from the camera to the baseball, we used principles of similar triangles.

  The relationship between the real-world diameter of the baseball d, its pixel width in the image w, the focal length of the camera f, and the distance from the camera to the object ℓ is given by:

  $$\frac{d}{\ell} = \frac{w}{f}$$

  Rearranging for distance:

  $$\ell = \frac{d \cdot f}{w}$$

  Where:
  - d = diameter of the baseball (inches or centimeters)
  - f = focal length of the camera (in pixels)
  - w = measured pixel width of the baseball in the captured image
  - ℓ = distance from the camera to the baseball

  This formula allows us to compute the distance dynamically in real time as the baseball moves through the frame.

Test: To test the distance formula, a ball was placed at 24 inches away. The computer then analyzed the number of pixels in diameter was was and used the logic of similar triangles to calculate the pixelated focal length of the iPhone 14 camera. Once this was acquired, the program will use these calculations to estimate distance. The picking up mechanism will be tested by driving the motors of the RC car as well as the picking up mechanism at different PWM values to see if the brooms can overcome the friction of the surfaces, such as the ground and the grass.

Learn: The distance calculations were successful with a slight margin of error. The similar triangle method to calculate the distance is a good enough estimation for the purposes of our rover. The picker upper mechanism can overcome the friction of both pavement and grass, making it a reliable picking up mechanism.

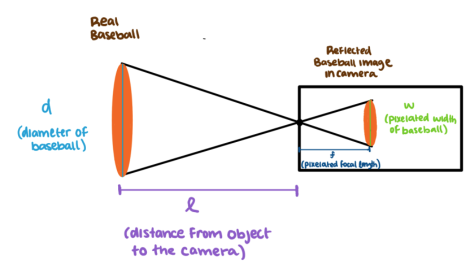

Fig. 5. Sketch of focal length calculation

III. Prototype 3

Plan: This prototype will focus on the efficient integration of the broom pick up mechanism and the wire assembly from an Arduino to 3 motor drivers to the motors.

- Requirements: Assemble the wiring to the 3 motors, the motor drivers and the Arduino. An Arduino Uno should be able to control the 3 motors at various PWM values.

The motors should be able to go forward and in reverse. The picking up mechanism should attempt to pick up a stationary ball.
- Design: Cardboard ramps were placed in between the each broom to ensure the ball will roll down sideways to go into our rover.
- Calculation: Adjusted broom rotational speed to match rover approach speed: $v = r\omega$.

Test: The rover was driven on pavement with a baseball placed in line with the rover and with enough distance so that the rover can easily drive forward and pick up the baseball without having to steer towards it. During testing, the wiring had faulty connections and there was no common ground. At first, when powering the motors, they still rotated and worked well, giving the team no indications of problems within the circuit despite the messiness of the wiring. However, after encountering rough terrain, the RC car rustled around and one of the motor drivers sparked and all three of the motor drivers and the Arduino fried.

Learn: The broom pick up mechanism is strong enough to pick up a baseball. If the speed of the brooms exceeds the motor speed of the RC car, the ball is likely to not slide down the ramps and will continue to be spit out in front. As for the failure of the motor drivers and Arduino, a valuable lesson learned was that it is important to create a common ground. If the team took a step back, planned better,used the knowledge acquired from circuits and consulted the Professor earlier, this could be prevented. Thus, although it is tempting to work fast and jump in, it is beneficial to plan effectively prior to testing to avoid costly and time-consuming mistakes.

IV. Prototype 4

Plan: This prototype will explore different strategies of steering the rover. Specifically, the team had to address wiring and hardware layout issues following the frying of essential components. Additionally, a stable iPhone stand was mounted for accurate testing, and enhanced rover maneuverability. The control based code continued to be developed, preparing for communication from Python to the Arduino which would allow for the RC car to move autonomously.

- Requirements: The mover must be able to steer quickly and effectively on different terrains. The code must make enhancements and be ready to deploy once rover can turn.
- Design: To fix the camera stability issue, we attached a phone tripod holder directly to the rover chassis using super glue and zip ties. Resistance bands will connect the two tires together to attempt a tank-style design for turning. Wiring issues were resolved by soldering loose connections and insulating them with electrical tape. Neatness of the circuit ensured easy troubleshooting. All components required to share a common ground, which significantly improved power delivery and reliability during tests. On the software side, we advanced our Python code to implement a fully autonomous state machine with four distinct modes:
  - Mode 0 – Standby
  - Mode 1 – Align (center the ball)

Fig. 6. Prototype 3 with implemented phone stand

Fig. 7. Updated motor wiring configuration

  - Mode 2 – Drive (approach the ball)
  - Mode 3 – Search (scan environment)

Mechanically, tank-style turning method involving resistance bands wrapped between wheels. The idea was to improve turning torque and simulate a differential drive. However, the implementation was unreliable—the bands slipped frequently and did not provide consistent performance. After these tests, caster wheels deemed to be a more viable option, which offered better maneuverability and were easier to implement.

- Calculation: Necessary wheel separation was estimated to apply proper band tension in the tank-style configuration. The calculations were based on torque and turning radius, but in practice, the friction losses and slipping made it impractical.

Test: The resistance bands were tested by driving either the left or the right motor forward to see if turning was possible. The caster wheels were then installed and were tested in the same fashion as the resistance bands. The code was tested by one team member hold a ball in front of the camera and move around to test the ability for the rover to track a ball all while testing the quality of the rover's steering, the phone stand, and the durability of the wiring.

Fig. 8. Conceptual tank-style rover configuration

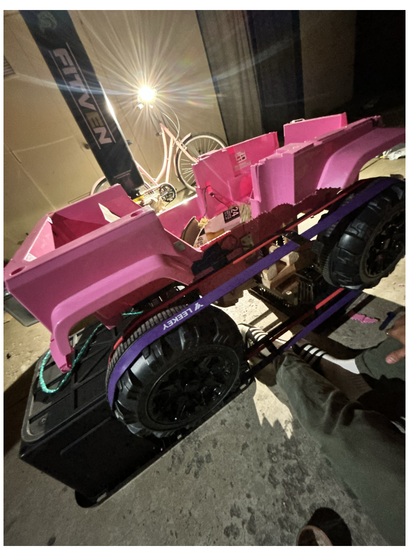

Fig. 9. Implemented tank-style configuration using resistance bands

Results: During preliminary testing of the rover, we experienced severe lagging when OpenCV was running. Initially the problem seemed to be a slow computer or unstable connection between the phone and the camera, but after substantial testing the real problem became apparent. Our python code updated the arduino at such a fast rate that the arduino could not process the data. Specifically our team observed the arduino reaching a critical point where it could not handle any more data inputs to the serial monitor which would effectively cause the software to freeze. This was a huge bottle neck because we could not continue testing until this problem was resolved. On the mechanical side, the phone mount provided a clear and stable view of the of the ball allowing approximately 23 feet of visibility. The soldered wiring held up well under vibration. The tank-style idea was implemented but performance was poor due to slipping and uneven torque transfer. The caster wheels allowed for quick, reliable steering, being able to turn instantly.

Learn: Over-complicating the mechanical design without robust prototyping can lead to performance issues. The resistance band approach added complexity without clear benefit, while the simple caster wheel fix was much more effective, even though it was more expensive. Structuring the code with separate autonomous modes also helped a lot—it made debugging easier and gave us more flexibility as we moved toward a full autonomous system. Analyzing and understanding the way the system was behaving when OpenCV was running and communicating to the arduino revealed a lot about how the system worked. Most importantly, this discovery allowed us to debug the freezing problem by simply limiting the inputs to the serial monitor so that the arduino serial monitor would not get saturated.

V. Prototype 5

Plan: The final goal was to achieve a fully autonomous retrieval cycle on a standard baseball field, ensuring the rover could reliably navigate and pick up baseballs under varied environmental conditions including concrete, grass, and turf. The code should incorporate PID control to accurately align the rover with the detected baseball and take advantage of the quick steering.

- Requirements: The rover must be able to detect when a ball is stationary, align its ball collecting channel towards the ball, drive forward in order to ensure pickup, then rotate continuously to find other balls to pick up nearby.
- Design: This prototype focused on final integration. The improved turning of the rover allowed a focus on tuning the PID controller to match different surfaces. Chamfers were added to reduce drag, especially during pickup.
- Calculation: PID gains were tuned (K_p, K_i, K_d) for each surface:
  - Concrete: Required low power and minimal gains.
  - Grass: Demanded high motor power and higher proportional gain.
  - Turf: Performed well with intermediate tuning.

  The integral and derivative terms were adjusted to reduce overshoot and improve steady-state behavior.

Test: The rover operated on concrete first. It responded well and needed only light tuning, but camera angle adjustments were critical to align with the ball. On thick grass, we increased motor power and PID gains. The robot initially struggled to turn, so we modified the turning logic. On turf, the rover completed multiple successful autonomous retrievals using the final gains. Vision and motor systems worked well together.

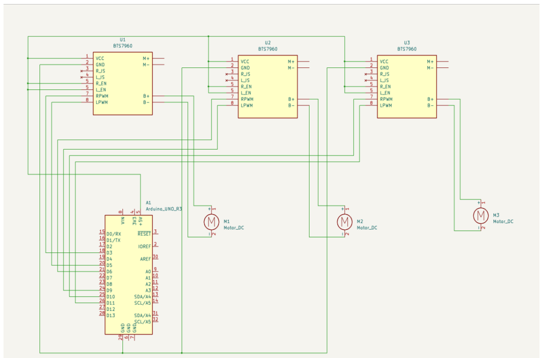

Fig. 11. Complete circuit diagram

Learn: Adapting a PID controller to accommodate many different terrains proves to be quite difficult, and it was much easier to tune the rover to specific situations it would encounter such as the concrete, turf and grass. Long grass required much more gain due to high rolling resistance. Overshoot was reduced by tuning the derivative and integral terms. In some scenarios, like the concrete, the derivative and integral terms were unnecessary. After several full runs, the system consistently retrieved baseballs. The combination of mechanical upgrades, gain tuning, and improved software logic made this prototype our most successful.

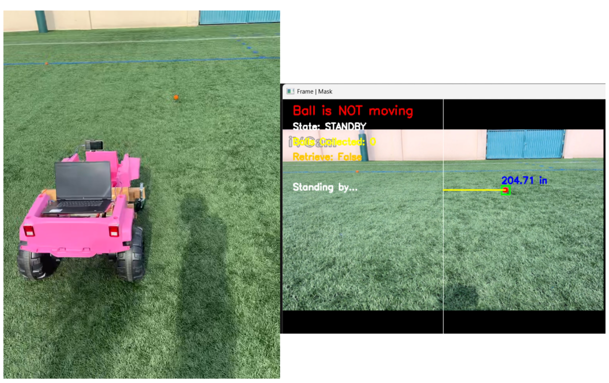

Fig. 10. Final prototype of rover displaying end goal of autonomously picking up baseballs

VI. Reflection

From the start, this project was very ambitious. The goal was not to just make an RC car, but to build a fully autonomous rover that could detect, track, and retrieve baseballs outdoors across different terrains. That scope pushed the team technically, but also forced us to learn how to work effectively as a team.

One of the biggest takeaways was the importance of planning and communication. Early on, building was often rushed without a complete strategy, which led to setbacks—burned motor drivers, disconnected systems, and delays during integration. Moving forward, tasks were done slower and methodically, tasks were divided more intentionally, and check ins were regulated on both hardware and software progress. Better coordination helped the team work more efficiently and avoid repeated mistakes.

A valuable lesson learned was that prototyping is essential towards all components the system. Testing things in small pieces helped build understanding of what would work, what wouldn't, and where the design needed to evolve. For example, trying out the tank-style drive system revealed its limitations early, which saved time in the long run. The iterative nature of prototyping improved the team's vision, control, and collection mechanisms in a realistic, test-driven way.

Another lesson was the value of merging code and hardware sooner. Once the team moved to a shared software structure with autonomous modes, it was easy to iterate quickly and test the full system in real conditions. That shift helped bring the whole project together.

A major takeaway from this project was realizing how difficult it is to build something that works reliably across different environments. Whether it was detecting the baseball, tuning the PID controller for smooth alignment, or making sure the picker upper didn't spit the ball back out—every component had to be adjusted based on where the rover was operating. The way the rover moves changes drastically depending on the surface—grass, turf, or concrete—which meant the PID settings had to be constantly tweaked. What worked well on grass would cause overshooting on concrete, and smoother surfaces required far less correction. There was no one-size-fits-all solution.

At first, this constant tuning was frustrating. But over time, the team learned how to test in real conditions, recognize what needed adjustment, and make quick, thoughtful changes. It took a lot of trial and error, but that process ultimately made the rover more reliable and adaptable to real-world challenges. In the end, the ability to tune and refine each part of the system was one of the biggest factors behind the prototype's success.

In the end, the final rover fulfilled the core objective. It completed multiple autonomous retrievals, handled real-world conditions, and reflected a ton of learning. Beyond the technical achievements, experience was gained in team dynamics, design iteration, and building something ambitious from scratch. While there is still room for refinement—such as strengthening the intake system and enabling adaptive gain tuning—the team finished with a reliable, functional system and a solid foundation for future development.

Acknowledgments

We thank Dr. Elliot Hawkes, Dr. Trevor Marks, and Ryan Abell for their support throughout the quarter.