Objective: Orientation Detection of a High Speed Projectile

Background

In baseball analytics, understanding the exact orientation of the ball at release is important to calculate its true spin. Your task is to design and implement a perception pipeline that identifies the orientation of a baseball by detecting its seam patterns using a single high-speed monocular camera.

- **Target:** An official baseball (9-inch circumference, 108 hand-sewn red stitches).
- **Environment:** A laboratory with controlled high-intensity strobe lighting.
- **Camera:** A high-speed monocular camera mounted 3–5 meters from the pitch path.

Part 1: System Design

Before processing pixels, you must ensure the data is usable. Describe the following on how you'd design the system & why

1. Calculate the maximum allowable exposure time (shutter speed) to keep motion blur of the red seams under 2 pixels when the ball is traveling at 100 MPH.
2. At 5 meters, what lens focal length and sensor resolution are required to ensure the ball occupies at least 200 pixels in diameter?
3. Discuss the trade-offs between a traditional Hough Circle Transform vs. a Deep Learning (e.g., YOLO/segmentation) approach for real-time edge deployment
4. Explain the visual challenge of detecting spin when the ball's axis of rotation is parallel to the camera's optical axis (bullet spin). How does your detection logic change if the seams are effectively "stationary" in the center of the ball?

Part 2: Basic Implementation

Provide a Python prototype focused on orientation detection. You may use any tools/libraries with our provided dataset

1. Implement a robust method to detect the circular boundary of the baseball in a noisy, high-speed frame
2. Implement a pipeline that takes the cropped "ball-region" from above and extracts the pixel coordinates of the red seam segments
3. Implement a pipeline that takes a raw image of a baseball and extracts the pixel coordinates of the red seam segments. Your code should be robust to slight motion blur and lighting variations.

Part 3: Bonus Implementation

4. Using the seam segments detected in Part2, implement an approach to determine the ball's 3D orientation relative to the camera.
    - If you have 5 consecutive frames of the ball, implement a simple logic to ensure the estimated orientation is physically consistent (i.e. the ball shouldn't rotate 180 degrees in 1 millisecond).
5. Write a test case that validates your coordinate transformation from the Ball-Local frame to the Camera-Reference frame.

Part 4: AI Collaboration Log (Mandatory if used)

Document your use of AI tools (Claude, ChatGPT, etc.) in this exercise

- **Prompt History:** List all prompts used to solve the problem
- **Critical Review:** Describe one instance where an AI suggested a "standard" CV approach that you realized would fail in a high-speed robotics context, and how you pivoted.
- **Optimization:** Did you use AI to optimize your code for the timing or processing performance? If so, what was the performance gain?

Deliverables

1. **System Design Document:** As defined in Part 1
2. **Prototype:** Modular code implementing the 3-part pipeline
3. **AI Usage Report:** As defined in Part 4