Projectile Motion to Hit Fixed Targets: Methodology Document

Gia Dvalishvili

Kutaisi International University
Introduction to Numerical Programming

1 Problem Statement

The objective of this project is to calculate the initial conditions (velocity v_0 and angle θ) required for a projectile to hit fixed targets detected in an input image. The process involves:

- Detecting circular targets in the image.
- Using the shooting method to compute the initial conditions.
- Visualizing the results through animations of the projectile motion.

The **shooting method** is suitable because it iteratively determines the optimal v_0 and θ that meet the target boundary conditions. Its flexibility makes it ideal for handling varying trajectories and physical constraints.

2 Mathematical Formulation

2.1 Equations of Motion

The equations governing projectile motion are:

$$x(t) = x_0 + v_0 \cos(\theta) \cdot t, \tag{1}$$

$$y(t) = y_0 + v_0 \sin(\theta) \cdot t - \frac{1}{2} g t^2, \tag{2}$$

where:

- x_0, y_0: Initial position.
- v_0: Initial velocity.
- θ: Launch angle.
- g: Acceleration due to gravity (9.81 $\mathrm{m/s^2}$).
- t: Time.

2.2 Boundary Conditions

For a target at (x_t, y_t), the task is to find v_0 and θ such that:

$$x(t_{\text{impact}}) = x_t, \tag{3}$$

$$y(t_{\text{impact}}) = y_t. \tag{4}$$

The time of impact t_{impact} is computed using $x(t)$, and $y(t)$ is verified to ensure the projectile reaches the desired height.

2.3 Gaussian Blur

Gaussian blur is applied to the image to reduce noise and improve target detection. The Gaussian kernel is defined as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \tag{5}$$

where σ is the standard deviation of the Gaussian distribution. This operation smoothens the image by averaging pixel intensities in a localized area.

2.4 Hough Circle Detection

Circular targets are detected using the Hough Transform, specifically the `HoughCircles` function in OpenCV. The method identifies circles by searching for local maxima in a parameter space defined by:

$$(x - a)^2 + (y - b)^2 = r^2, \tag{6}$$

where (a, b) is the circle center and r is the radius.

3 Numerical Methods

3.1 Shooting Method

The shooting method iteratively solves for v_0 and θ:

1. Initialize ranges for v_0 and θ.
2. Compute the trajectory using Equations (1) and (2).
3. Compare the trajectory endpoint with the target (x_t, y_t).
4. Minimize the error $|y(t_{\text{impact}}) - y_t|$.

3.2 Numerical Properties

- **Truncation Error**: Results depend on the resolution of v_0 and θ. Higher resolution improves accuracy but increases computation time.
- **Stability**: The equations are stable due to their bounded nature and reliance on straightforward physics.

4 Algorithm

4.1 Target Detection

1. Convert the input image to grayscale.
2. Apply Gaussian blur to reduce noise using Equation (5).
3. Use OpenCV's `HoughCircles` to detect circular targets.
4. Map detected circles to grid coordinates.

4.2 Velocity and Angle Calculation

1. Iterate over a range of v_0 and θ.
2. For each pair, calculate the projectile's trajectory.
3. Check if the trajectory endpoint is within an acceptable error of the target.
4. Record the best v_0 and θ minimizing the error.

4.3 Animation

1. Generate trajectories for each target using the calculated v_0 and θ.
2. Create sequential animations visualizing projectiles hitting targets.

5 Results

5.1 Best-Case Scenarios

- **Clean Images**: Images with minimal background noise and large, well-defined circular targets are easiest to process.
- **Moderate Targets**: Targets at moderate distances and heights with trajectories that are neither too steep nor too flat.

5.2 Worst-Case Scenarios

- **Noisy Images**: Images with significant background details or small, indistinct targets may lead to incorrect detection.
- **Extreme Trajectories**: Targets that are very high or require very shallow angles can be challenging due to physical or computational constraints.

6 Conclusion

The project successfully computes initial conditions for projectiles to hit fixed targets. It handles clean images and moderate targets well, but noisy images and extreme trajectories remain challenging. Future work could include:

- Optimizing the shooting method for faster convergence.
- Accounting for air resistance for realistic scenarios.
- Enhancing target detection for noisy environments.