Robin Radar Systems C++ assignment

Introduction

Modern radar systems provide value by extracting useful information from raw radar images. Simply displaying the reflected energy is not sufficient for customers. Depending on the radar system, such images are collected at a frequency of about once a second. Important first steps include the detection of moving objects, which are referred *to as plots,* and correlating these into trajectories of the objects, known as *tracks*. Then, tracks are given a classification such as *bird* or *aircraft*. Finally, algorithms that extract useful information from all steps before are run.

Conceptually, the following happens:

1. In an image processing pipeline, a radar image is analyzed in order to find (potential) plots.
2. A tracking algorithm analyzes the shape and intensity of plots over time and space and combines them into tracks.
3. Things such as the shape and size of plots, but also behavior such as speed and altitude, are used to assign a classification to a track.
4. Configurable and often customer-specific algorithms are used to extract useful information and generate alarms or log things in a database.

Natural phenomena such as rain, waves and moving trees complicate things here, as does the erratic/difficult-to-predict behavior of birds, drones, and sometimes cars and aircraft.

Customers are typically only interested in certain kinds of tracks in specific locations, with some kind of specific behavior. For example, an alarm should be raised when a flock of birds enters the runway between 7 am and 11 pm. Or the flight direction of birds should be logged to a database in order to detect whether the migration season has started.

Assignment

Your task is to analyze the motion of a single 2D track. We provide a track and an area of interest as two simple text files. The question that must be answered is whether this track will (likely) enter the area of interest.

We’re primarily interested in how you approach the problem — your assumptions, structure, and clarity of thought. It's okay to make simplifications as long as they’re documented. We expect the assignment to take about **2–4 hours**, please don’t spend more time than that.

Input

The input of your program is two files:

1. *Track.txt* - Each line contains a detection point, one per line: `time, x, y`, e.g:

```
0,-122.714,-64.3468
0.8,-119.444,-63.8805
...
```

2. *Area.txt* - A list of four `x,y` points forming a rectangle:

```
-8.0, 8.0
 8.0, 8.0
 8.0,-8.0
-8.0,-8.0
```

Output

The assignment is to write a C++ program to analyze the motion of the track. For simplicity, results should be written to the console.

Question 1: Is the track on course to enter the area?
Question 2: If so, provide an estimate as to *when* it will enter the area and *where*.
Question 3: Provide an estimate of the *speed* of the track and the *direction* it is moving in (as an angle with the x-axis).

Guidelines:

- Use C++ 17 or newer
- Use only standard C++ (including the standard library)
- Please include build instructions
- Code readability and structure matter more than performance
- The coordinate system is unitless — you may assume x and y are in an abstract 2D plane. Time is given in seconds.
- Please let us know if something is not clear