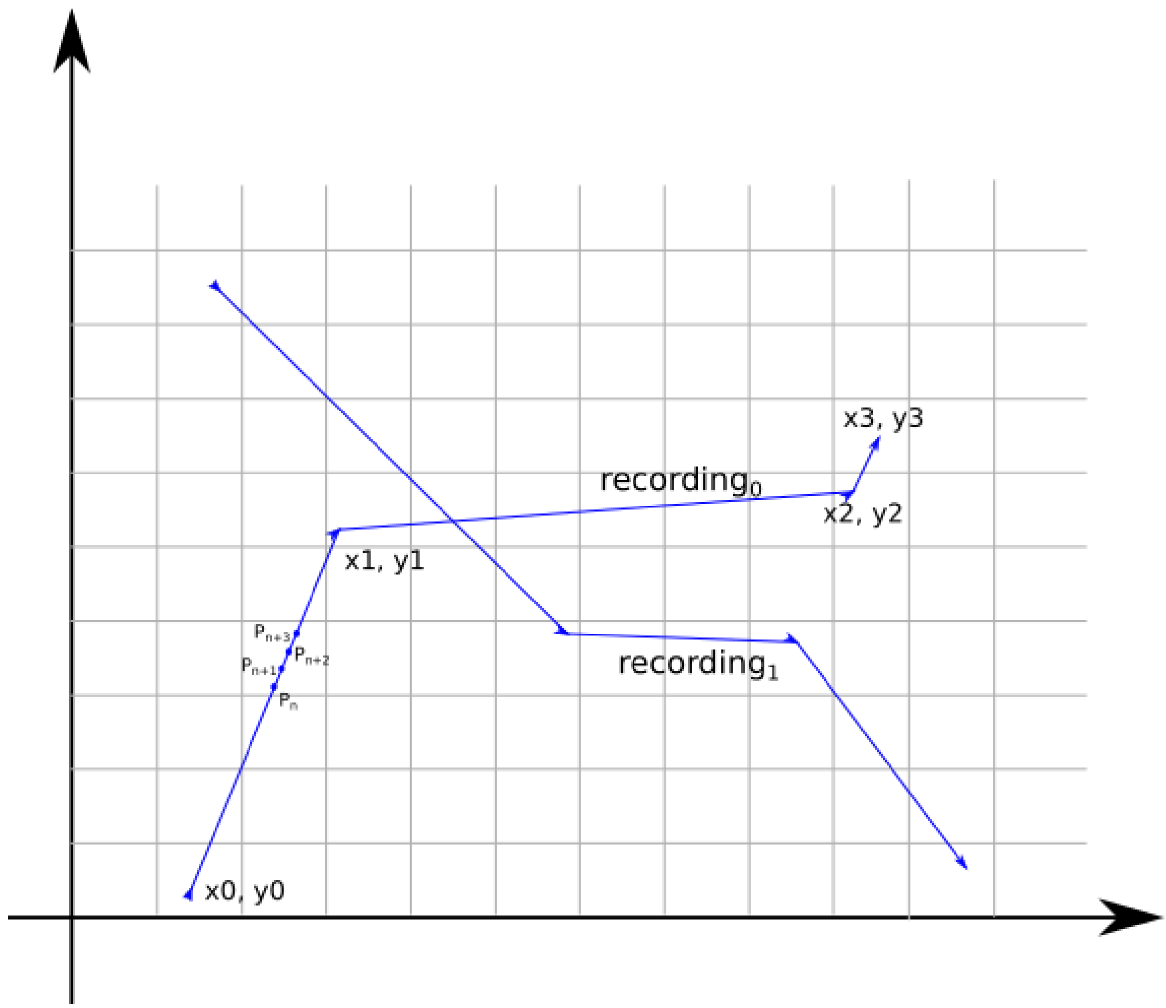
Esri R&D Center Vienna – C++ code assignment

Goal

The goal of this assignment is to 1. understand your current experience with tools / frameworks / libraries related to our daily work and 2. see what code written by you looks like. The result of this assignment is used during a face-to-face meeting as a base for further technical discussions.

Assignment

Complete all or some of the below tasks in a time that you feel is reasonable. If you get stuck on one of the tasks you may skip it and take notes to discuss your problems in the following meeting - or reach out to us via email. If not specified, you are free to choose whatever solution / library you think is the best choice for a given task.



The recordings – collections of raw data collected using our internal apps – are illustrated in the plot above (recording0 and recording1). The data was collected by walking straight between different ground truth points ((t0, x0, y0), (t1, x1, y1), (t2, x2, y2), ...) with constant speed. In the recording, the reported magnetic field has been collected at measurement points (P0~ Pn, Pn+1, Pn+2....). The magnetic field was collected in three directions (magx, magy, magz) and assigned the timestamp it was collected at (t). The data is stored in Protocol Buffers-files – one per recording.

1. In C++, read the given Protocol Buffers-files
    a. Please find our Protocol Buffers definition attached. You can build the C++ bindings using "protoc --cpp_out=build *.proto"
2. Store relevant data using a performant data structure
    a. Magnetic data is located in a field called "magnetics". Ground truth points are located in "positions".
3. Compute the position of each measurement point (Pn) by linear interpolation
    a. Hint: we know the position and timestamp of ground truth points (t, x, y), the timestamp of the measurement point, and the speed are constant. Therefore the position can be computed:
        i. xpn = (tpn - t0)/(t1-t0) * (x1 - x0) + x0
        ii. ypn = (tpn - t0)/(t1-t0) * (y1 - y0) + y0
4. Compute the magnetic field amplitude
    a. mag = (magx^2 + magy^2 + magz^2)^0.5
5. Compute the average magnetic field amplitude in each map grid. The grid is a rectangle grid with size of 5m start from position (0, 0)
    a. Hint: first get all the measurement points in each grid, and then compute their mean value
6. Create an integration test that covers as much of the above code as possible / reasonable
7. Persist the (grid) magnetic field strength map in a file
    a. Use a human-readable format like CSV, JSON, etc
8. Create bash-script to automatically install necessary dependencies, run integration test and your program afterwards
9. Create a short document summarizing your software architecture and your thoughts about it if applicable. Also make sure to describe any technical hurdles you had to overcome and how you did it
10. Upload your code to a git repository and grant access to it for ttaschauer@esri.com