Assignment

Question
A robot carrying a smartphone (Samsung Galaxy Note 8) was driven on a sidewalk and has collected images along with various sensor data such as GPS and IMU. The phone was mounted in landscape orientation with the rear camera facing forward. Unfortunately, all of those sensors operate at different frequencies and produce noisy measurements. You would like to estimate the 2D trajectory of this robot (we assume the difference in altitude is negligible) given all of this sensor data.

Tips: The GPS data should give you a pretty good starting point for the trajectory. There are a couple of images provided to give you some idea of how the robot is moving and a text file that gives you timestamps for all collected images. There are also text files for each sensor with timestamps and readings

Tasks
You're expected to write code to do the following:
1. Read the provided text files and extract the sensor data.
2. Match the timestamps of the different sensors. You may need to interpolate, integrate and smooth some sensor data for the next step.
3. Estimate the trajectory of the robot relative to the first frame using as much (useful) sensor information as possible and try to filter out noise (e.g. Kalman filter).
4. Visualize the estimated trajectory.
5. Given two frame ids as input, return the relative pose of the robot from the first to the second frame. It is sufficient to provide only the 2D translation as a vector and the change in orientation around the z-axis as a scalar.

Guidelines
1. You can use Python, Java or C++. You may also use Jupyter Notebook.
2. Document your code, the algorithm used, how to build and run, etc.
3. The code should be as self-contained as possible.
4. If you use code snippets or libraries written by someone else, you need to provide references.

Bonus
How could you further improve the trajectory estimation using images? Please implement your solution using the provided images and explain how to extend your approach if images for the complete trajectory were available.

Submission
To submit your code, push it into a **private** git repository, and let us know the access method to the repository. Include all necessary dependencies if needed.