Assignment 2 – Zero Velocity Update

For this assignment, you will investigate and implement zero velocity update (ZUPT) methods for foot motion reconstruction. As has been mentioned in the lecture, ZUPT is essential to obtaining correct motion trajectory by stopping double integration errors from carrying forward from each step.

What's provided?

We have provided MATLAB code file *assignment2.m* with an empty ZUPT function that just returns the current velocity unmodified. A description of the high level functions are described below. You should be able to read and understand the logical flow of the program without too much trouble. You do not need to understand the specifics of the functions already implemented.

- *ReadData* reads a raw data file and output a data structure usable in MATLAB
- *DetermineInitialWindow* determines the initial static period of an experiment and uses that period to perform various calibrations
- *preprocess* performs pre-processing of input data from ReadData and returns gravity compensated acceleration signal
- *trajectory_estimation* performs the actual trajectory estimation through double integration of gravity compensated acceleration data. ***zero_velocity_update*** (the one you need to modify) is within this top level function

Task 1 – Data Collection

You should perform the following data collection to obtain data for subsequent tasks.

1. Wear a single sensor on one shoe (on the tip of the shoe)
2. Stand still, start data collection program, and keep still for 5 seconds
3. Walk through 10 meters (33 tiles), stop to still and stop data collection program
4. Record the total walking distance

Task 2 – Hard ZUPT

In this task, you will attempt to implement a hard (simple) ZUPT method.

First, we will explain the code within *trajectory_estimation.* This function simply first performs an integration of the gravity compensated acceleration signal to obtain velocity, and then integrate the velocity to obtain displacement. After the first integration, you will have the opportunity to use the function *zero_velocity_update* to look through the velocity data vector and determine points at which it should be 0.

To implement hard ZUPT, consider the following:

- When is the velocity of your foot 0?

- Can these 0 points (or sequence of points) be found from the gravity compensated acceleration data, from the gyroscope data or from the combination?
- Can you implement the *zero_velocity_update* function so that you can scan through both the gravity compensated acceleration and/or gyroscope data to find the points where velocity is supposed to be 0, and update the velocity vector?
- Is the resulting reconstruction better?
    - Hint: it should be

Task 3 – Soft ZUPT

- Looking at the reconstruction obtained using hard ZUPT, is the trajectory good? How smooth is it? How much error is it making?
- Plot the velocity vector after you have applied your ZUPT, do you see a possible cause?
    - Hint: consider the sharp changes that has been introduced by the ZUPT
- What can be done to improve from task 2? This is an open ended question, but at least you should consider smoothing the velocity data from the hard ZUPT.

Hand-ins

You should hand in the following material. All code should be in a runnable state against your data:

- Collected data and ground-truth step and total distances
- Plot of the original assignment2.m code
- Code from Task 2
- Plot of Task 2 after implementing hard ZUPT
- Code from Task 3
- 2-3 sentences explaining what you have implemented in Task 3
- Plot of Task 3 after implementing soft ZUPT