Q1: When running "runtime.m", we are asked for a movement type: 1 for saccades and 3 for pursuits. I don't see anywhere in the documentation to see how our program should differ when a different movement type value is given. What is the intention of this value?

A: You need to select option 1 for saccades. Option 3 exists for historical reasons only. You will work with a different code for Project 3 which specifically targets classification of smooth pursuit.

Q2: Each Eye Movement Detection (EMD) function prototype that you provided takes an "eye\_record" struct-array as a parameter. It seems that part of this struct is filled in for us with data that makes sense: (x,y) position in degrees and (x,y) velocity in degrees. What is the frequency of these rows? For example, is the second row after 1 ms. of time and the third row after 3ms, meaning frequency is 1 row/ms.?

A: The sampling frequency of your data is 120Hz. Thus, you can assume that you can a sample every 1000/120 milliseconds. In reality, the distribution of eye movement samples is non-uniform, but for the purposes of this class you can assume it is.

Q3: Each function prototype requires us to return several variables (eye\_record, fixation\_filtered\_EMD, etc), but there doesn't seem to be any documentation on what these return values should contain. For instance, I assume we have to fill in several of the empty columns in eye\_record, but based on the IVT example, we don't need to fill out every column and the meaning of each column is non-obvious to me. The same is true for the other return values.

A: You don’t have to worry about those additional variable. You just need to assign eye\_record(t).xy\_movement\_EMD to 1, 2, or 4. 1 means fixation, 2 saccade and 4 noise. The remainder of the code should take care of the rest for you. If you find any explicit bugs please let me know and I award points for your efforts depending on what you find.

Q4: Lines 51-53 of "runtime.m" say we need to create Global variables related to our method. What is the purpose of these global variables? From my understanding, each algorithm should be self-contained within its own function, so global variables seem unnecessary. What is the purpose then of these global variables and what kind of global variables should we be creating?

A: Provided code is inefficient and does not followed rules of good coding for modern programming. Please create variable for your algorithm in the file EMD\_[name of your algorithm].m The idea is that by changing the line [eye\_record,fixation\_filtered\_EMD,saccade\_filtered\_EMD,pursuit\_detected\_EMD] = EMD\_\_[name of your algorithm] (eye\_record); you should be able to run your code

Q5: The project description only mentions uncommenting a line in "main.m" and doing the rest of the work in your individual EMD function. However, it seems that we need to edit other provided functions too (see question 4). Are there any of the other provided functions we need to work on, or can we just focus on our own functions?

A: There are two functions that you need to edit: 1) main where you have [eye\_record,fixation\_filtered\_EMD,saccade\_filtered\_EMD,pursuit\_detected\_EMD] = EMD\_\_[name of your algorithm] (eye\_record);

And the actual function that implements your algorithm.

You don’t have to put any modification to runtime.m or other functions.

Unless you are fixing the bugs, in which case refer to the answer above.

Q6: The eye\_record struct has values '\*\_velocity\_measured\_deg'. However, this seems to be an error since velocity cannot be measured in degrees (a unit of distance). I am assuming that this is degrees per millisecond. Is that correct?

A: Very correct observation. '\*\_velocity\_measured\_deg' represents velocity in degrees per second. The better name for that variable should be '\*\_velocity\_measured\_deg\_sec'

Q7: Are we going to be discussing SQnS, FQnS, and FQlS in class, or should we look these up ourselves to better understand how our algorithms will be measured?

A: We will discuss them in class, but I strongly recommend that you would read a corresponding paper that defines them in detail – “Standardization of automated analyses of oculomotor fixation and saccadic behaviors”.

Q8: How much of pre-implemented I-HMM algorithms available online are we allowed to use?

A: In every project, you have to do a considerable amount of work yourself. If you just blindly use a pre-existing implementation of some hmm function, ask yourself how much of this algorithm do you understand? Do you understand why Baum-Welch / Verterbi are present? How do they improve or have a chance to improve the performance on top of the I-VT method that already pre-classifies the signal initially?

If you don’t understand how it works inside then implement it from scratch. In your report please outline what exact work you did and why what you did makes sense.