|  |
| --- |
| PROJECT#1 |
| [Eye Movement Classification – Fixations and Saccades]  Due Date: specified on TRACS  Human Computer Interaction |

**PLEASE READ THE ISTRUCTIONS CAREFULLY – THEY CONTAIN A LOT OF DETAILS. MAKE SURE TO DOUBLE AND TRIPLE CHECK EVERYTHING.**

**Objective:** The objective of this project is to implement one of the eye movement detection algorithms and get a skill of basic eye movement detection. The project can be completed by teams of 1-3 people, depending on the algorithm.

**Project:**

Select your teammate/s and choose an eye movement classification algorithm from the following list presented in class. The implementation is to be done in MATLAB. All CS department lab computers have MATLAB installed. MATLAB tutorials are available online.

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm type** | **Description** | **Maximum points for successful implementation** | **Maximum number of team members** |
| I-DT | Dispersion-Threshold Identification | 85 | 1 |
| I-MST | Minimum Spanning Tree Identification | 90 | 1 |
| FIR 2/5 tap velocity\* | Finite Input Response Filter – 2/5 tap velocity | 95 | 1-2 |
| FIR 2/5/7 tap acceleration\* | Finite Input Response Filter – 2/5/7 tap acceleration | 96 | 1-2 |
| I-KF\*\* | Kalman Filter Identification | 100 | 1-2 |
| I-HMM | Hidden Markov Model Identification | 105 | 1-3 |
| Your Algorithm | If you decide to create your own algorithm please let me know | 105 | 1-3 |

**Additional penalties:** a)Each member of a group will be penalized if the number of people in a group exceeds the allowed maximum. The penalty is 40 pts for each additional person. b) if an implementation of an algorithm is just a disguised version of the I-VT the grade for the project will be reduced to 0 pts.

\* These algorithms are described in the course book, Salvucii’s paper, and Tole’s paper that are uploaded on TRACS website

\*\* Papers explaining the Kalman Filter implementations are presented at http://www.cs.txstate.edu/~ok11/publications.html website. Also a separate lecture is given in class about the Kalman filter

GRADING CRITERIA

1. **SUBMISSION**
   1. Create a zip archive of **ALL** of your source files (not just the ones you have modified), your report file, and your presentation file. Name your archive *ET\_Project1\_classification [Name of the Teammates].zip.*  Upload the archive on the Assignment tab on TRACS. It is your responsibility to check that the file was submitted on TRACS.
   2. You have one day of late submission. If you submit late 10 points will be deducted from your project!

|  |  |  |
| --- | --- | --- |
| *#* | ***Description*** | ***Penalty (pts)*** |
| 1.1 | You did not submit the project 1, and your teammate(s) also DID NOT | Project receives 0 grade |
| 1.2 | You did not submit the project 1, but your teammate(s) DID | −10 |
| 1.3 | You did not submit files as ONE archive file (zip or rar) | −10 |
| 1.4 | One of your support files or more does not have proper format. The formats are allowed to be submitted are:  Report: DOC, DOCX, PDF.  Presentation slides: PPT, PPTX, PDF; | −10 |
| 1.5 | Submitted code is not in MATLAB, but in some other programming language. | Project receives 0 grade |
| 1.6 | One day of late submission | -10 |

1. **CHEATING POLICY**
   1. If a copy (from your classmate or from the Internet) is caught, all involved submissions (original as well as the copies) will be penalized. It is your responsibility to guard your work. Secure the read/write access of your directories. Any copy will result in ZERO grade for the assignment for both parties. The only exception is when you report the theft of your work in advance.
2. **CODE**

You MUST use the MATLAB code posted on TRACS together with the assignment. Select and implement the algorithm that you have selected in one of the functions with corresponding name. Uncomment this function from the function list presented in “main.m”. You can get maximum 7 points for the report.

When debugging and fine-tuning your algorithm you will need to select the thresholds in your algorithm that allow behavior scores (FQnS, SQlS, FQlS) as close as possible to their ideal values, otherwise your work will be penalized. Ideal behavior score values for the stimulus that was recorded for this assignment are FQnS=73.4%, FQlS=0°, and SQnS=100%.

|  |  |  |
| --- | --- | --- |
| *#* | *Description* | *Penalty, pts* |
| 3.1 | Following instructions provided in your report your code cannot be used to replicate the results that you claim in your report and the oral presentation (discussed below) | Project receives 0 grade |
| 3.2 | Your code cannot replicate the results exactly, but you can prove that mismatch is a mistake, and another set of good behavioral scores can be obtained by running your **already submitted** code | −10 |
| 3.3 | Submitted code does not compile and run | Project receives 0 grade |
| 3.4 | Submitted code is hidden I-VT but is presented on paper as some other algorithm. Only I-HMM can use I-VT as a PART of the algorithm | Project receives 0 grade |
| 3.5 | Unfinished code that cannot produce behavioral scores and the marked eye positional plots | Project receives 0 grade |
| 3.6 | Either eye positional plots or behavioral scores are missing from the output. | −50 |
| 3.7 | Classification data points are not in two dimensions | −10 |
| 3.8 | Your code does not have comments | −5 |
| 3.9 | If you get FQnS, SQnS with the difference more than 10% but less than 15% of their ideal values OR 2° <= FQlS < 3° | −5 |
| 3.10 | If you get FQnS, SQnS with the difference more than 15% but less than 25% of their ideal values OR 3° <= FQlS < 4° | −10 |
| 3.11 | If you get FQnS, SQnS with the difference more than 25% of their ideal values OR 3° <= FQlS < 4° | −50 |

1. **REPORT**

Write report file ET\_Project1\_Report\_[Name of the Teammates].doc (docx). You can get maximum 46 points for the report.

|  |  |  |
| --- | --- | --- |
| *#* | *Description* | *Penalty in points if the corresponding item is not provided* |
| 4.1 | Provide your name and names of your teammates. | −2 |
| 4.2 | Name of the algorithm selected for identification. | −2 |
| 4.3 | Identify what responsibilities each teammate had in this project, e.g., coding, documentation, debugging, etc. Indicate in a tableapproximate amount of time spent in planning, coding, debugging, and report writing in hours for each team member. Each category must be specified. If you were the only member of your team, you must still document the amount of time spent on each task. | −3 |
| 4.4 | Provide the pseudocode for your eye movement classification algorithm. Be specific enough that somebody would be able to implement your algorithm by following your pseudocode. | −10 |
| 4.5 | Provide a brief description of classification algorithm implementation challenges you have encounter during the project. Provide a brief description of how those challenges were solved. If shortcuts were used identify those shortcuts and the reasons why you have used them | −10 |
| 4.6 | Answer the question “Can the smooth pursuit of eye movement be identified by your algorithm?” If yes please describe how it can be potentially done? If not explain why it would not be possible to identify Smooth Pursuit of Eye Movements | −5 |
| 4.7 | Write down all threshold (parameter) values that you have employed in your algorithm | −6 |
| 4.8 | Provide qualitative and quantitative score values for fixations and saccades (SQnS, FQnS, and FQlS) | −6 |
| 4.9 | Provide all the references (scientific papers, websites) that you have read to accomplish the implementation of your algorithm | −2 |

1. **ORAL PRESENTATION**

Prepare an Oral presentation (not more than 10 minutes). One of the team members must be ready to present during class on the due date. You can get maximum 47 points for the presentation. If not ready to present in class at the presentation date 47 points will be deducted from the project.

|  |  |  |
| --- | --- | --- |
| *#* | *Description* | *Penalty if the point is NOT presented, pts* |
| 5.1 | Provide the name of the algorithm that you have selected for the presentation | −2 |
| 5.2 | Provide the pseudocode for your algorithm | −15 |
| 5.3 | Highlight (literally) the changes between your algorithm and the algorithm presented in the class lecture slides or class textbook or any other material source that you employed as a starting point for your algorithm | −5 |
| 5.4 | Discuss in a separate slide the ambiguities presented in the original pseudocode presented in the class lecture slides or class textbook or any other material source that you employed and how you solved these ambiguities in your implementation | −5 |
| 5.5 | Display in a separate slide specific threshold values that you have employed in your algorithm | −5 |
| 5.6 | Display in a separate slide behavioral scores (SQnS, FQnS, and FQlS) that you have obtained with your algorithm for the threshold specified | −5 |
| 5.7 | Discuss in a separate slide the implementation challenges that you have encountered | −5 |
| 5.8 | Discuss in a separate slide cases when you think your algorithm will fail. E.g. such cases might be represented by such things as small corrective saccades, blinks during fixations, beginning and the end of the saccades | −5 |

1. **EXTRA POINTS**
   1. If you present detailed analysis of your eye movement classification algorithm by analyzing several threshold values (at least 15 combinations/values) you will be able to get 3 extra points. A grader will assign the certain amount of points.
   2. One best projects will be selected based on a) difficulty of the algorithm, b) efficiency of the code, and c) the cleanness of the implementation. The best project will get 10 extra points.
2. **FAQ**

Q:

I was trying to figure out how to run your program in the MATLAB IDE. Is there any documentation on running your program or a tutorial? I have been doing some of my own research on it but I wanted to make sure there wasn't in documentation you can specifically provide on the IDE itself. Also, do I have to un-comment code in your program in order to run the default IVT algorithm example

A:

A tutorial on the MATLAB IDE can be found here <http://www.mathworks.com/videos/getting-started-with-matlab-68985.html?s_iid=ML2012_tutorials>

The MATLAB documentation is very useful for specific questions regarding syntax, functions, etc: http://www.mathworks.com/help/matlab/

Q:

How to run the default code that has I-VT implemented as an example?

A:

Open file runtime.m in MATLAB. After you open the script you can run it by pressing F5. You don't have to uncomment anything to run the default I-VT algorithm. To implement your algorithm please open a file with the name corresponding to your algorithm (e.g., EMD\_IKF.m) and insert your code there. In the file main.m change line [eye\_record,fixation\_filtered\_EMD,saccade\_filtered\_EMD,pursuit\_detected\_EMD] = EMD\_IVT(eye\_record);

to

[eye\_record,fixation\_filtered\_EMD,saccade\_filtered\_EMD,pursuit\_detected\_EMD] = EMD\_IKF(eye\_record);

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