Eye Tracking Data Analysis

Project in Eye Tracking 2014

Group C

Filip Povolny	25 %	Fixation detection algorithm
Sami Pietinen	25 %	MFD, MSA computation, plotting
Kritika Katwal	25 %	Data preprocessing
Keshav Gupta	25 %	Report

Objective

In this project raw eye tracking data has been analyzed. The dataset consists of 6 subjects where each subject had approximately 20 sessions. In each session subject had to look at an image and recognize it. Data from one session is the sequence of raw gaze locations x, y recorded with sampling frequency 1000 Hz.

Tasks

1. Raw data was loaded and preprocessed.
2. Modified algorithm from [1] was used for fixation and saccade detection. It is based on velocity and we tried 2 different settings by changing its parameters.
3. Mean fixation duration (MFD) and Mean Saccade Amplitude (MSA) were computed. MFD is the sum of all fixation points divided by the total number of fixations and it was calculated for each subject and for all the subjects together. Similarly, MSA is the sum of all the saccade amplitudes divided by the total number of saccades. Saccade amplitude is the angular distance that eye travels in one saccade. Standard deviation was computed for all the measurements like MFD, MSA categorized by known, unknown and overall images.
4. CSV file was created and named accordingly.
5. The charts were created using Matlab.

Data Preprocessing

1. The data provided was converted for easier working from .csv into .mat format and wanted subjects were fetched in function `get_subjects()`.
2. Raw data (x, y coordinates in pixels) were converted to angles in degrees (example in Figure *1*).

$$(x_{deg}, y_{deg}) = \left(\frac{x_{pix}}{97}, \frac{y_{pix}}{56}\right)$$

Fixation Detection Algorithm

This algorithm uses velocity-based approach of saccade and fixation detection with an adaptive threshold. Its implementation is in function `get_saccades()`.

1. The x, y coordinates are converted into instantaneous visual angles (example in Figure *2*).

$$iva = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

2. Signal was filtered to reduce noise with normalized filter kernel of different sizes and coefficients, similar to [2]:

1/9	2/9	3/9	2/9	1/9

Table 1: 5-tap normalized velocity filter

1/16	2/16	3/16	4/16	3/16	2/16	1/16

Table 2: 7-tap normalized velocity filter

The output was the velocity with removed noise (see Figure *3*).

3. Filtered signal was multiplied with sampling frequency to get velocity in deg/s. The maximum theoretical value is 1000 deg/s. All the values higher than this threshold, probably errors, were set to 1000 deg/s.
4. In order to find saccade peaks, saccade peak threshold was empirically estimated:
$$thres_{saccade\ peak} = \mu_{noise} + 3\sigma_{noise}$$
where $noise$ is all the velocity values with amplitude < noise amplitude, μ is the mean and σ is the standard deviation of those values. We tried 2 different noise amplitudes:
$$ampl_{noise} = 100 - 150\ °/s$$
All velocities > $thres_{saccade\ peak}$ were considered as saccade peaks (see Figure *4*).
5. For each saccade peak the signal was scanned backwards to find the saccade onset point. The saccade onset threshold was defined:
$$thres_{saccade\ onset} = \mu_{noise} + \sigma_{noise}$$
Then the saccade onset point is the velocity sample, whose value is lower than $thres_{saccade\ onset}$ and the value of the previous sample is greater.
6. Analogically the saccade offset using saccade offset threshold was found (example in Figure *5*).
$$thres_{saccade\ offset} = thres_{saccade\ onset}$$
7. Finally, the path points were classified either as fixations or saccades (example in Figure *6*).

Graph Plots

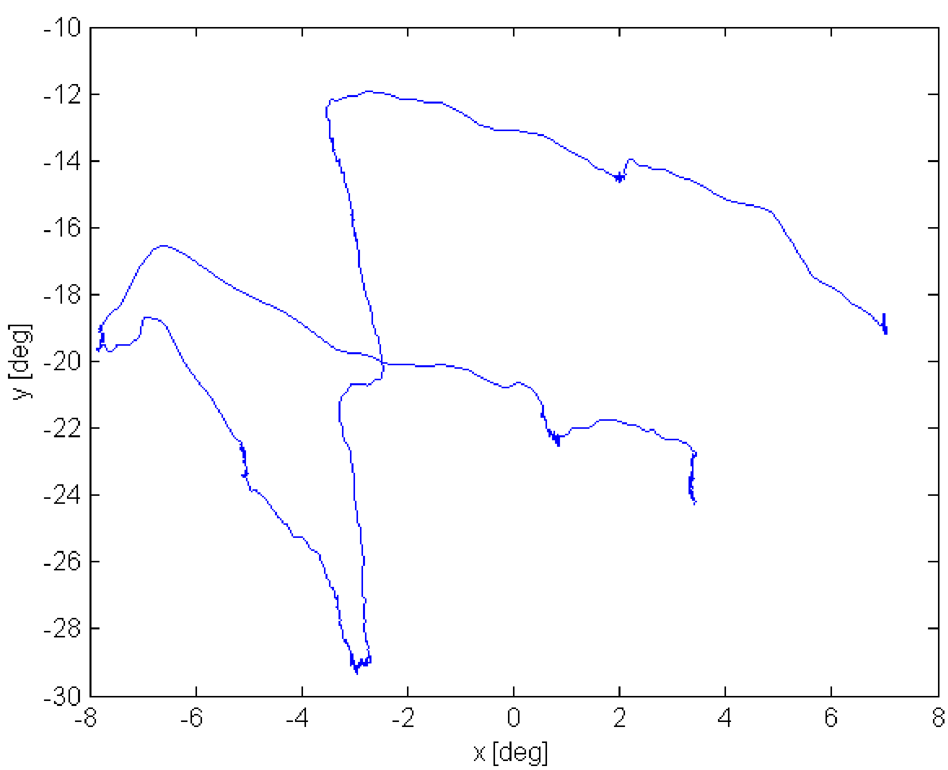

Figure 1: Gaze path on the image

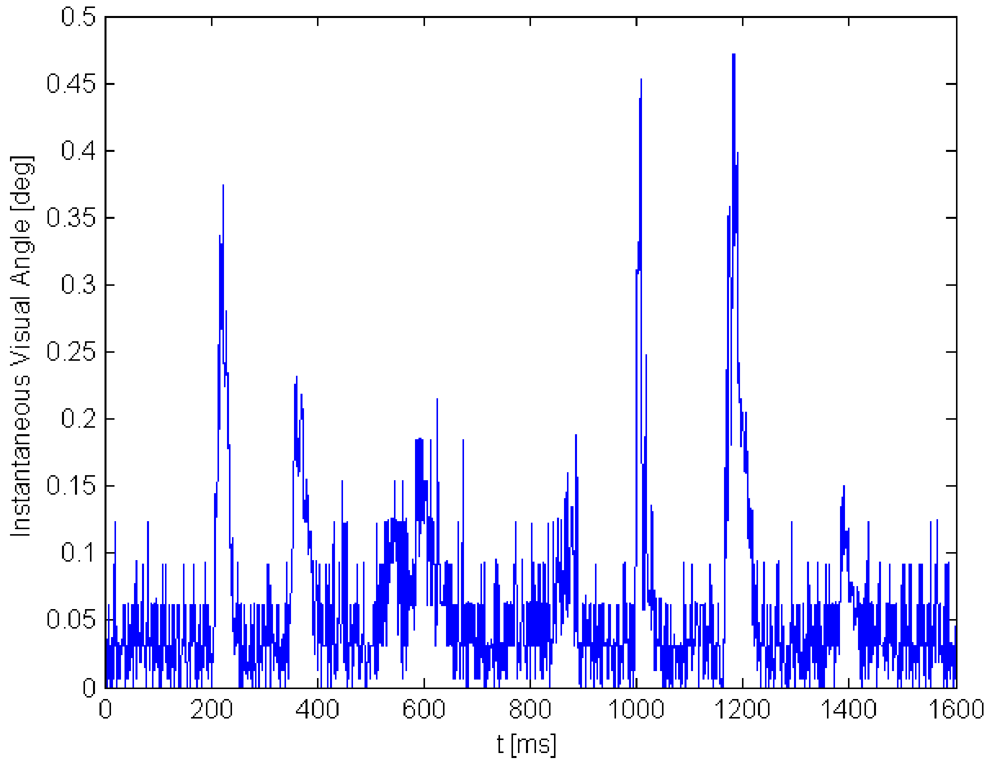

Figure 2: Instantaneous visual angles - angular distances between consecutive gaze points

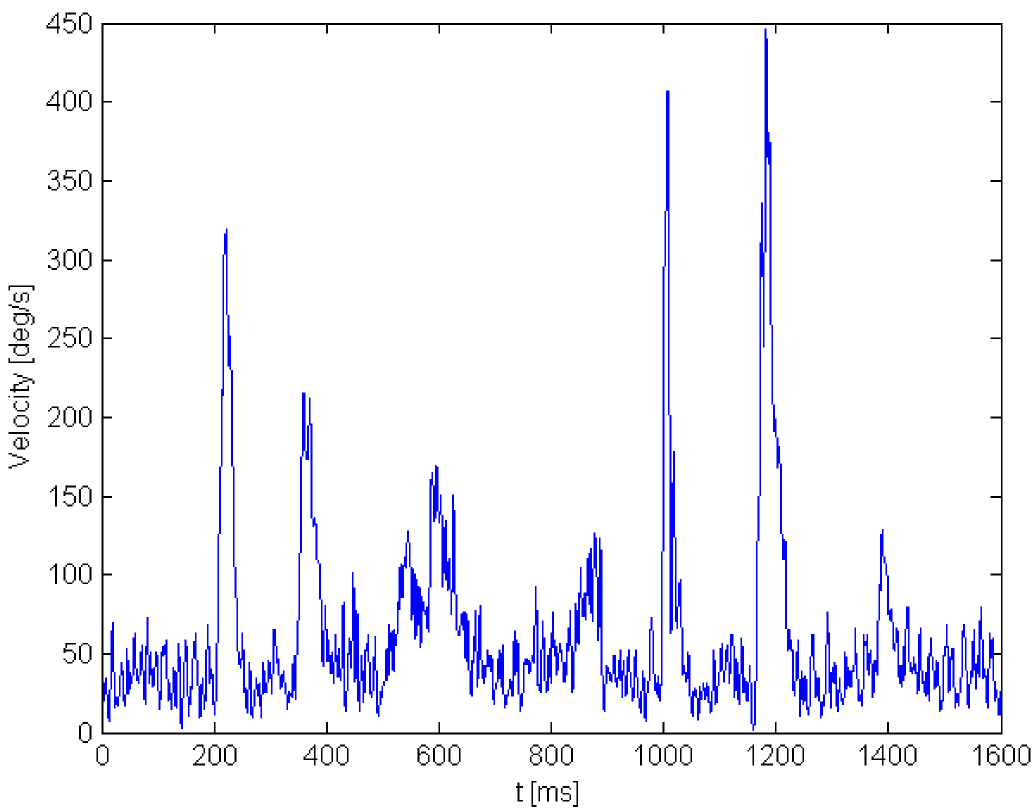

Figure 3: Velocity - instantaneous visual angles filtered with velocity filter

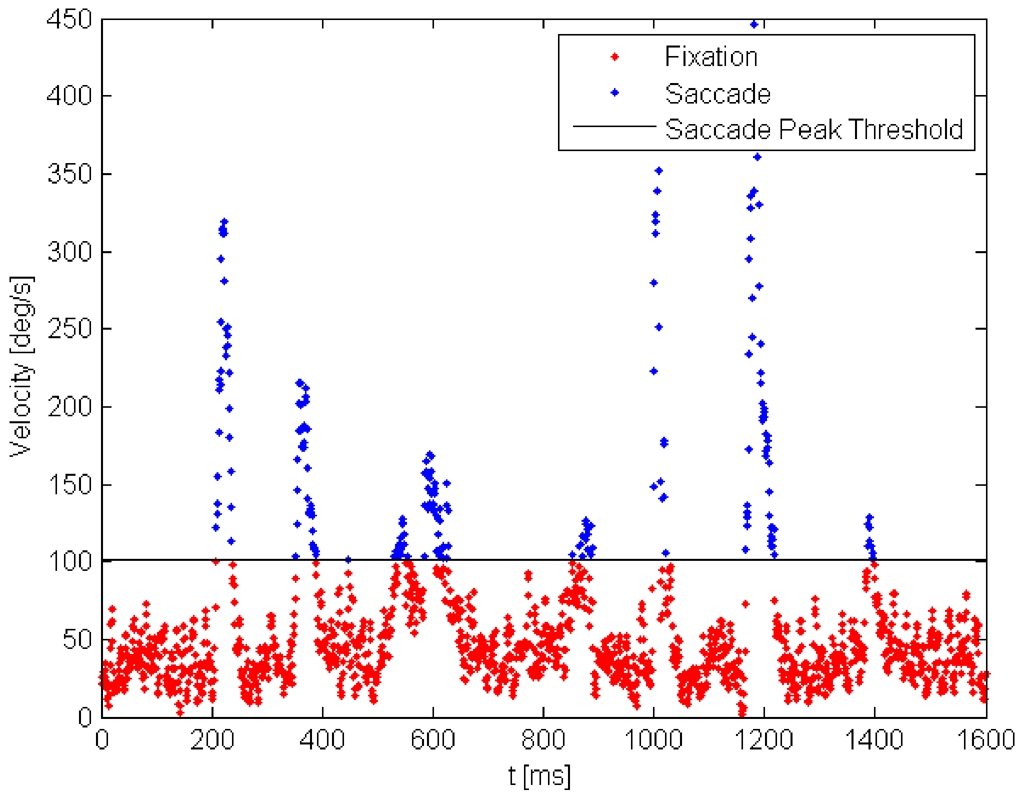

Figure 4: Blue points above saccade peak threshold represent saccade peaks.

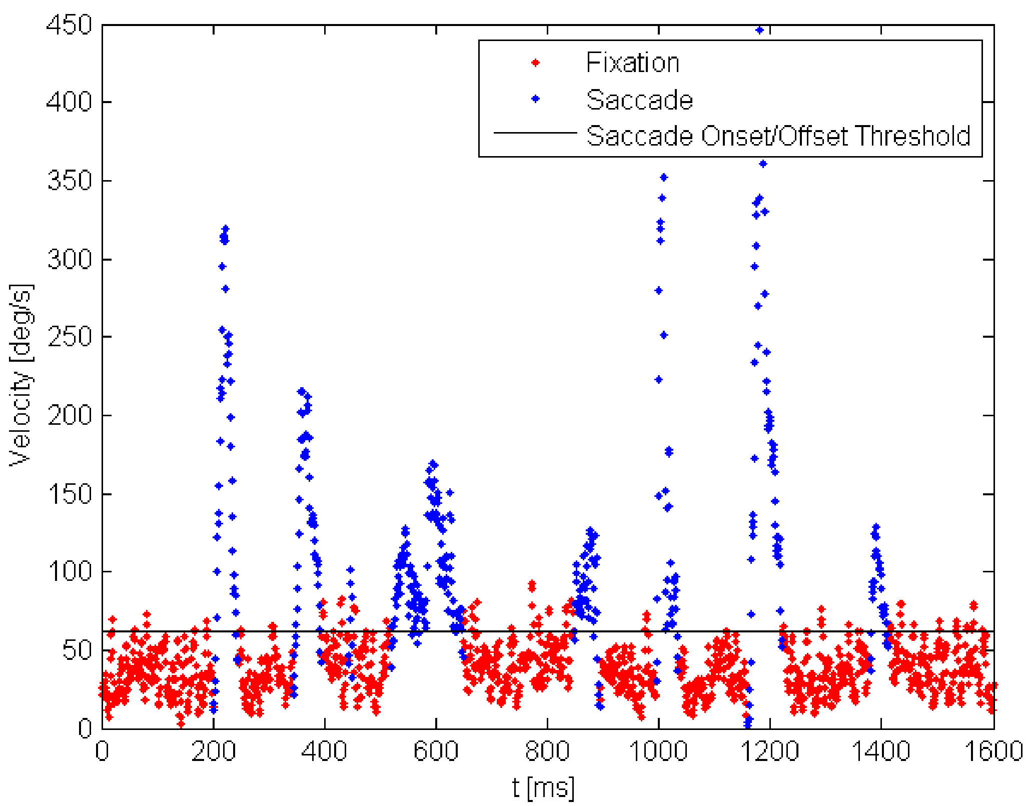

Figure 5: Whole saccades (blue) and fixations (red). Black line represents the saccade onset/offset threshold

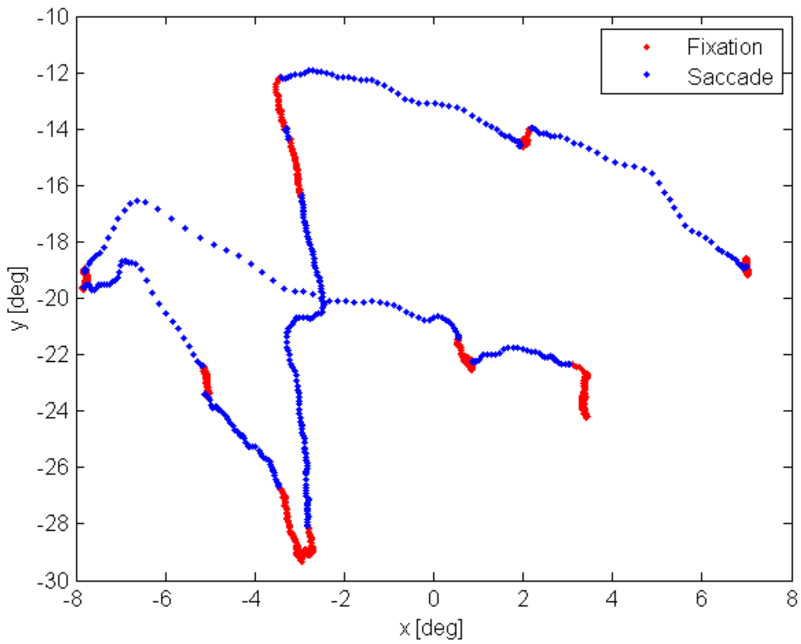

Figure 6: Gaze location points classified as fixation (red) or saccade (blue).

Results

See results for subjects s16 – s20 separately in Figure 7 and aggregated in Figure *8*. The results are computed in function `get_msd_msa()` and saved in .csv file in function `save_results()` in the format specified below and finally plotted in function `plot_results()`.

```
sid MFD_true MFD_SD_true MFD_false MFD_SD_false …
    MSA_true MSA_SD_true MSA_false MSA_SD_false …
    MFD_overall MFD_overall_SD MSA_overall MSA_overall_SD
```

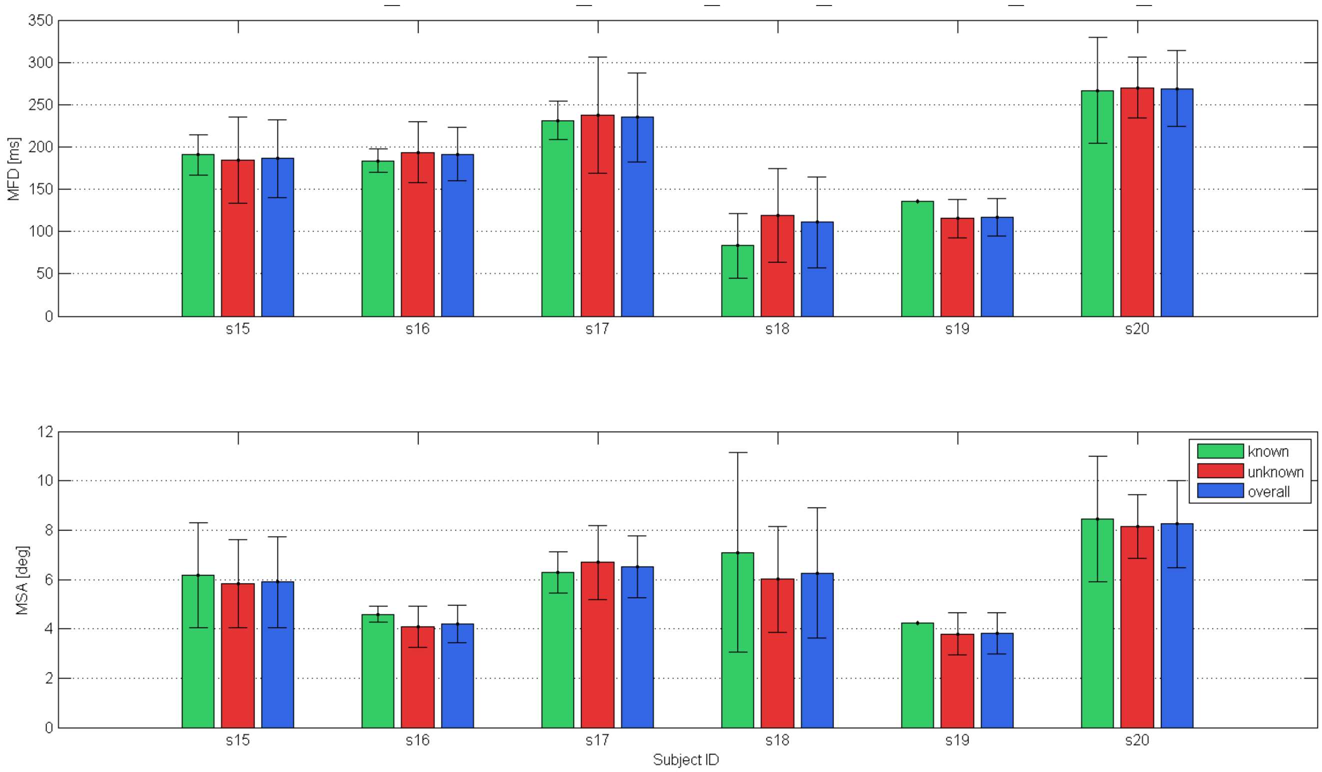

Figure 7: Mean Fixation Duration (upper) and Means Saccade Amplitude (lower) and the Standard Error of the Mean (Standard

Deviation) for different subject categorized for known, unknown and overall images.

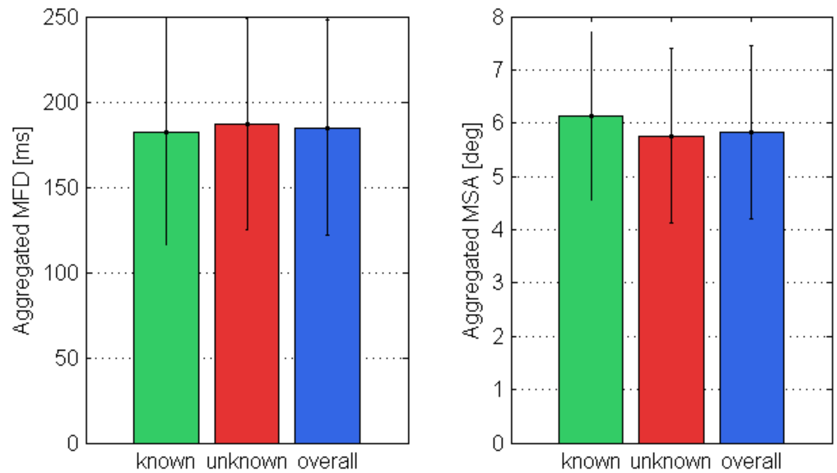

Figure 8: Aggregated MFD (left) and MSA (right), including the standard deviations, for all the subjects categorized for known, unknown and overall images.

Discussion

Looking at the above results it is evident that the difference between the MSA and MFD for known, unknown and overall is not significant enough to interpret any observation as yet.
The performance of the fixation detection algorithm could not be evaluated because the data provided was raw and lacked any additional information to make proper validations. The only way for us was to look at the graph plots and try to find the approximate areas of fixations.
This algorithm is not very well suited for short saccades and fixations because of the small difference in their velocities. Therefore, for noise amplitude was more suitable to choose $ampl_{noise} = 100\ °/s$ that rather splits fixations into more segments, but detects even short saccades. The size of the velocity filter affects the results insignificantly.

Usage

Run script `et.m` in Matlab command line.

References

[1] M. Nyström and K. Holmqvist, "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data," vol. 42, no. 1, pp. 188-204, 2010.

[2] A. T. Duchowski, Eye Tracking Methodology: Theory and Practice, vol. 2, London: Springer, 2007.