Eye Movement Classification Software user manual

Rev 1.5

6/12/2012

Created by Dr. Alex Karpov
Edited by Dr. Oleg Komogortsev
ok11@txstate.edu

Contents

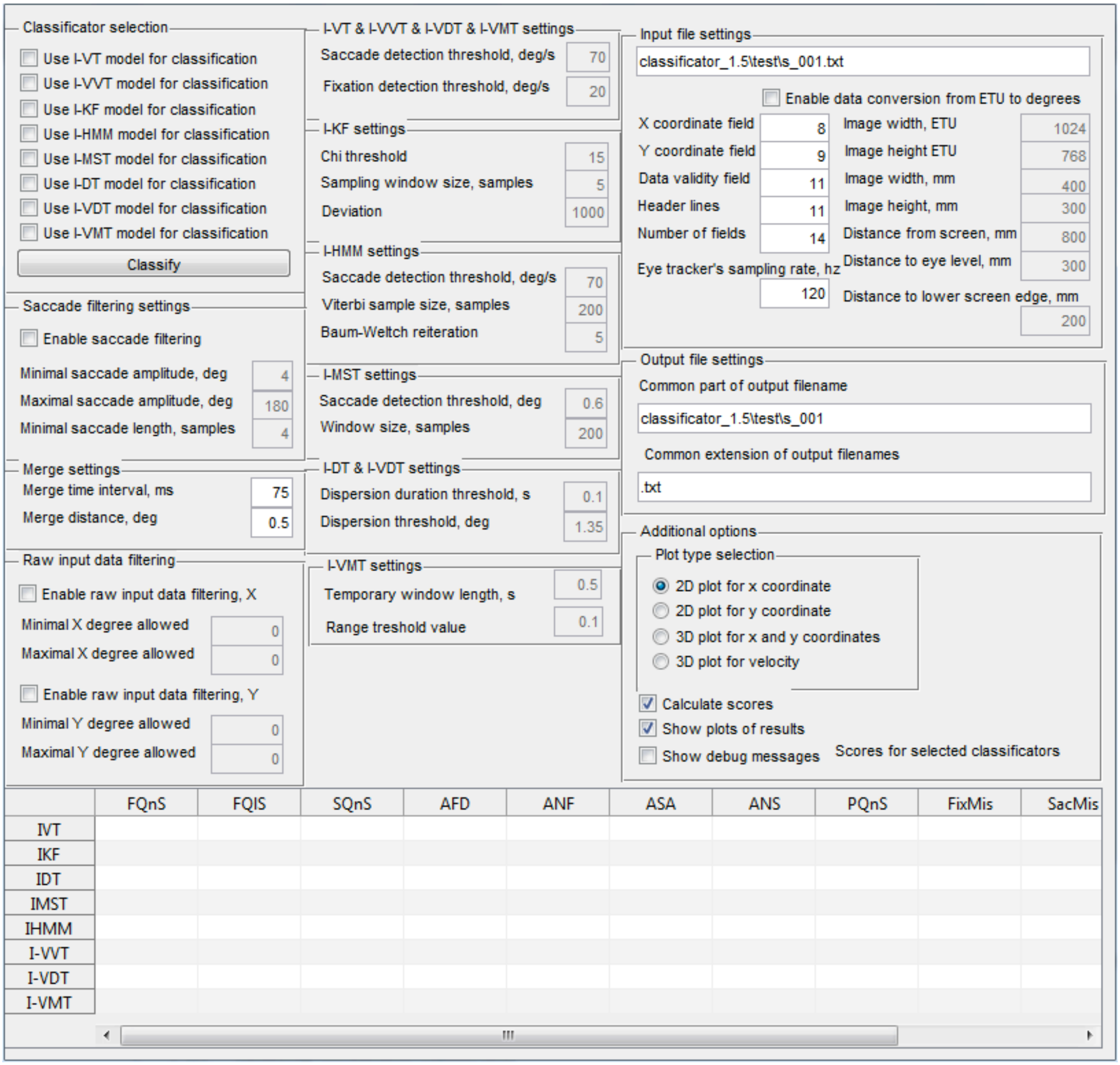
Eye movement’s classifiers user manual.

There are two way to use this software. First of them is to use GUI that available from classificator_run1_5. The second way is use API available from classificatory classes.

Note: runme.m is just an outdated example of API usage and shouldn’t be used directly.

GUI User manual.

After running classificator_run program user gets the following window:



Classificator selection

- □ Use I-VT model for classification
- □ Use I-VVT model for classification
- □ Use I-KF model for classification
- □ Use I-HMM model for classification
- □ Use I-MST model for classification
- □ Use I-DT model for classification
- □ Use I-VDT model for classification
- □ Use I-VMT model for classification

Saccade filtering settings

- □ Enable saccade filtering
- Minimal saccade amplitude, deg: 4
- Maximal saccade amplitude, deg: 180
- Minimal saccade length, samples: 4

Merge settings

- Merge time interval, ms: 75
- Merge distance, deg: 0.5

Raw input data filtering

- □ Enable raw input data filtering, X
- Minimal X degree allowed: 0
- Maximal X degree allowed: 0
- □ Enable raw input data filtering, Y
- Minimal Y degree allowed: 0
- Maximal Y degree allowed: 0

I-VT & I-VVT & I-VDT & I-VMT settings

- Saccade detection threshold, deg/s: 70
- Fixation detection threshold, deg/s: 20

I-KF settings

- Chi threshold: 15
- Sampling window size, samples: 5
- Deviation: 1000

I-HMM settings

- Saccade detection threshold, deg/s: 70
- Viterbi sample size, samples: 200
- Baum-Weltch reiteration: 5

I-MST settings

- Saccade detection threshold, deg: 0.6
- Window size, samples: 200

I-DT & I-VDT settings

- Dispersion duration threshold, s: 0.1
- Dispersion threshold, deg: 1.35

I-VMT settings

- Temporary window length, s: 0.5
- Range treshold value: 0.1

Input file settings

classificator_1.5\test\s_001.txt

- □ Enable data conversion from ETU to degrees
- X coordinate field: 8
- Y coordinate field: 9
- Data validity field: 11
- Header lines: 11
- Number of fields: 14
- Eye tracker's sampling rate, hz: 120
- Image width, ETU: 1024
- Image height ETU: 768
- Image width, mm: 400
- Image height, mm: 300
- Distance from screen, mm: 800
- Distance to eye level, mm: 300
- Distance to lower screen edge, mm: 200

Output file settings

- Common part of output filename: classificator_1.5\test\s_001
- Common extension of output filenames: .txt

Additional options

Plot type selection

- ● 2D plot for x coordinate
- ○ 2D plot for y coordinate
- ○ 3D plot for x and y coordinates
- ○ 3D plot for velocity

- ☑ Calculate scores
- ☑ Show plots of results
- □ Show debug messages

Scores for selected classificators

	FQnS	FQlS	SQnS	AFD	ANF	ASA	ANS	PQnS	FixMis	SacMis
IVT										
IKF										
IDT										
IMST										
IHMM										
I-VVT										
I-VDT										
I-VMT										

As you may see this window contains several panels.

First of them is 'Classificator selection'. Here you can choose which classifiers will be used in the processing of your data. At this moment five classifiers available:

- I-VT – velocity threshold classifier, classify saccades and fixations only;
- I-VVT – velocity threshold classifier, classify saccades, fixations, and smooth pursuits;
- I-KF – Kalman filter classifier, classify saccades and fixations only;
- I-HMM – Hidden Markov model classifier, classify saccades and fixations only;
- I-MST – minimum spanning tree classifier, classify saccades and fixations only;
- I-DT – dispersion threshold classifier, classify saccades and fixations only;
- I-VDT – velocity threshold classifier united with dispersion threshold classifier, classify saccades, fixations and smooth pursuits;
- I-VMP/I-VMT – velocity threshold classifier united with classifier based on San Agustin idea of classification, capable to detect saccades, fixations, and smooth pursuits.

Note that brief description of each algorithm is provided in [1], the pseudocode for each algorithm is presented in [2].

After each selection the appropriate tabs settings for the chosen classifier became active. For example after selecting I-VT classifier the corresponding 'I-VT settings' tab became active.

Each of this classifier setting tabs contains settings specific to a classifier.

I-VT settings:

- 'Saccade detection threshold' field set the value of speed threshold value that used for differentiate between saccades and fixations. If the speed of movement from one eye position to next eye position is less than this value it means that this eye movement belongs to fixation. Elsewise it is a saccade.
- 'Fixation detection threshold' field in disabled for this classifier.

I-VVT settings:

- 'Saccade detection threshold' field set the value of speed threshold value that used for differentiate between saccades and fixations. If the speed of movement from one eye position to next eye position is greater than this value it means that this eye movement belongs to saccade. Elsewise it is a fixation or smooth pursuit and should be check by other threshold.
- 'Fixation detection threshold' field set the value of speed threshold value that used for differentiate between fixations and smooth pursuits (saccades were filtered prior that stage of

classification). If the speed of movement from one eye position to next eye position is greater than this value it means that this eye movement belongs to smooth pursuit. Elsewise it is a fixation.

I-KF settings:

- ‘Chi threshold’ field set the threshold value for χ^2-distribution. If χ^2-distributions value is less than this threshold it means that we detected a fixation. Elsewise we deal with saccade.
- ‘Sampling window size’ field set the number of samples for which we calculate χ^2-distribution.
- ‘Deviation’ field set the deviation value between predicted and measured values for χ^2-distribution calculation.

I-HMM settings:

- ‘Saccade detection threshold’ field have the same meaning as in I-VT classifier.
- ‘Viterbi sample size’ field set the number of samples that classifier use as one data set. If set too high then is not enough machine accuracy to calculate statistical parameters.
- ‘Baum-Welch reiteration’ field set the number of iterations of Baum-Welch algorithm. It has sense to set this value equal 4 or 5.

I-MST settings:

- ‘Saccade detection threshold’ field set the distance in degrees between two adjacent eye focus positions. If this distance is less than the thresholds value it mean that this eye movement belongs to fixation. Elsewise it is a saccade.
- ‘Window size’ field set up the number of samples that classifier use during data processing. If set too high it can significantly increase time of detection for saccades. It has sense to set this value a little lesser than average fixation duration.

I-DT settings:

- ‘Dispersion duration threshold’ field determine how much time we should use for dispersion calculation during data processing.
- ‘Dispersion threshold’ field set the threshold value for dispersion of selected samples in degrees. If dispersion is less than that value it means that we detect the fixation. Elsewise we have a deal with saccade.

I-VDT settings:

- ‘Saccade detection threshold’ field set the value of speed threshold value that used for differentiate between saccades and fixations. If the speed of movement from one eye position to next eye position is greater than this value it means that this eye movement belongs to saccade. Elsewise it is a fixation or smooth pursuit and should be check at next stage of classification.

- ‘Dispersion duration threshold’ field in I-DT tab determine how much time we should use for dispersion calculation during data processing.
- ‘Dispersion threshold’ field in I-DT tab set the threshold value for dispersion of selected samples in degrees. If dispersion is less than that value it means that we detect the fixation. Elsewise we have a deal with smooth pursuit.

I-VMP settings:

- ‘Saccade detection threshold’ field in the I-VT tab set the value of speed threshold value that used for differentiate between saccades and fixations. If the speed of movement from one eye position to next eye position is greater than this value it means that this eye movement belongs to saccade. Elsewise it is a fixation or smooth pursuit and should be check at next stage of classification.
- ‘Temporary window length, s’ field determine how much time we should use for dispersion calculation during data processing.
- ‘Range threshold value’ field determine set the threshold value for dispersion of selected samples in range between 0 and 1.

Default parameter values for each classification algorithm are provided as suggested in [1]

Below ‘Classificator selection’ panel situated ‘Saccade filtration settings’ tab. On this tab user may set up settings for filter parasitic saccades. Such saccades may occur due to equipment failure or incorrect settings of the classifier. Settings on this panel become available after activation the ‘Enable saccade filtration’ check box. This panel has three fields:

- ‘Minimal saccade amplitude’ field set the minimal allowed amplitude for saccade. All saccades with amplitude lesser then this value will be discarded. Such saccades may occur when saccade threshold value was too low and some fixations were classified by classifier as saccade with low amplitude.
- ‘Maximal saccade amplitude’ field set the maximal allowed amplitude for saccade. All saccades with amplitude greater than this value will be discarded. Such saccades may occur after eye tracker failure. For example, during blink eye tracker may return 90^{o} as eye position and marked as valid data reading. This may lead to saccade with very large amplitude.
- ‘Minimal saccade length’ field set the minimal allowed length for saccade in samples. It helps for filter out too short saccades. The reason for occur such saccades usually the same as above.

Below ‘Saccade filtration settings’ tab situated ‘Merge settings’ panel. On this panel user may change merge function settings namely:

- ‘Merge time interval’ field setup a maximal time interval between two fixations that allow merging them.
- ‘Merge distance’ field setup a maximal distance between two fixations that allow merging them.

Default parameter values for each classification algorithm are provided as suggested in [1]

Below ‘Merge settings’ tab situated ‘Raw input data filtering’ panel. On this panel user can enable input data filtration based on the range of allowed degree for X and Y axis separately. The settings for range values become available after activation corresponding checkboxes. If this filter is active all data that ont in the allowed range will be marked as noise.

‘Input file settings’ allow user to set up parameters regarding input file. Text field at the top of the panel contains full input file name. Input file have to be structured as described in data structure section of this manual. To successfully read the data from the input file user have to set up five fields:

- ‘X coordinate’ field setup the number of column where x coordinate of eye position is situated;
- ‘Y coordinate’ field setup the number of column where y coordinate of eye position is situated;
- ‘Data validity’ field setup the number of column where data validity mark is situated. Input file may not have such field. If this is the case then user have enter 0 to this field.
- ‘Header lines’ field setup the number of lines in the input file header. Such lines just skipped by reader. If actual eye tracker data go from the first line, user should enter 0 into this field.
- ‘Number of fields’ field setup the number of fields in the line with data of input file.
- ‘Eye tracker’s sampling rate’ field setup the sample rate for input file. This value show how much samples was read from eye tracker per second.

In discussed panel situated ‘Enable data conversion from ETU to degrees’ check box. After enabling this option user can provide input data with ETU coordinates. Classifiers have to converse all input data from ETU to degrees. But this mode is not recommended. If user decides to use this mode he have to enable this checkbox. After that several fields become available:

- ‘Image width, ETU ‘ field setup the width of the eye tracker virtual image in the ETU;
- ‘Image height, ETU’ field setup the height of the eye tracker virtual image in the ETU;
- ‘Image width, mm’ field setup the width of the eye tracker virtual image in the mm;
- ‘Image height, mm’ field setup the height of the eye tracker virtual image in the mm;

- ‘Distance from screen’ field setup the distance between the screen and the subject;
- ‘Chin rest height’ field setup the height of the chin rest that used during the data collection;
- ‘Chin rest position’ field setup the distance between the eye tracker and the chi rest.

‘Output file settings’ panel allow user to set up output filenames. This panel contains two fields:

- ‘Common part of output filename’ field contain the common part for all output files;
- ‘Common extension of output filename’ field contain the common extension for all output files;

Output filenames are created from these values as follows. Common part of output filename combined with unique part and with the common extension of output filename. Unique part of output filename is different for every classifier. For example for I-VT classifier such unique parts will be _ivt_saccade_list for saccades; _ivt_noise_list for noise records; _ivt_pursuits_list for smooth pursuits; and _ivt_fixation_list for fixations. For other classifiers unique parts will be changed accordingly.

Below this tab ‘Show debug messages’ checkbox is placed. After switching on this control user will get some informational messages on the MatLab console.

‘Additional options’ tab contains two checkboxes that allow to user make some general result processing. One of this checkboxes named ‘Calculate scores’ enable scores calculations. These scores will be displayed at the table in right bottom of the window. For unused classifiers scores will be containing ‘N/A’. Another checkbox on this panel is a ‘Show plots of results’ checkbox. After enable user get the possibility to make plots of results of four kinds. These kinds are:

- 2D plot for x coordinate – user get a 2D plot of x coordinate for eye tracker data and classified results for every selected classifier.
- 2D plot for y coordinate – user get a 2D plot of y coordinate for eye tracker data and classified results for ever selected classifier
- 3D plot for x and y coordinates – user get a plot of eye tracker data and classified results in 3D for every selected classifiers.
- 3D plot for velocity – user get the same plots as above with velocity added for every selected classifiers.

Be aware that for using these settings classifiers have to get an access to stimulus data used during the eye recording session. Currently stimulus used in our experiments is hardcoded.

The legend of these plots mean as follows:

- ‘stimulus’ is a readings of artificial eye stimulus that was used during experiment session for subjects;

- ‘eye tracker’ is a raw data readings from eye tracker.
- ‘noise’ includes several different eye position types. First of them is consist from all eye positions that was marked by eye tracker as noise during the session. For this category x and y coordinates were reset to 0. Second category includes data from saccades that were discarded during saccades filtration. Be aware that filtration of micro saccades with amplitude less than 0.5 degrees is hardcoded. And the last category of noise consists from eye positions initially marked as fixation but discarded due their short duration (less than 100 ms.). This filtration is also hardcoded.
- ‘saccade’ consists of all eye positions classified as saccade by classifier.
- ‘fixation’ consists of all eye records classified as fixation by classifier.
- ‘centroid’ includes all centroids (coordinates of eye fixation) of fixations.
- ‘pursuit’ includes all eye positions classified as smooth pursuit by classifier (if any).

Due to filtration eye trackers data is not necessary have to be the same as set of noise, saccades, and fixations points. For example in some cases eye trackers position trajectory and fixation trajectory may differ due to merge.

Tab scores for selected classificatory contains behavior scores as defined in [1]. The scores can be employed for meaningful classification threshold selection or identify cases when classification by a specific algorithm fails. The computation of the behavior scores is tied to the stimulus described in [1], if different type of stimulus is employed **scores_computation_class** must be modified to reflect the changes in the stimulus.

Data Format.

Input data file format is as follows. It has to be an ASCII text file with several fields delimited by spaces or semicolons ‘:’. Every field has to be a number.

Output data file format is case specific. The headers in the output files are self-explanatory.

API User Manual.

Classifiers consist of several classes. Each of these classes provides several methods and properties. The examples of usage of these classes may be seen in SCORES.m and runme.m files. Generally classifiers implemented in these classes:

- **classificator_IVT_class** is implementation of I-VT classifier;
- **classificator_IDT_class** is implementation of I-DT classifier;
- **classificator_IKF_class** is implementation of I-KF classifier;

- **classificator_IHMM_class** is implementation of I-HMM classifier;
- **classificator_IMST_class** is implementation of I-MST classifier;
- **classificator_IVVT_class** is implementation of I-VVT classifier;
- **classificatory_IVDT_class** is implementation of I-VDT classifier;
- **classificatory_IVMT_class** is implementation of I-VMT/I-VMP classifier;
- **scores_computation_class** is implementation of class for draw plots and scores computational.

Here are the common methods and properties for all classifiers:

- **input_data_name** property contains full input data filename for reading;
- **x_field** property contains the number of x coordinate field in the input file;
- **y_field** property contains the number of y coordinate field in the input file;
- **v_field** property contains the number of data validation field in the input file;
- **header_count** property contains the number of lines in the header of the input file;
- **fields_count** property contains the number of data fields in the lines of input file;
- **debug_mode** property contains the Boolean switch to enable output of additional information to MatLab console;
- **delta_t_sec** property contains the time step between to eye position measurement (this property is backward to **sample_rate** property below);
- **sample_rate** property contains the sample rate of the eye tracker (this property is backward to **delta_t_sec** property above);
- **image_width_mm** property contains image width in mm for conversion from ETU to degree and backward;
- **image_height_mm** property contains image height in mm for conversion from ETU to degree and backward;
- **image_width_etu** property contains image width in ETU for conversion from ETU to degree and backward;
- **image_height_etu** property contains image height in ETU for conversion from ETU to degree and backward;

- **distance_from_screen** property contains distance in mm between screen and the subject of eye position readings;
- **chin_rest_height** property contains the height of the chin rest that used during the experiment session;
- **chin_rest_position** property contains the position of chin rest during the experiment session;
- **eye_records** property contains the list of eye tracker data records;
- **saccade_records** property contains the list of detected saccades;
- **fixation_records** property contains the list of detected fixations;
- **noise_records** property contains the list of noise data;
- **minimal_saccade_amplitude** property contains the minimal allowed amplitude for saccade;
- **maximal_sacade_amplitude** property contains the maximal allowed amplitude for saccade;
- **minimal_saccade_length** property contains the minimal allowed saccade length;
- **use_degree_data_filtering_X** property enable or disable usage of input data filtering based on range of allowed angles for X axis;
- **use_degree_data_filtering_Y** property enable or disable usage of input data filtering based on range of allowed angles for Y axis;
- **minimal_allowed_X_degree** property contains the minimal allowed angle in degrees for eye boll in X axis;
- **maximal_allowed_X_degree** property contains the maximal allowed angle in degrees for eye boll in X axis;
- **minimal_allowed_Y_degree** property contains the minimal allowed angle in degrees for eye boll in Y axis;
- **maximal_allowed_Y_degree** property contains the maximal allowed angle in degrees for eye boll in Y axis;
- **merge_fixation_time_interval** property contains the maximal time interval between two fixations that allow to merging them into the one fixation;
- **merge_fixation_distance** property contains the maximal distance between centroids of two fixations that allow to merging them into the one fixation;
- **unfiltered_saccade_records** property contains the list of unfiltered saccades;

- **filtered_saccade_records** property contains the list of filtered saccades;
- **basename_output_filename** property contains common part of output filenames;
- **basename_output_extension** property contains common extension for output files;
- **error_code** property contains an error code for I/O errors that tale a place during input file reading;
- **error_message** property contains a verbose description for **error_code**;

And here are the unique properties for every classifier:

- **saccade_detection_threshold** property (I-VT);
- **chi_threshold**, **window_size,** and **deviation** properties (I-KF);
- **dispersion_duration_sec_threshold,** and **dispersion_threshold** properties (I-DT);
- **saccade_detection_threshold,** and **window_size** properties (I-MST);
- **saccade_detection_threshold, viterbi_sample_size,** and **baum_welch_reiterations_count** properties (I-HMM);
- **saccade_detection_threshold, fixation_detection_threshold** property (I-VVT);
- **saccade_detection_threshold, dispersion_duration_sec_threshold, idt_dispersion_threshold** properties I-VDT;
- **saccade_detection_threshold, window_duration_sec_threshold, san_agustin_threshold** properties (I-VMP/I-VMT).

Common methods for all classifiers include:

- **read_data()** method read data from the input file with provided settings;
- **convert _from_ETU_to_degree()** method converts eye tracker data from ETU to degree (this method is backward to method below);
- **convert_from_degree_to_ETU()** method converts eye tracker data from degree to ETU (this method is backward to method above);
- **classify()** method is perform an eye data tracker data classification. In order to proper filtering of unnecessary data by degree this method should be in-between two executions of **eye_tracker_data_filter_degree_range** methods;
- **merge_records()** method is split classified eye tracker data into the several lists according data classification;

- **saccade_filtering()** method execute a saccade filtration according provided amplitude thresholds and saccade length.

- **eye_tracker_data_filter_degree_range()** method execute a data filtration according provided ranges of acceptable angles. Any data with angle out of the provided range will be mark as noise if filtration is enabled for corresponding axis. In order to properly filter unnecessary data this method should be called twice. First time rights before **classify()** method and second time immediately after.

- **setup_output_names()** method setup all output filenames according common parts;

- **write_datafiles()** method actually output classified lists of records.

For auxiliary data processing and output was implemented **scores_computation_class**. It has properties and methods as follows:

- **eye_records, saccade_records, fixation_records, noise_records, sample_rate, delta_t_sec** properties generally have the same meaning as the same properties of classifiers;

- **generate_stimulus_CS_TR_2009_16()** method is a hardcoded stimulus;

- **draw_graphics(mode,name)** method draw the plot named by **name**. **Mode** may be equal to 1,2,3,4. The description of these modes sees in GUI User Manual.

- **FQnS, FQlS, SQnS, AFD, ANF, ASA, ANS, PQnS, PQlS_P, PQlS_V, FPMS,** and **SPMS** properties contains the behavior scores computed as a result of classification.

The lists of records have the following structure:

- **eye_records** is a two dimensional array. First and second fields contains x and y coordinates respectively, third field contain velocity in this sample, fourth field hold classification, fifth is for validity mark and the sixth field for time mark (sample number);

- **saccade_records, fixation_records, noise_records, pursuit_records, filtered_saccade_records, unfiltered_saccade_records** have the same structure. They are cell array each cell of which contains the two dimensional arrays. First and the second fields of this array contains x and y coordinates respectively. The third field contains a sample number (time mark).

- **stimulus_records** contains stimulus data. This is a two dimensional array, first and second fields of which contains x and y coordinates respectively, the third field is a time mark (sample number) and the forth one is a record type.

Input Data Included with the Distribution

Three subjects' files out of the recording specified in [1] are provided in the distribution.

Changes from release 1.1.

- Added three new classifiers I-VMP, I-VDT, and I-VVT with their properties;

Changes from release 1.0.

- Bugs with empty lists of saccades, fixations and noises are fixed;
- New field delimiter in input file added – ‘:’;
- New filter added – allowed ranges of angles for horizontal and vertical axis.

Changes from previous distribution.

- Bug fixes in I-MST and I-HMM implementations;
- Speed improvements in I-MST implementation;
- Code clarification and redesign;
- GUI interface.

TODO list.

- Code optimization and speed improvements;
- Parallelization;
- GUI improvements;
- Convenient work with files and directories;
- Individual manual saccade editor;
- User stimulus routines;
- Implement a selecting between three modes: 1D with X axis, 1D with Y axis, and 2D mode (using in current and previous versions) for improving the quality of classification.

References

[1] O. V. Komogortsev*, et al.*, "Standardization of Automated Analyses of Oculomotor Fixation and Saccadic Behaviors," *Biomedical Engineering, IEEE Transactions on,* vol. 57, pp. 2635-2645, 2010.

[2] O. V. Komogortsev*, et al.*, "Qualitative and Quantitative Scoring and Evaluation of the Eye Movement Classification Algorithms," Technical Report. Texas State University - San Marcos, http://ecommons.txstate.edu/, San Marcos, 2009.