UBC’s

Eye Movement Data Analysis Toolkit (EMDAT)

User Manual

Version 0.2

Samad Kardan

Acknowledgement

Development of EMDAT has been done under the supervision of Dr. Cristina Conati.

EMDAT is partially based on Python Eye Tracking Library (PTEL) developed by Nicholas FitzGerald.

# Introduction

This manual describes how to use EMDAT library for analyzing eye tracking data collected by a Tobii eye tracker. It is important for the reader to be familiar with eye tracking concepts and Tobii Studio software before using EMDAT.

## Basic Concepts

An eye-tracker provides eye-gaze information in terms of *fixations* (i.e., maintaining eye-gaze at one point on the screen) and *saccades* (i.e., a quick movement of gaze from one fixation point to another), which are analyzed to derive a viewer’s attention patterns. EMDAT uses a large set of basic eye-tracking features, described by as the building blocks for comprehensive eye-data processing. These features are built by calculating a variety of statistics upon the basic eye-tracking measures described in .

**Table .** Description of basic eye tracking measures

|  |  |
| --- | --- |
| Measure | Description |
| Fixation rate | Rate of eye fixations per milliseconds |
| Number of Fixations | Number of eye fixations detected during an interval of interest |
| Fixation Duration | Time duration of an individual fixation |
| Saccade Length | Distance between the two fixations delimiting the saccade (d in ) |
| Relative Saccade Angles | The angle between the two consecutive saccades (e.g., angle y in ) |
| Absolute Saccade Angles | The angle between a saccade and the horizontal (e.g., angle x in ) |

Of these measures, *Fixation rate*, *Number of Fixations* and *Fixation Duration* are widely used (e.g., [11, 15–17]); we also included *Saccade Length* (e.g., distance *d* in ), *Relative Saccades Angle* (e.g., angle *y* in ) and *Absolute Saccade Angle* (e.g., angle *x* in ), as suggested in , because these measures are useful to summarize trends in user attention patterns within a specific interaction window (e.g., if the user’s gaze seems to follow a planned sequence as opposed to being scattered). Statistics such as sum, average and standard deviation can be calculated over these measures with respect to: (*i*) the full experiment window, to get a sense of a user’s overall attention; (*ii*) specific areas of interest (AOI from now on) identifying parts of the interface that are of specific relevance for understanding a user’s attention processes.

**Fig. .** Saccade based eye measures

# Input Files for using EMDAT

## Files exported from Tobii Studio

There four types of files that are used by EMDAT which can be directly exported from Tobii studio software.

All-Data.tsv: This file contains all the data recorded by the eye-tracker. The eye-tracker samples at a constant rate (e.g., every 8 milliseconds for a Tobii T120 sampling at 120Hz), and records values for a wide variety of features including gaze location, pupil dilation etc.

Fixation-Data.tsv: The individual points of gaze-location data output in the All-Data file can be aggregated into "Fixations" according a clustering algorithm. The algorithm groups gaze samples which are close enough in distance and time into Fixations, which have a location (centre of the corresponding gaze samples), and a duration.

**Note:** When exporting the data, the Tobii software allows you to choose one of the available methods for clustering Fixations (varies in different versions of Tobii studio). Make sure you use the same method for every subject in a given experiment, to ensure consistency.

Event-Data.tsv: This file includes all the non-gaze events which occur, such as mouse-clicks and key-presses. It also includes all the user defined events that are added post experiment using Tobii studio. This is important for aligning the experimental timing data with the eye-tracking.

‘.aoi’ file: This file contains the definition of the Areas of Interest that were defined and exported using the AOI tool in Tobii studio. The AOIs are defined as a single polygon. EMDAT also supports dynamic AOIs which are active at certain time intervals. For defining dynamic AOIs you need to use an extended version of the ‘.aoi’ file as defined in the next section.

## Additional files

‘.aoi’ file: EMDAT supports dynamic AOIs which are active at certain time intervals. For defining dynamic AOIs you need to use the extended version of the ‘.aoi’ file. The extended version of '.aoi' files has pairs of lines for each AOI of the form:

aoiname<tab>point1x,point1y<tab>point2x,point2y<tab>...<newline>

#<tab>start1,end1<tab>...<newline>

Where in the first line, *aoiname* is the name of the AOI, *point***i***x* and *point***i***y* define the x and y coordinates of the each vertex in the polygon that defines boundaries of the AOI. In the second line, *start***j** and *end***j** define the start and end of each time interval that this AOI is active in milliseconds. The second line starts with ’#' and is optional. If the second line does not exist the AOI will be active throughout the whole session (a.k.a. a global AOI).

‘.seg’ file:

The '.seg' files have lines of the form:

scene\_name<tab>segment\_name<tab>start\_time<tab>end\_time<newline>

Where *scene\_name* is the id of the Scene that this Segment belongs to, *segment\_name* is the id of the Segement, and *start\_time* and *end\_time* determines the time interval for the Segment.

# EMDAT Core files

All of the EMDAT files have been documented according to Google’s documentation style for Python[[1]](#footnote-1).

## Segment Class

A [Segment](segment.html) is a class that represents the smallest unit of aggregated eye data samples with a conceptual meaning related to the experiment attached to it by the experimenter (e.g., performing a sub task, the interval in which user was looking at the screen uninterrupted, etc.). This class is the equivalent  
of segments as defined in Tobii studio.

## Scene Class

A [Scene](Scene.html#Scene) is a class that represents one scene (e.g., when certain visual elements are present at the screen) in the experiment. The Scene is designed to aggregate [Segment](file:///C:\Research-Samad\EMDAT\doc\segment.html)s related to a target conceptual entity or activity in the experiment. A Scene should have at least one [Segment](file:///C:\Research-Samad\EMDAT\doc\segment.html) assigned to it. From the technical point of view, the [Scene](file:///C:\Research-Samad\EMDAT\doc\scene.html) class is used to combine multiple [Segment](file:///C:\Research-Samad\EMDAT\doc\segment.html)s and calculate the aggregated features for this new entity as a whole. This class is the equivalent of a scene as defined in Tobii studio.

While Scene and Segment classes have many similarities (e.g., they share a lot of features), you should note that they have different intended purposes. The Segment class covers one basic continues interval of eye gaze samples and calculates the features for that interval. The Scene class enables the researcher to look at the eye gaze data in a higher level of abstraction and get away from the raw data by covering a set of Segments. For example, if a researcher is interested in see how participants react to a specific visual element (e.g., a virtual agent) that appears under certain conditions on the screen, each instance of its appearance will be covered by a Segment and the overall picture is captured by a Scene that includes all the relevant Segments. As another example, if two activities of interest are happening having two separate Scenes with their respective Segments enables looking at these two activities separately even if in some cases they overlap.

## AOI and AOI\_Stat classes

In EMDAT, the boundary of an Area of Interest ([AOI](AOI.html)) is defined as a polygon on the screen. You can optionally define a second polygon inside the first polygon to be excluded from the [AOI](file:///C:\Research-Samad\EMDAT\doc\AOI.html). An [AOI](file:///C:\Research-Samad\EMDAT\doc\AOI.html) can be always active (a global [AOI](file:///C:\Research-Samad\EMDAT\doc\AOI.html)) or can be active during certain time intervals.

In order to calculate the features for an [AOI](file:///C:\Research-Samad\EMDAT\doc\AOI.html) instance, you need to create an [AOI\_Stat](AOI.html#AOI_Stat) instance and map it to a target AOI object by passing it to the [AOI\_Stat](file:///C:\Research-Samad\EMDAT\doc\AOI.html#AOI_Stat) constructor. The resulting [AOI\_Stat](file:///C:\Research-Samad\EMDAT\doc\AOI.html#AOI_Stat) will calculate all features related to the given [AOI](file:///C:\Research-Samad\EMDAT\doc\AOI.html) and store them for later reference.

## Recording Class

Recording is a class used to hold all the data from one [Recording](Recording.html) (i.e., one complete experiment session) for one participant. It also has some useful functions for reading different types of files.

## Participant Class

Participant is the parent class for a project specific class that holds the information for one Participant in the experiment. There are some project specific functionalities that should to be overridden by the child class in order to use EMDAT. This class is the only component of EMDAT that is not fully implemented by design in order to address the needs of different projects. The following two methods cover the project specific aspects of EMDAT:

**Partition: This method is responsible for generating the list of Scenes and their corresponding Segments. Please see the BasicParticipant class for a sample implementation of this function that reads ‘.seg’ files and generates the Scene list.**

**read\_participants: This method is responsible for generating the full name of the files used for each participant (e.g., the full name of the ‘All-Data.tsv’ file for the first participant is ‘P1\_All-Data.tsv’ and so forth). Then it generates a list of the Participant objects by passing the file names to the Participant’s constructor method.**

# Using EMDAT

## Exporting files from Tobii studio

As explained in explained in Chapter 2, in order to use EMDAT on your data, you will need to export ‘**All-Data.tsv’**, ‘**Fixation-Data.tsv’** and **‘Event-Data.tsv’** files for each participant in your experiment. Please refer to your Tobii studio manual for details of how this can be done.

## Generating ‘.seg’ and ‘.aoi’ files

The other two types of information that is needed by EMDAT are the definition of AOIs (‘.aoi’ file) and the definition of Scene and Segments (‘.seg’ files).

If you are using AOIs that are active throughout the experiment for one user, then you can easily define and export AOIs using AOI tool in Tobii studio. However, if your AOIs are active only in certain time intervals for each user, you may need to create customized ‘.aoi’ files for each user according to the extended ‘.aoi’ format as explained in section 2.2.

When it comes to defining Scene and Segments, depending on the experiment design you can have a very basic one line ‘.seg’ file for each user such as:

MainScene<tab>Seg1<tab>0<tab>120000<newline>

This line defines one Scene with only one Segment that covers the whole experiment for the user (120 seconds in the above example). You can also have more complex set of Scenes with multiple Segments or even Scenes with overlapping time intervals. All these different designs are supported by the ‘.seg’ file format described in Section 2.2.

## Changing the BasicParticipant class

EMDAT distribution includes an example implementation of the Participant class, called BasicParticipant. This class implements a typical scenario where all the input files are provided and the experiment only includes general AOIs. You can directly use this code for your project if you have a similar use case. The only change that may be necessary is in the **read\_participants method where the full names of the files are generated.**

**If you have a more complicated scenario** you will need to implement a more complicated **Partition method that generates the Scene list based on the ‘Events-Data.tsv’ files and/or some external log files generated by the software that was under study. It is important to note that, you will need to add a mechanism to generate ‘.aoi’ files for your dynamic AOIs as well.**

## Project specific Configurations

### Data Validation

### Automatic Restoration of Invalid samples

1. Available at [http://google-styleguide.googlecode.com/svn/trunk/pyguide.html#Comments](http://google-styleguide.googlecode.com/svn/trunk/pyguide.html%23Comments) [↑](#footnote-ref-1)