Eye-Clicker

System Requirements

SFWR ENG 4G06 / MECHTRON 4TB6
GROUP 8

Adam Gordon 400036676
Bowen Yuan 400005913
Jacky Chan 400015317
Jeffrey Sun 400022051
Leo Li 400032949
Tongfei Wang 001437618

Contents

1 Purpose

The purpose of the Eye-Clicker is to give our users an alternate method to use their computer. With our device, a user will be able to control their computer or laptop without having to use a mouse, and instead use their eyes. This is especially helpful to users who have limited mobility with their arms, those who are injured, or simply someone who wants to rest. Given an alternate method of using a computer, our device can help more people use a computer comfortably and be connected to the digital world we live in.

Eye tracking has become fairly popular over recent times, and our Eye-Clicker will utilize this concept to provide a full user experience that mimics the behavior of a mouse. The Eye-Clicker will identify where the user is looking at to position the cursor to that spot on the screen. It will also track a unique set of simple eye movements/actions as well as certain voice commands to deliver the appropriate response. Whether the user wishes to click, drag or enter text the Eye-Clicker will provide all those functionalities simply though tracking the eye and voice control.

To achieve this goal, we will train a base AI model to predict where the user looks on the screen based on the position of their eyes. Since the position of the user relative to the screen and camera angles may vary, the user will further calibrate the model in order to finetune it for their usage. This will allow for the best opportunity to achieve good accuracy when tracking the user's eyes.

1.1 Scope

The project will be based around tracking user's eye movements and making the cursor react, including but not limited to moving the cursor, left-clicking and click and dragging. This will be achieved through image processing, more specifically, human eye recognition and a well-developed algorithm to control mouse actions.

In-scope functionality items for the Eye-Clicker including the following:

- Calculate the position the user is looking on the computer display using trained AI models.
- Calibration system to increase the accuracy of the Eye-Clicker by adjusting for camera position and user position
- Allows user to perform a left-click as well as dragging with certain special actions that can be performed with eyes (e.g. blink an eye, staring, etc.)
- Be able to let user disable the Eye-Clicker system to prevent misoperation.
- Voice control is available to allow the user to quickly and ergonomically execute commands such as enabling the Eye-Clicker.

The following items are out of scope: Allows user to control the cursor with voice input

1.2 Context Diagram With Boundaries

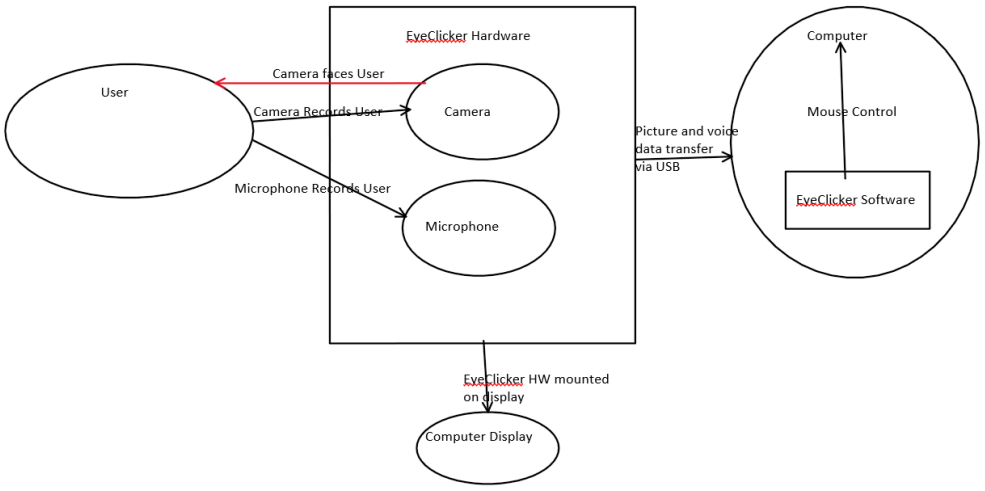

Figure 1: Context Diagram

1.3 Diagram of Components

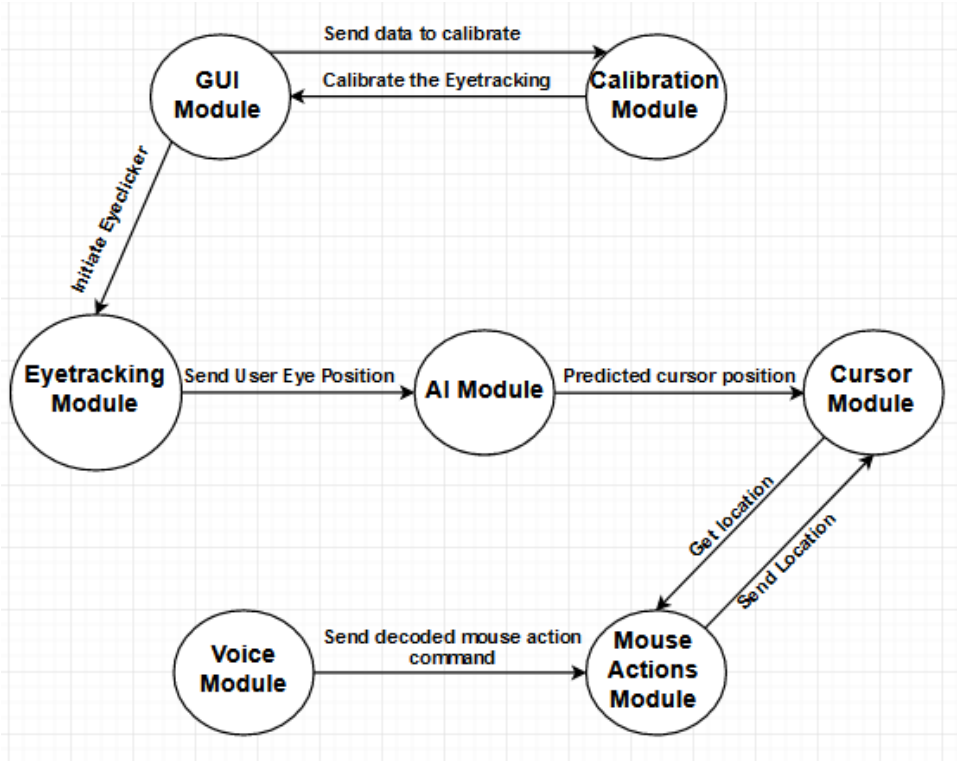

Figure 2: Diagram of Components

1.4 Assumptions

A1	There will be sufficient lighting in the environment of eye detection.
Rationale	The eyes must be visible to the camera for the system to detect where the user is looking on the screen.

A2	There is only one pair of eyes in the view of the camera that belongs to the same person.
Rationale	Having more than one pair of eyes will be very confusing to the system and it is difficult to determine who is the primary user.

A3	The camera is looking generally in the direction of the user's face. The eyes are easily seen and are not at an obscure angle.
Rationale	This helps with the detection of the eyes and will greatly improve the accuracy of eye tracking.

A4	The user must recalibrate if the camera or user position is altered.
Rationale	The calibration is relative to the user and camera position. Recalibration is necessary for eye detection as the new position will be foreign to the system.

A5	The user's eyes are not extremely far from the camera such as more than 1 meter away.
Rationale	If the eyes are really small, the error of margin will be very high as very small movements could be the difference between the left and right ends of the screen.

A6	The user follows the calibration steps appropriately.
Rationale	If the user does not follow the guidelines of the calibration and perform all the steps, the eye detection may be skewed.

2 System Variables

2.1 Monitored and Controlled Variables

Name	Type	Range	Units	Comment(s)
image_frame	Binary	N/A	N/A	Video input captured by web camera
coord_landmark_left	Two dimensional array of floating numbers	[0,1]	Percent	Coordinates of points around user's left eye(in percentage)
coord_landmark_right	Two dimensional array of floating numbers	[0,1]	Percent	Coordinates of points around user's right eye(in percentage)
coord_pupil	Array of floating numbers	[0, 1]	Percent	Coordinates of points indicating left and right pupil positions(in percentage)
get_cursor_xy	Boolean	[0, 1]	Percent	Signal to tell the module to output the current position of the cursor
voice_input	Binary	N/A	N/A	Voice commands from the user
coord_pupil_training	Two dimensional array of floating numbers	[0, 1]	Percent	Array of coordinates of points indicating left and right pupil positions(in percentage)
coord_pupillooking_training	Two dimensional array of floating numbers	[0, 1]	Percent	Array of coordinates of points indicating where the user's eyes are looking at on the screen(in percentage)
GUI_start	Boolean	[0,1]	N/A	Signal for initializing the Eyetracking Module
GUI_exit	Boolean	[0,1]	N/A	Signal for exiting the GUI Module
GUI_calibration	Boolean	[0,1]	N/A	Signal for initializing the Calibration Module

Name	Type	Range	Units	Comment(s)
predicted_cursor_position	Array of floating numbers	[0, 1]	N/A	The coordinate where the cursor will move to
cursor_xy	Array of floating numbers	[0,1]	N/A	Current coordinate of the cursor(in percentage)
click_left	Boolean	[0,1]	N/A	Perform a double left click action if Click_left = 1
click_right	Boolean	[0,1]	N/A	Perform a right click action if Click_right = 1
init_calibration	Boolean	[0, 1]	N/A	Initially 0, 1 when user clicks Calibrate, set to 0 when calibration is initiated
calibration_in_proccess	Boolean	[0, 1]	N/A	Set to 1 when calibration is in process, else 0
predicted_cursor_position	Array of floating numbers	[0, 1]	N/A	Coordinates of points indicating where the user's eyes are looking at on the screen(in percentage)
Init_eyetracking	Boolean	[0,1]	N/A	Initialize the Eyetracking Module
Init_calibration	Boolean	[0,1]	N/A	Initialize the Calibration Module

2.2 Constants

3 Behaviour Overview

- **Eye-Tracking Module:** Provide eyelid and pupil position information based on web-cam video. This data will be communicated to different components of the system.
- **Cursor Module:** Responsible for moving the cursor to given locations as well as getting the current coordinates of the cursor.
- **Voice Module:** Responsible for detecting voice commands from the user and initiating the response.
- **Mouse Action Module:** Responsible for executing mouse click actions.
- **Calibration Module:** Responsible for increasing the accuracy of the Eye-Clicker.
- **AI Module:** Responsible for training the data by machine learning algorithms and using the algorithm to detect the desired cursor position.
- **GUI Module:** Responsible for interacting with users by displaying information and provide actions.

4 Component Traceability

Component Module:	Performance Requirement:
Eye-Tracking Module	Performance Requirement 1
	Performance Requirement 2
	Performance Requirement 3
	Performance Requirement 4

Component Module:	Performance Requirement:
Cursor Module	Performance Requirement 3

Component Module:	Performance Requirement:
Voice Module	Performance Requirement 5
	Performance Requirement 6

Component Module:	Performance Requirement:
Calibration Module	Performance Requirement 1

Component Module:	Performance Requirement:
AI Module	Performance Requirement 1
	Performance Requirement 3

5 Component Overview

5.1 Eye-Tracking

5.1.1 Description

This module will capture video input from a web camera at a minimum of 10fps (frames per second). The web camera will constantly retrieve image frames of the user and process the image data by the algorithm written with an openCV library. The output of the algorithm will be the coordinates of the points around the user's eyes and the point indicating the pupil's position (in percentage relative to the screen, for example the middle point will have a coordinate of (0.5, 0.5)).

5.1.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comments
image_frame	Video Frame	N/A	N/A	Video input captured by web camera
eye_tracking_activated	Boolean	[0, 1]	N/A	True if this module is activated

Outputs:

Output Name	Output Type	Range	Units	Comments
coord_landmark_left	Two dimensional array of floating numbers*	[0,1]	Percent	Coordinates of points around user's left eye (in percentage)
coord_landmark_right	Two dimensional array of floating numbers*	[0,1]	Percent	Coordinates of points around user's right eye (in percentage)
coord_pupil_left	Array of 2 floating numbers	[0, 1]	Percent	Coordinates of points indicating left pupil position (in percentage)
coord_pupil_right	Array of 2 floating numbers	[0, 1]	Percent	Coordinates of points indicating right pupil position (in percentage)

*the array will be in essence a list of points

5.1.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
image_frame	Video Frame	User eye blinked	Drop the image frame so it will not be analyzed
eye_tracking_activated	Boolean	N/A	N/A

5.1.4 Timing Constraints

The algorithm must be efficient enough to give out the coordinate data within the limited delay time (350 ms).

5.1.5 Initialization

The module is initialized by clicking activate in the GUI module.

5.1.6 Web Camera specifications

Video resolution: 1920*1080
Interface: USB 2.0/1.1

5.2 Cursor

5.2.1 Description

This module will receive predicted cursor position from the trained AI model and move the cursor to the predicted location.
The module will start to move the cursor once the getting position signal is set to true.

5.2.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comments
predicted_cursor_position	Array of floating numbers	[0, 1]	N/A	The coordinate where the cursor will move to
get_cursor_xy	Boolean	[0, 1]	N/A	Signal to tell the module to output the current position of the cursor

Outputs:

Output Name	Output Type	Range	Units	Comments
cursor_xy	Array of floating numbers	[0,1]	N/A	Current coordinate of the cursor(in percentage)

5.2.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
predicted_cursor_position	Array of floating numbers	N/A	N/A
get_cursor_xy	Boolean	N/A	N/A

5.2.4 Timing Constraints

The module must provide cursor coordinates in real time for accuracy purpose.

5.2.5 Initialization

The module is initialized at the start-up of the Eye-Clicker.

5.3 Voice

5.3.1 Description

This module will receive sound information from a microphone. The microphone is constantly retrieving sounds from the surroundings and it is activated when the user says "Hey Eye-Clicker". The available commands are "Hey Eye-Clicker, left click", and "Hey Eye-Clicker, right click".

5.3.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comments
voice_input	Binary	N/A	N/A	Voice commands from the user

Outputs:

Output Name	Output Type	Range	Units	Comments
click_left	Boolean	[0,1]	N/A	Perform a double left click action if click_left = 1
click_right	Boolean	[0,1]	N/A	Perform a right click action if click_right = 1

5.3.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
voice_input	Byte Data Stream	Unknown voice commands	N/A

5.3.4 Timing Constraints

The information must process in time for ease of use for the user.

5.3.5 Initialization

The module is initialized at the start-up of the Eye-Clicker.

5.4 Mouse Actions

5.4.1 Description

This module will communicate with the computer system to perform clicking actions. It takes certain outputs from Voice module as the input, and performs the corresponding mouse click actions.

5.4.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comments
click_left	Boolean	[0,1]	N/A	Output from Voice module
click_right	Boolean	[0,1]	N/A	Output from voice module

Outputs:

Output Name	Output Type	Range	Units	Comments
left_click	N/A	N/A	N/A	OS left click actions when Click_left = 1
right_click	N/A	N/A	N/A	OS right click actions when Click_right = 1

5.4.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
click_left	Boolean	Click_left = 0	Do nothing when Click_left = 0
click_right	Boolean	Click_right = 0	Do nothing when Click_right = 0

5.4.4 Timing Constraints

The module should process the input right after Voice module gives an output with a maximum response time of 3 seconds (Performance Requirement 5).

5.4.5 Initialization

The module works alongside with Voice module. It is initialized at the same time as the Voice module.

5.5 Calibration

5.5.1 Description

This module will be used to calibrate the Eye-Clicker to improves its accuracy. It will do this by training an AI model. The input and output data for the model will be gathered by entering "Calibration" through the GUI. Once the mode is entered, the system will ask the user to follow the cursor with their eyes. The Calibration module will then use the Cursor module to move the cursor a maximum of 30 seconds. During this 30 second calibration time the Eye-Tracking module will track the eye position coordinates in synchronous with cursor module tracking mouse coordinates. This data will then be used to further train the AI module uniquely for the user.

5.5.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comments
init_calibration	Boolean	[0, 1]	N/A	Initially 0, 1 when user clicks Calibrate, set to 0 when calibration is initiated

Outputs:

Output Name	Output Type	Range	Units	Comments
calibration_in_proccess	Boolean	[0, 1]	N/A	Set to 1 when calibration is in process, else 0

5.5.3 Timing Constraints

The maximum length of time the user will need to track the cursor with their eyes will be 30 seconds.

5.5.4 Initialization

The Calibration module will be initiated through the GUI menu.

5.6 AI module

5.6.1 Description

This module will be used to take the coordinate of landmark and pupil points as input to predict user's desired cursor position based on machine learning algorithm. At first, AI module will be used to train a basic algorithm based on the data eye-tracking module gathered. It will also be used to improve its prediction accuracy by taking user's calibration data as training data.

5.6.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comments
coord_landmark_left_traning	ArrayList of two dimensional array of floating numbers	[0, 1]	Percent	An arraylist of coordinates of points indicating left and right pupil positions(in percentage)
coord_landmark_right_traning	ArrayList of two dimensional array of floating numbers	[0, 1]	Percent	An arraylist of coordinates of points indicating where the user's eyes are looking at on the screen(in percentage)
coord_pupil_left_training	ArrayList of an array of 2 floating numbers	[0, 1]	Percent	An arraylist of coordinates of points indicating left pupil position (in percentage)
coord_pupil_right_training	ArrayList of an array of 2 floating numbers	[0, 1]	Percent	An arraylist of coordinates of points indicating left and right pupil positions(in percentage)
coord_landmark_left	Two dimensional array of floating numbers	[0,1]	Percent	Coordinates of points around user's left eye (in percentage)
coord_landmark_right	Two dimensional array of floating numbers	[0,1]	Percent	Coordinates of points around user's right eye (in percentage)
coord_pupil_left	Array of 2 floating numbers	[0, 1]	Percent	Coordinates of points indicating left pupil position (in percentage)
coord_pupil_right	Array of 2 floating numbers	[0, 1]	Percent	Coordinates of points indicating right pupil position (in percentage)

Outputs:

Output Name	Input Type	Range	Units	Comments
predicted_cursor_position	Array of floating numbers	[0, 1]	N/A	Coordinates of points indicating where the user's eyes are looking at on the screen(in percentage)

5.6.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
invalid_pupil_coordinate	Array of float	Coordinate number is out of range	N/A

5.6.4 Timing Constraints

1.The process of training the AI should not be more than 20 seconds.
2.The process of using the existing AI model to predict desired cursor position should not be more than 400ms.

5.6.5 Initialization

The module is initialized at the start-up of the Calibration module.

5.7 GUI

5.7.1 Description

The GUI(graphical user interface) module is a system of interactive visual components for our software. This module will display objects that convey information, and represent actions such as calibration and exit the program which can be taken by the user.

5.7.2 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comments
GUI_start	Boolean	[0,1]	N/A	Signal for initializing the Eyetracking Module
GUI_calibration	Boolean	[0,1]	N/A	Signal for initializing the Calibration Module

Outputs:

Output Name	Input Type	Range	Units	Comments
init_eyetracking	Boolean	[0,1]	N/A	Initialize the Eyetracking Module
init_calibration	Boolean	[0,1]	N/A	Initialize the Calibration Module

5.7.3 Exception Handling

Input Name	Input Type	Exception	Exception Handling
init_eyetracking	Boolean	ModuleAlreadyInited	Do nothing
init_calibration	Boolean	ModuleAlreadyInited	Do nothing

5.7.4 Timing Constraints

The GUI Module should move the cursor or execute the clicking no longer than 100 millisecond right after Cursor Module and Mouse Action Module give the output.

5.7.5 Initialization

The module is initialized at the start-up of the Eye-Clicker.

6 Likelihood of Change

Module	Likelihood of Change	Rationale
Eye-Tracking Module	Very Unlikely	Key Implementation aspect
Cursor Module	Very Unlikely	Key Implementation aspect
Voice Module	Very Unlikely	Key Implementation aspect
Mouse Actions Module	Very Unlikely	Key Implementation aspect
Calibration Module	Very Unlikely	Key Implementation aspect
AI module	Very Unlikely	Key Implementation aspect
GUI Module	Very Unlikely	Key Implementation aspect

7 Normal Operation

During the calibration stage, the user should be within the effective working distance of the Eye-Clicker (60cm of the screen). Sitting too far away from the camera causes inaccuracy or worsens the ability of Eye-Clicker to locate users' eyes and to control the cursor accurately.

During the working stage after calibration, the user now can navigate the cursor using their eye movements. Left-clicking and dragging are the basic functions the user can operate. These functions operate at the optimal level when the user is sitting at a stable position and as close to the position that the system was previously calibrated in.

Speech recognition is one of the solutions to help the user have more capability to do a task. When using speech recognition, the user should speak at a steady pace to obtain optimal performance. When the user wants to type something after clicking into a text box, it allows the user to speak to the system and the system will convert the speech into text. It can also activate "reading mode" by saying the command "Disable Eye-Clicker (Disable mouse clicking and dragging)" to avoid unintended actions when the user is staring at a spot for a long time (reading an article or watching a TV show).

8 Handling Undesired Event

- Unable to click the desired point on the screen. When the user is having trouble clicking the desired item, one should redo the calibration to improve the accuracy. To access the calibration test during run time, simply activate speech recognition and say "Recalibrate".
- Staring at the screen and accidentally activate clicking or dragging. When the user wants to read an article or watch a TV show, one might accidentally activate undesired actions like mouse left-clicking and mouse dragging. To solve this issue, utilize

Eye-Clicker's voice commands and say "Eye-Clicker Disable". When the user wants to use the Eye-Clicker again, say "Eye-Clicker Enable" to continue controlling the cursor with eyes' movements.

- Eye-Clicker cannot locate the user's eyes after turning his/her head away from the screen. When the user looks away from the screen for too long, Eye-Clicker might lose track of the locations of the user's eyes. To solve this problem, one should get closer to the camera and stare closer into it for a few seconds and then one can get back to the normal sitting position and possibly recalibrate.

9 Requirements

The scope of the project will be limited to meeting the requirements on a display size of 35.56cm (14inches) with a 16:9 aspect ratio while the user is 60cm. If desired the software will still be compatible with different display sizes, aspect ratios and viewing distances, however the accuracy may be degraded. This user setup constraint allows for simplification when training the AI model.

Performance Requirements:

- The deviation of detecting where the Eye-Clicker thinks the user is looking compared to the actual position the user is looking at must be smaller than 4cm while within effective working distance of the Eye-Clicker (60cm from the screen). This accuracy requirement will be met for only a user's display size of 35.56cm (14inches) with a 16:9 aspect ratio.
- The latency of detecting the users' eyes position must be less than 150 milliseconds (Eye-Tracker module).
- The latency of when then user looks at a spot on the display versus when the Eye-Clicker system moves the mouse to the predicted position on the display should be less 400ms (entire system).
- The latency of detecting whenever users is blinking must be shorter than 150 milliseconds.
- The latency of receiving, translating, and executing users' voice command must be shorter than 3 seconds after the user stops talking.
- The accuracy of speech recognition must be greater than 80 percent.

10 Appendix

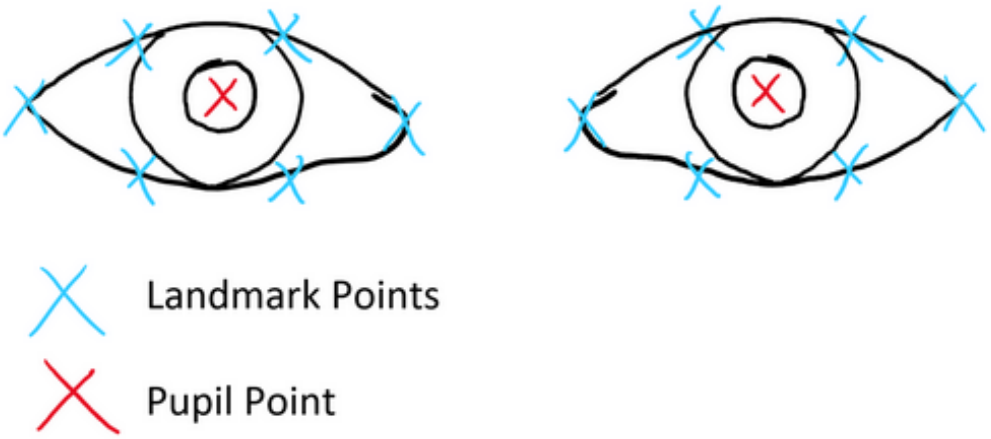

Figure 3: Eyetracking Coordinates