Gaze Behavior Inferencing Data Processing

Faculty Advisor: Sangjin Hong

Angelique Bryson, Ian Franco Del Rosario

Jecenia Carcamo, Angel Lee, Vinay Pillay

Stony Brook University
Electrical and Computer Engineering

VISION & DEVOUR

Introduction

The goal of the project is to collect and analyze gaze behaviour. Our project essentially is a monitoring system used for the analysis of an eye-gaze estimation system for the evaluation of media content. To analyze the gaze behaviour, the monitoring system will reflect the intention of content given from the content provider to accomplish a goal. If a content provider wants to emphasize a key content or focus on a main product they would want the user to concentrate on a particular image. The content provider will deliberately show the intentional image (Pattern B) while the user will focus on the gaze behaviour presented (Pattern A). We will study the Intention Flow, the image that the provider wants the user to look at, compared to the Gaze profile, the image that the user views.

In our application, a camera takes the user's gaze data of where they are looking at on the screen. The gaze data is coordinates (Ex: 600, 400) of where the user is looking at on the screen. Using GazePointer and GazeFlow API, the user looks at an image with specific Regions of Interest (ROI's) and the server converts the coordinates to a specific ROI the user is looking at. We also created a headset prototype to increase the accuracy of the gaze detection and due to COVID-19, we couldn't work on the prototype as much as we wanted to and instead created a 3D model of what we wanted our final headset to look like.

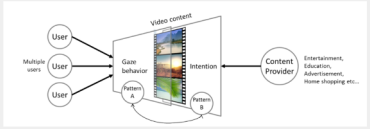

Figure 1: How content providers can use Gaze Detection

Engineering Specifications

Figure 2: Testing Position

Constraints vs. Intrusive Device

- Lower frame rate/resolution
- No eye glasses
- Very sensitive to movement
- Low lit environments
- Less accurate and precise

Device Parameters:

- Webcam at least HD resolution 720p, or 1280 x 720
- Windows 10, 4GB RAM minimum
- SSD drive, i5/i7 dual core processor or better

Environmental Parameters:

- User must be 50-80 cm from eye tracker
- Flat, static platform
- Avoid high amount of near-infrared spectrum light
- Contrast of webcam depends on ambient light

GazePointer and GazeFlowAPI

GazePointer is an eye tracking software made by Szymon Deja, a PhD student who works on human-computer interaction. GazeFlowAPI is the library that tracks the user's gaze, head position, and head rotation from the GazePointer application.

Figure 3: GazePointer GUI

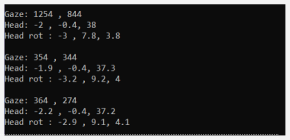

Figure 4: GazeFlowAPI Data

Client-Server Connection

The client and server setup is used to create a communication platform that allows gaze data to be sent to each other. The provider will be able to collect the data of the user's gaze behaviour from the client. The figure shows a basic layout of how the client program and the server program are connected. The client and server are connected by having the same IP address and port number.

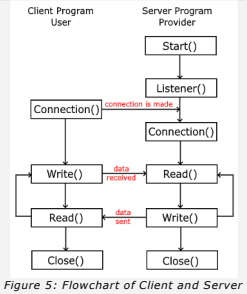

Figure 5: Flowchart of Client and Server

Region of Interest - Software Results

An ROI is defined as a location that a person can focus on. For our program, this will consist of a range of pixels on the screen. During our gaze collection, when a user looks at any point in this range, our system will record this interaction as a sample. This sample is stored as the region's index. Once a number of samples have been taken, it will be sent to our server, and the server will save this interaction into files for later processing.

Figure 6, 7, 8 - The content image, the ROI, and the composite overlay shown by the ROI maker.

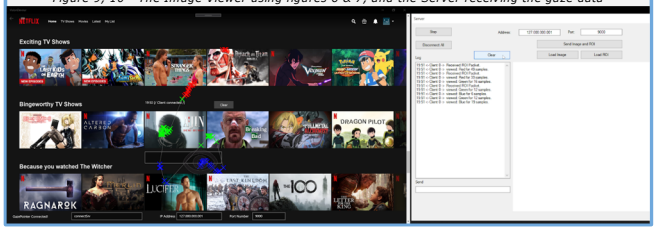

Figure 9, 10 - The Image Viewer using figures 6 & 7, and the Server receiving the gaze data

Initial Prototype

To build this prototype, we used a baseball helmet, a seven inch monitor and a camera. We decided to use a helmet and wood blocks because it was the only way the monitor would stay up and not fall. The areas of the prototype we decided that needed improvements were the appearance and size of the headset and the quality of the cameras being used.

Figure 11: User Wearing Prototype

Figure 12: Prototype

Simulation - Hardware Results

We used Rhinoceros 3D computer graphics and computer-aided design application software to create our final hardware design. By developing an intrusive device, we were successfully able to process the data of the human gaze with a camera at the level of the user. This will allow the user to move their heads and body freely, with better precision and accuracy.

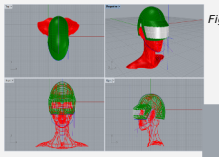

Figure 13: Detailed Simulation

System Features:

Real-time gaze
Client-Server Capability
ROI Maker
Calibration
Webcam eye tracking
Compatible with all webcams
Head position data
EyeMouse/Eye PC control

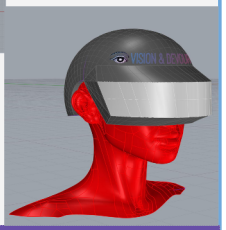

Figure 14: Final Product

Glossary

ROI - Region of Interest
API - Application Programming Interface
RAM - Random Access Memory
SSD - Solid State Drive
GUI - Graphical User Interface
3D - 3-Dimensional
HD - High Definition
PC - Personal Computer
COVID-19 - Coronavirus Disease 2019

Acknowledgements

We would like to thank our faculty advisor, Sangjin Hong, for giving us the opportunity to work with him as well as Szymon Deja, the creator of GazePointer and GazeFlowAPI.

Vision & Devour