Implicit Gaze Interaction for Information Visualization

by

Feiyang Wang

A thesis submitted to the
School of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Faculty of Science
University of Ontario Institute of Technology (Ontario Tech University)
Oshawa, Ontario, Canada
August, 2021

Thesis Examination Information

Submitted by: **Feiyang Wang**

Master of Science in **Computer Science**

Thesis Title: Implicit Gaze Interaction for Information Visualization

An oral defense of this thesis took place on August 3, 2021 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee: Dr. Patrick Hung

Research Supervisor: Dr. Christopher Collins

Examining Committee Member: Dr. Faisal Qureshi

Thesis Examiner: Dr. Loutfouz Zaman

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

This thesis presents a novel marker-free method for identifying screens of interest when using head-mounted eye-tracking for visualization in cluttered and multi-screen environments. The presented approach offers a solution for discerning visualization entities from sparse backgrounds by incorporating edge-detection into the existing pipeline. The system allows for both more efficient screen identification and improved accuracy over the state-of-the-art ORB algorithm. To make use of this pipeline in visualization applications, a model is introduced to track a user's interest in rendered visualization objects by collecting the gaze data and calculating the object group's interest scores across selected visual features. With the interest model, We offer an implicit gaze interaction system that provides subtle interaction supports to improve group-of-interest objects visibility and to ease object selection in crowded regions of information visualizations.

Keywords: gaze estimation; feature detection; attention modelling; implicit interaction; overplotting visualization

Author's Declaration

I, Feiyang Wang, hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I authorize University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Feiyang Wang

Statement of Contributions

I hereby certify that I am the sole author of this thesis. Part of the work described in Chapter 3 has been published as:

F. Wang, A. J. Bradley, and C. Collins. "Eye Tracking for Target Acquisition in Sparse Visualizations". *ACM Symposium on Eye Tracking Research and Applications.* ETRA '20 Short Papers. Association for Computing Machinery, Stuttgart, Germany, 2020. ISBN: 9781450371346. DOI:10.1145/3379156.3391834.

I performed the majority of the synthesis, experiments, and writing of the manuscript. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the main source of the creative works and/or inventive knowledge described in this thesis.

Acknowledgements

I would like to express my gratitude to all those who helped me during the writing of this thesis.

My deepest gratitude goes first and foremost to Professor Christopher Collins, my supervisor, for his constant encouragement and guidance. The door to Prof. Collins' office was always open whenever I ran into a trouble spot or had a question about my research or writing. Without his consistent and illuminating instruction, this thesis could not have reached its present form.

Second, I would also like to thank the experts who participated in the defense and provided feedback for my thesis and research projects: Dr. Adam Bradley, Dr. Faisal Qureshi, Dr. Loutfouz Zaman and Mariana Akemi Shimabukuro. Without their passionate participation and input, the defense could not have been successfully conducted.

Finally, I must express my very profound gratitude to my family and to my girlfriend for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Tables

List of Figures

1 Introduction

In previous years, many eye-tracking studies focus on collaborative environments where detecting multiple displays and tracking the gaze on which screen is a known challenge [13, 60, 65]. In this context, to locate gaze on a screen, first the front view camera (see Figure 1.1b) image is used to determine the location and identity of the screen in the field of view, and the eye camera is used to determine the gaze point within the detected screen. Determining which display is the one of interest in cluttered, multi-screen environments in which the user is moving, is the error-prone and rate-limiting step. In response to this, state-of-the-art systems use one of two approaches. The first is to calibrate the eye tracker to the corners of the screen using ArUco markers[1] or enabling the eye tracker to easily identify the space of interest (e.g., screen itself, a visualization, another application, etc.) in relation to the marked corners [37]. The second is the ORB (Oriented FAST and Rotated BRIEF) feature detection algorithm which can extract unique key points from corner or edges of a target image and match them with key points from the interest area of the user's gaze [82].

When combined with information visualizations, the first approach is inconvenient since the markers could occupy the screen space and interfere with the visualization design. While the second approach with visualizations, some characteristics, such as cluttered or sparse plotting and white backgrounds, causes problems for feature matching of target acquisition. One possible solution is to place a textured pattern behind the visualization so that the ORB algorithm has sufficient visual features to discern the screen accurately. However, this approach can interfere with visualization design.

Although various types of eye-tracking systems (eye-tracker device and application) exist, they can be divided into two categories: diagnostic and interactive [29]. In human-computer interaction, cognitive science, and information visualization, diagnostic eye-tracking studies play an important role in quantitatively measuring people's visual attention as they solve visual tasks. For example, in a visual analytics study, an eye-tracker tracks a human subject who sits in front of a computer screen and reports the gaze-positions on the screen. Researchers then proceed to test their hypotheses by analyzing the collected data using visual and statistical ana-

[1] A synthetic square marker composed by a wide black border and an inner binary matrix which determines its identifier.

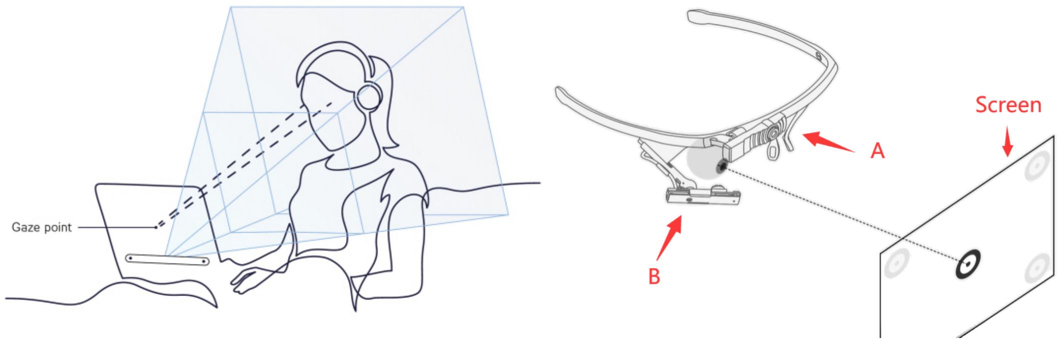
lytic techniques. Recently, a novel method (Data-of-interest (DOI) eye-tracking analysis) was presented to analyze eye-tracking data (accumulated as a stream of 2D gaze-samples) by relating the gaze samples to visual contents on data visualization [2]. When compared to traditional point-based or area of interest (AOI)-based analysis [8], the DOI method does not require to manually relate eye-tracking data to the visual stimulus or defined AOIs after a user study. The DOI instruments the visualization and can automatically map the gaze samples to data objects on the screen in real time. To exploit the analogy with the AOI nomenclature, such eye-tracked data objects are referred to as DOI. DOI analysis is able to answer many questions that traditional analysis approaches cannot, especially in experiments of significantly longer sessions and operating in data space.

For gaze interactive systems, eye-tracking is used to change a graphical interface based on a user's visual attention, such as using eye-tracking as an alternative to pointing devices (e.g., mouse, touch interface) or text inputs. However, gaze interaction within complex systems sometimes can be inefficient and error-prone since unintentional gaze fixations can interfere with a system's judgment about the intention of the user and trigger unintentional system responses. Also, holding the gaze on a point for a long time to explicitly activate a system component can cause both mental discomfort and physical exhaustion. Thus the use of gaze for this purpose has fallen out of wide use [34], and proved to be a bottleneck for designing advanced gaze interaction.

To respond to these challenges, this work first studies the solution for target screen acquisition on sparse visualization in a collaborative environment with multiple screens. Next, it investigates the application of the DOI method to visual attention modelling in visual analysis. Finally, we look into the use of gaze as a form of implicit interaction which can aid in using information visualizations without requiring long dwell times nor triggering disruptive visual changes due to unintentional gaze actions.

1.1 Motivation

Eye-tracking is a process of measuring human gaze behaviours and invaluable at explaining how people perceive, solve visual tasks, and use interfaces [29] and used widely in psychology and cognitive science to help researchers understand thought and affect mechanisms [79]. As eye-tracking technologies have become more portable, accurate, and inexpensive, it is possible for modern eye-tracking systems to precisely locate a user's gaze on different display devices (e.g., computer screens, projection screens, hand-held, and wearable displays) and classify gaze actions by typical measures (e.g., fixation duration, fixation count, saccade length). This presents an opportunity to revisit the use of gaze as a mode of interaction. Our research interest



(a) Remote Eye-tracker (Tobii[2]) (b) Head-mounted Eye-tracker (Pupil-core[3])

Figure 1.1: Two different types of eye-tracking devices. Figure (a) is a Tobii remote eye-tracker calculating the eye's position and gaze point on a computer monitor. For Pupil-core in figure (b), A is the front view camera and B is the eye camera.

falls into this area, especially in eye-tracking methodology, visual attention modelling, implicit gaze interaction and non-distracting guidance.

As the design space for gaze interaction extens from the development of eye-tracking technology, the potential of gaze used as an interaction input has increased as well. Compared to traditional eye-tracking of a single user on one screen, scenarios with multiple users or screens can also benefit from the gaze interaction. One example is to replace laser pointers with eye-trackers to indicate locations among multiple screens in a collaborative working environment; one can do that by removing a time-consuming step that requires the users to precisely specify which screen and what location they are looking at. In the aforementioned example, eye-tracking not only allows users to understand each other by knowing the location of other's gaze point, but also to simplify and to improve the communication accuracy. To achieve that, it is necessary to provide a reliable and robust system that guarantees the success of eye-tracking of multiple users at the same time across multiple screens. Eye-tracking within data visualization research, the data collected by the DOI method is highly granular with semantic content as it is linked to the data that powers the visualization. In addition to that, DOI does not require any human pre-processing, object viewing data can be collected and analyzed efficiently for many topics, using interactive visualizations, for visual attention modelling, to narrow the gap between the user and visualization.

[2] https://tech.tobii.com/technology/what-is-eye-tracking/
[3] https://docs.pupil-labs.com/core/

1.2 Contributions

The main contributions of this work are as follows:

1. A video processing pipeline for dynamic screen detection that can easily differentiate the boundary of the target screen and the others, which enables a more efficient and fluid experience.

2. An interest model to detect data-of-interest (DOI) and to predict a user's groups-of-interest (GOI); this model is informed by a gaze action detection algorithm and through the abstraction of common features of data objects from multiple dimensions.

3. Two interaction techniques that use the interest model to proactively detect gaze actions and respond with non-distracting and subtle support to tackle down classic visual analytic problems (e.g. information discovery, target selection), and improve the usability and readability of data visualizations.

1.3 Thesis Organization

This thesis is organized as follows: Chapter 2 discusses the previous research done on eye-tracking methodology and gaze interaction design. We then proceed to provide a pipeline to solve the problem of interest in Chapter 3. The proposed gaze-data based interest model is described in Chapter 4. Chapter 5 discuss using gaze as explicit or implicit input for human-computer interaction. Two applications of implicit gaze interaction were proposed to deal with overplotting visualization and improve existing target selection methods on graphical-user-interface. Chapter 6 proposes an evaluation plan with two studies for two implicit interaction application on visualization. We conclude in Chapter 7 with limitations and future directions for this research.

2 Related Work

There is a large unexplored design space at the intersection of visualization and gaze interaction because of previous technological constraints. To gain a better understanding of the background behind choosing to focus on gaze interaction and information visualization, it is necessary to review several current trends and research areas in the field surrounding eye-tracking. The following chapter gives insight into the three basic thesis building topics: gaze estimation on screen-based applications, attention modelling with eye-tracking data, and implicit gaze interaction and application for data visualization.

2.1 Gaze Estimation on Screen-based Applications

A wide variety of eye tracking applications exist. They can broadly be categorized into diagnostic and interactive applications [29]. Interactive applications were initiated in the early 1980s [9] and further developed by Ware and Mikaelian [99]. A large number of novel applications has been proposed to use gaze information for improved interaction with screen-based applications. In terms of gaze estimation on displays, they can be divided into two classifications: remote (i.e., eye trackers placed at a display) and head-mounted. Gaze interaction with screens is mostly done through remote eye trackers and significant attention has been given to applications that assist disabled people [43]. In our work, a camera frame pipeline is proposed to detect contours of screens without using visual markers in advance and discern the correct screen for gaze interaction with ORB feature detection, which increases the accuracy of gaze estimation on visual content with large white space (e.g., information visualization) and enhance the stability of screen calibration when a user moving his head in a multiple screens collaborative environment.

2.1.1 Remote Eye-trackers

Remote eye-trackers can be used for interaction with attentive user interfaces [97] (e.g., gaze contingent displays [30] and EyePliances [89]), on public screens [83], or multiple screens [87]. Accurate gaze estimation on displays remains a significant challenge — particularly when remote eye trackers are used. They lack mobility and multiple user interaction since such trackers only allow a single user to interact with a display at any point in time, and any interaction is restricted to the tracking range of typically 50–80 cm in a central area in front of the display [91]. Previous work either focused on extending the tracking range of remote trackers [40, 70], or on calibration-free (spontaneous) interaction but was either limited to interaction along a horizontal axis, i.e., without full 2D gaze estimation [103, 104] or required dynamic interfaces [98]. Dostal et al. developed a system for multi-user gaze estimation on a big screen [25] and suffered from only being able to detect users when they stand separately facing the screen and react based on the user's proximity to the display. In this instance, eye-tracking is used to make sure the user is looking at the screen but is not used to determine exactly where they are focused. Stellmach et al. addressed the mobility (interacting from different positions/orientations) of users [92] by using an additional external tracking system. Later works used mobile remote eye-trackers (e.g., GazeDrone [54], EyeScout [53]) for position and movement independent gaze estimation on a large screen. Comparing to head-mounted eye-trackers, these eye-trackers are too complicated and expensive to be deployed in real-world scenarios.

2.1.2 Head-mounted Eye-trackers

Head-mounted eye-trackers may be more flexible for this purpose as they allow the user to move freely in front of the display. Early work on using head-mounted eye trackers for interaction still required calibration to a single, stationary display prior to first use [31]. More recent approaches aimed to estimate gaze dynamically but either required visual markers attached to the display [5, 49, 102] or in the environment to detect gaze on predefined interaction areas, e.g., to control a TV set [12]. While simplifying detection, the visual markers are limited by the need to place the markers on the objects. Both approaches require the display to be fully visible to the scene camera, which cannot be guaranteed at all times in mobile settings. This is a strong demonstration of the potential of this research area but is still limited by calibration issues.

With advances in computer vision, visual markers can be substituted with detecting the display directly in the scene camera's field of view. Mardanbegi et al. detect screens based on quadrilaterals found in the scene [65]. Turner et al. extended this to multiple displays (based on the displays' aspect ratios) by adding a second camera and a method for transparently switching between two calibrations [96]. GazeProjector suggests using head-mounted eye-trackers to automatically select feature points in a single calibration instead of screen borders for screen calibration to obtain accurate gaze estimation across multiple displays, which is more robust to changing light conditions and generalizes better to displays of arbitrary shape and size [59]. However, calibrating the screens based on their visual content may result in inaccuracy due to lacking feature points on the often white screens or competition between multiple displays in the view of a scene camera.

2.2 Spatial Relationships of Users and Displays

Tracking the spatial relationship of users can be done in two ways. First, external tracking equipment can be used to determine a user exact position in 3D space (and thus its spatial relationship to a display in the environment). The Proximity Toolkit makes use of such high-precision tracking equipment and provides an interface to acquire spatial relationships [26, 66]. While such a setup results in extremely high accuracy, it is cost-consuming for deployment and impractical for changing location frequently. Alternatively, the spatial relationship between a user and a display can be identified by using a camera. Many approaches temporarily show on-screen visual markers [94] or use dynamic markers following a camera's position [77]. More recently, natural feature tracking was used to determine spatial relationships. Herbert et al. used a ScaleInvariant Feature Transform (SIFT) to determine the camera's spatial relationship to a display [41]. Their system tried to identify a screenshot of the display in the device's camera stream. Virtual Projection extended this approach to dynamically updated displays [6]. Touch Projector further allowed for tracking multiple displays provided that display contents differ sufficiently [10]. Based on these underlying concepts, GazeProjector applies progressive algorithms FAST and FREAK (with their significantly improved matching accuracy [1]) to improve the tracking efficiency by increasing the frame rate from 10 fps in Virtual Projection [6] to more than 20 fps. In our work, a reference contour comparison approach is offered and used to reduce the frequency of using complicated computer vision algorithms for screen calibration, without reducing its accuracy. In Chapter 5, the relationship between the frame rate and the resolution of a scene camera is discussed with an experiment since the resolution may influence both the accuracy of gaze estimation and the frame rate per second of the estimation.

2.3 Eye-tracking Data Analysis

Eye-tracking data is traditionally interpreted in the space of rendered visual stimuli where gazes were recorded and used one of two analysis paradigms: point based or area of interest (AOI) based [8]. Point-Based analysis methods treat each gaze sample as independent points while AOI analyses aggregate gazes into areas of interest and operate at this higher level of abstraction. The major disadvantage of these two approaches requires analyzers to relate point-based gazes with the semantic contents or define AOIs over stimuli manually after the recording. Hence, the analysis process takes a prolonged time when the count of stimuli and visual contents increases. What's more, the process becomes prohibitively inefficient for interactive and dynamic stimuli (e.g. video) since analyzers have to define AOIs for each frame of a video.

Several solutions are proposed to overcome this weakness. One example is the automatic AOI annotations using gaze clustering algorithms [28, 78, 85]. However, an increase of complexity of visual contents in stimuli may increase the difficulty of the AOI annotation process. Stellmach et al. proposed the object of interest (OOI) concept for 3D stimuli where eye-trackers collect gaze points on the surface of 3D objects available in a scene [93]. Additionally, Steichen et al. [90] and Kurzhal et al. [58] suggested the possibilities of dynamic AOI annotations in the case of computer generated visual content. Sayeed et al. improved eye-tracking data analysis with his method data-of-interest (DOI) by automatically detecting which data objects a user of a visualization views [2]. As such, DOI is the mapping of gaze samples to data objects rather than pure pixel positions, which can answer questions that AOI cannot since DOI data can be significantly more granular and larger than AOI data. DOI can not only give visualization scientists a convenient and fast way to analyze and understand the quality of visualizations but also has a potential to model user's attention to a visualization. However, it is still not enough to tell a user's attention just by knowing what data objects he has seen. We need a method to distinguish and summarize the common feature from the collected data-of-interests to tell what colour or shape of the or which location of the objects are user viewed most. In Chapter 4, a top-down approach is proposed that divides the data-of-interest into different groups by one salient visual feature, and we refer to this method as group-of-interest.

2.4 Visual Attention Modelling

Modeling visual attention in images and videos has been an important area of research in psychophysics, computational modeling and neurophysiology. Current attention models generally fall into two main categories: *bottom-up* approaches and *top-down* approaches. Bottom-up attention models (stimulus driven) are based on the low-level features of the visual scene, while top-down models (goal driven) are determined by phenomenons such task, goals, experience and knowledge.

Seminal work by Koch and Ullman [55] used a purely bottom-up model that decodes a scene based on pre-attentive visual features (e.g. color, depth, and direction of motion) to create a saliency map – a two-dimensional topological map that encodes conspicuity across the entire scene. The central thesis of their work is salient features within a stimulus "stands out", thus attracting overt attention. They used a winner-take-all neural network to determine the most salient locations, and defined rules for shifting the processing focus which can be biased by proximity and similarity preferences. Much of the existing work on computational modeling of selective attention have adopted the idea of bottom-up feature extraction and saliency map (e.g. [44], [45], and [46]) to simulate human viewing behavior. Alvitta et al. proposed a mouse-click based hidden Markov model with the bottom-up approach for modeling and detecting attention during visual data exploration and demonstrate how their method can be used to predict future attending regions and actions [74].

Based on the DOI method and bottom-up attention approach, we contributed an interest model to represent a user's attention to salient features and related it in the form of scores presenting different interest levels (we refer to as group-of-interest) to each group of data objects on a visualization. This method improves the DOI method by automatically relating the DOI and predefined data object groups to study further the user's interestedness in the objects classed by various salient visual features and insight into visualization.

2.5 Explicit and Implicit Interaction with Gaze

The eye tracker provides information about the location of the screen that the user is looking at. Applications can use this data for explicit eye input that requires conscious control by the user, or can use it as an implicit source of information, often in combination with other input modalities (attentive user interfaces) [16].

In traditional human-computer interaction involving eye-tracking technology, gaze has always been used explicitly as a purposeful and attention-demanding control medium to engage with computers and replace other control devices such as mouse or microphone to assist disabled people. As the eye-tracking system improved and became inexpensive, gaze interaction became widely studied and extended to various areas including (gaze guesture [23, 27, 49], target selection [20, 91, 92, 101], screen attendance [25, 98, 103]).

As eye-tracking systems are getting increasingly capable of predicting the gaze location and modeling attention with advanced analysis methods, implicit gaze interactions may proliferate and partly replace explicit interactions. A public display that shows content when it senses gaze from human presence or a recommendation engine that utilizes user actions for social recommendations are typical examples [26]. In recent works, gaze is treated as an implicit input to infer user's attention or cognitive abilities for the system to dynamically adjust the interface or representation of the visualization [21, 73, 90]. In this thesis, we contribute interaction designs related to two different topics and support information access and discovery by improving the readability and usability of the visualization with implicit gaze interaction.

The first topic is a classic problem of overwhelming information that comes from overplotting visualizations, such as a scatter plot. Matejka et al. [67] offered a solution that dynamically optimizes a value to change the scatter plot's opacity as the data points increase. However, this method can only alter the opacity of the data points globally by manual adjustment, which randomly sacrifices data points partly to show others. In our work, we contribute a visualization that can automatically and dynamically detect user-interested data points and raise them to the top by interacting with implicit gaze input and conducting visual transition. The advantage of our implicit interaction system is that it can provide non-distracting guidance to show the information of interest to a user while the guidance is not being sensed by the user and keeps the integrity of the visualization.

The second one is target selection, a basic task for acquiring graphical-user-interface (GUI) components such as buttons, icons and menu options. With the increment in both size and resolution of computer displays, it becomes less available and efficient for a user to acquire small visual elements surrounded by multiple nearby objects on the large display with the traditional cursor techniques [51, 100]. One promising solution is the Bubble Cursor which can dynamically adjust the cursor's activation area until the closest target is captured [39]. This is equivalent to expanding the boundary of each target to the Voronoi region with the target center being the region center, so that the Voronoi diagram [69] defined by all targets fills the whole screen space. To improve target selection performance, the state of art work tried to calculate Voronoi tessellation based on the mouse movement to resize the effective area of the targets actively [19]. Mungguen et al. [20] improved Bubble Cursor by replacing the cursor from mouse to gaze. Based on these two ideas, we contribute a Gaze Additive Voronoi Cursor that takes gaze as implicit input as well as the cursor, changing the targets' effective area according to the gaze movement.

3 Marker-Free Gaze Tracking Pipeline

Using eye-tracking technology to pinpoint gaze in multi-screen environments is a challenge because first the display of interest must be detected, then the gaze point localized in screen coordinates. One common solution is placing ArUco markers at the corners of a screen. An ArUco marker is a synthetic square marker composed by a wide black border and an inner binary matrix which determines its identifier (id) [36]. The black border facilitates its fast detection in the world-camera image and the binary codification allows its identification and the application of error detection and correction techniques. In the circumstance of visual design, distraction is inevitable if ArUco markers are used, since the black square markers at the corners can divert the user's attention and interfere visual consistency. To avoid this, the ORB feature detection algorithm is used as an alternative approach, as it can detect the target screen based on the content on it rather than the ArUco markers.

ORB is basically a fusion of the FAST keypoint detector [80, 81] and BRIEF descriptor [14] with many modifications to enhance the performance. More detailed information of ORB is introduced in Section 3.3. One of the main difficulties in implementing head-mounted eye tracking with visualization is that the over abundance of white space often confuses the eye tracking system during calibration and on-going real-time gaze tracking. One solution is to place a textured pattern behind the visualization so that the ORB algorithm has sufficient visual features to discern the screen accurately. Similar to the ArUco markers, this approach can interfere with visualization design between points of interest and the background. Thus, we want to avoid that.

In this thesis, we contribute a method for multi-screen detection that allows users to forego ArUco markers and background textures in their visualizations. This method is designed to improve interaction times by removing a major step at the beginning of an engagement (marker registration) and allows visualizations to appear as designers intend without the compromise of manipulated background textures. Our method uses a pipeline approach to process data from eye tracking glasses. The high-level overview is shown in Figure 3.1 and consists of preprocessing to determine candidate screen contours in the world-view, ORB detection to determine the correct screen, and transformations of the gaze point to visualization-relative coordinates.

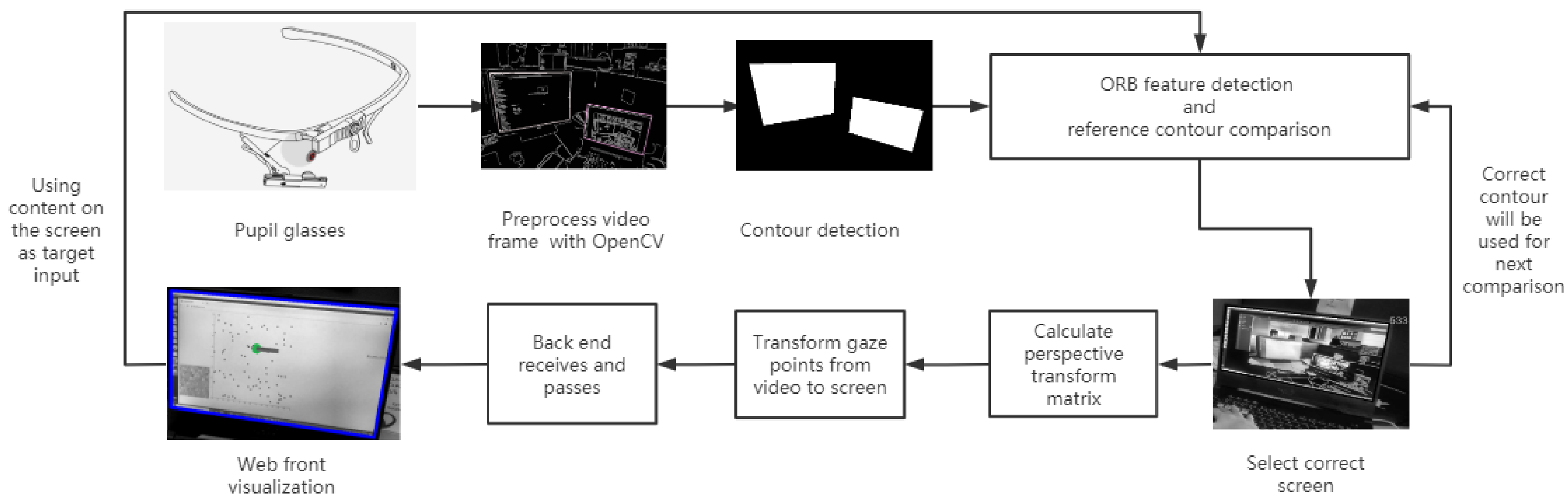


Figure 3.1: Our pipeline begins with the eye-tracking glasses providing a video frame and gaze point data to our program. OpenCV Filters are used to preprocess the frame and detect screen contours, followed by ORB feature detection and reference contour comparison to detect the correct screen. The identified screen is used for both calculating the homography matrix and the next frame detection. With the matrix, gaze points are transformed from 2D video plane to 2D screen plane. Finally, the back end server passes the gaze points to the web front end (showed in green point).

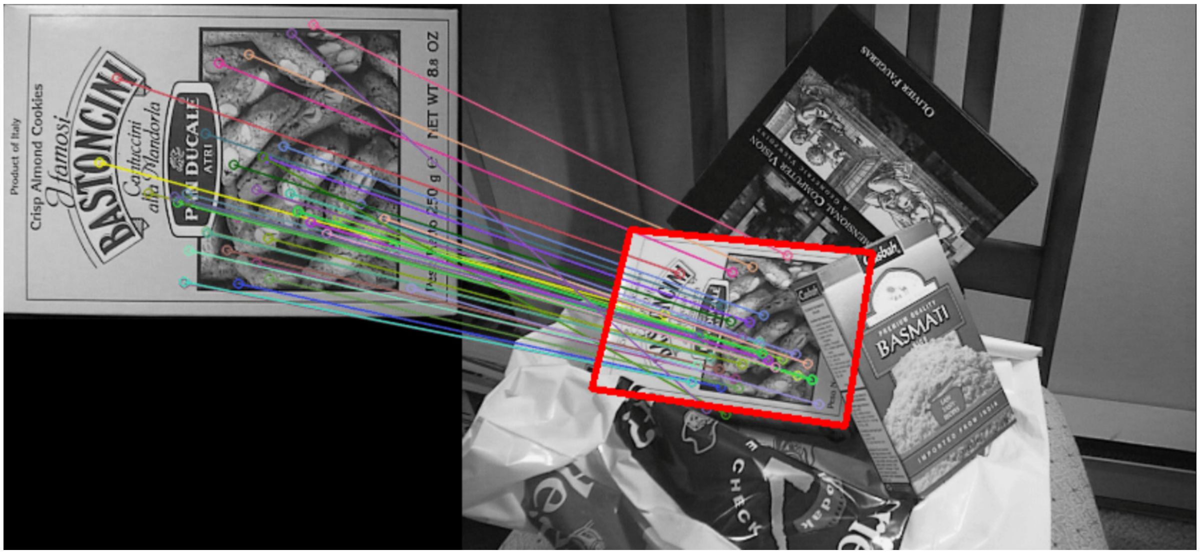


Figure 3.2: An example of feature matching with the ORB algorithm [47].

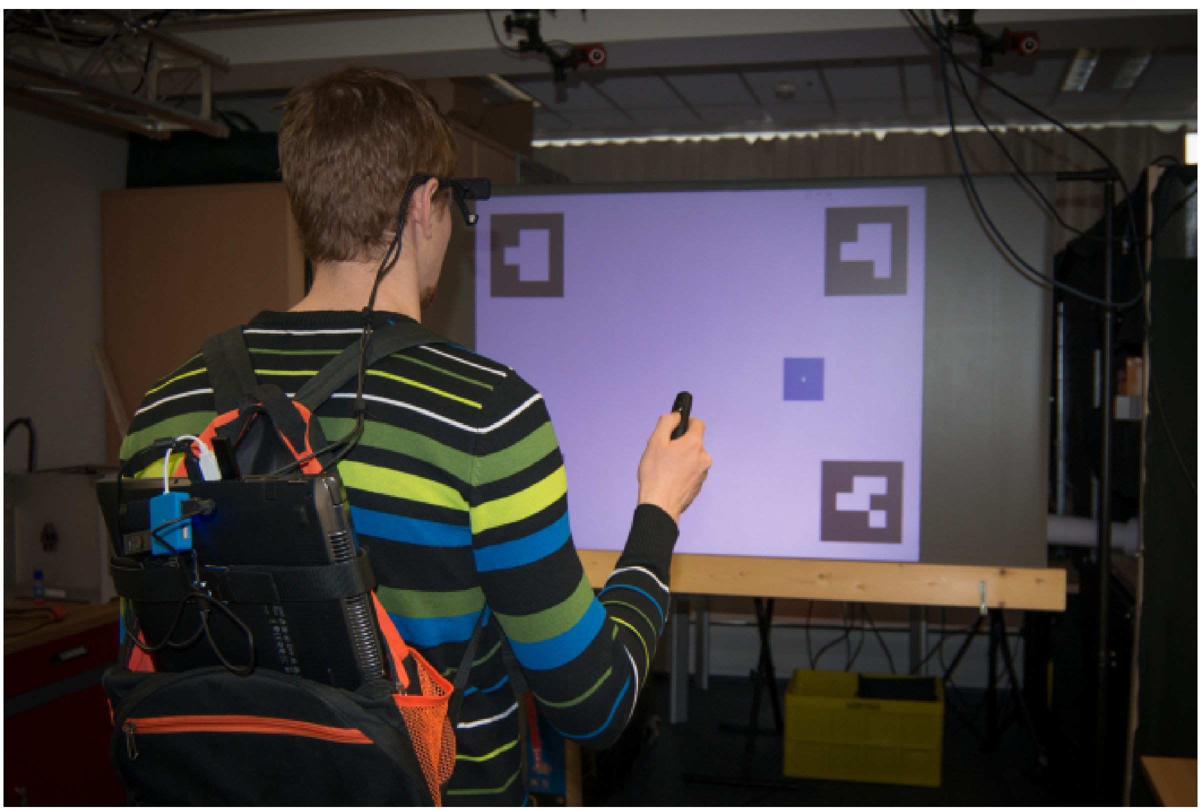


Figure 3.3: An example of detecting a screen through ArUco markers using a mobile eye tracker [5].

Collaborative environments where more than one person is working around multiple screens can also benefit from this method. By implementing our approach, researchers can design applications that focus on:

1. Modelling visual attention of users.
2. Visual transition of style or representation on the data-object according to user's attention.
3. Offering implicit gaze interactions.
4. Awareness and conflict in collaboration with eye-tracking technology.

3.1 Gaze Estimation and Calculation

For the eye-tracking software and hardware, Pupil Capture Software and Pupil Core Glasses were used [52]. The Pupil Core has two cameras: the world-view camera records the scene in front of the user and the eye camera is pointed inwards to capture eye movement. We chose this type of input because our long term goal is a collaborative environment with multiple eye-tracked users to produce dynamic interactions. The eye position (pupil) is detected in each camera frame, using the Pupil Lab 2D algorithm, and the gaze position in the world camera frame was estimated after a 5-point calibration. Both the world camera frame and the gaze position are then sent to an open port that our software is listening on. The system we produced can detect both big (wall-sized) and small (desktop) screens and is capable of recognizing which screen the user is focusing their gaze on when multiple displays are present.

3.2 Candidate Screen Detection

In order to transform the gaze point from coordinates in the world view image to coordinates relative to the target visualization, the contour of the correct screen needs to be detected. In Figure 3.1, image filters from OpenCV are used to pre-process each frame from the world camera with the following steps: (1) Reduce video frame noise and detail with Gaussian Blur filter (kernel size 3×3). (2) Detect a wide range of edges in the video frame with the Canny Feature Detection (see Figure 3.4). To counter various lighting environments, Canny Feature Detection's threshold is adjustable when the system is working (in our case, with 3 as weight factor, and a threshold from 50 to 80 performs best). Here, the Canny Feature Detection does not only ignore the boundary of small objects but also keeps edges of screens. (3) Find contours (a list of arrays that store coordinates of an edge). (4) Remove unqualified contours. Contours with

Figure 3.4: A video frame with Canny Detection. The satisfied contours are highlighted with coloured boundaries.

less than 3 edges and over 5 edges are discarded in this step since our target is 4-edged screens. Contours with 3 or 5 edges are corrected to 4 edges with the `approxPolyDP` function from OpenCV. If there is only one contour left from the pre-processing, the contour will be recognized as the target screen and directly passed to calculate the homography matrix described in the Section 3.5. Otherwise, the detected contours (see Figure 3.5) are then passed to the next step. In addition, small contours with an area falling under a threshold percentage of the full world-camera video frame (currently 10%) are discarded as they are likely too small for the user to read.

3.3 ORB Feature Detection

The ORB algorithm is based on FAST (Features from Accelerated Segment Test) and BRIEF (Binary Robust Independent Elementary Features). It increased their scale invariance and rotation invariance — the image's scale and rotation change do not affect the detection result in ORB. FAST is used to detect key points (a pixel with surrounding pixels which satisfy a specific condition) of an image or a video frame. Normally, key points are centred around blobs, edges, and prominent corners. After FAST selecting the key points, BRIEF is needed to generate a descriptor for each key point. A descriptor is a binary code of constant length, which stores some characteristics about the keypoint. In this thesis, descriptors are the way to compare the

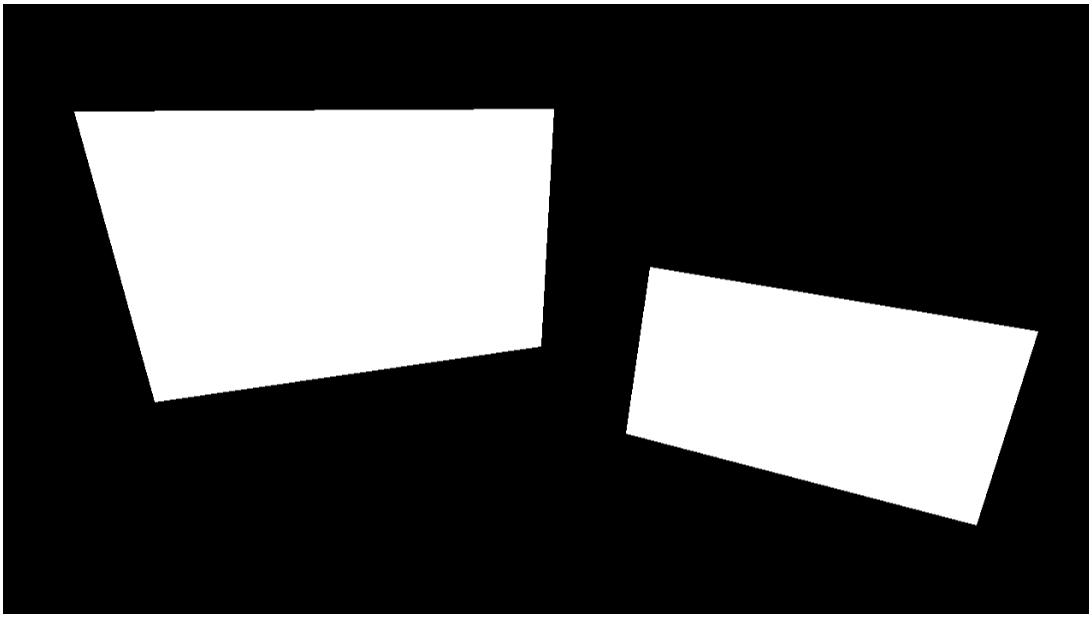


Figure 3.5: A black and white image of detected contours. White spaces show the space of the screens in this video frame.

key points between the front camera video frame and target screen frame for calculating the homography. Figure 3.6 is an example: here $(x1, y1)$ and $(x2, y2)$ can be matched by comparing their descriptors since the descriptors are equivalent to each other and unique from the image they belong to. In computer graphics, the nature of graphics transformation is matrix multiplication. The screen's content did not change from the left book to the right book, only the scale and rotation changed. The left image pixel coordinates are multiplied by a matrix (H) to transform into the right book's pixel coordinates. Thus, the matrix H is the homography. We will discuss more about homography in Section 3.6.

It has been shown in previous implementations that the ORB algorithm works well when being used with photographs [59], but when the algorithm was tested with sparse visualizations such as scatterplots, the success rate dropped dramatically. The original algorithm suffers from an abundance of mismatches between the video frame and the actual screen when using information visualizations (see Figure 3.8). Those mismatches are not only detrimental to the accuracy of the gaze point transformation but also slow the speed of the software. In response to this, we designed a new method that uses ORB as a starting point. One of the main drawbacks of the ORB algorithm in this instance is that the system has a hard time deciphering images that have a lot of uniformity (e.g. blank backgrounds). This is compounded when visualizations are on the screen with many items that are not differentiated by size and colour. To address this issue, our method is designed to restrict ORB detection to the contents of candidate contours

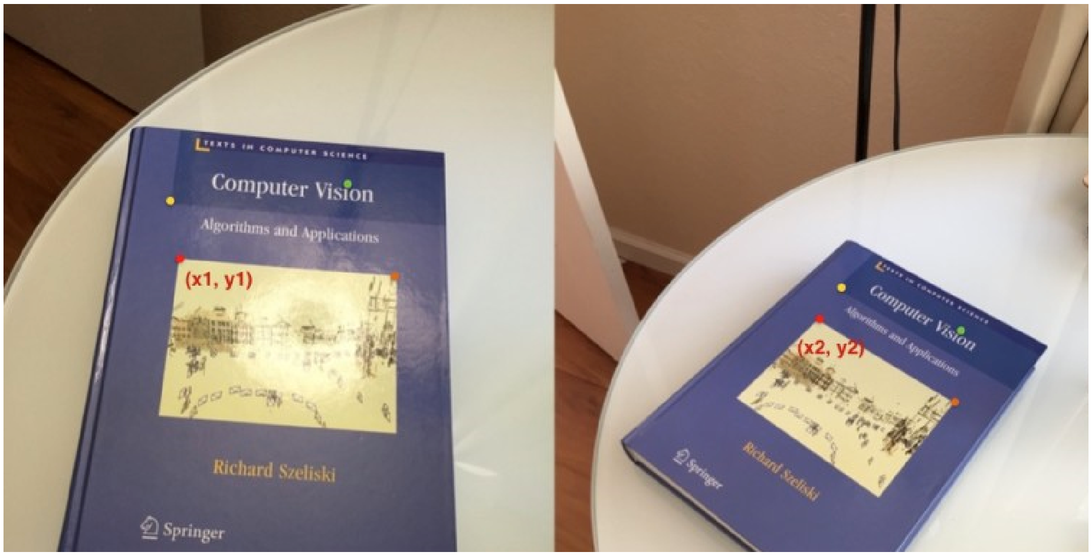


Figure 3.6: Same book in two images are related by homography [64]. Point *(x1,y1)*, *(x2,y2)* and other matched points are annotated by the same colour.

only, reducing the possibility of spurious feature matches in the world camera video stream. This combination of computer vision and ORB has shown great potential for eye tracking and sparse visualizations.

Consider two scenarios. The first scenario is data analysis with only one screen. In this case there is no need to use ORB since there is only one contour from the last step. The only detected contour is recognized as the correct screen. The second scenario is there are multiple contours were detected from the previous step. Our method detects key points only from inside contours one by one, drastically reducing mismatches.

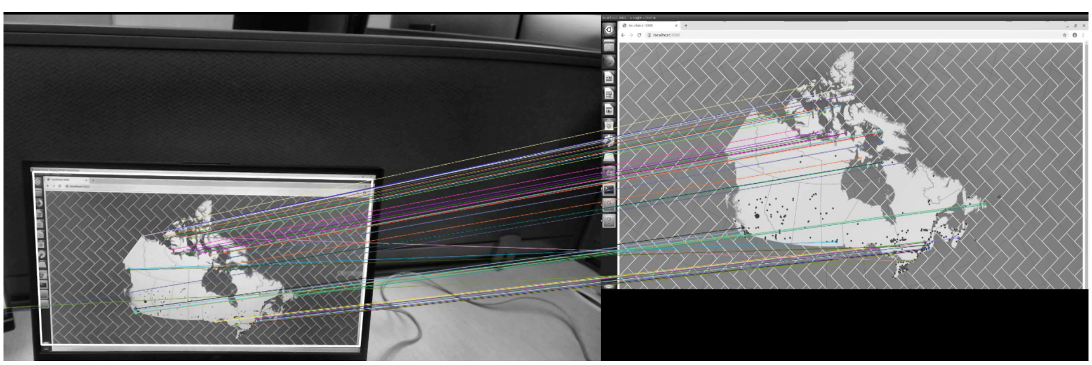


Figure 3.7: A successful example of ORB feature detection and matching with a textured background visualization. Showing a sample of matched keypoints.

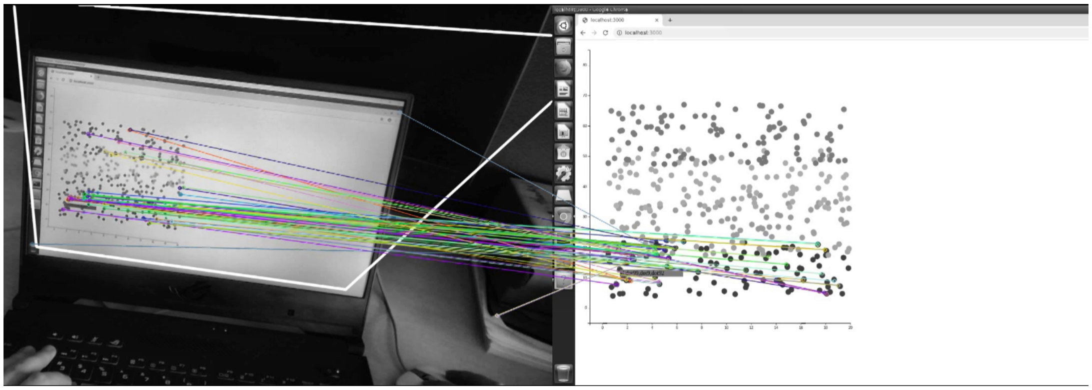


Figure 3.8: A failure example ORB feature detection and matching with a large white space visualization. Note the detected screen contour (white polygon) is incorrect.

3.4 Reference Contour Comparison

Computational cost is another problem with the original ORB algorithm, incurring latency in interactive scenarios. To reduce this processing overhead, we have devised a way to reduce the invocations of ORB. Our method compares the previous video frame's output contour to the current frame's candidate contours. If a candidate contour's central point is inside the last frame's output contour, it is selected for the homography transformation and ORB detection is skipped. Otherwise, the ORB feature detection is used to match candidate contours to detect the correct screen.

To prevent a compounding impact of an erroneous detection, ORB feature matching is run every 10 frames irrespective of the reference contour comparison. ORB matching overrides the reference contour matching when it runs. In testing, this was found to balance accuracy against computational demand. Our system forwards the contour boundaries of the detected screen in the frame to the next step in our pipeline, which will not only find the edges of the screens but also find the correct screen in the frame.

3.5 Gaze Point Transformation

Since the user's gaze point is relative to the world camera video, homography matrix [95] was used to transform the user's gaze location from the 2D world camera video plane to the 2D screen plane given the corners of the screen contour in the frame. Given the screen-relative gaze point, it is possible to identify which window the user is looking at on the correct screen in multiscreen environments. Thus the gaze information can be passed to the correct program, such as visualizations or other applications, for use populating interest models, gaze-interaction

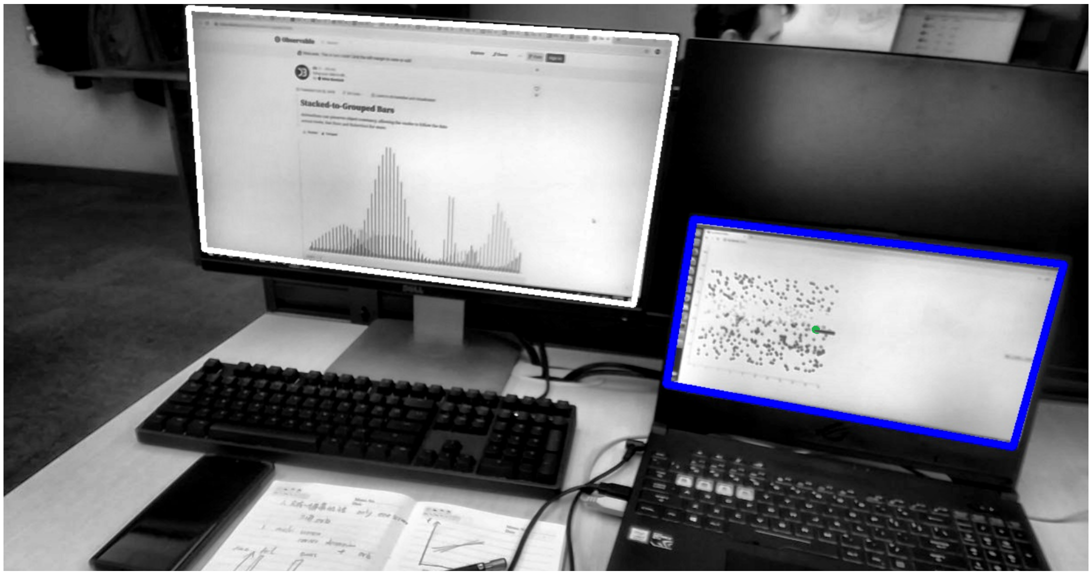


Figure 3.9: Our method is able to discern between multiple screens in space with low computational overhead and recognize complex visualizations with large amounts of white space (a challenge when using head-mounted eye tracking glasses with visualizations). The blue frame shows our screen detection, and the green dot shows the gaze point.

techniques, or other purposes. To process the real-time information a back end server was used to capture the time, the user's ID, and each individual user's gaze data.

3.6 Demo

Our demo, shown in Figure 3.9, is built on an HTML front end with JavaScript. The demo listens and collects the gaze data from our back end server. Each object in the web front end is assigned an event handler to check if the user's gaze is on it or not. When users are looking at one object, the data model records this action, how long, when, which part of the visualization, and which user is looking at it. Also, each object is given an attribute of which user currently "owns" each object. This attribute gives us a lot of flexibility to design interactions between multiple users.

3.7 Experiment

To analyze the performance of our method, an experiment was designed and run to test the original ORB algorithm against our improved version using on-screen information visualizations.

Table 3.1: Accuracy of screen detection by method and setup

	Dense Visualization		Sparse Visualization	
	Stable one screen	***Moving two screens***	***Stable one screen***	***Moving two screens***
ORB feature detection	87.1%	0%	13.6%	0%
Our method	99.7%	96.5%	96.9%	90.9%

The experiment varied three factors: (i) The content of the visualization on the screen (sparse visualization as in Figure 3.8 or dense visualization as in Figure 3.7). (ii) The screen detection method (our method and ORB feature detection). (iii) The viewing context (stable one screen or head and gaze moving slowly between two screens). In the two screen condition, the second screen showed a distractor bar chart visualization as shown in Figure 3.9. With the three factors, we have recorded eight videos with the pupil-core glasses in the lab with two monitors. The measure for the experiment was the count of frames in which the target screen bounds were detected correctly (all four corners of the target screen aligned correctly). This was assessed manually by the lead researcher viewing each frame and counting the correct detections.

For each of the four conditions (one/two screen $\times$ sparse/dense visualization) we captured raw video and eye-tracking data of a user looking at the target visualization on WSL (Windows Subsystem for Linux). For each test, 500 frames of data were captured. This number of frames was found to be sufficient to consistently differentiate methods. The output from the Pupil cameras was post-processed twice: by our pipeline and the ORB algorithm alone.

As seen in Table 3.1, our method outperformed the original ORB algorithm while not requiring ArUco markers or the need to add a textured background to break up the abundance of white space. In particular, when the user's gaze was moving between the two screens, the ORB algorithm alone could not tell the difference between the screen with the active information and the distractor.

Multiple resolutions were then tested with one and two users using our method to pinpoint the limits of the scalability of our approach. Starting without the reference contour check (RCC) (i.e., ORB on every frame), the results are shown in Figure 3.10. The performance decreased dramatically with resolutions above 1280/720 (time per frame increased). If the ORB output is checked every frame the entire approach is simply too slow. Based on these results the reference contour comparison was added. Because users are often keeping their heads relatively still while looking at visualizations, this consistency is relied on to skip ORB detection and increase the overall speed of the system in the arrange case.

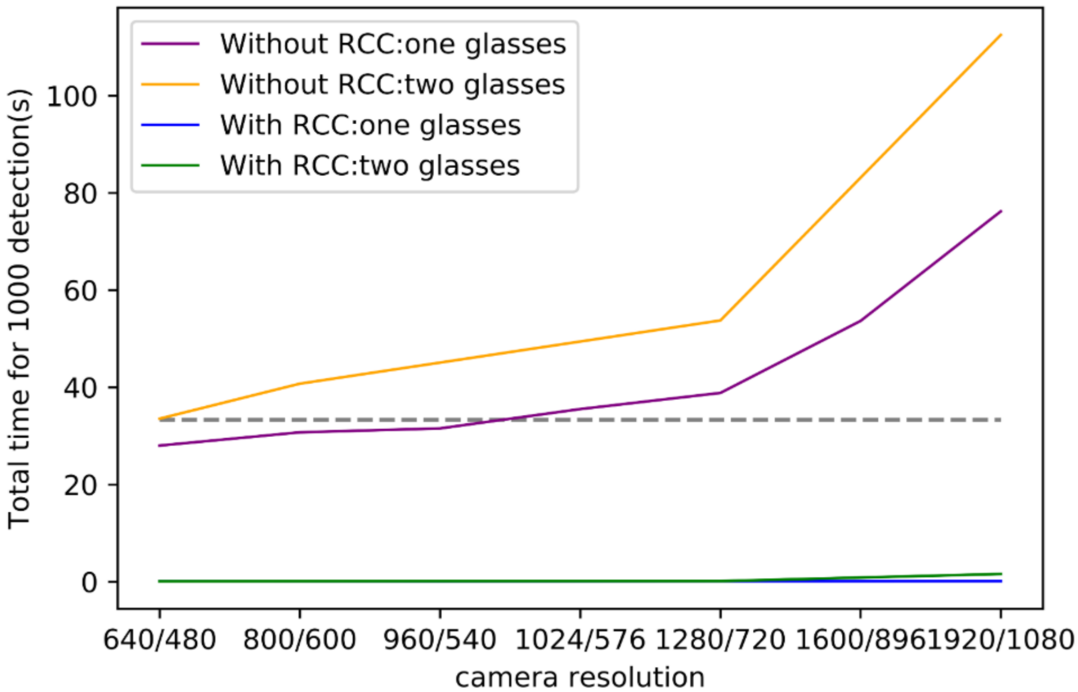


Figure 3.10: Timing results of our method with and without the reference contour check (RCC). The dotted gray line is the baseline for processing 30fps.

The speed increase is visible in Figure 3.10 where the blue and green lines (with RCC) are significantly lower than the orange and purple ones (without RCC). Overall by using our method which combines edge detection, ORB, and the heuristic RCC, efficiency and accuracy are improved over the ORB alone.

Although using ArUco markers does not require as many steps as our method and it is faster than our method, removing the extra display step is desirable for situations such as public visualizations where calibration is a barrier to engagement (deploy problem, occupy the screen of visualization). Markers can also be visually distracting or aesthetically displeasing. Also, our method can work perfectly even a screen is only partly visible to the front camera (see Figure 3.11).

3.8 Collaborative Environment

In our work, a visualization can be programmed to show different information to different users because the system knows which user is wearing which eye-tracking headset and where they are looking; this opens up new avenues for interaction based on individual user gaze.

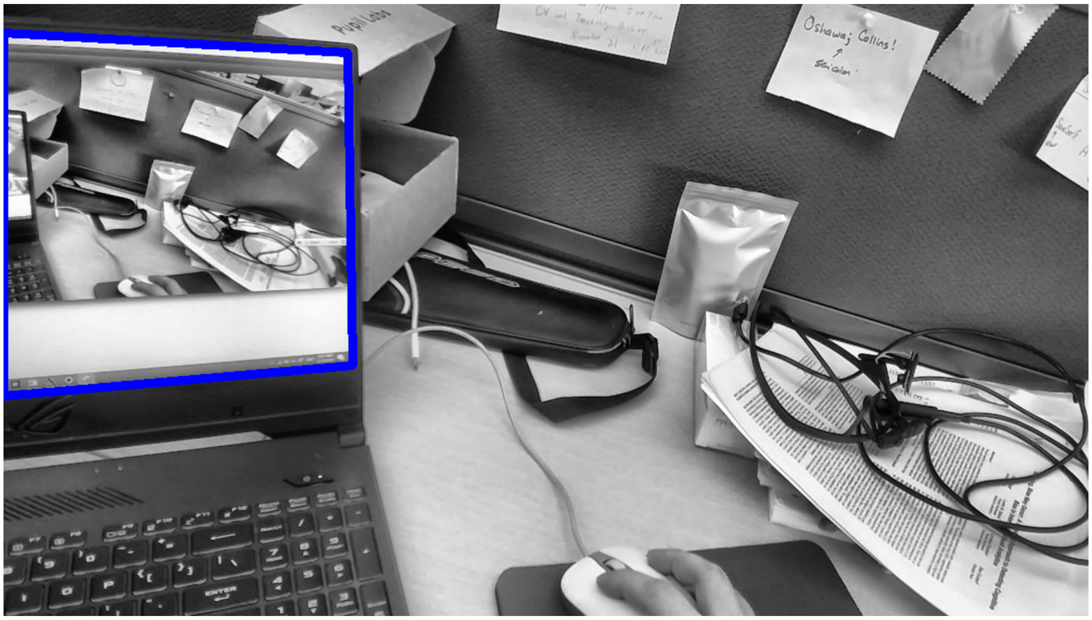


Figure 3.11: The half screen is detected by our method in blue rectangle.

Collaborative environments where more than one person is working around multiple screens can also benefit from this method. By implementing the approach, researchers can design applications that focus on:

1. Modelling visual attention of users.
2. Visual transition of style or representation on the data-object on the screen.
3. Offering implicit gaze interactions.
4. Awareness and conflict in collaboration with eye-tracking technology.

4 Gaze Data and Interest Model

Eye tracking studies in visualization research have provided insights into how people interpret and interact with visualizations. The analysis of gaze behavior provides information about the distribution of visual attention over time and visual strategies employed in interpreting a visualization or in working with a complex visual analytics system. Typical measures derived from gaze data are fixation duration, fixation count, saccade length, and numerous other aggregated values [42]. All measures can be indicators for specific perceptual or cognitive aspects (e.g., cognitive load [38], working memory [75]) that are potentially interesting for the assessment of a visualization.

In the beginning of eye-tracking analysis, gaze data were collected and interpreted as gaze-coordinates in the space of rendered visual stimuli. Analysts usually related these data to the semantic content of the stimuli offline by manually inspecting gaze heatmaps visually or defining an area of interest (AOI) [8]. For studies involving many subjects, long sessions, and interactive content, this process is inefficient. Researchers found that gaze coordinates can be related to rendered visual objects in real time and yield an account of which objects a user views at any given time since the content's layout is known when being generated [2]. The output is a list of granular data-objects users viewed at any time in an experiment (e.g., individual nodes in a network, 3D objects in a scene) and refers to these objects as data-of-interest (DOI). As such, the DOI approach can capture users' data interests from interactive visualizations over long periods of time. Moreover, DOIs are characterized by a rich set of attributes derived from the data that the DOIs are defined on, and from the visual context in which they are displayed [48].

Modelling visual attention has been an important area of research in computational modelling. Current attention models generally fall into two main categories: *bottom-up approaches* and *top-down approaches*. Bottom-up attention models (stimulus driven) are based on the low-level features of the visual scene, while top-down models (goal driven) are determined by phenomenons such task, goals, experience and knowledge. It is found that user's attention is usually attracted by a salient visual feature due to the type of the visualizations or the goal of tasks in the studies [33]. The salient feature can be the shape, colour, label and location of rendered visual objects on a chart or a trend (increase, decrease, peak, bottom), a region or specific values in visualization data.

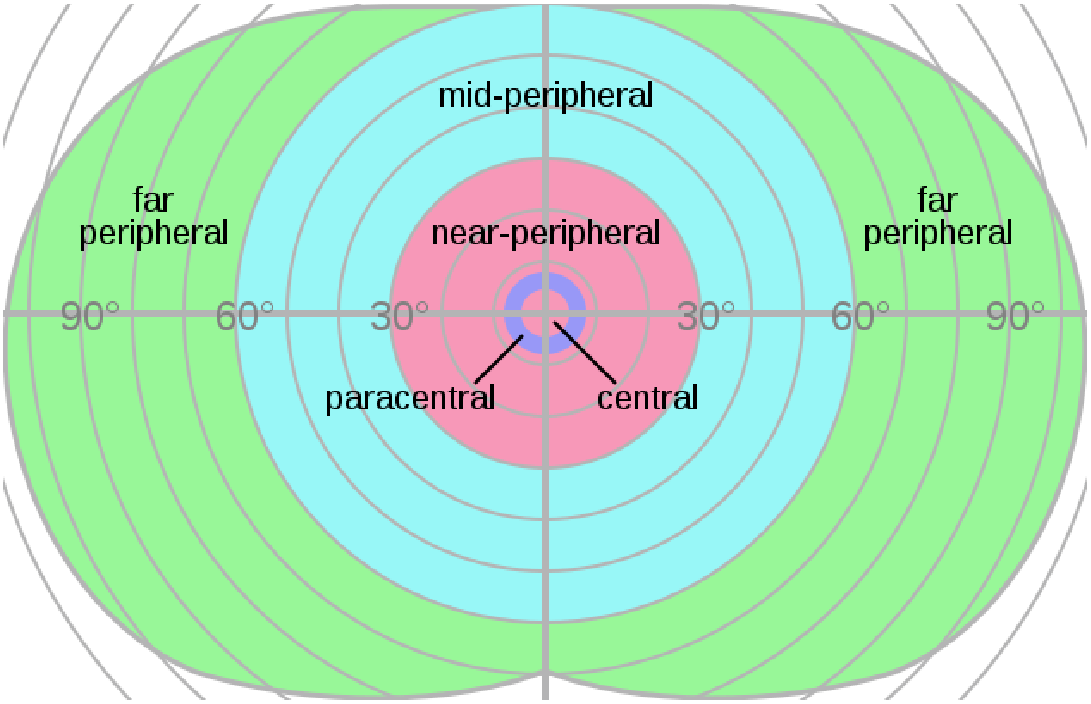


Figure 4.1: Field of view of human eye [105].

In this chapter, we propose a top-down approach that divides DOI into groups by one salient visual feature for modelling and detecting user's attention on groups of DOI during visual data exploration. This allows us to study further the user's cognitive state (especially the interestedness in the groups) and insight into visual stimuli. We refer to these groups as group-of-interests(GOI). Another reason why choosing the top-down approach is that humans are born to use gaze implicitly and this will be discussed in next chapter. Based on the top-down approach, a model is designed that uses GOIs as input, calculates scores for GOIs with a customize weight formula and predicts future attending targets or regions in complex and interactive visualization. We will demonstrate how this model can be used to improve the readability and usability of the visualization by designing an implicit interactive system which can offer appropriate support or guidance to users in the next chapter.

4.1 Gaze Action

The definition of a gaze action is as follows: In a period of time, a discrete gaze movement from one position to another above a threshold speed. The time interval is related to the frequency of eye-tracker camera and the person's eyes movement. The data provided by the eye-tracker

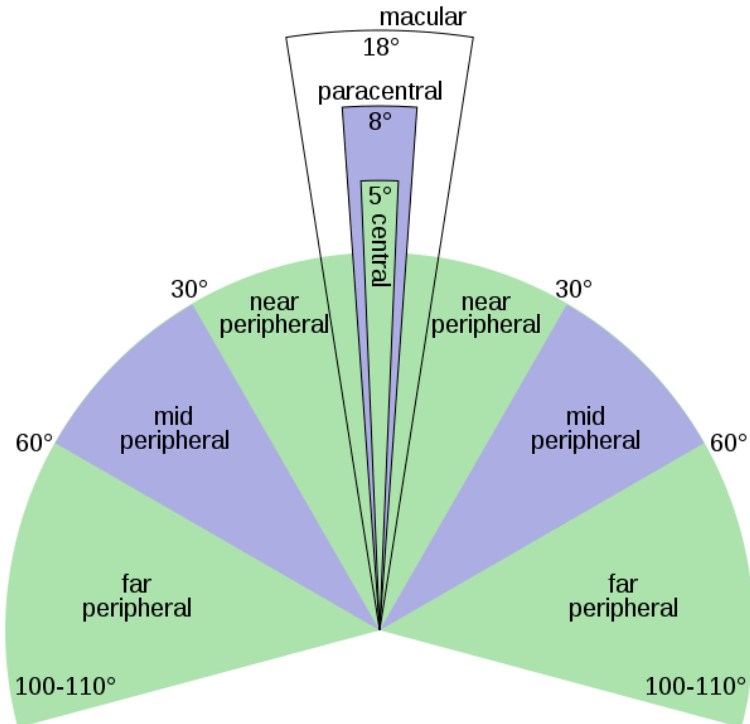


Figure 4.2: Peripheral vision of the human eye [106].

and our pipeline is a series of continuous x and y coordinates of the gaze location on the visual stimuli, and the rate of the gaze input depends on the recording frequency of eye-tracker camera. In the interest model, there is a listener constantly monitoring the coordinates of the gaze point. For every new (x, y) gaze location input, the speed of the gaze movement (by degrees per second) is calculated by measuring the moving distance and time interval between the last and new gaze input and calculating their quotient. A minimum threshold distance is needed to register a gaze action since the gaze data from the eye-tracker is unstable and varies slightly during the detection. If there is no limitation for the distance, many repetitive gaze actions will be counted and will influence user interest prediction accuracy.

Nevertheless, the model cannot examine how many visual rendered objects a user can see with only a gaze location. A circular region should be set up to represent the user's viewing region on the screen. A person with normal vision can see objects within a field of about 180 degrees with both eyes when looking straight ahead [86], but much of this is peripheral. The *central 5* degrees seen by both eyes is called *central vision*, and it is *central 8* degrees for *paracentral vision*. It is also known as "seeing" vision (see Figure 4.1 and Figure 4.2), because it is the vision you use to look directly at something. To improve the interest model's performance, the model used 5 degrees as a parameter to compute a region of the central vision that

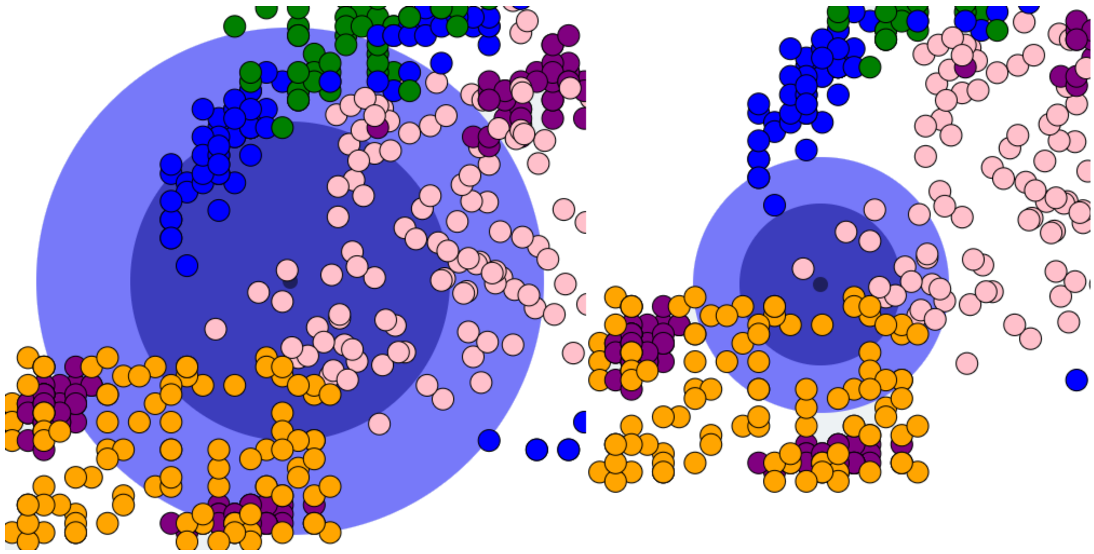


Figure 4.3: Region of central and paracentral vision on the screen. The left figure shows the region of visions when distance *d* equals 60 cm, and the *d* for the right picture is 30 cm. On both figures, the semi-transparent light blue circle is the region of paracentral vision. It is semi-transparent deep blue for central vision. The black point inside the two regions is the gaze point.

changes based on the user's distance to the screen. From the definition of central vision, four components are needed for calculating user's central vision *r* (in pixels) on the screen:

(1) distance (*d*) between the user and the screen, (2) size of the screen, (3) resolution of the screen, (4) angular size of the central vision.

With the *d*, angle central vision (α) and tangent function, we can obtain the radius of the central or paracentral vision in centimetres on the screen (see Figure 4.3). Since the graphic on the screen is essentially drawn in pixels, the size and resolution of the screen are needed to transfer the radius from centimetres into pixels. In Formula 4.1, the radius is converted from centimetres to inches by multiplying 0.393701. To convert inches to pixels, PPI (pixels per inch) of the screen is needed. In Formula 4.2, PPI is calculated by the size and resolution of the screen. d_i is the length of the screen diagonal by inches. w_p and h_p are the horizontal and vertical resolution of the screen. Then, the interest model is able to search and check rendered visual objects inside the central vision for each gaze action. The gaze action's speed can transfer from pixels per second to centimetres per second in the same formula but in the opposite way.

$$r = d \times \tan(\alpha) \times 0.393701 \times PPI \tag{4.1}$$

$$PPI = \sqrt{\frac{w_p^2 + h_p^2}{d_i}} \tag{4.2}$$

Gaze actions can be mainly classified into three types [61]. First, the slow period when the gaze is more or less still and visual information taken in is referred to as a *fixation*, which is characterized by low positional dispersion, low velocity, and a duration of about 200–300 ms. Second, when the gaze is shifting from one position to another, the action is referred to as a *saccade*, which is a very rapid movement with typical velocities ranging from 30 to 500 deg/s and durations ranging from 30 to 80 ms. Last, when the observed objects are moving, e.g., when watching a dynamic scene, other gaze actions may occur that are related to the movement in the scene. One such eye movement is the *smooth pursuit*. Knowing the type of gaze action can benefit visualization researchers when designing interactive systems. These actions inspire different strategies to make the system we built more efficient and robust.

4.2 Object Selection

After successfully detecting the gaze actions, a selection algorithm is needed for the interest model to pick up the visually rendered objects that the user can see within the central vision region for each gaze action. The interest model uses a customized selection algorithm for the gaze to select the objects that the user is able to see. It is worth noting that the algorithm will neglect objects that are occluded over a threshold percent and partly or hardly visible to the user. There are two circumstances for the selection algorithm: (1) If one or more objects are detected inside the user's central vision in a gaze action, all the visible objects (not occluded) are selected as the targets that the user saw during that gaze action (see Figure 4.4). (2) Similar to the bubble cursor selector [39], for a gaze action with no objects inside the user central vision, the algorithm will try to find the object with the shortest Euclidean distance to the gaze point. The reason for handling the second scenario this way is due to the inevitable minor offset of the gaze data provided by the eye-tracker or the pipeline. With the selection algorithm, the interest model can still able to select objects with deviated gaze data in some situations. On the other hand, a threshold is set to prevent the algorithm from selecting objects with unlimited distance. The speed of the gaze action also plays an important role in the selection algorithm. Gaze selection only works when the speed of the gaze actions is lower than 30 deg/s (e.g. a fixation). For gaze actions over 30 deg/s, the interest model will discard the saccade actions when calculating users' preference on visual content since humans cannot attend to detailed information near the gaze point while moving their eyes.

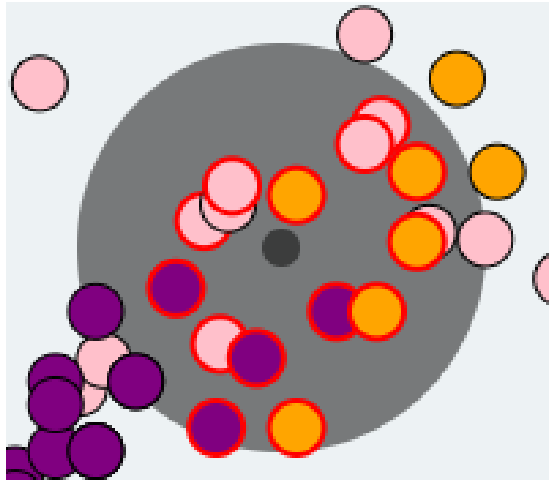


Figure 4.4: Visual objects selected (surround by red edge) within central vision by gaze (grey circle). Occluded objects or objects outside the region have a black edge.

4.3 Interest Score Formula

In order to investigate gaze actions and the degree of interest (DOI) for the groups of visual objects in a data visualization study, we created a mathematical formula to calculate an interest score.

$$S_{ij} = \frac{n_i}{n_{all}} \times (log(n_i) + 1) \times \sqrt{\Delta t} \times \frac{1}{d_j + 1} \tag{4.3}$$

As Formula 4.3 shows, four components contribute to the interest score (S_{ij}) of one gaze action. In the formula, *i* is the index of a group with visited objects that have the same salient feature (e.g. visited dots with same colour). n_i and n_{all} means the number of objects from group *i* and the number of all objects visited by that single gaze action. The n_i divided by n_{all} is the proportion of group *i* to total visited objects. For n_i, a group *i* with more visited objects should receives a higher interest score. On the other hand, the increment of the interest score should be flattened when n_i increases significantly since human one-time memory is limited. Hence, the logarithm is used for n_i*+1* to reduce the growth of the score and avoid a result of zero when n_i equals 1. For the duration of the gaze action, square root is applied on the delta time *t* to reduce the weight contribution.

The interest score serves as the metric to represent the user's interest level in a object group with the same salient visual feature (e.g. points of the same colour). For a gaze action, the model will calculate an interest score for each group, which considers the proportion of a group to the total, the duration of the gaze action, the number of visited objects, the distance between the objects and the gaze point.

The formula also takes the distance d between the gaze point and the readable object into consideration since the distance is inverse to the interestedness of the user. People are naturally more interested and focused on the center of the point they are looking at. The shorter the distance, the more user is concentrated on the information provided by the objects. However, group interest score is calculated, and the distance is varied for every single object inside the user's central vision. We offer a solution to this by choosing the smallest distance of a visible object *j* in the group. As d_j decrease, 1 divided by d_j + 1 will increases and converge on 1 when d_j equals to 0.

4.4 Interest Score Calculation

The calculation of the interest scores for group-of-interest (GOI) is the core module of the interest model. For a gaze action, the bubble-cursor-like algorithm selects a list of objects viewed by the user, which are the data-of-interest (DOI)s. Then they are separated into groups by a predefined salient feature (e.g. colour), and the interest model calculates the group interest scores respectively. Next, these scores are stored in a map object and put into an array where a sliding window counts total interest scores and generates a summary of user's interest every 10 gaze actions. In Figure 4.5, the red bracket indicates a sliding window with a length of 10 gaze actions. By adding up all 10 gaze action interest scores by groups in the sliding window, the interest model compares each group's total score and selects the highest group as the user's current interest group. After obtaining the score at *Time 1* and 3 or more subsequent gaze actions were detected, the sliding window shifted to *Time 2* with a *Moving Step* and found the next interest group of the user. To achieve an accurate result and avoid switching the result too frequently, the interest model only changes when the group with the highest score is dominated (50% higher than other groups) in sliding window counting.

The frequency of counting the interest scores (the moving step of sliding window) for GOI is a question worth discussing. The higher the frequency is, the more the GOI is related to the user's immediate and short term interest in current visual stimuli. On the contrary, if the frequency is set relatively low, the interest scores will represent the user's longer GOI. The wise way is to integrate the frequency with the use cases or the purpose of the tasks and then decide the best solution. With the interest model's result, the door to understanding the user's interest and reading experience about the visualized data and designing complex as well as natural gaze interaction has opened. In the next chapter, two applications are introduced to evaluate the usability and accuracy of the interest model.

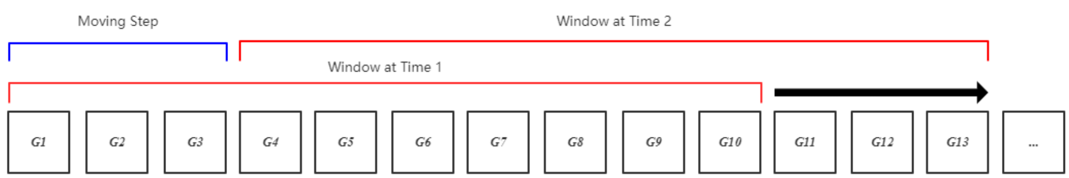


Figure 4.5: A sliding window (in red bracket) with the length of ten gaze actions. After being full and 3 or more succeeding gaze actions input to the interest model. The sliding window shifts with a moving step (blue bracket).

5 Implicit Gaze Interaction

Gaze is widely used in psychology, cognitive science and visualization. In many studies [23, 27, 84], gaze worked as a direct control modality to work with interactive systems and fulfill task goals. A person's gaze point can be used in a variety of ways to control user interfaces, alone or in combination with other input modalities, such as a mouse, keyboard, sensors, or other devices. A major field within gaze interaction research is to find more efficient and useful ways to facilitate human-computer interaction and explore novel user interfaces. However, gaze interaction within complex systems sometimes can be inefficient and error-prone [34]. This makes users to gaze intentionally at items, which goes against innate way of using our eyes. Examples can be found in recent research where gaze points and gaze gestures are used to type or control a complicated user interface [57, 63, 71]. Before clarifying the problem in depth and exploring an alternative path for gaze interaction design research, it is necessary to understand the nature of human gaze behaviour and the difference between explicit interaction and implicit interaction.

In most gaze interaction experiments, participants are asked to gaze directly and continuously to fulfill a series of interactive tasks. This purposeful and attention-demanding way of engaging with computers is referred to as *explicit interaction*, where the appropriateness depends on the assumption that the user has conducted an action to achieve a specific effect intentionally. Using gaze explicitly is conflicted with our information retrieval process since unaware and unintentional fixations can interfere with a system's judgment about the intention of users. Hence, users were asked to gaze steadily at things over a few seconds to reduce unintentional fixation registrations and ensure their intention is successfully understood by the system [57, 63, 71]. Long-held gazes can cause both mental discomfort and physical exhaustion for users, which is known as gaze fatigue [76], and thus their use is a bottleneck for designing advanced gaze interaction.

Previously, it was thought that when the eye observed something, the mind also thought about ("retrieved") that object. This is called the "eye-mind hypothesis" [50]. However, more recent research has challenged this, revealing that eye movements say nothing about the underlying cognitive retrieval process, as the process controlling the switch in gazes is independent of the process controlling retrieval [4]. In other words, we cannot assume that someone is pay-

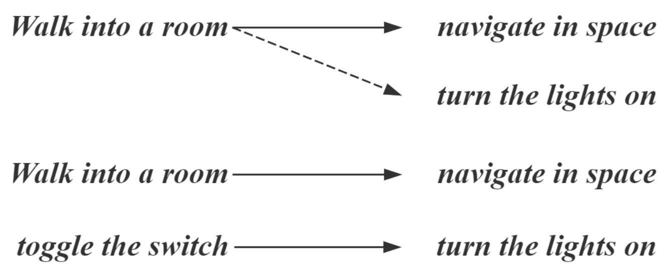


Figure 5.1: Depiction of implicit (above) and explicit (below) ways of turning on the lights. The dashed line shows an input–effect pair that is expected to be implicit. A common pattern in implicit interactions is the co-occurrence (or bundling) of multiple effects as a result of the same input, eliminating the need for additional actions. (Figure from the work of Serim and Jacucci [88].)

ing attention to something just by knowing what they are looking at. There are several types of gaze fixations. Fixations can be classified as an unaware (unconscious), such as randomly gazing while thinking. They can be visually motivated (unintentional) fixations, such as looking at something to see it, or interaction motivated (intentional) fixations, such as looking at something to activate or select it.

In the last two decades, there is an increasing interest in exploring interactions that differ from traditional explicit interactions. A term that is often used to represent these new types of user engagements is implicit interaction, defined as the "user's attitude towards an input–effect relationship in which the appropriateness of a system response to the user input (i.e., an effect) does not rely on the user having conducted the input to intentionally achieve it" [88]. In other words, an action performed by the user that is not aimed to interact with a system for achieving a specific result but such a system can understand the input and able to respond with effects. For implicit interaction, appropriateness of a particular effect is instead understood from the user input, but does not rely on the user's intentionality. An *input* refers to any kind of data that originates from the user and available to the system. An *effect* refers to any outcome that is facilitated as a result of this user's action or data that happened either with or without system mediation. For example, walking into a room facilitates navigation in space, but can also cause the light turn on in the presence of a motion sensor.

Implicit interactions often suppose the existence of a primary and intentional activity. Observing through the lens of multiple *input–effect* pairs, situations are translated in which an input leads to multiple effects. Some of these effects are intended by the user and can explain why the user has conducted the action in the first place. Other effects can be unintentional but still appropriate for a given situation. The expected benefits of implicit interactions can be attributed to the decreased user effort that is achieved through this bundling (Figure 5.2) instead of an effect being unintentional. Considering gaze cannot always represent a user's

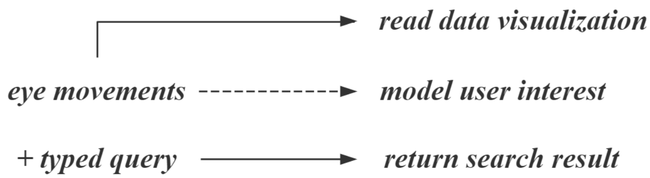


Figure 5.2: Diagram showing a gaze interaction that is designed to be implicit (dashed lines) within the context of other interactions. (Figure from [88].)

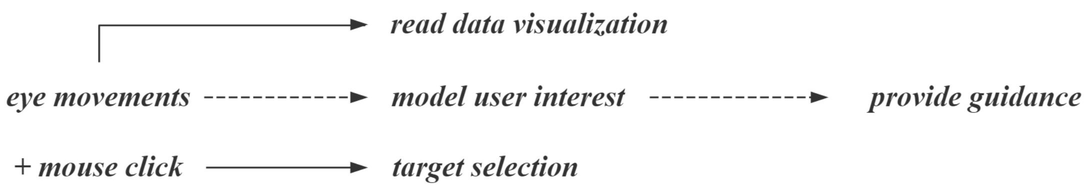


Figure 5.3: Implicit gaze interaction (dashed lines) design in our work. An interest model is built based on a user's gaze input. Our interactive system detects the user's shift in interest and provides visual guidance or, combined with a mouse click, and give invisible support to enhance target selection.

attention and the gaze fixation often being unintentional (visually motivated) or unconscious during information exploration, designing interactions that use gaze as *implicit input* should be a more reasonable choice, which can provide a better interactive experience to users with the combination of other interactions (see Figure 5.2).

The premise of this implicit gaze interaction is that the system needs to collect adequate information from continuous gaze data for modeling the user's attention and analyzing the intention behind it. To better understand the user's cognitive state and the attention of interest to an information visualization, previous chapter has introduced a gaze-data based interest model to analyze a user's DOI (data-of-interest) and predict the preference for visual content by calculating interest scores and comparing the GOIs (groups of interest). In the following sections, several implicit interaction designs for information visualization are presented to interact with gaze input. With the scores of group-of-interest resulting from the interest model and indicating the degree of user's attention on visual features, the goal of our designs is to improve the usability and readability of the visualizations by responding with effects in the form of visual guidance or non-distracting support, which allow users to interact more naturally and comfortably with the visualization during the information discovery.

5.1 Guidance in Visual Analytics

In the process of visual analytics supporting information discovery, users that are typically experts in their domain but novices to visual analytics, are not always able to fully follow the information delivered by a data visualization and complete analysis, due to the complexity of tasks and style of the visualization. What parameters should be set for analytical computations when suitable values are not clear upfront? How should analysts interpret visual representations of complex phenomena rather than plain information graphics? How to make proper progress in terms of many things to control during the data analysis process? Guidance can benefit these processes to help users gain a better insight into the data and narrow the gap that hinders the effective continuation of the data analysis [15]. Major modalities of providing visual assistance include textual or visual channels such as size, font, colour, highlighting, and animation, which can provide different levels of salience depending on which type of visual signal is applied. However, it is found that guidance systems can sometimes be harmful to the users [22]. For example, previous mixed-initiative systems will provide unnecessary help to users, which could be overly interruptive and distracting. Besides, providing inappropriate assistance that does not suit users' needs and context may be disappointing [56]. Additionally, it could create biases for analysts and lead the analysis down to an unhelpful, unimportant, or misleading path which will cause them lose objectivity. For example, recommendation algorithms could provide high similarity information to an analyst and keep him stuck in the same kinds of information group with one-sided or early conclusions, decreasing the probability to reveal new data.

To prevent the guidance from being distracting and annoying to users, in the following section we present an implicit gaze interaction design that supports visual analytics in a distinct way by conducting subtle visual transitions (system's response) outside the central vision. Since gaze input is a reflection of users' intention, offering non-distracting guidance or invisible support allows users to fully immerse themselves in the visual exploration without being interrupted or spending time on confusing guidance. In the following section, our design is applied to solving real-world problems by providing non-distracting support. To demonstrate the effectiveness of this idea, two demos were built to solve real-world problems, which are described in Section 5.2 and Section 5.3. The scenario of the first application is high density and overplotted information visualizations such as scatter plots and maps. In the prototype, the interest model generates recommendations with different levels of visual transition to improve the usability and readability of scatter plots. The second application uses implicit gaze interaction to improve the experience with existing interactive controls, especially for target selection. A

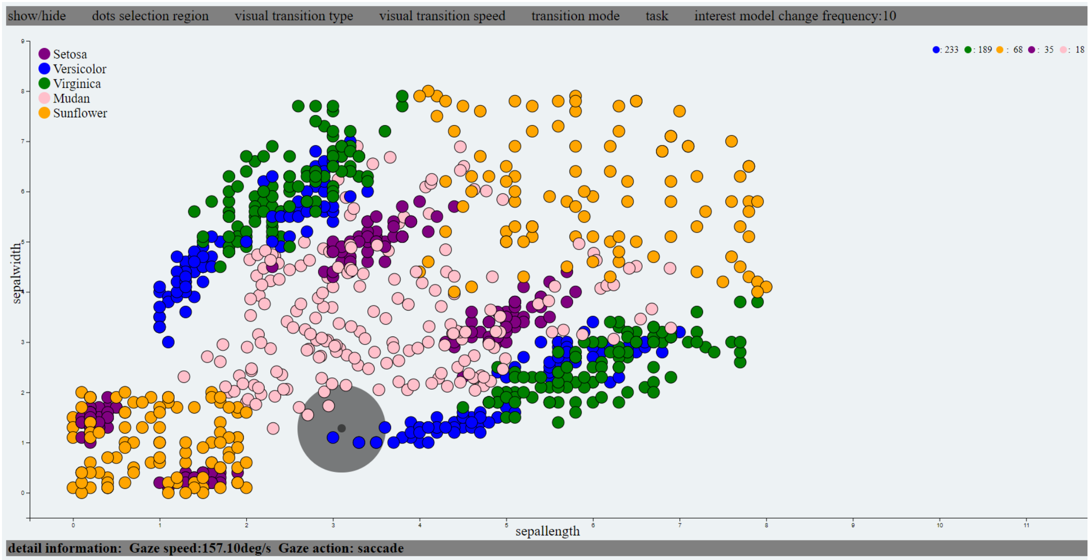


Figure 5.4: The interface of the overplotting scatter plot demo. On the top is the menu bar for various settings. On the bottom is real-time gaze state information (speed and type). On the right side below the menu bar is the interest scores for different colour groups.

study design is also purposed to evaluate our design compared to existing input device such as mouse, touch screen, and combined with mouse-based interaction in Chapter 6.

5.2 Overplotting in Visualization

Overplotting is one of the most classic issues in data visualization [7]. It occurs when the dataset is larger than the available visual space can accommodate. For example, the dots of a cluttered scatter plot will tend to overlap, making the graphic hard for users to read. Visualization scientists have attempted to solve this issue with various methods such as: decreasing dot size, using transparent dots, sampling only a fraction of the data, providing interactive zoom, and so on. These methods increase the readability of overplotting by making each dot more clear, but it also reduces the readability in other aspects. For example, it is hard to recognize a semi-transparent dot's colour in a high-density area of overplotting [67]. Decreasing the size of the dots may cause some dots to be barely visible to users, or make overlapping dots in the high-density area even harder to find.

Although the trouble of overplotting cannot be completely eliminated, it is still possible to reduce the complexity of overplotting by designing interactions so that users can proactively adjust the style of visualizations as they want, such as highlighting a specific group of data to bring it to the forefront. For example, scatter plot points are often associated with a categor-

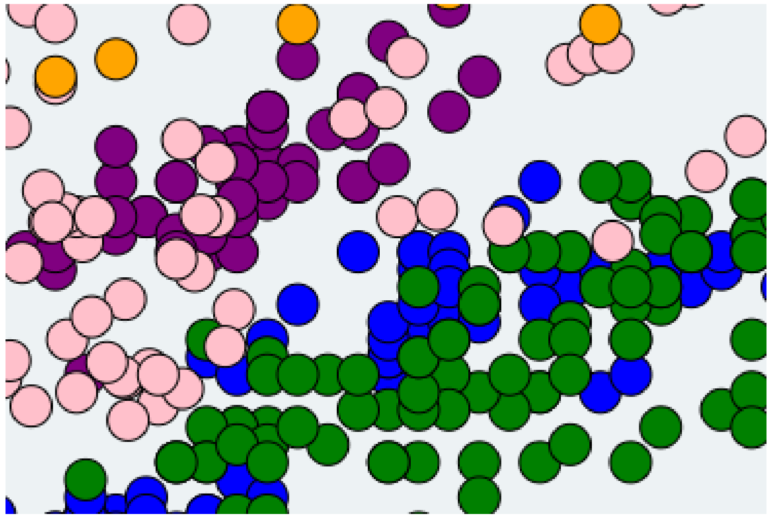


Figure 5.5: Overlapping objects of the scatter plot where certain dots behind are hard for users to identify.

ical data dimension (country, flower, animal) which can be used to create groups which can be interactively isolated. This categorical dimension is often encoded with a salient visual feature (colour, shape, size). A users' reading sequence is not only decided by the spatial (x,y) coordinate but also driven by the viewer's task, goals, and prior experience, which is also of tremendous importance in determining where a person will look in an image or a visualization [68].

This requires users to observe complicated user interfaces and learn how to use the interaction tools provided by the visualizations before or during the exploration. The preparation for visualization interactions sometimes can be overwhelming and frustrating to a visualization beginner or even an expert [11]. To leave out this time-consuming step and help users focus more on the visualizations, we contribute an implicit gaze interactive system that brings groups of points to the forefront of over-plotted scatter plots. The way the system processes implicit gaze input allows users able to explore the visualization according to their interest or purpose without any other intentional behaviours.

Our solution for improving the readability of over-plotted scatter plots is to conduct a visual transition to dynamically raise the hidden dots of interest in the overlapping areas to the top. Using the group-of-interest (GOIs), the idea is to bring the hidden dots to the front when the user is interested in that group of data, allowing the user to efficiently examine the whole dataset group by group. Thus, the question has become *when should the system change the rendering order and which group of dots should be raised*?

In the demo, we create an over-plotted scatter plot with the multidimensional Iris dataset [3, 35], using different colours to represent the various types of flowers. To make the demo similar to real-world scenarios, additional flower samples have also been added to the dataset to increase the data diversity as well as the complexity of the overplotting. The demo system used the interest model from Chapter 4 to process the user's implicit gaze input and obtain their GOI. After detecting that the user's GOI changed, the system will execute the visual transition to change the rendering order of dots based on the user's interest.

Since our purpose in designing the interactive visualization is to offer support as subtle and non-distracting as possible, the visual transition is constrained by two conditions. First, a speed limit to the gaze action is set for the transition. Since people do not pay attention to details and have difficulty seeing visual transitions while moving their eyes, the scatter plot will only change the rendering order of dots on the whole display if the user's gaze speed is over 30 deg/s (i.e., detected as a saccade). Secondly, when the gaze movement is below 30 deg/s, the visual transition will only occur outside the user's central vision, to reduce the user's possibility of discovering it. Restricting the transition to areas outside the central region also reduces the chance that something the user is currently examining will change while being examined, which could be frustrating. In Figure 5.6, three images show where a user is looking, and the gray circle is the central vision. The dots in the scatter plot from the first picture is the original rendering order. By detecting the user's group of interest in the second image, dots in purple (Setosa) outside the user's central vision were raised by our system to the top of other colour group dots. The rest dots from the interest group inside the user's central vision would rise after the user's gaze shifted to another position. Also, the user can use a mouse click to select an individual dot to raise it to the top and indicate its information (e.g., colour, unique ID) by showing a dialogue.

5.3 Improving Target Selection with the Gaze Additive Voronoi Cursor

Target selection is a basic task for acquiring graphical-user-interface (GUI) components such as buttons, icons and menu options. Most visual interactive applications, such as web browsers and information visualizations, still require users to frequently select and interact with a mouse or keyboard. With the increment in both size and resolution of computer displays, it becomes less available and efficient for a user to acquire small visual elements surrounded by multiple nearby objects on the large display with the traditional cursor techniques [51, 100]. One promising solution is the Bubble Cursor which can dynamically adjust the cursor's activation area until the closest target is captured [39]. This is equivalent to expanding the boundary of each target to the Voronoi region with the target center being the region center, so that the Voronoi diagram [69] defined by all targets fills the whole screen space (Figure 5.7).

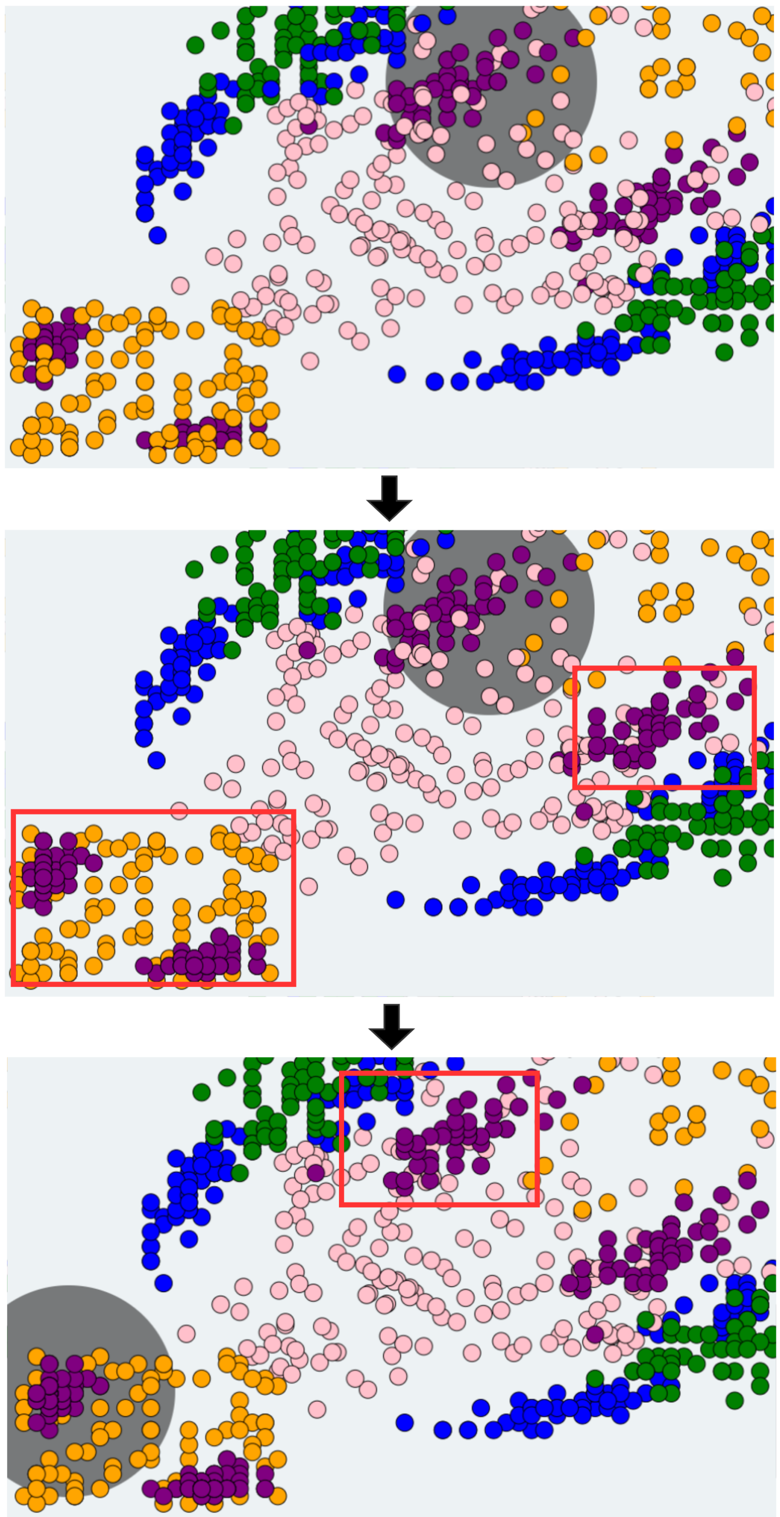
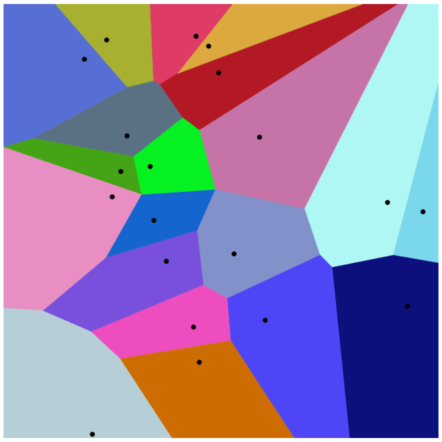


Figure 5.6: A screenshot of data objects' rendering order from the over-plotted scatter plot during the visual transition. It is noticeable that the dots from user's group-of-interest (purple) inside the user's central vision (gray circle) won't change until the gaze moved to the left bottom.



(a) Euclidean distance

(b) Manhattan distance

Figure 5.7: Voronoi diagrams under two different metrics [32].

To improve the performance of Voronoi tessellation in target selection, the state-of-the-art technique uses a weighted Voronoi diagram [72] to replace the static Voronoi diagram. A weighted Voronoi diagram in n dimensions is a generalization of a Voronoi diagram. The Voronoi cells in a weighted Voronoi diagram are defined in terms of distance metrics (see Figure 5.7). In weighted Voronoi diagrams, each site (target) has a weight that influences the distance calculation. The idea is that larger weights indicate more important sites, and such sites will get larger Voronoi cells. All Voronoi cells from a weighted Voronoi diagram need to be recalculated when a site's weight changes. In an additively-weighted Voronoi diagram, instead of only using the distance metric as measurement to find out the closest target, weights are subtracted from the distances and a target with a higher weight will occupy a larger Voronoi tessellation. Jacky proposed the Additive Voronoi Cursor (AVC) [19], which changes the weight of targets' Voronoi cells and resizes their effective areas after the mouse movement switches from the ballistic (moving towards a target) to correction (slow down and correct the final destination) phases. The overall target selection time is reduced since the correction cursor movement becomes easier. However, the error rates of the AVC method increased while the amplitude of the mouse movement was larger.

Inspired by this technique, we present the Gaze Additive Voronoi Cursor that combines the additively-weighted Voronoi cursor and our gaze-based interest model. Subjoining a content-aware weighting method for adjusting the weights of on-screen targets according to both the pattern of gaze movement and the local target density as well as the distribution, our goal is to provide a better target selection performance, which able to dynamically resize the effective ar-

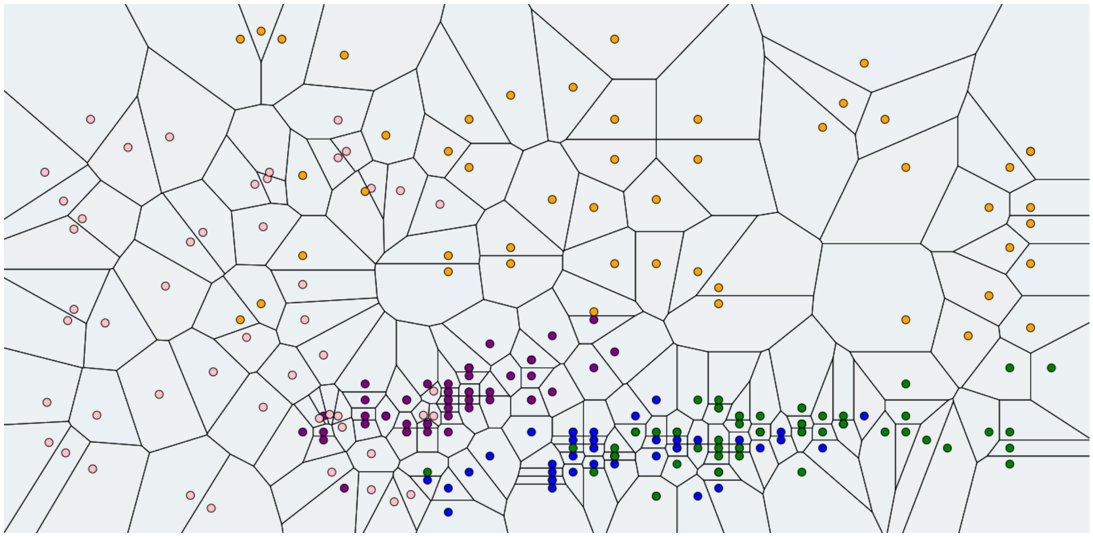


Figure 5.8: An example of the Voronoi scatter plot.

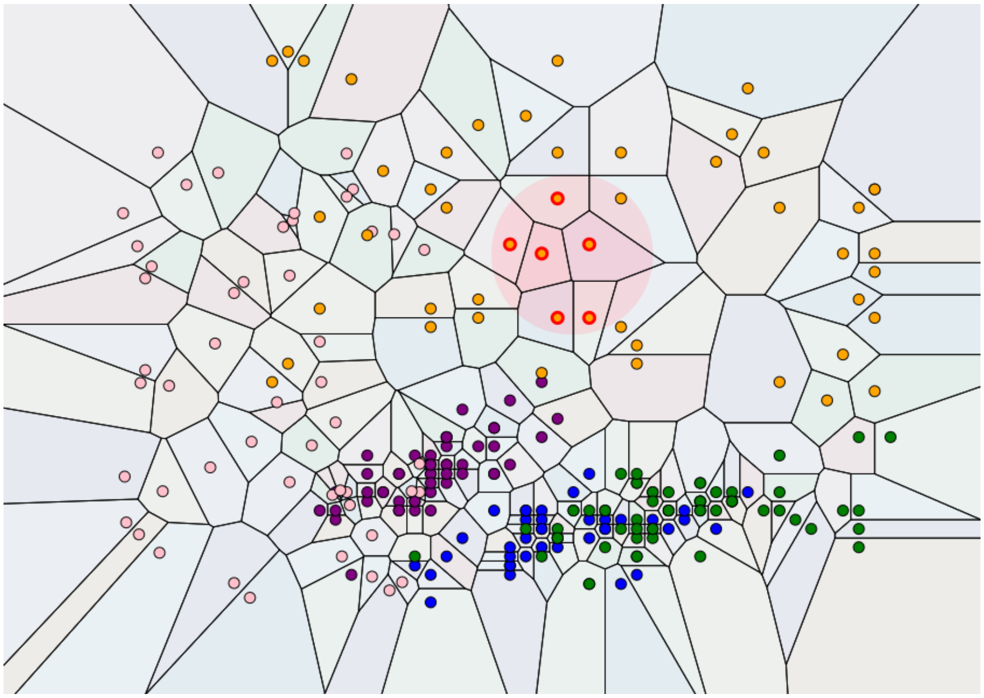
eas (Voronoi region) based on gaze location and result from the interest model. The advantages of using gaze to support target selection are:

1. Normally before moving the mouse to select, the user must notify the target in advance with a gaze [17]. Supporting by implicit gaze input, it is possible to offer the system more time than just using a mouse to prepare the interaction, which increases the target selection accuracy by speeding up the weighted Voronoi diagram calculation.

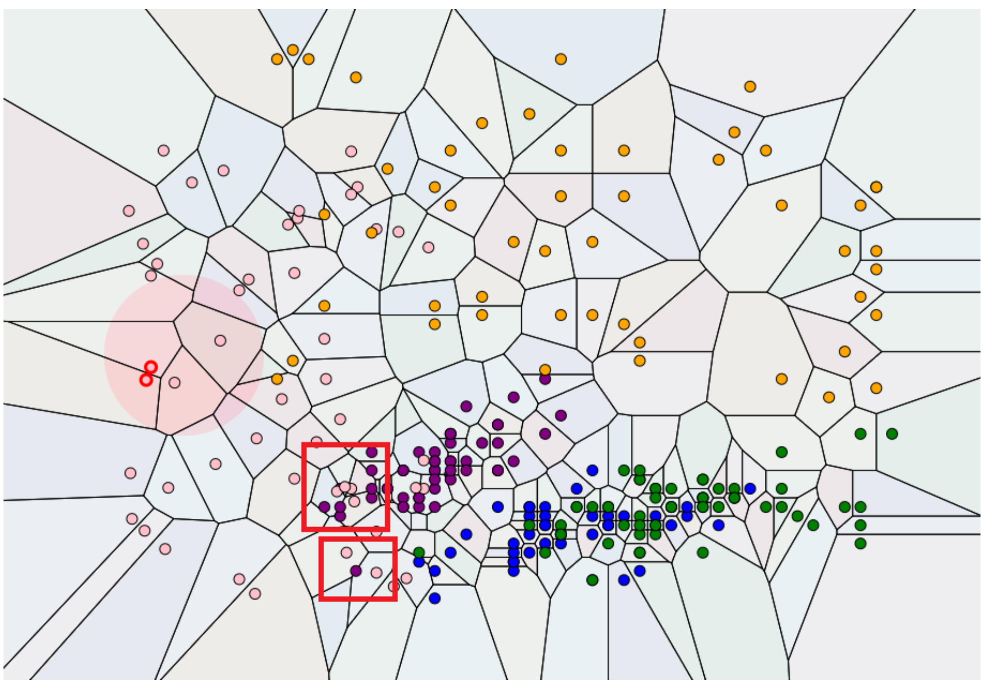
2. The Additive Voronoi Cursor may fail to predict the correct target when the amplitude of the mouse movement increase, which is highly related to individual behaviour. Using gaze is more natural for people than using a mouse [24, 62]. Eye movements between points of fixation are ballistic; that is, no correction for errors in trajectory are made while the eye is in motion [99]. In other words, using gaze for target selection is not only more accurate than mouse but also removes a step in the selection algorithm.

3. The on-screen targets are not always isolated from each other in real-world cases. If there are connections between the targets, the interest model can be used to detect group-of-interest and resize the interest group's effective areas for the selection, which is possible to improve the accuracy of selecting interested targets for the user. In Figure 5.9, the Voronoi tessellation (reachable area) of each target changed based on weight contribution from not only the user's fixation but also the result from the interest model.

After illustrating the benefit of our Gaze Additive Voronoi Cursor, it is also important to design proper implicit interaction behind the target selection to improve the experience for



(a) Voronoi tessellation when the user interested in the orange group.



(b) Voronoi tessellation when the user interested in the pink group. In the red square, the pink group targets' reachable regions were expanded by AGVC, which increase the success rate for user accessing pink targets. It is noticeable that the purple dot's regions were reduced, but the user can still select them by directly clicking.

Figure 5.9: Dynamically changing the weighted Voronoi tessellations according to user's different interests to groups of targets in real-time.

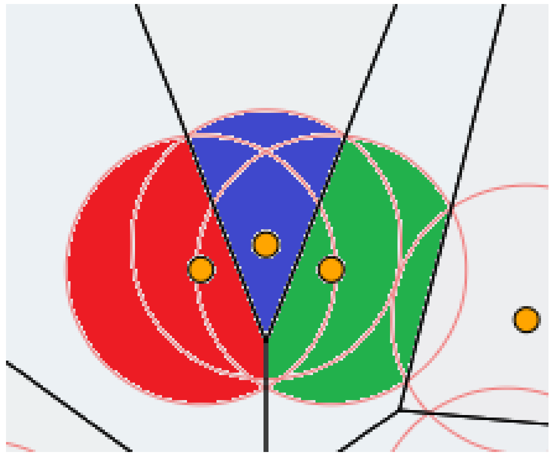


Figure 5.10: Reachable region of targets under weighted Voronoi tessellation.

users. Although users can select a target by clicking on its Voronoi region without moving the mouse directly on it, it is still necessary to constrain the target selection with a maximum distance between the mouse and the target inside the weighted Voronoi region (See Figure 5.10). This is because if there is no distance limitation for target selection, incorrect selective actions may be detected due to users' unconscious clicks on the space area. In Figure 5.10, several orange targets with their own weighted Voronoi region segmenting by black line are inside pink circles. The pink circle is the actual reachable region for targets that allow the user to select. The reachable region of the three targets in the middle are indicating with colour red, blue and green.

The success rate of target selection in high target-density area could be lower than a space with sparse targets since the high density requires higher accuracy of mouse clicks. To solve this issue, Gaze Additive Voronoi Cursor in high-density targets area uses a strategy where a target A could have an effective area containing other target B (see Figure 5.9b). In this situation, the reason why target B stays inside target A's reachable region without having its own one is because target B's weight may be much smaller than target A and the space around the target B is closer distance-wise to target A according to the calculation of the additive weighted Voronoi diagram. But the user can still select the target B by directly clicking on it. The Gaze Additive Voronoi Cursor can increase the accuracy of target selection in both sparse and crowded regions of targets.

6 Proposed Evaluation

We have built two demonstration applications based on our two different implicit interaction designs with Pupil-core glasses or Tobii eye-tracker. The demos are based on web-front d3.js visualizations and a back-end node.js server. Due to the Covid-19 pandemic, in-person studies and evaluations are not permitted currently. In lieu of a study and results, here we present the planned methodology to evaluate each of the applications. The study plans focus on understanding the impact of the gaze interest model and visualization adjustments based on the model on the efficiency of data analysis and the level of distraction for the user.

6.1 Participants

The study will take place in the Human-Centred Computing Lab at Ontario Tech University, to provide for control of the experimental conditions. According to previous work [74], we will recruit 30 participants who are using a computer and have normal or corrected-to-normal vision. Participants will be screened for colour vision deficiencies. In the future, we will use power analysis to determine how many participants are needed to detect a difference of a specific amount between conditions in our studies.

6.2 Apparatus

The required equipment for the studies: (1) A 16-inch laptop, (2) Pupil-core glasses or Tobii eye-tracker, (3) Mouse.

6.3 Procedure

There will be at least one investigator supervising and only one participant for each session. We will provide a paper instruction about the various precautions of the studies at the beginning. Next, the study supervisor will set up the eye-tracking device and lead a screen calibration for the participant. In order to avoid the influence of light factors on the study results, we will make sure the light setting for all participants is the same. Then we will open the demo and ask the participants to follow the guidance inside the demo. Each session will run for an hour and consist of two sub-studies as the following sections. During the session, we will record the screen and collect the participant's gaze and mouse data. The demo we provide will also measure the time spent and the interest model's result in each task. At the end of the study, the participant will be asked to answer questions and provide feedback. We will analyze the measurements and evaluate the accuracy and usability of our implicit gaze interaction design.

6.4 Study Interface

Our proposed application aims to present an adaptive visualization that automatically reacts to the implicit gaze input from users to help them explore the visualization. The study interface presents the state of the gaze input and output from our interest model, which will show information that the user might be interested in, without distraction, by conducting subtle gaze-enabled visual transitions or improving target selection accuracy by resizing the reachable region with Gaze Additive Voronoi Cursor.

First, the intermediary program from Chapter 3 takes and processes the gaze input from the eye-tracking devices (Pupil-core glasses or Tobii eye-tracker) and pushes it to a network port where the study application is listening. The study interface takes a series of gaze inputs and the interest model calculates and generates the interest scores based on the participant's past gaze actions. The gaze data is fed into the interest model as *user vectors* and the interest scores of each object groups for each prediction is displayed on an exploratory analytics interface.

We had some initial criteria for designing our interface based on the gaze input, different settings and modalities for the interest model and the implicit interaction system. Because our studies only need participants to finish tasks by using their eyes to read and mouse to interact with the data objects from a visualization, simplicity and intuitive design were key. A straightforward way is needed for users to understand the goal of the task, and an intuitive way to display the gaze action information and interest level with the visualization content and control variables.

We went with a single-page design with three components: menu bar, interest scores, and gaze state bar. In the menu bar, the investigator can change settings such as: the visibility of user's gaze point and objects that selected by gaze, the frequency of calculating the interest score. Since we need to compare the effectiveness with or without our implicit interaction in the evaluation, the interface is designed so the implicit interaction can be toggled. The interest scores are present at the right top corner showing the participant's interest level to the group of objects on the visualizations. The gaze state bar at the bottom shows the type and the speed of the participant's current gaze action. The design and selection of each component are described in Figure 6.1.

6.5 Study 1: Non-distracting Support for Occlusion Mitigation

In this study, the demo that provides non-distracting support for participants is going to be evaluated. Without visual guidance or support, participants can easily be stuck into their early bias and ignore important content before going through all the information in the visualization reading task. The demo in this study will collect the implicit gaze input from participants. It will calculate the interest scores and predict the group-of-interest based on the past gaze actions. Using this, it will provide non-distracting support to benefit participants to obtain information quickly and entirely in visualization reading tasks.

There are several questions that this study need to be answered from this study.

(a) Can our non-distracting support improve the speed of reading a cluttered scatter plot?

(b) Can our non-distracting support improve the readability of the over-plotted scatter plot?

(c) Can our interest model correctly predict participants' attention?

(d) Is our implicit gaze interaction non-distracting?

To answer questions (a) and (b), at the beginning of the study, the participants will be assigned to one of two groups with counterbalancing to carry out several tasks – reading data objects from a specific group or a group sequence. The first group will used the system *with the non-distracting support*, and the second group will have *no gaze support*. To determine whether our implicit interaction will distract the participants or not, the first group also needs to be split further into two subgroups with visual transition *everywhere* or *outside* the foveal region.

The data we plan to collect for each task are: the reading time, logged results from the interest model, and the answers to questions after a reading task. In order to minimize the interference of unrelated variables, the distribution and colour of the group information on the visualization is generated randomly.

Reading tasks will suggest specific actions, such as 'Review all the purple points to find if there is a trend.' Participants will be asked to complete the task as quickly and accurately as possible, and to press a button when done (to stop the timer). After the completion of each reading task, we will ask several questions, focusing on the distribution of the whole data set or group information in a specific area, as well as a single data object. In addition, the participant will be asked questions related to what they saw on the visualization such as: the distribution

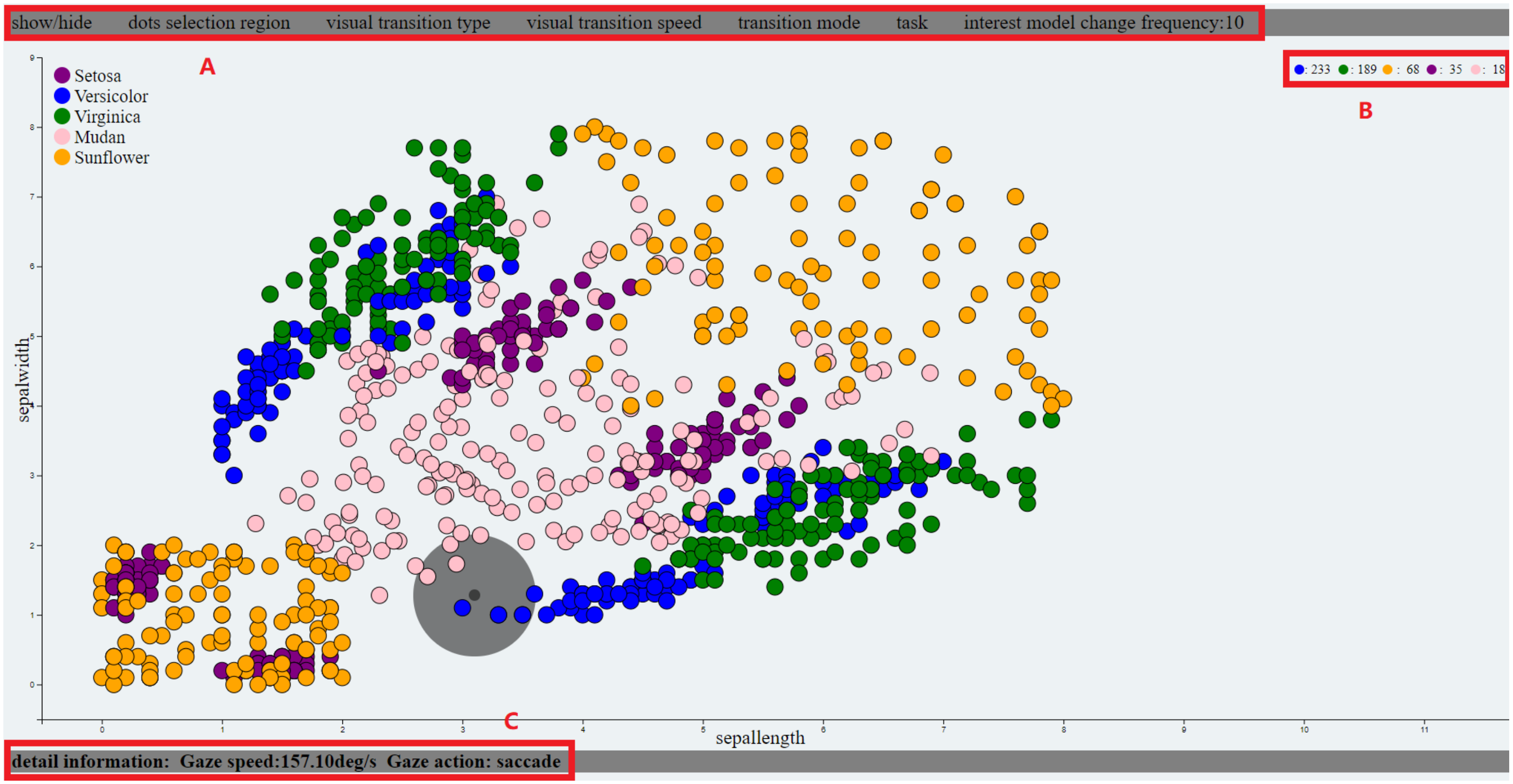


Figure 6.1: The Interface of our demo[a]. A is the menu bar of the demo for the investigator to adjust study parameters. B is the interest score result from the interest model. C is the current gaze state information.

[a] Link to the demo video: https://www.youtube.com/watch?v=khdJ86tkgd0

of a group that the participant was assigned to review, or a specific dot from the group but the participant have not seen. Here are example questions which could be used after the reading task:

Q1. Were you able to read all information?

Q2. What is the distribution of the purple group?

Q3. Have you noticed there is a dot at *(x,y)* location?

Q4. Did you notice any changes in the visualization? How do you think of changes and the mechanism behind them?

By measuring the completion time of the reading task for each participant, we can use the task as factor to run a comparative study and use inferential statistics to analyze the data to answer the question (a). From participants' feedback of *Q1* to *Q3*, we can answer the question (b). By manually verifying the interest model's result with the task goal, question (c) can be answered. Participant responses to *Q4*, grouped by different visual transition regions (everywhere or outside foveal region), will answer (d), the degree of distraction from implicit gaze interaction.

6.5.1 Alternative Interface: Maps

A map demo (see Figure 6.2) has also been built as an alternative option to this study since it is related to a real-world scenario. We randomly picked a crowded area in Toronto and used the Google Maps API to collect data of different types of point-of-interest (POI) such as bus stations, government buildings, groceries, hospitals, restaurants, subway stations in this region. This interface could also be tested to evaluate non-distracting support to help users to locate a target place in a cluttered map.

6.6 Study 2: Gaze Additive Voronoi Cursor for Target Selection

In this study, we will test the performance of our Gaze Additive Voronoi Cursor (GAVC) on target selection tasks. There are two concerned questions need to be answered in this study. First is comparing the mouse-based additive Voronoi cursor and gaze-based additive Voronoi cursor: which is faster and more accurate? The second question is the effectiveness of using group-of-interest to improve the success rate for a user selecting type-related or content-related targets such as buttons from a menu or data objects from the same colour group.

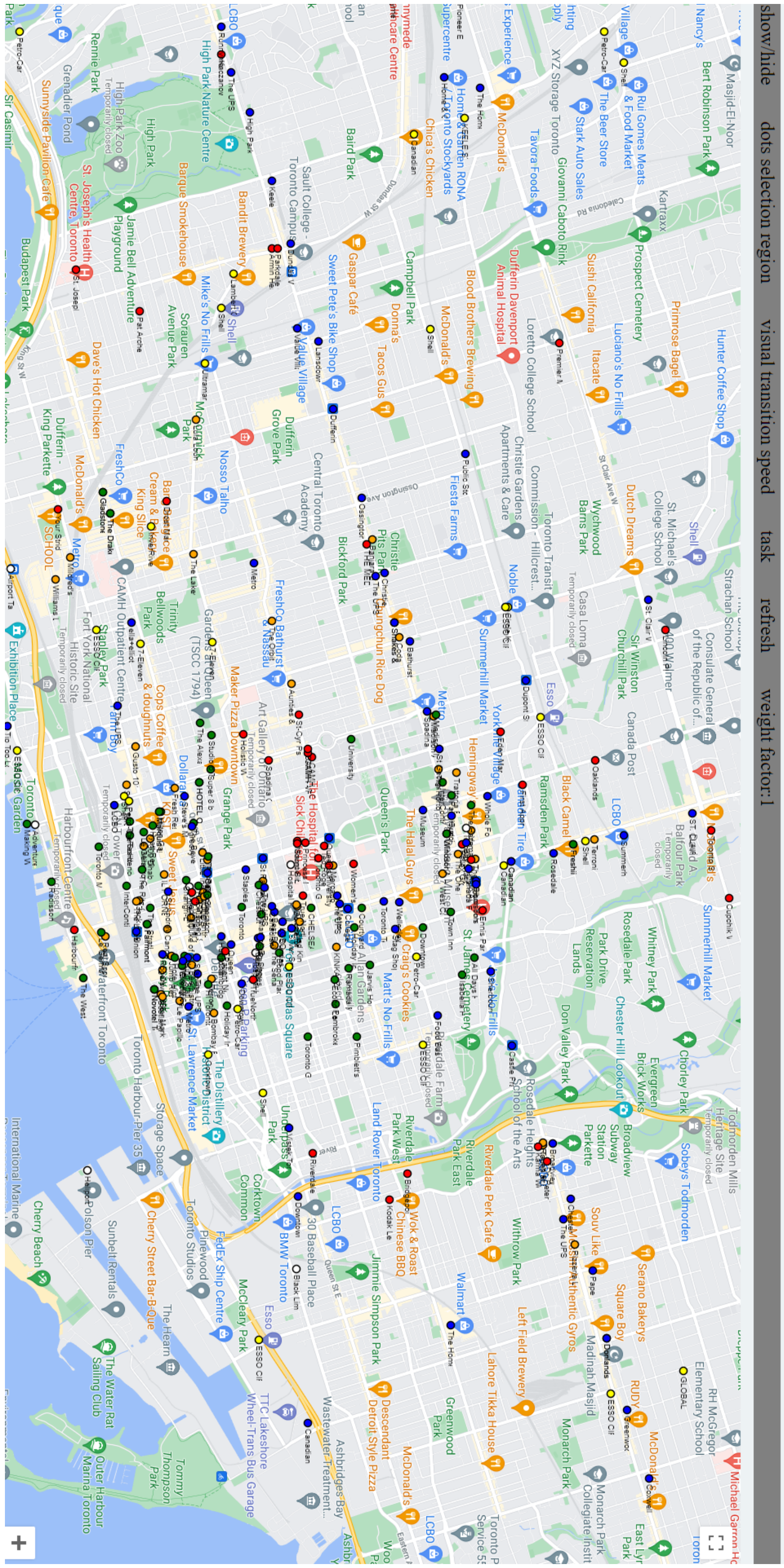


Figure 6.2: Interface of the Map demo with different point-of-interest categories marked with different coloured dots.

We can simply separate participants into two groups by the cursor technique (*GAVC* and *AVC*) for the first question. The demo will randomly highlight a target after each selection. Participants will be asked to select as many targets as possible within a certain period of time, like 5 minutes. In order to compare with the AVC, the GAVC will neglect the weight contribution from the interest model and focus on pure gaze movement, just like AVC rely on mouse movement. We will measure the number of selected targets and the error rate. Then, the average number of targets and error rate will be calculated for comparing gaze-based and mouse-based additive Voronoi cursors.

For the second question, participants can be divided into two groups with the interest model *on* or *off*. They will be asked to select all targets from a group several times. The measurement is similar to the first question. By comparing the average time and accuracy between the two groups, we can find out if the interest model is playing an important role in the information-related target selection.

6.7 Expectations from the Evaluation

In the first study, it is expected that non-distracting support will improve the speed and accessibility of the reading tasks since the visualization can bring dots from the user's group-of-interest to the front to remove the blindfold of overlapping. The key to the study's success or not is the accuracy of the interest model can model and predict the participant's interest or not. Normally in a region where the density of dots is low and the groups are isolated to each other, the interest model can work pretty well. However, in the situation where the interest model works in a high-density area where dots from different groups are overlapping each other, the performance of the model still remains uncertain. It is hard for the system only to use gaze to discern if a participant is focusing on an overlapped dots in a crowded area that are partly visible. One solution is to use a mouse to hover only the partly visible dots, but this requires objects with enough size on the visualization and high precision mouse movement from the participants. For the second study, it's more direct and easy for us to examine the performance and usability of the GAVC by measuring the error rate and time spend for each trial of selection task.

There is an interesting question for implicit gaze interaction that have not been fully test, verified or discussed before: Does implicit interaction become explicit when the user understands what is happening? At this moment we can only say it depends how well the user understand what is happening. For example, the participants might have a chance to notice the visual transition of dots from the group-of-interest outside the foveal region in the first study. Before they find out there's a relationship between the visited dots and the visual changes where the rest from the same group also are raised to the top, we think it is still implicit interaction. But as the participants understand the visualization can bring the dots to the front and start making use of this mechanism, the interaction has become explicit. For the second study, if the participants are never told that the effective area of targets are changing based on how they move their eyes. It will be implicit interaction forever. This question is worth discussing since using implicit interaction in an explicit way would be slower than just designing an appropriate explicit interaction to achieve the same result.

7 Conclusions

This chapter discusses the contributions, work limitations, conclusions and some ideas for possible future work directions for this research.

7.1 Contributions

One of the contributions is our video processing pipeline that innovates by combining a contour detection algorithm with an ORB feature matching algorithm to maximize the accuracy of a user's gaze estimation across multiple screens. The pipeline can automatically adapt to the incoming gaze data from the eye-tracker and smoothly differentiate the boundary of the target screen and the others, which reduces the cost of time of gaze estimation and improves the interaction experience by removing a step – marker registration. This also provides an opportunity for visualization specialists to design visualizations suitable for gaze interaction but does not need to sacrifice the design space for gaze estimation.

Furthermore, the result of the gaze estimation from our video processing pipeline was fed into our interest model based on the data-of-interest (DOI) method and a top-down approach. By relating the detected gaze action and salient features of multiple dimensions on the visualization, this model calculated the interest scores for group-of-interest (GOI) with a mathematical formula and predicts future attending object targets or regions in data visualization.

Finally, an implicit gaze system interaction was designed and implemented based on the video processing pipeline and the interest model result from users' continuous implicit gaze input. The implicit interaction system works by providing non-distracting guidance to improve the usability and readability of the visualizations, which allow users to interact more naturally and comfortably with the visualization during the information discovery. However, whether the system provides the intended benefit must be determined through a future experiment.

An evaluation plan is also prepared in Chapter 6 that this implicit gaze interaction system can be studied in different use cases, such as information access based on changing rendering order of objects on visualization and target selection like clicking a button.

7.2 Limitations and Future Work

7.2.1 Prediction with Machine Learning

To successfully predict the user's gaze location and targets of interest, we chose a traditional computer vision algorithm (contour detection algorithm and ORB feature detection) and built an interest model with a mathematical weighted formula rather than using machine learning methods. While machine learning becomes more competitive, the depth of deep learning neural networks and computational cost grew exponentially. When choosing machine learning as a method to solve the problems we face, it is necessary to carefully determine the probability of making a fuss over a trifle. Sometimes simple methods can achieve better results. As far as the current situation is concerned (the results achieved by our method), the use of machine learning will not offer us a big step forward but likely slow down our progress since we need to train machine learning models and optimize them. A drawback of machine learning is that it requires a significant amount of training data for a competitive performance, which can easily annoy or frustrate users. Nevertheless, recent work has proved there is a huge potential between machine learning and gaze interaction [18]. In future work, we can use machine learning algorithms to train a model with a deeper and more complicated framework for gaze estimation or predicting targets of interest that users will interact with.

7.2.2 Interaction Across Multi-users

Several implicit interaction designs have been proposed in previous chapters to benefit people from roaming information on the visualization. We believe our methods can also be extented to scenarios where multiple users are interacting with a visualization system. For implicit interaction that provides guidance or support across multi-users, interference and sharing of system response information from the interaction is our main consideration. It's a trade-off between conducting visual transition inside the foveal vision to protect other users and reducing the distraction by disturbing others. Unlike the design for a single user, the subtle guidance or support should only be visible to the related user when their gaze is joined together rather than being outside the foveal vision since exposing personal or customized data from the guidance to another user may cause confusion or privacy risk. On the other hand, designing gaze interaction with multiple users can benefit simultaneously and collaborative information discovery. For example, when the interest model finds out two users share the same interest in a data group, the overplotting visualization could bring it to the front with different visual effects, or the jointed gaze becomes a lens to zoom in location of their fixation.

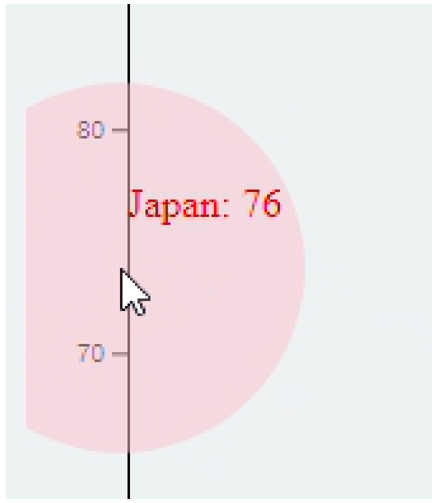


Figure 7.1: When the axis detected fixation (in pink circle), it showed the object and the value that the user just viewed with red text.

7.2.3 Other Types of Tasks and Visualizations

In our implicit gaze interaction design, we used a winner-take-all method to determine a user's attention — group of interest (the group with highest interest score from interest model). The reason for this is we are using implicit interaction, and the user must have a goal in the study. Otherwise, the implicit gaze input will become meaningless. What should the result look like in an open-ended task? This question cannot be answered at this moment. In some condition, it is possible to have a user who is interested in two different groups and the interest scores are close to each other. If our system only provides guidance to the winner group, it could create bias or mislead the user. One hidden direction of implicit interaction research is to analyze and study joint interest or attention. Another direction is to find out the perfect size of the interest scores moving window and the moving step. These require studies with more complicated tasks and measurement designs.

Also, this work only focuses on implicit gaze interaction with scatter plot visualization on middle-size screens. Suppose we extend the interaction to other sizes of screens such as large public displays or mobile phones. The designs can be totally different. Although existing implicit interactions have only included basic visualization (e.g. line graph, scatter plot, charts) and user interface. It is believed that the opportunities of using the gaze interaction combining advanced visualization and different types of tasks are enormous. We have some initial but interesting ideas related to the axis of charts with gaze interaction (see Figures 7.1 and 7.2).

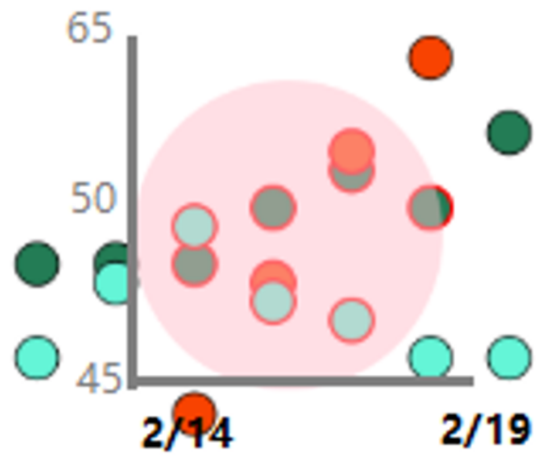


Figure 7.2: After a user fixed on a location (in pink circle) for several seconds, a reference axis would be shown to help the user read the values of the dots.

7.3 Conclusion

Our screen acquisition pipeline, gaze-data-based user interest model and implicit gaze interaction system are all promising steps towards ensuring gaze estimation success as well as improve the experience of visual analytics from a computer vision and human-computer interaction perspective. Our method is able to discern the target display that the user is looking, calculate an accurate gaze estimation, and feed the gaze data into an interest model to predict user's attention to data objects on visual stimuli. This may improve information readability and target accessibility of visualization.

Without a formal evaluation of our system, it is difficult to say that the current implicit interaction designs is successful in terms of interacting with the interest model and supporting information access and target selection in a meaningful way.

However, due to the Covid-19 pandemic we can only provide a prepared study plan to evaluate our method when the situation is back to normal. In summary, we were able to predict accurate gaze estimation across multiple screens, model user attention in the form of interest and design and implement an implicit gaze interaction system that supports user's visualization discovery and target selection.

Reference

1. A. Alahi, R. Ortiz, and P. Vandergheynst. "FREAK: Fast Retina Keypoint". *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 510–517. DOI: 10.1109/CVPR.2012.6247715.

2. S. S. Alam and R. Jianu. "Analyzing Eye-Tracking Information in Visualization and Data Space: From Where on the Screen to What on the Screen". *IEEE Transactions on Visualization and Computer Graphics* 23:5, 2017, pp. 1492–1505. DOI: 10.1109/TVCG.2016.2535340.

3. E. Anderson. *The irises of the Gaspé peninsula*. 1935. URL: https://docs.opencv.org/master/dc/dc3/tutorial_py_matcher.html (visited on 03/16/2021).

4. J. R. Anderson, D. Bothell, and S. Douglass. "Eye Movements Do Not Reflect Retrieval Processes: Limits of the Eye-Mind Hypothesis". *Psychological Science* 15:4, 2004. PMID: 15043638, pp. 225–231. DOI: 10.1111/j.0956-7976.2004.00656.x. eprint: https://doi.org/10.1111/j.0956-7976.2004.00656.x. URL: https://doi.org/10.1111/j.0956-7976.2004.00656.x.

5. M. Barz, F. Daiber, D. Sonntag, and A. Bulling. "Error-aware gaze-based interfaces for robust mobile gaze interaction". *Proc. ACM Symposium on Eye Tracking Research & Applications*. ACM, 2018. DOI: 10.1145/3204493.3204536.

6. D. Baur, S. Boring, and S. Feiner. "Virtual Projection: Exploring Optical Projection as a Metaphor for Multi-Device Interaction". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Association for Computing Machinery, Austin, Texas, USA, 2012, pp. 1693–1702. ISBN: 9781450310154. DOI: 10.1145/2207676.2208297. URL: https://doi.org/10.1145/2207676.2208297.

7. E. Bertini and G. Santucci. "Quality Metrics for 2D Scatterplot Graphics: Automatically Reducing Visual Clutter". *Smart Graphics*. Ed. by A. Butz, A. Krüger, and P. Olivier. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 77–89. ISBN: 978-3-540-24678-7.

8. T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. "State-of-the-Art of Visualization for Eye Tracking Data". *EuroVis - STARs*. Ed. by R. Borgo, R. Maciejewski, and I. Viola. The Eurographics Association, 2014. ISBN: 978-3-03868-028-4. DOI: 10.2312/eurovisstar.20141173.

9. R. A. Bolt. "Eyes at the Interface". *Proceedings of the 1982 Conference on Human Factors in Computing Systems*. CHI '82. Association for Computing Machinery, Gaithersburg, Maryland, USA, 1982, pp. 360–362. ISBN: 9781450373890. DOI: 10.1145/800049.801811. URL: https://doi.org/10.1145/800049.801811.

10. S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. "Touch Projector: Mobile Interaction through Video". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Association for Computing Machinery, Atlanta, Georgia, USA, 2010, pp. 2287–2296. ISBN: 9781605589299. DOI: 10.1145/1753326.1753671. URL: https://doi.org/10.1145/1753326.1753671.

11. J. Boy, L. Eveillard, F. Detienne, and J. Fekete. "Suggested Interactivity: Seeking Perceived Affordances for Information Visualization". *IEEE Transactions on Visualization and Computer Graphics* 22:1, 2016, pp. 639–648. DOI: 10.1109/TVCG.2015.2467201.

12. J. Breuninger, C. Lange, and K. Bengler. "Implementing Gaze Control for Peripheral Devices". *Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-Based Interaction*. PETMEI '11. Association for Computing Machinery, Beijing, China, 2011, pp. 3–8. ISBN: 9781450309301. DOI: 10.1145/2029956.2029960. URL: https://doi.org/10.1145/2029956.2029960.

13. X. Bu, M. Pei, and Y. Jia. "Attention Estimation for Input Switch in Scalable Multi-display Environments". *Neural Information Processing*. Ed. by A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, and D. Liu. Springer International Publishing, Cham, 2016, pp. 329–336. ISBN: 978-3-319-46681-1.

14. M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "Brief: Binary robust independent elementary features". *Proc. European Conference on Computer Vision*. Springer. 2010, pp. 778–792.

15. D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H. Schulz, M. Streit, and C. Tominski. "Characterizing Guidance in Visual Analytics". *IEEE Transactions on Visualization and Computer Graphics* 23:1, 2017, pp. 111–120. DOI: 10.1109/TVCG.2016.2598468.

16. S. Chandra, G. Sharma, S. Malhotra, D. Jha, and A. P. Mittal. "Eye tracking based human computer interaction: Applications and their uses". *2015 International Conference*

on Man and Machine Interfacing (MAMI). 2015, pp. 1–5. DOI: 10.1109/MAMI.2015.7456615.

17. M. C. Chen, J. R. Anderson, and M. H. Sohn. “What Can a Mouse Cursor Tell Us More? Correlation of Eye/Mouse Movements on Web Browsing”. *CHI ’01 Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’01. Association for Computing Machinery, Seattle, Washington, 2001, pp. 281–282. ISBN: 1581133405. DOI: 10.1145/634067.634234. URL: https://doi.org/10.1145/634067.634234.
18. X. Chen, A. Acharya, A. Oulasvirta, and A. Howes. “An Adaptive Model of Gaze-Based Selection”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2021. ISBN: 9781450380966. URL: https://doi.org/10.1145/3411764.3445177.
19. J. K. Cheung, O. K.-C. Au, and K. Zhu. “Additive Voronoi Cursor: Dynamic Effective Areas Using Additively Weighted Voronoi Diagrams”. *Human-Computer Interaction – INTERACT 2019*. Ed. by D. Lamas, F. Loizides, L. Nacke, H. Petrie, M. Winckler, and P. Zaphiris. Springer International Publishing, Cham, 2019, pp. 273–292. ISBN: 978-3-030-29387-1.
20. M. Choi, D. Sakamoto, and T. Ono. “Bubble Gaze Cursor + Bubble Gaze Lens: Applying Area Cursor Technique to Eye-Gaze Interface”. *ACM Symposium on Eye Tracking Research and Applications*. ETRA ’20 Full Papers. Association for Computing Machinery, Stuttgart, Germany, 2020. ISBN: 9781450371339. DOI: 10.1145/3379155.3391322. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3379155.3391322.
21. Ç. Çığ Karaman and T. M. Sezgin. “Gaze-based predictive user interfaces: Visualizing user intentions in the presence of uncertainty”. *International Journal of Human-Computer Studies* 111, 2018, pp. 78–91. ISSN: 1071-5819. DOI: https://doi.org/10.1016/j.ijhcs.2017.11.005. URL: https://www.sciencedirect.com/science/article/pii/S1071581917301611.
22. C. Collins, N. Andrienko, T. Schreck, J. Yang, J. Choo, U. Engelke, A. Jena, and T. Dwyer. “Guidance in the human–machine analytics process”. *Visual Informatics* 2:3, 2018, pp. 166–180. ISSN: 2468-502X. DOI: https://doi.org/10.1016/j.visinf.2018.09.003. URL: https://www.sciencedirect.com/science/article/pii/S2468502X1830041X.
23. A. De Luca, R. Weiss, and H. Drewes. “Evaluation of Eye-Gaze Interaction Methods for Security Enhanced PIN-Entry”. *Proceedings of the 19th Australasian Conference on*

Computer-Human Interaction: Entertaining User Interfaces. OZCHI '07. Association for Computing Machinery, Adelaide, Australia, 2007, pp. 199–202. ISBN: 9781595938725. DOI: 10.1145/1324892.1324932. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/1324892.1324932.

24. U. Demšar and A. Çöltekin. "Quantifying gaze and mouse interactions on spatial visual interfaces with a new movement analytics methodology". *PLOS ONE* 12:8, 2017, pp. 1–36. DOI: 10.1371/journal.pone.0181818. URL: https://doi.org/10.1371/journal.pone.0181818.

25. J. Dostal, U. Hinrichs, P. O. Kristensson, and A. Quigley. "SpiderEyes: Designing Attention- and Proximity-Aware Collaborative Interfaces for Wall-Sized Displays". *Proceedings of the 19th International Conference on Intelligent User Interfaces*. IUI '14. Association for Computing Machinery, Haifa, Israel, 2014, pp. 143–152. ISBN: 9781450321846. DOI: 10.1145/2557500.2557541. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2557500.2557541.

26. J. Dostal, U. Hinrichs, P. O. Kristensson, and A. Quigley. "SpiderEyes: Designing attention- and proximity-aware collaborative interfaces for wall-sized displays". *Proc. ACM Conf. on Intelligent User Interfaces*. 2014, pp. 143–152.

27. H. Drewes, A. De Luca, and A. Schmidt. "Eye-Gaze Interaction for Mobile Phones". *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*. Mobility '07. Association for Computing Machinery, Singapore, 2007, pp. 364–371. ISBN: 9781595938190. DOI: 10.1145/1378063.1378122. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/1378063.1378122.

28. G. Drusch, J. M. C. Bastien, and S. Paris. "Analysing eye-tracking data: From scanpaths and heatmaps to the dynamic visualisation of areas of interest". *International Conference on Applied Human Factors and Ergonomics*. Krakow, Poland, 2014. URL: https://hal.archives-ouvertes.fr/hal-01223743.

29. A. T. Duchowski. "A breadth-first survey of eye-tracking applications". *Behavior Research Methods, Instruments, & Computers* 34, 2016, pp. 455–470. DOI: 10.3758/BF03195475.

30. A. T. Duchowski, N. Cournia, and H. Murphy. "Gaze-Contingent Displays: A Review". *CyberPsychology & Behavior* 7:6, 2004. PMID: 15687796, pp. 621–634. DOI: 10.1089/cpb.2004.7.621. eprint: https://doi.org/10.1089/cpb.2004.7.621. URL: https://doi.org/10.1089/cpb.2004.7.621.

31. M. Eaddy, G. Blasko, J. Babcock, and S. Feiner. “My own private kiosk: privacy-preserving public displays”. *Eighth International Symposium on Wearable Computers*. Vol. 1. 2004, pp. 132–135. DOI: 10.1109/ISWC.2004.32.

32. B. Ertl. *File:Euclidean Voronoi diagram.svg, File:Manhattan Voronoi Diagram.svg — Wikimedia Commons, the free media repository*. [Online; accessed 25-February-2021]. 2020. URL: %5Curl%7Bhttps://commons.wikimedia.org/wiki/File:Manhattan_Voronoi_Diagram.svg,%20https://commons.wikimedia.org/wiki/File:Euclidean_Voronoi_diagram.svg%7D.

33. “Exploring visual attention and saliency modeling for task-based visual analysis”. *Computers & Graphics* 72, 2018, pp. 26–38. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2018.01.010. URL: https://www.sciencedirect.com/science/article/pii/S0097849318300104.

34. A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, and M. R. Morris. “Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design”. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI ’17. Association for Computing Machinery, Denver, Colorado, USA, 2017, pp. 1118–1130. ISBN: 9781450346559. DOI: 10.1145/3025453.3025599. URL: https://doi.org/10.1145/3025453.3025599.

35. R. Fisher. *The use of multiple measurements in taxonomic problems*. 1936. URL: https://docs.opencv.org/master/dc/dc3/tutorial_py_matcher.html (visited on 03/16/2021).

36. S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. “Automatic generation and detection of highly reliable fiducial markers under occlusion”. *Pattern Recognition* 47:6, 2014, pp. 2280–2292. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2014.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0031320314000235.

37. S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. “Automatic generation and detection of highly reliable fiducial markers under occlusion”. *Pattern Recognition* 47:6, 2014, pp. 2280–2292.

38. T. van Gog, L. Kester, F. Nievelstein, B. Giesbers, and F. Paas. “Uncovering cognitive processes: Different techniques that can contribute to cognitive load research and instruction”. *Computers in Human Behavior* 25:2, 2009. Including the Special Issue: State of the Art Research into Cognitive Load Theory, pp. 325–331. ISSN: 0747-5632. DOI: https:

//doi.org/10.1016/j.chb.2008.12.021. URL: https://www.sciencedirect.com/science/article/pii/S0747563208002318.

39. T. Grossman and R. Balakrishnan. "The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '05. Association for Computing Machinery, Portland, Oregon, USA, 2005, pp. 281–290. ISBN: 1581139985. DOI: 10.1145/1054972.1055012. URL: https://doi.org/10.1145/1054972.1055012.

40. C. Hennessey and J. Fiset. "Long Range Eye Tracking: Bringing Eye Tracking into the Living Room". *Proceedings of the Symposium on Eye Tracking Research and Applications*. ETRA '12. Association for Computing Machinery, Santa Barbara, California, 2012, pp. 249–252. ISBN: 9781450312219. DOI: 10.1145/2168556.2168608. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2168556.2168608.

41. L. Herbert, N. Pears, D. Jackson, and P. Olivier. "Mobile device and intelligent display interaction via scale-invariant image feature matching". English. *PECCS 2011 - Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems*. 1st International Conference on Pervasive and Embedded Computing and Communication Systems, PECCS 2011 ; Conference date: 05-03-2011 Through 07-03-2011. 2011, pp. 207–214. ISBN: 9789898425485.

42. K. Holmqvist and R. Andersson. *Eye-tracking: A comprehensive guide to methods, paradigms and measures*. 2017. ISBN: ISBN-13: 978-1979484893.

43. T. E. Hutchinson, K. P. White, W. N. Martin, K. C. Reichert, and L. A. Frey. "Human-computer interaction using eye-gaze input". *IEEE Transactions on Systems, Man, and Cybernetics* 19:6, 1989, pp. 1527–1534. DOI: 10.1109/21.44068.

44. L. Itti and C. Koch. "Computational modelling of visual attention". *Nature Reviews Neuroscience 2*, 2001, pp. 194–203. DOI: 10.1038/35058500.

45. L. Itti, C. Koch, and E. Niebur. "A model of saliency-based visual attention for rapid scene analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20:11, 1998, pp. 1254–1259. DOI: 10.1109/34.730558.

46. L. Itti and C. Koch. "A saliency-based search mechanism for overt and covert shifts of visual attention". *Vision Research* 40:10, 2000, pp. 1489–1506. ISSN: 0042-6989. DOI: https://doi.org/10.1016/S0042-6989(99)00163-7. URL: https://www.sciencedirect.com/science/article/pii/S0042698999001637.

47. A. Jakubovic and J. Velagic. "Image Feature Matching and Object Detection Using Brute-Force Matchers". 2018, pp. 83–86. DOI: 10.23919/ELMAR.2018.8534641.

48. R. Jianu and S. S. Alam. "A Data Model and Task Space for Data of Interest (DOI) Eye-Tracking Analyses". *IEEE Transactions on Visualization and Computer Graphics* 24:3, 2018, pp. 1232–1245. DOI: 10.1109/TVCG.2017.2665498.

49. F. Jungwirth, M. Haslgrübler, and A. Ferscha. "Contour-Guided Gaze Gestures: Using Object Contours as Visual Guidance for Triggering Interactions". *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. ETRA '18. Association for Computing Machinery, Warsaw, Poland, 2018. ISBN: 9781450357067. DOI: 10.1145/3204493.3204530. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3204493.3204530.

50. M. A. Just and P. A. Carpenter. "A theory of reading: From eye fixations to comprehension". *Psychological Review*. 1980, pp. 329–354. URL: https://doi.org/10.1037/0033-295X.87.4.329.

51. P. Kabbash and W. A. S. Buxton. "The "Prince" Technique: Fitts' Law and Selection Using Area Cursors". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '95. ACM Press/Addison-Wesley Publishing Co., Denver, Colorado, USA, 1995, pp. 273–279. ISBN: 0201847051. DOI: 10.1145/223904.223939. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/223904.223939.

52. M. Kassner, W. Patera, and A. Bulling. "Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction". *Adjunct Proc. ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 1151–1160. DOI: 10.1145/2638728.2641695.

53. M. Khamis, A. Hoesl, A. Klimczak, M. Reiss, F. Alt, and A. Bulling. "EyeScout: Active Eye Tracking for Position and Movement Independent Gaze Interaction with Large Public Displays". *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Association for Computing Machinery, Québec City, QC, Canada, 2017, pp. 155–166. ISBN: 9781450349819. DOI: 10.1145/3126594.3126630. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3126594.3126630.

54. M. Khamis, A. Kienle, F. Alt, and A. Bulling. "GazeDrone: Mobile Eye-Based Interaction in Public Space Without Augmenting the User". *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. DroNet'18. Association for Computing Machinery, Munich, Germany, 2018, pp. 66–71. ISBN: 9781450358392. DOI: 10.1145/3213526.3213539. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3213526.3213539.

55. C. Koch and S. Ullman. "Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry". In: *Matters of Intelligence: Conceptual Structures in Cognitive Neuroscience*. Ed. by L. M. Vaina. Springer Netherlands, Dordrecht, 1987, pp. 115–141. ISBN: 978-94-009-3833-5. DOI: 10.1007/978-94-009-3833-5_5. URL: https://doi.org/10.1007/978-94-009-3833-5_5.

56. J. A. Krisch. *Why You Hated Clippy, That Annoying Microsoft Paperclip*. 2016. URL: https://www.vocativ.com/319986/clippy-digital-assistant/index.html.

57. C. Kumar, R. Menges, D. Müller, and S. Staab. "Chromium Based Framework to Include Gaze Interaction in Web Browser". *Proceedings of the 26th International Conference on World Wide Web Companion*. WWW '17 Companion. International World Wide Web Conferences Steering Committee, Perth, Australia, 2017, pp. 219–223. ISBN: 9781450349147. DOI: 10.1145/3041021.3054730. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3041021.3054730.

58. K. Kurzhals, F. Heimerl, and D. Weiskopf. "ISeeCube: Visual Analysis of Gaze Data for Video". *Proceedings of the Symposium on Eye Tracking Research and Applications*. ETRA '14. Association for Computing Machinery, Safety Harbor, Florida, 2014, pp. 43–50. ISBN: 9781450327510. DOI: 10.1145/2578153.2578158. URL: https://doi.org/10.1145/2578153.2578158.

59. C. Lander, S. Gehring, A. Krüger, S. Boring, and A. Bulling. "GazeProjector: Accurate gaze estimation and seamless gaze interaction across multiple displays". *Proc. ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 395–404. DOI: 10.1145/2807442.2807479.

60. C. Lander, S. Gehring1, A. Krüger, S. Boring, and A. Bulling. "GazeProjector: Accurate Gaze Estimation and Seamless Gaze Interaction Across Multiple Displays". *UIST '15*. 2015.

61. L. Larsson, M. Nystro¨m, and M. Stridh. "Discrimination of fixations and smooth pursuit movements in high-speed eye-tracking data". *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2014, pp. 3797–3800. DOI: 10.1109/EMBC.2014.6944450.

62. D. J. Liebling and S. T. Dumais. "Gaze and Mouse Coordination in Everyday Work". *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. UbiComp '14 Adjunct. Association for Computing Machinery, Seattle, Washington, 2014, pp. 1141–1150. ISBN: 9781450330473. DOI: 10.1145/2638728.2641692. URL: https://doi.org/10.1145/2638728.2641692.

63. P. Majaranta, U.-K. Ahola, and O. Špakov. "Fast Gaze Typing with an Adjustable Dwell Time". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Association for Computing Machinery, Boston, MA, USA, 2009, pp. 357–360. ISBN: 9781605582467. DOI: 10.1145/1518701.1518758. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/1518701.1518758.

64. S. Mallick. *Homography Examples using OpenCV (Python / C ++)*. 1999. URL: https://learnopencv.com/homography-examples-using-opencv-python-c/.

65. D. Mardanbegi and D. W. Hansen. "Mobile gaze-based screen interaction in 3D environments". *Proc. Conf. on Novel Gaze-Controlled Applications*. ACM, 2011. DOI: 10.1145/1983302.1983304.

66. N. Marquardt, R. Diaz-Marino, S. Boring, and S. Greenberg. "The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies". *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Association for Computing Machinery, Santa Barbara, California, USA, 2011, pp. 315–326. ISBN: 9781450307161. DOI: 10.1145/2047196.2047238. URL: https://doi.org/10.1145/2047196.2047238.

67. J. Matejka, F. Anderson, and G. Fitzmaurice. "Dynamic Opacity Optimization for Scatter Plots". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2015, pp. 2707–2710. ISBN: 9781450331456. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2702123.2702585.

68. L. E. Matzen, M. J. Haass, K. M. Divis, Z. Wang, and A. T. Wilson. "Data Visualization Saliency Model: A Tool for Evaluating Abstract Data Visualizations". *IEEE Transactions on Visualization and Computer Graphics* 24:1, 2018, pp. 563–573. DOI: 10.1109/TVCG.2017.2743939.

69. E. D. Michel Marie Deza. *Encyclopedia of Distances*. Springer, Berlin, Heidelberg, 2013. Chap. Computer-Related Distances, pp. 339–347.

70. D. Model and M. Eizenman. "A General Framework for Extension of a Tracking Range of User-Calibration-Free Remote Eye-Gaze Tracking Systems". *Proceedings of the Symposium on Eye Tracking Research and Applications*. ETRA '12. Association for Computing Machinery, Santa Barbara, California, 2012, pp. 253–256. ISBN: 9781450312219. DOI: 10.1145/2168556.2168609. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2168556.2168609.

71. M. E. Mott, S. Williams, J. O. Wobbrock, and M. R. Morris. "Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times". *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Association for Computing Machinery, Denver, Colorado, USA, 2017, pp. 2558–2570. ISBN: 9781450346559. DOI: 10.1145/3025453.3025517. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3025453.3025517.

72. A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Inc., USA, 1992. ISBN: 0471934305.

73. M. Okoe, S. S. Alam, and R. Jianu. "A Gaze-enabled Graph Visualization to Improve Graph Reading Tasks". *Computer Graphics Forum* 33:3, 2014, pp. 251–260. DOI: https://doi.org/10.1111/cgf.12381. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12381. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12381.

74. A. Ottley, R. Garnett, and R. Wan. "Follow The Clicks: Learning and Anticipating Mouse Interactions During Exploratory Data Analysis". *Computer Graphics Forum* 38, 2019, pp. 41–52. DOI: 10.1111/cgf.13670.

75. L. M. K. Padilla, S. C. Castro, P. S. Quinan, I. T. Ruginski, and S. H. Creem-Regehr. "Toward Objective Evaluation of Working Memory in Visualizations: A Case Study Using Pupillometry and a Dual-Task Paradigm". *IEEE Transactions on Visualization and Computer Graphics* 26:1, 2020, pp. 332–342. DOI: 10.1109/TVCG.2019.2934286.

76. M. Parisay, C. Poullis, and M. Kersten-Oertel. "FELiX: Fixation-based Eye Fatigue Load Index A Multi-factor Measure for Gaze-based Interactions". *2020 13th International Conference on Human System Interaction (HSI)*. 2020, pp. 74–81. DOI: 10.1109/HSI49210.2020.9142677.

77. N. Pears, D. G. Jackson, and P. Olivier. "Smart Phone Interaction with Registered Displays". *IEEE Pervasive Computing* 8:2, 2009, pp. 14–21. DOI: 10.1109/MPRV.2009.35.

78. C. M. Privitera and L. W. Stark. "Algorithms for defining visual regions-of-interest: comparison with eye fixations". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22:9, 2000, pp. 970–982. DOI: 10.1109/34.877520.

79. K. Rayner. "Eye Movements and Cognitive Processes in Reading, Visual Search, and Scene Perception". In: *Eye Movement Research*. Ed. by J. M. Findlay, R. Walker, and R. W. Kentridge. Vol. 6. Studies in Visual Information Processing. North-Holland, 1995, pp. 3–22. DOI: https://doi.org/10.1016/S0926-907X(05)80003-0. URL: https://www.sciencedirect.com/science/article/pii/S0926907X05800030.

80. E. Rosten and T. Drummond. "Fusing points and lines for high performance tracking". *IEEE Int. Conf. on Computer Vision*. Vol. 2. IEEE. 2005, pp. 1508–1515.

81. E. Rosten and T. Drummond. "Machine learning for high-speed corner detection". *European Conf. on Computer Vision*. Springer. 2006, pp. 430–443.

82. E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". *Proc. Int. Conf. on Computer Vision*. IEEE CS, 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

83. J. San Agustin, J. P. Hansen, and M. Tall. "Gaze-Based Interaction with Public Displays Using off-the-Shelf Components". *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing - Adjunct*. UbiComp '10 Adjunct. Association for Computing Machinery, Copenhagen, Denmark, 2010, pp. 377–378. ISBN: 9781450302838. DOI: 10.1145/1864431.1864444. URL: https://doi.org/10.1145/1864431.1864444.

84. J. San Agustin, H. Skovsgaard, J. P. Hansen, and D. W. Hansen. "Low-Cost Gaze Interaction: Ready to Deliver the Promises". *CHI '09 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '09. Association for Computing Machinery, Boston, MA, USA, 2009, pp. 4453–4458. ISBN: 9781605582474. DOI: 10.1145/1520340.1520682. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/1520340.1520682.

85. A. Santella and D. DeCarlo. "Robust Clustering of Eye Movement Recordings for Quantification of Visual Interest". *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. Association for Computing Machinery, San Antonio, Texas, 2004, pp. 27–34. ISBN: 1581138253. DOI: 10.1145/968363.968368. URL: https://doi.org/10.1145/968363.968368.

86. J. Sardegna, S. Shelly, and A. Rutzen. *The Encyclopedia of Blindness and Vision Impairment*. Facts on File Series. Facts on File, 2002. ISBN: 9780816042807. URL: https://books.google.com/books?id=rjZluAAACAAJ.

87. M. Sasaki, T. Nagamatsu, and K. Takemura. "Cross-Ratio Based Gaze Estimation for Multiple Displays Using a Polarization Camera". *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. Association for Computing Machinery, New Orleans, LA, USA, 2019, pp. 1–3. ISBN: 9781450368179. DOI: 10.1145/3332167.3357095. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3332167.3357095.

88. B. Serim and G. Jacucci. "Explicating "Implicit Interaction": An Examination of the Concept and Challenges for Research". *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Association for Computing Machinery, Glasgow, Scotland Uk, 2019, pp. 1–16. ISBN: 9781450359702. DOI: 10.1145/3290605.3300647. URL: https://doi.org/10.1145/3290605.3300647.

89. J. S. Shell, R. Vertegaal, and A. W. Skaburskis. "EyePliances: Attention-Seeking Devices That Respond to Visual Attention". *CHI '03 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '03. Association for Computing Machinery, Ft. Lauderdale, Florida, USA, 2003, pp. 770–771. ISBN: 1581136374. DOI: 10.1145/765891.765981. URL: https://doi.org/10.1145/765891.765981.

90. B. Steichen, G. Carenini, and C. Conati. "User-Adaptive Information Visualization: Using Eye Gaze Data to Infer Visualization Tasks and User Cognitive Abilities". *Proceedings of the 2013 International Conference on Intelligent User Interfaces*. IUI '13. Association for Computing Machinery, Santa Monica, California, USA, 2013, pp. 317–328. ISBN: 9781450319652. DOI: 10.1145/2449396.2449439. URL: https://doi.org/10.1145/2449396.2449439.

91. S. Stellmach and R. Dachselt. "Look & Touch: Gaze-Supported Target Acquisition". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Association for Computing Machinery, Austin, Texas, USA, 2012, pp. 2981–2990. ISBN: 9781450310154. DOI: 10.1145/2207676.2208709. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2207676.2208709.

92. S. Stellmach and R. Dachselt. "Still Looking: Investigating Seamless Gaze-Supported Selection, Positioning, and Manipulation of Distant Targets". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Association for Computing Machinery, Paris, France, 2013, pp. 285–294. ISBN: 9781450318990. DOI: 10.1145/2470654.2470695. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2470654.2470695.

93. S. Stellmach, L. Nacke, and R. Dachselt. "3D Attentional Maps: Aggregated Gaze Visualizations in Three-Dimensional Virtual Environments". *Proceedings of the International Conference on Advanced Visual Interfaces*. AVI '10. Association for Computing Machinery, Roma, Italy, 2010, pp. 345–348. ISBN: 9781450300766. DOI: 10.1145/1842993.1843058. URL: https://doi.org/10.1145/1842993.1843058.

94. "System Software for Ubiquitous Computing". *IEEE Pervasive Computing* 5:01, 2002, pp. 70–81. ISSN: 1558-2590. DOI: 10.1109/MPRV.2002.993146.

95. R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

96. J. Turner, A. Bulling, and H. Gellersen. "Extending the Visual Field of a Head-Mounted Eye Tracker for Pervasive Eye-Based Interaction". *Proceedings of the Symposium on Eye Tracking Research and Applications*. ETRA '12. Association for Computing Machinery, Santa Barbara, California, 2012, pp. 269–272. ISBN: 9781450312219. DOI: 10.1145/2168556.2168613. URL: https://doi.org/10.1145/2168556.2168613.

97. R. Vertegaal and J. Shell. "Attentive user interfaces: The surveillance and sousveillance of gaze-aware objects". *Social Science Information* 47, 2008. DOI: 10.1177/0539018408092574.

98. M. Vidal, A. Bulling, and H. Gellersen. "Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets". *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '13. Association for Computing Machinery, Zurich, Switzerland, 2013, pp. 439–448. ISBN: 9781450317702. DOI: 10.1145/2493432.2493477. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2493432.2493477.

99. C. Ware and H. H. Mikaelian. "An Evaluation of an Eye Tracker as a Device for Computer Input". *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*. CHI '87. Association for Computing Machinery, Toronto, Ontario, Canada, 1986, pp. 183–188. ISBN: 0897912136. DOI: 10.1145/29933.275627. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/29933.275627.

100. A. Worden, N. Walker, K. Bharat, and S. Hudson. "Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons". *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. CHI '97. Association for Computing Machinery, Atlanta, Georgia, USA, 1997, pp. 266–271. ISBN: 0897918029. DOI: 10.1145/258549.258724. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/258549.258724.

101. D. Yu, X. Lu, R. Shi, H.-N. Liang, T. Dingler, E. Velloso, and J. Goncalves. "Gaze-Supported 3D Object Manipulation in Virtual Reality". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2021. ISBN: 9781450380966. URL: https://doi.org/10.1145/3411764.3445343.

102. L. H. Yu and M. Eizenman. "A new methodology for determining point-of-gaze in head-mounted eye tracking systems". *IEEE Transactions on Biomedical Engineering* 51:10, 2004, pp. 1765–1773. DOI: 10.1109/TBME.2004.831523.

103. Y. Zhang, A. Bulling, and H. Gellersen. "SideWays: A Gaze Interface for Spontaneous Interaction with Situated Displays". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Association for Computing Machinery, Paris, France, 2013, pp. 851–860. ISBN: 9781450318990. DOI: 10.1145/2470654.2470775. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2470654.2470775.

104. Y. Zhang, J. Müller, M. K. Chong, A. Bulling, and H. Gellersen. "GazeHorizon: Enabling Passers-by to Interact with Public Displays by Gaze". *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '14. Association for Computing Machinery, Seattle, Washington, 2014, pp. 559–563. ISBN: 9781450329682. DOI: 10.1145/2632048.2636071. URL: https://doi-org.uproxy.library.dc-uoit.ca/10.1145/2632048.2636071.

105. Zyxwv99. *File:Field of view.svg — Wikimedia Commons, the free media repository*. [Online; accessed 25-February-2021]. 2020. URL: %5Curl%7Bhttps://commons.wikimedia.org/wiki/File:Field_of_view.svg%7D.

106. Zyxwv99. *Peripheral vision of the human eye.svg — Wikimedia Commons, the free media repository*. [Online; accessed 25-February-2021]. 2020. URL: %5Curl%7Bhttps://commons.wikimedia.org/wiki/File:Peripheral_vision.svg%7D.