

# GazeSpace: Eye Gaze Controlled Content Spaces

Sven Laqua  
University College London  
Department of Computer Science  
Gower Street  
0044 (0)20 7679 0351  
s.laqua@cs.ucl.ac.uk

Shane Udaraka Bandara  
University College London  
Department of Computer Science  
Gower Street  
shaneucl@yahoo.co.uk

M. Angela Sasse  
University College London  
Department of Computer Science  
Gower Street  
0044 (0)20 7679 7212  
a.sasse@cs.ucl.ac.uk

## ABSTRACT

In this paper, we introduce GazeSpace, a novel system utilizing eye gaze to browse content spaces. While most existing eye gaze systems are designed for medical contexts, GazeSpace is aimed at able-bodied audiences. As this target group has much higher expectations for quality of interaction and general usability, GazeSpace integrates a contextual user interface, and rich continuous feedback to the user. To cope with real-world information tasks, GazeSpace incorporates novel algorithms using a more dynamic gaze-interest threshold instead of static dwell-times. We have conducted an experiment to evaluate user satisfaction and results show that GazeSpace is easy to use and a “fun experience”.

## Categories and Subject Descriptors

H5.2 [User Interfaces]: Input devices and strategies.

## General Terms

Design, Experimentation, Human Factors.

## Keywords

Pointing and Selection, Eye Tacking, Eye Gaze Interaction, Eye Controlled User Interfaces

## 1. INTRODUCTION

For over 20 years people have used keyboards and mice as primary input devices for personal computers. With continuous advances in technology, novel forms of interaction are emerging. With the shift towards mobile and ubiquitous computing, touch screens are used in many scenarios. People use a pen or finger to control tablet PCs, PDAs, smart phones or public terminals. With advances in eye-tracking technology [1, 2, 3, 7, 8 and 10] and projects that reduce implementation costs [9], eye gaze may be the next interaction technique to break into mainstream usage.

**Eye gaze interaction** is commonly regarded as a potential complement, if not a replacement for traditional input techniques. However, when discussing input techniques for computing systems it is crucial to distinguish two main steps in the interaction process. As today’s user interfaces (UI) mostly

apply a desktop metaphor, users are required to point (e.g.: move a mouse) and select (e.g.: press a mouse button). It is important to consider this distinction when analyzing potential eye gaze interaction techniques. Many existing eye gaze systems demonstrate satisfying results for gaze-based pointing (or gaze-pointing). However, this only fulfills half of the requirement for the described “point and select” interactions. Successfully implementing the selection part of the interaction is more challenging than the pointing part. Current eye gaze systems commonly make use of static dwell-times to achieve this second part of the interaction process [1, 6, 7 and 10].

## 2. BACKGROUND

Most existing eye gaze systems are designed for medical contexts (e.g.: Tobii P10), where they enhance quality of life for people with disabilities, who cannot use traditional input techniques. The most common applications for patients are eye-typing to communicate with their surroundings and means to control their environment (e.g.: light switches, or motor controls for wheel chair). With increasing accuracy (resolution of gaze pointing), flexibility (freedom of head movements) and decreasing costs [9], applications “will soon be practical for able-bodied users” [3]. But as able-bodied audiences have much higher expectations for quality of interaction and general usability, challenges to beat traditional input methods and user interfaces arise. This partly explains why much of existing research focuses on rather abstract tasks when evaluating prototypes using dwell-time selection [1, 6 and 10].

**Task complexity:** Testing simple selection tasks ensures that limited cognitive effort is required. Although one can argue that simple selection tasks help modeling future scenarios, e.g. of selecting menu items, the abstract nature of these experiments often excludes the *cognitive component*: people need to look at a number of elements, make a choice, and then select the appropriate element. Experiments testing selection based on colour or single letters [10] minimize cognitive load and thus simplify task complexity. We hypothesize that more complex information tasks work less efficiently with static dwell times (H1). As complex tasks involve unique mental processes, they might require tailored activation times.

Research on more realistic use-cases, involving more complex tasks, commonly combines gaze-pointing with alternative means for selection (e.g.: hotkeys [3], speech [8] or even EMG clicking [2]). Due to space restrictions, we only discuss the most recent one: Kumar’s EyePoint system [3] enables users to browse the World Wide Web by replacing mouse interactions with a combination of gaze-pointing and hotkeys for selection (using a keyboard). EyePoint proposes a “look-press-look-release action” to cope with accuracy limitations of current eye trackers when used with standard user interfaces. This incorporates (a) looking at an area of interest, (b) zooming into

this area by clicking a hotkey, (c) selecting the desired element within this scaled-up area by looking at it and (d) releasing the hotkey to confirm selection of the chosen element. The main benefit of this approach is its compatibility with standard desktop interfaces. However, this solution has a number of limitations: (1) It requires separate keys to control the interaction and (2) it complicates the interaction process by approx. factor 2: point-click (mouse) vs. look-press-look-release (EyePoint). Compared with traditional mouse-based interaction, EyePoint increased task completion time and resulted in much higher error rates [3].

### 3. THE GAZESPACE SYSTEM

We have developed GazeSpace to provide a simple eye gaze interface that offers an appealing alternative to using the mouse when browsing content spaces (e.g.: to read blogs and news, look at pictures or video clips).

**The user interface:** GazeSpace combines real-time gaze tracking with a minimalist contextual visualization. Building on a contextual user interface (see Focus-Metaphor [4, 5]), the screen layout allows for seamless exploration of information spaces. The novel layout also suggests being beneficial to overcome the Midas Touch problem, commonly found in eye gaze interaction systems. The main information area (content element) in the middle of the screen is surrounded by contextual navigation elements (see Figure 1). When selecting a contextual element, its state changes: It enlarges into a content element and moves to the centre of the screen replacing the previous element. This previously central element switches back to context state and moves to the periphery of the screen.



Figure 1. The GazeSpace interface

**Layered architecture:** A Tobii X50 eye-tracker and the Tobii Software Development Kit (SDK) have been used to develop the GazeSpace prototype. The SDK enables access to real-time gaze data through the Tobii Eye Tracker Components API (TetComp). GazeSpace is a 2-layer application consisting of a system layer and a visual layer. The *system layer* has been developed using Microsoft VisualBasic (VB) which enables real-time access to gaze data (50Hz results in 50 gaze points per second) through TetComp. The *visual layer* has been developed using Adobe Flash (formerly Macromedia). The Flash user interface (UI) is embedded within the VB application. Relevant information about a user's gaze and the state of the UI is communicated between VB and Flash (using Actionscript). The layered approach has been chosen for two reasons: (1) Flash alone is not capable of accessing system API's or DLL's, which are required to receive real-time gaze data, and (2) VisualBasic does not provide the flexibility for fast prototyping, which would enable a similar user experience as Flash does.

**Algorithms:** The GazeSpace prototype aims to enable people to explore information spaces entirely by using eye gaze. To cope with the increased task complexity, instead of using static dwell-time for activation, GazeSpace integrates a more flexible *gaze-interest-threshold*. This approach comes with another benefit: it is more robust (if the eye tracker loses the user's gaze, the last working state of the interface can be "frozen"). Visual feedback ensures that the user is aware of the problem (this will be explained in detail in the feedback section below). After repositioning, she can return to the last working state. Instead of using fixations, the implemented algorithms work on the actual raw data, delivered through TetComp. Two variations of this technique have been implemented:

The **static interest accumulation algorithm** (SIA) collects gaze over each target area (navigational elements) until a predefined threshold is reached for the first element. When the user moves away from one target area before threshold is reached, the interest counter remains at this level until (A) the user comes back and new interest is added, or (B) another target area reaches threshold (gets activated). In case B, the interest counter is set to 0. The logic behind this algorithm aims to follow natural decision making processes, where a user looks at certain navigational choices before deciding which element to activate. After getting a first impression about relevant elements (by gazing at them), the user's actual decision to activate one of the elements should be faster. Accordingly, by collecting interest (gaze) along this decision making process for each element, reaching the interest-threshold will be faster.

The **dynamic interest decay algorithm** (DID) works similar to the first algorithm (SIA). However, instead of simply stopping the interest counter for an element when a user looks elsewhere, an additional decay function is used to decrease the collected interest for each element based on a predefined timing value (e.g.: accumulated interest is reduced by 50% every 50 collected gaze points outside the target area). This approach follows the logic, that when a user is not looking at an element for a longer amount of time (after first focusing on it), she "loses interest". After a few seconds, information about the previous element will have left short-term memory and thus more time is required again to make a decision (longer activation time).

**Integrated user interface feedback:** Existing eye gaze systems provide a very basic appearance, with usability and aesthetics being subordinate to functionality. GazeSpace is designed to improve usability by providing continuous visual feedback to the user. Gaze-pointing and activation are facilitated through a *coloured border* for each navigational element, providing feedback through a *dynamic gaze-over state* (similar to mouse-over state). Being transparent in neutral state, colour intensity increases continuously whilst a user is looking at an element. When reaching interest-threshold (by collecting a user's gaze), colour intensity approaches 100%, signaling the user that the element is about to be selected (see Figure 2).



Figure 2. Contextual UI element in neutral state (left) and after collecting "attention" through gaze (right)

As all stationary (not head-mounted) eye tracking systems can only cope with moderate head movements, it cannot be guaranteed that the user's eye will be tracked the whole time. We believe that providing feedback, on whether the system is "active" or not will enhance overall usability of the eye gaze system. For this reason, GazeSpace integrates a *coloured background* providing visual feedback over the general tracking state of the system: *light red – no gaze found, light green – gaze tracked*.



Figure 3. Experimental setup for the GazeSpace experiment

## 4. EXPERIMENT

We conducted an experiment in the lab to measure system performance and user satisfaction with the GazeSpace prototype (see Figure 3). 20 participants took part in the experiment. The average age was 24.2 (age range 18 – 50) and the sample was balanced for gender (50% female).

Besides collecting general user feedback and testing system performance of the prototype, we compared the two algorithms discussed earlier ( $A_1$ : SIA and  $A_2$ : DID algorithm). In order to test the algorithms using a between-subjects design, we created two separate content spaces:  $C_1$ : 7 UI elements providing information on ePassports (see Figure 1), and  $C_2$ : 7 UI elements providing information on the government's new ID card scheme. Both content spaces were designed to create a similar appearance and offer similar amounts of pictorial and textual information. Pairing of algorithm and content space has been counterbalanced (combinations:  $A_1C_1$ ,  $A_1C_2$ ,  $A_2C_1$ ,  $A_2C_2$ ). Each participant conducted two sessions (e.g.:  $A_1C_2$  and  $A_2C_1$ ), ordering of sessions has been counterbalanced.

After calibration, participants could familiarize themselves with the GazeSpace system using different content. They also received a brief introduction into how the system works and what the visual feedback means. After this introductory phase, each session lasted approx. 10 minutes during which participants were given information tasks (e.g.: "What kind of personal information does the chip on the ePassport hold?") by the experimenter. The aim was to facilitate task-driven exploration of the content spaces. After conducting both sessions, participants had to fill in a questionnaire evaluating the system. They were also asked whether they found a difference in the interface (apart from content) and if yes, which of the sessions they preferred.

### 4.1 Results

The following analysis focuses on qualitative findings. With the aim to develop an appealing and simple to use eye gaze system, collecting meaningful user feedback on GazeSpace has been very important in this experiment.

#### 4.1.1 User Satisfaction

To capture participants' overall reaction to GazeSpace, 6 questions from the "Questionnaire for User Interface Satisfaction" (QUIS) have been used (e.g.: "The system was: Frustrating – Satisfying"). In addition, aspects of ease of use, accuracy, etc. were evaluated using 25 tailored usability questions. A 6-point Likert scale has been adopted throughout. Figure 4 shows the clustered and normalized results.

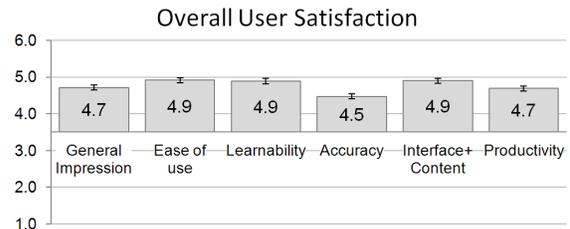


Figure 4. Questionnaire results

Participants consistently rated the GazeSpace prototype positive and particularly high for ease of use and learnability. This correlates with participants explicitly stating that GazeSpace is "very easy to use", "easy to understand", "self explanatory" and a "fun experience". However, there were of course also negative comments such as "speed of activation (not sufficient)", "eyes got tired after a while", "time to activate a bit long". The variety of comments shows that perception of the interaction experience using GazeSpace differed quite a lot between participants. The main aspect varying strongly was interaction speed. This finding might support H1 arguing that more complex information tasks are better supported by tailored activation times. The user interface feedback (coloured border and background) created diverse responses: Some found it useful; others found it "annoying". The analysis of the related usability questions showed that participants rated the coloured border feedback significantly higher than the coloured background feedback ( $t_{19} = 2.65$ ,  $p < 0.01$ ). What also became clear after conducting the experiment was that despite using a state-of-the art eye tracker, quality of calibrations and tracking ability varied strongly. While some participants were able to run a complete 10 minute session without hardly any interruptions of the gaze tracking, others had to cope with constant interruptions. This obviously biased participants' feedback. We therefore conducted a separate analysis of the questionnaire feedback using vision as independent variable (see next section).

#### 4.1.2 Impact of vision

Throughout all categories of user feedback, differences have been found when comparing feedback for participants with normal vision (7) and participants with corrected vision (13, using glasses or contact lenses). These differences have been significant for *general impression* ( $t_{18} = 2.06$ ,  $p < 0.027$ ), *learnability* ( $t_{18} = 2.06$ ,  $p < 0.027$ ), *accuracy* ( $t_{18} = 2.50$ ,  $p < 0.012$ ) and *Interfaces & Content* ( $t_{18} = 2.16$ ,  $p < 0.023$ ) (see Figure 5).

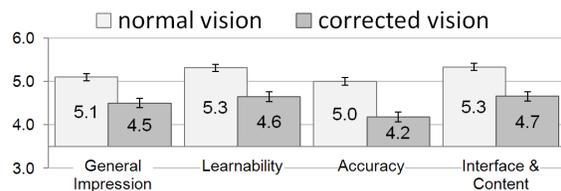


Figure 5. Impact of vision on user satisfaction

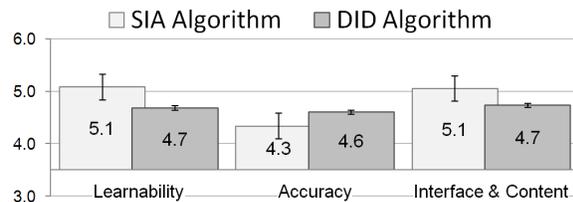
### 4.1.3 Impact of Algorithms

The comparison of the two tested algorithms (SIA and DID) revealed a higher preference for the static interest accumulation algorithm (see Table 1). Interestingly, 35% of participants did not notice any difference in the user interface of the two sessions (apart from content). Participants favoring the SIA algorithm usually stated faster interactions as the reason.

**Table 1. User preference of prototype versions**

no difference noticed	35%
With static algorithm (SIA)	45%
With dynamic algorithm (DID)	20%

When looking at the questionnaire feedback, using algorithms as independent variable, superior ratings for the static interest accumulation (SIA) algorithm for *learnability* and for *interface & content* have been found (see Figure 6). Although the SIA algorithm was preferred, participants surprisingly rated it less accurate. However, it needs to be noted that these differences are not significant.



**Figure 6. User satisfaction based on algorithm**

## 5. CONCLUSIONS

We have not compared the GazeSpace system with traditional mouse-based input, as the main aim of this early study was to investigate initial user responses to the GazeSpace prototype. Results show that participants throughout liked user interface and interaction technique. As the most common complaint mentioned has been the speed of interaction, future work should test lower activation thresholds (“if faster ... it would be fun and relaxing”). Many participants preferred the SIA algorithm, as it resulted in faster activations. We consciously decided for longer activation thresholds in the beginning of the experiment to prevent high error rates. These could have negatively biased participant’s general evaluation of this novel interaction technique. When using static dwell times for selection, speed of interaction and error rates will always influence each other. To compete with traditional input techniques, future eye gaze systems might require adaptive selection algorithms which can be tailored to individual users and task complexity.

Participants’ comments such as “aesthetically pleasing”, “user friendly” and “nice interface” further suggest that the user experience will be crucial when targeting future eye gaze systems at able-bodied audiences. The design of the user interfaces should complement the novelty of the interaction technique and rich feedback on the system’s state will further complement the user experience. Although feedback on the general tracking state seems to be quite useful, findings suggest a redesign into a more subtle and less disturbing feedback feature.

Looking at algorithm performance mainly revealed a difference in interaction speed. Although not in the focus of this paper, some initial eye tracking analysis suggests no significant differences in fixation counts, gaze time or fixation durations

between SIA and DID. Further studies and more detailed analysis of the eye tracking data might enable a more detailed discrimination. It also appears crucial to experiment with different settings for gaze-interest-thresholds and timing values for the decay function.

## 6. FUTURE WORK

Besides improving current algorithms, further development of the GazeSpace prototype will focus on redesigning the layered architecture of the application to enable direct web access. Achieving this milestone will result in the ability to browse “live” blog spaces simply through eye gaze. To enable realistic blog-reading, GazeSpace will require the integration of scrolling functionality to cope with larger amounts of text. Previous research has confirmed that this is a feasible feature [3]. Another crucial aspect for the development of future eye gaze systems will be the required effort for correcting a false interaction. The easier it will be to correct an error (wrong selection), the more tolerant users should be.

## 7. REFERENCES

- [1] Hansen, J. P. et al. Command Without a Click: Dwell Time Typing by Mouse and Gaze Selections. In *Proceedings of Ninth IFIP TC13 International Conference on Human-Computer Interaction (Interact '03)* (Zurich, Switzerland, Sept 1-5, 2003)
- [2] Junker, A. M. and Hansen, J. P. Gaze Pointing and facial EMG clicking. In *2<sup>nd</sup> Conference on Communication by Gaze Interaction (COGAIN 2006)* (Turin, Italy, Sept 4-5, 2006)
- [3] Kumar, M. et al. EyePoint: Practical Pointing and Selection Using Gaze and Keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07)* (San Jose, CA, 2007).
- [4] Laqua, S. and Brna, P. The Focus-Metaphor Approach: A Novel Concept for the Design of Adaptive and User-Centric Interfaces. In *Proc. Interact '05*, (Rome, Italy, 2005), 295-308.
- [5] Laqua, S., Ogbechie, N. and Sasse, M. A. Contextualizing the Blogosphere: A Comparison of Traditional and Novel User Interfaces for the Web. In *Proc. 21st BCS HCI Group Conference (HCI '07)* (Lancaster, UK, 2007).
- [6] Majoranta, P. et al. Effects of Feedback on Eye Typing with a Short Dwell Time. In *Proceedings of the Eye Tracking Research & Application Symposium (ETRA 2004)* (San Antonio, TX, 2004).
- [7] Miniotas, D. et al. Eye Gaze Interaction with Expanding Targets. In *CHI'04 extended abstracts on Human factors in computing systems* (Vienna, Austria, 2004), 1255-1258.
- [8] Miniotas, D. et al. Extending the Limits for Gaze Pointing through the Use of Speech. In *Information Technology and Control*, 2005, Vol. 34, No. 3, 225-230.
- [9] openEyes. <http://hcvl.hci.iastate.edu/cgi-bin/openEyes.cgi>, last accessed on 24 May 2007.
- [10] Silbert, L. E. and Jacob, R. J. K. Evaluation of Eye Gaze Interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '00)* (The Hague, Netherlands, 2000), 281-288.
- [11] Tobii P10. <http://www.tobii.com/default.asp?sid=553>, last accessed on 24 May 2007.