Eye-Controlled Interfaces for Multimedia Interaction

Chandan Kumar, Raphael Menges, and Steffen Staab
Institute for Web Science and Technologies (WeST)

In the digitized world, interacting with multimedia information occupies a large portion of everyday activities; it's now an essential part of how we gather knowledge and communicate with others. It involves several operations, including selecting, navigating through, and modifying multimedia, such as text, images, animations, and videos. These operations are usually performed by devices such as a mouse or keyboard, but people with motor disabilities often can't use such devices. This limits their ability to interact with multimedia content and thus excludes them from the digital information spaces that help us stay connected with families, friends, and colleagues.

When we interact with multimedia—which presents rich visual content—our eyes process relevant information. Eye activities can be tracked, and gaze tracking has been successfully used to analyze and evaluate cognition during multimedia learning.[1] So far, however, tracking of this visual channel has rarely been exploited to control interaction with multimedia information. Seminal work has started to employ eye-tracking technology for controlling interactions, targeting two major paradigms for interaction by eye tracking:[2] direct control and implicit observations.

Direct control refers to the deliberate, explicit use of eye movements by the user with the intention that such gaze signals are picked up for interaction commands, such as selecting, moving, or modifying an object or defining a new one—by text input, for example. *Implicit observations* of gaze signals have been used to enhance the viewing activity (such as for extended reading, as in Text 2.0[3] or to capture the varying importance of multimedia visual content[4]).

Direct control is effective but very slow and tiring for the human user. Implicit observations remain unobtrusive but can at best be considered a weak signal of what the user wants to accomplish. Thus, direct control through gaze tracking should be supplemented and merged with interactions derived from implicit observation to remain effective while improving the user friendliness of eye-tracking technologies in multimedia interaction. In this context, the EU-funded MAMEM project (Multimedia Authoring and Management using your Eyes and Mind) aims to propose a framework for natural interaction with multimedia information for users who lack fine motor skills (see the related sidebar for more information).

Here, we primarily focus on the gaze-based control paradigm (see Figure 1) that we've developed as part of our work at the Institute for Web Science and Technologies (WeST) within the scope of MAMEM project. We outline the particular challenges of eye-controlled interaction with multimedia information, including initial project results. The objective is to investigate how eye-based interaction techniques can be made precise and fast enough to not only allow disabled people to interact with multimedia information but also make usage sufficiently simple and enticing such that healthy users might also want to include eye-based interaction.

Challenges of Eye Input

Eye-tracking systems measure a person's eye movements so that the gaze point is established at any point in time. Different invasive or non-invasive methods for eye movement measurement have been investigated to improve gaze data estimation.[2] Eye-tracking technology has evolved, increasing precision and decreasing cost. However, using an eye gaze as input remains a challenge, due to the limitation of the visual angle, calibration errors, the drift, and inherent eye jitter, as well as the fact that the gaze positions reported by eye trackers don't correspond exactly to where the user is looking.

1070-986X/16/$31.00 © 2016 IEEE

Multimedia Authoring and Management Using Your Eyes and Mind

The MAMEM project started in May 2015, and it consists of eight different partners (see www.mamem.eu/project/consortium) collaborating to deliver the technology that will let people operate software applications and execute multimedia-related tasks using their eyes and mind. MAMEM especially targets individuals with motor disabilities (such as people with Parkinson's disease, muscular disorders, and tetraplegia). The common symptom of these disorders is the loss of the voluntary muscular control (while preserving cognitive functions), leading to a variety of functional deficits, including the ability to operate applications that require the use of a conventional interaction medium (mouse, keyboard, touchscreens, and so on). As a result, the affected individuals are marginalized and can't keep up with the rest of the society in a digitized world.

MAMEM's aims to better integrate these people into society by endowing them with the critical skill of accessing multimedia information content using novel and more natural interface channels. MAMEM also aims to make its technology persuasive and provide the principles for designing interfaces that will motivate disabled people to use them.

Eye movements occur as sequences of very fast (less than 100 ms) saccades, followed by relatively stable fixation periods (100–600 ms). Even during fixations, however, the eye isn't completely still but is characterized by some jitters. In addition, even the precision of recent eye trackers is still limited, because the trackers can only determine the angle of the user's gaze in relation to the screen up to approximately 0.5 degrees (or roughly 10 pixels on a 17-inch display observed from 60 cm away). These factors imply a loss of stability, causing the "eye cursor" to momentarily leave the target intended by the user for further multimedia interaction. *Direct control* is impeded by such factors. Consequently, for reliable eye-controlled interfaces, smoothing mechanisms are necessary to stabilize the gaze signal. Furthermore, interface elements must be adopted to negotiate the impact of limited accuracy.

Another major difficulty for input control by gaze interaction is the double duty performed by eyes. When using an eye tracker for control, the "normal" course of events changes substantially, because the eyes are both acting as an important sensory channel and providing motor responses to control the computer.[5] Instead of the hand providing motor responses to control the computer through external physical devices, the eye provides motor response through virtual or graphical controls that appear on the system's display. The eyes are thus overloaded with explicit control tasks, especially in complex scenarios of multimedia exploration and modification, making the interaction tedious and error-prone. Therefore, it's imperative for eye-controlled interfaces to use implicit eye signals to predict user intentions and support the explicit control of tasks.

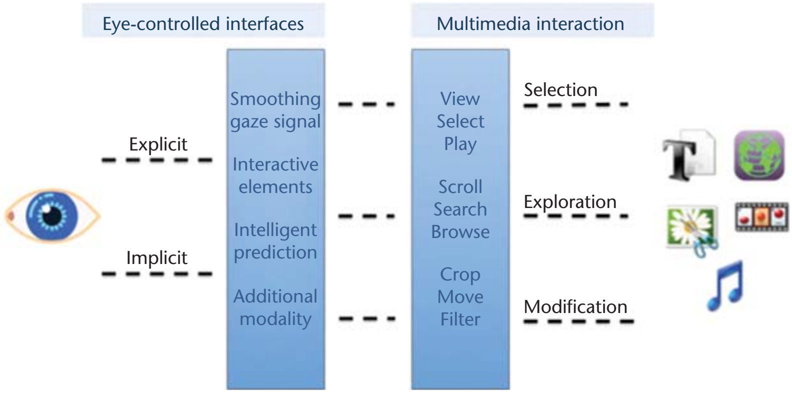


Figure 1. Eye-controlled interfaces for multimedia interaction. Examples include selecting, exploring, and modifying text, images, and videos. (Note that interaction with multimedia isn't limited to this set of operations.)

Furthermore, eyes engage in inspection and selection simultaneously while interacting with the interface. The most common method to distinguish between inspections and selections is to set a time threshold (or *dwell time*), with a click issued after the duration of the fixation exceeds a specified amount of time. The dwell time might lead to unintended activations, resulting from fixations used for inspection being confused with a selection. This issue is referred to as the *Midas-Touch problem*.[2] Appropriate visual feedback from the eye-controlled interfaces could play a vital role in helping users discriminate between inspection and selection. Moreover, an additional source of input confirmation could also be a significant measure for distinguishing between selections and inspections.

To address these challenges and increase the feasibility of eye-based interactions, we employ

Another significant element of eye-controlled interfaces is providing feedback to users with respect to their gaze activity.

the following four techniques: we smooth gaze signals to counteract eye jitter, design novel interface elements for eye-controlled interaction, add implicit intelligence to gaze-centered interfaces, and integrate additional input modalities.

Smoothing Signals

Researchers have pointed out the necessity of diminishing the effects of eye jitter to improve the stability of the eye cursor using smoothing. Oleg Špakov compared several eye-movement filters for use in HCI applications.[6] The comparison was based on the introduced delay, smoothness, and closeness to the idealized data. The study's outcome was that algorithms with state detection (fixation and saccade) and adapted processing generally performed better than others. To smooth the data from the eye tracker in real time, it's necessary to determine whether the most recent data point is the beginning of a saccade, a continuation of the current fixation, or an outlier relative to the current fixation. The x and y components of the raw gaze points are mapped according to two independent, linear functions.

For smoothing, we can also apply moving averages—that is, the average location of every *k* successive gaze points within a fixation window, where the fixations and saccades are treated separately. Manu Kumar and his colleagues presented a one-sided triangular filter to compute the fixation point as a weighted mean in the current fixation window.[7] They also applied two Kalman filters to process the eye-gaze data, one for the entire raw gaze data, and the other for the gaze data within fixation windows.

In addition to the smoothing effect, we can generate information about the start and end of fixations. This can be used to generate high-level gaze events. The algorithm proposed by Darius Miniotas and his colleagues, the *grab-and-hold algorithm,*[5] had the same effect as if the gaze were held on the desired target during periods of fixation, thus effectively reducing the probability of restarting the selection timer before the end of the dwell time.

Designing Interactive Interface Elements

Interfaces generally use their architecture to acclimatize to the type of input device used for operating the computer. The look and feel of the interface depends on the device selected for primary input. For example, when a mouse or keyboard is the primary input device, the interface-controllable elements, such as buttons, icons, menus, scrollbars, lists, and dialog boxes, will appear as they do in a conventional interface. However, when a different physical input device, such as an eye tracker, is the primary source, the look and feel of these elements must change to accommodate eye input.

Facilitating Accuracy

As noted, the gaze position acquired from eye-tracking devices doesn't exactly correspond to where the user is looking. This problem can be addressed with interface adaptations—such as enlarging targets when the eye-gaze interaction involves acquiring small targets. Moreover, when the user looks at the screen, the area around the gaze point can be linearly magnified and redisplayed in a zoom window.[8] A magnified view helps map gaze points to a desired target, so the desired actions can be easily and correctly performed. Distorting the gaze area around the desired target can also help systems better understand the user's gaze.

Another way to magnify a target is to temporarily expand the target itself rather than to zoom in on the area around it. When the user's gaze falls within the vicinity of the desired target, the target size could increase to include the gaze point into the enlarged target.

Visual Feedback

Another significant element of eye-controlled interfaces is providing feedback to users with respect to their gaze activity.[9] Most errors are induced by the lack of adequate feedback from the screen, because the slightest discrepancy between a user's eye movements and what he or she sees, feels, or hears can disrupt the experience. Designers thus must repurpose the feedback mechanisms for sensory information from eyes.

Using adequate visual graphics and animation as feedback could help users discriminate between inspections and activations and could reduce errors in interactive operations—for example, buttons get activated when the gaze hits them, and they shrink after activation to trigger the button. A colored overlay increasing in size would work as a visual representation of the remaining time until the trigger. Furthermore, the visual feedback shouldn't be unanimous for all kind of operations—that is, a scroll button should have a specific visual feedback indicating the page lengths that have been scrolled.

Figure 2 shows some examples of these elements. The top row in Figure 2a signifies a click emulation through eye-gaze interaction, where the animated highlighting over the icon shows the gaze duration, and the click is activated like a switch button. The bottom row indicates a sensor-like button, relevant for progressive elements like scrolling. Figure 2b shows the stages of eye typing, with the magnifying effect of character selection combined with the visual feedback of the user's gaze.

The discussed accuracy and interactive elements become essential to develop applications that use the eye as a direct control for interaction with multimedia information. In that regard, we have employed these heuristics in various multimedia interaction environments.

Figure 3 shows an example of gaze-adapted browsing (with GazeTheWeb), where the conventional browser interface is customized for eye gestures. The custom layout on the left and right indicates the enlarged graphical elements to select various browsing operations and visual feedback for interacting with these elements. The major central region is the Web view containing the content of the webpages. The left layout contains browser-centric functionality, such as opening new tabs, going backward and forward, and changing various browser settings. The right layout is for interacting with the webpage. The user can view, scroll, and navigate through the image and text content though gaze direction. To rapidly scroll up and down, dedicated buttons are used that act like sensors, where the visual saturation increases when the user looks at the buttons.

Clicking on images and text hyperlinks and navigating to different pages is an essential component of a user's Web-browsing behavior. For such navigational task, links might be very close to each other. Consequently, we applied the strategy of dynamically magnifying the observed screen portion when the user is looking at an intended hyperlink, so the link can be accurately selected within the enlarged page region. Similarly, Figure 4 shows the gaze-adapted Twitter application interface, where the user can perform all essential operations (such as tweeting, searching, following, or discovering) interactively via direct control through the eyes.

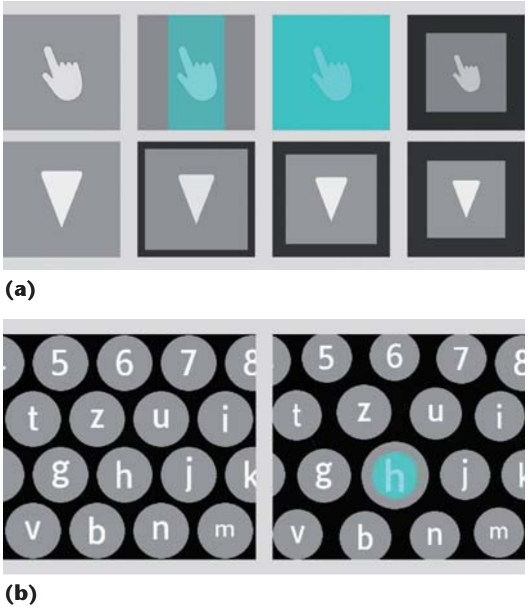


Figure 2. States of activation for interactive elements of eye-controlled interfaces: (a) visual feedback and (b) visual feedback with magnification on the right side of the screen.

Figure 5 shows the example of an eye-controlled interface for the game "Schau genau!" where gaze is used to control a butterfly. The player collects flowers and classifies photographs of flowers to earn points (see Figure 5a).[10] In this immersive game environment, several interaction elements were included with respect to the button size, shape, visual feedback, and so on. For example, Figure 5b shows the game screen space for the player inserting his name for his high score. The user can scroll horizontally through letters of the alphabet. The fixated letter enlarges until a dwell time is over and the letter is selected. If the player fixates another letter in the meantime, the old one is scaled back down. On the bottom of the game screen, the

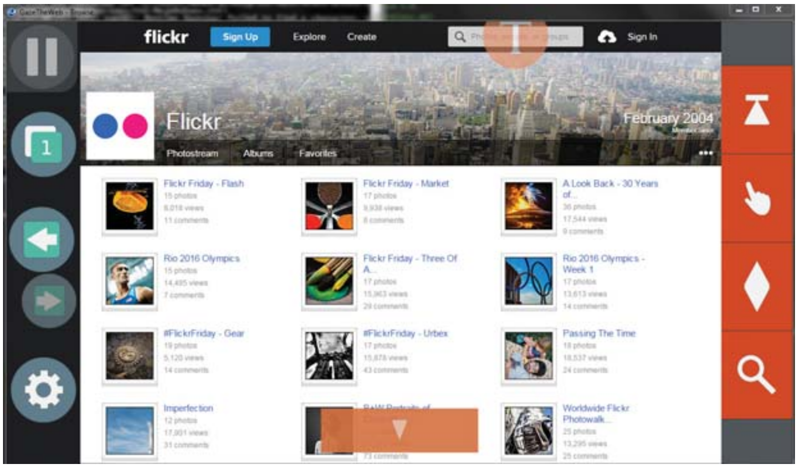
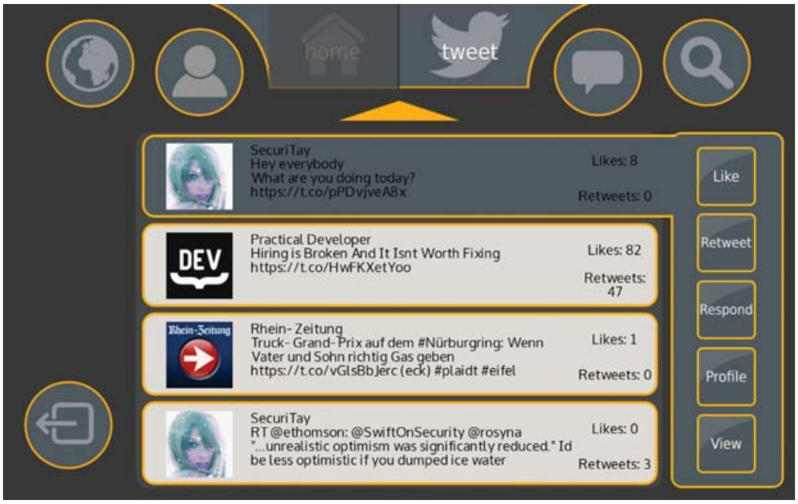


Figure 3. Eye-controlled Web browsing. Here, the user is exploring a collection of images.



SecuriTay
Hey everybody
What are you doing today?
https://t.co/pPDvjveA8x
Likes: 8
Retweets: 0

Practical Developer
Hiring is Broken And It Isnt Worth Fixing
https://t.co/HwFKXetYoo
Likes: 82
Retweets: 47

Rhein-Zeitung
Truck-Grand-Prix auf dem #Nürburgring: Wenn Vater und Sohn richtig Gas geben
https://t.co/vGlsBbJerc (eck) #plaidt #eifel
Likes: 1
Retweets: 0

SecuriTay
RT @ethomson: @SwiftOnSecurity @rosyna "...unrealistic optimism was significantly reduced." Id be less optimistic if you dumped ice water
Likes: 0
Retweets: 3

Figure 4. Eye-controlled social media browsing. Here, the user can select and interact with tweets.

player can either confirm the input or delete the latest written letter using individual selection buttons.

The interfaces in Figures 3–5 were well received by users (see the video demonstrations of the applications at https://west.uni-koblenz.de/en/research/mamem). The Schau genau! game was installed in a horticultural garden show in Landau (http://lgs-landau.de) in a stand-alone arcade cabinet. The game was played more than 2,900 times during the exhibition—a clear indication of its popularity and the acceptability of its adopted interaction elements. The Twitter interface shown in Figure 4 was highly appreciated by users in a lab study with 13 participants. The eye-controlled interface was compared against the conventional method of emulating a "mouse with eyes" (OptiKey; https://github.com/OptiKey/OptiKey/wiki), and it significantly outperformed the eye-mouse emulation in the metrics of system usability and mental workload.

Inspired by the success of these design and interaction elements, we proposed an *eyeGUI* library to design and develop GUIs suitable for eye-based input control. The eyeGUI library enables the manipulation and rendering of user interfaces for eye-tracking input.[11] A variety of elements, such as buttons, images, and text, can be used from the library to build a proper interface. All elements in eyeGUI are designed especially for eye tracking in terms of their size, appearance, and user interaction—for example, buttons get activated when the gaze hits them, and they shrink after activation to trigger the button. The eyeGUI library was developed in C++ 11 and is based on OpenGL. You can use it to build user interfaces for eye tracking by adding XML files as layouts and manipulating elements within these layouts via "listeners." The listeners can be accessed in the application environment to give every interface element its own functionality.

Adding Implicit Intelligence

Predicting a user's intent via implicit gaze signals can help enhance control functionality. Here, we describe how implicit observations might help with three example activities—scrolling, image search, and editing.

Scrolling

The act of scrolling is strongly coupled with the user's ability to engage with information via the visual channel. Therefore, the use of implicit eye-gaze information is a natural choice for enhancing multimedia content scrolling techniques.

For scrolling and reading, we propose using natural eye movements to control the motion of the windows for the user. In the GazeTheWeb interface, the user must explicitly activate automatic scrolling, turning it on and off by pressing a button using direct gaze control (see the right side of Figure 2a). In the auto scroll mode, the user has a smoother and more natural reading experience, because the scrolling is supported via implicit observation of user's gaze coordinates. The scroll direction is determined by noting the quadrant where the user is currently looking, outside a central neutral region. The scroll speed is proportional to the distance from the center of the screen.

Searching Images

Another focus of our work is to employ implicit gaze pattern for an improved search experience—that is, to make the GazeTheWeb interface more fluent and natural by collecting the feedback implicitly while users are involved in gaze-based browsing. Essentially, the user implicitly requests better images by zooming in, and this feedback is inferred from gaze data while the user looks at the images. This could significantly reduce the user's effort for explicitly refining search results.[4]

There have been some preliminary studies related to the use of implicit gaze information in image retrieval.[12] However, GazeTheWeb provides a more natural scenario of gathering implicit feedback to enhance the results while users are actively engaged in gaze-driven searching and browsing. For example, Figure 6a shows an image search scenario in which the user is browsing through results after searching for "plants." The user zooms in on the pictures for an enlarged view, providing implicit feedback on the region of interest. Based on the fixations and user attention, the system goal is to present refined results to support the user's interest (the desert plants shown in Figure 6b).

Editing

Gaze-based browsing not only offers a framework for eye-controlled access but also recognizes relevant signals regarding the user's region of interest. We plan to use such fixation data to enhance multimedia interaction—for example, to identify important content and use it for multimedia editing (such as image cropping). With the GazeTheWeb interface, users simply look at images while browsing, and we can use the gaze patterns to identify the important image content and automatically generate crops of any size or aspect ratio. The goal is to create appealing crops without explicit interaction. Furthermore, precise identification of relevant image content without explicit interaction is a vital feature. It lets us analyze and quantify the viewing behavior of images and how users select the region of interest for image editing. It also lets us analyze other useful functionalities, such as the automatic creation of snapshots or of thumbnails for adaptive Web documents.

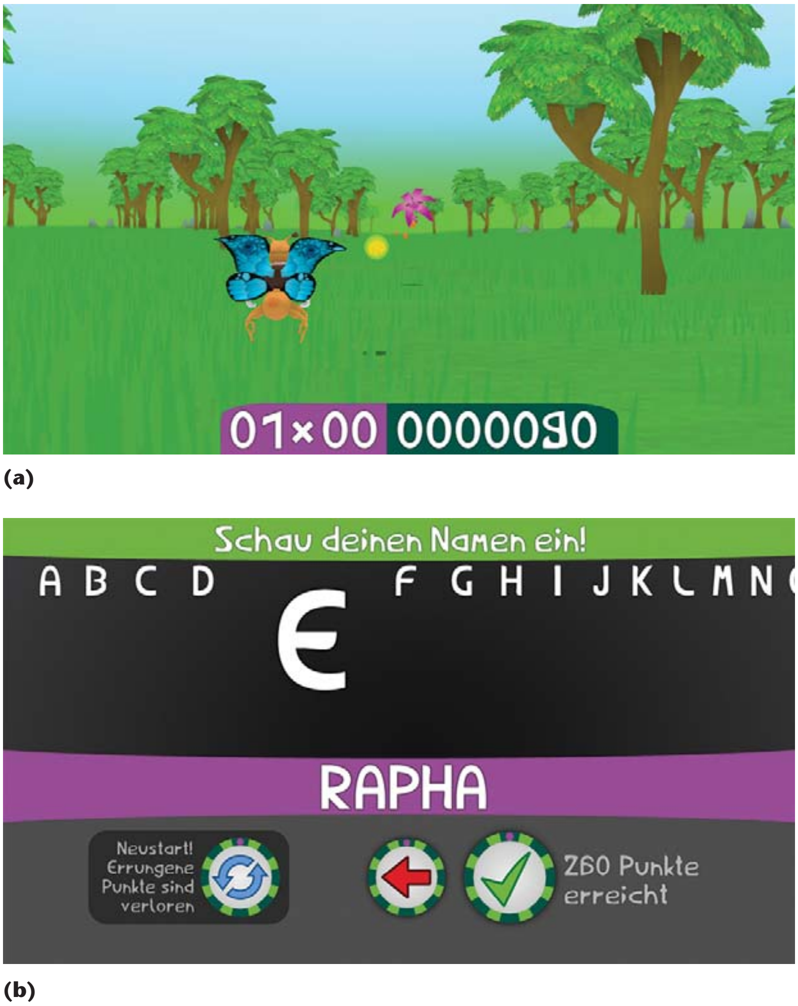

01×00 0000090
(a)
Schau deinen Namen ein!
A B C D E F G H I J K L M N
RAPHA
Neustart! Errungene Punkte sind verloren
260 Punkte erreicht
(b)

Figure 5. Eye-controlled gaming. The player (a) moves an animated object to select targets and (b) inserts a nickname for achieving a high score.

In this context, we have already conducted experiments and employed human fixation patterns to identify the most salient region of images, because defining a good crop requires a model that explicitly represents important image content. Our analysis of the Schau genau! game data implies that human fixations are very particular in identifying important image content. Figure 7 shows a sample image from the game data. The image on the far right is the visual saliency generated using eye fixations of players, which is more accurate compared to automatic saliency detection algorithms such as GBVS, Itti-Koch, and Signature saliency,[13] often used to generate automated crops of the picture.

Implicit gaze feedback is also relevant in personalizing the user experience—especially

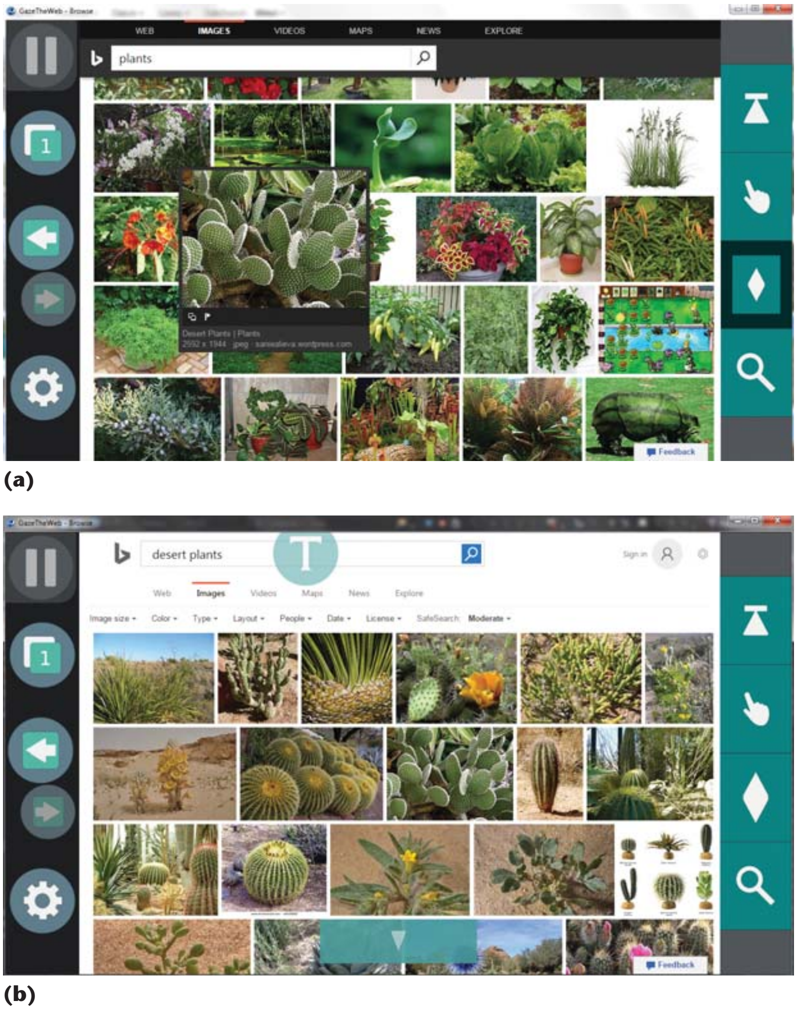


Figure 6. Gathering implicit feedback to enhance search results: After searching for plant images, the user (a) interacts with the results and zooms in on a desert plant. Based on this implicit observation of the user's gaze, the GazeTheWeb system (b) would re-rank the results and present desert plants.

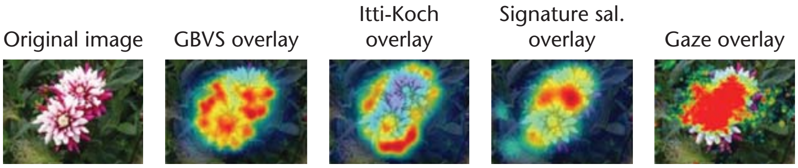

Original image | GBVS overlay | Itti-Koch overlay | Signature sal. overlay | Gaze overlay

Figure 7. Cropping: Saliency patterns of image using automated algorithms and human gaze pattern: the (a) original image and images with (b) the GBVS, (c) Itti-Koch, and (d) Signature saliency overlays and with the (e) gaze overlay.

given that the sensory input performance largely depends on the individual's ability to process the cognitive information (for example, some users might be able to easily select the target with smaller dwell times compared to the time needed by others).

Integrating Additional Modalities

To overcome with the Midas-Touch problem, researchers in the HCI field have proposed different multimodal systems integrated with eye-gaze input. One kind of multimodal system combines the user's gaze with a hardware button—such as the MAGIC technique for pointing tasks.[2] This technique first warps the cursor to the vicinity of the target when the user stares at and wants to select it. Then the user can use a manual-pointing device to confirm the selection. Currently, we offer a similar functionality in GazeTheWeb by integrating the keyboard with gaze input. The user can look at the desired target in the screen and press a predefined hotkey (the Enter key) for the desired action. Another kind of multimodal system fuses the gaze and speech.[7]

For people with motor disabilities who can't use an external physical input device, integrating other psycho-physiological signals seems more appropriate. Furthermore, eye signals attentively reflect brain activities—that is, where users are looking indicates what they're processing in their minds, and how long they're looking at something indicates how much processing effort is needed (one example is the eye-mind hypothesis[14]). In this regard, electroencephalogram data from Brain Computer Interface (BCI) devices provide insight into how the brain works and helps us understand stress and other neural artifacts that can be incorporated to enhance gaze-based interaction.

Brain signals from alpha and beta frequency bands, together with eye-tracking signals, could provide more control in a multimedia interaction environment. With our MAMEM partners, we're currently investigating several fusion techniques to enhance the performance of eye-controlled interfaces via BCI and biosensors. We're examining the sensorimotor rhythms (SMRs) to understand signals that indicate state changes while the user navigates on the GazeTheWeb browser for a reading (inspection) or selection task. This "switching" task or state change task is being tested with SMRs in conjunction with signals from the eye-tracking system to eradicate or reduce the Midas-Touch problem. Moreover, SMRs could support complex multimedia interaction tasks like image editing (rotating images with motor imagery actions).

Furthermore, we examine error related potential (ErrP), which offers a natural way to detect errors for automatic error correction (AEC) with the EEG sensor. ErrPs have been used for AEC in BCIs (such as to correct misspellings in P300 or cVEP-based spellers[15]) but not in combination with eye-controlled interfaces. At first, our focus will be to correct misspellings, which will be activated when an ErrP signal is detected. In the future, we'll use ErrP on the more complex error scenarios of multimedia interaction. **MM**

Acknowledgements

This work is part of project MAMEM, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 644780.

References

1. T. van Gog and K. Scheiter, "Eye Tracking as a Tool to Study and Enhance Multimedia Learning," *Learning and Instruction*, vol. 20, no. 2, 2010, pp. 95–99.
2. P. Majaranta and A. Bulling, "Eye Tracking and Eye-Based Human–Computer Interaction," *Advances in Physiological Computing*, Springer, 2014, pp. 39–65.
3. R. Biedert et al., "The Text 2.0 Framework: Writing Web-Based Gaze-Controlled Realtime Applications Quickly and Easily," *Proc. 2010 Workshop on Eye Gaze in Intelligent Human Machine Interaction*, 2010, pp. 114–117.
4. T.C. Walber, A. Scherp, and S. Staab, "Smart Photo Selection: Interpret Gaze as Personal Interest," *Proc. 32nd Ann. ACM Conf. Human Factors in Computing Systems*, 2014, pp. 2065–2074.
5. D. Miniotas, O. Špakov, and I.S. MacKenzie, "Eye Gaze Interaction with Expanding Targets," *CHI 04 Extended Abstracts on Human Factors in Computing Systems*, 2004, pp. 1255–1258.
6. O. Špakov, "Comparison of Eye Movement Filters Used in HCI," *Proc. Symp. Eye Tracking Research and Applications*, 2012, pp. 281–284.
7. M. Kumar et al., "Improving the Accuracy of Gaze Input," *Proc. 2008 Symp. Tracking Research & Applications*, 2007, pp. 65–58.
8. C. Lankford, "Effective Eye-Gaze Input into Windows," *Proc. 2000 Symp. Eye Tracking Research & Applications*, 2000, pp. 23–27.
9. P. Majaranta et al., "Auditory and Visual Feedback During Eye Typing," *CHI 03 Extended Abstracts on Human Factors in Computing Systems*, 2003, pp. 766–767.
10. C. Schaefer et al., "Schau genau!—An Eye Tracking Game with a Purpose," *Workshop on the Applications for Gaze in Games at CHI Play*, 2014.
11. R. Menges et al., "EyeGUI: A Novel Framework for Eye-Controlled User Interfaces," to be published in *Proc. 9th Nordic Conference on Human-Computer Interaction* (NordiCHI), 2016.
12. O. Oyekoya and F. Stentiford, "Perceptual Image Retrieval Using Eye Movements," *Int'l J. Computer Mathematics*, vol. 84, no. 9, 2007, pp. 1379–1391.
13. A. Borji and L. Itti, "State-of-the-Art in Visual Attention Modeling," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, 2013, pp. 185–207.
14. M.A. Just and P.A. Carpenter, "Eye Fixations and Cognitive Processes," *Cognitive Psychology*, vol. 8, 1976, pp. 441–480.
15. M. Spüler, W. Rosenstiel, and M. Bogdan, "Online Adaptation of a c-VEP Brain-Computer Interface (BCI) Based on Error-Related Potentials and Unsupervised Learning," *PLoS ONE*, vol. 7, no. 12, 2012, pp. 1–11.

Chandan Kumar is a postdoctoral researcher at the Institute for Web Science & Technologies (WeST), Universität Koblenz-Landau, Germany. Contact him at kumar@uni-koblenz.de.

Raphael Menges is a student research assistant at the Institute for Web Science & Technologies (WeST), Universität Koblenz-Landau, Germany. Contact him at raphaelmenges@uni-koblenz.de.

Steffen Staab is a professor for databases and information systems, and head of the Institute for Web Science & Technologies (WeST), Universität Koblenz-Landau, Germany. He also holds a chair for Web and Computer Science with the Web and Internet Science Research Group (WAIS) at the University of Southampton, UK. Contact him at staab@uni-koblenz.de.