Designing Gaze-supported Multimodal Interactions for the Exploration of Large Image Collections

Sophie Stellmach*, Sebastian Stober†, Andreas Nürnberger†, Raimund Dachselt*
Faculty of Computer Science
University of Magdeburg, Germany
{stellmach, dachselt}@acm.org; {stober, andreas.nuernberger}@ovgu.de

ABSTRACT

While eye tracking is becoming more and more relevant as a promising input channel, diverse applications using gaze control in a more natural way are still rather limited. Though several researchers have indicated the particular high potential of gaze-based interaction for pointing tasks, often gaze-only approaches are investigated. However, time-consuming dwell-time activations limit this potential. To overcome this, we present a gaze-supported fisheye lens in combination with (1) a keyboard and (2) and a tilt-sensitive mobile multi-touch device. In a user-centered design approach, we elicited how users would use the aforementioned input combinations. Based on the received feedback we designed a prototype system for the interaction with a remote display using gaze and a touch-and-tilt device. This eliminates gaze dwell-time activations and the well-known *Midas Touch* problem (unintentionally issuing an action via gaze). A formative user study testing our prototype provided further insights into how well the elaborated gaze-supported interaction techniques were experienced by users.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: [User-centered design, Input devices and strategies, Interaction styles]

General Terms

Design

Keywords

Eye tracking, multimodal interaction, gaze control

1. INTRODUCTION

One of the big challenges of computer science in the 21^{st} century is the digital media explosion. Steadily growing video, image, and music collections demand new approaches to keep this content accessible to a user. In contrast to queries posed as text or as an example (e.g., a similar object), exploratory systems address scenarios where users cannot formulate such a query – e.g., because the retrieval goal cannot be clearly specified in advance. A common approach to this problem is to provide an overview visualization of the content together with some means for navigation.

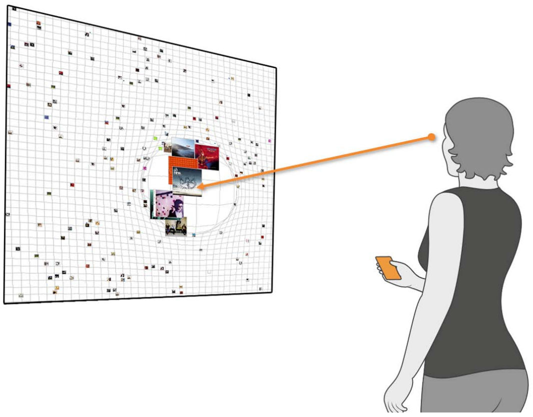

Figure 1: The envisioned setting for gaze-supported interactions using a large remote display.

Representing large image collections on a screen may lead to the challenge how to provide sufficient details while still maintaining an adequate context due to the limited screen space. While available display sizes are increasing, suitable focus and context techniques remain crucial for the exploration of large data representations. A comprehensive review about focus and context techniques is presented by Cockburn et al. [5]. This issue plays an important role in various application areas including the exploration of information visualizations, geographic information systems, virtual 3D scenes, but also the interaction with 2D graphical user interfaces (GUIs).

A fisheye lens is one solution to locally emphasize items of interest while still maintaining context information (local zooming). In this regard, information about a user's visual attention can support the accentuation of important content (i.e., data that are currently looked at). This is also beneficial for gaze-based selections (see e.g., [15, 2, 11]) as small target items are difficult to hit via gaze. Several researchers have indicated the high potential of gaze-based interaction for efficient pointing tasks (e.g., [13, 23, 2]) as gaze often

[*] User Interface & Software Engineering Group
[†] Data & Knowledge Engineering Group

NGCA '11, May 26-27 2011, Karlskrona, Sweden

precedes a manual action. However, the *Midas Touch* problem (unintentionally issuing an action via gaze) poses one of the major challenges for gaze-based interaction [13, 12]. One approach to overcome this problem is a suitable combination with additional input modalities (see e.g., [4]).

This motivated us to design gaze-supported remote interactions for exploring extensive amounts of images displayed, for example, on large screens as illustrated in Figure 1. For this purpose smartphones provide interesting features for interacting with remote multimedia content, such as accelerometers for tilt and throw gestures and touch-sensitive screens [7]. In the scope of this paper, we term these type of devices as *touch-and-tilt* devices. In addition, we look at a combination of gaze and a keyboard. Thereby, the keyboard can be seen as a representative modality for other input devices that have distinct physical buttons, such as gaming controller or remote controls for television sets.

Our work contributes to a deeper investigation of gaze-supported techniques for the exploration of large image collections using an adaptive non-linear fisheye lens (among others). The remaining paper is structured as follows: First, we discuss how gaze has been applied for the interaction with fisheye lenses in Section 2. In Section 3 we introduce a multimedia retrieval tool that we enhanced for gaze-supported interaction with (1) a keyboard and (2) a mobile touch-and-tilt device. Following a user-centered design approach, potential users have been interviewed on how they would operate such input combinations for browsing in a large set of images. The resulting user-elicited gaze-supported interaction techniques are presented in Section 4. Based on these insights, we propose a conflict-free set of multimodal interaction techniques which we implemented in a prototype system (see Section 5). This system was evaluated in a formative user study which is reported in Section 6. The received qualitative results and the concluding discussion in Section 7 provide guidance on how to further improve the design of gaze-supported interaction techniques.

2. RELATED WORK

Eye pointing speed and accuracy can be improved by target expansions [15, 2]. For this purpose, Ashmore et al. [2] describe gaze-based fisheye lenses to locally magnify the display at the point-of-regard (POG) which allows preserving the peripheral resolution. Another approach is to decrease the size of peripheral windows (*minification*) to preserve the focal window at the original resolution [9].

Ashmore et al. [2] point out that hiding the fisheye lens during visual search helps the user to get a better overview before making a selection. In addition, they also claim that a localized target expansion has the advantage of maintaining detail and context. Miniotas et al. [15], however, express that the benefits of *dynamic* target expansions are arguable due to inaccurate and jittering eye movements. As they point out themselves, this can be compensated by specialized algorithms to stabilize the eye cursor [24, 15].

Cockburn et al. [5] provide a comprehensive review about focus-and-context techniques including fisheye views. As an example, Ashmore et al. [2] use an underlying elastic mesh for the fisheye deformations with a flat lens top (with a constant zooming level). They use a single lens with a gaze dwell-based activation. Another and particularly promising fisheye lens technique is the *Spring–Lens* from Germer et al. [10] for distorting images based on a mass-spring model. While the *SpringLens* is also a context-preserving magnifying glass (similar to the fisheye lens used by Ashmore et al. [2]) it has the additional advantage to be able to magnify multiple areas of custom shape and to apply data-driven distortions in real-time. After all, while the *SpringLens* represents an interesting distortion technique, it has not yet been combined with gaze input. In this regard, Shoemaker and Gutwin [18] also describe fisheye lenses for multi-point interactions, however, they only use single-mouse input. Interestingly, they apply a dwell-time (via mouse) to trigger the fisheye lens instead of using an additional button. Therefore, this approach could be of interest for the adaptation to a gaze-only interaction.

Facilitating a touch-and-tilt device for the gaze-supported exploration of large image collections on a remote display has not been investigated so far (at least to the best of our knowledge). However, several works present gaze-supported interaction with large displays for target selections [3] or for example in combination with freehand pointing [21] and hand gestures [22].

In a nutshell, gaze-based fisheye lenses have not been combined with additional input devices yet. Also, they have not been used for the exploration of large image collections. However, previous work provides a good foundation and leads on what to consider for gaze-supported fisheye lenses, such as hiding the lens if not explicitly required [2, 18].

3. GAZE GALAXY

This section describes the general design of our exploratory multimedia retrieval system called *GazeGalaxy* with its implemented *SpringLens* technique. The underlying application is outlined in Section 3.1. As our aim is to advance the system for a novel gaze-supported interaction, the interaction tasks and functionality mappings that are available in *GazeGalaxy* are specifically addressed in Section 3.2. The novel gaze-supported interaction techniques, however, will be described in Section 5 based on initial interviews with potential future users that are reported in Section 4.

3.1 System Description

GazeGalaxy is based on a multimedia retrieval system for multi-facet exploration of music [19] and image collections [20] as presented by Stober et al. The application uses a galaxy-metaphor to visualize a collection which can comprise several thousand objects. As illustrated in Figure 2, an overview of the entire collection is presented by displaying some of the items as spatially well distributed thumbnails for orientation. The remaining content is represented as individual points (“stars”). The initial distribution of the objects is computed using multi-dimensional scaling [14]. This dimensionality reduction technique produces a 2D projection of the high-dimensional dataset while trying to correctly preserve the distances between the objects.[1]

This results in neighborhoods of similar objects (i.e. with small pairwise distances) based on features the user can weight, such as a particular color distribution [20]. Users can enlarge interesting neighborhoods with a *SpringLens*-based fisheye lens causing more thumbnails to be displayed in a specific region and at a larger size (see Figure 2).

[1] Object distances are computed from content-based features that are automatically extracted.

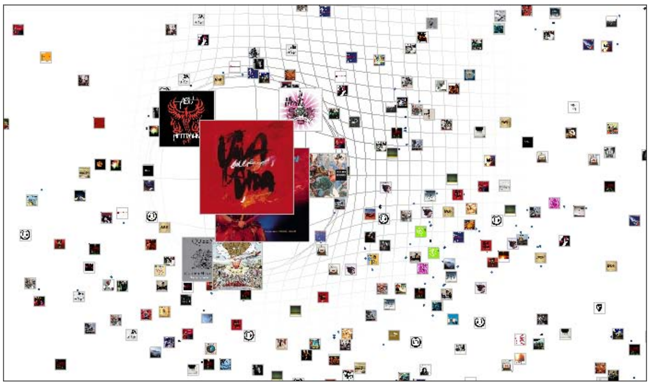

Figure 2: Screenshot of *GazeGalaxy*. (The partial grid is only displayed to show the underlying mesh structure. Inverted color scheme for printing.)

3.2 Interaction Tasks and Mappings

GazeGalaxy supports common interaction tasks for the exploration of information spaces [17], such as *overview*, *zoom + pan*, and *details on demand*. Additional actions, for example, to toggle filter modes, are also implemented but will not be emphasized further in this paper as we focus mainly on the lens interaction and on panning and zooming in the workspace as these are crucial tasks for various application contexts. The interaction tasks that we want to investigate further in the scope of this paper are listed in Table 1 with the currently implemented functionality mappings using keyboard and mouse. With respect to these task mappings, we want to additionally point out that some tasks are currently mapped repeatedly to different input variants. Panning can, for example, be performed by either dragging the workspace with a mouse or by using the cursor buttons on the keyboard. This gives users a clear and non-conflicting variety to choose the technique that fits best to their individual interaction style.

4. PRE-USER SURVEY

Aiming for a user-centered development, we conducted interviews with several volunteers to incorporate users already at an early stage of the design process. This helped us to find out how they would spontaneously use eye gaze with either a keyboard or a touch-and-tilt device to interact with the *GazeGalaxy* tool. This is a similar procedure as presented by Nielsen et al. [16] for the development of natural interaction interfaces.

Participants. Eight un-paid volunteers (six male, two female) participated in our interview. All were students or employees at a local university. Six participants had prior experience with eye tracking technology. All were familiar with mobile touch-and-tilt devices.

Procedure. First, the *GazeGalaxy* tool was introduced to each participant. The main interaction tasks (see Table 1) were demonstrated using a mouse and a keyboard. Afterwards, each participant was asked to imagine to work with *GazeGalaxy* on a large remote display using a combination of (1) *keyboard and gaze* and (2) *touch-and-tilt and gaze*. They were encouraged to actually stand up and hold a smartphone (an iPhone) to better put themselves in the envisioned situation. Then the interaction tasks were discussed in the order as listed in Table 1 for (1) and (2). For this, we asked each participant how he/she could imagine to perform a certain action with each input combination. Participants could describe multiple approaches for interaction.

Task **Change...**	Default interaction technique **Keyboard & Mouse**
Lens position	Right-click + Drag mouse pointer
Lens magnification	Right-click + Mouse-wheel
Pan	Cursor keys or Left-click + Drag
Zoom	Mouse-wheel or Press `+`/`-` keys
Thumbnail size	Press `PageUp`/`PageDown` keys

Table 1: The main interaction tasks that are available in *GazeGalaxy* with the current task mappings.

4.1 User-elicited Task Mappings

In the following, we discuss the qualitative feedback received from the eight interviewees. As assumed, the interviewees were concerned that gaze should not be overloaded with too many tasks as an undisturbed visual inspection of the screen is essential for exploration tasks. Hence, all interviewees expressed the opinion that the gaze input should be explicitly activated by a mode switch (more about possible ways to toggle modes will be explained in the following paragraphs).

4.1.1 Keyboard and Gaze

For the combination of a keyboard and an eye tracker, all interviewees agreed that the gaze may substitute the mouse well as a pointing device, for example, for setting the *lens position*. However, all interviewees asked for the possibility to switch modes so that the gaze input would not be active at all times to decrease the mental workload (and the *Midas Touch* problem). Thus, participants proposed an approach similar to switching between upper and lower case characters on a typewriter: Either continuously hold a certain key (e.g., `Shift`) to activate a mode or discretely toggle modes by pressing a key (e.g., the `Caps Lock` key). While six interviewees preferred the former approach, two also suggested a combination of both.

Panning the workspace could be performed using the cursor keys as indicated by six interviewees. Two participants could also imagine to use their gaze at the screen borders for panning (similar to the approach described by Adams et al. [1]). Thus, if looking at an item close to the upper screen border, this item will slowly move towards the center of the screen. In this respect, one interviewee also proposed to look at an item and hit a certain key (e.g., `C`) to center the currently fixated item.

For adapting the other interaction tasks (*lens magnification*, *zoom level*, and *thumbnail size*), all interviewees indicated that they would use keyboard buttons. One user expressed the idea that holding a certain button could initiate displaying a graphical scale for selecting, for example, a *thumbnail size* by gaze. This approach is also applicable to other aspects such as the *zoom level* and *lens size*. Except for this idea the integration of additional GUI elements was not proposed.

4.1.2 Touch-and-tilt and Gaze

Most controversy was on how to map panning and zooming tasks to a combination of gaze and a touch-and-tilt device - not resulting in any prevailing overall preferences. First of all, seven participants preferred holding the touch-and-tilt device in one hand with a vertical layout ("as you would normally hold a cell phone"). Although one participant also proposed to use the horizontal layout holding the device in both hands, his main preference remained for the first approach. This has important implications for the touchscreen design as most interface elements should then be easily reachable by the thumb (as illustrated in Figure 1).

Lens position. Six interviewees would follow an analog approach for positioning the lens as for the keyboard and gaze combination: while touching the mobile screen the lens is positioned at the point-of-regard. Also a discrete toggle using a virtual button on the touch device has been mentioned by two people. Two interviewees would not use gaze for adapting the lens position, but instead would use the touchscreen as a corresponding representation of the large display. Thus, if the user tabs into the left upper corner on the touchscreen, the lens will be positioned at the left upper corner of the large remote display.

Lens size. While six respondents would use a virtual slider with a one-touch slide gesture for imitating the mouse scroll wheel, also a pinch gesture was proposed.

Panning. While three people suggested to look at the screen borders to pan (as previously mentioned for the keyboard condition), another three would rather use a panning gesture on the touch device. Two interviewees could imagine to use the tilt functionality. Another idea was to stick (glue) the workspace to the current gaze position while touching a designated area on the mobile display.

Zooming. While all interviewees agreed on setting the zooming pivot by gaze (i.e., a zoom is performed at the position where the user currently looks at), answers on how to zoom in and out differed significantly. Three interviewees voted for tilting the mobile device forward and backward to zoom in and out. Two respondents mentioned a pinch gesture on the touchscreen; two others a slide gesture. One person would use a pull gesture (quickly moving the device towards the user) to zoom in and a throw gesture (quickly moving the device away from the user) to zoom out (see e.g., [7]). Also, the possibility to display additional GUI elements on the large display was mentioned, however, not preferred.

Thumbnail size. Participants suggested to use a pinch gesture (as mentioned by three respondents), a touch slider (two respondents), a virtual button on the touchscreen (one respondent), to tilt the device while continuously touching it for a mode switch (one respondent), and to turn the mobile device like a compass (one respondent).

The gathered user feedback provided valuable first insights for deciding on suitable techniques for a more natural gaze-supported interaction. However, further analysis for the specification of unambiguous interactions is required as several techniques were mentioned repeatedly for different tasks, such as tilting the device to pan, zoom, and adapt the thumbnail size. Especially for the combination of touch, tilt, and gaze input we discuss design considerations in more detail in the next section.

5. DESIGN OF GAZE-SUPPORTED INTERACTIONS

We structured the received user feedback and elaborated individual interaction sets that are free of ambiguous mappings (i.e., conflict-free): one set for condition (1) *keyboard and gaze* and one for condition (2) *touch-and-tilt and gaze*. In general, we decided on the interaction technique that was most frequently mentioned for a certain task (given that this technique has not already been assigned to another task). This approach worked well except for condition (2) as no clear overall user preferences could be identified for panning and zooming. An overview of the elaborated interaction sets is listed in Table 2. In the following, we explain the individual sets in more detail.

5.1 Keyboard & Gaze

The gaze is used to indicate where to position the fisheye lens and where to zoom in. These actions will, however, only be carried out, if an additional key is pressed on the keyboard (e.g., pressing the `PageUp` key to zoom in or holding the `Ctrl` key for lens positioning). The other tasks are mapped to different buttons on the keyboard as listed in Table 2.

5.2 Touch-and-tilt & Gaze

Before deciding on particular task mappings, several basic conditions had to be specified that came up during the interviews. As several respondents expressed that they would prefer holding the touch-and-tilt device in one hand and ideally only use the thumb to interact, we decided to refrain from using multitouch input such as a pinch gesture and instead focus on single-touch gestures (at least for the scope of this paper). Furthermore, two people were concerned that using a tilt could unintentionally be performed while talking or moving around. Thus, just as well as for gaze-based input (the *Midas Touch* problem), we decided that an additional explicit action (namely touching a certain area on the touchscreen) is required to issue a gaze- or tilt-based action. Finally, the need to shift the visual attention between mobile and remote display should be kept to a minimum. Thus, the interface and interaction with the mobile device should be designed to allow *blind interaction*. This means that regions on the touchscreen should be large enough and arranged in a way to allow users to interact with it without looking at the mobile screen.

Based on these basic assumptions we implemented a first prototype for exploring data in the *GazeGalaxy* tool via gaze and touch-and-tilt input. Preliminary graphical user interface designs for the touch-and-tilt device are shown in Figure 3. The design of this prototype and the elaborated interaction techniques are discussed in detail in the following.

We distinguish three different modes: (a) pan+zoom, (b) fisheye lens, and (c) thumbnail size. While we combined mode (a) and (b) on one screen of the multitouch device, we decided to use an additional screen for altering the thumbnail size. This was also motivated by the idea to integrate new tasks (e.g., for filtering the displayed content) in the future that would hardly fit on one single screen while still providing the possibility for a blind interaction. The tabs can be switched at the top of the touchscreen to leave the centered and easier to reach regions for more frequent tasks.

The first screen is divided into two active areas: a large region for the pan+zoom mode to the left and a smaller

Task	Different input combinations	
Change...	**Keyboard & Eye Tracker**	**Touch-and-tilt device & Eye Tracker**
Lens position	Look + Hold key	Look + Touch
Lens magnification	Press keys (e.g. 8 and 2 on the num pad)	Touch slide gesture
Pan	Cursor keys or Look at screen borders	Relative panning on touchscreen Look at screen borders
Zoom	Look + Press +/- keys	Look + Touch + Tilt
Thumbnail size	Press `PageUp`/`PageDown` keys	Touch slide gesture (with mode switch)

Table 2: The main interaction tasks that are available in the *GazeGalaxy* tool and possible functionality mappings to different multimodal input combinations.

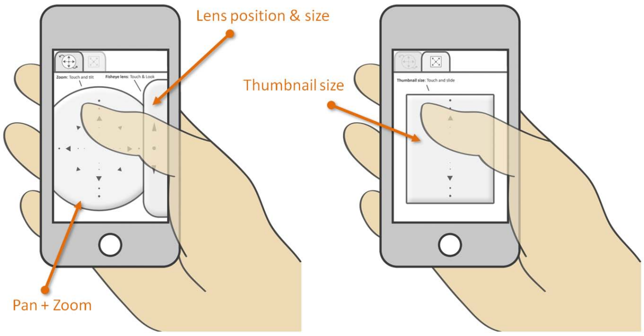

Figure 3: A first user interface prototype for the touch-and-tilt device.

area for the lens mode to the right (see Figure 3). As soon as one of the two areas is touched, the corresponding mode is activated. Thus, no actions (whether by gaze or tilt) will be performed if there is no touch event. As mentioned before this aims for preventing the *Midas Touch* problem.

Fisheye lens mode. The fisheye lens is positioned by looking at a location on the remote display while putting a finger on the lens area at the right of the first screen. If the finger slides up from the initial touch position, the lens size will be increased (and vice versa).

Pan+Zoom mode. If the user touches the pan area, the pan+zoom mode will be activated. Once this mode is active, panning can either be performed by looking at the screen borders (from the remote screen) or by panning on the touchscreen (as also described in [6]). For our prototype we use rectangular active regions at the screen borders for gaze panning. This eventually is the same approach as used by Adams et al. [1] for their screen panning regions. If the user looks at the left screen border, the workspace will shift to the right. This allows to move items that are currently close to the left border closer to the center of the screen.

Panning via touch is based on relative touch positions in the pan area on the multitouch screen. This means that no matter where the user touches the pan area initially, an upwards movement of the finger will lead to shifting the workspace up (analog for the other directions). This further supports a blind interaction.

As soon as touching the pan area, the user can also zoom by tilting the device forward and backward as also proposed by Dachselt and Buchholz [6]. For this, the orientation of the mobile device when activating the pan+zoom mode is used as a starting position. Tilting the device with respect to this starting position by at least a certain threshold will result in a zoom action. Thus, we use the relative positioning data instead of defining absolute positions for how to hold the device. Dachselt and Buchholz [6] use absolute positions which may cause problems due to differing physical constraints (i.e., some people cannot bend their wrists as much as others).

Thumbnail size mode. The second screen on the mobile device can be reached by touching the second tab at the top. Here the thumbnail size can be altered by performing a slide gesture. Moving the finger up results in larger and down in smaller thumbnail sizes.

After all, the elaborated interaction sets do not require any gaze dwell-time activations and thus should allow for a more fluent and quick gaze-based interactions. We made a particular point of eliminating the *Midas Touch* problem by accompanying gaze-based and tilt interactions with an additional explicit action such as pressing a button or touching the mobile screen.

5.3 System Setup

We enhanced *GazeGalaxy* to support various input modalities. For gathering gaze data we use a Tobii T60 table-mounted eye tracker. As touch-and-tilt device we use an iPhone/iPod Touch, but the system could easily be adapted to other multitouch smartphone devices as well. Finally, a computer is needed for executing *GazeGalaxy*. For our system setup, we run *GazeGalaxy* on the same system that hosts the Tobii T60 eye tracker. The devices can communicate among each other using the local area network. Thus, to communicate with the iPhone/iPod the computer requires a wireless network adapter. The communication is handled by a Virtual Reality Peripheral Network (VRPN) interface that we have extended to support various input devices. An overview of the system setup is presented in Figure 4. In the near future, the system will be further extended by additional input modalities, such as a mobile binocular eye tracker. This will allow to test the implemented interaction techniques with larger remote displays as originally envisioned in Figure 1.

6. FORMATIVE USER STUDY

To obtain first indications on the usability of our elaborated prototype for the combination of gaze and touch-and-tilt input, we conducted a formative qualitative user study

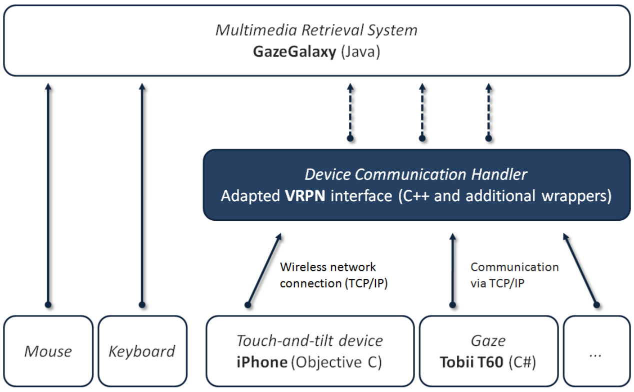
Multimedia Retrieval System
GazeGalaxy (Java)
Device Communication Handler
Adapted VRPN interface (C++ and additional wrappers)
Wireless network connection (TCP/IP)
Communication via TCP/IP
Mouse
Keyboard
Touch-and-tilt device
iPhone (Objective C)
Gaze
Tobii T60 (C#)
...

Figure 4: Schematic overview of the system setup for the gaze-supported multimodal interaction with *GazeGalaxy*.

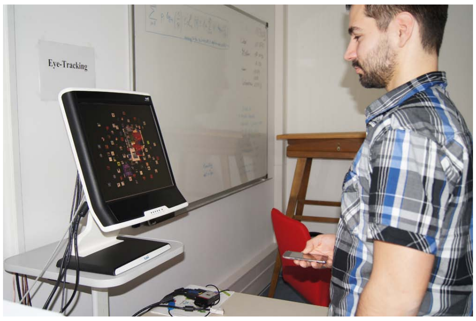

Figure 5: A participant standing in front of the Tobii T60 eye tracker to interact via gaze and an iPod Touch with the *GazeGalaxy* tool.

that gathered participants' impressions for using these techniques with *GazeGalaxy*.

Participants. Six staff members of a local university participated in the study (all male, with an average age of 29). While three people had already participated in the pre-user survey, for the other three people the topic and the presented techniques were completely new. All participants were right-handed.

Apparatus. We used the system setup described in Section 5.3. The Tobii T60 allows determining screen gaze positions at a frame rate of 60 Hz based on corneal-reflections that are identified in streamed video data. For stabilizing the gaze cursor, we used a *speed reduction* technique of Zhang et al. [24]. This means that raw gaze data were partially integrated with the previous gaze position. In addition, a minimal threshold distance (30 pixels) had to be traveled via gaze to assign the current value as new gaze cursor position. For the gaze panning, rectangular pan regions extending to 100 pixels at each screen boarder were defined (at a 1280 x 1024 screen resolution).

As touch-and-tilt device we used an iPod Touch (2nd generation). This device allows multitouch interaction on a mobile screen and provides a three-axis accelerometer for tilting. The graphical user interface from the iPod was designed according to the screen prototype illustrated in Figure 3.

The eye tracker was positioned on an elevated rack (see Figure 5) so that the participants could comfortably stand in front of it. This should give them a better feeling for the remote interaction with a distant display and should show us how users would hold the mobile device in such a situation.

Procedure. Participants were welcomed and briefly introduced to the purpose of the survey (to get first impressions on some novel interaction techniques). First, it was checked that the eyes of a user could be correctly detected by the eye tracker. Otherwise the distance and angle to the eye tracker screen had to be adjusted. Secondly, an eye tracker calibration was performed. Then the *GazeGalaxy* tool was started and the interaction tasks with their particular mappings were explained in the order as listed in Table 2. The participants could directly test the techniques and were encouraged to verbally express their thoughts about the interaction. After all techniques had been explained, the participants could test the different interaction techniques in any order to explore a large image collection (with 800 pictures). At the end, the experimenter concluded by asking the participants about particular advantages and disadvantages of each technique (again in the order as presented in Table 2).

6.1 Results

Overall the combination of gaze and a remote touch-and-tilt interaction was perceived very positively. Three participants described it as intuitive and easy to understand. Nevertheless, interacting via gaze felt unfamiliar (mentioned by two participants). Some participants were actually surprised that the interaction worked that well as they had never used gaze input before. Some minor remarks were mentioned how to further improve the interactions especially concerning better visual feedback about the current gaze position and active mode. In principle, all participants found it good that an additional explicit action (i.e. a touch-event in a designated area) had to be performed to enable gaze and tilt input. However, one participant suggested that he did not find this intuitive as it could be confusing which area must be touched to issue a certain command. Two participants found it particularly interesting that the rough gaze position is sufficient for interaction. Although the interaction with only one hand was considered convenient, the possibility to turn the iPod and use a two-handed interaction on a horizontal screen layout was proposed by one participant again (see Section 4).

Lens position. Setting the fisheye lens position via gaze was described as natural and useful. Only moving the lens if touching the device was found very helpful, as the fear existed to easily loose orientation (as mentioned by five participants). Three participants would have liked to use the gaze lens while also being able to pan and zoom. This is currently not supported for the assumed thumb-only interaction. One participant proposed a persistent gaze mode as mentioned in the pre-user survey for the `Caps Lock`-mode.

Lens size. Changing the magnification of the fisheye lens via a slider on the right side of the screen was found good (four participants). However, the layout may need to be adjusted for left-handers. The slider could be improved by a flick gesture to set a motion in action. One participant wished for a larger slider area and for arranging the slider to the opposite side as he argued that he could reach the opposite side better with his thumb. Additional ideas for changing the lens size included a dwell-time based increase in size or incorporating the distance from a user's head to the screen.

Panning via touch. All participants found it good to be able to move the workspace by sliding the finger on the touchscreen. As mentioned before, an additional flick gesture to start a certain motion that slowly subsides would be nice. One person proposed that instead of using a touch event, a "flight-mode" could be activated to move through the scene by tilting the iPod.

Panning via gaze. As described in Section 5, it is also possible to move the view by looking at the screen borders. Three participants found this particularly useful for moving items that are cut off by a screen border towards the screen center. In this respect, two participants mentioned that they find this technique ideal for fine adjustments of the view, while they would rather use the touch panning to cross larger distances. In addition, larger margins were desired (as indicated by two participants) as it was sometimes difficult to hit the screen corners via gaze. It was also suggested to increase the panning speed towards the screen borders. Furthermore, two participants missed some form of feedback for the active margin regions such as a highlight-on-hover effect.

Zooming via tilt and gaze pivoting. It was found very intuitive by all participants that the view zooms in at the location that is currently looked at. While most participants liked to tilt forward/backward for zooming, one participant intuitively double-tabbed on the panning region to zoom, and another one tilted left and right. Both participants expressed that they in general do not like to use the forward and backward tilt as they fear to need to twist their wrists uncomfortably. After explaining that our technique uses the device's relative motions after touching the panning area (see Figure 5.2), they dismissed their concerns.

Thumbnail size. Using a touch slide gesture for changing the thumbnail size was found good. However, four participants disliked the need to switch between tabs. On the one hand, three participants explained that they did not like to look down from the primary screen (the Tobii monitor) to the mobile device to switch modes. In this regard, it was proposed to rearrange the tab icons to switch between tabs even without looking at the mobile screen (*blind interaction*), for example, by locating each tab icon at different screen corners. On the other hand, the second mode only contained the feature to change the thumbnail size and although this is not a frequent task, it may disrupt the workflow. Thus, different solutions were offered by the participants, including to integrate this task into the first tab-screen or to use a multitouch gesture after all (e.g., a pinch).

7. DISCUSSION

All in all, the combination of gaze, touch, and tilt input for interacting with a remote display was perceived as very promising in the formative user study. At first sight, such a combination appears to be a step backwards with respect to intuitive interaction design. Although using gaze for pointing is considered very intuitive, having to use another input channel simultaneously increases the effort and complexity of the interaction. However, as participants from both studies reported, this is accepted, because this allows for a more relaxed gaze-based interaction for the following reasons:

- The *Midas Touch* problem is avoided as users can communicate via an additional input channel whether an action is really intended.
- For the same reason, there is no need for dwell-time activations which otherwise would slow down the interaction.
- The different input modalities complement well for supporting multiple tasks simultaneously (such as panning and zooming), which is difficult for gaze-only interaction.

The importance of well-designed feedback was confirmed as users want to be assured that the system understood their intentions correctly and that the intended mode has been activated. Visual feedback about the current mode could, for example, be indicated by adapting the cursor's shape and color as done by Istance et al. [12] for different gaze interaction conditions. Feedback also plays an important role for identifying tracking problems (whether of the gaze or touch and tilt data) – e.g., if the communication to the devices is temporarily lost. At the current stage, the *GazeGalaxy* application's only direct feedback is the focus visualization by the SpringLens. Further possibilities also incorporating haptic feedback (e.g., vibrations) and auditory feedback (e.g., beep sounds) for not overloading the visual input channel need to be investigated.

Recalling the original application scenario, exploratory search, the combination of a fisheye-based visualization with gaze input for focus control indeed seems to be very promising according to received user feedback (which confirms findings from Ashmore et al. [2]). The *GazeGalaxy* application could clearly be further elaborated – especially in terms of additional gaze-contingent visualizations [8] to emphasize objects of (visual) interest. Furthermore, the possibilities for blind interaction with the system should be extended: Here, using the orientation of the touch-and-tilt device (i.e. vertical or horizontal layout) as mode switch is an interesting option as an alternative to the tabs interface.

For the near future, porting the system to a large display as shown in Figure 1 is pursued using a mobile eye tracker. Finally, the implemented novel interaction techniques in *GazeGalaxy* need to be compared against other input combinations in terms of retrieval performance and interaction efficiency for the originally intended setting (some exploratory task).

8. CONCLUSION

This paper presented a detailed description of a user-centered design process for gaze-supported interaction techniques for the exploration of large image collections. For this purpose, gaze input was combined with additional input modalities: (1) a keyboard and (2) a mobile tilt-enabled multitouch screen. The integration of user feedback at such an early stage of the design process allowed for the development of novel and more natural gaze-supported interaction techniques. While gaze acted as a pointing modality, the touch and tilt actions complemented the interaction for a multifaceted interaction. Based on user-elicited interaction techniques we developed an extended multimedia retrieval system, *GazeGalaxy*, that can be controlled via gaze and touch-and-tilt input to explore large image collections. First user impressions on the implemented interaction techniques were gathered and discussed. Results indicate that gaze input may serve as a natural input channel as long as certain design considerations are taken into account. First, gaze data is inherently inaccurate and thus interaction should not rely on precise positions. Using the gaze positions for setting a fisheye lens and zooming in at the point-of-regard were described as intuitive. Secondly, users should be able to confirm actions with additional explicit commands to prevent unintentional actions.

9. ACKNOWLEDGMENTS

We want to thank all volunteers for their insightful comments in our evaluations. This research was supported by the *German National Merit Foundation*, the German Research Foundation (DFG) project *AUCOMA*, and the German Ministry of Education and Science (BMBF) project *ViERforES-II* (01IM10002B).

References

[1] N. Adams, M. Witkowski, and R. Spence. The inspection of very large images by eye-gaze control. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 111–118, 2008.

[2] M. Ashmore, A. T. Duchowski, and G. Shoemaker. Efficient eye pointing with a fisheye lens. In *Proceedings of Graphics Interface 2005*, GI '05, pages 203–210, 2005.

[3] H.-J. Bieg. Gaze-augmented manual interaction. In *Proceedings of CHI'09 - Extended abstracts*, pages 3121–3124, 2009.

[4] E. Castellina and F. Corno. Multimodal gaze interaction in 3d virtual environments. In *COGAIN '08: Proceedings of the 4th Conference on Communication by Gaze Interaction*, pages 33–37, 2008.

[5] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, Vol. 41:1–31, 2009.

[6] R. Dachselt and R. Buchholz. Throw and tilt – seamless interaction across devices using mobile phone gestures. In *Lecture Notes in Informatics, Vol. P-133*, MEIS '08, pages 272–278, 2008.

[7] R. Dachselt and R. Buchholz. Natural throw and tilt interaction between mobile phones and distant displays. In *Proceedings of CHI'09 - Extended abstracts*, pages 3253–3258, 2009.

[8] A. T. Duchowski, N. Cournia, and H. Murphy. Gaze-contingent displays: A review. *CyberPsychology & Behavior*, 7(6):621–634, 2004.

[9] D. Fono and R. Vertegaal. Eyewindows: evaluation of eye-controlled zooming windows for focus selection. In *Proceedings of CHI'05*, pages 151–160, 2005.

[10] T. Germer, T. Götzelmann, M. Spindler, and T. Strothotte. SpringLens – Distributed Nonlinear Magnifications. In *Eurographics - Short Papers*, pages 123–126. EGPub, 2006.

[11] D. W. Hansen, H. H. T. Skovsgaard, J. P. Hansen, and E. Møllenbach. Noise tolerant selection by gaze-controlled pan and zoom in 3d. In *Proceedings of ETRA'08*, pages 205–212. ACM, 2008.

[12] H. Istance, R. Bates, A. Hyrskykari, and S. Vickers. Snap clutch, a moded approach to solving the midas touch problem. In *Proceedings of ETRA'08*, pages 221–228. ACM, 2008.

[13] R. J. K. Jacob. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of CHI'90*, pages 11–18, 1990.

[14] J. Kruskal and M. Wish. *Multidimensional scaling.* Sage, 1986.

[15] D. Miniotas, O. Špakov, and I. S. MacKenzie. Eye gaze interaction with expanding targets. In *Proceedings of CHI'04 - Extended abstracts*, pages 1255–1258, 2004.

[16] M. Nielsen, M. Störring, T. Moeslund, and E. Granum. A procedure for developing intuitive and ergonomic gesture interfaces for HCI. In *Gesture-Based Communication in Human-Computer Interaction*, volume 2915 of *Lecture Notes in Computer Science*, pages 105–106. Springer Berlin / Heidelberg, 2004.

[17] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, 1996.

[18] G. Shoemaker and C. Gutwin. Supporting multi-point interaction in visual workspaces. In *Proceedings of CHI'07*, pages 999–1008, 2007.

[19] S. Stober and A. Nürnberger. A multi-focus zoomable interface for multi-facet exploration of music collections. In *7th International Symopsium on Computer Music Modeling and Retrieval*, pages 339–354, 2010.

[20] S. Stober, C. Hentschel, and A. Nürnberger. Multi-facet exploration of image collections with an adaptive multi-focus zoomable interface. In *WCCI '10: Proceedings of 2010 IEEE World Congress on Computational Intelligence*, pages 2780–2787, 2010.

[21] D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 33–42, 2005.

[22] B. Yoo, J.-J. Han, C. Choi, K. Yi, S. Suh, D. Park, and C. Kim. 3d user interface combining gaze and hand gestures for large-scale display. In *Proceedings of CHI'10 - Extended abstracts*, pages 3709–3714, 2010.

[23] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of CHI'99*, pages 246–253, 1999.

[24] X. Zhang, X. Ren, and H. Zha. Improving eye cursor's stability for eye pointing tasks. In *Proceedings of CHI'08*, pages 525–534, 2008.