Gaze vs. Mouse: A Fast and Accurate Gaze-Only Click Alternative

Christof Lutteroth, Moiz Penkar, Gerald Weber
University of Auckland
Auckland 1010, New Zealand
{christof, moiz, gerald}@cs.auckland.ac.nz

ABSTRACT
Eye gaze tracking is a promising input method which is gradually finding its way into the mainstream. An obvious question to arise is whether it can be used for point-and-click tasks, as an alternative for mouse or touch. Pointing with gaze is both fast and natural, although its accuracy is limited. There are still technical challenges with gaze tracking, as well as inherent physiological limitations. Furthermore, providing an alternative to clicking is challenging.

We are considering use cases where input based purely on gaze is desired, and the click targets are discrete user interface (UI) elements which are too small to be reliably resolved by gaze alone, e.g., links in hypertext. We present Actigaze, a new gaze-only click alternative which is fast and accurate for this scenario. A clickable user interface element is selected by dwelling on one of a set of confirm buttons, based on two main design contributions: First, the confirm buttons stay on fixed positions with easily distinguishable visual identifiers such as colors, enabling procedural learning of the confirm button position. Secondly, UI elements are associated with confirm buttons through the visual identifiers in a way which minimizes the likelihood of inadvertent clicks. We evaluate two variants of the proposed click alternative, comparing them against the mouse and another gaze-only click alternative.

Author Keywords
Eye gaze tracking, web browser navigation

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: User Interfaces—*Input devices and strategies*

INTRODUCTION
Eye gaze trackers make it possible to monitor the user's gaze, i.e., detect where the user is looking on a surface such as a computer display. The technology has been used in research for a long time, and is now slowly finding its way into the consumer market. The price of entry-level eye gaze trackers has fallen dramatically, and some mobile devices implement eye control features such as pausing video in the absence of detected gaze. Eye gaze tracking has characteristics which make it interesting for human-computer interaction. When used for pointing, it is natural and easy to use [27] and is one of the fastest input mechanisms [29, 25]. Users naturally look at a target object before performing any action, so the gaze also reflects users' intentions [9, 16]. This raises the prospect of using it in a general-purpose input method.

Terminology & Scope: In order to use eye gaze tracking effectively for input, we need a *click alternative* [23], i.e., a method which can be used to activate a clickable user interface element (*clickable* for short), analogous to a mouse. There are two typical applications for clicking: first, precision-clicking of any one of a near-continuum of positions, such as in paint tools or in text-editors, and secondly clicking of *discrete clickables* such as hyperlinks and buttons. The second application (discrete clickables) is sufficient for many common activities such as browsing. Similarly to important related works [5, 1, 28, 23], we restrict our investigation to discrete clickables. For continuous precision clicking, e.g., for graphics and text editing, different click alternatives would have to be used. Moreover, we consider only *gaze-only* click alternatives, meaning those which rely only on gaze input, and not *hybrid* click alternatives using extra hardware such as buttons. Finally, we support targets which are too small to be resolved reliably by gaze pointing alone, e.g., hyperlinks.

Motivation: There are many applications for gaze-only click alternatives. They are widely used for accessibility in medical conditions where button-based clicking cannot be used easily. There are industrial and medical use cases which require touchless interaction, e.g., if the users' extremities are contaminated or otherwise engaged. Other modalities may be impractical in some situations, e.g., a noisy environment can preclude voice input. A good gaze-only click alternative may also be of interest for everyday situations where mouse and touch are inconvenient.

Creating a workable gaze-only click alternative is challenging. Gaze pointing is not accurate enough even for typical discrete clickables. This limitation arises not only because of the current gaze tracking technology, but is also due to physiological limitations of the eye [18]. Furthermore, while gaze pointing seems fairly straightforward, activating ("clicking") is not [11]. Activating a clickable after the user looks at it for certain time ("dwell") leads to inadvertent clicks (the "Midas touch" problem). This problem remains an important concern for almost all gaze-only click alternatives. Unsurprisingly, none of the existing gaze-only click alternatives comes close to the mouse as a gold standard in terms of combined speed, accuracy and ease of use (see Related Work).

In this work we present a novel gaze-only click alternative called *Actigaze*[1]. It is similar to an existing gaze-only click alternative, *Multiple Confirm* [23], which we found applied in practice (e.g., for accessibility), but comes significantly closer to the speed and accuracy of the mouse. Multiple Confirm uses gaze-pointing to pre-select a set of potential targets, i.e., the user first dwells near the target. Then *confirm buttons* are shown for each of the potential targets (Figure 1): these are dwell-activated buttons which are shown in the periphery of the users' active work area, together with textual labels for the targets. To disambiguate between the potential targets, the user dwells on one of the confirm buttons and the corresponding target is "clicked". Multiple Confirm avoids inadvertent clicks with this separate confirm step and disambiguates between potential targets in areas crowded with clickables. However, finding the right confirm button for a target is slow.

Contributions: Actigaze addresses the shortcomings of Multiple Confirm. Potential targets are colored and the periphery of the users' work area is filled with confirm buttons which have corresponding colors (Figure 2). Finding the confirm button for a target is done by simple color matching. A key innovation is that the confirm buttons are *stable*: first, they have fixed positions, and secondly, their colors are fixed and can be learned so as to become part of procedural memory. We present two variants of Actigaze which can be selected on user preference: *Static Coloring*, where all clickables are colored immediately, and *Dynamic Coloring*, where only clickables close to the current gaze position (i.e., potential targets) are colored. In general, *visual identifiers* other than colors can also be used, e.g., patterns or icons. The main contributions are:

1. Stable confirm buttons with fixed colors which enable learning. Since the confirm buttons do not require animation, they can be placed outside a screen.
2. Stable coloring: The colors are pre-assigned when the user interface is first rendered and remain constant afterwards.
3. Optimized coloring: The coloring of the clickables is done in such a way as to minimize the likelihood of inadvertent clicks, by avoiding close occurrences of the same color. An efficient algorithm to calculate the coloring is provided.
4. Two variants (Static Coloring and Dynamic Coloring) offering a trade-off between visual appearance and speed.
5. An a-posteriori gaze analysis technique for Static Coloring which simplifies the state machine of this variant.
6. Optimized color choice: A set of colors proposed as visual identifiers which are easily distinguishable and preserve contrasts.
7. An experimental comparison of Actigaze with the mouse and Multiple Confirm.

[1]`http://www.actigaze.com`

BACKGROUND AND RELATED WORK

The accuracy of gaze tracking is limited for two reasons: first, technical limitations, which may improve as the technology evolves; secondly, physiological limitations which cannot be overcome by better geometric gaze tracking. Due to the size of the human fovea the geometric axis of the eye is not always aligned with the gaze point, and the gaze position is constantly fluctuating due to involuntary eye movements. Moreover, when gaze is used to perform actions in a user interface, it is difficult to separate perception from intention, which can result in unintentional actions, e.g., if clicks are triggered by dwell [13]. Various gaze click alternatives have been devised to perform actions or mimic the mouse [11]. They can be classified as: direct, indirect and auxiliary.

Direct click alternatives use the gaze to point directly onto the target but vary in the mechanism used to trigger the activation. Direct alternatives are inherently limited by the inaccuracy of gaze tracking [18] and *cannot be used with general, unmagnified user interfaces*, which contain clickables such as hypelinks which are simply too small. The most obvious and natural direct alternative is *dwell* (or fixation), which activates a clickable after the gaze dwells on it for a defined time interval. For very simple object selection tasks, dwell can be significantly faster than the mouse [25]; it has been used for specialized UIs such as carefully-designed menus [21, 28]. However, short dwell times lead to inadvertent clicking, and long dwell times are difficult due to involuntary eye movement [22].

Some click alternatives use gaze-pointing together with hardware buttons for clicking. Hardware buttons seem slightly faster than simple dwell with a typical 0.4 second threshold, but are less accurate as people have been found to tend to click before the gaze has fully settled on the target [29]. Researchers in this area agree that these click alternatives should be reserved for applications where an additional hardware button is feasible and where the click targets are sufficiently large [32, 18, 30].

Blinks, winks and other facial movements have been used in direct alternatives to trigger activations [19, 7]. While good detection rates can be achieved, such movements are usually not used consciously. Short blinks occur naturally about 10 times per minute, so click alternatives use longer blinks for disambiguation. However, longer blinks affect vergence and the fixation point is lost. Some users find eyebrow raises to be more tiring than blinking while others find blinking to be more disruptive than eyebrow raises [7].

Some direct alternatives use gaze movement patterns. Møllenbach et al. [20] studied single point-to-point movements for selecting objects in areas near the edge of the screen. Users gazed at a target and then at the opposite screen edge to complete the selection. Møllenbach et al.

found shorter, horizontal movements to be more efficient than longer or vertical ones. Huckauf et al. [12] studied anti-saccades for target selection: when a user dwells on a clickable, a copy of it appears beside it and can be activated by glancing in the direction opposite to the copy. Compared to simple dwell, anti-saccades resulted in shorter click times but also in more errors.

Different magnification techniques have been proposed to enable the use of direct alternatives for smaller clickables. In the ERICA system [17] users dwell on a region to magnify it, and then select a point of interest with a second dwell. Finally, users choose a mouse action (left click, right click, etc.) using dwell-activated buttons. Skovsgaard et al. [26] compared two-step and three-step magnification for dwell-based target selection, finding three-step magnification to be more accurate than two-step magnification, albeit with similar subjective ratings and selection times. Ashmore et al. [2] compared a dwell-activated fish-eye lens with a continuous fish-eye zoom for dwell-based target selection, with the lens performing better than the zoom and simple dwell. Fish-eye lenses distort the border of the magnified area but avoid occlusion of screen content, as seen with simple magnification techniques. The Bubble Cursor [8] dynamically zooms the gazed-at area so that only one target is selectable at any time. EyePoint [14] magnifies the region around the gaze point when a hotkey is pressed, allowing users to perform mouse click operations using a normal keyboard. Magnification is useful for smaller clickables but disrupts the UI visually, especially when triggered by unintentional dwells, and adds to the overall click time.

Indirect click alternatives provide an additional selection mechanism for disambiguating between potential targets, so they do not rely so much on precise gaze coordinates. They can be used in general user interfaces with smaller clickables, without disrupting the user experience as much as magnification does. Dwell has been combined with *confirm buttons* [29, 22, 23]: users first gaze near a clickable by dwelling and then confirm the selection by looking at a confirm button. To compensate for gaze tracking inaccuracy, confirm buttons are presented for all the clickables within a certain radius around the center of dwell. In previous work we compared different existing confirm button based click alternatives and found that arranging confirm buttons directly around a dwell in the user interface ("Radial Confirm") is too distracting as it occludes content and draws the gaze away [23]. The best performing alternative (Multiple Confirm) places the confirm buttons in the margin of the display area, with labels for clickables (Figure 1). Its accuracy is close to that of the mouse, with an average click time of 7.4 seconds, compared to 2.1 seconds for the mouse [23].

Some gaze-controlled web browsers use indirect alternatives with more than two dwells for clicking. Abe et al. [1] proposed dwell-activated directional buttons for moving a cursor and activating a selected link. Castellina & Corno [5] proposed dwell-activated directional buttons for selecting a group of links tagged with running numbers, and activating a link in the group with dwell-activated number buttons. Alternatively, all links are tagged with running numbers and can be 'dialed' with a dwell-activated numeric keypad. In such methods, especially those using selection cursors, the click time grows with the number of clickables, so they are slower than the click alternatives using confirm buttons. Castellina & Corno [5] found participants preferred the 'dialing' approach over ERICA magnification [17] for browsing.

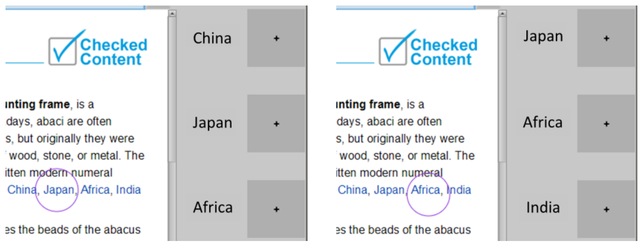

Checked Content
nting frame, is a
days, abaci are often
s, but originally they were
wood, stone, or metal. The
itten modern numeral
China, Japan, Africa, India
es the beads of the abacus
China
Japan
Africa
India
+

Figure 1. Existing Multiple Confirm click alternative: confirm buttons are unstable as the dwell position changes, even if referring to the same link (left vs. right).

Eye gaze gestures have also been studied as indirect click alternatives. Drewes & Schmidt [6] evaluated gaze movement around a dialog window either in clockwise or anti-clockwise direction for yes/no responses, as well as other gestures. They found gestures can be reliable and efficient ($\approx$1.9 seconds for dialog), and UI edges and corners are natural anchor points. Gestures can be efficient for constrained selection tasks but are error prone and too limited as a more general click alternative. Many users find them tiring, preferring techniques which require minimal deliberate eye movements [13].

Auxiliary click alternatives use gaze tracking with a physical pointing device for disambiguation between different targets. MAGIC [31] moves the pointer quickly to the gaze position using the mouse for finer movements and clicking, to speed up pointing. The Rake Cursor [3] shows a grid of multiple mouse pointers simultaneously, moving the whole grid with the mouse and selecting the active pointer in the grid by gaze. It successfully reduces mouse movements as the pointer closest to a target can be used. The Gaze-enhanced User Interface Design (GUIDe) [15] combines gaze with keyboard and mouse to improve various common tasks.

STABLE CONFIRM BUTTONS

Many of the named approaches come with a trade-off: direct alternatives by themselves are too inaccurate for small targets, magnification disrupts the user experience, indirect approaches involving more than two dwells are slow, and gaze gestures can trigger only a very limited number of options. Multiple Confirm is a plausible gaze-only click alternative for discrete clickables. It resolves ambiguities and prevents inadvertent clicks. However, it is still considerably slower than the mouse. A likely reason is that the confirm buttons are *unstable*, i.e., their button labels and positions change (Figure 1) as the gaze moves, making it harder to find the right button and preventing users from using procedural memory. To address this, we propose here a technique which allows the confirm buttons to be *stable*, i.e., stay at constant positions with constant *visual identifiers*.

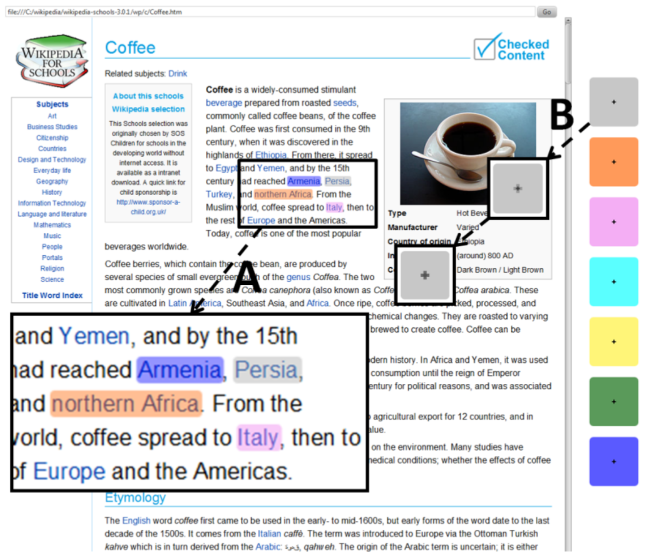

Coffee
Checked Content
A
B
and Yemen, and by the 15th
had reached Armenia, Persia,
and northern Africa. From the
world, coffee spread to Italy, then to
of Europe and the Americas.
Etymology

Figure 2. When dwelling, clickables near the gaze are colored dynamically (A) and associated with stable confirm buttons (right margin). The crosshair of a confirm button is animated during dwell (B).

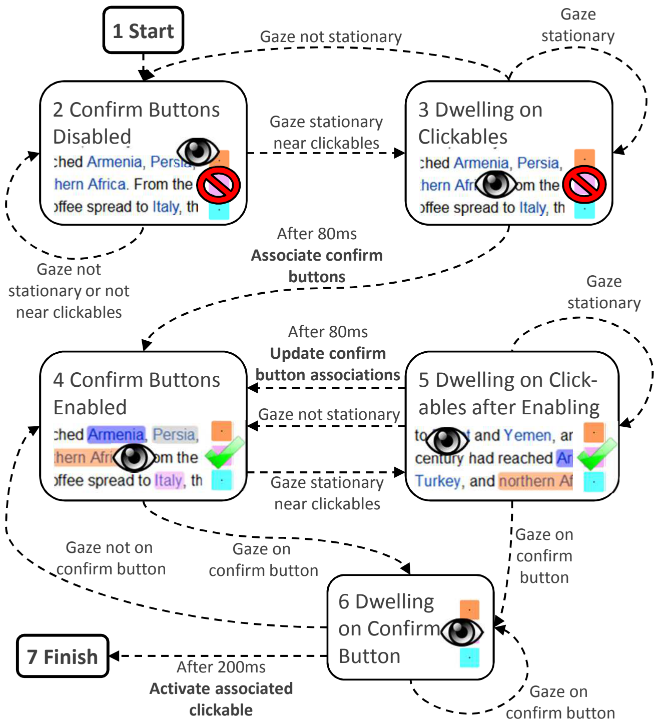

1 Start
Gaze not stationary
Gaze stationary
2 Confirm Buttons Disabled
Gaze stationary near clickables
3 Dwelling on Clickables
Gaze not stationary or not near clickables
After 80ms Associate confirm buttons
After 80ms Update confirm button associations
Gaze stationary
4 Confirm Buttons Enabled
Gaze not stationary
5 Dwelling on Clickables after Enabling
Gaze stationary near clickables
Gaze not on confirm button
Gaze on confirm button
Gaze on confirm button
6 Dwelling on Confirm Button
7 Finish
After 200ms Activate associated clickable
Gaze on confirm button

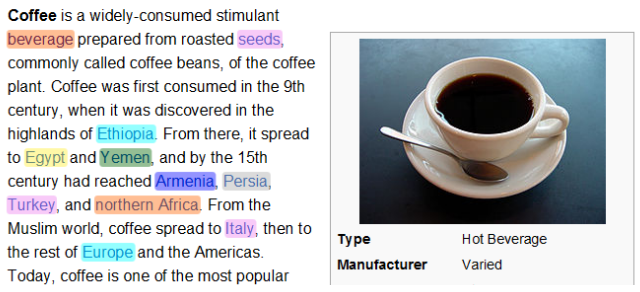
Figure 3. State machine of clicking with stable confirm buttons.

Figure 2 shows a web browser with stable confirm buttons in a margin on the right and colors as visual identifiers. There are two levels of stability, and both can contribute to learning by the user and to increased speed. First, the confirm buttons and their colors are stable and do not change. They could in principle even be placed outside the interactive display. As a result, the position of the colors can be learned and become part of procedural memory. Peripheral vision can help to locate confirm buttons with clear visual identifiers. Secondly, the *assignment* of visual identifiers to clickables can be kept stable as long as the user interface does not change. Given the same browser settings, the hyperlink “Persia” in Figure 2 can always be associated with the gray button in the top-right position. Among other things, this enables the concept of static coloring explained below.

All clickables are assigned visual identifiers right from the start, but because the number of confirm buttons is limited (seven in Figure 2), only a subset of clickables can be *associated* with confirm buttons at any time. Actigaze associates all clickables within a certain *association radius* ($\approx$ 1 cm in our prototypes) around the gaze point with the confirm buttons which have the corresponding visual identifiers. This makes Actigaze a multiple-confirm technique and compensates for tracking inaccuracy. If the user dwells on a confirm button for long enough, the clickable associated with that confirm button is activated. There is at most one clickable associated with each confirm button at any time. The necessary steps for the assignment of visual identifiers are discussed later. As the gaze wanders, different clickables are close to the gaze point and the association has to change. This gives rise to the two variants of Actigaze: Static Coloring and Dynamic Coloring. Static Coloring shows the *assigned* visual identifiers of all the clickables simultaneously (Figure 4); Dynamic Coloring shows the visual identifiers only for the *associated* clickables (Figure 2). We discuss Dynamic Coloring first as it illustrates the association process.

State Machine: The state machine in Figure 3 specifies how the association of clickables with confirm buttons is maintained based on dwell. The notation follows the one we used in [23]. At first no clickables are associated with the confirm buttons, i.e., the confirm buttons are disabled (2). This is important to prevent inadvertent clicks immediately after a user has activated a clickable: the user’s gaze may still linger on a confirm button after activating it by dwell. When the gaze dwells within the association radius of clickables (3), the clickables are associated with the confirm buttons after a certain time (4), using the pre-defined assignment. The associations between confirm buttons and clickables are updated as the gaze dwells near other clickables (5). While the confirm buttons are enabled (4 and 5), the associated clickables are colored. In order to activate one of them the user can immediately move the gaze to the corresponding confirm button (6) and activate the clickable through dwell (7). In our prototypes we used an *association dwell threshold* of 80 ms and an *activation dwell threshold* of 200 ms.

Stable confirm buttons are always visible and static, which means there is a lower potential for distraction compared with buttons which appear or change dynamically. However, clickables have to be modified to indicate their assigned visual identifier to the user, and this can potentially distract or interfere with the content of a UI. In the Dynamic Coloring variant, the UI’s overall appearance is changed to only a limited degree, since only the clickables currently associated with confirm buttons are colored (Figure 2). This has the additional benefit that users receive feedback about which click-


Coffee is a widely-consumed stimulant beverage prepared from roasted seeds, commonly called coffee beans, of the coffee plant. Coffee was first consumed in the 9th century, when it was discovered in the highlands of Ethiopia. From there, it spread to Egypt and Yemen, and by the 15th century had reached Armenia, Persia, Turkey, and northern Africa. From the Muslim world, coffee spread to Italy, then to the rest of Europe and the Americas. Today, coffee is one of the most popular

Type	Hot Beverage
Manufacturer	Varied

Figure 4. All links are colored statically in a way which minimizes the likelihood of inaccurate clicks.

ables are associated. Furthermore, the dwell time threshold controlling how soon new clickables are associated can be varied. This achieves a tradeoff between potential distraction and speed: a higher threshold means the clickables are changed less frequently but it slows down the clicking, and vice-versa for a lower threshold.

Static Coloring: An alternative to Static Coloring is that all clickables are colored all the time. This is the Static Coloring variant of Actigaze (Figure 4). This does somewhat modify the UI's overall appearance. However, there are no dynamic changes which could distract the user. It should be remembered that instead of colors one can use patterns or shapes (e.g., clock hands or numbers) as visual identifiers; this could mitigate the visual impact of Static Coloring. Furthermore, Static Coloring can be used with non-active displays such as paper posters or slow displays such as e-ink (then using above alternatives for colors). Static Coloring is faster than Dynamic Coloring because the user sees the color of a target straight away. That is, the time the user needs to register the color (before moving on to the corresponding confirm button) overlaps with the dwell time. In Dynamic Coloring, the color of a target is only shown after the dwell; so the time the user needs to register the color of the target starts only after the dwell, adding extra time.

Confirm buttons should have a visual anchor in the center (e.g., a crosshair) to help users hold their gaze. For on-screen buttons this anchor should be animated while the gaze dwells on the button in order to provide dynamic feedback (e.g., by gradual highlighting until activation, as shown in Figure 2 B). Note that only the anchor should be animated, as animations outside the button center can draw the gaze away. Stable confirm buttons can be off-screen to save screen real estate, e.g., as markers arranged around the screen on the case of a mobile device, as gaze trackers can usually also track the gaze slightly outside the screen region. This would make it more challenging to give dynamic feedback during activation. However, with an adequate button size, dynamic activation feedback is not necessary (e.g., 2.8 $\times$ 2.8 cm in our prototypes).

ASSIGNMENT OF VISUAL IDENTIFIERS

Clickables are associated with confirm buttons if they fall within the aforementioned association radius. If we want to have a stable assignment, no two clickables within any association radius can have the same visual identifier (as we must never have two clickables associated with the same confirm button simultaneously). Consequently, in an *assignment* of visual identifiers to clickables, the minimum distance between any clickables with the same visual identifier should be sufficiently maximized. This will work better with more visual identifiers. In particular, it can be used to disambiguate between narrow targets such as hyperlinks in adjacent lines. In principle, various algorithms can be used to find a (near) optimal assignment in this sense. We found that the following greedy algorithm is both efficient and sufficient in practice.

Algorithm for color assignment: The algorithm iterates over all clickables in their order of appearance from the top to the bottom of the page. The algorithms assigns each clickable l greedily the optimal color based on the coloring of previous clickables. The optimal color is the one for which all clickables of the same color are the furthest away from the current clickable. To find that color c the algorithm goes for each color backwards through the previously assigned colors (it can also be done for all colors in one pass). For each previous clickable l' with color c the algorithm checks the distance $\delta_m(l, l')$ between the two clickables. A natural distance measure is the minimum distance between any two pixels from each clickable. The algorithm maintains the minimum such distance. Previous clickables of the same color (including the closest one) can be found with an efficient data structure. The search window for the closest clickable of the same color is limited by the size of the screen. Concerning asymptotic runtime, the most important case where asymptotic behavior is of concern is webpages, since there is no maximum length. The algorithm is linear in the page length and quadratic in link density, i.e., in the number of clickables on the screen. This is good news since asymptotic behavior in page length is of most concern, while link density is self-delimiting, i.e., screens can only have a limited number of links for readability reasons.

The algorithm can be adapted to cater for other requirements, e.g., if confirm buttons are small (mobile devices), clickables which are close could be preferably associated with confirm buttons which are not next to each other, to improve accuracy.

Color choice: If colors are used as visual identifiers, it is important that they are easily distinguishable in order to not hamper the visual search for the right confirm buttons. We designed the palette of visual identifier colors in Figure 2 on the basis of criteria for the distinguishability of colors [10]. These criteria include separating the colors in the CIE LUV color space, separating the colors linearly (any three colors should not be on the same line in the color space), and choosing different color categories (i.e., colors from different regions associated with common color names such as "green" and "purple").

The color visual identifiers of clickables should not dominate a user interface, and for readability reasons should provide good contrasts with the colors used for fonts in labeled clickables. We address the former by reducing brightness and the latter by reducing saturation (Figure 4). Notably, our palette replaces the red in Haley's palette [10] with gray. Haley has already reported difficulties with similar colors in the red-

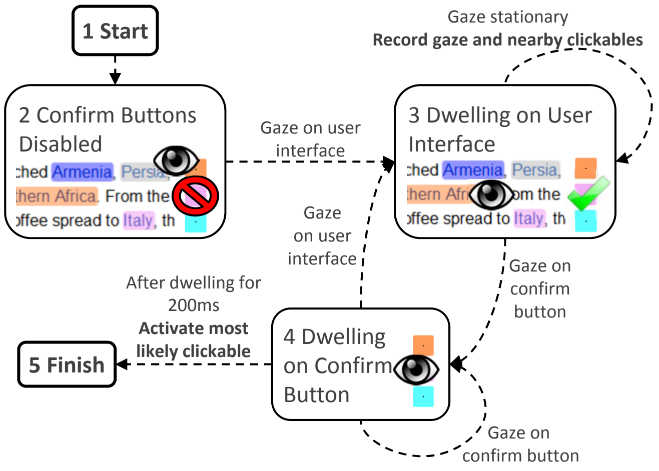

1 Start
2 Confirm Buttons Disabled
Gaze on user interface
3 Dwelling on User Interface
Gaze stationary
Record gaze and nearby clickables
Gaze on user interface
Gaze on confirm button
After dwelling for 200ms
Activate most likely clickable
5 Finish
4 Dwelling on Confirm Button
Gaze on confirm button

Figure 5. State machine of clicking with a-posteriori gaze analysis.

yellow region (having low color distance and linear separation). With the reduced brightness and saturation the distinction of red and orange showed as particularly problematic in our pilot studies. Moreover, red is a dominant and semantically loaded color. During our experiments our participants – including a red-green color blind male – had no problems distinguishing the chosen colors. Optimal ordering of visual identifiers on confirm buttons is a future work. On the one hand, one may choose a natural ordering which may facilitate visual search, e.g., by wavelength. On the other hand, similar visual identifiers may be separated to improve accuracy.

A-POSTERIORI GAZE ANALYSIS

In the Static Coloring variant, since there is no visual feedback about the current association of clickables with confirm buttons, there is in fact no need to associate clickables with confirm buttons early on. While the user is dwelling on a confirm button, the click alternative can decide *a-posteriori* which clickable should most likely be activated, based on an analysis of recorded gaze data. This simplifies the state machine, as shown in Figure 5. Such an a-posteriori analysis can retrace the gaze path backwards in time from the confirm button which the user is currently dwelling on. It can consider various criteria to decide probabilistically if a target and which target should be activated, such as:

Proximity Clickables close to the gaze path are more likely.

Color match The color of clickables must match the confirm button color.

Recency Clickables which lie on or near a more recent point in the gaze path are more likely.

Dwell time Clickables where the gaze dwelled on or nearby for longer (or above a threshold) are more likely.

For our desktop prototype (Figure 4) we found the *most-recent matching dwell* criterion to be sufficient. This uses all of the above criteria and results in the same behavior as the state machine in Figure 3. After dwelling on a confirm button with a certain color, the clickable with the same color which was most-recently dwelled on or nearby for at least 80 ms is activated. Only the clickables within an association radius of the dwell are considered (the same radius as for Dynamic Coloring).

EVALUATION

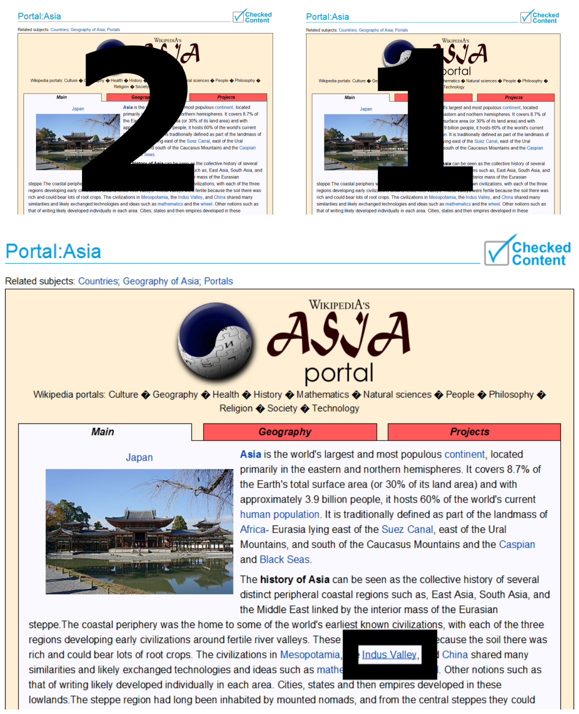
We performed an experiment to evaluate the usability of the two Actigaze variants, Dynamic Coloring and Static Coloring, and compare them with the Mouse and the predecessor method Multiple Confirm. In previous work we reviewed several gaze-only click alternatives and identified Multiple Confirm as one of the best [23]. Consequently, Multiple Confirm serves as a baseline for gaze-only click alternatives, and the mouse as a baseline for point-and click technologies in general. We did not choose direct dwell (without confirm) as a baseline because it is known to be too inaccurate – our previous experiments showed that small links are almost impossible to click with dwell alone.

Hypotheses: We expect Multiple Confirm to be significantly slower and the mouse to be significantly faster than all other alternatives. Furthermore, we expect Static Coloring to be faster than Dynamic Coloring (as discussed in the previous sections).

Setup: Dynamic and Static Coloring were implemented as described in the previous sections, with Static Coloring using a-posteriori analysis. To achieve a fair comparison, we used an identical experimental apparatus for all conditions where possible. We integrated all click alternatives into the same custom webbrowser (Figures 1, 2 and 4), using the same overall screen layout and confirm buttons of similar size and shape. We also adjusted all conditions to use the same dwell thresholds where possible, so that no condition would be disadvantaged by confounding factors. We used a Tobii X2-30 Wide remote eye gaze tracker mounted below a 23 inch LCD screen with a Dell optical mouse.

Task: Similar to Penkar et al., we used hyperlink clicking tasks to measure click time and accuracy for a typical use case. The user is shown a Wikipedia webpage with typical medium font size, and a short, regular countdown appears on top of the page (Figure 6 top). When the countdown finishes, the hyperlink to click is highlighted with a black rectangle (Figure 6 bottom), which remains visible until a link is clicked. The user is asked to click on the highlighted target as quickly and accurately as possible, using one of the click alternatives. No scrolling is required. The click time is the time between the appearance of the black rectangle and a "click" by the user. Accuracy is measured by counting the number of misclicks, i.e., clicks on an incorrect target.

This task design addresses some of the problems reported by Penkar et al.: in their study, the user had to memorize a sequence of links and then click them quickly one after another. As a result, the measurements also included time for recall and visual search, and participants were trained with the same links which the click times were measured for. Our design avoids additional time for recall and visual search and training with the same links, allowing us to obtain more realistic click time measurements. Finally, instead of using online


Portal:Asia

Checked Content

Related subjects: Countries; Geography of Asia; Portals

WIKIPEDIA'S ASIA portal

Wikipedia portals: Culture ◆ Geography ◆ Health ◆ History ◆ Mathematics ◆ Natural sciences ◆ People ◆ Philosophy ◆ Religion ◆ Society ◆ Technology

Main | Geography | Projects

Japan

Asia is the world's largest and most populous continent, located primarily in the eastern and northern hemispheres. It covers 8.7% of the Earth's total surface area (or 30% of its land area) and with approximately 3.9 billion people, it hosts 60% of the world's current human population. It is traditionally defined as part of the landmass of Africa- Eurasia lying east of the Suez Canal, east of the Ural Mountains, and south of the Caucasus Mountains and the Caspian and Black Seas.

The **history of Asia** can be seen as the collective history of several distinct peripheral coastal regions such as, East Asia, South Asia, and the Middle East linked by the interior mass of the Eurasian steppe.The coastal periphery was the home to some of the world's earliest known civilizations, with each of the three regions developing early civilizations around fertile river valleys. These cause the soil there was rich and could bear lots of root crops. The civilizations in Mesopotamia, Indus Valley, d China shared many similarities and likely exchanged technologies and ideas such as mathe Other notions such as that of writing likely developed individually in each area. Cities, states and then empires developed in these lowlands.The steppe region had long been inhabited by mounted nomads, and from the central steppes they could

Figure 6. Task: after a countdown (top-left and top-right) the link to click is highlighted with a black rectangle (bottom).

Wikipedia pages, we used the offline corpus Wikipedia for Schools[2] in order to avoid irregular load times.

Procedure: The procedure is illustrated in Figure 7. We used a within-subjects design, so every participant performed the click tasks in all four conditions. The order of conditions was counterbalanced to compensate for order bias and training effects. First participants were given a demographics questionnaire and an introduction of eye gaze tracking and the experimental tasks. For each condition, the gaze tracker was calibrated and participants had two minutes to browse freely and familiarize themselves with the click alternative if required (Free Training). 40 clicking tasks were performed; the first ten were training clicks and were not included in the data and the following 30 clicks were timed. A representative sample of click targets was chosen, with approximately an equal number of "easy", "medium" and "hard" targets in every set of training and timed clicks. For "easy" targets only one or two hyperlinks would be within the association radius around the target, for "medium" targets three or four, and for "hard" targets five or more. In the cases where the need for recalibration became evident, the current click was completed and the next click started after recalibration. Participants filled out i) a System Usability Scale (SUS) questionnaire [4] after using each click alternative and ii) a post-questionnaire with overall preference rankings and open answers at the end. The experiment took approximately 40 minutes for each participant. Each participant was rewarded with a shopping voucher.

[2]`http://schools-wikipedia.org/`

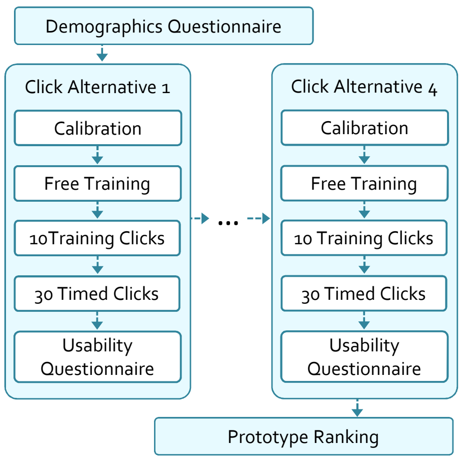


Figure 7. Experiment procedure.

Framework: There is currently no standard benchmark for and no framework supporting the development of gaze based click alternatives. To provide a stable testbed for the evaluation of gaze click alternatives, we created a framework based on a custom WebKit-based web browser written in Java. It supports common eye gaze trackers from Tobii and EyeTribe, and makes it possible to implement gaze click alternatives as plugins using an event-based interface. This made it easier to implement all click alternatives consistently and avoid confounding factors. The framework was also used to automate the experiment procedure. Sequences of hyperlink clicking tasks can be scripted in text files. The experimental procedure can be controlled at any time via keyboard hotkeys. Click times, accuracy measurements and experiment parameters such as condition and task are recorded in an event log, with a tabular structure which can be analyzed by all common statistics applications. The framework includes implementations of various click alternatives, an offline version of Wikipedia, and a script with the clicking tasks used in this study. We decided to make it open-source software and freely available to other researchers on the web[3].

Participants: A total of 25 volunteers, 6 females and 19 males from three different ethnicities aged between 18 and 45 (median 24), were recruited from the City Campus of the University of Auckland. Most of them were students, reported to be confident computer users and readers of English text, and had never used an eye gaze tracker before. Nine participants were wearing glasses or contact lenses.

Results

Every participant performed 30 timed tasks in each condition, resulting in a total of 750 measured clicks for each alternative. The data collected in this study are available online[4]. Figure 8 and Table 1 summarize the experiment results. The click times have a strongly skewed, non-normal distribution (Figure 9). Therefore we used non-parametric statistics to analyze the data.

[3]`http://github.com/aucklandhci/gazebrowser`

[4]`http://github.com/aucklandhci/gazebrowser/tree/master/datasets/`

Table 1. Summary of click times, misclicks, System Usability Scale (SUS) scores and preference ranks.

	Multiple Confirm	Dynamic Coloring	Static Coloring	Mouse
Median time (s)	2.61	1.67	1.46	1.1
95% CI	[2.37, 2.99]	[1.57, 1.77]	[1.34, 1.55]	[1.02, 1.14]
Misclicks	30 (4%)	16 (2.1%)	26 (3.5%)	14 (1.9%)
SUS avg.	74.8	78.32	76.24	90.88
95% CI	±6.02	±4.35	±6.02	±3.35
Std. dev.	14.58	10.55	14.58	8.13
Rank avg.	2.88	2.40	2.92	1.76

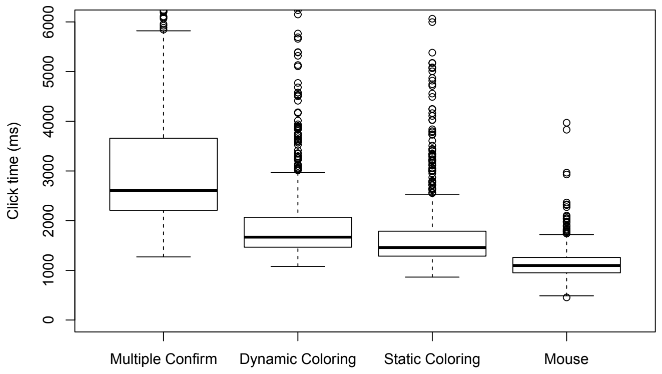


Figure 8. Boxplot of click times.

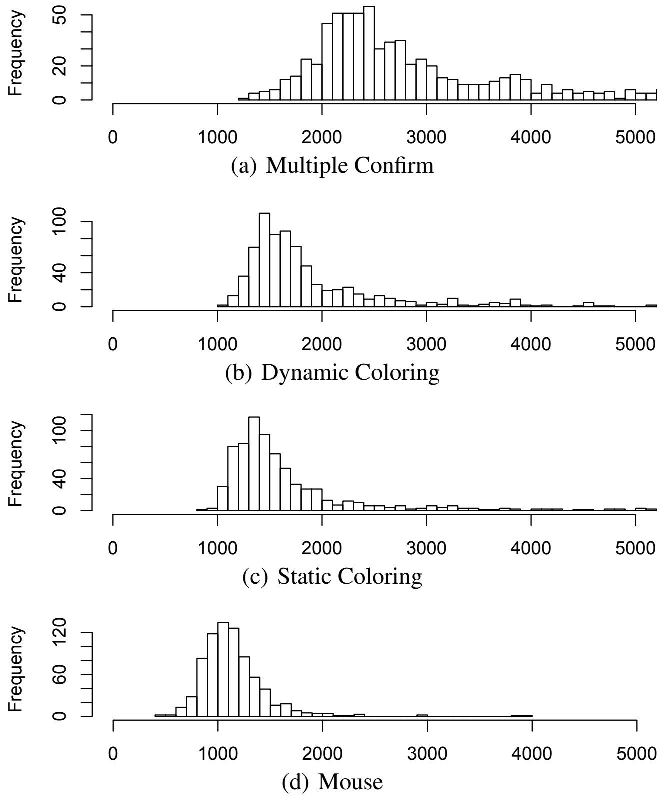


Figure 9. Frequency distributions of click times in milliseconds.

A Friedman rank sum test shows a significant effect of the click alternative on click time ($\chi^2 = 1479.47, df = 3, p \ll 0.001$). One-tailed Wilcoxon signed-rank tests show significant differences between Multiple Confirm and Dynamic Coloring ($Z = 18.26, p \ll 0.001$), Dynamic Coloring and Static Coloring ($Z = 8.44, p \ll 0.001$), and Static Coloring and Mouse ($Z = 19.83, p \ll 0.001$). All null hypotheses about click time can be rejected with high confidence.

The click time frequency distributions (Figure 9) show that the majority of click times for Multiple Confirm fall between 2 and 3 seconds, while the majority of click times for Static Coloring, Dynamic Coloring and Mouse fall between 1 and 2 seconds. Multiple Confirm has a very long tail with many slow click times. All alternatives have reasonably few misclicks ($\leq 4\%$).

A Friedman rank sum test shows a significant effect of the click alternative on the SUS scores ($\chi^2 = 24.68, df = 3, p \ll 0.001$). Two-tailed Wilcoxon signed-rank tests with Holm correction do not show significant differences between Multiple Confirm and Static Coloring ($Z = -0.20, p = 0.85$) or Dynamic Coloring and Static Coloring ($Z = 0.93, p = 0.72$), but there is a significant difference between Dynamic Coloring and Mouse ($Z = -4.19, p \ll 0.001$). A Friedman rank sum test shows a significant effect of the click alternative on preference rank ($\chi^2 = 12.73, df = 3, p = 0.005$). Two-tailed Wilcoxon signed-rank tests with Holm correction do not show significant differences between Multiple Confirm and Static Coloring ($Z = -0.27, p = 0.81$), Multiple Confirm and Dynamic Coloring ($Z = 1.34, p = 0.44$), Dynamic Coloring and Static Coloring ($Z = -1.47, p = 0.44$) or Dynamic Coloring and Mouse ($Z = 1.77, p = 0.32$), but there are significant differences between Static Coloring and Mouse ($Z = 2.84, p = 0.02$) and Multiple Confirm and Mouse ($Z = 2.86, p = 0.02$).

For some participants the eye gaze tracker did not perform well and had to be recalibrated. This was also commented on by some participants. Almost all participants remarked positively about the gaze click alternatives during the experiment, and many participants said that gaze clicking with the Actigaze alternatives felt faster than the mouse. Some participants commented that when using Multiple Confirm, they had difficulties finding the right confirm button. Some participants found the colors used in the Actigaze alternatives distracting.

Discussion

The click time and accuracy achieved with the variants of Actigaze are clearly better than Multiple Confirm, which used to be the best click alternative, and consequently better than existing gaze click alternatives. Although Actigaze is still significantly worse than the mouse, its accuracy and median click times come fairly close. It is hard to beat the mouse as the established gold standard; however, many participants erroneously perceived gaze clicking to be faster, probably because eye movement involves less physical effort than hand movement.

The increased performance is in line with the mental models associated with Multiple Confirm and the Actigaze variants. In Multiple Confirm the user performs a visual search for the correct confirm button which is typically linear in the number of confirm buttons due to their instability; on average half the confirm button labels need to be checked, making it fairly slow. By contrast, in Actigaze the visual search is nonexistent or greatly reduced after an initial training effect.

It has to be noted that for any dwell-based click alternative the dwell time adds a hard penalty, e.g., increasing the dwell time by 100 milliseconds is expected to increase all click times by that amount. The current activation dwell threshold of 200 milliseconds is chosen as being acceptable for the user and to help avoid inadvertent clicks. By subtracting the dwell thresholds we can see how much time is spent on the actual movement of the gaze: after subtracting 80 ms and 200 ms for the gaze thresholds, the remainders for the Actigaze alternatives are 1.39 and 1.18 seconds respectively. This shows that compared to the Mouse, much of the overhead of gaze clicking can be explained by dwell.

The longer click times and especially the outliers for the gaze alternatives were mostly due to gaze tracker inaccuracy, according to our observations. If gaze tracker calibration was successful then there were no problems of this kind, so this inaccuracy is not inherent, e.g., it is not due to physiological factors. Some of the negative feedback was caused by problems with the gaze tracker. These problems, combined with the necessity of a calibration process and incidental recalibration, clearly set gaze clicking apart from the mouse in terms of usability and likely had a negative effect on the usability scores and rankings. Actigaze works with any gaze tracker and will benefit from improvements in this technology. Specifically auto-calibration techniques [24] could be of great benefit for its usability.

Threats to Validity

A possible threat to validity is the implementation and adaptation of the Multiple Confirm click alternative. The alternative was implemented according to the specifications given in [22], but it is hard to recreate exactly the same system. A different eye gaze tracker was used. Moreover, the dwell parameters were adjusted in accordance with the Actigaze dwell alternatives. However, comparing the click time distribution in our experiment with those presented in [22], our implementation of Multiple Confirm seems to have performed faster than in the previous experiment. Hence, using it as a baseline seems valid.

There may be a social desirability bias in the preference data of our experiment, as many participants were from our department and may have known us. However, the study was conducted by a research assistant who was not involved in the other stages of the research, and a script was followed to reduce such effects. Anonymity was emphasized, participants were given privacy when filling out questionnaires, and they were encouraged to give accurate answers.

Some limitations of the equipment should be noted. Although the eye gaze tracker allowed for some head movement, a few participants moved too much during the experiment and had to be repositioned. Excessive movement was mostly caused by participants getting excited about the new technology (some participants turned to the experimenter to express their interest) or when participants moved away from the eye tracker while relaxing in between tasks. This may have had a negative effect on the click alternatives.

CONCLUSION

We have presented Actigaze, a novel, purely gaze-based click alternative for discrete clickables. With the design and evaluation of Actigaze, we have made the following contributions:

- Techniques to achieve stability of confirm buttons to enable the use of procedural memory.
- An algorithm to assign visual identifiers to clickables in order to avoid inadvertent clicks.
- Two variants which trade off visual appearance and speed.
- An a-posteriori analysis technique which simplifies the state machine of one of the variants (Static Coloring).
- A palette of easily distinguishable color visual identifiers.
- An experimental comparison of Actigaze against the mouse and the Multiple Confirm click alternative.

The results of the user study with 25 participants show that Actigaze significantly improves performance over the Multiple Confirm click alternative [23], which is used in practice. In our experiment the click time was improved by about a second, without loss of accuracy. To help other researchers, we make our framework for the development and evaluation of gaze click alternatives freely available.

There are several future works. Actigaze can be combined with auto-calibration techniques in order to reduce setup time. The time thresholds could be tuned or adapted dynamically to reduce the immanent time penalties of purely gaze-based interaction. A longitudinal study of Actigaze could verify long-term usability and the performance gain expected from frequent use. With practice the use of confirm buttons may become ‘automatic’, i.e., without conscious cognitive effort similar to touch typists who do not have to think about the locations of the keys.

With gaze tracking for mobile devices becoming more reliable and mainstream, it would be logical to explore the use of Actigaze specifically for mobile use. Further studies can also explore the added value of Actigaze with other types of interfaces, e.g., for the conventional WIMP-style. For web browsing Actigaze could be studied in the presence of other gaze-controlled features such as scrolling, forward/backward navigation buttons and form input. Another stream of research would be to consider how gaze clicking could work together with other input modalities, e.g., touch, voice and gestures.

REFERENCES

1. Abe, K., Owada, K., Ohi, S., and Ohyama, M. A system for web browsing by eye-gaze input. *Electronics and Communications in Japan 91*, 5 (2008), 11–18.

2. Ashmore, M., Duchowski, A. T., and Shoemaker, G. Efficient eye pointing with a fisheye lens. In *Proc. Graphics Interface*, Canadian HCI Society (2005), 203–210.
3. Blanch, R., and Ortega, M. Rake cursor: improving pointing performance with concurrent input channels. In *Proc. CHI*, ACM (2009), 1415–1418.
4. Brooke, J. SUS - A quick and dirty usability scale. *Usability evaluation in industry 189* (1996), 194.
5. Castellina, E., and Corno, F. Accessible web surfing through gaze interaction. In *Proc. Conference on Communication by Gaze Interaction (COGAIN)* (2007).
6. Drewes, H., and Schmidt, A. Interacting with the computer using gaze gestures. In *Proc. INTERACT*. Springer, 2007, 475–488.
7. Grauman, K., Betke, M., Lombardi, J., Gips, J., and Bradski, G. Communication via eye blinks and eyebrow raises: video-based human-computer interfaces. *Universal Access in the Information Society 2* (2003), 359–373.
8. Grossman, T., and Balakrishnan, R. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. CHI*, ACM (2005), 281–290.
9. Hayhoe, M., and Ballard, D. Eye movements in natural behavior. *Trends in Cognitive Sciences 9*, 4 (2005), 188 – 194.
10. Healey, C. G. Choosing effective colours for data visualization. In *Proc. Visualization*, IEEE (1996), 263–270.
11. Huckauf, A., and Urbina, M. H. On object selection in gaze controlled environments. *Journal of Eye Movement Research 2*, 4 (2008), 4.
12. Huckauf, A., and Urbina, M. H. Object selection in gaze controlled systems: What you don't look at is what you get. *ACM Trans. Appl. Percept. 8* (February 2011), 13:1–13:14.
13. Jacob, R. J. K. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Trans. Inf. Syst. 9* (1991), 152–169.
14. Kumar, M., Paepcke, A., and Winograd, T. Eyepoint: practical pointing and selection using gaze and keyboard. In *Proc. CHI*, ACM (2007), 421–430.
15. Kumar, M., and Winograd, T. GUIDe: Gaze-enhanced UI design. In *Proc. CHI*, ACM (2007), 1977–1982.
16. Land, M. F. Eye movements and the control of actions in everyday life. *Progress in Retinal and Eye Research 25*, 3 (2006), 296 – 324.
17. Lankford, C. Effective eye-gaze input into windows. In *Proc. Symposium on Eye Tracking Research & Applications (ETRA)*, ACM (2000), 23–27.
18. MacKenzie, I. S. An eye on input: research challenges in using the eye for computer input control. In *Proc. Symposium on Eye-Tracking Research & Applications (ETRA)*, ACM (2010), 11–12.
19. Majaranta, P., and Räihä, K.-J. Twenty years of eye typing: systems and design issues. In *Proc. Symposium on Eye Tracking Research & Applications (ETRA)*, ACM (2002), 15–22.
20. Møllenbach, E., Lillholm, M., Gail, A., and Hansen, J. P. Single gaze gestures. In *Proc. Symposium on Eye-Tracking Research & Applications (ETRA)*, ACM (2010), 177–180.
21. Ohno, T. Features of eye gaze interface for selection tasks. In *Proc. Asia Pacific Conference on Computer Human Interaction (APCHI)* (1998), 176–181.
22. Penkar, A. M., Lutteroth, C., and Weber, G. Designing for the eye: design parameters for dwell in gaze interaction. In *Proc. Australian Computer-Human Interaction Conference (OzCHI)*, ACM (2012), 479–488.
23. Penkar, A. M., Lutteroth, C., and Weber, G. Eyes only: Navigating hypertext with gaze. In *Proc. INTERACT*, Springer (2013), 153–169.
24. Pfeuffer, K., Vidal, M., Turner, J., Bulling, A., and Gellersen, H. Pursuit calibration: Making gaze calibration less tedious and more flexible. In *Proc. UIST*, ACM (2013), 261–270.
25. Sibert, L. E., and Jacob, R. J. K. Evaluation of eye gaze interaction. In *Proc. CHI*, ACM (2000), 281–288.
26. Skovsgaard, H., Mateo, J. C., Flach, J. M., and Hansen, J. P. Small-target selection with gaze alone. In *Proc. Symposium on Eye-Tracking Research & Applications (ETRA)*, ACM (2010), 145–148.
27. Stampe, D. M., and Reingold, E. M. Selection by looking: A novel computer interface and its application to psychological research. *Studies in Visual Information Processing 6* (1995), 467–478.
28. Urbina, M. H., Lorenz, M., and Huckauf, A. Pies with eyes: the limits of hierarchical pie menus in gaze control. In *Proc. Symposium on Eye-Tracking Research & Applications (ETRA)*, ACM (2010), 93–96.
29. Ware, C., and Mikaelian, H. H. An evaluation of an eye tracker as a device for computer input. In *Proc. CHI*, ACM (1987), 183–188.
30. Yeoh, K. N., Lutteroth, C., and Weber, G. Eyes and keys: An evaluation of click alternatives combining gaze and keyboard. In *Proc. INTERACT*, Springer (2015).
31. Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (magic) pointing. In *Proc. CHI*, ACM (1999), 246–253.
32. Zhang, X., and MacKenzie, I. S. Evaluating eye tracking with iso 9241-part 9. In *Proc. HCI International (LNCS 4552)*, Springer (2007), 779–788.