Spatio-Temporal Modeling and Prediction of Visual Attention in Graphical User Interfaces

Pingmei Xu
Princeton University
pingmeix@princeton.edu

Yusuke Sugano
Max Planck Institute for Informatics
sugano@mpi-inf.mpg.de

Andreas Bulling
Max Planck Institute for Informatics
bulling@mpi-inf.mpg.de

ABSTRACT

We present a computational model to predict users' spatio-temporal visual attention on WIMP-style (windows, icons, menus, pointer) graphical user interfaces. Like existing models of bottom-up visual attention in computer vision, our model does not require any eye tracking equipment. Instead, it predicts attention solely using information available to the interface, specifically users' mouse and keyboard input as well as the UI components they interact with. To study our model in a principled way, we further introduce a method to synthesize user interface layouts that are functionally equivalent to real-world interfaces, such as from Gmail, Facebook, or GitHub. We first quantitatively analyze attention allocation and its correlation with user input and UI components using ground-truth gaze, mouse, and keyboard data of 18 participants performing a text editing task. We then show that our model predicts attention maps more accurately than state-of-the-art methods. Our results underline the significant potential of spatio-temporal attention modeling for user interface evaluation, optimization, or even simulation.

Author Keywords

Visual attention; saliency; interactive environment; graphical user interfaces; physical action; spatio-temporal modeling

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces – Evaluation/Methodology

INTRODUCTION

Human gaze serves a dual purpose in human-computer interaction. For one, gaze is appealing for hands-free interaction with pervasive interfaces, since it is faster than the mouse for pointing [44] and intuitive and natural to use [52]. Gaze therefore has a long history as an input modality for tasks ranging from desktop control [27], eye typing [35] and target selection [47] to password entry [10], cross-device content transfer [51], and notification display [19]. Common to all of these uses is that gaze is implemented as an explicit input, i.e. an input that users actively employ to interact with an interface. At the same time, gaze naturally indicates what we are interested in and what we attend to. Gaze has therefore also been used as an implicit input, e.g. for recognition of users' activities and cognitive processes [11, 12] or as a measure of users' visual attention while interacting with a user interface [9, 33].

CHI'16, May 07 - 12, 2016, San Jose, CA, USA

ACM 978-1-4503-3362-7/16/05 $15.00
DOI: http://dx.doi.org/10.1145/2858036.2858479

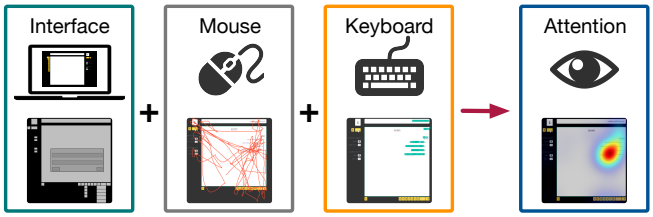


Figure 1: We present a computational model to predict users' spatio-temporal visual attention on a graphical user interface based solely on information about the interface as well as users' mouse and keyboard input.

A fundamental limitation for both uses of gaze in human-computer interaction is that estimating gaze requires special-purpose eye tracking equipment. Unfortunately, eye trackers may not always be available, have to be calibrated to each user prior to first use, and tracking is limited to a confined area in front of the interface [34]. Even more importantly, eye trackers themselves only provide users' current and past gaze locations. They do not provide information on which locations or components the user will likely attend to or interact with in the future. Such information is valuable for intelligent user interfaces, for example, to proactively adapt to users' needs [42]. While computer vision methods that only require ordinary cameras have matured considerably [55, 57], they are still inferior in terms of gaze estimation accuracy.

A promising solution to both problems may be provided by computational models of visual attention, i.e. models that mimic basic perceptual concepts to reproduce human attentive behavior [25]. Typically taking a single image as input, these models aim to predict those locations whose local visual attributes significantly differ from the surrounding image, and which are therefore most likely to be attended to next by the observer. Originally introduced in neuroscience, these models were successful in a range of fields, most notably computer vision for tasks such as object detection or image segmentation. Despite close parallels to interactive systems, such as the problem of predicting the next interaction location, few works have investigated the use of such mod-

els [36, 48]. However, existing attention models are limited because they mainly rely on visual components, and they capture neither user input, such as from mouse and keyboard, nor interface information, such as components, nor do they take the history of user interactions with the interface into account.

We aim to address these limitations by presenting a computational model to predict users' spatio-temporal visual attention on WIMP-style (windows, icons, menus, pointer) graphical user interfaces. Like existing models of bottom-up visual attention in computer vision, our model does not require any eye tracking equipment. Instead, it predicts attention solely using information available to the interface, specifically users' past mouse and keyboard input actions and the UI components they interacted with. This is in contrast to computer vision models that use only image features, such as intensity, color or orientation [25]. To study our model in a principled way, we introduce a method to synthesize new user interface layouts that are functionally equivalent to real-world interfaces, such as from Gmail, Facebook, or GitHub. Finally, using these synthesized layouts, we quantitatively analyze attention allocation and its correlation with user input and UI components using ground-truth gaze, mouse, and keyboard data of 18 participants performing a text editing task. We further show that our model predicts attention maps more accurately than state-of-the-art methods.

We believe that computational modeling of spatio-temporal visual attention in the GUI space has significant potential to answer fundamental questions relevant for attentive user interfaces, such as how well attention is aligned and coordinated with other input, e.g. from mouse and keyboard, how and when users allocate their attention to different GUI components relevant for a given task, how they develop and employ task-specific attention allocation behavior, and how consistent such behavior is for one user and across different users.

RELATED WORK

Our work builds on existing methods for (1) computational modeling of visual attention, (2) gaze prediction from other input modalities, (3) modeling and prediction of user behavior during user interface interactions.

Computational Modeling of Visual Attention

Modeling visual attention is an active area of research [4]. Current attention models either take only visual features calculated from an image (so-called bottom-up models) or task-related features (top-down models) as input. The output is a map that topographically encodes the probability of visual attention over the whole image. Itti et al. proposed one of the first bottom-up models [26]. The model computes normalized center-surround difference maps of individual image features, such as orientation, intensity and contrast. Other approaches predicted attention using Bayesian models [56], information maximization [7], bottom-up graphical models [22], and region covariance [18]. To incorporate temporal information, more recent models fuse static and dynamic attention maps, for example, to estimate visual attention in video [8, 31]. Similarly, data-driven approaches that use machine learning and large amounts of training gaze data have become popular, such as SALICON [28] and the Judd model [30]. Because bottom-up models use only image information, their performance degrades considerably for situations in which attention is mainly influenced by top-down factors, such as the user's intent, goals, or tasks. To address this limitation, another line of research focused on modeling attention from top-down factors, such as task [39] or scene context [16].

In contrast, only a few previous works investigated the use of bottom-up models for HCI purposes. Masciocchi et al. and Still et al. showed that attention maps generated using the original Itti model [25] correlate well with fixations during free-viewing of web pages [36, 48]. Shen and Zhao [43] developed a learning-based bottom-up model to predict attention on web pages while Borji et al. [5] introduced a method for computational modeling of top-down visual attention in driving video games. In contrast, our model is suited for general WIMP-style user interfaces and is explicitly designed for non-free-viewing conditions, i.e. conditions in which the user performs a particular task, since these conditions are most important for interactive human-computer interfaces.

Gaze Prediction from Other Input Modalities

To address challenges associated with eye tracker availability and usability, a large body of work explored the use of other input modalities as a gaze replacement. For example, several works demonstrated that mouse position can be used as a proxy for gaze location [1, 2, 21, 24, 37] and mouse click positions can be even used to calibrate an eye tracker [49]. Other works studied the correlation between cursor position and gaze in more detail, including the temporal relationship between gaze and cursor. For example, Bieg et al. [3] showed that gaze generally leads the mouse cursor, while Liebling and Dumais [32] studied the temporal relationship between cursor and gaze and showed that they get closest around 100 – 250 ms before a click event. A similar relationship was identified between typing behavior and gaze location [29, 53]. Other works investigated eye-hand coordination during specific tasks, such as target selection [45], visual search [3, 15], web browsing [13], or web search [23, 38, 41]. None of these works integrated information from multiple inputs, such as mouse and keyboard, into a joint model, nor did they combine these inputs with information about interface components or past actions of the user to predict attention spatio-temporally.

Modeling User Behavior in UI Interactions

Given that users' input is closely related to their tasks, several works used them as features to model user behavior in UI interactions. Mouse movements and position, for example, were used to infer user intent during web search [20], user states in video browsing [54], or user tasks in e-learning applications [17]. Others directly incorporated gaze information to recognize user activities in web [14] and visualization interfaces [50, 46]. In contrast to what is proposed here, all of these methods focused on user modeling and activity recognition. The link to attention prediction, in particular in a spatio-temporal fashion and for general-purpose user interfaces, as well as the idea of combining multiple inputs with information about the interface, were not considered.

Application	Sample	G1	G2	G3	G4	G5	G6	G7	G8
Email		✔	✔	✔		✔			✔
Blog		✔	✔	✔	✔	✔	✔		✔
Photo comment			✔	✔		✔	✔	✔	
Update status			✔	✔		✔	✔		
Share content		✔	✔			✔	✔		
Short message			✔		✔	✔			
Product review		✔	✔			✔		✔	✔
Documentation		✔	✔		✔	✔	✔		

Figure 2: Application scenarios, sampled products, and groups of UI components for text editing tasks. Components are grouped into eight categories: title/short description (G1), main content (G2), text formatting (G3), meta information/setting (G4), finish button (G5), profile icon (G6), image (G7), and new window (G8).

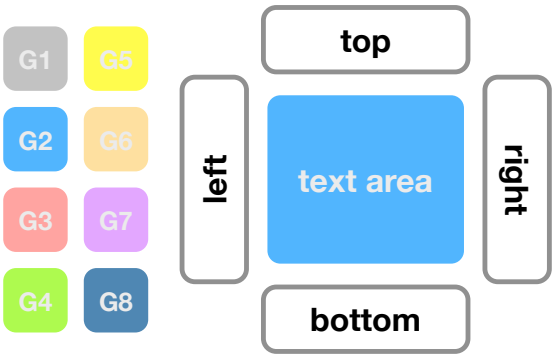


Figure 3: To synthesize new interfaces for the text editing task, we group UI components by functionality and randomly place them into four areas around the main input area.

MODELING INTERACTION PATTERNS

In this work, we take a data-driven approach to analyze and predict visual attention in the GUI space. We chose text editing interfaces because they represent real-world interactive tasks in terms of the most common styles of interactions related to GUI. To maximize the generality of the model and analysis, it is essential to collect data covering a range of different UI layouts and functionalities. However, the variation of real-world implementations is huge and it is challenging to collect a sufficient amount of samples.

To address these problems, we first create a generic model of UIs based on real-world examples. This model is not only beneficial to synthesize realistic UI pattern samples for data collection, but also guides the feature design for learning-based attention prediction.

(a) (b)

Figure 4: Sample real-world application interfaces for Gmail (a) and Blogger (b) with overlaid color masks indicating the different functional UI groups.

Real-World UI Samples

To cover the most frequent uses of text editing in everyday life, we first sampled 8 web application examples from the most popular websites: writing an email (Gmail), composing a blog (Blogger), commenting on a photo on a social network (Instagram), posting a status on a social network (Facebook), sharing a link (Tumblr), posting a short message (Twitter), writing a product review for online shopping (Amazon), and documenting code (GitHub README).

Figure 4 shows examples of the UI samples. From these examples, we can see that UI layouts and functionalities are not completely random but are composed of some common design patterns. These UI patterns can be characterized by 1) functionalities, 2) layout, and 3) appearances.

Functionality

Although each application is composed of various UI elements, their fundamental functionalities can be summarized into fewer categories. We first grouped each UI components into several function categories: title/short description (G1), main content (G2), text formatting (G3), meta information/setting (G4), finish button (G5), profile photo (G6), image (G7) and new window (G8).

Figure 2 shows how these function categories correspond to the real-world examples. It can be seen that the categorization is general enough to cover real-world variations of the UI patterns.

Layout

We characterized different UI layout patterns using a layout grid. The grid consists of a main text area and four spaces, left, right, top and bottom to the main text area (see Figure 3). Figure 4 (a)-(b) show how each UI example corresponds to the layout grid.

Appearance

In contrast to the functionality and layout, it can be seen that appearance of the design pattern has a larger degree of freedom. At least from the examples above, we cannot observe any tendencies about color and size of the UI components. This indicates the fundamental difficulty of attention predic-

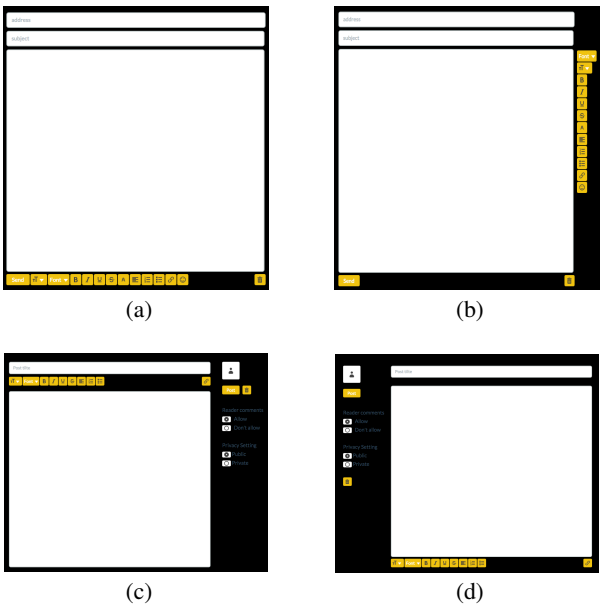


Figure 5: Sample synthesized interfaces for email (a, b) and blog (c, d). The interfaces have randomized layouts but contain UI components with the same functionality as the real-world interfaces.

tion during text editing tasks: visual information becomes less significant in this case.

Layout Synthesis

Based on these observations, we define our UI model as follows. We first identify key UI components for each application example and match the components across different applications; for example, a "send" button for sending an email is matched with a "publish" button for publishing a blog. In other words, one application definition gives a subset of function categories with application-specific semantic definitions.

Given these function lists, we further randomized the positioning of UI components by assigning the location of each UI group into the layout grid. G1 and G2 always corresponded to the text area, and the UI components G3, G4, G5, G6, and G7 were arranged into these four subareas as groups, as shown in Figure 5 (a)-(d). In each subarea, the number of UI groups was set to no more than three to avoid clutter. For G8, which refers to pop-up windows such as the input form where the user supplies a URL when inserting a link, it was always set to the center when it became visible.

For UI appearance we used a consistent design for the size, texture and icon background to remove the bias caused by the specific samples that we chose. To increase the appearance variation, we randomly assigned the color theme for each task, choosing from 4 options: blue, green, red, and yellow.

DATA COLLECTION

We used our method to synthesize 30 different – yet functionally equivalent – user interface layouts derived from real-world interfaces, such as Gmail, Facebook, and GitHub (see Figure 2). We then designed a user study to collect behavioral data (mouse, keyboard, and gaze behavior) of multiple users interacting with these different layouts. This approach allowed us to collect large amounts of data that covers realistic daily-life interaction scenarios.

Participants and Apparatus

We recruited 18 participants (6 females and 12 males, aged between 20 and 30 years) through mailing lists. They were paid 10 EUR for participating in our one-hour recording. All participants used at least one of the applications on a daily basis. All of them had normal or corrected-to-normal vision.

The experiment was conducted on a desktop PC and a display with a resolution of 1920×1200 px and refresh rate of 60 Hz. Stabilizing their heads using a chinrest, the distance of their eyes from the screen was about 55 cm. Gaze data was recorded using a Tobii TX300 stationary eye tracker running at 300 Hz and providing an accuracy of $0.5°$. We used the Tobii Studio software to analyze the gaze data; fixations and saccades were automatically detected by the software using default parameters. An optical mouse was used with the same sensitivity and acceleration for all participants. The experiment software was run in a Chrome web browser in full-screen mode (black background). We implemented the software in JavaScript to log the meta information of UIs and users' activities. Specifically, we logged the positioning and appearance of UI components and recorded the time series of mouse positions (x, y) where the origin is the upper left corner of the screen, mouse clicks, keyboard typing, and dynamic changes of UIs such as the appearance of float windows and drop-down menus.

Procedure

The experiment was split into three recording sessions with 6–7 text editing tasks each. We calibrated the eye tracker before each session using Tobii's default 9-point calibration routine. To reduce fatigue, participants took at least a 2-minute break between sessions. The order of tasks was randomized across participants. The UI layout of each task was selected from 30 layouts covering eight real-world interfaces as described in the previous section. The color theme was randomly selected from 4 options. Before each task we provided participants with a general and vague hint as to what they could write about (e.g. "Please write a blog entry about your hometown", "Please write an email to invite your friend for dinner") to reduce the time and effort required to contemplate the content. The synthetic GUIs were fully functional and participants were encouraged to use their full functionality, including text formatting, but also to complete the text editing task as quickly as possible by writing a chunk of text no more than five sentences long. On average, the overall study time of each participant was one hour, and the participants performed 6.1 ($\sigma = 6.6$) mouse clicks per task.

In total, we recorded gaze and interaction data for 245 text editing tasks. We recorded 34,695 fixations, 3,884 mouse clicks, and 43,158 key presses. The average completion time for one task was 98.4 s ($\sigma = 73.8s$). After the experiment, we

asked participants to rate the realism of the text editing tasks compared to what they were used to in their daily lives from 1 to 5 (1: not realistic at all, 5: fully realistic). The average rating was 4.3 ($\sigma = 0.7$).

ANALYSIS OF VISUAL ATTENTION

Several previous works investigated correlations between gaze and mouse movements [13, 41] as well as gaze and cursor position [23] in specific tasks, such as web browsing. However, none of these works compared users' visual attention, mouse and keyboard input, and interaction behavior with UI components on a graphical user interface in a controlled and principled manner. The analysis gives us a general overview on how visual attention is allocated in the UI space, and guides us in establishing the attention prediction model.

Correlations with User Input

We first analyzed the correlation between attention and user input including mouse and cursor. Figure 6 shows an example for one user-task pair. As illustrated in Figure 6 (b), we observed 3 frequent patterns of the interaction between attention (eye), action (mouse and keyboard) and user interface in the coordinate space (AAUC), including pattern 1) mouse following the eye, pattern 2) eye focusing on cursor during text editing, and pattern 3) mouse remaining stationary while the eye inspected text content or UI components.

To evaluate this observation quantitatively, we examined the correlation between eye movement and user input. For mouse input, we used the recorded mouse movement which is the mouse (x, y) position. For keyboard input, we used the (x, y) position of the cursor, which refers to the blinking mark placed in the text area to indicate a proposed position for insertion. In the recording, the cursor was only visible when the current, focused UI component is a text input area. We extracted the position and timestamps of mouse and cursor per user, per task. Since eye fixations occurred every 200–500 ms and the positions of the mouse and cursor were logged whenever they changed, we sampled time uniformly every 10ms and interpolated the positions by assigning the previous eye fixation and most recent mouse and cursor position to each timestamp. We computed the Pearson product-moment correlation coefficient, which measures the linear dependence between two variables giving a value between $+1$ and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation. We also estimated the lag of a signal compared with the fixation by finding the peak of the cross-correlation of the two time sequences.

Table 1 shows that eye and mouse correlate in both directions. Additionally, the mouse tends to lag behind gaze by more than 100 ms on average. Both results support the finding of AAUC-pattern 1. Besides, the strong correlation in the x direction for eye-cursor indicates that when the participant was editing text, the focus of attention mainly landed right on the next position for adding new content, which supports AAUC-pattern 2. Furthermore, since the active time for the mouse and cursor is non-overlapping, by combining the mouse and cursor, we got a higher correlation between attention and user input in both the x and y directions.

	Attention-action correlation in pixel space			
Signal	*x*	*lag x (ms)*	*y*	*lag y (ms)*
Mouse	0.35	114	0.51	193
Cursor	0.44	2	0.36	9
Combined	0.60	-	0.56	-

Table 1: Correlation and lag in the x and y direction between eye movement and user input including the mouse trajectory, the cursor position, and the combination of both.

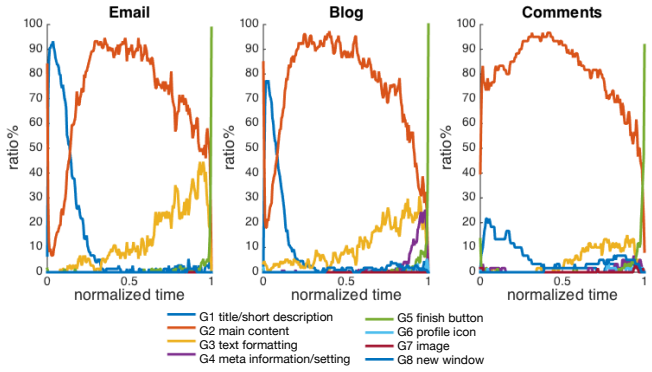


Figure 7: Statistically there exist some typical work flows for users to complete a text editing task even across different applications. At each time, the fraction of dwell for one UI group is computed by calculating the portion of tasks in the dataset with that group of UI as the focused UI group.

Interaction Flow Patterns

To understand visual attention, temporal evolution of attention allocation is also an important factor. When interacting with a GUI, visual attention can be strongly influenced by the semantic meaning of the interface components. Having a task in mind, the user performed a sequence of basic computing tasks (such as selecting a piece of text and editing the format of selected text) to achieve the goal. Although different users may choose different methods to accomplish each basic computing task (such as menu item selection or pressing a key combination when changing the format of a word to bold), if there exists a typical workflow for users to complete a task at a higher level, then it can serve as a useful cue that we can rely on to predict the current interface component of interest and the corresponding attention allocation.

To answer this question, we evaluated the distribution of the focused UI group over time. At each time point, the focused UI group was determined by mouse selection. As seen in Figure 7, at the beginning of the task, participants tended to start on the general description of the task, for instance the title of the blog or the subject of the email. They then spent most of their time working on the main content while simultaneously changing the format of the text. Before finishing the task, they edited the meta information, for instance, choosing a privacy setting for the post. Generally speaking, the distribution of the focused UI group at each time point also suggests

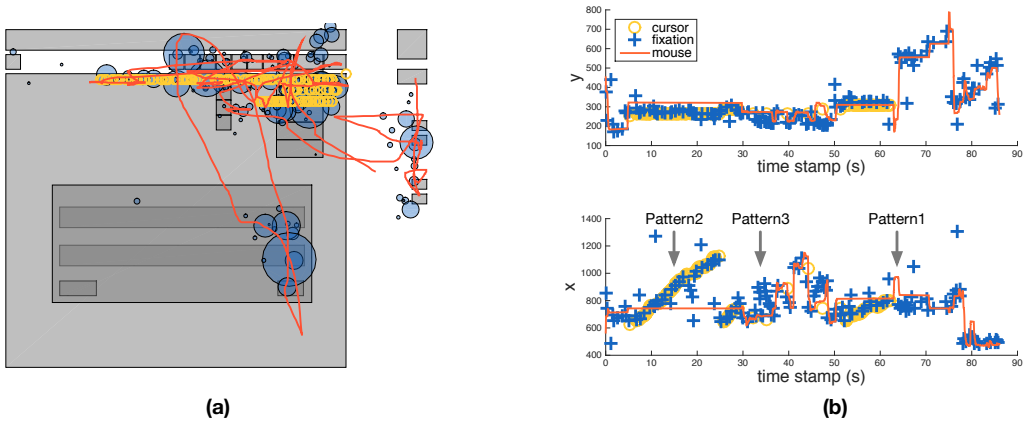


Figure 6: Sample spatio-temporal interaction patterns. (a) Bounding boxes of UI components and tracks of eye, mouse, and cursor position. (b) The same positions over time. Also shown are three frequently occurring patterns: 1) mouse follows the eye, 2) eye focuses on cursor during text editing, and 3) the mouse remains stationary while the eye inspects text or UI components.

the probability distribution of visual attention focusing across UI groups. Furthermore, from the perspective of top-down attention prediction, the fact that this distribution changes over time implies that to accurately predict "what" is attended to visually by the user, a model should (automatically) capture the temporal evolution of the focused UI group as well.

SPATIO-TEMPORAL MODELING OF VISUAL ATTENTION

Figure 8 shows an overview of our approach for modeling users' visual attention on a GUI. Our model takes information about the interface as well as users' mouse and keyboard actions as input, computes individual feature channels from the raw data recorded over time, and predicts joint spatio-temporal attention maps, which indicates the likelihood of users' attention focusing on each location over time.

In a general form, estimating the attention map is a regression problem. For each pixel location of the target UI space, we compute a feature vector $\mathbf{m}$ based on the feature channels extracted from the raw data. Then we seek a function mapping from $\mathbf{m}$ to v which is the pixel value of the ground-truth attention map corresponding to the same location. In our case, ground-truth attention maps of training data are obtained using an eye tracker. We use a generalized linear model parametrized by a weight vector $\mathbf{w}$ to represent this mapping; the predicted attention value at each pixel is indicated by the weighted sum of the pixel value at the corresponding location from the observed maps. Mathematically, the goal is to optimize the following objective function:

$$\arg\min_{\mathbf{w}} \sum_{U} \sum_{T} \sum_{P} \| \sum_{i=k:K} w_k m_k - v \|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

where K is the number of feature channels, U indicates the set of users, T indicates the set of tasks performed by a user, P is the number of pixel samples used for training and λ is a regularization parameter to prevent overfitting. This is a standard optimization problem which has a closed-form solution.

In this work we consider two cases, static and dynamic attention prediction, depending on whether the temporal aspect is taken into account. For dynamic attention prediction we further discuss two cases, offline and online, depending on whether the testing is conducted during or after the recording.

Features

As discussed in the previous section, three AAUC patterns are frequently observed, and they illustrate close correlation with eye movement for the mouse, keyboard, and interface components, respectively. To combine these three factors (mouse, keyboard, and interface) into the feature for training a predictive model, we use the following information from the raw data: mouse and cursor positions and UI element locations.

For the mouse and cursor positions, we create a binary map with those pixels set to 1 that have a mouse/cursor dwell and 0 for the others, given a set of (x, y) positions which correspond to a set of time points. Which set of time points to use depends on the training/testing setting, and will be discussed in a later section. For the bounding boxes of UI groups, we create a binary map for each of them based on the location of the bounding box (1 for pixels inside the box, 0 for others). Then we convolved these binary maps with a Gaussian filter (with a cut off frequency $-6dB$) in the same fashion as computing the attention map from fixations, as shown in Figure 8. This yields one mouse map, one cursor map, and G UI maps as features.

Since the fixation locations are very sparse compared to the total number of pixels, the numbers of fixated pixels and non-fixated ones are very unbalanced. To address this issue we adopt a similar strategy as Judd et al. [30] during training: for each attention map, we first keep all pixels with a value above a threshold, then randomly sample the same number of pixels from the rest. We set the threshold as a fixed fraction of the maximum value in the attention map. We only use the sampled pixels for training the model. During testing, we compute the feature vector for each pixel location and predict its value within the attention map.

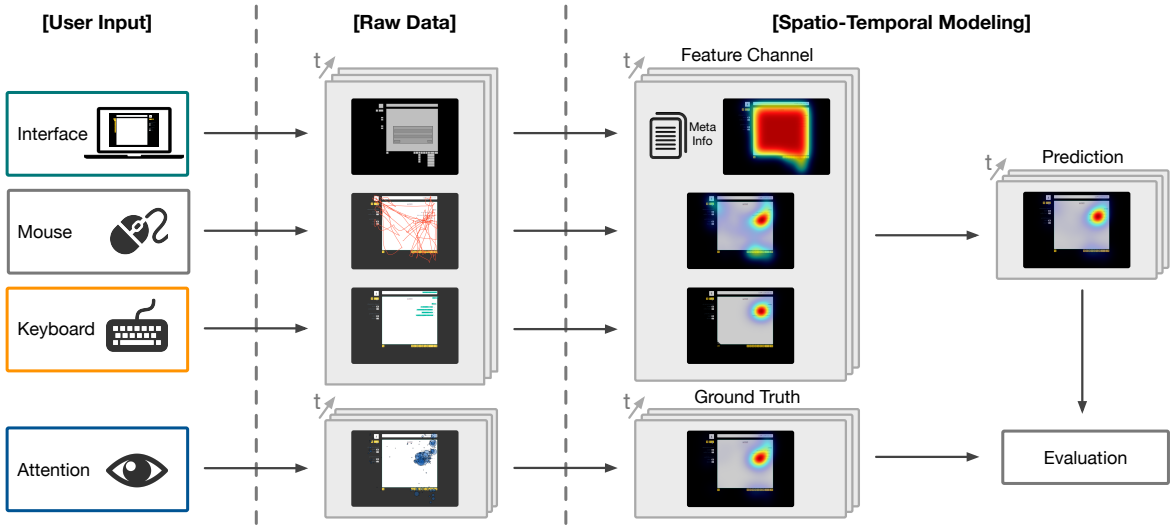


Figure 8: Our model for visual attention prediction in graphical user interfaces takes information about the interface as well as users' mouse and keyboard actions as input. The model then computes individual feature channels from the raw data recorded over time and predicts joint spatio-temporal attention maps. For evaluation purposes, we compare these predicted attention maps with ground truth attention information recorded using an eye tracker.

Static Attention Prediction

We first address the problem of predicting a static attention map which indicates the spatial distribution of the attention of one user using one interface performing one task. This case assumes an offline analysis, and the goal is to predict the whole attention distribution over the GUI space given a recording of user activities on one UI and one task.

The procedure is similar to previous bottom-up models dealing with static images, where the fixations with positions (x, y) of one user performing one task are accumulated over time and the corresponding attention map is used as the ground truth for training as shown in Figure 9. For mouse/cursor, we first accumulated the set of (x, y) positions into a binary map, and then applied Gaussian convolution. In order to train a generalized model for tasks with a different set of UI groups, if a certain UI group doesn't appear we set its UI map to all zero values; otherwise, we compute a UI map based on the location and size of its bounding box.

Dynamic Attention Prediction

While the static model is expected to capture spatial attention distribution efficiently, it is often more important to predict time-dependent localization of visual attention. As discussed previously, the attention allocation to each UI group changes over time (see Figure 7). The dynamic attention prediction model emphasizes this temporal aspect of users' attention and aims to predict the spatial location of attention over time instead of predicting one global distribution. In this section, we consider both offline and online analysis scenarios.

Offline Model

We first normalize the time span of each task to $[0, 1]$, then evenly sample T time points and train a model $\mathcal{M}(t)$ for each time point t. In this case, there is only one fixation used as the ground truth at time t as shown in Figure 11. The positions of fixation, mouse, and cursor at t are defined as the last observed previous position. For each time t we extract features from a time window $[t - d, t + d]$ around t. Unlike the static case, these maps are not aggregated but used as a set of $2d + 1$ maps. Another important aspect of the offline analysis is that we can also know the normalized timestamp t of the test data. Therefore the attention prediction at each time t is directly computed by $\mathcal{M}(t)$.

Online Model

For interactive tasks, it is also important to assume an online analysis scenario. In contrast to the offline model, in this case, the total duration of the current task is unknown. Hence it is required to first predict the normalized time t from the user's previous activities. To this end, we additionally train a normalized time prediction model during the training. We first compute the histogram of the fraction of dwell time on each UI group as in Figure 7 for $[0, t]$, and train a linear regression model to map this histogram to the normalized time t. During testing, we first estimate the normalized time t and then choose the corresponding $\mathcal{M}(t)$ for prediction.

RESULTS

Our model was trained on half of the users and tested on the other half. For dynamic attention prediction we evenly sampled 200 time points in the range $[0, 1]$. For the offline model we set $d = 10$ data points. The threshold that we used for sampling the training data was 0.1 times the maximum value of the fixation map.

Metrics

To evaluate the accuracy of our proposed models, we used the following metrics commonly used in the visual saliency literature:

	Static attention prediction accuracy				
Method	*NSS*	*AUC-Judd*	*AUC-Borji*	*CC*	*Similarity*
GT	3.99	0.97	0.96	1.00	1.00
Gaussian	0.61	0.76	0.75	0.20	0.27
Mouse	3.19	0.95	0.86	0.80	0.66
GBVS [22]	2.62	0.89	0.82	0.63	0.49
SALICON [28]	2.92	0.93	0.77	0.74	0.53
CovSal [18]	2.89	0.54	0.82	0.69	0.52
Ours	**3.43**	**0.96**	**0.89**	**0.86**	**0.73**

Table 2: Performance comparison for static attention prediction as well as the ground-truth (GT) performance.

	Dynamic attention prediction accuracy				
Method	*NSS*	*AUC-Judd*	*AUC-Borji*	*CC*	*Similarity*
Gaussian	0.86	0.78	0.77	0.13	0.10
Mouse	4.54	0.86	0.77	0.47	0.42
GBVS [22]	2.07	0.91	0.86	0.27	0.18
SALICON [28]	2.97	0.94	0.84	0.35	0.27
CovSal [18]	2.44	0.92	0.79	0.29	0.23
Ours (offline)	**6.26**	**0.98**	**0.97**	**0.65**	**0.26**
Ours (online)	**5.53**	**0.98**	**0.97**	**0.59**	**0.24**

Table 3: Performance comparison for dynamic attention prediction.

- *Normalized Scan-Path Saliency (NSS)* [40]. This measure is calculated as the mean value of the normalized attention map s at n fixation locations: $NSS = \frac{1}{n}\sum_{i=1}^{n}\frac{s(x_i,y_i)\mu_s}{\sigma_s}$.
- *Area Under ROC curve (AUC)*. The attention map is treated as a binary classifier; the pixels with a value above a threshold are classified as fixated, while the rest are classified as non-fixated pixels. By thresholding over this map and plotting the true positive rate versus false positive rate curve (ROC curve), AUC is calculated as the area underneath the curve. Different variations of the AUC metric exist; we used *AUC-Judd* [30] and *AUC-Borji* [6].
- *Correlation Coefficient (CC)*. This measure is the linear correlation coefficient between human attention map h and a predicted attention map s (CC=0 for uncorrelated maps): $CC(s,h) = \frac{cov(s,h)}{\sigma_s\sigma_h}$.
- *Similarity*. This measures the similarity between two different attention maps when viewed as distributions (SIM=1 means the distributions are identical). The maps are first normalized to sum to 1, then $Similarity(s,h) = \sum_i \min(s_i, h_i)$.

In order to obtain a ground truth attention distribution, we convolve a Gaussian filter across the users' fixation locations [30].

Static Attention Prediction

We compared our model with three state-of-the-art attention models: graph-based visual saliency (GBVS) [22], Saliency in Context (SALICON) [28], and attention from region covariances (CovSal) [18]. We also employed a naive baseline that always predicts a central Gaussian, and we included human performance by directly using the location of fixations. Table 2 summarizes prediction scores of these models. As can be seen from the table, our model performs best across all employed metrics. In particular, the model achieves a NSS of 3.43 (human: 3.99), an AUC-Judd of 0.96 (ground truth: 0.97), and an AUC-Borji of 0.89 (ground truth: 0.96). While the CC (0.86) and Similarity (0.73) metrics are slightly worse (ground truth: 1.00), our model still outperforms all other models by a considerable margin. The next best attention predictions are achieved using only mouse information, which is competitive with our model particularly for the AUC-Judd (0.95) and AUC-Borji (0.86) metrics. As expected, the naive baseline performs the worst among all models. These results underline the importance of combining information capturing users' actions as well as UI components for static attention prediction. We speculate that for user interfaces with more complex appearances, such as an image editor or a news feed on a social network, the predictive power of traditional attention models can be leveraged to improve performance even further.

Figure 9 shows sample attention maps for static attention prediction using our model, individual user inputs, and established bottom-up attention models for the email writing task. Note that, as per definition, the maps for fixation, mouse, and cursor are computed by accumulating samples over the whole task. As can be seen in the figure, the attention map predicted by our model matches well with the ground-truth gaze data obtained from the eye tracker. In contrast, information on mouse and keyboard input alone is not sufficient to model the complex spatio-temporal attention patterns that emerge during the email writing task. While mouse information correctly predicts a high-attention area at the bottom of the screen, it misses the high-attention area at the top left (and vice versa when using only keyboard information). The figure also shows that established bottom-up attention models lag even further behind. While the high-attention areas predicted by GBVS mainly align with high-contrast regions of the user interface, SALICON and CovSal only predict a central uninformative region.

Dynamic Attention Prediction

The output for dynamic attention prediction is a sequence of fixation (x, y) positions over time, as shown in Figure 10. Since there is only one fixation for each point in time during testing, for evaluation purposes, we chose two scores (*NSS* and *Auc-Judd*). Both scores directly use fixation positions (instead of attention maps, as ground truth) to evaluate the accuracy of proposed models.

As shown in Table 3, our model performs best across both metrics for both the offline and online case. For known users, the best performance is achieved by our offline (NSS 6.26, AUC 0.97) and online models (NSS 5.53, AUC 0.98). They are followed, at a considerable distance, by the mouse (NSS 4.54, AUC 0.77) as well as the bottom-up models and the naive baseline. Similar performance behavior can be seen for the case of new users. These results show that in the case of dynamic attention prediction, there is a large performance gap between pure bottom-up models and models that take users'

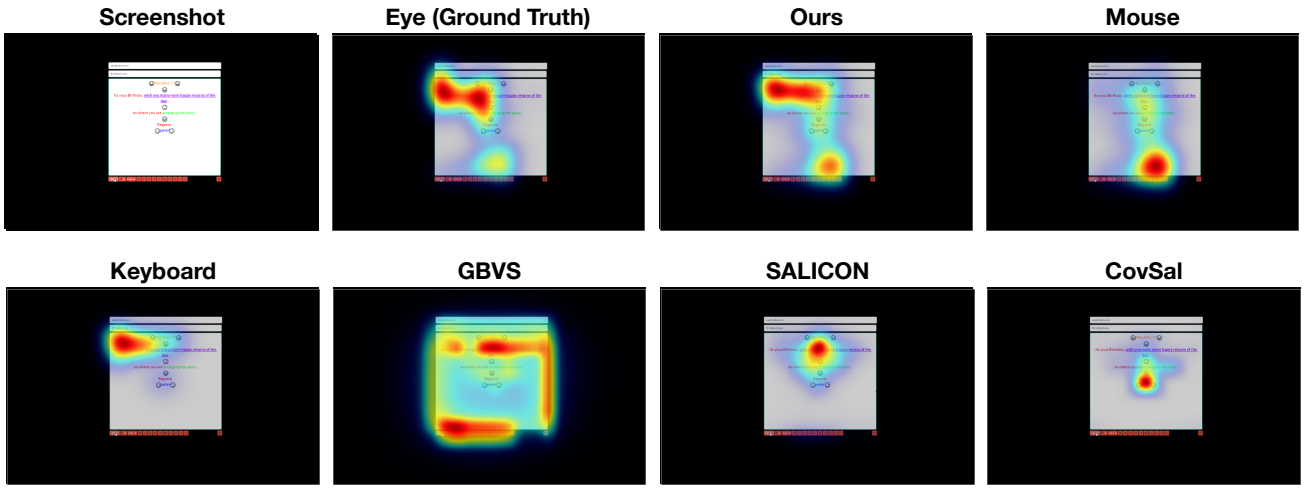


Figure 9: Sample attention maps for the email writing task for static attention prediction using the different models: spatio-temporal visual attention model (Ours), mouse or keyboard input, graph-based visual saliency (GBVS), Saliency in Context (SALICON), attention from region covariances (CovSal). Ground-truth attention map was obtained using an eye tracker.

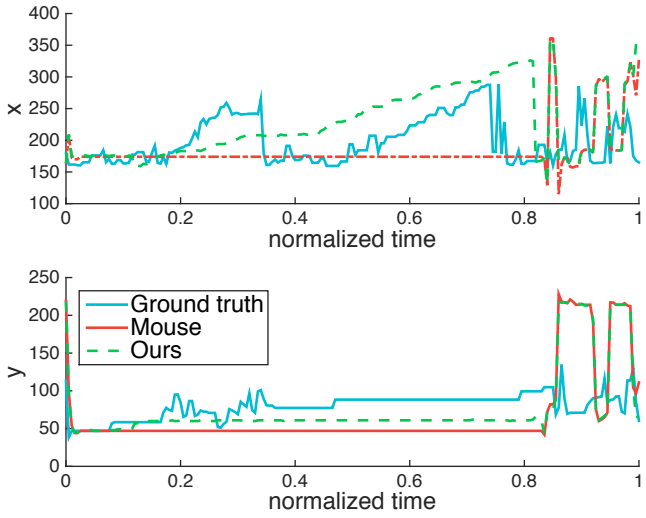


Figure 10: The predicted (x, y) fixation position over time for dynamic attention prediction using different models. The time is normalized to $[0, 1]$.

actions into consideration. In an interactive setting, the fixation location of the current point in time greatly depends on users' previous activities and current mental state, so the visual information plays a less important role in this case.

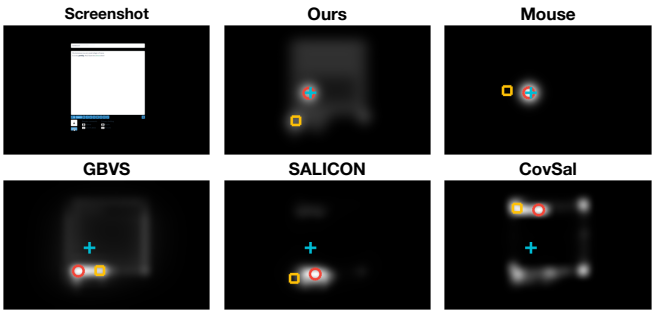
In Figure 6 we mentioned that three patterns of AAUC are often observed. A prediction model should have the capacity to capture these patterns. As illustrated in Figure 11, our model successfully achieves this goal by designing the features based on previous observations, which also explains the performance gap between our models and other methods.

Furthermore, we found that our regression methods for predicting the normalized time point for online testing provided reasonable results with time prediction error 0.19 (σ=0.15). This, again, confirms the observation that there exist typical workflows that can be generalized, although the task, the UI, and the users vary across each recording. Among different settings of our models, the offline model performs better than the online model by adding information from further time-stamps in the feature.

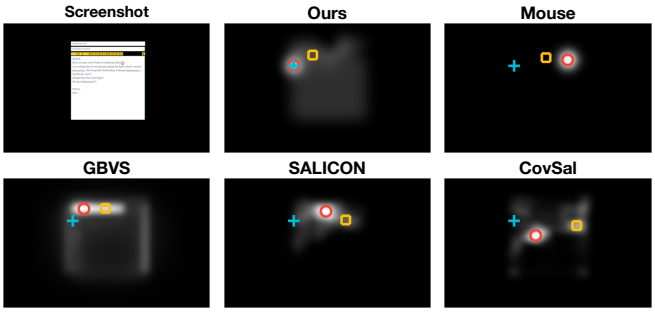
DISCUSSION

In this work we presented a computational model for predicting users' spatio-temporal visual attention for graphical user interfaces. The key advantage of our model is that it does not require any eye tracking equipment. Instead, it predicts attention solely using information available to the interface, specifically users' mouse and keyboard input as well as the UI components they interact with. By taking text-editing as a use case and through extensive evaluations, we demonstrated that our model outperforms state-of-the-art methods by a large margin. We showed that visual attention prediction in an interactive environment can be dramatically improved by taking both information about the interface as well as users' mouse and keyboard actions into account.

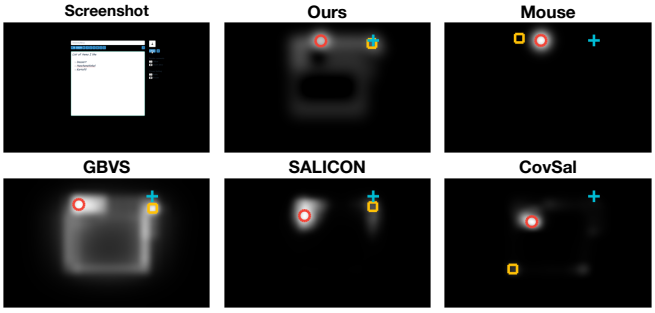
To evaluate our model, we introduced a method to group user interface components according to their functionality and to synthesize new user interface layouts derived from real-world interfaces. This method proved very useful, as it enabled us to reduce the influence of other factors on visual attention. In addition, by focusing on the functionality of UI components, we were able to study different application scenarios jointly and develop a generic model applicable to all scenarios. The method has considerable potential beyond the current study, because it can be easily scaled up to a much larger number of interfaces and users. It can also be easily extended to other tasks beyond text editing, and we therefore believe it can be developed into a general tool for studying UI design variations in a principled manner.



(a)



(b)



(c)

Figure 11: Sample screenshots of text editing tasks and corresponding maps for next fixation prediction by different models. The blue cross indicates the human fixation, and the red circle and the yellow square are the first and second maximum point of each map after non-maximum suppression, respectively. By incorporating mouse and cursor input, our model provided correct prediction when AAUC pattern 1 (a) and AAUC pattern 2 (b) occurred. By taking the information of interface into account as well, our model even predicted accurately by a second guess when AAUC pattern 3 (c) occurred.

We recorded a dataset that contains synchronized mouse and keyboard as well as gaze data on synthesized UI layouts. By analyzing the data, we were able to identify common patterns of interaction across visual attention, physical action, and user interface components. We also found that different users share consistent attentive behavior when performing similar tasks. Taken together, these findings show that computational modeling of spatio-temporal visual attention has significant potential to answer fundamental questions in attentive user interface design, such as how users allocate visual attention over different GUI components and how physical input, visual attention, and task interplay with each other. Spatio-temporal modeling of users' visual attention therefore also has potential for more general application scenarios, such as user interface evaluation, optimization, or simulation.

While the results from our evaluations are promising, we identified several opportunities for extending and improving our model. For example, future work could evaluate the proposed modeling approach for other types of user interfaces, interaction tasks, or platforms, such as public displays or handheld personal devices. Since visual attention, available input modalities, and coordination with other inputs will differ across these different platforms, these evaluations will provide valuable insights into the generality of the approach. In addition, while in this work we focused on mouse and keyboard input only, it will be very interesting to see whether and how other inputs can be incorporated into the model, such as gesture, speech, or touch input.

CONCLUSION

In this paper, we presented a computational model to predict users' spatio-temporal visual attention on graphical user interfaces. In order to systematically control and study the influence of different sources of information on visual attention, we also introduced a method to utilize synthesized user interface layouts. We then conducted data collection studies and demonstrated the effectiveness of the proposed model by comparing our model with state-of-the-art methods. We believe that our work provides valuable tools for understanding users' behavior in interactive environments.

Acknowledgements

This work was funded, in part, by the Cluster of Excellence on Multimodal Computing and Interaction (MMCI) at Saarland University, the Alexander von Humboldt Foundation, and a JST CREST research grant.

REFERENCES

1. E. Arroyo, T. Selker, and W. Wei. 2006. Usability tool for analysis of web designs using mouse tracks. In *Ext. Abstr. CHI*. 484–489. DOI: http://dx.doi.org/10.1145/1125451.1125557
2. R. Atterer, M. Wnuk, and A. Schmidt. 2006. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *Proc. WWW*. 203–212. DOI: http://dx.doi.org/10.1145/1135777.1135811
3. H. J. Bieg, H. Reiterer, and H. H. Bülthoff. 2010. Eye and pointer coordination in search and selection tasks. In *Proc. ETRA*. DOI: http://dx.doi.org/10.1145/1743666.1743688
4. A. Borji. 2013. State-of-the-art in visual attention modeling. *IEEE TPAMI* (2013). DOI: http://dx.doi.org/10.1109/TPAMI.2012.89
5. A. Borji, D. N. Sihite, and L. Itti. 2011. Computational modeling of top-down visual attention in interactive

environments. In *Proc. BMVC*. 1–12. DOI: http://dx.doi.org/10.5244/C.25.85

6. A. Borji, H. R. Tavakoli, D. N. Sihite, and L. Itti. 2013. Analysis of scores, datasets, and models in visual saliency prediction. *ICCV* (2013). DOI: http://dx.doi.org/10.1109/ICCV.2013.118
7. N. Bruce and J. Tsotsos. 2006. Saliency based on information maximization. In *Proc. NIPS*.
8. N. D. B. Bruce and J. K. Tsotsos. 2008. Spatiotemporal saliency: towards a hierarchical representation of visual saliency. *Proc. WAPCV* (2008). DOI: http://dx.doi.org/10.1007/978-3-642-00582-4_8
9. A. Bulling. 2016. Pervasive Attentive User Interfaces. *IEEE Computer* 49, 1 (2016), 94–98. DOI: http://dx.doi.org/10.1109/MC.2016.32
10. A. Bulling, F. Alt, and A. Schmidt. 2012. Increasing the Security of Gaze-Based Cued-Recall Graphical Passwords Using Saliency Masks. In *Proc. CHI*. 3011–3020. DOI: http://dx.doi.org/10.1145/2207676.2208712
11. A. Bulling, C. Weichel, and H. Gellersen. 2013. EyeContext: Recognition of High-level Contextual Cues from Human Visual Behaviour. In *Proc. CHI*. 305–308. DOI: http://dx.doi.org/10.1145/2470654.2470697
12. A. Bulling and T. O. Zander. 2014. Cognition-Aware Computing. *IEEE Pervasive Computing* 13, 3 (July 2014), 80–83. DOI: http://dx.doi.org/10.1109/mprv.2014.42
13. M. C. Chen, J. R. Anderson, and M. H. Sohn. 2001. What can a mouse cursor tell us more? Correlation of eye/mouse movements on web browsing.. In *Proc. CHI*. DOI: http://dx.doi.org/10.1145/634067.634234
14. F. Courtemanche, E. Aïmeur, A. Dufresne, M. Najjar, and F. Mpondo. 2011. Activity recognition using eye-gaze movements and traditional interactions. *Interacting with Computers* 23, 3 (2011), 202–213. DOI: http://dx.doi.org/10.1016/j.intcom.2011.02.008
15. A. L. Cox and M. M. Silva. 2006. The role of mouse movements in interactive search. In *Proc. CogSci*. 1156–1161.
16. K. Ehinger, B. Hidalgo-Sotelo, A. Torralba, and A. Oliva. 2009. Modeling search for people in 900 scenes. *Visual Cognition* (2009). DOI: http://dx.doi.org/10.1080/13506280902834720
17. A. Elbahi, M. A. Mahjoub, and M. N. Omri. 2013. Hidden Markov model for inferring user task using mouse movement. In *Proc. ICTA*. 1–7. DOI: http://dx.doi.org/10.1109/ICTA.2013.6815305
18. E. Erdem and A. Erdem. 2013. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision* (2013). DOI: http://dx.doi.org/10.1167/13.4.11
19. J. E. Garrido, V. M. R. Penichet, M. D. Lozano, A. Quigley, and P. O. Kristensson. 2014. AwToolkit: attention-aware user interface widgets. *In Proc. AVI* (2014). DOI: http://dx.doi.org/10.1145/2598153.2598160
20. Q. Guo and E. Agichtein. 2008. Exploring mouse movements for inferring query intent. In *Proc. SIGIR*. 707–708. DOI: http://dx.doi.org/10.1145/1390334.1390462
21. Q. Guo and E. Agichtein. 2010. Towards predicting web searcher gaze position from mouse movements. In *Ext. Abstr. CHI*. 3601–3606. DOI: http://dx.doi.org/10.1145/1753846.1754025
22. J. Harel, C. Koch, and P. Perona. 2007. Graph-based visual saliency. In *Proc. NIPS*.
23. J. Huang, R. White, and G. Buscher. 2012. User see, user point: Gaze and cursor alignment in web search. In *Proc. CHI*. DOI: http://dx.doi.org/10.1145/2207676.2208591
24. J. Huang, R. W. White, and S. Dumais. 2011. No clicks, no problem: using cursor movements to understand and improve search. In *Proc. CHI*. 1225–1234. DOI: http://dx.doi.org/10.1145/1978942.1979125
25. L. Itti and C. Koch. 2001. Computational modelling of visual attention. *Nature reviews neuroscience* 2, 3 (2001), 194–203. DOI: http://dx.doi.org/10.1038/35058500
26. L. Itti, C. Koch, and E. Niebur. 1998. A model of saliency based visual attention for rapid scene analysis. *PAMI* (1998). DOI: http://dx.doi.org/10.1109/34.730558
27. R. J. K. Jacob. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proc. CHI*. 11–18. DOI: http://dx.doi.org/10.1145/97243.97246
28. M. Jiang, S. Huang, J. Duan, and Q. Zhao. 2015. SALICON: Saliency in context. In *Proc. CVPR*. DOI: http://dx.doi.org/10.1109/CVPR.2015.7298710
29. R. Johansson, Å. Wengelin, V. Johansson, and K. Holmqvist. 2010. Looking at the keyboard or the monitor: relationship with text production processes. *Reading and writing* 23, 7 (2010), 835–851. DOI: http://dx.doi.org/10.1007/s11145-009-9189-3
30. T. Judd, K. Ehinger, F. Durand, and A. Torralba. 2009. Learning to predict where humans look. In *Proc. ICCV*. DOI: http://dx.doi.org/10.1109/ICCV.2009.5459462
31. J. Li, Y. Tian, T. Huang, and Gao W. 2010. Probabilistic multi-task learning for visual saliency estimation in video. *IJCV* (2010). DOI: http://dx.doi.org/10.1007/s11263-010-0354-6

32. D. J. Liebling and S. T. Dumais. 2014. Gaze and Mouse Coordination in Everyday Work. In *Adjunct Proc. UbiComp '14*. 1141–1150. DOI: http://dx.doi.org/10.1145/2638728.2641692

33. P. P. Maglio, R. Barrett, C. S. Campbell, and T. Selker. 2000. SUITOR: An attentive information system. In *Proc. IUI*. 169–176. DOI: http://dx.doi.org/10.1145/325737.325821

34. P. Majaranta and A. Bulling. 2014. *Eye Tracking and Eye-Based Human-Computer Interaction*. Springer Publishing, 39–65. DOI: http://dx.doi.org/10.1007/978-1-4471-6392-3_3

35. P. Majaranta and K. J. Räihä. 2002. Twenty years of eye typing: systems and design issues. In *Proc. ETRA*. 15–22. DOI: http://dx.doi.org/10.1145/507072.507076

36. C. M. Masciocchi and J. D. Still. 2013. Alternatives to Eye Tracking for Predicting Stimulus-Driven Attentional Selection Within Interfaces. *Human–Computer Interaction* 28, 5 (2013), 417–441. DOI: http://dx.doi.org/10.1080/07370024.2012.731332

37. V. Navalpakkam and E. Churchill. 2012. Mouse Tracking: Measuring and Predicting Users' Experience of Web-based Content. In *Proc. CHI*. 2963–2972. DOI: http://dx.doi.org/10.1145/2207676.2208705

38. V. Navalpakkam, L. Jentzsch, R. Sayres, S. Ravi, A. Ahmed, and A. Smola. 2013. Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *Proc. WWW*.

39. R. J. Peters and L. Itti. 2007. Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention. In *Proc. CVPR*. DOI: http://dx.doi.org/10.1109/CVPR.2007.383337

40. R. J. Peters, A. Iyer, L. Itti, and C. Koch. 2008. Objects predict fixations better than early saliency. *Journal of Vision* (2008). DOI: http://dx.doi.org/10.1167/8.14.18

41. K. Rodden, X. Fu, A. Aula, and I. Spiro. 2008. Eye-mouse coordination patterns on web search results pages. In *Ext. Abstr. CHI*. 2997–3002. DOI: http://dx.doi.org/10.1145/1358628.1358797

42. D. D. Salvucci and J. R. Anderson. 2000. Intelligent gaze-added interfaces. In *Proc. CHI*. 273–280. DOI: http://dx.doi.org/10.1145/332040.332444

43. C. Shen and Q. Zhao. 2014. Webpage Saliency. In *Proc. ECCV*. 33–46. DOI: http://dx.doi.org/10.1007/978-3-319-10584-0_3

44. L. E. Sibert and R. J. K. Jacob. 2000. Evaluation of eye gaze interaction. In *Proc. CHI*. 281–288. DOI: http://dx.doi.org/10.1145/332040.332445

45. B. Smith, J. Ho, W. Ark, and S. Zhai. 2000. Hand eye coordination patterns in target selection. In *Proc. ETRA*. DOI: http://dx.doi.org/10.1145/355017.355041

46. B. Steichen, G. Carenini, and C. Conati. 2013. User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proc. IUI*. 317–328. DOI: http://dx.doi.org/10.1145/2449396.2449439

47. S. Stellmach and R. Dachselt. 2012. Look & touch: gaze-supported target acquisition. In *Proc. CHI*. 2981–2990. DOI: http://dx.doi.org/10.1145/2207676.2208709

48. J. D. Still and C. M. Masciocchi. 2010. A Saliency Model Predicts Fixations in Web Interfaces. In *Proc. MDDAUI*.

49. Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike. 2015. Appearance-Based Gaze Estimation With Online Calibration From Mouse Operations. *IEEE THMS* PP, 99 (2015), 1–11. DOI: http://dx.doi.org/10.1109/THMS.2015.2400434

50. D. Toker, C. Conati, B. Steichen, and G. Carenini. 2013. Individual user characteristics and information visualization: connecting the dots through eye tracking. In *Proc. CHI*. 295–304. DOI: http://dx.doi.org/10.1145/2470654.2470696

51. J. Turner, A. Bulling, J. Alexander, and H. Gellersen. 2014. Cross-Device Gaze-Supported Point-to-Point Content Transfer. In *Proc. ETRA*. 19–26. DOI: http://dx.doi.org/10.1145/2578153.2578155

52. M. Vidal, A. Bulling, and H. Gellersen. 2013. Pursuits: Spontaneous Interaction with Displays based on Smooth Pursuit Eye Movement and Moving Targets. In *Proc. UbiComp*. 439–448. DOI: http://dx.doi.org/10.1145/2468356.2479632

53. Å. Wengelin, M. Torrance, K. Holmqvist, S. Simpson, D. Galbraith, V. Johansson, and R. Johansson. 2009. Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior research methods* 41, 2 (2009), 337–351. DOI: http://dx.doi.org/10.3758/BRM.41.2.337

54. B. Westphal and T. Syeda-Mahmood. 2002. On learning video browsing behavior from user interactions. In *Proc. WWW*.

55. E. Wood and A. Bulling. 2014. EyeTab: Model-based gaze estimation on unmodified tablet computers. In *Proc. ETRA*. 207–210. DOI: http://dx.doi.org/10.1145/2578153.2578185

56. L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrel. 2008. SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision* (2008). DOI: http://dx.doi.org/10.1167/8.7.32

57. X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. 2015. Appearance-Based Gaze Estimation in the Wild. In *Proc. CVPR*. 4511–4520. DOI: http://dx.doi.org/10.1109/CVPR.2015.7299081