Int. J. Human-Computer Studies 111 (2018) 78–91

Contents lists available at ScienceDirect

International Journal of Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Gaze-based predictive user interfaces: Visualizing user intentions in the presence of uncertainty

Çağla Çığ Karaman*, Tevfik Metin Sezgin

Intelligent User Interfaces Lab, Department of Computer Engineering, Koç University, Istanbul, 34450, Turkey

ARTICLE INFO

Keywords:
Implicit interaction
Activity prediction
Task prediction
Uncertainty visualization
Gaze-based interfaces
Predictive interfaces
Proactive interfaces
Gaze-contingent interfaces
Usability study

ABSTRACT

Human eyes exhibit different characteristic patterns during different virtual interaction tasks such as moving a window, scrolling a piece of text, or maximizing an image. Human-computer studies literature contains examples of intelligent systems that can predict user's task-related intentions and goals based on eye gaze behavior. However, these systems are generally evaluated in terms of prediction accuracy, and on previously collected offline interaction data. Little attention has been paid to creating real-time interactive systems using eye gaze and evaluating them in online use. We have five main contributions that address this gap from a variety of aspects. First, we present the first line of work that uses real-time feedback generated by a gaze-based probabilistic task prediction model to build an adaptive real-time visualization system. Our system is able to dynamically provide adaptive interventions that are informed by real-time user behavior data. Second, we propose two novel adaptive visualization approaches that take into account the presence of uncertainty in the outputs of prediction models. Third, we offer a personalization method to suggest which approach will be more suitable for each user in terms of system performance (measured in terms of prediction accuracy). Personalization boosts system performance and provides users with the more optimal visualization approach (measured in terms of usability and perceived task load). Fourth, by means of a thorough usability study, we quantify the effects of the proposed visualization approaches and prediction errors on natural user behavior and the performance of the underlying prediction systems. Finally, this paper also demonstrates that our previously-published gaze-based task prediction system, which was assessed as successful in an offline test scenario, can also be successfully utilized in realistic online usage scenarios.

1. Introduction

For several years, great effort has been devoted to developing gaze-based prediction models that capture human behavior patterns naturally accompanying virtual interaction tasks such as reading an electronic document, or manipulating a virtual object (Fig. 1) (Bader et al., 2009; Bednarik et al., 2012; Campbell and Maglio, 2001; Çığ and Sezgin, 2015a; Courtemanche et al., 2011; Steichen et al., 2013).

However, existing models are generally evaluated in terms of prediction accuracy, and within offline scenarios that assume perfect knowledge about user's task-related intentions and goals. Such scenarios are called wizard-based test scenarios. Note that, in this paper, "online usage" does not refer to real-life usage scenarios. Online/offline distinction is made not based on how realistic the user interface is but based on whether the predictions are fed back to the user during interaction. In an example offline wizard-based test scenario, the users are asked to either select an object, or to manipulate a previously selected object (Bader et al., 2009). Collected data with labels corresponding to user intentions are then used to compute the accuracy of the related intention prediction model. The output of the prediction model is in no way shown to the users. In other words, in the wizard-based test scenarios, the loop between the user and the prediction system is open, i.e. the user is fed hardwired and perfect visual feedback via the user interface irrespective of predictions made by the prediction system (Fig. 2a). Existing studies do not take into account how these models would perform in the absence of wizards. They also do not examine how/if the prediction errors affect the quality of interaction. In this paper, we eliminate the wizard assumption and close the loop between the user and the prediction system. We achieve this by feeding highly accurate but imperfect predictions (since we do not have prediction systems that can perform with 100% accuracy yet) made by the prediction system to the user via appropriate visualizations of the user interface (Fig. 2b). By means of a thorough usability study, we seek answers to the following research questions: (1) How should a user interface adapt its behavior accord-

* Corresponding author.
E-mail addresses: ccig@ku.edu.tr (Ç. Çığ Karaman), mtsezgin@ku.edu.tr (T.M. Sezgin).

https://doi.org/10.1016/j.ijhcs.2017.11.005
Received 18 April 2017; Received in revised form 29 November 2017; Accepted 30 November 2017
Available online 2 December 2017
1071-5819/

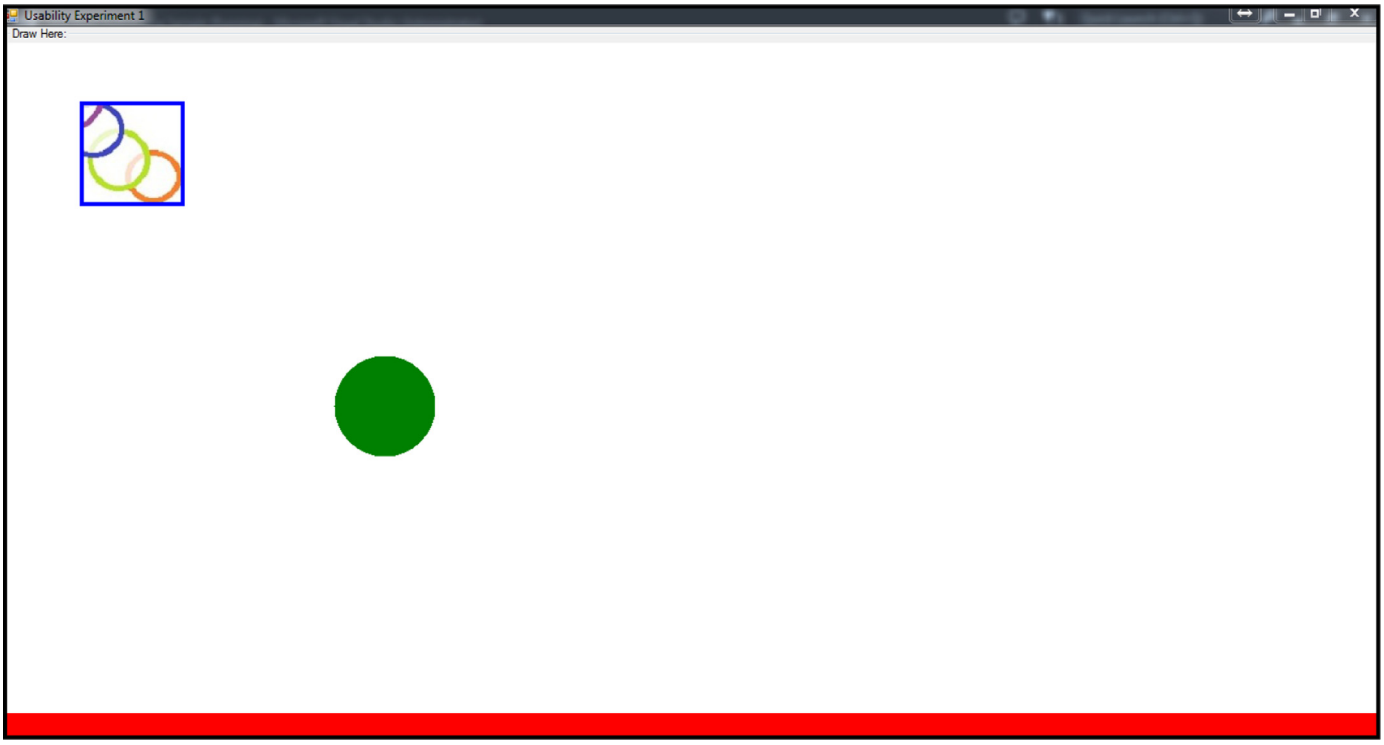


Fig. 1. Screen capture of one of our predictive user interfaces visualizing a virtual interaction task. User's task is to drag the blue square (located on the upper-left of the screen) onto the center of the green circle (located on the bottom-right of the screen). We use our gaze-based virtual task prediction model to predict user's task-related intentions and goals in real-time. Furthermore, we assist the user by automatically triggering various user interface adaptations that reflect these predictions. By adaptation, we mean the adaptation of the screen contents in terms of the visibility of visual feedback corresponding to possible tasks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

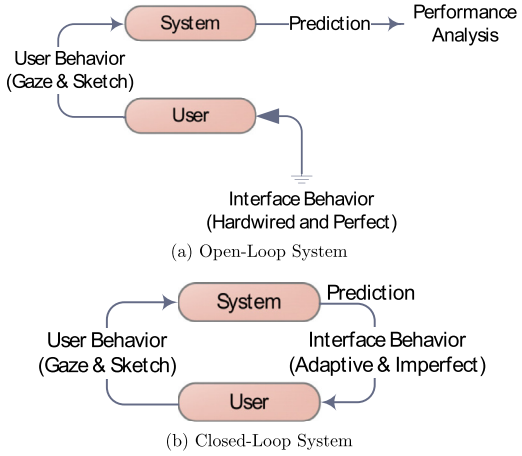


Fig. 2. Closing the loop between the user and the prediction system. The user behavior affects system prediction which in turn may affect user behavior.

ing to real-time predictions made by the underlying prediction system? (2) Will adaptations affect user behavior and inhibit performance of the prediction system (that assumes natural human behavior)? (3) Will prediction errors affect user behavior and inhibit performance of the prediction system? (4) Does users' compatibility with the prediction system have implications for the design of such interfaces?

Section 2 gives a summary of related work on gaze-based predictive interfaces. Section 3 provides details on our usability study, proposed adaptive visualization approaches, and proposed gaze-based predictive user interfaces. Section 4 describes the evaluation of our predictive user interfaces in terms of performance, usability, and perceived task load. Section 5 concludes with a discussion of our work and a summary of future directions.

2. Related work

Explicit interfaces (e.g. text terminals and graphical user interfaces) rely on direct commands from the user to the computerized system. In contrast, implicit interfaces sense and reason about user actions that are not primarily aimed to interact with a computerized system to automatically trigger appropriate reactions (Schmidt, 2000). In order to reason about user actions with innovative sensors like eye trackers, implicit interfaces model human behavior by extracting useful and usable patterns while users keep their normal habits and ways of interaction. The advantage of implicit interfaces is that the users do not need explicit commands, prior knowledge, or training to interact with the system. Shortcomings of the command-based explicit interaction model are especially highlighted in mobile computing systems where the ability to input commands is limited. In this paper, we show that well-designed intelligent user interfaces can assist the users by implicitly generating commands based on previously learned models of eye gaze behavior. Related work falls under two broad categories: gaze-based virtual task predictors and gaze-contingent user interfaces.

2.1. Gaze-based virtual task predictors

To the best of our knowledge, there is no line of work that uses online feedback from a gaze-based task prediction model to build a user interface that dynamically adapts itself to user's spontaneous task-related intentions and goals. The majority of the related work focuses solely on generating prediction models and evaluating them in terms of prediction accuracy. However, these systems pay little attention to how prediction models would perform in online usage scenarios. In this paper, we address the multi-faceted goal of building a real-time user interface

that dynamically captures and predicts user's task-related intentions and goals based on eye-movement data, and proactively adapts itself according to these predictions.

Among the earliest examples of *gaze-based virtual task predictors* is work by Campbell and Maglio (2001). They use a wide range of eye movement patterns in order to classify reading, skimming, and scanning tasks. This was followed to a great extent by studies concentrating on intention prediction, i.e. predicting whether the user wants to interact with the system or not during natural interaction. For instance, Bader et al. (2009) use a probabilistic model to predict whether the user intends to select a virtual object or not with 80.7% average accuracy. Similarly, Bednarik et al. (2012) use SVMs to predict whether the user intends to issue a command or not with 76% average accuracy. Both prediction tasks are examples of binary classification. To the best of our knowledge, none of these works have carried out formal studies to evaluate the proposed prediction models in online usage scenarios that involve real users interacting with predictive user interfaces driven by these models.

There are only a few studies that take intention prediction one step further and attempt multi-class intention prediction of virtual tasks. The first notable example is by Courtemanche et al. (2011). This work utilizes eye movements discretized in terms of interface-specific areas of interest (AOI) in addition to keystroke and mouse click events created by the user during interaction. They use HMMs to predict which of the three Google Analytics tasks (i.e. evaluating trends in a certain week, evaluating new visits, and evaluating overall traffic) the user is currently performing with 51.3% average accuracy. The second example is by Steichen et al. (2013). Their domain is information visualization with graphs including bar graphs and radar graphs. Similarly, they rely on interface- and graph-specific AOIs for feature extraction, and Logistic Regression to predict which of the five information visualization tasks (retrieve value, filter, compute derived value, find extremum, and sort) the user is currently performing with 63.32% average accuracy. In subsequent studies, the same group of authors propose different user interface adaptations for graphs (e.g. highlighting, drawing reference lines, and recommending alternative visualizations) (Carenini et al., 2014), and study the effects of these adaptations on a user's performance, both in general and in relation to different visualization tasks and individual user differences (Conati et al., 2014). However, as the authors also mention in a recent publication (Steichen et al., 2014), they have still not published a fully integrated adaptive information visualization system that is able to dynamically provide adaptive interventions that are informed by real-time user behavior data.

2.2. Gaze-contingent user interfaces

A closely-related research area focuses on *gaze-contingent user interfaces* (Duchowski et al., 2004). Gaze-contingent user interfaces utilize gaze data for adapting the user interface contents as we do. However, they rely simply on the instantaneous location of a user's focus of attention. Besides, they do not contribute probabilistic prediction systems or sophisticated gaze-based feature extraction mechanisms to the literature. Nevertheless, for completeness sake, our literature review covers works in this area as well.

Although very few publications address the issue of building gaze-based predictive user interfaces, gaze-contingent user interfaces have attracted much attention from research teams in the last decade. Gaze-contingent user interfaces alter the on-screen view presented to the user based on the focus of a user's visual attention. These interfaces are utilized for improving usability in information visualization applications and promoting engagement and learning in e-tutoring applications, etc. Despite manifesting the large potential benefits of gaze-contingent user interfaces in numerous application areas, all existing works have the following shortcomings in common: (1) They are rule-based, i.e. they tie specific actions to specific regions on the screen and trigger the user interface for an adaptation only based on the duration of eye gaze on these specific regions. (2) In these systems, there is no probabilistic prediction algorithm that directs the adaptive behavior of the user interface. Accordingly, there is no effort to tackle challenges associated with uncertainty or prediction errors. (3) There is no systematic analysis investigating whether and how these user interface adaptations affect user's natural gaze behavior. (4) Lastly, there are very few formal studies to assess the usability and perceived task load associated with these user interfaces.

One of the first examples of gaze-contingent user interfaces is proposed by Starker and Bolt (1990). Their system uses dwell time to determine which part of a graphical interface a user is interested in, and then provides more information about this area via visual zooming and synthesized speech. Streit et al. (2009) and Okoe et al. (2014) have notable contributions that use gaze data for adapting the contents of information visualization systems. Streit et al. (2009) use gaze data to enlarge visualization or maximize clarity of focused regions in 2D scenes, and to navigate 3D scenes. Okoe et al. (2014) use gaze data to improve a user's speed and accuracy in determining whether two nodes are connected in a graph by dimming out or highlighting edges according to user's view focus, and manipulating saliency of sub-graphs around nodes viewed often. Several publications have appeared in recent years documenting the use of gaze-contingent user interfaces in intelligent tutoring systems. Sibert et al. (2000) use dwell time to detect difficulties in identifying words during reading tasks and assist users by providing visual (via highlighting) and auditory cues. Wang et al. (2006) and D'Mello et al. (2012) use gaze data to alleviate disengagement during learning by providing visual and auditory feedback to "unattentive" students looking away from the screen.

To the best of our knowledge, among the existing works that aim to build gaze-contingent user interfaces, there is no work that addresses the problem of adapting the user interface contents in line with user's task-related intentions and goals inferred via probabilistic models of user behavior.

3. Usability study

Consider the tasks described in Fig. 3. We have a gaze-based virtual task prediction system that can accurately distinguish between these tasks. In this paper, we propose to use online feedback from this system to build a user interface that dynamically adapts itself to user's spontaneous task-related intentions and goals. This gives rise to the following research questions: (1) How should a user interface adapt its behavior according to real-time predictions made by the underlying prediction system? (2) Will adaptations affect user behavior and inhibit performance of the prediction system (that assumes natural human behavior)? (3) Will prediction errors affect user behavior and inhibit performance of the prediction system? (4) Does users' compatibility with the prediction system have implications for the design of such interfaces?

3.1. Demographics

We conducted our usability study on 19 participants (17 males, 2 females) recruited from undergraduate and graduate students of our university's engineering faculty on a voluntary basis. Our participants were aged 20–26 years old (M = 23.3, SD = 2.0). 10 participants had normal vision, while the remaining 9 had corrected-to-normal vision. 15 participants had dark-colored eyes, while the remaining 4 had fair-colored eyes. On a scale between 1 (none) to 5 (application developer), participants were moderately experienced with tablets (M = 3.7, SD = 0.9), and less so with pen-based tablets (M = 2.4, SD = 1.2) and eye trackers (M = 2.5, SD = 0.8).

3.2. Setup

We used a Tobii X120 stand-alone eye tracker and a tablet to collect synchronized gaze and pen data, respectively. Tobii X120 operates with

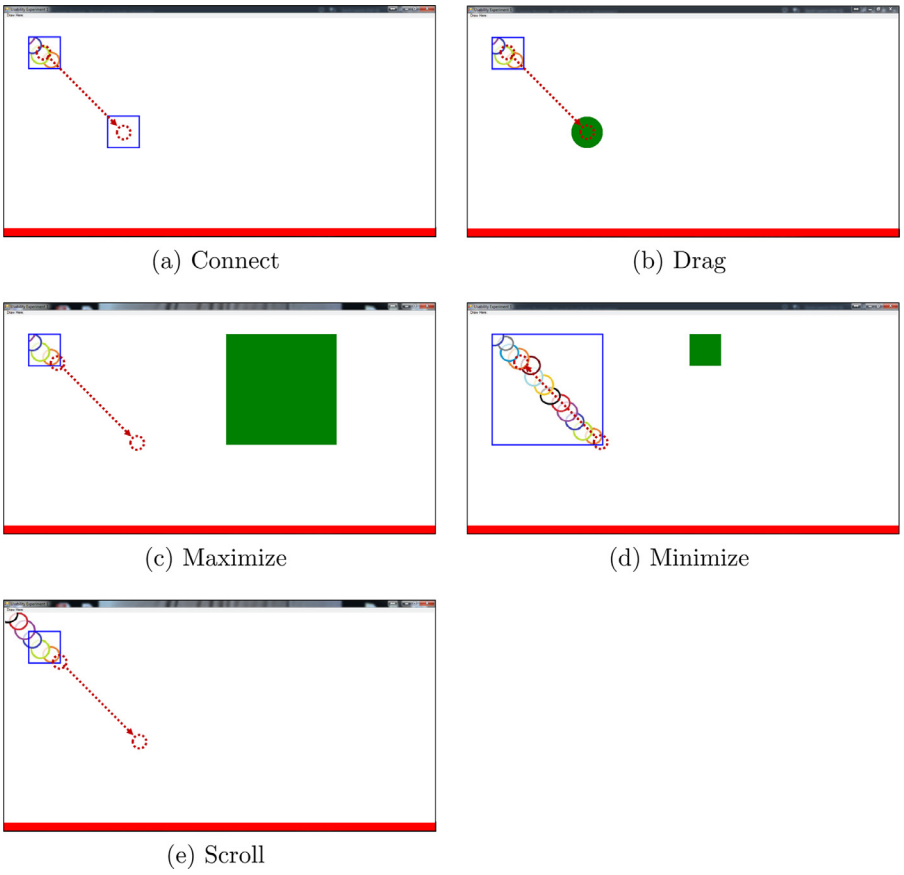


Fig. 3. Pen-based virtual interaction tasks included in our research. Demonstrative examples of how each task can be performed are visualized with dotted visualizations. Starting and ending positions of the exemplary pointer motion is visualized with dotted circles whereas direction of the exemplary pointer motion is visualized with a dotted arrow connecting the starting and ending positions. It is important to note that the dotted visualizations only serve as a reference within this paper, and they are not meant to be visible to the user during the usability study.

a data rate of 120 Hz, tracking accuracy of 0.5°, and drift of less than 0.3°. The tracker allows free head movement inside a virtual box with dimensions $30 \times 22 \times 30$ cm. For displaying our user interfaces accompanied by user's pen position on the tablet, we used a 18.5″ Samsung wide screen LED monitor connected to a PC with Intel Core i5-2500 3.30 GHz CPU and 8GB RAM. Our interfaces were implemented in C++ using the Visual Studio 2013 IDE. Detailed description of the physical setup can be found in our previous paper (Çığ and Sezgin, 2015a).

3.3. User interfaces

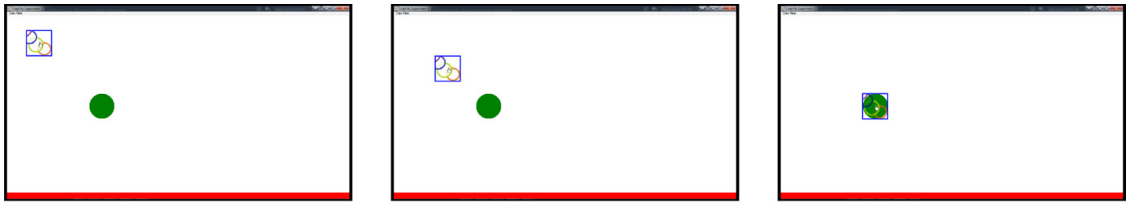
To answer the research questions posed above, we designed and implemented 5 different user interfaces that collectively serve as a generalized, context-free, and non-application-specific test bed. The first two are wizard-based interfaces and will be respectively referred to as *wizard UI*, and *after-the-fact wizard UI*. Wizard-based interfaces assume that there exists a "wizard" which knows and informs the underlying prediction system about the user's intentions, thereby allowing the system to provide the user with correct visual feedback at any moment during interaction. The remaining three are realistic predictive interfaces that eliminate the wizard assumption and will be respectively referred to as *after-the-fact predictive UI, real-time predictive UI*, and *subtle real-time predictive UI*. Our predictive interfaces demonstrate alternative ways of visualizing real-time predictions, and hence each produce an answer to the first question. To answer the second and third questions, we compare the predictive interfaces with the wizard-based interfaces with respect to system performance (measured in terms of prediction accuracy), usability, and perceived task load. To answer the fourth question, we search for a correlation between users' compatibility with the prediction system and measured performance on different predictive interfaces.

3.3.1. Wizard UI

Wizard UI can be thought of as the "gold standard" among our interfaces. It is designed to resemble as closely as possible the WIMP-based user interfaces that users are familiar with. Accordingly, in this wizard interface, the underlying prediction system has no command over the interface and prediction results are not visualized by means of any interface adaptations. Expectedly, the user is unaware of predictions errors. In other words, the loop between the user and the prediction system is open, i.e. the user is fed hardwired and perfect visual feedback via the user interface irrespective of predictions made by the prediction system (Fig. 4). We use the system performance, usability, and perceived task load of this wizard interface as the upper bound and evaluate our proposed predictive interfaces in comparison with this interface. Underlying prediction systems have been trained with multimodal user data previously collected via a nearly identical user interface (that also does not visualize predictions). Therefore, system performance of this interface is expected to surpass others. Usability and perceived task load of this interface is similarly expected to surpass others since it is deliberately designed to resemble traditional WIMP-based user interfaces.

3.3.2. After-the-fact wizard UI

We have a prediction system that can accurately distinguish between intended user actions (i.e. with approximately 90% success rate for 5 actions). Users can greatly benefit from a user interface that reflects user's task-related intentions and goals in real-time. For this purpose, the loop between the user and the prediction system must be closed, i.e. highly



(a) Onset of Action (b) During Action (c) End of Action

Fig. 4. Screen captures of *wizard UI* during a *drag* task. Images serve as illustrations of how our interface looks at the onset, during, and at the end of the user's pen action, respectively. Position of the manipulated object changes in accordance with the user's pen action. Note that the user is fed visual feedback about the current task, and that task only.

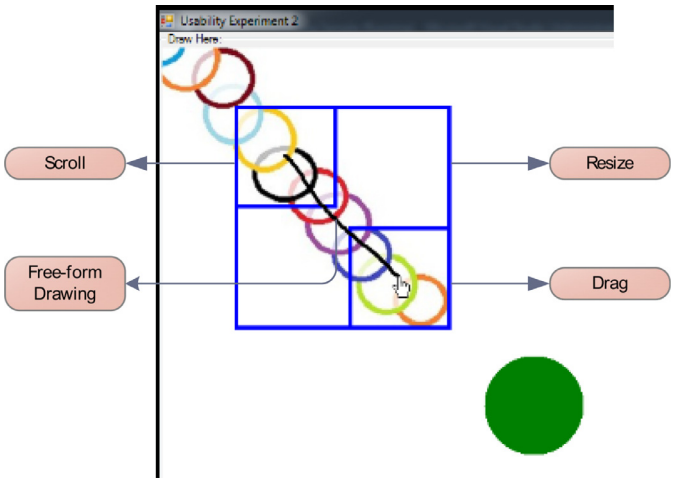
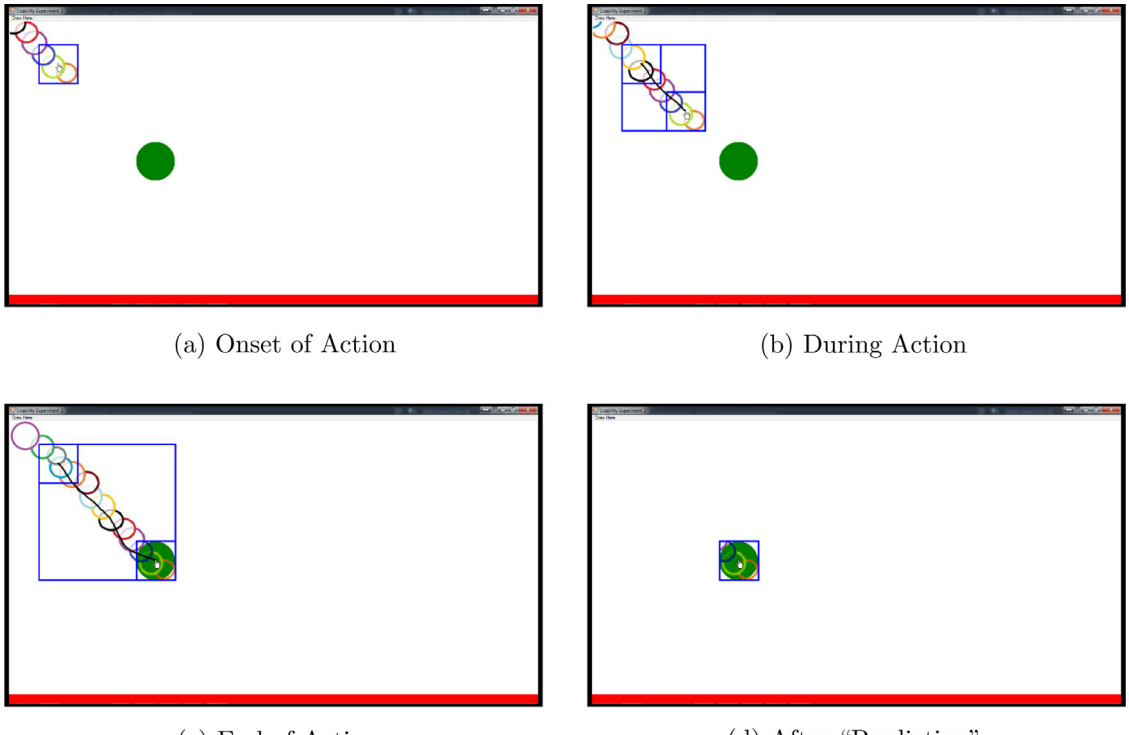


Fig. 5. We introduce a novel visualization paradigm for gaze-based predictive user interfaces where effects of all possible actions are visualized simultaneously for the duration of an action. This paradigm that we will refer to as *simultaneous visualization* can be utilized for providing visual feedback to users in the presence of uncertainty.

accurate but imperfect predictions made by the prediction system must be fed to the user via appropriate visualizations of the user interface. In line with the feedback principle of design (Norman, 1988), the user interface must provide immediate and appropriate visual feedback about the effects of user's actions from the start to the end of an action. However, the prediction system can say its final word on the user's action only once the action is completed. The challenge here is to find a novel way of providing real-time feedback about user's action-related intentions and goals throughout an action while the user's intention is still uncertain. In other words, the challenge is uncertainty visualization.

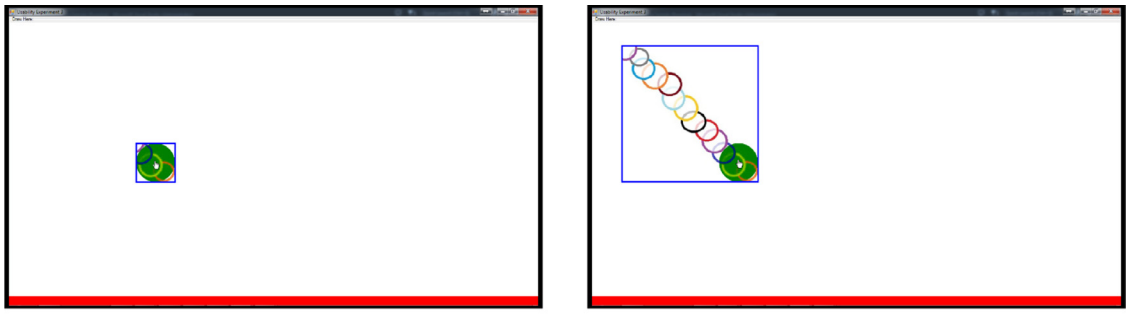
After-the-fact wizard UI is our first step towards tackling the uncertainty visualization challenge. We propose a novel user interface approach where effects of all possible actions are visualized simultaneously for the duration of an action (Fig. 5). When the action is finalized, irrelevant effects fade out and only the effects of the intended action remain visible (Fig. 6). We expect that the user's eyes will focus on the effects of the intended action and irrelevant effects will not affect user behavior thereby inhibit performance of the prediction system (that assumes natural human behavior). This user interface will serve as a means of testing this argument. Note that this interface is also a wizard interface, i.e. once the action is completed, the intended action information



(a) Onset of Action (b) During Action

(c) End of Action (d) After "Prediction"

Fig. 6. Screen captures of *after-the-fact wizard UI* during a *drag* task. Effects of all possible actions are visualized simultaneously from the onset until the end of the action. When the action is finalized, a prediction is made about the user's intended action. Accordingly, irrelevant effects fade out and only the effects of the intended action (i.e. *drag*) remain visible (Fig. 6d). However, there is no prediction really since the intended action information is provided by the wizard.



(a) After a Correct Prediction (b) After an Example Incorrect Prediction

Fig. 7. Screen captures of *after-the-fact predictive UI* during a *drag* task. Screen captures in Fig. 6 also apply to this interface with only one difference. In this case, the intended action information is provided by the underlying prediction system. Hence, when the action is finalized, the user may see effects of an unrelated action due to possible prediction errors. For example, Fig. 7b shows what the UI looks like if user's intended action is incorrectly predicted as a *maximize* task instead of a *drag* task.

is provided by the wizard instead of some underlying prediction system. Accordingly, this user interface is also free from prediction errors.

3.3.3. After-the-fact predictive UI

After-the-fact predictive UI can be regarded as a realistic version of *after-the-fact wizard UI*, where the wizard assumption is eliminated and the intended action information is provided by the underlying prediction system instead of the wizard. Accordingly, when the user completes an action, s/he may see effects of an unrelated action if the underlying system produces a prediction error (Fig. 7). This interface that we propose for visualizing prediction results can be employed in an online usage scenario, hence system performance, usability, and perceived task load of this interface is of great interest to our usability study.

3.3.4. Real-time predictive UI

Showing the effects of irrelevant actions for the entire duration of an action can lead to a heavily cluttered interface as the number of possible actions increases. We offer to use transparency as a solution. More specifically, we envision a user interface where increasing levels of transparency indicates decreasing likelihoods of an action being the intended action. When an action starts, it becomes possible to produce progressively more accurate prediction results in real-time from the start to the end of an action. Since our prediction system is of probabilistic nature, it is also possible to acquire the likelihoods of an action being the intended action in real-time. On that account, we propose another novel user interface approach where effects of all possible actions are visualized simultaneously for the duration of an action with dynamically changing levels of transparency (Fig. 8). This allows us to create a less cluttered and more responsive real-time predictive interface that does not wait until the end of an action to make a prediction.

Every 500 ms, the underlying prediction system feeds the user interface with a list of probability values each denoting the likelihood of an action being the intended action. This, in turn triggers the scene to be redrawn according to the updated likelihood values (Fig. 9). We employ the following steps to create a mapping from the likelihood value p to the alpha value α to determine the transparency level of each effect. Likelihood values range from 0 to 1 and alpha values range from 0 to 255 (0 indicating full transparency and 255 indicating full opacity). If we directly map the likelihood values to alpha values, the effect of an action might fully disappear as its likelihood value approaches too close to 0. To make sure that effects of all actions are visible at all times, we increment the likelihood value of each effect by a base likelihood value of 0.25. Note that for all actions the initial value of p^* is set to 0.25. Then we map the likelihood values to acquire alpha values in the range [64 255] using the following formulas:

$$p^* = 0.75 * p + 0.25 \quad (1)$$

$$\alpha = \lceil p^* * 255 \rceil \quad (2)$$

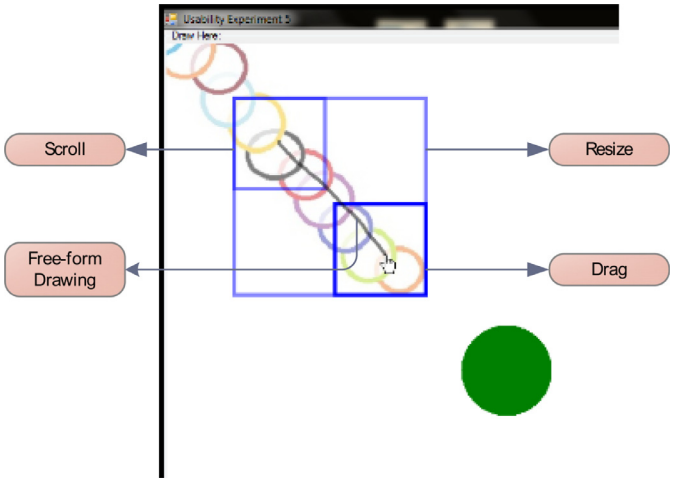


Fig. 8. We introduce another novel visualization paradigm that we will refer to as *adaptive transparency*. It can similarly be utilized for uncertainty visualization in gaze-based predictive user interfaces. In this paradigm, the user interface dynamically adapts itself according to user's real-time intentions and goals. In this respect, our novel visualization paradigm is similar to as-you-type suggestions (i.e. incremental search or real-time suggestions) used in popular search engines or predictive keyboard applications for mobile devices.

Note that a similar methodology applies to the previously described *after-the-fact predictive UI* where the alpha value is fixed at 255, i.e. all effects are fully opaque at all times.

3.3.5. Subtle real-time predictive UI

Subtle real-time predictive UI can be regarded as a more subtle version of *real-time predictive UI*, where the base likelihood value is twice as large, and hence the range of alpha values starts at a higher level. In this case, the likelihood values are mapped in a similar fashion to acquire alpha values in the range [128 255] using the following formulas:

$$p^* = 0.50 * p + 0.50 \quad (3)$$

$$\alpha = \lceil p^* * 255 \rceil \quad (4)$$

Note that similarly, for all actions the initial value of p^* is set to 0.50. This increase in the base likelihood value results in decreased fluctuation of transparency levels, and hence a more stable interface (Fig. 10).

3.4. Procedure

Each participant was subjected to each of the five user interface conditions, resulting in a repeated measures design. The order of conditions presented to each participant was randomized based on the Latin square method (using a 5 × 5 Latin square). During each condition, participants

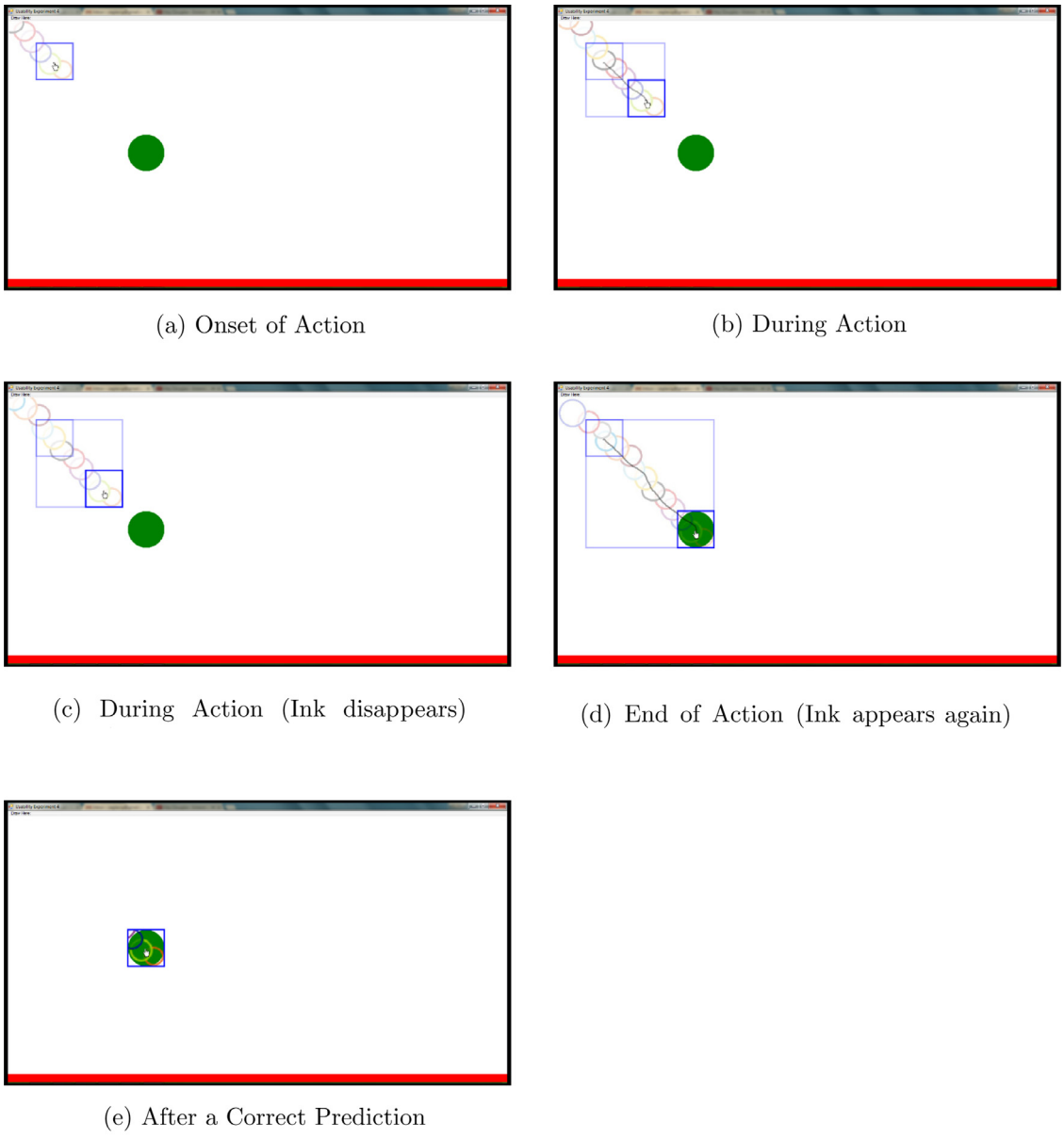


Fig. 9. Screen captures of *real-time predictive UI* during a *drag* task. Effects of all possible actions are visualized simultaneously from the onset until the end of the action. These effects have dynamically changing levels of transparency indicating the likelihood of each action being the intended action at any instant during interaction. It is possible for effects of unlikely actions to disappear as in Fig. 9c based on the instantaneous prediction results. Visibility fluctuation may be found plausible by some users and distracting by others, further analysis in Section 4 will seek an answer to this question among others.

were instructed to complete 5 randomized repeats of 5 tasks (Fig. 3). The order of tasks presented during each condition was randomized as well. It took each participant about 30 min to complete the study. By means of our usability study, we compiled a database of eye gaze, pen, and predicted task label data from 19 participants for 5 randomized repeats of 5 tasks in 5 different user interface conditions. In-between the conditions, participants received 5 practice runs corresponding to each of the 5 tasks in the upcoming user interface condition.

Overall, our usability study consisted of 4 main stages. In the **first stage**, participants were presented with the study guidelines. During this stage, we informed the participants in advance about the various visual effects they might face while performing the tasks (such as visual feedback corresponding to unrelated tasks, or changes in transparency). More specifically, we asked them to concentrate on the given tasks emphasizing the fact that these effects did not determine or affect their success by any means. In addition to this, we requested the participants to keep their eyes on the display device, use a single stroke to complete each task, and maintain an appropriate distance to the eye tracker (which could be monitored and adjusted via the status bar that stayed green as long as the participant was inside the gaze tracking range). In the **second stage**, participants were asked to complete the standard built-in 9-point calibration procedure posed as an “attention test” in order to conceal any hints of eye tracking. **Third stage** was the main data collection stage. Participants received the tasks one by one. At the beginning of each task, prerecorded non-distracting (in terms of avoiding unsolicited gaze behavior) audio instructions were delivered via headphones. Transcripts of the audio instructions given to the participants for each task are listed as follows[1]:

- Connect: Connect the centers of the two squares

[1] Note that the instructions for the drag, maximize, and minimize tasks contain color information which will not show in a B/W copy of Fig. 3. For these tasks, the object to be manipulated (dragged/maximized/minimized) is the one on the left side of each screen.

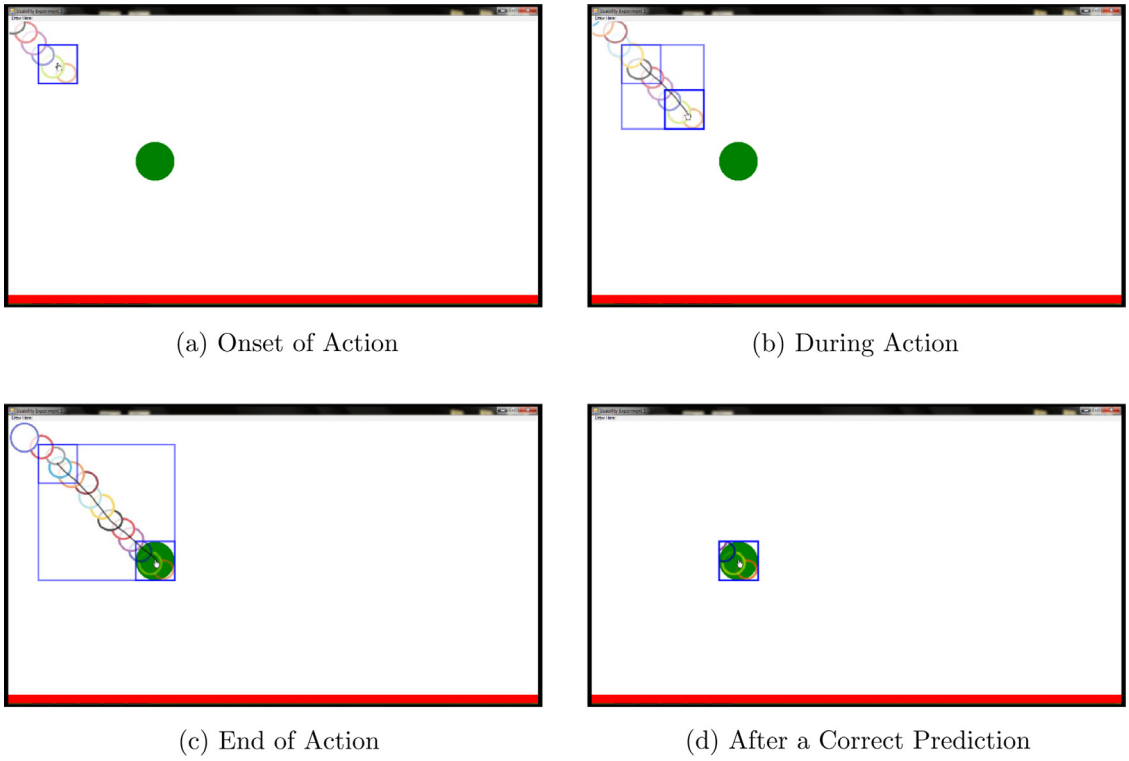


Fig. 10. Screen captures of *subtle real-time predictive UI* during a *drag* task. Similarly, effects of all possible actions are visualized simultaneously with dynamically changing levels of transparency. When compared with the previous interface, effects of all actions are more pronounced at all times and it is not possible for effects of unlikely actions to disappear due to the increase in the base likelihood value.

- Drag: Drag the blue square onto the center of the green circle
- Maximize: Increase the size of the blue square to match the size of the green square
- Minimize: Decrease the size of the blue square to match the size of the green square
- Scroll: Pull the chain until the color of the last link is clearly visible

For each task, participants were asked to manipulate the object in a certain way. The objects could be manipulated by holding and pulling/pushing them in the desired direction using the pen. Desired pen motion started at the center of the object and followed a diagonal line of 10.5 cm. However, the participants were free to manipulate the object as they see fit and decide when the task was complete. We believe this flexibility in task completion criteria is necessary to elicit natural behavior from participants. In order to manipulate the object, participants used the pen-based tablet and the display. A hand-shaped visual cursor was rendered on the display to indicate the position of the user's pen on the tablet. If anything went wrong during a task (e.g. the percentage of gaze data flagged *valid* by the eye tracker was less than 80% or the participant accidentally made redundant/irrelevant pen movements), the current task was repeated. In the **fourth and final stage** of our usability study, a questionnaire was administered to collect qualitative data about the usability and perceived task load associated with our user interfaces as well as demographic data. For the questionnaires, we gathered our user interfaces into three groups: first group consisted solely of *wizard UI*, second group consisted of the after-the-fact interfaces, and third group consisted of the real-time interfaces. Therefore, users were asked to submit three answers instead of five to each of the questionnaire items. This grouping approach is necessary since users cannot differentiate between different flavors of after-the-fact and real-time interfaces without knowing further details about our usability study, perhaps the most important being the presence of underlying prediction systems. For the questionnaire, we compiled a series of Likert-type questions based on the System Usability Scale (SUS) (Brooke, 1996) and the NASA Task Load Index (NASA-TLX) (Hart and Staveland, 1988) assessment tools. SUS gives a high-level subjective view of usability while NASA-TLX rates perceived workload. Both tools allow the researchers to add scores of individual questions to yield a single score on a scale of 0–100. Since some questions (e.g. "How much physical activity was required?") are irrelevant to our usability study, we have excluded them from our questionnaire. As a result, we included the following list of questions in our study:

SUS questions to assess usability (with items on a 5-point likert scale)

- I thought the system was easy to use.
- I found the system unnecessarily complex.
- I would imagine that most people would learn to use this system very quickly.
- I thought there was too much inconsistency in this system.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

(Note that positively- and negatively-worded questions were alternated so that the participants have to read each statement and make an effort to think whether they agree or disagree with it.)

TLX-NASA questions to assess perceived performance, effort, and frustration (with items on a 20-point likert scale)

- How successful were you in accomplishing what you were asked to do?
- How hard did you have to work to accomplish your level of performance?
- How insecure, discouraged, irritated, stressed, and annoyed were you?

3.5. Underlying gaze-based task prediction systems

In the previous sub-sections, we have repeatedly referred to *underlying prediction systems* that provide intended action information to our

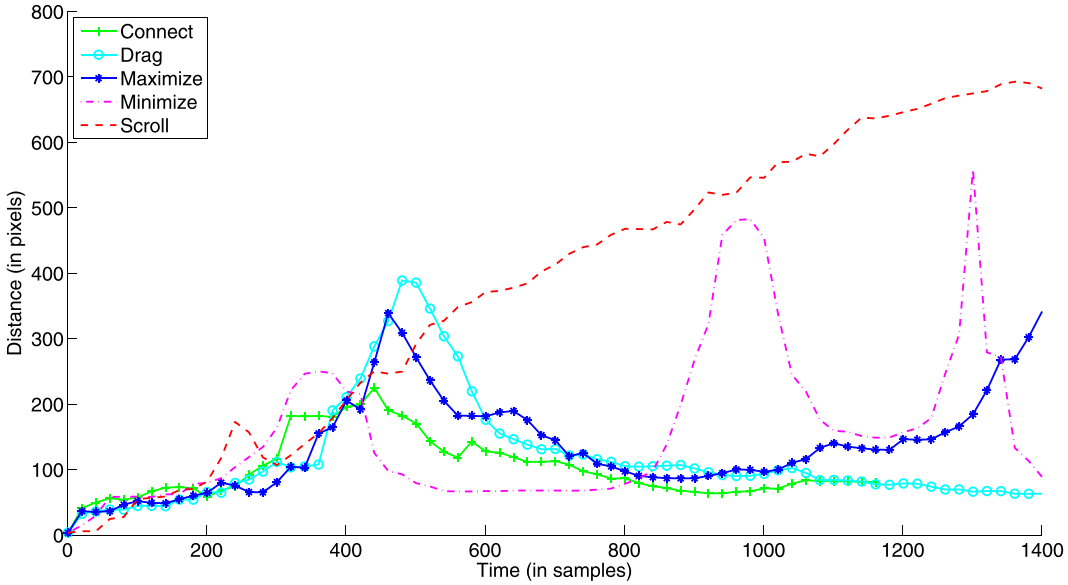


Fig. 11. Characteristic signals obtained from sketch-gaze distance signals of each task.

user interfaces. These systems are in fact statistical prediction models trained with machine learning algorithms on previously collected user data. In total, we have two major task prediction systems: an after-the-fact prediction system, and a real-time prediction system. The former is integrated into our *after-the-fact predictive UI* whereas the latter is integrated into our *real-time predictive UI* and *subtle real-time predictive UI*. In this sub-section, we describe these systems in detail.

3.5.1. After-the-fact task prediction system

Our after-the-fact prediction system builds upon our previously-published work on gaze-based prediction of pen-based virtual interaction tasks (Çığ and Sezgin, 2015a). In our previous paper, we present an after-the-fact task prediction system for the same set of tasks that we include in the current paper. In the current paper, we modify the existing system to the needs of a responsive real-time user interface. More specifically, we decrease the average time it takes for the existing system to determine the type of a newly completed action from 1.125 s to 0.039 s.

Our after-the-fact prediction system waits until the ongoing action is completed to provide intended action information. It outputs a single value denoting the predicted action from the list of possible actions. More specifically, it outputs a single value from the set {1, 2, 3, 4, 5} since we have five tasks in total. To determine the type of a newly completed action, this system extracts three kinds of features from the collected gaze and pen (sketch) data. These features are: (1) evolution of instantaneous sketch-gaze distance over time, (2) spatial distribution of gaze points collected throughout a task, and (3) IDM visual sketch features (Ouyang and Davis, 2009). Detailed description of each feature can be found in our previous paper (Çığ and Sezgin, 2015a).

We focus on optimizing the computational time of the first feature since we have previously demonstrated that it is this feature (more specifically the Dynamic Time Warping (DTW) library it utilizes) that causes the performance bottleneck (Çığ and Sezgin, 2015b). The first feature models the time-wise evolution of the instantaneous distance between pen tip and gaze direction over time using a time-series signal. Initially, one or multiple characteristic signals are computed per task (Fig. 11). When it comes to determining which task a new signal belongs to, similarity of the new signal to each of the characteristic signals is measured. For computing the similarity of two given signals, an open-source MATLAB-based DTW is used (Felty, 2004). To reduce the time requirement of this similarity computation, we have replaced the MATLAB-based library with another library that is written and compiled in the more efficient C programming language (DeBarr, 2006). Numerically, this allows us to process a single action in 0.039 s instead of 1.125, an improvement by a factor of approximately 30 times.

Using the optimized version of our feature extraction mechanism, we train our after-the-fact prediction system following the standard three-step machine learning pipeline. The first step involves extracting feature vectors from a set of data samples. To this end, we extract the features described earlier to obtain three separate feature vectors for each completed action in the database. The first two feature vectors are combined via feature-level fusion and the third feature vector is merged with this combination via classifier-level fusion, both decisions taken based on our previous findings on how information fusion technique effects accuracy values in our context (Çığ and Sezgin, 2015a). Note that for extracting the feature vectors, we use the same set of data samples collected in our previous study (Çığ and Sezgin, 2015a). The second step of the pipeline involves training prediction models using the extracted feature vectors. For this purpose, we train a single Support Vector Machine (SVM) model using the Gaussian radial basis function (RBF) kernel. In this step, we do not partition the input data into disjoint folds for training and testing, and instead use the whole data for training our model since we will use real-time user data during the usability study for testing purposes, which in fact constitutes the third and final step of the pipeline.

3.5.2. Real-time task prediction system

Our real-time prediction system provides on-the-fly intended action information from the start to the end of action. It outputs a list of probability values each denoting the likelihood of an action being the intended action. More specifically, it outputs five likelihood values each in the range [0 1] since we have five tasks in total.

Training of our real-time prediction system is similar to the training of our after-the-fact prediction system except for one major difference. We use our real-time prediction system to create responsive interfaces that dynamically adapt themselves according to user's real-time intentions and goals, and do not wait until the end of an action to make a prediction. This requires a specialized training approach as we have previously proposed in Çığ and Sezgin (2015b). In line with this approach, we train five separate SVM models capturing the characteristics of each task during different time intervals. Accordingly, the first model captures the characteristics of each task in the first 500 ms while the second model captures the characteristics of each task in the first x milliseconds where x is between 500 and 1000, etc. Our real-time prediction system

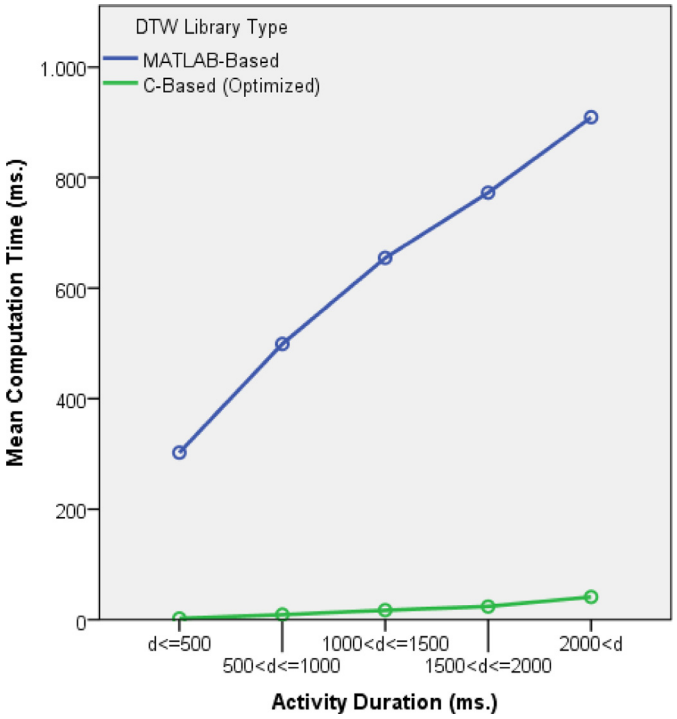


Fig. 12. Mean computation times obtained with each DTW library as a function of time elapsed from the start of a task. Note that with the MATLAB-based DTW library, it is not even possible to update the user interface every 500 ms according to user's real-time intentions and goals since after a point, it takes more than 500 ms for the prediction system to determine the likelihood values for the ongoing action.

in fact consists of these five separate SVM models. Every 500 ms, our real-time prediction system uses the appropriate SVM model to compute and feed the user interface with a list of probability values each denoting the likelihood of an action being the intended action. This, in turn triggers the scene to be redrawn according to the updated likelihood values.

Similar to the after-the-fact prediction system, the real-time prediction system uses SVM models trained using the Gaussian radial basis function (RBF) kernel, and uses the whole data for training the models instead of partitioning the input data into disjoint folds for training and testing. Moreover, our real-time prediction system uses the same kinds of features for feature extraction, and combines separate feature vectors using the same information fusion techniques. Computational time is ever more important since our real-time prediction system is specifically trained to enable responsive interaction. Therefore, for the first kind of feature, the same optimized DTW library is used (Fig. 12).

4. Evaluation

We have proposed five different user interfaces. The first two are wizard-based interfaces. The first interface is the "gold standard" due to its deliberate resemblance to the WIMP-based user interfaces that users are accustomed to. More specifically, in this wizard interface, the underlying prediction system has no command over the interface and prediction results are not visualized by means of any interface adaptations. Expectedly, the user is unaware of predictions errors. Despite their advantages, wizard-based interfaces are not suited to realistic usage scenarios since they assume perfect knowledge about user's action-related intentions and goals. The reality, however, dictates uncertainty about user's intentions and goals unless we have prediction systems that can perform with 100% accuracy. The remaining three interfaces are predictive interfaces. They have each been designed with the goal of building an adaptive user interface that visualizes user's intentions and goals in the presence of uncertainty. In this section, we evaluate the predictive interfaces relative to the wizard interfaces, taking the performance (measured in terms of prediction accuracy), usability, and perceived task load of the first wizard interface as the upper bound. Hence, we both formally test our underlying prediction systems in reasonable scenarios that eliminate the wizard assumption, and propose multiple solutions to the uncertainty visualization challenge faced while designing predictive user interfaces.

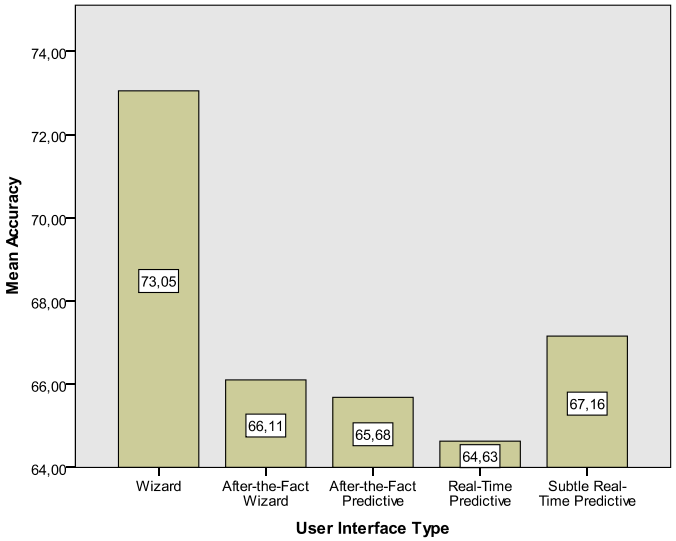


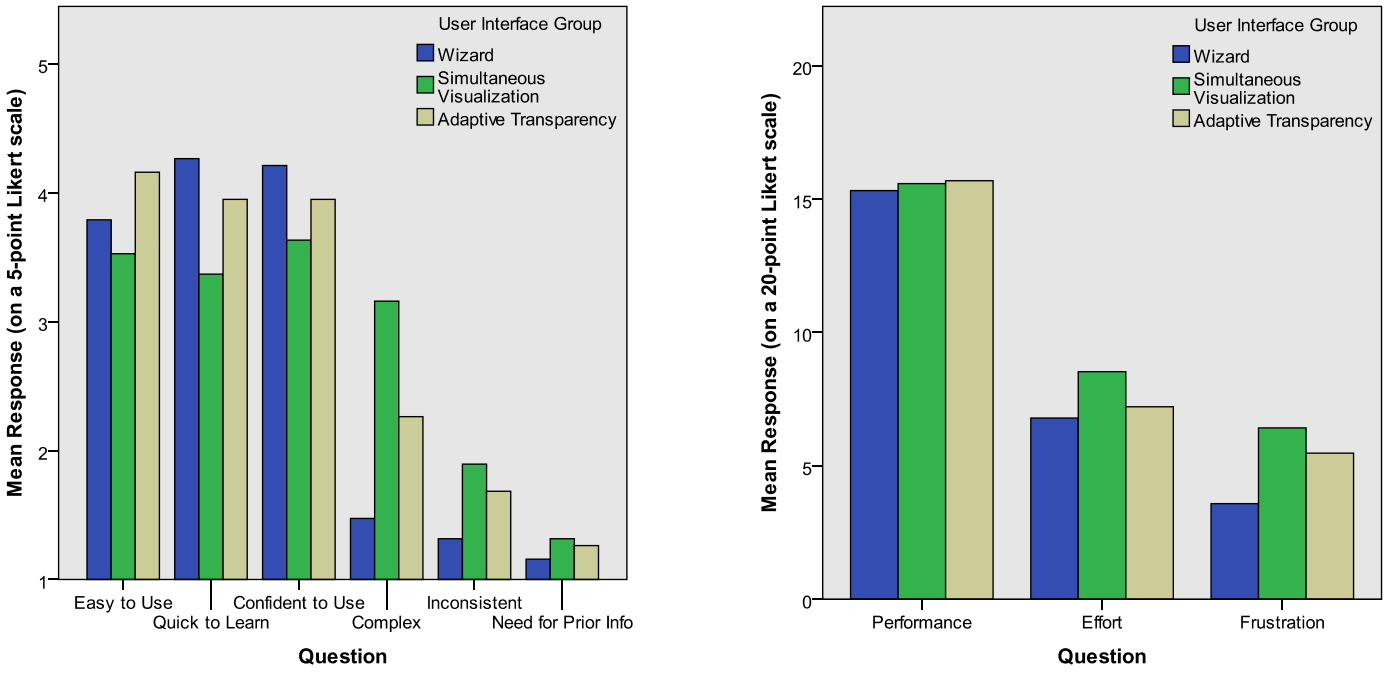
Fig. 13. Marginal mean accuracy score for each user interface averaged over all users.

We present our evaluation results under four main titles. In Section 4.1, we compare our interfaces quantitatively and qualitatively without taking subjective differences into consideration, i.e. by inspecting significant differences between mean scores of each user interface averaged over all users. Then in Section 4.2, we demonstrate that subjective differences are too prominent and significant to be overlooked in the context of our usability study. Therefore in Section 4.3, we perform quantitative and qualitative analysis using a repeated measures design. Taking the subject-based analysis one step further, we offer a statistical method to predict which predictive user interface will be more suitable for each user in terms of system performance. This personalized approach boosts system performance and provides users with the more optimal interface.

4.1. Subject-independent results

4.1.1. Quantitative (accuracy)

We have *intended* and *predicted* task label data collected from 19 participants for 5 randomized repeats of 5 tasks in 5 different user interface conditions. For each user interface, we compute the marginal mean of accuracy by taking the percentage of correctly predicted tasks over all 475 tasks (Fig. 13). *Wizard UI* has the highest accuracy among the others. As we have previously mentioned, superior performance of *wizard UI* is expected due to the fact that the underlying prediction systems have been trained with multimodal user data previously collected via a nearly identical user interface (that also does not visualize predictions). More specifically, neither *wizard UI* nor our previously-published user interface involve simultaneous effect visualizations, adaptive changes in transparency, and erroneous predictions. Despite the similarity of these interfaces, accuracy of *wizard UI* is 73% whereas accuracy of our previously-published interface was reported as 88% (Çığ and Sezgin, 2015a). We believe this difference is caused by the fact that *wizard UI* was tested on a different group of participants than the one which provided the multimodal data for training and testing our previously-published interface. This performance degradation can conceivably be avoided by training the underlying prediction systems using only the



(a) Usability (b) Perceived Task Load

Fig. 14. Marginal mean qualitative results for each user interface measured in terms of usability and perceived task load, and averaged over all users.

current user's data or data collected from users who exhibit similar hand-eye coordination behaviors to the current user's.

Following *wizard UI, subtle real-time predictive UI* has the second highest accuracy, surpassing even *after-the-fact wizard UI* that is free of prediction errors. This indicates that *subtle real-time predictive UI* is the best candidate for solving the uncertainty visualization challenge while minimizing accuracy degradation.

4.1.2. Qualitative (usability and perceived task load)

Overall, usability of the real-time interfaces was rated higher than usability of the after-the-fact interfaces. More specifically, users found the real-time predictive interfaces easier to use, quicker to learn, and they felt more confident using them. In addition, users found the real-time predictive interfaces simpler, more consistent, and they needed less prior information before using them. Likewise, perceived task load of the real-time interfaces was rated lower than the after-the-fact interfaces, i.e. users perceived themselves as more successful in completing the tasks while spending less effort and feeling less frustrated with the real-time predictive interfaces compared to the after-the-fact interfaces.

These results (also summarized in Fig. 14) demonstrate that despite the complex mechanisms involved, usability and perceived task load of the real-time predictive interfaces (grouped under adaptive transparency) was rated superior to that of the after-the-fact interfaces (grouped under simultaneous visualization). This indicates that it is beneficial to decrease the clutter and increase the responsiveness of the interfaces by dynamically changing levels of transparency.

4.2. A personalized approach to uncertainty visualization

Performance of a user during interaction with a novel predictive user interface is conceivably linked to the user's *compatibility* with the interface. We use the term *compatibility* to refer to how well the interface collects, reasons about, and visualizes the user's intentions and goals. Highly *compatible* users which receive relatively more accurate feedback about their intentions and goals are more likely to perform better with

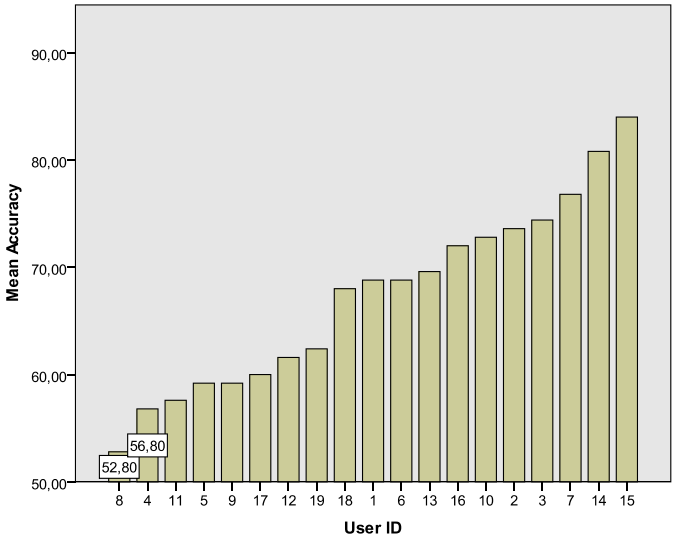


Fig. 15. Mean accuracy score for each user averaged over all user interfaces. Note that a boxplot analysis of the corresponding data marks the two users with the lowest accuracy scores as mild outliers. However, we have not eliminated their data from future analysis since they are not marked as extreme outliers, and similar users are likely to use our interfaces.

and have a high opinion about a novel predictive user interface. In addition to our main research questions, we also aim to find answers to these reasonable claims on personalized differences in *compatibility* with our predictive user interfaces.

Detailed inspection of the accuracy scores reveals high levels of variability among users. Variability is primarily due to subjective differences in *compatibility* with our gaze-based task prediction systems (Fig. 15). The majority of users produce information-rich hand-eye coordination behaviors that enable our gaze-based task prediction systems to achieve

high accuracy scores irrespective of user interface type. On the other hand, a number of users do not lend themselves well to our gaze-based task prediction systems. Variability is also secondarily due to subjective differences in user interface inclinations/preferences. For instance, we observe that some users are not as affected by prediction errors, others perform better in real-time predictive interfaces compared to after-the-fact predictive interfaces, etc. There is no single common pattern among users summarizing the relationship between user interface type and mean accuracy score. Based on these observations, we take variability among users into consideration when comparing the accuracies of different user interfaces in the following sub-sections. To this end, we adopt a repeated measures design that provides a way of accounting for variability, thus decreasing non-systematic variance and increasing sensitivity and power of comparisons between different user interfaces. Furthermore, we utilize variability to our advantage by proposing a personalized approach to uncertainty visualization instead of a unified one. This personalized approach fundamentally involves offering each particular user with the user interface that s/he performs better with and prefers more.

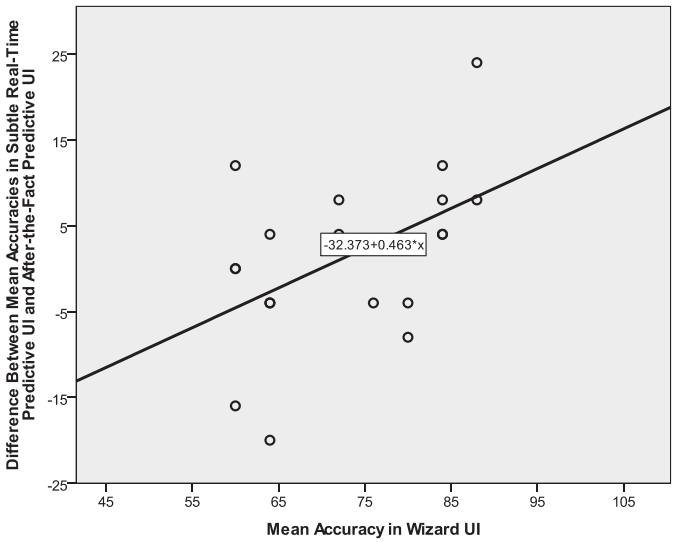


Fig. 16. Users with high accuracy values in *wizard UI* also have favorable accuracy values in *subtle real-time predictive UI*.

4.3. Repeated measures design

4.3.1. Quantitative (accuracy)

Our research primarily seeks answers to the questions of whether user interface adaptations or prediction errors affect user behavior thereby inhibit performance of the underlying prediction systems (that assume natural human behavior). To find answers to these questions, we conducted a repeated measures ANOVA that compares the effect of user interface type on mean accuracy scores. Mauchly's Test of Sphericity indicated that the assumption of sphericity had not been violated ($\chi^2(9) = 11.918,\ p = 0.220$), and therefore no corrections were used. There was a significant effect of user interface type on mean accuracy scores, ($F(4, 72) = 3.287,\ p = 0.016$). Post-hoc tests using the Bonferroni correction revealed that user interface adaptations elicited a slight degradation in accuracy scores for *after-the-fact predictive UI* (65.68 ± 1.99) and *subtle real-time predictive UI* (67.16 ± 3.05) conditions compared to *wizard UI* condition (73.05 ± 2.44). However, neither reduction was found statistically significant ($p = 0.15$ and $p = 0.43$, respectively), indicating the suitability of these two predictive interfaces for solving the uncertainty challenge. The reduction was minimal in *subtle real-time predictive UI* condition, further emphasizing the superiority of this user interface. On the other hand, *real-time predictive UI* condition (64.63 ± 2.58) elicited a significant degradation ($p = 0.043$) in accuracy scores compared to *wizard UI* condition, ruling out the candidacy of this interface for solving the uncertainty challenge. Furthermore, there was no significant effect of absence/presence of prediction errors on accuracy scores ($p = 1.00$) across *after-the-fact wizard UI* (66.11 ± 2.73) and *after-the-fact predictive UI* conditions (two conditions that differ only in the absence/presence of an underlying prediction system, and hence of prediction errors). On the basis of these findings, we can conclude that *after-the-fact predictive UI* and *subtle real-time predictive UI* can be used for uncertainty visualization in gaze-based predictive interfaces without significantly affecting user behavior and inhibiting performance of the underlying prediction systems.

4.3.2. Qualitative (usability and perceived task load)

We have demonstrated in Section 4.1.2 that when subjective differences are not taken into consideration, usability and perceived task load of the real-time interfaces are rated superior to usability and perceived task load of the after-the-fact interfaces. In this sub-section, we show that repeating the qualitative analysis using a repeated measures design, and hence taking subjective differences into consideration leads us to the same conclusion. To make a concise statement, instead of analyzing responses to individual questions on usability, we compute a single score summarizing all aspects of usability by subtracting the sum of responses to negatively-worded questions from the sum of responses to positively-worded questions.[2]

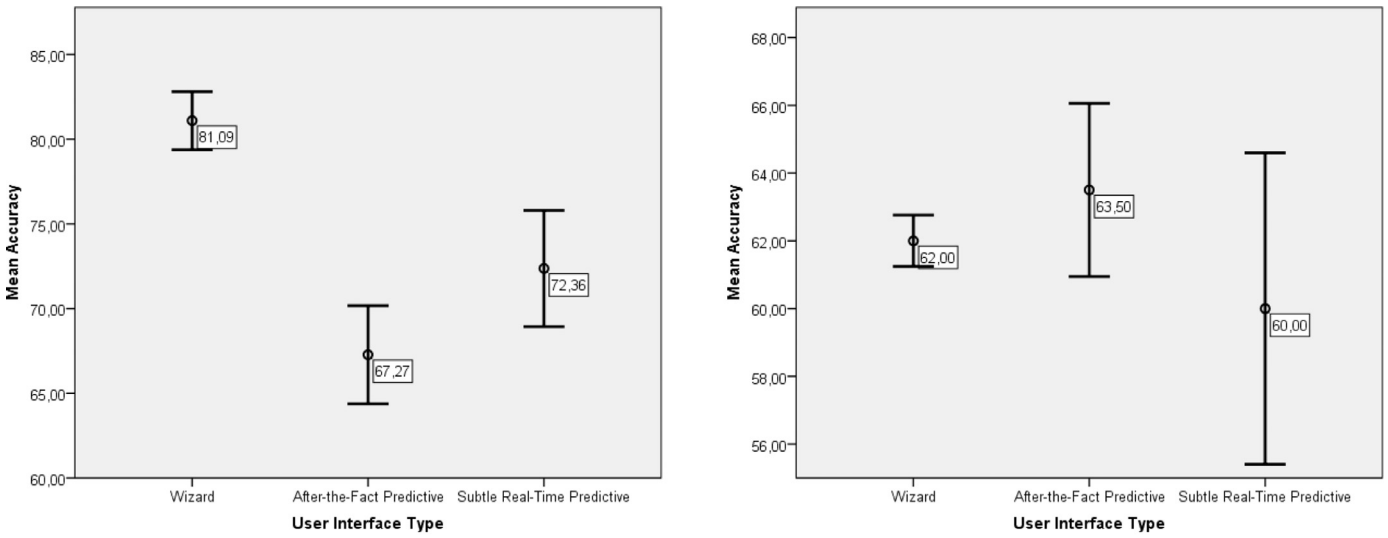
We conducted a repeated measures ANOVA to compare the effect of visualization paradigm on usability. Mauchly's Test of Sphericity indicated that the assumption of sphericity had not been violated ($\chi^2(2) = 2.830,\ p = 0.243$), and therefore no corrections were used. There was a significant effect of visualization paradigm on usability, ($F(2, 36) = 6.545,\ p = 0.004$). Post-hoc tests using the Bonferroni correction revealed that usability of *simultaneous visualization* paradigm condition (4.16 ± 4.10) is statistically lower than usability of both "gold standard" (8.32 ± 2.81) and *adaptive transparency* paradigm (6.84 ± 3.69) conditions ($p = 0.016$ and $p = 0.027$, respectively). On the other hand, no significant difference was found between usability of "gold standard" and *adaptive transparency* paradigm conditions ($p = 0.746$). We also conducted a repeated measures ANOVA to compare the effect of visualization paradigm on perceived task performance, however no significant effects were found.

4.3.3. Correlation analysis and detection of user groups based on quantitative evidence

Following the quantitative and qualitative comparative analysis of our user interfaces in a repeated measures design, we created a mapping based on correlation analysis to predict a user's *compatibility* with our gaze-based task prediction systems based on his/her performance in *wizard UI*. *Compatible* users are assigned to *subtle real-time predictive UI* whereas *incompatible* users are assigned to *after-the-fact predictive UI*. This personalized mapping and subsequent user interface assignment approach enables us to offer each particular user with the user interface that s/he performs better with and prefers more. In this manner, we achieve a mean accuracy score that surpasses the individual mean accuracy scores of both user interface types.

We ran a Pearson product-moment correlation to determine the relationship between a user's mean accuracy score in *wizard UI* and difference between his/her mean accuracy scores in *subtle real-time predictive UI* and *after-the-fact predictive UI*. There was a statistically significant positive correlation ($r = 0.485,\ n = 19,\ p = 0.035$). The corresponding

[2] Positively-worded questions are concerned with ease of use, learnability, and confidence whereas negatively-worded questions are concerned with complexity, inconsistency, and need for prior information. Note that the tools we use for usability and perceived task load assessment (Brooke, 1996; Hart and Staveland, 1988) allow the researchers to add scores of individual questions to yield a single score.



(a) *Compatible* Users (b) *Incompatible* Users

Fig. 17. Personalization boosts system performance. Note that among our participants, 11 were predicted as *compatible* users and the remaining 8 were predicted as *incompatible* users. Error bars indicate ±1 standard error.

linear regression equation (Fig. 16) was estimated as follows:

$$\text{Difference} = -32.373 + 0.463 \times \text{Accuracy in Wizard UI} \quad (5)$$

Using this equation and a given user's mean accuracy value in *wizard UI*, we can predict whether the user will perform better in *subtle real-time predictive UI* or *after-the-fact predictive UI*. Since the correlation is positive, users with high accuracy values in *wizard UI* also have favorable accuracy values in *subtle real-time predictive UI*. We refer to users with high accuracy values in *wizard UI* (Difference ≥ 0) *compatible* users and offer them *subtle real-time predictive UI*. On the other hand, we refer to users with relatively lower accuracy values in *wizard UI* (Difference < 0) *incompatible* users and offer them *after-the-fact predictive UI*. This personalized approach yields mean accuracy scores of 72.36% and 63.5% for *compatible* and *incompatible* users, respectively (Fig. 17). Averaged over all users, mean accuracy score raises up to 68.63%, surpassing the individual mean accuracy scores of all our predictive user interfaces. Note that the reported mean accuracy scores correspond to the leave-one-out cross-validation accuracy scores.

4.3.4. Qualitative reasoning and statistical analysis behind user groups

We have created an intelligent system that can predict which user interface a particular user will perform better with based on his/her *compatibility* with our *wizard UI*. More specifically, we offer *subtle real-time predictive UI* to *compatible* users and *after-the-fact predictive UI* to *incompatible users*. In this sub-section, we show that in addition to boosting system performance, personalization provides users with the more optimal visualization approach (measured in terms of usability and perceived task load).

Overall, *compatible* users did not prefer the after-the-fact interfaces as much as *incompatible* users. They found these interfaces less easy to use (3.18 vs. 4.00), more complex (3.45 vs. 2.75), and they felt less confident using them (3.36 vs. 4.00). Moreover, they perceived themselves as less successful in completing the tasks (15.00 vs. 16.38) while spending more effort (9.82 vs. 6.75) and feeling more frustrated with these interfaces (7.27 vs. 5.25). We further ran a Pearson product-moment correlation to determine the relationship between a user's rating of usability[3] for the real-time predictive interfaces only and difference between his/her mean accuracy scores in *subtle real-time predictive UI* and *after-the-fact predictive UI*. Note that the latter factor determines the user group of a particular user. There was a statistically significant positive correlation ($r = 0.576,\ n = 19,\ p = 0.01$). This further emphasizes the inclination of *compatible* users towards the real-time interfaces.

5. Future work and concluding remarks

We have presented the first line of work that uses online feedback from a gaze-based task prediction model to build a user interface that dynamically adapts itself to user's spontaneous task-related intentions and goals. Since it is not yet possible to train prediction models that can perform with 100% accuracy, we have proposed novel approaches to providing visual feedback in the presence of uncertainty. From another point of view, we have closed the loop between the user and the prediction system by feeding highly accurate but imperfect predictions made by the prediction system to the user via appropriate visualizations of the user interface. Our novel approaches for visualizing uncertainty, namely *simultaneous visualization* and *adaptive transparency*, have been realized via wizard-based user interfaces and different flavors of predictive user interfaces. To assess the performance, usability, and perceived task load of our interfaces, we have conducted a thorough usability study with 19 participants and 5 frequently employed virtual interaction tasks. Among these interfaces, *after-the-fact predictive UI* and *subtle real-time predictive UI* stand out as the best candidates for solving the uncertainty visualization challenge. Both interfaces are able to visualize user's task-related intentions and goals in the presence of uncertainty, and without significantly affecting user behavior and inhibiting performance of the underlying prediction systems. Moreover, the latter has comparable usability and perceived task load to WIMP-based user interfaces. Furthermore, we have offered a method to predict which predictive user interface will be more suitable for each user in terms of system performance. Personalization boosts system performance and provides users with the more optimal visualization approach.

Building complex real-world user interfaces utilizing our prediction models and exploring their usability characteristics is an essential follow-up to what we presented. We believe that various existing software tools can possibly be improved if we have a way of correctly guessing the user's intentions during interaction. Practical

[3] To make a concise statement, instead of analyzing responses to individual questions on usability, we compute a single score summarizing all aspects of usability by subtracting the sum of responses to negatively-worded questions from the sum of responses to positively-worded questions.

application scenarios may involve professional diagramming software, electronic circuit design software, digital photography organizing tools, and mind mapping tools. In all these scenarios, the interface consists of objects (i.e. flowchart shapes, circuit elements, photos, rectangles representing concepts, etc.) that need to be manipulated (i.e. dragged, resized, connected, etc.) multiple times. In the existing interfaces, the user has to explicitly switch the mode of operation via unnatural and imposed mode switching mechanisms and interaction rules. Well-known examples to such mechanisms are locating the four-headed arrow to drag an object and locating the double-headed arrow that can only be found at the corners and edges to resize an object. We believe we can use our prediction system to create interfaces where the user does not have to make a specific gesture or locate the correct button to repeatedly switch the interaction mode in-between different manipulation tasks. These novel predictive interfaces will especially profit mobile devices where screen size limitations and absence of a physical mouse make high precision pointing impossible (also known as the fat finger problem).

Until we can build prediction models that can perform with 100% accuracy, we need to find a way to handle prediction errors. Although we have demonstrated that there is no significant effect of absence/presence of prediction errors on accuracy scores in our context, it is possible that users might confuse system errors with user-induced errors and diverge from natural gaze behavior in an effort to avoid them. In turn, this divergence will conceivably reduce the quality of the user's experience with the interface as well as the accuracy of our prediction systems that assume natural user behavior. In consequence, several questions remain to be addressed with respect to detecting and recovering from prediction errors: What will be the degree of initiative on the system's sides – "will the system act, offer to act, ask if it should act, or merely indicate that it can act?" (Ju et al., 2008) How can we detect prediction errors? Will it be possible for users to correct prediction errors by overriding? How can we design transitions between implicit and explicit interaction, so that users can interrupt or stop a proactive system action? How can we establish shared understanding between the user and the system without interrupting the interaction flow? Formal user studies will be needed to obtain definitive answers to such questions.

On the basis of the promising findings presented in this paper, work on the remaining issues is continuing, and will be presented in future papers. One remaining issue concerns mismatch between training and testing conditions of our gaze-based task prediction models. The mismatch is firstly due to the fact that our models were evaluated using a different group of participants than the one which provided the multimodal data for training them. In our future research we intend to concentrate on training the underlying prediction models using only the current user's data or data collected from users who exhibit similar hand-eye coordination behaviors to the current user's. The mismatch is secondly due to the fact that our models were trained with offline interaction data that do not involve user interface adaptations or prediction errors. Nevertheless, our models were tested in an online setting. Therefore, further research is required to investigate the performance of new prediction models trained using multimodal data collected during the usability study presented in this paper. Finally, note that we have acquired quite promising results despite the presence of mismatches, and we believe that alleviating the mismatch problem will further boost the performance of our prediction systems. Another issue concerns *compatibility* prediction. We predict a user's *compatibility* with our gaze-based task prediction systems based on his/her performance in *wizard UI*. *Wizard UI* is designed to resemble as closely as possible the WIMP-based user interfaces that users are familiar with. Further study into predicting a user's *compatibility* based on his/her natural gaze behaviors during interaction with prominent browsers/operating systems (e.g. while the user is freely browsing the web or organizing digital photo albums) would be of interest.

Acknowledgments

The authors gratefully acknowledge the support and funding of TÜBİTAK (The Scientific and Technological Research Council of Turkey) under grant numbers 110E175 and 113E325 and TÜBA (Turkish Academy of Sciences).

References

Bader, T., Vogelgesang, M., Klaus, E., 2009. Multimodal integration of natural gaze behavior for intention recognition during object manipulation. In: Proceedings of the Eleventh International Conference on Multimodal Interfaces. ACM, New York, NY, USA, pp. 199–206.

Bednarik, R., Vrzakova, H., Hradis, M., 2012. What do you want to do next: a novel approach for intent prediction in gaze-based interaction. In: Proceedings of the Symposium on Eye Tracking Research and Applications. ACM, New York, NY, USA, pp. 83–90.

Brooke, J., 1996. Sus - a quick and dirty usability scale. Usability Eval. Ind. 189 (194), 4–7.

Campbell, C.S., Maglio, P.P., 2001. A robust algorithm for reading detection. In: Proceedings of the 2001 Workshop on Perceptive User Interfaces. ACM, New York, NY, USA, pp. 1–7.

Carenini, G., Conati, C., Hoque, E., Steichen, B., Toker, D., Enns, J., 2014. Highlighting interventions and user differences: informing adaptive information visualization support. In: Proceedings of the Thirty-second Annual ACM Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp. 1835–1844.

Çığ, Ç., Sezgin, T.M., 2015a. Gaze-based prediction of pen-based virtual interaction tasks. Int. J. Hum.-Comput. Stud. 73, 91–106.

Çığ, Ç., Sezgin, T.M., 2015b. Real-time activity prediction: a gaze-based approach for early recognition of pen-based interaction tasks. In: Proceedings of the Twelfth Sketch-Based Interfaces and Modeling Symposium. Eurographics Association, Aire-la-Ville, Switzerland, pp. 59–65.

Conati, C., Carenini, G., Hoque, E., Steichen, B., Toker, D., 2014. Evaluating the impact of user characteristics and different layouts on an interactive visualization for decision making. Comput. Graph. Forum 33 (3), 371–380.

Courtemanche, F., Aïmeur, E., Dufresne, A., Najjar, M., Mpondo, F., 2011. Activity recognition using eye-gaze movements and traditional interactions. Interact. Comput. 23 (3), 202–213.

DeBarr, D., 2006. Constrained dynamic time warping distance measure. https://www.mathworks.com/matlabcentral/fileexchange/12319-constrained-dynamic-time-warping-distance-measure/.

D'Mello, S., Olney, A., Williams, C., Hays, P., 2012. Gaze tutor: a gaze-reactive intelligent tutoring system. Int. J. Hum.-Comput. Stud. 70 (5), 377–398.

Duchowski, A.T., Cournia, N., Murphy, H.A., 2004. Gaze-contingent displays: a review. Cyberpsychol. Behav. Social Networking 7 (6), 621–634.

Felty, T., 2004. Dynamic time warping. http://www.mathworks.com/matlabcentral/fileexchange/6516-dynamic-time-warping/.

Hart, S.G., Staveland, L.E., 1988. Development of nasa-tlx (task load index): results of empirical and theoretical research. Adv. Psychol. 52, 139–183.

Ju, W., Lee, B.A., Klemmer, S.R., 2008. Range: exploring implicit interaction through electronic whiteboard design. In: Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work. ACM, New York, NY, USA, pp. 17–26.

Norman, D.A., 1988. The Design of Everyday Things. Basic Book.

Okoe, M., Alam, S.S., Jianu, R., 2014. A gaze-enabled graph visualization to improve graph reading tasks. Comput. Graph. Forum 33 (3), 251–260.

Ouyang, T.Y., Davis, R., 2009. A visual approach to sketched symbol recognition. In: Proceedings of the Twenty-first International Joint Conference on Artifical Intelligence, pp. 1463–1468.

Schmidt, A., 2000. Implicit human computer interaction through context. Pers. Technol. 4 (2–3), 191–199.

Sibert, J.L., Gokturk, M., Lavine, R.A., 2000. The reading assistant: eye gaze triggered auditory prompting for reading remediation. In: Proceedings of the Thirteenth Annual ACM Symposium on User Interface Software and Technology. ACM, San Diego, CA, USA, pp. 101–107.

Starker, I., Bolt, R.A., 1990. A gaze-responsive self-disclosing display. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp. 3–10.

Steichen, B., Carenini, G., Conati, C., 2013. User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In: Proceedings of the Eighteenth International Conference on Intelligent User Interfaces. ACM, New York, NY, USA, pp. 317–328.

Steichen, B., Conati, C., Carenini, G., 2014. Inferring visualization task properties, user performance, and user cognitive abilities from eye gaze data. TiiS 4 (2), 1–29.

Streit, M., Lex, A., Müller, H., Schmalstieg, D., 2009. Gaze-based focus adaption in an information visualization system. In: IADIS International Conference Computer Graphics, Visualization, Computer Vision and Image Processing, pp. 303–307.

Wang, H., Chignell, M., Ishizuka, M., 2006. Empathic tutoring software agents using real–time eye tracking. In: Proceedings of the Symposium on Eye Tracking Research and Applications. ACM, New York, NY, USA, pp. 73–78.