Designing Implicit Interfaces for Physiological Computing: Guidelines and Lessons Learned Using fNIRS

ERIN TREACY SOLOVEY, Drexel University
DANIEL AFERGAN, Tufts University
EVAN M. PECK, Bucknell University
SAMUEL W. HINCKS and ROBERT J. K. JACOB, Tufts University

A growing body of recent work has shown the feasibility of brain and body sensors as input to interactive systems. However, the interaction techniques and design decisions for their effective use are not well defined. We present a conceptual framework for considering implicit input from the brain, along with design principles and patterns we have developed from our work. We also describe a series of controlled, offline studies that lay the foundation for our work with functional near-infrared spectroscopy (fNIRS) neuroimaging, as well as our real-time platform that serves as a testbed for exploring brain-based adaptive interaction techniques. Finally, we present case studies illustrating the principles and patterns for effective use of brain data in human–computer interaction. We focus on signals coming from the brain, but these principles apply broadly to other sensor data and in domains such as aviation, education, medicine, driving, and anything involving multitasking or varying cognitive workload.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces

General Terms: Human Factors

Additional Key Words and Phrases: Brain–computer interfaces, physiological computing, implicit interfaces

ACM Reference Format:
Erin T. Solovey, Daniel Afergan, Evan M. Peck, Samuel W. Hincks, and Robert J. K. Jacob, 2015. Designing implicit interfaces for physiological computing: Guidelines and lessons learned using fNIRS. ACM Trans. Comput.-Hum. Interact. 21, 6, Article 35 (January 2015), 27 pages.
DOI: http://dx.doi.org/10.1145/2687926

1. INTRODUCTION

As computing systems become increasingly complex and intelligent, the role and expectations of the human in the computer system change as well. With recent advancements in computer processing speed, algorithm sophistication, and autonomy capabilities, we would expect the advanced human–computer system to be more efficient and effective. However, these advancements in computing have also led to increased demands on

35

This research was supported by Google Inc. and the NSF (NSF Grant Nos. IIS-1065154 and IIS 1218170 as well as NSF Grant #1136996 awarded to the Computing Research Association for the CI Fellows Project). Any opinions, findings, conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of Google Inc. or the National Science Foundation.
Authors' addresses: E. T. Solovey, College of Computing & Informatics, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104; email: erin.solovey@drexel.edu; D. Afergan, S. W. Hincks, and R. J. K. Jacob, Computer Science Department, Tufts University, 161 College Avenue, Medford, MA 02155; emails: {afergan, shincks, jacob}@cs.tufts.edu; E. M. Peck, Computer Science Department, Bucknell University, 1 Dent Drive, Lewisburg, PA 17837; email: evan.peck@bucknell.edu.

DOI: http://dx.doi.org/10.1145/2687926

people to keep up with vast amounts of data being generated and to perform numerous simultaneous tasks. We now see bottlenecks in systems and errors made due to the limited perceptual and communication capabilities of the human, whose abilities have remained relatively stagnant. Computer systems have a limited ability to detect the full spectrum of information that is naturally and effortlessly generated by the human during computer usage, and thus do not know when to throttle information output.

For example, during natural communication, humans generate accompanying visual and auditory cues that supplement their dialogue and allow the other part to adapt behavior appropriately. At the same time, several physiological changes occur that may or may not be detected by the other person. During human–computer interaction, humans may also generate these additional signals automatically, but there is limited support for the computer to detect and utilize this information. Detecting these signals in real time and incorporating them into the user interface could improve the communication between the computer and the human user with little additional effort required of the user [Lee and Tan 2006; Tan 2006; Cutrell and Tan 2008; Zander et al. 2008]. This communication improvement would lead to technology that is more effective because it supports the user's changing cognitive state. Such improvements are increasingly valuable as technology becomes more powerful and pervasive while our cognitive abilities do not change considerably.

In order to automatically infer the user's changing cognitive state in real time, researchers have explored performance data, interaction history (e.g., keystrokes), and environmental context to assess the user's current state [Fogarty, Hudson and Lai 2004; Hudson et al. 2003; Starner, Schiele and Pentland 1998], while others use computer vision to detect facial expressions or other behavioral measures. Physiological measures are also emerging as continuous indicators of cognitive state changes [Fairclough 2009; Mandryk, Atkins and Inkpen 2006; Nacke et al. 2011]. Brain imaging and brain sensing techniques aim to get close to the source by looking at changes in brain activity during task performance [Grimes et al. 2008; Hirshfield et al. 2009, Zander et al. 2010] and are becoming realistic tools for HCI research.

With brain, body, behavioral, and environmental sensors, it is now possible to capture subtle changes in the user's cognitive state in real time. This opens up new doors in human–computer interaction research. This information can be used as continuous input to interactive systems, making the systems more in sync with the user, providing appropriate help and support when needed. However, brain, body, and other sensor data differ from most existing input modalities (e.g., mouse, keyboard) because they do not directly manipulate a device. To achieve this goal, the interactive system must be designed carefully to take advantage of this more subtle new class of input, leading to *implicit interfaces*.

In this article, we present a conceptual framework for considering implicit input from the brain, along with design principles and patterns we have developed from our work in this area. These have emerged from investigations into the potential of functional near-infrared spectroscopy (fNIRS) brain sensing in interactive systems. To provide context, we begin by describing a series of controlled, offline fNIRS studies that lay the foundation for real-time brain-based systems. These studies illustrate the types of cognitive states that can be measured reliably with fNIRS as well as the best practices and tradeoffs in fNIRS data analysis. We then describe the real-time platform we have built that classifies a user's cognitive state based on brain data in real time and that serves as a testbed for exploring brain-based adaptive interaction techniques. We then present an overview of several examples of brain-based adaptive systems that we have built and studied, which illustrate these principles and patterns, and demonstrate effective use of brain data in human–computer interaction. We focus specifically on signals coming from the brain, but these principles can be applied broadly to other similar sensor data, and this work has applications in many domains such as health care, aviation, education, medicine, driving, and anything involving multitasking or varying cognitive workload.

2. BACKGROUND

2.1. Neuroimaging Background and HCI Considerations

Noninvasive neuroimaging techniques, primarily developed for clinical settings, have been powerful tools for understanding brain structure and function as well as for diagnosing brain injuries or disorders. *Structural imaging* techniques, such as computed tomography (CT), generate brain images of the static structure of the brain, as well as brain tumors and injuries. These provide valuable snapshots of the state of the brain but are not used in brain–computer interfaces, which require measurement of the changing state of the brain due to cognitive activity. In order to obtain this dynamic state information, *functional neuroimaging* can be used to capture changes over time during activities. For example, functional magnetic resonance imaging (fMRI) is widely used to generate three-dimensional images of the brain showing the blood oxygen level–dependent (BOLD) hemodynamic response to stimuli and activities. These hemodynamic changes in blood volume and oxygenation are an indirect measure of brain activity. Similar to fMRI, positron emission tomography (PET) scans provide three-dimensional images of blood flow, blood oxygen, and metabolic function of cells but are mainly used for investigating organs for cancers and other diseases. fNIRS also measures blood oxygen changes and is discussed in detail later. Unlike the hemodynamic neuroimaging tools, electroencephalography (EEG) and magnetoencephalography (MEG) provide a more direct measure of neuronal activity by recording electrical signals on the scalp generated by neurons firing in the brain.

Since the neuroimaging tools described previously were originally intended for use in clinical or laboratory settings, they often place restrictions on the patient or study participant that are not reasonable for realistic human–computer interaction settings [Tan and Nijholt 2010]. Besides being expensive, PET, fMRI, and MEG require subjects to sit or lie down in unnatural positions and remain essentially motionless [Lee and Tan 2006]. In addition, PET requires ingestion of hazardous material and fMRI exposes subjects to loud noises that may interfere with the study [Izzetoglu et al. 2005]. Plus, the powerful magnetic field prevents computer usage in both fMRI and MEG. These factors make it impractical to use these techniques in a realistic interactive situation. Because it is less intrusive, more portable, and less expensive than these other technologies, EEG has seen wide use in BCI research. For example, it has been used to classify tasks [Lee and Tan 2006], measure cognitive load [Grimes et al. 2008], and support human-aided computer vision [Shenoy and Tan 2008] as well as limited communication [Keirn and Aunon 1990; Schalk et al. 2004; Wolpaw et al. 1991]. However, it can have a significant setup time, and electronic devices in the room can interfere with the signal. It has limited spatial resolution but high temporal resolution. In addition, most EEG systems require gel to be applied to the scalp, although devices are being developed that use dry electrodes. Because these disadvantages are not prohibitive, EEG has been the main technology used in brain–computer interface research.

2.2. Functional Near-Infrared Spectroscopy in HCI Settings

Like other neuroimaging techniques, fNIRS was designed primarily for laboratory and clinical settings. However, restrictions such as long setup time, highly restricted position, intolerance to movement, and other limitations that are inherent to other brain sensing and imaging devices are not factors when using fNIRS. By using fNIRS, researchers can have access to the user's cognitive state in realistic HCI laboratory

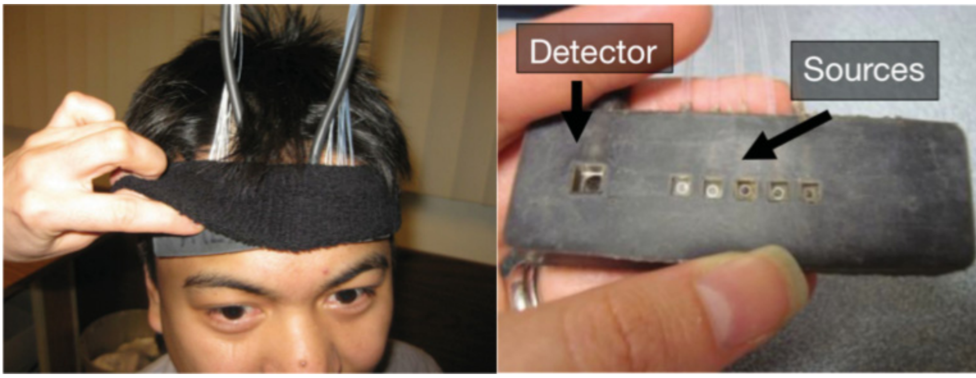

Fig. 1. A functional near-infrared spectroscopy sensor consists of light sources emitting two wavelengths of near-infrared light as well as a light detector (right). Two sensors can be placed under a headband to monitor oxygenation changes in the brain (left).

conditions. This section provides background on how fNIRS works, as well as considerations that are particularly relevant for adopting fNIRS in HCI settings.

2.2.1. Technical Background. fNIRS is a noninvasive and lightweight device that detects changes in oxygenated and deoxygenated blood in a region of the brain by using optical fibers to emit near-infrared light [Chance et al. 1988]. These light sources are arranged on a headband or cap, along with light detectors (Figure 1), making them portable, easy to use, and quick to set up. The light sources send two wavelengths of near-infrared light into the forehead, where it continues through the skin and bone 1 to 3cm deep into the cortex. Biological tissues are relatively transparent to these wavelengths, but the oxygenated and deoxygenated hemoglobin are the main absorbers of this light. After the light scatters in the brain, some reaches the light detector on the surface. By determining the amount of light sensed by the detector, the amount of oxygenated and deoxygenated hemoglobin in the area can be calculated using the modified Beer-Lambert Law [Delpy et al. 1988]. Because these hemodynamic and metabolic changes are associated with neural activity in the brain, fNIRS measurements can be used to detect changes in a person's cognitive state while performing tasks.

2.2.2. Sensor Placement. The basic technology described previously is common to all fNIRS systems, and the measured signal depends on the location of the probe. The most common placements are on the motor cortex [Sitaram et al. 2007] and the prefrontal cortex [Ehlis et al. 2008; Mappus et al. 2009], although other regions have also been explored [Herrmann et al. 2008]. The sensors used in this research were designed for the forehead, one of the most effective placements because of the absence of hair, which absorbs light and degrades the fNIRS signal. Thus, the anterior prefrontal cortex, which lies behind the forehead, is the main target in the studies described here. This area of the brain deals with high-level processing [Ramnani and Owen 2004], such as working memory, planning, problem solving, memory retrieval, and attention.

2.2.3. Hemodynamic Response Latency. An important characteristic of fNIRS data is that the hemodynamic response being measured is a slow response that occurs over 5 to 8 seconds. This is in contrast with EEG, which measures brain activity at the millisecond level. It should be noted that some studies have reported measurement of a fast fNIRS signal [Wolf et al. 2002], which would provide near-instantaneous measurements, but it has not been extensively explored and most studies look at the slow response. Due to this latency, it may be inappropriate to design interfaces that require immediate action from brain activity or to use fNIRS sensor data as direct, explicit input to the system as has been done with EEG. Instead, the continuous recognition of a particular cognitive state with fNIRS is suitable for use as implicit input, which is discussed later.

2.2.4. Signal Artifacts in HCI Settings. To explore the feasibility of fNIRS use in HCI, we previously examined whether typical human behavior (e.g., head and facial movement) or computer interaction (e.g., keyboard and mouse usage) interfere with brain measurement using fNIRS [Solovey et al. 2009]. We found that typing and mouse clicking do not interfere with the fNIRS signal. However, it may be necessary to avoid or control major head movements and frowns. Other artifacts such as minor head movements, heartbeat, and respiration may be corrected using filtering. There are many types of filtering algorithms that can help reduce the amount of noise in data (for a review, see Tak and Ye [2014]). Methods include short separation regression [Gagnon et al. 2012], adaptive finite impulse response (FIR) filtering, Weiner filtering [Devaraj et al. 2004; Izzetoglu et al. 2005], adaptive filtering [Devaraj et al. 2004], and principal component analysis [Huppert and Boas 2005; Matthews et al. 2008; Sitaram et al. 2007]. Matthews et al. [2008] note that FIR can be used in real time if accelerometers are used simultaneously on the head to record head motion. The other methods are mainly offline procedures, making them less practical for real-time systems. The guidelines [Solovey et al. 2009] are followed in the research described in the remaining sections to robustly detect cognitive changes for implicit input in realistic HCI laboratory settings.

2.3. Implicit Input in Human–Computer Interaction

Typically, most human–computer interaction techniques use *explicit input*, where the user consciously manipulates a device (e.g., mouse or keyboard) to indicate a desired command or action in the system. In contrast, *implicit inputs* are user actions or situational contexts that the system understands as input but that were not actively chosen by the user to interact with the system [Schmidt 2000]. For example, mobile applications frequently use location as an implicit input to the application, which then can improve the user experience by putting the interaction in context. Similarly, sensors can be embedded in the environment to recognize movement of objects or people in order to adapt the environment appropriately for the context (e.g., lights turning on as a person enters a room). Such implicit input is fundamental to the fields of ubiquitous computing and context-aware systems but mainly focuses on situational and environmental context and not on cognitive state as context. With lower costs for noninvasive brain and body sensing, we recently have seen a growing interest in employing such sensors in interactive systems for a wide audience, providing implicit contextual input of the user's cognitive state and diverging from the traditional, explicit brain–computer interaction model that has been effective for users with disabilities.

2.4. Cognitive State as Implicit Input

As brain–computer interfaces have traditionally focused on users with disabilities, they often employ brain signals as the primary input [Blankertz et al. 2007; Kennedy et al. 2000; Pfurtscheller, Flotzinger and Kalcher 1993; Schalk et al. 2004; Wolpaw et al. 1991]. Users concentrate on a certain type of thought (such as imagined hand movement) in order to *explicitly* control the system. This requires concentration, effort, and training and often seems unnatural. Some require implanted electrodes in the skull [Kennedy et al. 2000; Moore et al. 2001; Moore and Kennedy 2000] or long training periods with limited bandwidth [Millán et al. 2004]. While these systems provide this group of users with a valuable communication channel, they likely will not see wider adoption due to the low bandwidth compared to other available methods for nondisabled users. However, there have been recent examples of brain sensing used as explicit input for healthy users to make selections or control the interface, for example in a game context [Kuikkaniemi et al. 2010; O'Hara, Sellen and Harper 2011] or with a multitouch table [Vi and Subramanian 2012; Yuksel et al. 2010].

In contrast to these *active* and *reactive* BCIs [Zander et al. 2014] where users focus on the intentional brain activity of the user, Cutrell and Tan [2008] suggested that untrained users may benefit from systems that use pattern recognition and machine learning to classify signals users naturally give off when using a computer system. They suggest that the *implicit* commands inferred from a user's changing brain activity may be the most promising, universal application of brain–computer interaction. Zander's pioneering work has laid a foundation for such *passive brain–computer interfaces* [Zander et al. 2008; Rötting et al. 2009; Zander et al. 2010; Zander and Koethe 2011; Zander and Jatzev 2012; Zander et al. 2014]. Jacob et al. [1993] anticipated the notion of passive physiological monitoring, and recent advancements in physiological sensors have allowed them to become feasible. With brain and physiological sensing, the notion of *implicit input* extends to include the internal cognitive state of the user as context for the interaction. Systems are now being developed that use brain and body sensors to automatically discover aspects of the user's cognitive state and use this information as passive or implicit input to a system, augmenting any explicit input from other devices and increasing the bandwidth from humans to computers [Ayaz et al. 2013; Afergan et al. 2014a, 2014b; Derosiére et al. 2013; Grimes et al. 2008; Hirshfield et al. 2009, 2011; Lee and Tan 2006; Zander et al. 2009]. However, there are many challenges due to the ill-defined paradigms for implicit user interface design as well as the inherent noise in the signals. We address both of these concerns below by introducing design principles for user interfaces with implicit input of user state, as well as best practices for signal processing, data reduction, feature selection, and classification of noisy input signals for implicit user interfaces.

3. COGNITIVE STATE DETECTION WITH FNIRS

To explore the potential of fNIRS as an input modality, we have investigated its use for reliable detection of cognitive states that have direct relevance to human–computer interaction and have also built tools to facilitate both offline and real-time exploration of fNIRS signals. These are described in the sections to follow to provide further background for the design considerations and case studies described in this article and as a foundation for future work.

3.1. Offline Feasibility Studies for Cognitive State Detection

Through a series of controlled, offline studies, we have investigated cognitive states that can be classified reliably using fNIRS data, focusing on multitasking and varying workload scenarios that have direct relevance to many HCI scenarios. To explore cognitive workload, we used the ***n*-back task** paradigm, which has been shown to manipulate workload. To look more specifically at multitasking scenarios, we followed protocols developed by Koechlin et al. [1999] to induce various types of ***cognitive multitasking*** behavior. These experiments are described later and are the foundation for the real-time systems that we've developed. Researchers have also explored the fNIRS response during mental arithmetic tasks [Hoshi et al. 1997], changes in workload [Bunce et al. 2011; Herff et al. 2013; Izzetoglu 2003, 2004; Shimizu et al. 2009; St. John et al. 2004] and affect [Leon-Carrion et al. 2006], and motor tasks [Coyle et al. 2007; Sitaram et al. 2007].

3.1.1. Cognitive Workload (n-Back). One of the most widely used working memory tasks is the *n*-back test [e.g., Gevins and Cutillo 1993; Ayaz et al. 2007, 2012; Molteni et al. 2009; McKendrick et al. 2014; Fishburn et al. 2014]. In this task, participants are

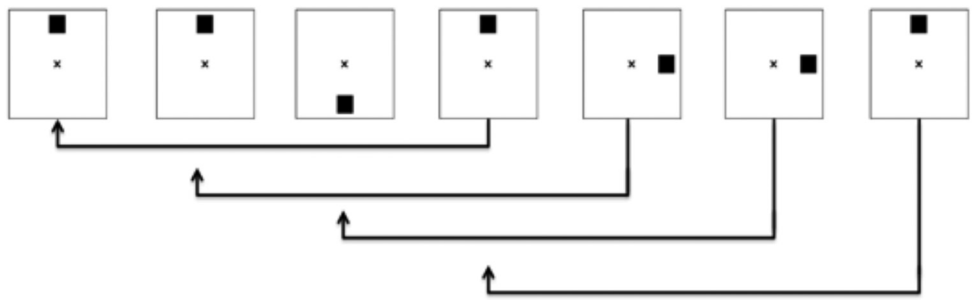

Fig. 2. 3-back visuospatial task. Participants were instructed to respond if the current square appears at the same location as the one presented three trials back in the sequence.

presented a sequence of stimuli and must indicate if the currently presented stimulus is the same as the stimulus presented n steps before it, where n generally is 0, 1, or 2. The stimuli are each presented for a short period, and then participants are given a period with no stimulus to respond. The n-back test is widely used and has been established to incite increasing levels of short-term working memory as n increases [Baddeley 2003; Owen et al. 2005], as well as yield distinguishable fNIRS signals [Ayaz et al. 2007; Cui et al. 2010; Herff et al. 2013; Li et al. 2005; Peck et al. 2013]. It has been shown in multiple studies [Mehler et al. 2009, 2012; Reimer et al. 2009] to increase task demand on a user as n increases "without requiring direct conflict with manual control or visual processing demands" [Mehler et al. 2009]. It has been used in studying spatial processing, working memory demands, and training [Molteni et al. 2009; McKendrick et al. 2014; Fishburn et al. 2014].

As n increases, participants remember more stimuli, and cognitive load increases. Two versions of this test are a visuospatial matching task (Figure 2), where shapes or letters are presented on a screen and participants indicate matches, and an audio recall version, where participants listen to stimuli and respond verbally with the number n stimuli ago.

3.1.2. Cognitive Multitasking. In addition to general workload, we have investigated the classification of cognitive multitasking states based on fNIRS data. This work builds from experiments described in Koechlin et al. [1999, 2000] with the goal of designing interfaces that recognize these states and behave in appropriate ways to support multitasking. We extended Koechlin et al.'s work [1999] by utilizing fNIRS instead of fMRI, which is not practical in HCI settings [Solovey et al. 2011]. Since the fNIRS sensors are placed on the forehead, they are particularly sensitive to changes in the anterior prefrontal cortex, where Koechlin et al. [1999] showed distinct activation profiles during *delay*, *dual,* and *branching* tasks. These states were defined as follows:

1. *Branching* is illustrated by the following scenario: *A user is tackling a complex programming task but is interrupted by an incoming email from her boss that is time sensitive*. It occurs when the user must "hold in mind goals while exploring and processing secondary goals" [Koechlin et al. 1999]. It involves aspects of both the *delay task* and the *dual task* states described next (Figure 3). Since this is challenging to users, automatically sensing this state would be valuable to HCI.

2. *Delay Task* occurs when the secondary task is ignored and therefore requires little attentional resources (e.g., *A user is tackling a complex programming assignment and at the same time gets instant messages that the user notices but ignores)*. The secondary task mainly delays response to the primary task.

3. *Dual Task* entails frequent task switching without the need to maintain information about the previous task (e.g., switching between responding to emails and

Fig. 3. Branching task. Primary and secondary tasks both require attentional resources to be allocated, and the primary task goal must be kept in mind over time.

responding to software support issues being logged). These tasks are referred to as dual task because there are two tasks that require attentional resources. These situations could also utilize adaptive support in the user interface, but the adaptive behavior would be distinct from that of branching.

Using fMRI for brain imaging, Koechlin et al. [1999] demonstrated that these three multitasking activities had different signatures in the anterior prefrontal cortex, the area that is best for measuring with fNIRS. Koechlin et al. [2000] later showed that even during branching, there were distinct activation profiles that varied depending on whether the participant could predict when task switching would occur or whether it was random. The experimental setup was almost identical to the earlier study, except that in all conditions, the branching paradigm was used. There were two experimental branching conditions and a control: *random* branching, *predictive* branching, and *control* branching. We showed that these states could, in fact, be distinguished using fNIRS as well, and that they could apply in realistic human–computer interaction tasks [Solovey et al. 2011]. The significance of these experiments lies in the fact that all task conditions look similar to an outside observer. They have the same stimuli and the same possible user responses. With brain sensing, however, it became possible to distinguish the conditions based on the distinct mental processes (and thus, distinct blood flow patterns) elicited by each task.

In addition, the cognitive states identified in these experiments have direct relevance to many HCI scenarios, particularly when a user is multitasking. Automatically recognizing that the user is experiencing one of these states provides an opportunity to build adaptive systems that support multitasking. For example, by recognizing that most interruptions are quickly ignored, as in the *delay* condition, the system could limit these types of interruptions or reduce their salience as appropriate. Further, if a user is currently experiencing a *branching* situation, the interface could better support maintaining the context of the primary task, whereas during *dual-task* scenarios this would be unnecessary. Finally, distinguishing between *predictive* and *random* scenarios could trigger the system to increase support when the user's tasks become unpredictable.

3.1.3. Toward Broader Cognitive State Detection. Further research is ongoing to identify additional cognitive states that can be detected with fNIRS. Prior fMRI studies provide indicators for the types of tasks that may activate the region being probed and often serve as a starting point for fNIRS work [e.g., Solovey et al. 2011]. However, activity deep in the brain will not be detected since fNIRS sensors probe only a few centimeters

Table I. Summary of Participants Analyzed and Trial Details

	Visuospatial n-Back A	Visuospatial n-Back B	Visuospatial n-Back C	Cognitive Manual Task
Subjects analyzed	16	8	2	14
Sessions analyzed	16	8	10	14
Total subjects	28	16	7	14
Total sessions	28	16	35	14
Trials per session	30	16	30	30
Trial length (sec)	25	40	25	30
Sampling rate (Hz)	11.79	6.25	11.79	11.79

into the brain cortex. Finally, the placement of the probes will impact the signals detected as described earlier.

3.1.4. Enabling Real-Time Brain Input. The preliminary offline studies build a foundation for fNIRS-based adaptive user interfaces by illustrating significant differences in the fNIRS signal in specific scenarios that could be used in HCI. By examining the n-back and multitasking paradigms with fNIRS, we brought research on cognitive activity to a system that is practical for HCI by showing that fNIRS sensors could detect states previously studied with fMRI (which cannot be used in HCI settings). Although all analysis was done offline, we found significant differences in the signals that suggest that a real-time classifier may be possible. These well-defined, validated, and simple tasks provide the basis for the *calibration phase* in most of our real-time studies described later. By using validated tasks, we can build a labeled dataset of brain data prior to running the real-time system. Based on the calibration tasks, a model can be built to recognize distinct multitasking and high-workload states and adapt behavior appropriately to better support the user.

3.2. Improving Feature Selection and Machine-Learning Approaches

While considerably more practical, fNIRS and EEG data can be noisy and less reliable than the more intrusive techniques (e.g., fMRI, MEG, surgically implanted electrodes), requiring machine-learning algorithms that can handle such data. To dissect the machine-learning process, we conducted an offline analysis of high-low workload manipulations for 46 sessions (39 subjects) collected over four experiments. For the purpose of evaluating the cleanest possible data, we focused on half of our dataset, using the half with the highest average classification accuracy (according to eight machine-learning algorithms operating over all evaluated features and time segments); an analysis on the full set of data yields similar conclusions but coats the dissected sources of meaning in the datasets in more noise. All experiments tested cognitive workload and consisted of a set of easy and hard trials in equal quantity (see Table I). The first three experiments were visuospatial n-back experiments, while the fourth was a cognitive manual task where users processed visual stimuli and pressed keys accordingly in a timely manner. To assess both the quality of data and efficacy of different machine-learning options, we conducted 46 leave-one-out cross-validations on the data, where the data held out in each fold was a single trial (30 in three experiments and 16 in the fourth). The machine-learning algorithm thus built a model over all but one trial, whose class (easy or hard) it predicted, repeating this procedure until every individual trial had supplied the testing case. In total, this resulted in 1,328 $((16{+}10 + 14) * 30 + (8 * 16))$ unique testing cases; in other words, the listed classification accuracies represent the model trained and testing on 1,328 separate occasions. If the fNIRS data was not indicative of the trial's associated class, the algorithm would likely classify roughly half of the trials correctly, and thus a classification accuracy well above 50%

Table II. Average Classification Accuracy by Algorithm

Algorithm	Average
Support Vector Machine: (*Weka SMO)*	74.3%
Multinomial Logistic Regression (*Weka)*	71.7%
Support Vector Machine: LibSVM (*Weka wrapper)*	67.6%
Adaboost on Decision Stump (Weka)	67.6%
Logistic Model Tree (LMT, Weka)	67.6%
Simple Logistic Regression (Weka)	67.5%
Naive Bayes (Weka)	65.4%
3-Nearest Neighbor (Weka)	64.4%

Table III. Classification Accuracy by Feature

Linear Slope	Standard Deviation	Mininum	Time to Peak	Absolute of Mean	Maximum	Mean	Absolute of Slope	Full Width at Half Maximum	Second Derivative	Average Accuracy
66.3	63.6	60.4	59.9	59.3	59.0	58.4	57.5	57.1	46.8	
✓	✓									69.0
✓	✓	✓								69.2
✓	✓	✓	✓							71.3
✓	✓	✓	✓	✓						70.8
✓	✓	✓	✓	✓	✓					70.9
✓	✓	✓	✓	✓	✓	✓				71.1
✓	✓	✓	✓	✓	✓	✓	✓			71.4
✓	✓	✓	✓	✓	✓	✓	✓	✓		72.5
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	73.3

over many trials suggests the presence of class-predictive information in the datasets and the capability of the associated algorithm to discover it. It is important to note that the confidence interval around the chance value is affected by the number of trials examined [Mueller-Putz et al. 2008]. What follows is a suite of evaluations on different machine-learning algorithms in order to (1) provide basic guidelines for how to build useful fNIRS analysis tools, (2) give deeper insight into what components of the data inform mental state, and (3) make explicit tradeoffs to consider both for the design of experiment and for the choice of machine-learning algorithm. In these results, the absolute classification accuracies are not as important as the relative accuracies, which indicate the effect of various combinations and choices of classification approach.

3.2.1. Machine-Learning Algorithms. Table II shows a comparison of different machine-learning algorithms. Weka's *SMO support vector machine* (with a polynomial kernel and cache (1.0) parameters) outperformed the other algorithms. It is worth noting that we examined different parameters (Weka's RBF and Puk kernels and different sizes for the cache) for *SMO,* but no combination of values led to superior results. In addition, procedures that tailored the selection of kernel and cache to the present training set using cross-validation had only a negligible impact on accuracy.

3.2.2. Descriptive Statistics. The second row of Table III shows the classification accuracies when the machine-learning algorithm is permitted to train on only one statistical description of the time segment at each channel. The analysis indicates that linear slope (the difference between the last and first value divided by the number of observations) appears to be the most informative way to describe the time segment and the best starting point for analysis. The subsequent rows show accuracies when the next most independently productive feature is added to the selection. These results reflect

Table IV. Classification Accuracy by Time Segment

1/1	1/2	2/2	1/3	2/3	3/3	1/4	2/4	3/4	4/4	1/5	2/5	3/5	4/5	5/5	Average Accuracy
70.1%	63.0%	66.6%	58.5%	65.5%	66.0%	57.1%	62.5%	64.9%	66.6%	56.5%	60.4%	60.4%	64.2%	65.3%	
✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	74.3%
✓	-	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	72.9%
✓	-	-	-	-	-	✓	✓	✓	✓	-	-	-	-	-	73.5%
✓	-	-	-	-	-	-	-	-	-	✓	✓	✓	✓	✓	73.5%
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	74.2%

Table V. Classification Accuracy by Channel

A-DC4 (left, 830, 3.5cm)	66.7%
B-DC8 (right, 690, 3.0cm)	66.6%
B-DC7 (right, 690, 2.5cm)	66.3%
A-DC2 (left, 830, 2.5cm)	64.1%
B-DC6 (right, 690, 2.0cm)	63.7%
B-DC5 (right, 690, 1.5cm)	63.6%
A-DC1 (left, 830, 2.0cm)	63.2%
B-DC3 (right, 830, 3.0cm)	63.0%
A-DC3 (left, 830, 3.0cm)	61.8%
A-DC7 (left, 690, 2.5cm)	61.7%
B-DC1 (right, 830, 2.0cm)	61.2%
A-DC5 (left, 690, 1.5cm)	61.1%
A-DC8 (left, 690, 3.0cm)	60.5%
B-DC2 (right, 830, 2.5cm)	59.9%
B-DC4 (right, 830, 3.5cm)	59.7%
A-DC6 (left, 690, 2.0cm)	58.8%

the intuitive phenomenon that describing the trial in terms of more statistical features (even the least predictive ones) tends to boost accuracy.

3.2.3. Time Segments. To evaluate where in the time segment information is concentrated as well as determine optimal blocks for partitioning a trial, we evaluated the performance of classifiers on individual separate windows of the dataset and multiple aggregated windows of the dataset. In Table IV, the first row labels different windows of the dataset, so that 1/1 stands for the whole time segment treated as one cohesive unit and 4/5 stands for the fourth fifth of the dataset, the content starting at the 60 percentile timestamp and ending at 80%. The next row shows classification accuracies when the algorithm only examines that particular window. These results suggest that the task-predictive information is insulated toward the end of the trial, which is consistent with the fact that the measurement technique relies on the slow movement of blood. The rows that follow show classification accuracies when many of these time segments are aggregated together. These comparisons would suggest that it makes sense to partition the data into subsegments, but two or three suffice. One can partition the dataset into thirds, fourths, or fifths, but this does not buy the algorithm much new information. Examining the whole, the first half, and the second half appears to extract the bulk of information.

3.2.4. Channels. Most physiological sensors provide multiple channels of data. These experiments were run on an ISS Imagent with 16 channels of fNIRS data made up of two detectors with four linearly arranged light sources associated with each detector (Figure 1). The source–detector distances are listed in the Table V. An analysis of machine-learning performance on only one channel reveals that while all channels are independently informative, there was little difference between detectors sampling the left versus right prefrontal cortex and between those calibrated to identify 830- versus 690-nanometer reflected light (Table V). It is, however, worth noting that the two detectors farthest from the light source (3.5cm on the left and 3.0cm on the right) had the highest one-channel classification accuracy, which is expected since their positions enable them to sample the deepest neural tissue, while the shallow channels mainly contain noise and artifacts originating closer to the skin.

3.2.5. Feature Selection and Machine-Learning Discussion. The previous experiments show the effect of various parameter and preprocessing approaches. From this, we can recommend that a trial be broken into segments. For each channel in these segments, a set of statistical features can be computed. In our work, we have found the highest classification accuracy when using slope and standard deviation over a segment. For potential performance improvements, additional features could be added such as minimum, time to peak, absolute of mean, mean, largest, absolute of slope, full width at half maximum, absolute of slope, and second derivative. Using Weka's SMO support vector machine shows promise and can be used in classification.

This analysis suggests that even rudimentary filtering and attribute selection approaches can provide reasonable classification accuracy. Future work could focus on expanding the feature set. In particular, a suite of features that describes amplitudes at various frequencies on Fourier transformed segments may provide complementary information. Other features that better approximate how the segment changes over time are also promising candidates for a better algorithm.

3.3. Real-Time Brain Input Platform

Building from the offline investigations, we developed a real-time platform for studying brain-based adaptive, implicit user interfaces. It focuses on capturing brain activity from functional near-infrared spectroscopy sensors, but the main components can be used for other brain and body sensors as well. This platform expands the functionality of our Online fNIRS Analysis and Classification (OFAC) system [Girouard et al. 2010] and Brainput system [Solovey 2012]. The system learns to identify brain activity patterns occurring as a user experiences various cognitive states. It provides a continuous, supplemental input stream to an interactive system, which uses this information to modify its behavior to provide better support for the user. Thus, we can use noninvasive methods to detect signals coming from the brain that users naturally and effortlessly generate while using a computer system. By following the principles described earlier, this additional information can lead to systems that respond appropriately to changes in the user's state. After describing the architecture of the real-time brain input platform, we will describe applications built on top of the platform as well as user evaluation studies demonstrating that passive brain input significantly improves several performance metrics in various applications and domains.

In our implicit brain input platform, there are four high-level phases: baseline, calibration, modeling, and classification. While similar pipelines exist for other physiological computing systems [e.g., Zander et al. 2010], here we describe it specifically within the context of fNIRS and highlight some of the unique design considerations necessary to construct a real-time fNIRS classification system.

3.3.1. Baseline Phase. Collecting a baseline measure is a standard practice in fNIRS studies and it is used for later calculating changes in the oxygenated and deoxygenated hemoglobin values from the baseline values. Usually a baseline measure is collected for 30 to 60 seconds, and then an average baseline measure is calculated over this period. During this period, the user is asked to relax and think of nothing while focusing on a focal point on a computer screen. This can also be collected at intermediate points within a study, if it is desired to have a baseline that is closer in proximity to the activity period.

3.3.2. Calibration Phase. In order for the system to successfully classify fNIRS data and turn it into meaningful information for a user interface, training or calibration data is collected. The calibration session provides brain data recorded during validated tasks to train a machine-learning classifier for the individual that will be using the system, since there can be variation in brain processes across different people. A participant completes a set of tasks designed to elicit understood cognitive states. Examples of such tasks are described in Section 3.1. By performing these known tasks repeatedly, we generate a dataset of labeled data. This dataset is used in the modeling phase to build a machine-learning model that learns to find specific patterns in this data that indicate one cognitive state or another in future, unlabeled data.

The main architecture supports various calibration task paradigms. The only requirement is that indicators designate the start and end of a trial and the label that should be used for that trial. This may be done by following an accurately timed protocol or by sending markers into the acquisition software to indicate the start, end, and label for a period. The system continually reads fNIRS data from the acquisition software (part of the ISS Boxy system). Once the end of a single training trial is indicated, a labeled training example for the machine-learning model is generated and stored until the end of the calibration session. A training example consists of a sequence of data points from each of the sensor channels, so we have simultaneously collected several sequences that together are a training example.

3.3.3. Modeling Phase. Once the calibration period has ended, the modeling phase begins. During this phase, the baseline data and calibration data are used together to create a classification model for future brain data. The data from these periods is preprocessed as described next and then a machine-learning model is built.

Several preprocessing steps are taken to convert the raw data coming from the acquisition software into meaningful information. Although preprocessing strategies vary widely, we describe a set of best-practice approaches that we have found to be effective. For each of the channels, the mean baseline value across each of the samples is taken. This is used in both the modeling and classification phases. We then calculate the absorption coefficients for the training set, which uses the baseline measure, differential path length factors for the two wavelengths, and source–detector distances. An elliptic low-pass filter with a cutoff frequency of 0.025Hz, stoppage frequency of 0.03Hz, max ripple of 3dB, and stopband attenuation of 50dB is used to filter the data. We normalize the data by channel using a z-score. We calculate change in oxygenation for a sample by subtracting the value of the first sample point (or the mean over a short period before the stimulus) from all points in the sequence to look at the changes from the same starting point. Although all trials last for the same length of time, it is possible that the training examples could have slightly different lengths due to differences in sampling. To ensure that all training examples have the same number of features, the examples are shortened to the length of the shortest example in the training set. Once all of these preprocessing steps are performed on the training data, we build a classification model.

To build a classification model, we use Weka [Hall et al. 2009], which provides support for numerous machine-learning classifiers. The system can be modified easily to use any of the classification algorithms included in the Weka toolkit, and we described results of various choices in Section 3.2. If the training set is unbalanced, the smaller class is oversampled so that the classes are balanced before training.

3.3.4. Real-Time Classification Phase. Once the training and modeling phases are complete, the system enters the classification phase. As sensor readings are received from the acquisition software, they are collected into a sequence of the same length as the training examples. A sliding window is created as new data comes in, and the continuous sequences of data are analyzed. Each sequence is preprocessed in the same manner as the training data and then sent to the machine-learning classifier, which classifies the sequence in real time. For each example, the machine-learning model predicts which class it belongs to and can optionally give a probability value. These classifications are sent via TCP/IP to an interactive application, which can make adaptations as necessary. Because the system is not involved with deciding on the adaptations and when to implement adaptations, it is generalizable and can be ported to many applications with only minimal changes.

3.3.5. Replay Mode. In addition to the normal online mode where signals are classified and sent in real time, the system supports replay mode, which simulates the analysis and classification of previously recorded data and is useful in experimenting with various adaptive strategies.

4. DESIGNING IMPLICIT INTERFACES FOR PHYSIOLOGICAL COMPUTING

The platform we have developed allows us to explore adaptive behavior to find the best strategies for use in implicit interactive systems and experimentally evaluate them. While our work has focused on fNIRS input, implicit user interfaces utilizing brain and body sensor data all share important characteristics and together define a new class of user interfaces. Below, we describe the design principles and framework that has emerged from our work in developing interactive systems with implicit fNIRS input.

4.1. Design Principles for Implicit Input

By definition, implicit input is passively obtained from the user, unlike explicit input from devices such as a mouse or keyboard. In addition, sensor data is often noisy, is constantly changing, and is continuous, unlike a discrete menu selection or mouse click. Further, the machine-learning classification algorithms only provide estimates of cognitive state, with some inherent level of uncertainty. The nature of this input requires careful consideration to ensure successful user interface design. We outline high-level principles that can guide development of interfaces that can take advantage of implicit input channels such as that coming from brain and body sensors (e.g., passive brain–computer interfaces). Many of these principles also apply to other similar noisy input channels.

For nondisabled users, passive channels of input are most useful when *augmenting other input devices* and providing a *supplemental channel* that indicates user state, instead of being the primary source of input in a system. In addition, because physiological data can be noisy, the adaptations must be *resilient to misclassifications*. One way to address this is to test the confidence and probability of predictions in order to influence the interface only when the system is reasonably certain of physiological state. Because the prediction might not always be correct, a designer should *avoid irreversible ormission-critical adaptations*. Instead, the adaptations must be used in a paradigm where the benefits outweigh the costs, and where a high number of correct adaptations can improve performance more than the damages from incorrect classifications, such as the user experiencing a loss of control. The adaptation should make *subtle, helpful changes* to the interface that would not be too disruptive if the user's state is misinterpreted. For example, cognitive state information may be used to change future interactions or to prechoose defaults, rather than to make prominent changes directly to the current display. Other potential types of interfaces would be those with multiple views or with limited screen real estate. The brain data could be used to make tradeoffs based on the user's cognitive state. Interface adaptations, which run the risk of causing confusion and adding to the user's workload, must be designed carefully to avoid performance decrements. In particular, care must be taken to avoid surprising or confusing the user by making unexpected changes to the interface. This calls for care and subtlety in the interface design.

4.2. Adaptive Strategies in Interactive Systems

Based on the high-level principles described above, we propose a conceptual framework in which adaptations are categorized by their target functional level and immediacy. Specifically, adaptations can affect the *semantic* or *syntactic* levels of a system [Foley and Van Dam 1982] and can be implemented as either *immediate* or *future* changes to the interface. This framework is meant for guidance for developers creating novel adaptive systems, rather than the establishment of a rigid organization system.

The *semantic* level of a system refers to the functions performed and the system's internal values and parameters, while the *syntactic* level of a system refers to the sequence of inputs and outputs but not the values of these operations [Foley and Van Dam 1982; Jacob 1983, 1986]. Thus, we can think of the semantic changes as ones that affect the behavior of the system and the goals and actions of a user, whereas syntactic changes are based on the user interface and do not modify the content of the application [Jacob 2001].

The interactive system can be adjusted in two different levels of immediacy. *Immediate* changes affect the elements currently on screen or being interacted with, while *future* changes adjust variables and elements that have not yet appeared. Immediate adjustments have the advantage of mapping directly to the user's experience and having a direct effect. However, they need to be done subtly in order to not disrupt the user experience. Future changes can be effective because stronger changes to the system may occur without surprising the user. However, future changes to the state of the system may be difficult to implement and evaluate.

We propose that these strategies can be implemented in conjunction with each other to produce distinct adaptive strategies for interactive systems [Afergan 2014]. Next, we outline how each of these adaptations affects a system, following the design principles discussed earlier. We then illustrate these principles through descriptions of several systems that we have built and evaluated.

In ***immediate semantic*** adaptations, the main display stays consistent; however, actions triggered by interacting with on-screen elements change according to implicit cognitive state input. The system may take control of elements such as timing or actions of elements that are currently displayed, change the effects of input devices, or adapt autonomy levels. This is the basis of Prinzel and Wilson's biocybernetic adaptive automation loops [Prinzel et al. 2000; Wilson and Russell 2007].

In ***future semantic*** adaptations, the display does not change, but over time the underlying functionality may change based on the implicit cognitive state input. An example of this might be search results, where information can vary in content. Physiological sensors could monitor user state during interaction with the information and map visual designs with metrics such as engagement or preference. Over time, an intelligent system could compare the user's state across different information delivery mechanisms and slowly gravitate toward personalized interfaces that elicit better performance and cognitive measures.

In ***immediate syntactic*** adaptation, information on screen is modified, filtered, or emphasized according to a user's cognitive state. This can be done via methods such as changing the peripheral data or layering of information on a display and may aid the focus of the user by making critical information more salient at critical moments.

Constant updates across the entire interface or changes in the display format may be jarring and unsettling for users and disrupt their ability to form cohesive mental models of the system. Instead, subtle modifications to inactive elements on screen may clarify the display for the user. Here, we can leverage the high temporal resolution of physiological sensors. Rather than evaluating the entire interface as one cohesive entity, we can evaluate individual interface elements and personalize them in a way that they might best serve the user or cater to the user's current cognitive state.

In a ***future syntactic*** adaptation, we change the upcoming layout of a system based on cognitive state. Combining user state with predictions of how the user will react to user interface elements, the system can decide on how to appropriately present information to the user. We can modify the type of visualization or stimuli, level of detail, number of options initially visible in a menu, or size of visual elements to provide what might be most suitable for the user.

While these strategies differ in implementation, they follow the high-level design principles described in Section 4.1 to minimize user disruption and make subtle, meaningful changes to an interface. The context of an application will dictate which strategy is the most appropriate, and we can suggest best practices based on our experience exploring several applications using the platform we have built for real-time, implicit brain input described in Section 3.3.

5. CASE STUDIES

To illustrate the conceptual framework described above, we present case studies from systems and experiments we have published previously. These systems demonstrate ***immediate*** and ***future*** adaptations on the ***semantic*** or ***syntactic*** level of the user interface. These case studies also show the flexibility of the real-time adaptive platform (Section 3.3), which supported various calibration, modeling, and classification approaches in order to provide the appropriate implicit brain input to the interactive system being studied.

5.1. Immediate Semantic Adaptive Strategy: Adapting Autonomy Levels

One strategy for designing implicit interfaces described previously is to enable ***immediate semantic*** changes via an adaptive backend in the system. We demonstrated this by integrating the sensing platform with an adaptive robot architecture that could adapt robot behavior to better support and collaborate with the human operator [Solovey et al. 2012]. The focus in this project was to adapt the autonomy level of the robots, based on cognitive state classification. The participant supervised two simulated autonomous robots in a navigation task in which the target was an area with a high signal strength. This required the participant to constantly switch context between the two robots, maintaining information about each robot's current location. This type of task is similar to the branching scenario described in Section 3.1.2 and has been linked with increased activation in the prefrontal cortex [Koechlin et al. 1999; Solovey et al. 2011]. The **calibration phase** consisted of a series of *branching* and *nonbranching* tasks described by Koechlin et al. [1999]. During the **classification phase,** the participant had a graphical input panel to give commands such as "turn right" or "take a reading" to measure the signal strength in the current location. In addition, the system implicitly provided cognitive state input and the system could adapt behavior based on this. One robot went into autonomous mode whenever a *branching* state was detected, based on the model built from the calibration data. This allowed the operator to focus on the other robot's progress. The robot exited autonomy mode when a *nonbranching* state was detected, and the participant was required to give instructions to the robot about where to explore. In the autonomous mode, the robot would take over the search task, periodically sensing signal strength and making appropriate course adjustments

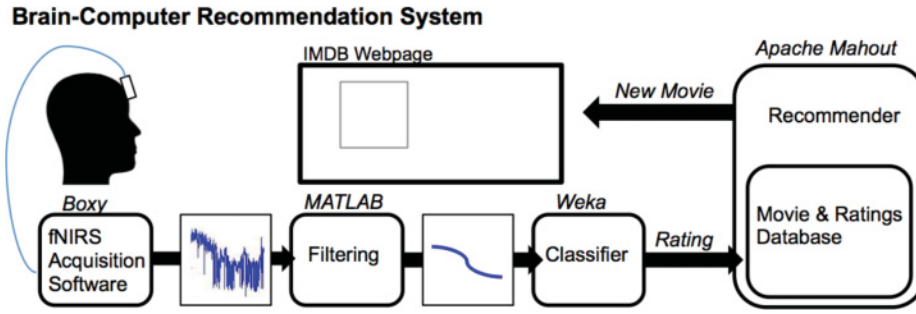

Fig. 4. Brain–computer recommendation system.

to ensure progress toward target location. However, the robot could be interrupted by the operator and given a command manually and would return to autonomous behavior after completing requested action.

This system was designed with the principles described earlier. As in most brain and body sensing systems, the data is noisy and constantly changing. Here, we used the brain data to change the autonomy level of one of the robots. This does not affect the primary navigation task, and the user can always provide commands to the robot, even during autonomy mode. This is similar to if you were working with another human who may recognize that you are busy and try to help out as appropriate.

5.2. Future Semantic Adaptive Strategy: Information Filtering System

Another strategy for constructing adaptive systems that adhere to good design guidelines is to modify the ***future*** output of a system with ***semantic*** changes. When new information enters a system, we can use physiological input to modify the selection, format, or timing of delivery of that information. In particular, when users have little expectation about the content of incoming information, this adaptation strategy is more likely to adhere to the high-level design principles of being *subtle* and *unobtrusive*. Poor classifications of state do not severely disrupt the user's interaction or mental model of the interface and, as a result, are more robust to misclassification problems that frequently accompany physiological computing systems. As an example, search results, product recommendations, and suggested music playlists are all personalized based on information filtering algorithms. The cost of potentially delivering a poor product recommendation is relatively low compared to the benefit of discovering relevant new products that may have been previously unknown to the user.

To illustrate this strategy, we describe a study exploring the feasibility of using fNIRS data as input to information filtering algorithms that suggested improved movie recommendations [Peck et al. 2013]. By building a functioning movie recommendation system that reacts to implicit fNIRS input (Figure 4), we demonstrate the potential of modifying the future output of a system as an adaptive strategy for physiological computing systems.

5.2.1. Evaluating Brain-Driven Movie Recommendations. During this study, the **calibration phase** consisted of sending fNIRS examples to a machine-learning model on known values, or in this case, movies that we already know the participant likes or dislikes. At the start of the experiment, we provided participants with a list of movies picked from IMDB's list of 250 best movies and 100 worst movies and asked them to select their top three and bottom three. Participants viewed a timed slideshow of selected movie web pages during which we recorded their brain activity with fNIRS.

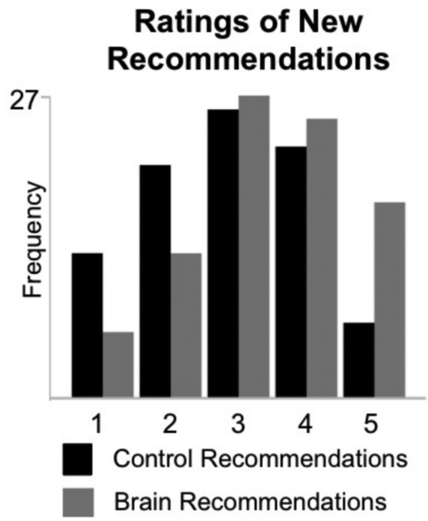

Fig. 5. Histogram of ratings in the two conditions. We see that brain recommendations tend to be rated higher—mostly 3s, 4s, and 5s.

In the **classification phase**, participants viewed two trials (control and recommender conditions), each of which consisted of a string of 20 movie websites, viewed sequentially. In the *control condition*, a series of predefined movies with average ratings are used for all participants. This serves as a baseline for a system that is not driven by brain input. In the *brain recommender condition*, implicit preference ratings, as predicted by our fNIRS data classifier, are fed into a movie recommendation engine. We show the same start movie as the control condition, but new movies are selected based on previous preference values. For example, the third movie is based on recorded preferences for the first and second movies.

5.2.2. Improved Recommendations. The study had 14 participants aged 19 to 28. To evaluate the success of our system, we analyzed two dependent measures:

(1) *Recommendation ratings by condition*: How did participants rate movies in the brain recommender condition in comparison to the control condition?
(2) *Recommendations over time*: A good recommender should improve over time as it constructs a more accurate picture of the user's likes and dislikes. Does the brain-driven recommender give better recommendations over time?

5.2.3. Recommendation Ratings by Condition. The key finding was that the brain recommender provided higher-rated movies than the control condition as the experiment session progressed. The distribution of all ratings in each condition is displayed in Figure 5. We would not expect a recommendation system to perform well until it had seen enough examples to provide suitable recommendations. Running Mann-Whitney's U test on movies 14 to 20 revealed a significant effect of condition (the mean ranks of the control condition and brain recommendation condition were 10.46 and 18.54, respectively; $U = 41.5$, $Z = 2.88$, $p < 0.01$, $r = 0.54$). We also found that 125 out of 280 (45%) movie recommendations in the brain condition were unique selections, meaning that each participant saw an average of nine movies no other participant viewed. These results support our primary hypothesis that the brain-driven recommendation system recommended movies that catered to the participant's individual preferences.

5.2.4. Recommendation over Time. Independent of the control condition, we find that recommendations from our system improved over time, suggesting that the preference model was gradually learning about the user. Across all participants, we analyzed the median rating given to movies at each time point (1–20) for each condition. For the brain recommender, we ran a linear regression and found that the total number of movies seen was a predictor of rating ($b = 0.046$, $t(20) = 2.541$, $p = 0.021$). This means that over the course of 20 movies, the median recommendation improved by roughly 1 rating point (from 3 to 4 out of 5). The overall model fit was $R^2 = 0.223$. By comparison, applying a regression to the control condition determined that the number of movies seen did not predict movie rating ($b = 0.004$, $t(20) = 0.154$, $p = 0.898$).

5.2.5. Discussion. Despite classifying user preference at relatively low accuracy levels, we were able to increase the user's satisfaction with the system largely because the user had very little expectation about future movie recommendations. Again, this acts as the primary advantage of manipulating the future semantics of a system. We can view the potential value of this adaptive technique in two ways.

First, we can augment systems that already modify or prioritize future output to the user. For example, services like Amazon or Netflix already have sophisticated recommendation algorithms that rely on nonphysiological information sources. In this case, one can envision Amazon or Netflix combining brain ratings with other implicit signals, such as purchase history and viewing history, to improve the overall accuracy of their model. Additionally, the user may be engaged in a high-performance task where avoiding disruptions is critical. These implicit measures help preserve user attention because they do not force an externalization of subjective feelings onto a rating scale. In either case, the role of the brain is minimal, positively nudging the user's engagement with information over a long period of time.

Second, we can construct new adaptive systems that have not been able to collect enough information about the user (either because the user is busy or this is no natural input mechanism) to modify its future output. For example, we can imagine a car radio station that naturally adapts its music to individual preferences without any intervention from the user. Or we could construct a notification system that waits until an opportune time in order to interrupt users with new emails. In this case, any help we can provide the user is better than the existing state of the art. If we classify user state incorrectly, the system will still behave in a manner that is nearly indistinguishable from its original functionality.

This work provides a simple snapshot of the potential of using brain signals to modify the ***future semantic*** output of a system. With increasing consumer interest in personalization, we believe that this adaptation strategy has the potential to be integrated in a number of applications.

5.3. Immediate Syntactic Adaptation: Dynamic Difficulty and Task Allocation

Here, we provide an example of ***immediate syntactic*** adaptation in a system for unmanned aerial vehicle (UAV) path planning. It has been shown that avoiding extended periods of too low or too high workload in a task may lead to a state of immersion and increased engagement. We modulated the number of UAVs that an operator controlled according to signals of low and high workload and were able to decrease operator error and increase task engagement [Afergan et al. 2014a]. Although modifying information in an *immediate syntactic* manner has the potential to disrupt a user's mental model of a system, we can mitigate this risk by making subtle changes only when we are confident of user state and aid the user in staying focused on the task and improving performance.

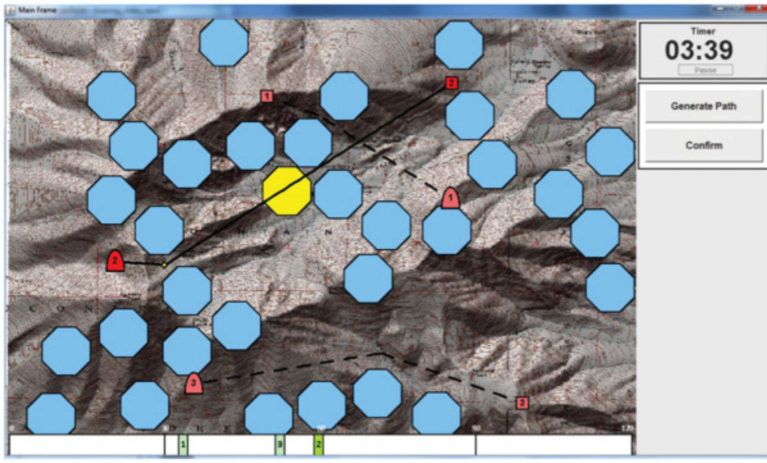

Fig. 6. View of UAV operator simulation. Participants guided UAVs to targets while avoiding dynamic obstacles.

5.3.1. Study Design. In order to induce periods of known high and low visuospatial workload during the **calibration phase**, participants performed 30 trials of a visuospatial 1-back and 3-back task [Baddeley 2003], as described in Section 3.1.1. For each trial, participants saw 10 stimuli appearing for 0.5 seconds and had 2 seconds to respond after each. Each trial totaled 25 seconds, and the hemodynamic patterns from these trials were used to calibrate the system by creating a machine-learning model of two levels of workload for each participant.

During the **classification phase,** participants' task was to guide between three and seven UAVs to a series of targets while avoiding obstacles that appeared and disappeared on screen (Figure 6). Operators were instructed that obstacles, shown as teal octagons, were no-fly zones, and that while UAVs could fly through them, there would be a large penalty for doing so. If entered, obstacles should be exited as soon as possible. Leaving UAVs idle for a long period of time would also result in a penalty, so participants were motivated to balance performing the task quickly and without collisions.

The participants were instructed that they were part of a team of UAV operators and that vehicles would be passed off from their control to other operators, and other operators' vehicles would be passed to them. Thus, participants were prepared for vehicles to appear and disappear during the task. To prevent disruption of the user's mental model of the scenario, UAVs were only removed if there were no obstacles in its path, meaning that the UAV should not demand any of the user's attentional resources and thus the user would not be distracted by the change.

Based on the implicit brain input, UAVs were removed and added during extended periods of high or low workload, respectively. In the *adaptive* condition, UAVs were added and removed according to brain signals correlating with low and high workload. After a UAV was added or removed, there was a 20-second period where no additional vehicles were added or removed. This prevented the user from having to rapidly switch contexts. In the *nonadaptive* condition, the simulation did not keep track of user state and intermittently added and removed UAVs (a random interval between 20 and 40 seconds). This timing was determined based on a series of pilots to correspond as closely as possible to the average number of additions and removals we observed in the adaptive condition.

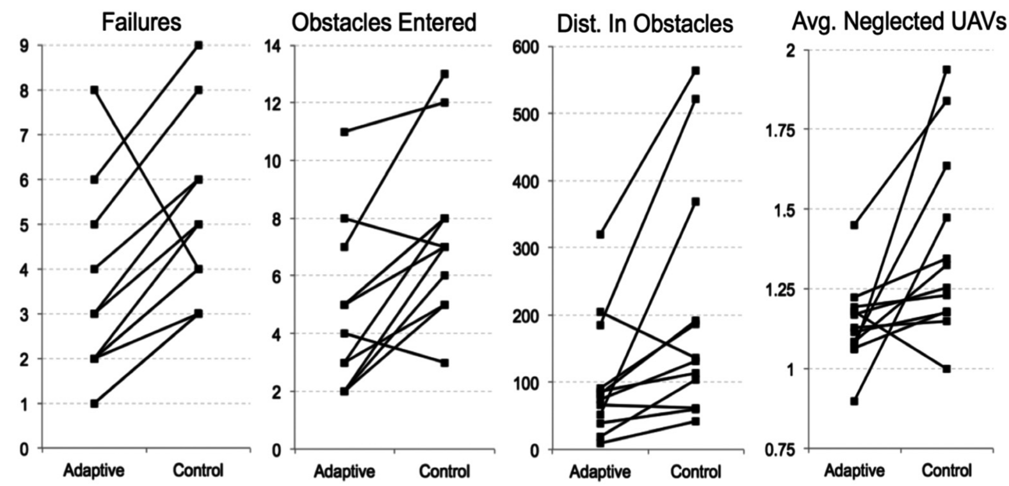

Fig. 7. We use a slopegraph to plot the effects of condition on UAV operator performance for each participant. The four measures showed significantly ($p < 0.05$) better performance in the adaptive condition interface (upward sloping lines), and at least ten of the 12 subjects performed better in the adaptive condition in each of these measures.

5.3.2. Increased Awareness. Across conditions, participants controlled roughly the same number of UAVs over time (no significant difference in number of UAVs). While participants completed the same number of successful trials across conditions, their failure rate was significantly higher in the nonadaptive condition (Figure 7). With a paired t-test, we found that participants had significantly fewer failures in the adaptive condition ($M = 3.25$; $SD = 2.14$) than in the nonadaptive condition ($M = 5; SD = 1.95$) ($t(11) = 3.17; p < 0.01$; Cohen's $d = 0.92$). In addition, although there was no significant difference in the number of obstacles that appeared in the UAVs' paths across conditions (since the obstacles randomly moved over time), participants entered an average of 4.75 ($SD = 2.77$) no-fly zones in the adaptive condition and 7.42 ($SD = 2:81$) no-fly zones in the nonadaptive condition ($t(11) = 4.14; p < 0.01$; Cohen's $d = 1.2$).

Because participants neglected fewer UAVs despite all other factors being consistent across conditions, we hypothesize that they paid more attention when workload was manipulated according to the implicit cognitive state input. We also conclude that the user's increased awareness in the adaptive condition, as demonstrated by distance traveled inside obstacles and neglected UAVs, suggests that the adaptive mechanism successfully preserved the user's mental model of the scenario while modulating the challenge level.

5.4. Future Syntactic Adaptation: Toward Personalized Visualizations

Similar to the future content modifications in the fNIRS-driven movie recommendation engine, we can imagine a system that modifies the ***future syntactic*** representation of content delivered to a user. Such a system would have the same advantage as when we proposed to modify the future semantics of a system. Most significantly, in scenarios where users do not have strong expectations about incoming information, modifying the future syntax may be imperceptible to users, minimizing the negative impact of misclassifications and noisy input from brain sensors.

As an example, most current visualization systems are created with a "one size fits all" approach that leverages general design guidelines. However, research increasingly suggests that a user's personality, experience, and cognitive state can impact a user's performance and analytical capabilities with an information visualization design. By monitoring cognitive state as people interact with a visualization, we may be able to identify the most effective visual designs and personalize future representations with the end goal of optimizing interaction.

In a study intended to move toward this goal, we used fNIRS to capture brain activity during user interactions with different graph types [Peck et al. 2013]. In a complex working memory task in which participants engaged with different visualizations, we found not only that participants differed in which visual design they believed was most cognitively taxing but also that this difference was clearly encoded in fNIRS brain data. As a result, we believe that fNIRS can be sensitive to the cognitive impact of visual design in analytical tasks, and this information may be used to optimize design to the user's current state.

Although there is little existing work that successfully makes syntactic changes in real time, there is an increasing body of research that suggests that this approach may be fruitful [e.g., Afergan et al. 2013; Kardan and Conati 2012; Peck et al. 2012; Steichen et al. 2013]. Thomas and Cook [2005] note that the development of intelligent systems that can aid users in performing analytical tasks is a significant direction for visual analytics research. Similar to the success of information filtering systems on the web, we believe that personalizing the syntax of future information can positively impact users' engagement with information and act as a viable adaptation technique for brain–computer interfaces.

5.5. Case Studies Discussion

These examples of real-time adaptations show that we can improve performance by using physiological data as a passive input to a system. This input may be particularly useful because when a user is in a state of information or memory overload, the user will not have spare cognitive resources to also manually or directly indicate his or her state. While these examples are particular to fNIRS, they are general enough to be used with other physiological sensors. However, fNIRS has a number of distinct advantages that makes it ideal for passive input. It is noninvasive and comfortable and requires a very short and easy setup. Because the signal is relatively robust to movement artifacts, it is ideal for situations where the user is performing a task normally. However, fNIRS is best used for interactive systems that require a slow response indicating a persistent state, according to specific states that have been measured so far (mainly cognitive workload and multitasking). In addition, data filtering and feature definition can aid in classification accuracy.

6. IMPROVING REAL-TIME ADAPTIVE BEHAVIOR

For successful use of brain, body, or other sensor data, interactive systems should handle noisy signals as well as occasional misclassification or user state. The adaptive interaction strategies suggested previously minimize the impact of misclassifications. To further reduce the impact of misclassifications, we can modify the way that a system triggers adaptations. In this section, we discuss the use of classification probability to construct more gentle and unobtrusive adaptive systems.

6.1. Modifying Adaptive Mechanism Based on Confidence in Model

Consider our movie recommendation system [Peck et al. 2013], in which the model was only capable of achieving a reasonable classification accuracy by differentiating between periods of low and high preference. In a real-world scenario, user preference is often more nuanced. When users feel uncertain or neutral about a movie, they typically do not assign ratings of extremely low preference or extremely high preference, as neither is a fair representation of their state. By extracting a measure of confidence in a model's classification accuracy, it is possible to design a more nuanced approach to triggering adaptive behavior and avoid extreme, discrete responses by the computer.

In the case of the movie recommendation system, we used a channel-based voting system to create an ad hoc value of confidence in the user model. In this implementation, a separate classifier was constructed for each of the 16 information channels on the fNIRS device (2 probes $\times$ 4 distances $\times$ 2 wavelengths). Each time the adaptive system polled the model for a prediction of user state, each channel provided its own classification of user state, which was then translated into a voting percentage. For example, if 16/16 channels agreed on a user state, the confidence in that value would be 100%. If 8/16 channels agreed on a user state, then the confidence in that value would only be 50%.

A system deployed in a real-world scenario might ignore all classifications that fall beneath some confidence threshold, ensuring that the information integrated into the user model is more likely to be reliable. However, due to experimental time constraints, the movie recommendation system needed to provide a new movie prediction with each and every movie viewed. Thus, we mapped our confidence values to the 5-star movie rating system: 50% to 60% confidence in a high-preference classification mapped to 3 stars, 60% to 80% confidence mapped to 4 stars, and 80% to 100% mapped to 5 stars. While classification is not equivalent to preference intensity, using a graded mapping strategy allowed us to use a more gentle recommendation when we were uncertain of the accuracy of our input. As a result, despite having relatively low classification accuracies in the movie recommendation system, we were able to boost the user's median rating of movie recommendations from 3 out of 5 stars to 4 out of 5 stars.

6.2. Modifying Adaptive Trigger Based on Probability Value

One can also use machine learning to provide probability values that help guide our predictions and adaptations. While a classifier must provide a prediction each time it is given, many machine-learning techniques can also assign the likelihood that a data sample belongs to each of the classes. By using these probability values, we can see how representative a sample is and this can help glean how valuable these predictions are. This technique has been used for MRI-based diagnosis as well [Kuncheva et al. 2010; Nouretdinov et al. 2011].

We [Afergan et al. 2014a] used the probability values returned from an LIBSVM machine-learning classifier to determine the probability values of a classification over time. The UAV simulation kept an 8-second sliding window of prediction values (with predictions every 0.5 seconds) of low and high workload and only made adaptations when overall probability was above 80% for that time period for predicting low or high probability. In order for adaptation to occur, there needed to be several consecutive high-probability predictions of a common class. This scheme allowed the system to effectively filter classifications that were likely inaccurate while permitting the confident predictions to drive adaptation.

7. CONCLUSIONS

In this article, we draw on our experience designing interactive systems that employ functional near-infrared spectroscopy to overcome as well as complement some of the drawbacks of other neuroimaging systems for HCI settings. Because it is an emerging technique, there have been relatively few studies showing specific measurements with fNIRS and their appropriate use in HCI. This article describes foundational studies exploring the feasibility and potential of fNIRS for HCI, as well as the real-time platform we have built for studying such systems. With this system, brain activity data can be used as a continuous input stream to an interactive system, making the system more in sync with the user and providing appropriate help and support when needed.

However, the adaptations must be done judiciously in order to help the user. To facilitate this, we present a conceptual framework along with design principles and patterns for designing interfaces that effectively take advantage of supplementary, implicit input channels such as the cognitive state information coming from neuroimaging tools. These are illustrated by the case studies (Section 5), in which we built and evaluated several systems that effectively use the fNIRS input to adapt an interactive system to better support the user.

Implicit, passive brain computer interfaces show promise to increase the bandwidth between the user and the computing system without additional work or conscious thought on the part of the user. These are an early step toward computers that can interpret the user's cognitive state and adapt accordingly. The ability to capture subtle changes in the user's cognitive state in real time opens up new doors in human–computer interaction research.

ACKNOWLEDGMENTS

We would like to thank Beste Filiz Yuksel, Tomoki Shibata, Francine Lalooses, A. J. Jenkins, Sergio Fantini, Angelo Sassaroli, Remco Chang, and the VALT Lab at Tufts as well as Missy Cummings and the Humans and Automation Lab at MIT.

REFERENCES

D. Afergan. 2014. Using brain-computer interfaces for implicit input. *Adjunct Proc. ACM UIST 2014*, ACM Press.

D. Afergan, E. M. Peck, R. Chang, and R. J. K. Jacob. 2013. Using passive input to adapt visualization systems to the individual. In *Proceedings of the CHI 2013 Workshop: Many People, Many Eyes: Aggregating Influences of Visual Perception on User Interface Design*.

D. Afergan, E. M. Peck, E. T. Solovey, A. Jenkins, S. W. Hincks, E. T. Brown, R. Chang, and R. J. K. Jacob. 2014a. Dynamic difficulty using brain metrics of workload. In *Proceedings of the ACM CHI 2014*. ACM.

D. Afergan, T. Shibata, S. W. Hincks, E. M. Peck, B. F. Yuksel, R. Chang, and R. J. K. Jacob. 2014b. Brain-based target expansion. In *Proceedings of the ACM UIST 2014*. ACM.

H. Ayaz, M. Izzetoglu, S. Bunce, T. Heiman-Patterson, and B. Onaral. 2007. Detecting cognitive activity related hemodynamic signal for brain computer interface using functional near infrared spectroscopy. In *Proceedings of the IEEE/EMBS Conference on Neural Engineering*. 342–345.

H. Ayaz, B. Onaral, K. Izzetoglu, P. A. Shewokis, R. McKendrick, and R. Parasuraman. 2013. Continuous monitoring of brain dynamics with functional near infrared spectroscopy as a tool for neuroergonomic research: Empirical examples and a technological development. *Frontiers in Human Neuroscience* 7, 1–13.

H. Ayaz, A. Shewokis, S. Bunce, K. Izzetoglu, B. Willems, and B. Onaral. 2012. Optical brain monitoring for operator training and mental workload assessment. *Neuroimage* 59, 36–47.

H. Ayaz, B. Willems, B. Bunce, P. A. Shewokis, K. Izzetoglu, S. Hah, A. R. Deshmukh, and B. Onaral. 2010. Cognitive workload assessment of air traffic controllers using optical brain imaging sensors. *Advances in Understanding Human Performance: Neuroergonomics, Human Factors Design and Special Populations*, T. Marke, W. Karwowski, and V. Rice (Eds.). CRC Press Taylor & Francis, 21–31.

A. Baddeley. 2003. Working memory: Looking back and looking forward. *Nature Reviews Neuroscience* 4(10), 829–839.

S. C. Bunce, K. Izzetoglu, H. Ayaz, P. Shewokis, M. Izzetoglu, K. Pourrezaei, and B. Onaral. 2011. Implementation of fNIRS for monitoring levels of expertise and mental workload. In *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems*, Springer, Berlin, 13–22.

B. Chance, E. Anday, S. Nioka, S. Zhou, L. Hong, K. Worden, C. Li, T. Murray, Y. Ovetsky, D. Pidikiti, and R. Thomas. A novel method for fast imaging of brain function, non-invasively, with light. *Optics Express* 10, 2 (1988), 411–423.

X. Cui, S. Bray, and A. L. Reiss. 2010. Speeded near infrared spectroscopy (NIRS) response detection. *PLOS ONE* 5, 11.

E. Cutrell and D. Tan. 2008. BCI for passive input in HCI. In *Proceedings of the ACM CHI 2008*. ACM.

D. T. Delpy, M. Cope, P. van der Zee, S. Arridge, S. Wray, and J. Wyatt. 1988. Estimation of optical path length through tissue from direct time of flight measurements. *Physics in Medicine and Biology* 33 1433–1442.

G. Derosiére, K. Mandrick, G. Dray, T. E. Ward, and S. Perrey. 2013. NIRS-measured prefrontal cortex activity in neuroergonomics: Strengths and weaknesses. *Frontiers in Human Neuroscience* 7, 583.

A. C. Ehlis, C. G. Bähne, C. P. Jacob, M. J. Herrmann, and A. J. Fallgatter. 2008. Reduced lateral prefrontal activation in adult patients with attention-deficit/hyperactivity disorder (ADHD) during a working memory task: A functional near-infrared spectroscopy (fNIRS) study. *Journal of Psychiatric Research* 42, 13, 1060–1067.

S. H. Fairclough. 2009. Fundamentals of physiological computing. *Interacting with Computers* 21, 1–2, 133–145.

F. A. Fishburn, M. E. Norr, A. V. Medvedev, and C. J. Vaidya. 2014. Sensitivity of fNIRS to cognitive state and load. *Frontiers in Human Neuroscience* 8, 76.

J. Fogarty, S. E. Hudson, and J. Lai. 2004. Examining the robustness of sensor-based statistical models of human interruptibility. In *Proceedings of the ACM CHI 2004.* ACM.

J. D. Foley and A. Van Dam. 1982. *Fundamentals of Interactive Computer Graphics.* Addison-Wesley Systems Programming Series, Reading, MA.

L. Gagnon, R. J. Cooper, M. A. Yücel, K. L. Perdue, D. N. Greve, and D. A. Boas. 2012. Short separation channel location impacts the performance of short channel regression in NIRS. *NeuroImage* 59, 3, 2518–2528.

A. S. Gevins and B. C. Cutillo. 1993. Neuroelectric evidence for distributed processing in human working memory. *Electroencephalography and Clinical Neurophysiology* 87, 128–143.

A. Girouard, E. T. Solovey, L. M. Hirshfield, K. Chauncey, A. Sassaroli, S. Fantini, and R. J. K. Jacob. 2009. Distinguishing difficulty levels with non-invasive brain activity measurements. In *INTERACT 2009*.

D. Grimes, D. S. Tan, S. Hudson, P. Shenoy, and R. Rao. 2008. Feasibility and pragmatics of classifying working memory load with an electroencephalograph. In *Proceedings of the ACM CHI 2008.* ACM.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter* 11, 1, 10–18

C. Herff, D. Heger, O. Fortmann, J. Hennrich, F. Putze, and T. Schultz. 2013. Mental workload during n-back task-quantified in the prefrontal cortex using fNIRS. *Frontiers in Human Neuroscience* 7, 935.

M. J. Herrmann, T. Huter, M. M. Plichta, A. C. Ehlis, G. W. Alpers, A. Mühlberger, and A. J. Fallgatter. 2008. Enhancement of activity of the primary visual cortex during processing of emotional stimuli as measured with event-related functional near-infrared spectroscopy and event-related potentials. *Human Brain Mapping* 29, 1, 28–35

L. M. Hirshfield, A. Girouard, E. T. Solovey, R. J. K. Jacob, A. Sassaroli, Y. Tong, and S. Fantini. 2007. Human-computer interaction and brain measurement using functional near-infrared spectroscopy. In *Proceedings of the UIST 2007.* ACM.

L. M. Hirshfield, E. T. Solovey, A. Girouard, J. Kebinger, R. J. K. Jacob, A. Sassaroli, and S. Fantini. 2009. Brain measurement for usability testing and adaptive interfaces: An example of uncovering syntactic workload with functional near infrared spectroscopy. In *Proceedings of the ACM CHI 2009.* ACM.

L. M. Hirshfield, K. Chauncey, R. Gulotta, A. Girouard, E. T. Solovey, R. J. K. Jacob, A. Sassaroli, and S. Fantini. 2009. Combining electroencephalograph and near infrared spectroscopy to explore users' instantaneous and continuous mental workload states. In *Proceedings of the HCII'09*.

S. Hudson, J. Fogarty, C. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. Lee, and J. Yang. 2003. Predicting human interruptibility with sensors: A Wizard of Oz feasibility study. In *Proceedings of the ACM CHI 2003.* ACM Press.

K. Izzetoglu, S. Bunce, M. Izzetoglu, B. Onaral, and K. Pourrezaei. 2003. fNIR spectroscopy as a measure of cognitive task load. *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE.* IEEE 4, 3431–3434.

K. Izzetoglu, S. Bunce, B. Onaral, K. Pourrezaei, and B. Chance. 2004. Functional optical brain imaging using near-infrared during cognitive tasks. *International Journal of Human-Computer Interaction* 17, 2, 211–227.

M. Izzetoglu, K. Izzetoglu, S. Bunce, H. Ayaz, A. Devaraj, B. Onaral, and K. Pourrezaei. 2005. Functional near-infrared neuroimaging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 13, 2, 153–159.

R. J. K. Jacob. 1983. Using formal specifications in the design of a human-computer interface. *Communications of the ACM* 26, 4, 259–264.

R. J. K. Jacob. 1986. A specification language for direct-manipulation user interfaces. *ACM Transactions on Graphics* 5, 4, 283–317.

R. J. K. Jacob. 2001. Open Syntax: Improving access for all users. In *Proceedings of the WUAUC 2001.* ACM.

Z. A. Keirn and J. I. Aunon. 1990. A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering* 37, 12, 1209.

E. Koechlin, G. Basso, P. Pietrini, S. Panzer, and J. Grafman. 1999. The role of the anterior prefrontal cortex in human cognition. *Nature* 399, 1–4.

E. Koechlin, G. Corrado, P. Pietrini, and J. Grafman, 2000. Dissociating the role of the medial and lateral anterior prefrontal cortex in human planning. *Proceedings of the National Academy of Sciences* 97, 7651–7656.

L. I. Kuncheva, J. J. Rodríguez, C. O. Plumpton, D. E. Linden, and S. J. Johnston. 2010. Random subspace ensembles for fMRI classification. *IEEE Transactions on Medical Imaging* 29, 2, 531–542.

J. C. Lee and D. Tan. 2006. Using a low-cost electroencephalograph for task classification in HCI research. In *Proceedings of the ACM Symposium on User Interface Software and Technology 2006*. ACM Press.

J. Leon-Carrion, J. Damas, K. Izzetoglu, K. Pourrezai, J. F. Martín-Rodríguez, J. M. Barroso y Martin, and M. R. Dominguez-Morales. 2006. Differential time course and intensity of PFC activation for men and women in response to emotional stimuli: A functional near- infrared spectroscopy (fNIRS) study. *Neuroscience Letters* 403, 1–2, 90–95.

C. Li, H. Gong, Z. Gan, and Q. Luo. 2005. Monitoring of prefrontal cortex activation during verbal n-back task with 24-channel functional NIRS imager. *Proceedings of SPIE* 5630, 883.

R. L. Mappus IV, G. R. Venkatesh, C. Shastry, A. Israeli, and M. M. Jackson. 2009. An fNIR based BMI for letter construction using continuous control. In *ACM CHI 2009 Extended Abstracts*, ACM, 3571–3576.

R. Mandryk, M. Atkins, and K. Inkpen. 2006. A continuous and objective evaluation of emotional experience with interactive play environments. In *Proceedings of the ACM CHI 2006.* ACM.

R. McKendrick, H. Ayaz, R. Olmstead, and R. Parasuraman. 2014. Enhancing dual-task performance with verbal and spatial working memory training: Continuous monitoring of cerebral hemodynamics with NIRS. *Neuroimage* 85, 3, 1014–1026.

B. Mehler, B. Reimer, and J. F. Coughlin. 2012. Sensitivity of physiological measures for detecting systematic variations in cognitive demand from a working memory task: An on-road study across three age groups. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 54, 3, 396–412.

B. Mehler, B. Reimer, J. F. Coughlin, and J. A. Dusek. 2009. Impact of incremental increases in cognitive workload on physiological arousal and performance in young adult drivers. *Transportation Research Record: Journal of the Transportation Research Board* 2138, 2, 6–12.

E. Molteni, G. Baselli, A. M. Bianchi, M. Caffini, D. Contini, L. Spinelli, A. Torricelli, S. Cerutti, and R. Cubeddu. 2009. Frontal brain activation during a working memory task: a time-domain fNIRS study. *Progress in Biomedical Optics and Imaging* 10, 1.

G. Mueller-Putz, R. Scherer, C. Brunner, R. Leeb, and G. Pfurtscheller. 2008. Better than random: A closer look on BCI results. *International Journal of Bioelectromagnetism* 10, 52–55.

L. E. Nacke, M. Kalyn, C. Lough, and R. L. Mandryk. 2011. Biofeedback game design: Using direct and indirect physiological control to enhance game interaction. In *Proceedings of the ACM CHI 2011.* ACM.

I. Nouretdinov, S. G. Costafreda, A. Gammerman, A. Chervonenkis, V. Vovk, V. Vapnik, and C. H. Fu. 2011. Machine learning classification with confidence: application of transductive conformal predictors to MRI-based diagnostic and prognostic markers in depression. *Neuroimage* 56, 2, 809–813.

A. M. Owen, K. M. McMillan, A. R. Laird, and E. Bullmore. 2005. N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies. *Human Brain Mapping* 25, 1, 46–59.

E. M. Peck, B. F. Yuksel, L. Harrison, A. Ottley, and R. Chang. 2012. Towards a 3-dimensional model of individual cognitive differences. In *BELIV 2012: Beyond Time and Errors: Novel Evaluation Methods for Visualization, 2012*.

E. M. Peck, B. F. Yuksel, A. Ottley, R. J. K. Jacob, and R. Chang. 2013. Using fNIRS brain sensing to evaluate information visualization interfaces. In *Proceedings of the ACM CHI 2013.* ACM.

E. M. Peck, D. Afergan, and R. J. K. Jacob, 2013. Investigation of fNIRS brain sensing as input to information filtering systems. In *Augmented Human 2013*.

L. J. Prinzel, F. G. Freeman, M. W. Scerbo, P. J. Mikulka, and A. T. Pope. 2003. Effects of a psychophysiological system for adaptive automation on performance, workload, and the event-related potential P300 component. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 45, 4, 601–613.

N. Ramnani and A. M. Owen. 2004. Anterior prefrontal cortex: insights into function from anatomy and neuroimaging. *Nature Reviews Neuroscience* 5, 3, 184.

M. Rötting, T. Zander, S. Trösterer, and J. Dzaack. 2009. Implicit interaction in multimodal human-machine systems. In *Industrial Engineering and Ergonomics*, Springer, Berlin, 523–536.

G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw. 2004. BCI2000: A general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering* 51, 6, 1034.

A. Schmidt. 2000. Implicit human computer interaction through context. *Personal Technologies* 4, 2, 191–199.

P. Shenoy and D. S. Tan. 2008. Human-aided computing: utilizing implicit human processing to classify images. In *Proceedings of the ACM CHI 2008.* ACM.

T. Shimizu, S. Hirose, H. Obara, K. Yanagisawa, H. Tsunashima, Y. Marumo, and M. Taira. 2009. Measurement of frontal cortex brain activity attributable to the driving workload and increased attention. *SAE International Journal of Passenger Cars-Mechanical Systems* 2, 1, 736–744.

R. Sitaram, H. Zhang, C. Guan, M. Thulasidas, Y. Hoshi, A. Ishikawa, K. Shimizu, N. Birbaumer. 2007. Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain-computer interface. *Neuroimage* 34, 4, 1416–1427.

E. T. Solovey, A. Girouard, K. Chauncey, L. Hirshfield, A. Sassaroli, F. Zheng, S. Fantini, and R. Jacob. 2009. Using fNIRS brain sensing in realistic HCI settings: Experiments and guidelines. In *Proceedings of the ACM UIST 2009*.

E. T. Solovey, F. Lalooses, and K. Chauncey, et al. 2011. Sensing cognitive multitasking for a brain-based adaptive user interface. In *Proceedings of the ACM CHI*, ACM.

E. T. Solovey, P. Schermerhorn, M. Scheutz, A. Sassaroli, S. Fantini, and R. J. K. Jacob. 2012. Brainput: Enhancing interactive systems with streaming fNIRS brain input. In *Proceedings of the ACM CHI*. ACM.

T. Starner, B. Schiele, and A. Pentland. 1998. Visual contextual awareness in wearable computing. In *Proceedings of the ISWC.*

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 *Proceedings of the ACM IUI 2013.* ACM.

M. St. John, D. A. Kobus, J. G. Morrison, and D. Schmorrow. 2004. Overview of the DARPA augmented cognition technical integration experiment. *International Journal of Human-Computer Interaction* 17, 2, 131–149.

S. Tak and J. Ye. 2014. Statistical analysis of fNIRS data: A comprehensive review. *NeuroImage* 85, 1, 72–91.

D. Tan and A. Nijholt. 2010. Brain-computer interfaces and human-computer interaction. In *Brain-Computer Interaction: Applying our Minds to Human-Computer Interaction*, D. S. Tan and A. Nijholt (Eds.). Springer, London, 3–19.

J. J. Thomas and K. A. Cook, eds. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics.* IEEE CS Press. http://nvac.pnl.gov/agenda.stm.

D. Toker, C. Conati, G. Carenini, and M. Haraty. 2012. Towards adaptive information visualization: On the influence of user characteristics. In *User Modeling, Adaptation, and Personalization.*

C. T. Vi and S. Subramanian. 2012. Detecting error-related negativity for interaction design. In *Proceedings of the CHI 2012.* ACM.

A. Villringer, J. Planck, C. Hock, L. Schleinkofer, and U. Dirnagl. 1993. Near infrared spectroscopy (NIRS): A new tool to study hemodynamic changes during activation of brain function in human adults. *Neuroscience Letters* 154, 101–104.

G. F. Wilson and C. A.Russell. 2007. Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 49(6), 1005–1018.

M. Wolf, U. Wolf, J. H. Choi, R. Gupta, L. P. Safonova, L. A. Paunescu, A. Michalos, and E. Gratton. 2002. Functional frequency-domain near-infrared spectroscopy detects fast neuronal signal in the motor cortex. *Neuroimage* 17, 1868–1875.

J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris. 1991. An EEG-based brain-computer interface for cursor control. *Electroencephalography and Clinical Neurophysiology* 78, 3, 252.

T. O. Zander, C. Kothe, S. Welke, and M. Roetting. 2008. Enhancing human-machine systems with secondary input from passive brain-computer interfaces. In *Proceedings of the 4th International BCI Workshop & Training Cours*e.

T. O. Zander, J. Brönstrup, R. Lorenz, and L. R. Krol. 2014. Towards BCI-based implicit control in human–computer interaction. In *Advances in Physiological Computing.* Springer, London, 67–90.

T. O. Zander and S. Jatzev. 2012. Context-aware brain–computer interfaces: Exploring the information space of user, technical system and environment. *Journal of Neural Engineering* 9, 1.

T. O. Zander and C. Kothe. 2011. Towards passive brain–computer interfaces: Applying brain–computer interface technology to human–machine systems in general. *Journal of Neural Engineering* 8, 2.

T. O. Zander, C. Kothe, S. Jatzev, and M. Gaertner. 2010. Enhancing human-computer interaction with input from active and passive brain-computer interfaces. In *Brain-Computer Interfaces*, Springer, London, 181–199.

Received January 2014; revised October 2014; accepted October 2014.