Face Interface: A Methodology for Experiential Learning of Input Modalities

by

Jon William Wetzel

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2007

Author ....
Department of Electrical Engineering and Computer Science
May 25, 2007

Certified by ....
Edwin J. Selker
Associate Professor
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Face Interface: A Methodology for Experiential Learning of Input Modalities

by

Jon William Wetzel

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2007, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Computer Science and Engineering

Abstract

This thesis demonstrates that creating a system with a visual representation of the face which mirrors the user's facial gestures appears to solve problems in teaching a user to use the new input affordances of face-based interfaces. From experiences with the Attentive Interaction Design Toolkit, "Attention Meter," a methodology for helping the user design their own interactions with an unfamiliar input modality is created. This methodology is then applied to face-based interfaces, through a program called "Face Interface." The subsequent evaluations of Face Interface through its revisions show that it is ready to successfully apply the methodology through further experimentation.

Thesis Supervisor: Edwin J. Selker
Title: Associate Professor

Acknowledgments

Thanks to

- Professor Ted Selker for his help and guidance, and his role in introducing me to the world of academic research.
- Jackie Lee, co-creator of the Attentive Interaction Design Toolkit, Attention Meter.
- Heymian Wong (UROP) for all her help improving Attention Meter for Face Interface.
- Ana Pandalfo for her help in preparing the experiment.
- Everyone in the MIT Media Lab "Context Aware Computing" group, for all their input, support, and for being fun to work with these three and a half years.
- Jon and Sylvia Wetzel for their eternal love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Vision-based systems and even eye-tracking have been researched for decades[1]. The advent of the use of the AdaBoost algorithm and hierarchical classifiers has made it possible to detect and track larger anatomical features, particularly the face[2]. While the idea of building cameras to track anatomical features is not new[3], the use of video input in user interfaces has been relatively uncommon. With open source libraries capable of tracking faces at the frame-rate of a standard web-cam[4], the age of the face may be in sight.

Face Interface is a program for controlling a computer with a face. It allows the user to define a mapping between facial gestures, such as nodding and winking and actions at the OS level, such as keyboard input and executing programs. To help the user quickly become proficient at using their facial gestures as an input, Face Interface includes an avatar which reflects the user's successful gestures like a mirror. This work of this thesis shows this is a working implementation of a methodology which solves problems of teaching the user to find use for new input affordances.

1.1 Related Work

Commercial face-based mouse pointers, such as Natural Point's TrackIR[5], and Origin Instrument's HeadMouse[6] use the tracking of a single point on the face to move the mouse pointer. When coupled with an on-screen keyboard these can allow full control of the computer using only the face as an input. However, completing the basic mouse operations such as double clicking, right clicking, and drag and drop require a separate application[7] which must be triggered by on-screen keyboard. This slows down the user since they must move to the on-screen keyboard in between different mouse movements; and also, this adds extra modes to the interface, which are not always desirable. Face Interface does not strictly emulate the mouse through the position of the nose or other central facial feature, but instead uses the facial gestures from Attention Meter as its input.

Use of the language of facial gestures in interfaces has been demonstrated; researchers have produced mouse emulation systems exist which can use nodding and shaking for different kinds of mouse clicks[8], or use the shape of the mouth to perform clicking and dragging operations[9]. One recent system used head gestures to let the head tracking system know when it is not working[10]. Still other systems have tried using nodding to mean "OK" and shaking to mean "Cancel"[11, 12] in the many pop-up dialogs found in today's interfaces. However, these emulators are not affected by the context of the application they are in. The design of Face Interface makes it capable of matching gestures (nodding, opening the mouth) to controls on an app-by-app basis. For example, it has been conjectured that nodding is good for scrolling while reading documents[12], but in a media player one might rather have nodding move to the next track, or play a selected song rather than making the play list scroll. From our experiences with Attention Meter have produced our own set of observations about the use of these gestures (see Table 2.1). For instance, we have found that nodding is not good for selection activities (see section 2.2).

For many of the aforementioned hands-free systems which use the face as a pointer, Face Interface could improve this by eliminating some repetitive on-screen keyboard movements. Or perhaps, in the future, mouse pointer control can be added to Face Interface, allowing it to work on it's own to serve those unable to use the mouse and keyboard as existing systems do now.

For the able-bodied, Face Interface needs not to completely replace existing input methods like the keyboard and mouse; it allows users to use their face to augment their interaction as they desire. Already commercial systems have found uses for face control in games[5]. By designing Face Interface with the possibility to keep the current application in mind, we create a platform for use of facial gestures in any context.

1.2 Overview of the Thesis

This work of this thesis shows that providing a visual, experiential learning[13] environment solves problems of teaching the user to use the new input affordances of face-based interfaces. It begins with a chapter on the technological foundation for Face Interface, Attention Meter (chapter 2). In this chapter Attention Meter is reviewed in its use in five projects. The lessons learned in this review motivate our the methodology behind Face Interface.

Chapter 3, Face Interface, begins with a description of the methodology for promoting experiential learning in design toolkits. Then, the design and implementation of Face Interface is explained in detail. All versions of the user interface are presented and evaluated.

Chapter (4), proposes a user study to prove the methodology and guide future directions for Face Interface.

Finally, Chapter 5, Conclusion, includes a discussion of the evaluation and a outlining of future work.

Chapter 2

Attention Meter: An Attentive Interaction Design Toolkit

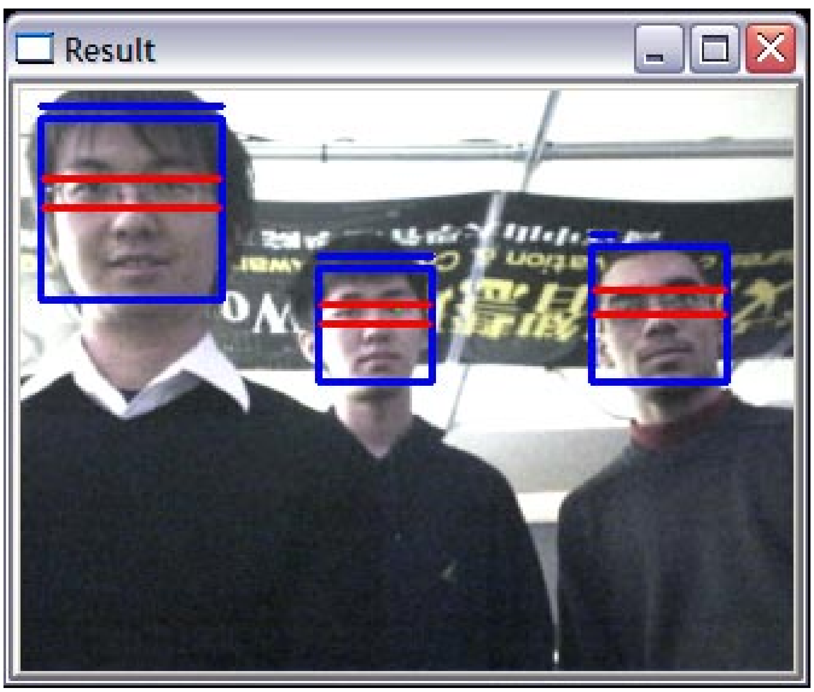

Figure 2-1: Attention Meter tracks multiple faces and computes an attention score for each face and the entire screen.

Face Interface extends the usability of our previous work, the Attentive Interaction Design Toolkit, also known as Attention Meter[14]. It was co-developed by Chia-Hsun Jackie Lee and Jon Wetzel in the Fall of 2005. Attention Meter is both the technology and motivation behind Face Interface, and so its history is included here. The original goal of Attention Meter was to create a toolkit for designers to build attention-aware face-based interfaces using the most mainstream vision hardware available, a webcam.

2.1 Implementation

Attention Meter is able to track the size and position of multiple faces, detect whether the face is moving or still, detect gestures of nodding and shaking, detect opening and closing of the eyes and mouth, and compute attention scores for each individual face and the scene as a whole. This data can be output graphically to windows on the screen (Fig. 2-1), or as text to a file on disk or a network stream. To make Attention Meter usable as a toolkit, this text file output was made easily readable by Adobe Flash[15]. Attention Meter is made so a designer may write a program in any language they desire to use the output.

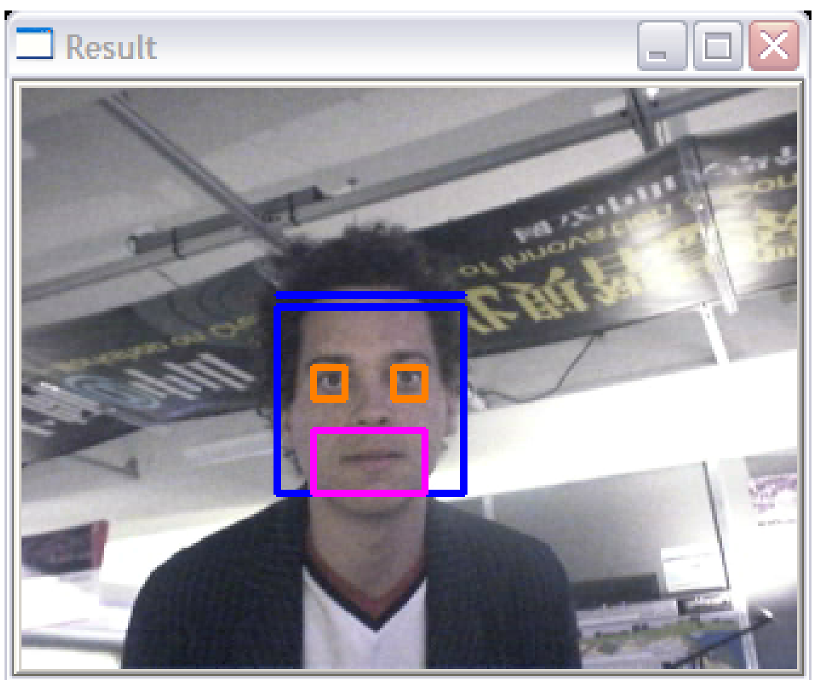

Figure 2-2: The "Result" window captures a face at full attention, mouth closed.

The "Result" window (Fig. 2-2)of Attention Meter displays the output in a graphical way. Each feature is drawn onto the camera image, with boxes around the face, mouth and eyes that change color to indicate the states such as nodding or mouth open. A line above each face shows its individual attention score, and the group attention is shown as a large green bar in another window. Providing a way to view and understand the output in realtime was a key part of making a platform that even novice programmers could use to build engaging interactive environments[14].

2.2 A Review of Attention Meter

2.2.1 Asian Reality Design Workshop

At the Asian Reality 2005 design workshop[16] in Taiwan, architecture students learned to use Attention Meter under the guidance of Chia-Hsun Jackie Lee, an experienced interaction designer. Using Adobe Flash[15] with Attention Meter, the groups created interactive spaces. The students divided into groups and produced three projects over the course of two days: Taiker KTV (karaoke experience), Scream Market (nightmarket simulation), and Eat-Eat-Eat (game set in a market)[14].

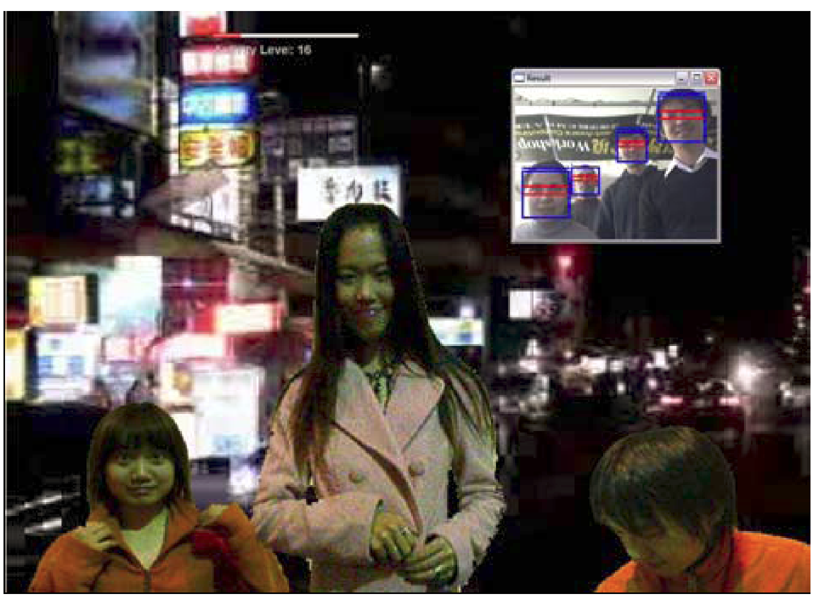

Figure 2-3: The ScreamMarket simulated night market.

Figure 2-4: The TaiKer-KTV interactive karakoe environment.

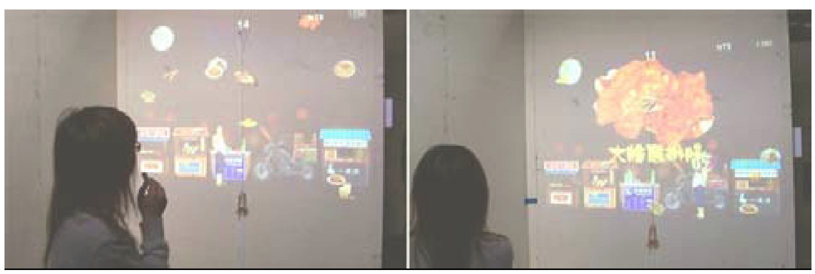

Figure 2-5: The Eat-Eat-Eat game.

When Attention Meter was launched, our focus was more on the "Attentive Interface" idea. The projects, however, highlighted different aspects of Attention Meter. The "Scream Market" group focused on the attention measuring abilities of Attention Meter and augmented them using a microphone. They created an interactive display portraying a nightmarket scene (Fig. 2-3), which became more lively based on the number of people watching and how loud they shouted.

In contrast, the other two projects did not focus the attention factor. "Taiker-KTV" made the karaoke video more exciting as more people shook their heads (Fig. 2-4), and "Eat-Eat-Eat" used face-tracking to move an avatar on the screen and catch food as it fell from the sky (Fig. 2-5). Though the toolkit was an "attention meter", two of the three projects did not use the attention score. When creating face-based interfaces, the designers found intuitive ways to apply gestures to actions. This motivated us to pursue face-based interfaces and eventually became one of the core ideas of Face Interface.

2.2.2 PlaceMap Controller

Figure 2-6: CampusMap, the MIT campus version of PlaceMap

Eager to experiment more with face-based interfaces, the next application of Attention Meter used it to perform zoom, pan, and select operations in "CampusMap" (Fig. 2-6). CampusMap was the MIT version of "PlaceMap," a interactive, web-based, social networking site complete with instant messaging and the ability to see your buddies on a map[17]. The goal of this project was to create a direct manipulation[18] interface based off of Attention Meter's output data.

Unlike participants, who were guided by someone experienced in creating interactive spaces with the face-based modality, the designer of the PlaceMap controller was only given the Attention Meter source code to gain knowledge of how each feature worked. In the resulting design panning the map was done by moving one's head in one of the four cardinal directions. Zooming in and out was controlled by moving one's head closer and farther away, respectively. Finally, nodding was used to select an object.

Nodding had been successfully used in the "Taiker-KTV" karaoke environment. In the context of group karaoke, nodding continuously was a social gesture used by the user to indicate they were having a good time, and the environments engaged them accordingly[14]. When the PlaceMap controller interface triggered the episodic action of selection with the continuous action of nodding, an impedance match problem occurred. Users had to nod continuously to select their target. This problem was compounded by the fact that nodding disrupts the visual focus of the eye, which they were using to aim in the first place.

From this example we observed that simply describing what Attention Meter is outputting does not convey the affordances to the designer. Designers need to understand both how gestures are made and how they provide affordances before applying them. In the Taiwan workshop, students played with Attention Meter before coming up with their interface designs. By watching themselves in the result window, these designers got a feel for how the gestures worked easily. This was a factor in the successful creation of the projects discussed previously.

Another factor to be noted is that these designers were architecture students, trained in design. More importantly, they were also guided by the more experienced Chia-Hsun Jackie Lee, who instructed them to keep the ability to observe people's actions in mind in their designs. Using his knowledge of what makes an immersing interaction, he helped them to find comfortable mapping. Contrast this with the designer for the PlaceMap controller, who was an engineering student without training in user interfaces and was designing the project to be controlled by people, not to react to what they might do naturally. Attention Meter has no way of directly conveying any principles of the affordances to the designer using it.

This showed us that in order to be effectively used by the widest audience, a toolkit for building interfaces should be able to convey information about the affordances as well. We compiled the information from previous projects for future use, and the result of our analysis of the use of facial gestures as affordances can be found in table

2.1. In the future, Face Interface may convey this information more directly (see section 5.2.2).

Input Signal	Uses
Horizontal/Vertical Position	Direct manipulation selection by position (Eat-Eat-Eat)
Distance from Camera	Selection, activation (Attentional Vending)
Nodding/Shaking, Mouth Opening/Closing	Set and hold a state/repeat an action until gesture ends (TaiKer KTV)
Blink Rate/Winking	Toggle state, "clicking" (Eye Bed[19])
Face Appearing/Disappearing	Set state based on number of faces. (Scream Market)

Table 2.1: Analysis of the use of affordances in Attention Meter

2.2.3 Attentional Vending Machine

Figure 2-7: The Attentional Vending Machine.

The goal of the Attentional Vending machine (Fig. 2-7) was to demonstrate the value of greetings in interactive systems. Attention Meter added attention-awareness to a Pepsi machine so that it could use sound and video to introduce itself to those who approached and not annoy people just walking by. This makes for a simple, effective demonstration, and made the Attentional Vending machine a popular exhibit during sponsor open houses.

The Attentional Vending project successfully used Attention Meter with only minor adaptations. When the designers of began, adverse lighting and background conditions added false positives to the face tracker. However, they quickly found that using some simple algorithmic techniques they could filter out the errant data and create a robust experience. Once this was accomplished, scenarios that varied based on the number of people worked without further problems. The "Result" window of Attention Meter, which designers use to understand and debug their projects, has no such post-processing. This feature was added to Face Interface as well.

2.2.4 Lessons from Attention Meter

In summary, we have learned three important lessons from the Attention Meter projects. From the Asian Reality Design Workshop[16] we learned that a realtime graphical representation of the input makes the technology more accessible. From the PlaceMap controller project we learned that the interface should be aware of and try to convey the affordances of facial gestures. Finally, from the Attentional Vending Machine we learned that adding some post-processing to the output can help users avoid some of the inherent pitfalls of the technology (in this case, lighting and false positives in the face tracker).

Chapter 3

Face Interface: A System for Experiential Learning of the Facial Input Modality

In this chapter, we describe a design methodology for building interface design toolkits capable of teaching designers input modalities. Then, we describe how it is applied to face-based interfaces through the example of Face Interface.

3.1 Methodology

From the experience with Attention Meter, we learned three principles:

- A real-time graphical representation of the input allows user to gain understanding of the input device through experimentation.
- The affordances of the input modality should be conveyed to the designer, as many are not trained to recognize them.
- Post-process the input as much as possible, not to limit the user's ability to make gestures, but to lower the amount of caused by errors caused by inherit difficulties in the input technique.

Face Interface is a non-programmer-friendly extension of Attention Meter, which allows users to map their facial gestures to keystrokes and/or file launching. Simply put, the user can control their computer with their facial gestures. The next section describes the specifics of how Face Interface is is implemented to meet the above three principles.

3.2 Implementation: Face Interface

At the core of Face Interface is Attention Meter, the Attentive Interaction Design Toolkit. The Face Interface program consists of an enhanced version of Attention Meter, representations for facial gestures, automated user actions, and rules (mapping the gestures to the actions), and a graphical user interface for operating all three. The design has progressed under the Industrial Design Intelligence paradigm, with attention given to ascetics, engineering, and evaluation at every stage. The various versions are discussed in Section 3.2.3.

This section begins with a description of the enhancements made to Attention Meter, continues with an overview of the Face Interface architecture, and then recounts the design progression the graphical user interface. The hardware platform is described in the final subsection.

3.2.1 Attention Meter Enhancements

Detecting Mouth Open/Close

Opening the mouth is a very easy gesture for the user to make and does not require moving the head. In older versions of Attention Meter, the open mouth and smile detection feature lacked robustness, so it was replaced with a more accurate mouth change detection. This enabled one to guess when the person is communicating. However, this is not very useful as an affordance, since it requires the user to hold their mouth absolutely still for large periods of time. Thus, we decided to implement new functionality to detect when the mouth was open.

The old mouth analysis algorithm was dropped because of its robustness issues, so a new algorithm was implemented as follows:

- Find the mouth region (fixed portion of face) and convert it to gray scale.

- Take the average of this region. Divide it by some constant to get the threshold for a "dark" pixel.

- Threshold the gray scale region to convert it to black and white. Divide the number of the "dark" pixels by the total number in the mouth region to get the ratio of dark to all.

- The ratio will be high when the mouth is open (from dark areas inside the mouth) and low when it is closed. The threshold, unfortunately, is highly dependent on the lighting, and must be calibrated. A function is provided to do this.

Detecting Winks

Winking is another gesture which is fast and does not require moving the entire head. Attention Meter had a system in place to detect blinks, making implementing wink detection easy. We thought that it would also be an easy gesture that people do naturally in social situations. While it turned out that it's not always the case that a user can successfully wink,

In order to improve wink detection, a new algorithm was implemented with which was similar to the mouth one above, except instead of using the ratio, the pixel count of the regions were compared to each other. Despite several combinations of smoothing and iterative thresholding, in the end nothing was more robust than our previous technique of finding the edge where the cornea meets the white of the eye.

The main improvement here was to distinguish between the two eyes, changing the face from merely having a "blink" attribute to actually keeping track of whether the eyes are open independently. This made detecting winks possible. The threshold required to recognize a cornea-white edge pixel was also change to be able to be set through the Attention Meter, so this feature can be calibrated as well.

Single-Face Tracking

Because most desktop applications are built for use by a single user with a single mouse and keyboard, using multiple faces as input was a poor fit for Face Interface. Instead of tracking all the faces in the image, the Attention Meter in Face Interface locks onto a single face of interest. When choosing which face to track initially, it chooses the face closest to the center, since it is likely to be the computer's primary user. We have not given up completely on multiple-face tracking and hope to include it in Face Interface in the future (see section 5.2.1).

3.2.2 Face Interface Architecture

Overview

In code, Face Interface is referred to by it's original name, User-Defined Facial Interaction (UDFI). UDFI can be divided into four distinct parts.

- The enhanced version of Attention Meter (section 3.2.1)
- The interface between Attention Meter and Face Interface, UDFIAMI
- The representations of gestures, actions, and the rules mapping between the two.
- The graphical user interface (GUI), which implements the first two points of the methodology in section 3.1.

While Attention Meter is written in (native) C++, Face Interface is written using Microsoft Visual C++ .Net 3.0[20]. This was done to simplify the process of building the GUI. Face Interface is a multi-threaded program. When the program begins, it starts in the GUI code and launches the main window. Attention Meter is immediately launched in another thread. When Face Interface is ready to use Attention Meter output, it launches a third thread for UDFIAMI. This thread can be deactivated when not in use, but the Attention Meter thread is always running.

Representations

The UDFI data classes represent facial gestures, actions that may result from those gestures, and the rules that map those gestures to those actions. All of these classes are "managed" in the Visual C++ .NET sense, supporting automatic garbage collection and serialization. Saving is done automatically by serializing the entire list of rule objects and saving it to disk each time the user adds, edits, or removes a rule.

`UdfiEvent (UdfiEvent.h, UdfiEvent.cpp)`

The UdfiEvent class represents events from Attention Meter-facial gestures. The class contains an enum representing all the different types of events (nod start, shake stop, etc.). A complete list of the events can be found later in table 3.1. The delegate method, "UdfiEventHandler" is also defined here. Other objects wishing to listen for UdfiEvents must add an instance of this delegate to their own event handling method. For the events involving faces entering/exiting regions of the view, a helper class called UdfiRegion was created. This class includes a method for determining if a point is inside a region.

`UdfiAction (UdfiAction.h, UdfiAction.cpp)`

The UdfiAction class is an abstract class for representing any action that might be triggered by an event. All UdfiActions have a Run method and a Name property. The specifics of the implementation are flexible from there. So far there are two direct subclasses of UdfiAction: UdfiTypingAction and UdfiRunAction. UdfiTypingAction uses the Microsoft .NET "SendKeys" class to send keyboard output. Its constructor takes in a string in the SendKeys format, and when Run is called that string is passed to SendKeys. UdfiRunAction uses the .NET "Process" class to launch a file specified at construction. If the file is an executable it is run directly; if it is a document capable of being read or opened with another program, the OS chooses the default application to open it with and does so.

`UdfiTrigger (UdfiTrigger.h, UdfiTrigger.cpp)`

The UdfiTrigger class represents a rule in Face Interface. UdfiTrigger objects contain a sequence of UdfiEvents and a sequence of UdfiActions. An UdfiTrigger begins listening for UdfiEvents, looking for the one which will match the first UdfiEvent in its sequence. When it finds a match, it begins looking for the next event. If the event is a repeat of the prior event, it continues waiting for the next one. If an incoming event does not match the prior event or the desired next event, the trigger resets, and begins waiting for the first element of the sequence. When the last event in the sequence is matched, the UdfiTrigger then calls Run for each UdfiActions in it's action sequence. This class is also where application-awareness will be implemented (see section 5.2.4).

Interface between Attention Meter and Face Interface

UDFI follows the event/listener design pattern, using Visual C++ Managed events. UDFIAMI polls Attention Meter for the frame number, and when a new frame completes UDFIAMI reads the data on the single-tracked face and raises any appropriate events listed in Table 3.1.

Events used by Face Interface as input
Face Appeared/Disappeared
Face Entered/Exited Region
Mouth Open/Closed
Left/Right Wink Started/Wink Stopped
Shake Start/Stop
Nod Start/Stop

Table 3.1: The list of events Face Interface may react to. A region is an area of camera view.

Event data is passed using UdfiEvent objects. UdfiTriggers and any GUI classes which need to know subscribe to these events.

Following the third point in our methodology (section 3.1, UDFIAMI applies some simple wait logic to the Attention Meter output to make the results less noisy. For instance, it waits until a face has reached a certain attention level before raising the Face Appeared event, to avoid the false positives that more interesting backgrounds are prone to expressing. Similar noise reduction is present for nodding, shaking, and winking in the form of short time delays.

UDFIAMI is defined in the UdfiAMI class in “UdfiAMI.h” and “UdfiAMI.cpp”. These files also include the event definition.

3.2.3 Face Interface Graphical User Interface

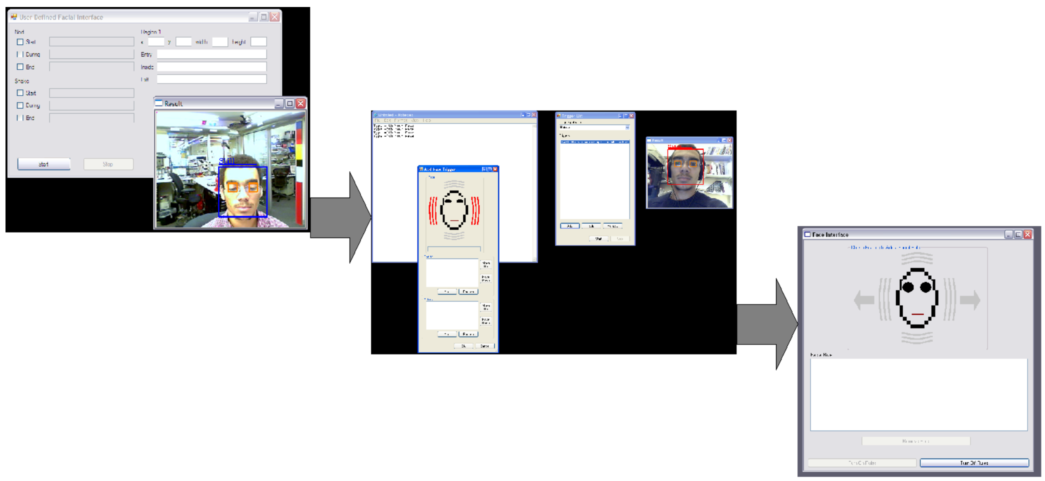

Figure 3-1: A side by side comparison of the three versions the of Face Interface UI.

Lessons have been learned from the progressively developed GUI for Face Interface. From the experience of using the first, we learned how a visual avatar might promote experiential learning. From the second we gained insight on how to focus the user experience on that learning, by condensing the user interface into a single window. In this section, each version is described and evaluated.

Concept Version

The GUI of Face Interface went through three versions. The first, “UDFI” (User-Defined Facial Interaction) was a proof of concept for mapping facial gestures to text input. It relied on the “Result” window from the original Attention Meter for visual

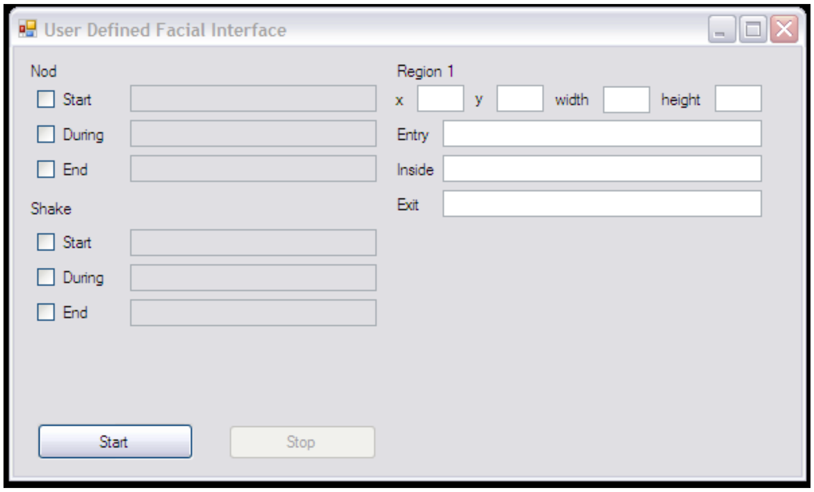

Figure 3-2: Face Interface proof of concept model (v1, pre-avatar).

feedback. The evaluation of this version is discussed in section 3.2.3, but from it we learned that a visual representation of the face is helpful for building associations.

The interface contains a text box for each gesture, and when the "Start" button is pressed, the text boxes are frozen, and their contents are input to the keyboard every time their respective gesture is detected. This simulated keyboard input is done using the Microsoft "SendKeys" class. When the user presses "Stop" simulated keyboard input stops, and become the text boxes become editable again.

Evaluation of Concept Version

The concept version was critiqued by other members of the Context Aware Computing group at the MIT Media Lab. There was some confusion as to what to do when first presented with the set of text boxes. One suggestion was to add a graphic of a face, so that users could more easily relate the gestures to the text they were entering. Then came the idea to use this graphic as a replacement of the result window, which led directly to the avatar (Fig. A-2) found in the subsequent alpha version.

Another suggestion was to remove the limitation of having only one mapping per gesture, allowing users to create a language of gestures if they wanted and to support multiple actions as results. The result was to abstract the mappings into rules, and allow the user to create rules out of any number of gestures and actions.

Finally, this version did not support easy experiential learning because it relied on the Attention Meter "Result" window to give feedback to the user. The color-coded squares which displayed the output in this window were not quickly learned by those in the group not familiar with Attention Meter, which limited their experimentation with it.

Alpha Version

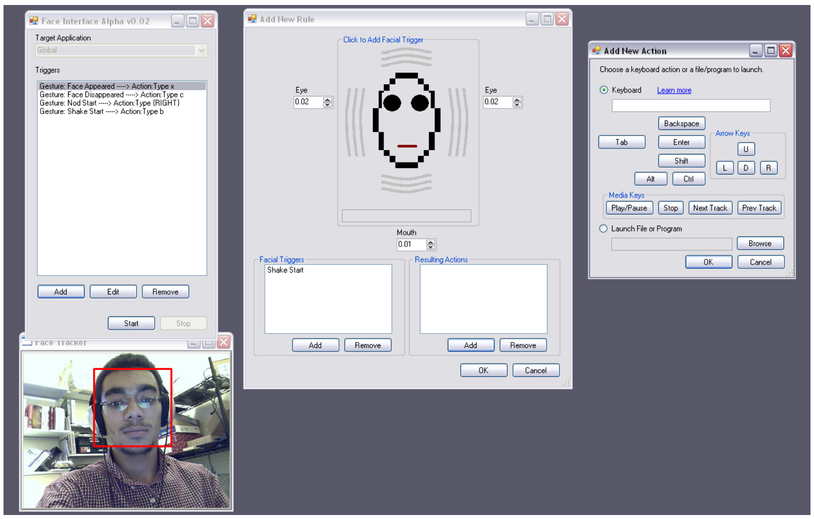

Figure 3-3: The alpha version of Face Interface, choosing an action for a new rule.

The alpha version (Fig. 3-3) was more generalized, allowing the user to create rules mapping sequences of facial gestures to sequences of actions (automated keyboard input and running programs). It also addressed the experiential learning problem by including and an avatar (Fig. A-2) which reflected the user's facial gestures like a mirror, in real time. Now as the user creates their rules, the system allows the user to create a natural mapping[21] between the visible features of the face and the actions they choose. In order to better understand how the effect of the avatar on users proficiency with performing the gestures using them as affordances, a experiment was proposed (section 4).

Another feature added in this version was the automatic saving of the user's rules. Every time the user changed the rules they were copied to disk, and when the program is first launched the saved rules were loaded.

Evaluation of Alpha Version

The alpha version of Face Interface has been field tested with demonstrations to sponsors. At the 2007 Digital Life Consortium[22] and 2007 Things That Think Consortium [23] events included an open house of the Context Aware Computing group. Here, Face Interface was demonstrated by controlling a media player. Initially, rules were made so that when a person looked at the screen the video played and when no one was watching the video would stop. Once the idea was explained, sponsors began making their own suggestions, such as using shaking to change movies and nodding to skip forward. These were from the meanings they inferred from the gestures: "I don't like this movie" for shaking and "I get the point, let's keep going" for nodding. Not many suggested uses for winking, partially because not all of them could wink. Overall most sponsors were impressed and took cards with the website for Attention Meter and Face Interface for future reference. Attendees of both consortiums took DVDs containing the software.

While the interface was well received by sponsors and included all of the suggested working features, there were still improvements to be made. Further evaluation by the Context Aware Computing group found that the many windows (Fig. 3-3) were distracting, leading to the experimental version which directs the users focus carefully, using only a single small pop-up window for keystroke entry (Fig. A-5).

It was also observed that neither sponsors nor group members tried using multiple gestures in their rules. This led to the simplification of one gesture per rule, eliminating much of the UI overhead.

Experiment Version

However, the alpha version's UI was found to be needlessly complex with its multi-leveled dialog boxes, so a third, more streamlined version was created. To simplify the experiment, the program was changed to create rules mapping one gesture to one Google Maps operation (which are backed by keyboard shortcuts). The avatar was

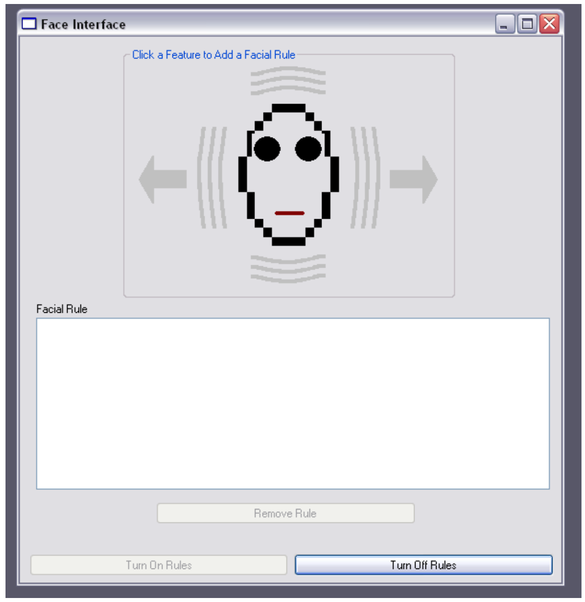

Figure 3-4: Main window of Face Interface, experiment version

moved to the main window and the flow of the program changed to center around it.

Evaluation of Experimental Version

One problem raised by the new GUI is the avatar is hard to figure out on first use. We have found that for first time users, despite the explanatory text at the top, it is not obvious that you should click on the portions of the face. To address this, buttons will be used to create the avatar, instantly allowing the users to know they are supposed to click on the features.

The second problem with the experimental version is it suffers from a loss of generality, since the user an only create rules which map one gesture to one keypress. However we felt that keeping the lexicon of gesture of language small was best for the experiment, since the time to learn is limited. The experimental version of Face Interface will be further evaluated through the experiment in the following section. The problem of generality, and others will be corrected in the next release version. Other proposed changes can be found in the future work section (section **??**)

3.2.4 Platform

Attention Meter is written in C++ (the primary language of OpenCV is C). In order to easily add a complex GUI while remaining in C++ development of Face Interface was done in C++ using Microsoft Visual Studio .NET. The ability to use the Visual Studio designer and managed code to ease most of the GUI creating process is offset by the fact that the user must install the visual C++ redistributable (included with the release) and the .NET framework, which limits Face Interface to the Windows platform. In the future it may be ported to a more cross-platform GUI library such as GTK+.

Face Interface was developed and run on a 1.66 GHz Intel Core 2 processor. Testing on single-core systems was limited, but the multi-threaded nature of the program leads it to run more smoothly on multi-core processors. RAM usage maxed out around 50MB, making memory a non-issue on current hardware. The camera used was a Logitech QuickCam Pro 5000[24] with RightLight™automatic light adjustment. While the Logitech camera system does contain some face tracking ability, this was not used by Attention Meter or Face Interface (it appears to be used only internally by the camera and its driver software).

Chapter 4

Face Interface User Study

The goal of this research is to demonstrate that our design methodology will improve users ability to create a interface in an unfamiliar modality using Face Interface, which is based on our previously mentioned methodology (section 3.1). This will be done by performing a user study with the experimental version (section 3.2.3) of Face Interface, which provides an interactive avatar in the main window. This chapter describes our evaluations so far, the design of the experiment, and several directions for future research.

The purpose of this experiment is to demonstrate that the methodology (section 3.1) will help users more quickly learn to design and use interactions using an unfamiliar input modality. This will be done using Face Interface and the face-based input modality provided by the gesture recognition of Attention Meter.

Our method is to use take a group of first-time Face Interface users and give them a task to perform using a simplified version of Face Interface. Half of the users will have a Face Interface program with the avatar feature, the other half will not. The hypotheses is that users with the avatar feature will find the program easier to use. Ease of use will be determined by the speed it which it takes them to complete the task, how often they change their rules, and their subject feedback on a questionnaire.

4.0.5 Structure of Experiment

The experiment begins with a consent form (appendix figure B-1) and intro survey (appendix figure B-5), asking for background information on their education, occupation, and how long they have used a computer. Once the survey is done the subject is given the first instruction page (appendix B-6) and given an unannounced five minutes to familiarize themselves with the interface. When this period is up, the subject is given task sheet (appendix figure B-7) and their time to complete it is recorded. This stage may last up to 15 minutes. Once complete, the user is given the exit survey (appendix figure B-8) and their compensation.

4.0.6 Experimental Version of Face Interface

To test the benefits of the avatar for promoting experiential learning, two versions of the experimental versions will be used. Both programs have one primary window with the avatar at the top and the list of rules at the bottom, and in both versions the avatar is present and clickable, but for half of the users the avatar remains frozen, regardless of what the user does with their face. Mousing over feature of the avatar explain which gesture they correspond to. Clicking on features of the avatar brings up a pop up window which asks for a keystroke to match to that gesture. If the user selects a keystroke and presses Ok, the rule is created.

Rules can be removed by selecting them and clicking the "Remove" button. Rules are active by default. If the user wants to stop their rules they can click "Turn Off Rules" to do so. They can then reactivate them by pressing "Turn On Rules". The software will be instrumented to log the time of all rule changes, allowing us to review what the user did during their time with the program. The "ease of use" measurement will depend on the time and number of changes as well as the user's self-reported difficulty.

Chapter 5

Conclusion

5.1 Discussion

Attention Meter succeeded in allowing designers to create engaging interactive spaces, however they worked most effectively with guidance with someone more experienced with the facial input modality. The goal of Face Interface is to discover a way to make learning new input modalities easier. The principles in our methodology could apply to haptics, body gestures, facial expressions or even possibly speech or other new modalities, but we found the face to be the most relevant example.

Given the information-laden nature of the face, the unobtrusiveness of vision based interfaces, and the proliferation of web cams, face-based interfaces may someday be as common as the mouse. Attention Meter has shown that even novice programmers can create new interactions using their face[14]; however, the interface they use to design their interactions have had variable impact on their ease of use. Some have more success with particular input features[25], others have struggled with using the input because of noisy detection or poor use of the affordances the input provides (see Section 2.2). The work of this thesis provides a technique for users to define their own face-based interactions and allows them to quickly master using their face as an input device. The continued re-evaluation of Face Interface has brought improvements in usability and encourages further research.

5.2 Future Work

This section outlines several directions future research may take.

5.2.1 Multiple Faces

While most desktop computer applications are made for a single user, those seeking to make more interactive spaces find value in observing groups of people. Attention Meter is capable of tracking multiple faces, so it is quite possible that the Face Interface user interface can be enhanced to support this ability.

5.2.2 Exposing the Affordances

Depending on the results of the experiment (section 4), it may be necessary to be more deliberate in exposing the affordances as we have discovered them (see Table 2.1). This could calls for changes as small as augmenting the avatar with text, to as large as adding a suggestion system.

5.2.3 Improved Eye Tracking and Face Direction Finding

Researchers at OMRON Corporation[26] have developed a single-camera face direction and eye-gaze finder[27]. This will allow for affordances to be created from subtle head movement gestures, such as tilting the head to one side, and eye-movement gestures, such as glancing at the edges of the screen. These affordances might also be used for pointing, as mentioned in section 5.2.5.

5.2.4 Context-of-Application Aware

It was planned to add application context to the rules in Face Interface, so that the rules would be able to affect different applications in different ways. For example, there could be a rule for nodding to scroll down while in a web browser, and another rule for nodding to change tracks while in a media player, and Face Interface would apply the correct rule depending on which application has focus. However, this feature was not vital for the experiment and so will be included in a subsequent version. The change will be a simple addition to the representation (UdfiTrigger class), but will require some more thoughtful UI work to let the user to choose the application.

5.2.5 Accessibility

As mentioned in Section 1.1, systems exist to allow for face-based pointer control for accessibility purposes. User-defined systems like Face Interface could augment these face-based pointers, allowing users to use gestures as shortcuts to avoid tedious motions. Alternatively, a face-based or eye-based pointer could be added to Face Interface.

5.2.6 Features from Affective Computing

The facial gestures that Face Interface works with are raw motor operations. Research in affective computing is creating programs and systems which can interpret facial video at a higher level. Systems like Self-Cam[28] will be able to detect whether the user is interested or bored, agreeing or disagreeing, concentrating or thinking. By applying this technology to Face Interface, we add new affective affordances which could lead users to more naturally work with their computers, conveying information to them as they would another person.

5.2.7 Exploring More Modalities

The idea that users can learn better from the experience of experimenting comes from experiential learning.[13] We expect that our methodology (Section 3.1) will hold true for other new input modalities as well as existing ones. In the future there could be a "Body-Gesture Interface" or a "User-Defined Haptic Interface" programs based on Face Interface.

Appendix A

Face Interface User Interface

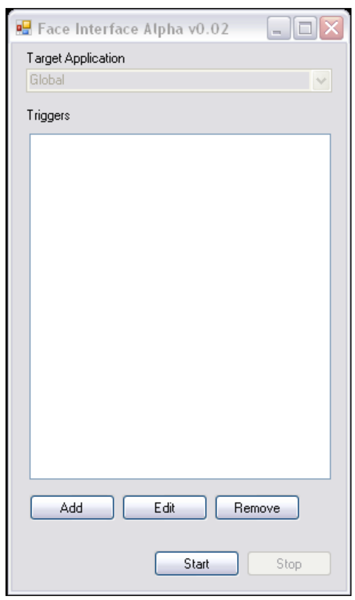

Figure A-1: The Face Interface Alpha main window.

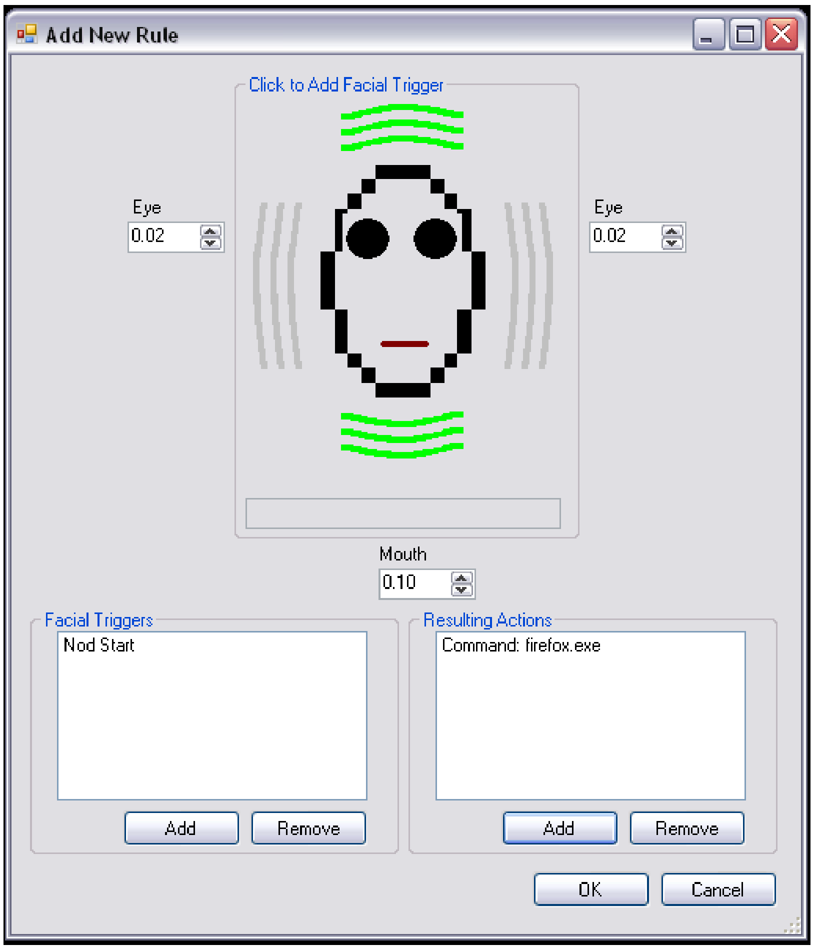

Figure A-2: The Face Interface Alpha rule editing window, with Avatar

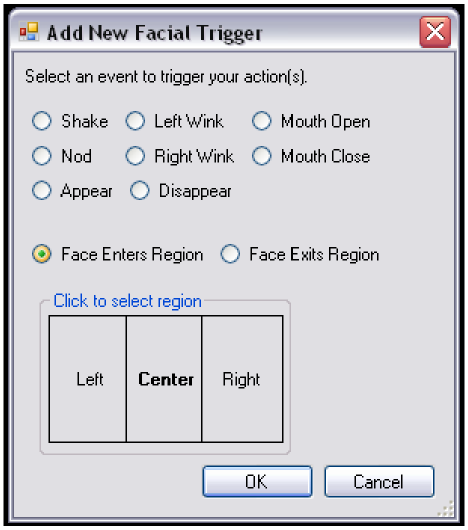

Figure A-3: The Face Interface Alpha gesture chooser

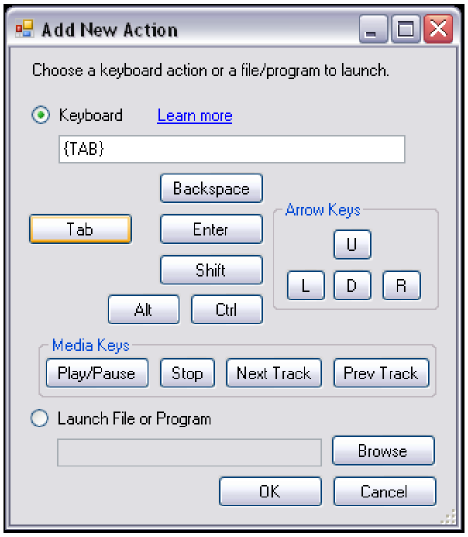

Figure A-4: The Face Interface Alpha action chooser

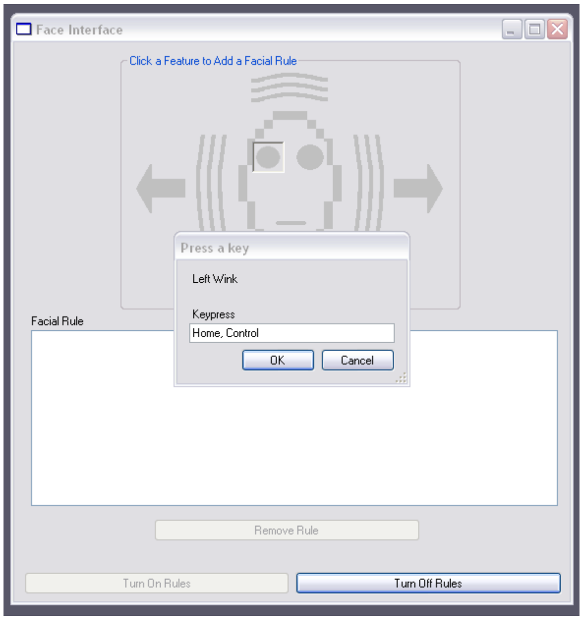

Figure A-5: The keystroke entry box in Face Interface (experiment version) appears after the user clicks on the desired gesture (in this case, a left wink).

Appendix B

Experiment Materials

CONSENT TO PARTICIPATE IN
NON-BIOMEDICAL RESEARCH

Face Interface User Study

You are asked to participate in a research study conducted by Professor Selker and Research Assistant Jon Wetzel, from the MIT Media Lab at the Massachusetts Institute of Technology (M.I.T.). Results from this experiment will be contributed to Jon Wetzel's Masters of Engineering thesis. You were selected as a possible participant in this study because you meet the age and language requirements. You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

- **PARTICIPATION AND WITHDRAWAL**

Your participation in this study is completely voluntary and you are free to choose whether to be in it or not. If you choose to be in this study, you may subsequently withdraw from it at any time without penalty or consequences of any kind. The investigator may withdraw you from this research if circumstances arise which warrant doing so.

- **PURPOSE OF THE STUDY**

This study is designed to help us learn how people learn to use our new computer program, "Face Interface", to use facial gestures to control a computer in the Windows environment.

- **PROCEDURES**

If you volunteer to participate in this study, we would ask you to do the following things:

1. **Read an instruction sheet and overview of Face Interface, then spend some time familiarizing with Face Interface. (Up to 10 minutes)**
2. **Read and perform a task using Google Maps and Face Interface. (Up to 15 minutes).**
3. **Complete a survey to give us your impressions with Face Interface. (Up to 5 minutes)**

The total time should be at most 30 minutes.

Figure B-1: Experiment Handout - Consent Form Page 1/4

- **POTENTIAL RISKS AND DISCOMFORTS**

We do not foresee any risks or discomforts. However, should problems arise, you are encouraged to contact us (see contact information below).

- **POTENTIAL BENEFITS**

From this study we hope you will get to express your views on our new Face-based interface. You may also become more interested in face-based technology in the future.

- **PAYMENT FOR PARTICIPATION**

You will receive a $10.00 voucher upon completion of the study.

- **CONFIDENTIALITY**

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission or as required by law.

While there will be a webcam for use in the "Face Interface", neither video or audio recording will be taken nor will your image ever appear in any of our data. At no time will we record anything that could compromise your confidentiality.

- **IDENTIFICATION OF INVESTIGATORS**

If you have any questions or concerns about the research, please feel free to contact:

Principal Investigator:
Jon Wetzel
MIT Media Laboratory
20 Ames Street, E15-320n
Cambridge, MA, USA 02142
jwwetzel@media.mit.edu
(617) 452-5518

Research Supervisor:
Ted Selker
MIT Media Laboratory
20 Ames Street, E15-322
Cambridge, MA, USA 02142
selker@media.mit.edu
(617) 253-0291, Fax (617) 253-0910

Figure B-2: Experiment Handout - Consent Form Page 2/4

- **EMERGENCY CARE AND COMPENSATION FOR INJURY**

"In the unlikely event of physical injury resulting from participation in this research you may receive medical treatment from the M.I.T. Medical Department, including emergency treatment and follow-up care as needed. Your insurance carrier may be billed for the cost of such treatment. M.I.T. does not provide any other form of compensation for injury. Moreover, in either providing or making such medical care available it does not imply the injury is the fault of the investigator. Further information may be obtained by calling the MIT Insurance and Legal Affairs Office at 1-617-253 2822."

- **RIGHTS OF RESEARCH SUBJECTS**

You are not waiving any legal claims, rights or remedies because of your participation in this research study. If you feel you have been treated unfairly, or you have questions regarding your rights as a research subject, you may contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T., Room E25-143B, 77 Massachusetts Ave, Cambridge, MA 02139, phone 1-617-253 6787.

Figure B-3: Experiment Handout - Consent Form Page 3/4

SIGNATURE OF RESEARCH SUBJECT OR LEGAL REPRESENTATIVE

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

____
Name of Subject

____
Name of Legal Representative (if applicable)

____________________________________ ____________
Signature of Subject or Legal Representative Date

SIGNATURE OF INVESTIGATOR

In my judgment the subject is voluntarily and knowingly giving informed consent and possesses the legal capacity to give informed consent to participate in this research study.

____________________________________ ____________
Signature of Investigator Date

Figure B-4: Experiment Handout - Consent Form Page 4/4

Face Interface User Study Intro Survey

Subject ID Number________

Date (Month/Day/Year):__________/____________/_____________

Major (if student):____

Degrees
Earned:____

Time Started:_________:_____________________________

How long have you been using a computer? (in years): ________________________

Have you used an on-line map program before (e.g. Google Maps, Mapquest):

____

If yes, which one(s)?

____

____

Figure B-5: Experiment Handout - Entrance Survey

Instructions

Your task is to use the program, *Face Interface,* to control Google Maps using only your face. This task will have two parts:

1. Familiarize yourself with Face Interface
2. Perform a Google Maps task using only your face.

1. Familiarize yourself with Face Interface

Face Interface allows you to use facial gestures, such as nodding, to automatically type as if you had pressed keys on the keyboard.

Google Maps has the following controls with keyboard shortcuts:

Map Control	Keyboard Shortcut
Pan Left	Home
Pan Right	End
Pan Up	Page Up
Pan Down	Page Down
Zoom In	+ (next to backspace)
Zoom Out	- (same row as backspace, next to "+")

For more information on Google Maps, see the "Help" link (at the top of the web page, towards the right.)

Face Interface allows you to control the features of Google Maps using facial gestures. Facial gestures include:

- Hiding and showing your face (can be done by turning the head away or covering with hands)
- Winking
- Nodding/Shaking your head
- Opening your mouth
- Moving your head to the left or right

The main window, labeled "Face Interface", contains all the controls for Face Interface.

- Use the imagery to select the gesture you want to create a rule for. This will make a "Choose a keystroke" dialog appear. Press they key you want to map to this gesture and then click "OK" when finished.
- Facial Rules appear in a large list box. To remove an existing rule, select the rule and then click "Remove Rule".
- If you want to disable your rules, click "Turn Off Rules". Pressing the "Turn On Rules" button will then re-enable them.

Figure B-6: Experiment Handout - Instructions

2. Perform a Google Maps task using only your face.

A web browser with Google Maps loaded is already on your desktop. It is currently centered on Tokyo, Japan. Using only your face and Face Interface, you must move the map view as close as possible to MIT, in Cambridge, Massachusetts, USA.

To do so you will have to add rules to control the zoom and pan features. Feel free to try changing your rules as needed to make the task as easy as possible.

Figure B-7: Experiment Handout - Timed Task

Face Interface User Study Exit Survey

Subject ID Number________

End Time _________:________________

How difficult would you rate this task? (check one)

Easy______ Somewhat Easy______ Moderate______ Somewhat Difficult______Difficult______

Rate the features on a scale of usefulness for Google Maps (1 = least useful, 5 = most useful)

Nodding	1______	2______	3______	4______	5______
Shaking	1______	2______	3______	4______	5______
Winking	1______	2______	3______	4______	5______
Face Appearing/Disappearing	1______	2______	3______	4______	5______
Mouth Open/Closed	1______	2______	3______	4______	5______
Face Moving Left/Right	1______	2______	3______	4______	5______

Would you use Face Interface with Google Maps on a regular basis? (check one)

No_____ Maybe Not_____ Maybe Yes_____ Yes_____

If Face Interface let you choose any keys, would you use Face Interface with other applications (this may include games)?

No_____ Maybe Not_____ Maybe Yes_____ Yes_____

If not "No", which applications?

____

____

Figure B-8: Experiment Handout - Exit Survey

We are looking for participants!

Face-Controlled Computing

Operate a computer using your face.

Make **$10.00** in less than 30 minutes.

Participants must be over 18 years old and have good command of the English language.

For more info and sign up e-mail:
Face.Interface.Experiment@gmail.com

Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com
Face Interface Experiment Sign up: Face.Interface.Expriement@gmail.com

Figure B-9: Experiment Poster

	Massachusetts Institute of Technology Committee on the Use of Humans as Experimental Subjects	Application # (assigned by COUHES)	
		Date	

APPLICATION FOR APPROVAL TO USE HUMANS AS EXPERIMENTAL SUBJECTS (EXEMPT STATUS FORM)

Please answer every question. Positive answers should be amplified with details. You may mark N/A where the question does not pertain to your application. Any incomplete application will be rejected and returned for completion.

I. BASIC INFORMATION

1. Title of Study	
Face Interface User Study	
2. Investigator	
Name: Jon Wetzel	Building and Room #: E15-320n
Title:	Email: jwwetzel@mit.edu
Department: EECS	Phone: (617) 452-5518
3. Faculty Sponsor. *If the investigator is a student and the research is for a thesis or dissertation, then a faculty sponsor must be identified and sign below.*	
Name: Ted Selker	Email: selker@media.mit.edu
Title: Associate Professor	Phone: (617) 253-6968
Affiliation: MIT Media Lab	
4. Funding. *If the research is funded by an outside sponsor, the investigator's department head must sign below.*	
Outside Sponsor:	Contract or Grant Title:
Contract or Grant #:	OSP #:
5. Human Subjects Training. *All study personnel in research **MUST** take and pass a training course on human subjects research. MIT has a web-based course that can be accessed from the main menu of the COUHES web site. COUHES may accept proof of training from some other institutions. List the names of all study personnel and indicate if they have taken a human subjects training course.*	
Jon Wetzel – COUHES Trained Ted Selker – COUHES Trained	
6. Anticipated Dates of Research	
Start Date: 5/16/2007	Completion Date: 7/16/2007

I. STUDY INFORMATION

1. Purpose of Study. *Please provide a brief statement of the background, nature and reasons for the proposed study. Use non-technical language.)*
The purpose of this study is to learn how users can learn to use our program, Face Interface, under two different conditions. One set of users will have a copy of the program with an interactive avatar of a face, while the other has only a written set of instructions. In this study users will read a set of instructions, and perform a task using the program.

APPLICATION FOR APPROVAL TO USE HUMANS AS EXPERIMENTAL SUBJECTS (EXEMPT FORM) – revised 8/5/2003

- 1 -

Figure B-10: COUHES Exempt Form, Page 1/2

In this experiment, Face Interface is a program for allowing users to control Google Maps using their face. To do this it can recognize some facial gestures, such as winking, nodding, shaking, and opening the mouth. In our study, Face Interface will enable the user to control Google Maps by making rules which map the facial gestures onto commands in Google Maps. For instance, a person could make a rule in which nodding causes the map to zoom in.

2. Study Protocol. *Please provide an outline of the proposed research. You should provide sufficient information for effective review by non-scientist members of COUHES. Define all abbreviations and use simple words. Unless justification is provided, this part of the application must not exceed 2 pages. Attaching sections of a grant application is* ***not*** *an acceptable substitute. Include copies of any questionnaire or standardized tests you plan to use. If your study involves interviews, submit an outline of the types of questions you will include. Your research outline should include a description of:*

A. Experimental procedures:

1. **Read and sign consent form. (if COUHES considers necessary)**
2. **Read the instruction sheet and overview of Face Interface. Spend 5 minutes familiarizing with Face Interface.**
3. **Read and perform the Google maps task. (Up to 15 minutes).**
4. **Complete exit survey.**
5. **Receive Compensation. ($10 MIT Voucher)**

B. Type and number of subjects involved:
20 people, ages over 18 years.

C. Method of recruitment:
Posters on MIT campus

D. Length of subject involvement:
30 minutes maximum

E. Location of the research:
E15-320

F. Procedures for obtaining informed consent *(COUHES may waive this requirement)*:
We will provide written consent form at COUHES discretion.

G. Procedures to ensure confidentiality:
No data will be associated with personal information (name, etc.). No video data will be recorded at any time. Each subject will be given a randomly generated four-digit ID number. Surveys results will be stored in electronic format and kept in the laboratory on computers without network connection. Any paper copies will be destroyed once their data is stored electronically. All data will be saved until the results of the experiment are analyzed.

3. HIPAA Privacy Rule. *If you are in any way working with protected health information you must comply with the HIPAA Privacy Rule. If your study has exempt status you will generally be dealing with de-identified data and your study will not be subject to the Privacy Rue. Please refer to the definition of de-identified data on the COUHES web site.*

APPLICATION FOR APPROVAL TO USE HUMANS AS EXPERIMENTAL SUBJECTS
(EXEMPT FORM) – revised 8/5/2003
- 2 -

Figure B-11: COUHES Exempt Form, Page 2/2

Bibliography

[1] Young, L., and Sheena, D. 1975. Survey of eye movement recording methods. Behavior Research Methods and Instrumentation 7, 397–429.

[2] Viola, P., Jones, M. Robust real-time face detection. In Proceedings of the ICCV 2001 Workshop on Statistical and Computation Theories of Vision, ICCV'01, volume 2, page 747, Vancouver, Canada, July 2001.

[3] Selker, T. Image Based Focusing. In ROBOTICS & INDUST. INSPECTION., 1983, vol. 360, 96-99

[4] OpenCV - Intel Open Source Computer Vision Library, http://www.intel.com/

[5] NaturalPoint TrackIR, http://www.naturalpoint.com/

[6] Origin Instruments HeadMouse, http://orin.com/

[7] Origin Instruments Dragger32, http://orin.com/

[8] Graveleau, V., Mirenkov, N., Nikishkov, G., A Head-Controlled User Interface. In Proceedings of the 8th Int. Conf. on Humans and Computers HC 2005, Aizu-Wakamatsu, 31 Aug-2 Sep. 2005, pp. 306-311.

[9] Tu J., Huang T., Tao H., Face as Mouse Through Visual Face Tracking. In Proceedings of the Second Canadian Conference on Computer and Robot Vision (CRV05)

[10] Kjeldsen, R. 2006. Improvements in vision-based pointer control. In Proceedings of the 8th international ACM SIGACCESS Conference on Computers and Acces-

sibility (Portland, Oregon, USA, October 23 - 25, 2006). Assets '06. ACM Press, New York, NY, 189-196. DOI= http://doi.acm.org/10.1145/1168987.1169020

[11] Davis, J.W., Vaks, S., A perceptual user interface for recognizing head gesture acknowledgements, Proceedings of the 2001 workshop on Perceptive user interfaces, November 15-16, 2001, Orlando, Florida.

[12] Morency, L., Darrell T., Head Gesture Recognition in Intelligent Interfaces: The Role of Context in Improving Recognition. In IUI06, January 29February 1, 2006, Sydney, Australia.

[13] D. A. Kolb. Experimental Learning: Experience as the Source of Learning and Development. Englewood Cliffs, NJ: Prentice-Hall.

[14] Lee, C. J., Wetzel, J., and Selker, T. 2006. Enhancing interface design using attentive interaction design toolkit. In ACM SIGGRAPH 2006 Educators Program (Boston, Massachusetts, July 30 - August 03, 2006). SIGGRAPH '06. ACM Press, New York, NY, 18. DOI= http://doi.acm.org/10.1145/1179295.1179314

[15] Adobe Flash, http://www.adobe.com/

[16] Asian Reality design workshop, http://nightmarket.org/

[17] The PlaceMap Project, http://placemap.mit.edu/

[18] Ben Shneiderman, Designing the user interface (2nd ed.): strategies for effective human-computer interaction, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1992

[19] Selker, T., Burleson, W., Scott, J. and Li, M. Eye-Bed, Workshop on Multimodal Resources and Evaluation in conjunction with the Third International Conference on Language Resources and Evaluation (LREC). June 2002.

[20] Microsoft .NET Framework, http://www.microsoft.com/

[21] Norman, D. A. Psychology of Everyday Things. Basic Books, 1988

[22] Digital Life Consortium, http://dl.media.mit.edu

[23] Things That Think Consortium, http://ttt.media.mit.edu/

[24] Logitech QuickCam Pro 5000, http://www.logitech.com/

[25] Lee, C.J., Jang, C.I., Chen, T.D., Wetzel, J., Shen, Y.B., Selker, T. Attention meter: a vision-based input 6 toolkit for interaction designers. Ext. Abstracts CHI 2006, ACM Press (2006), 1007-1012

[26] Omoron Corporation, http://www.omron.com/

[27] Kinoshita, K., Ma, Y., Lao, S., and Kawaade, M. 2006. A fast and robust 3D head pose and gaze estimation system. In Proceedings of the 8th international Conference on Multimodal interfaces (Banff, Alberta, Canada, November 02 - 04, 2006). ICMI '06. ACM Press, New York, NY, 137-138. DOI= http://doi.acm.org/10.1145/1180995.1181026

[28] Teeters, A., El Kaliouby, R., and Picard, R. 2006. Self-Cam: feedback from what would be your social partner. In ACM SIGGRAPH 2006 Research Posters (Boston, Massachusetts, July 30 - August 03, 2006). SIGGRAPH '06. ACM Press, New York, NY, 138. DOI= http://doi.acm.org/10.1145/1179622.1179782