Gesture output: Eyes-free output using a force feedback touch surface

Conference Paper · April 2013

DOI: 10.1145/2470654.2481352

6 authors, including:

Max Plauth
Hasso Plattner Institute
13 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Pedro Lopes
Hasso Plattner Institute
31 PUBLICATIONS **169** CITATIONS
SEE PROFILE

Patrick Baudisch
Hasso Plattner Institute
177 PUBLICATIONS **5,148** CITATIONS
SEE PROFILE

Gesture Output: Eyes-Free Output Using a Force Feedback Touch Surface

Anne Roudaut, Andreas Rau, Christoph Sterz, Max Plauth, Pedro Lopes, Patrick Baudisch
Hasso Plattner Institute, Potsdam, Germany
roudauta@gmail.com, {andreas.rau, christoph.sterz, max.plauth}@student.hpi.uni-potsdam.de,
{pedro.lopes, patrick.baudisch}@hpi.uni-potsdam.de

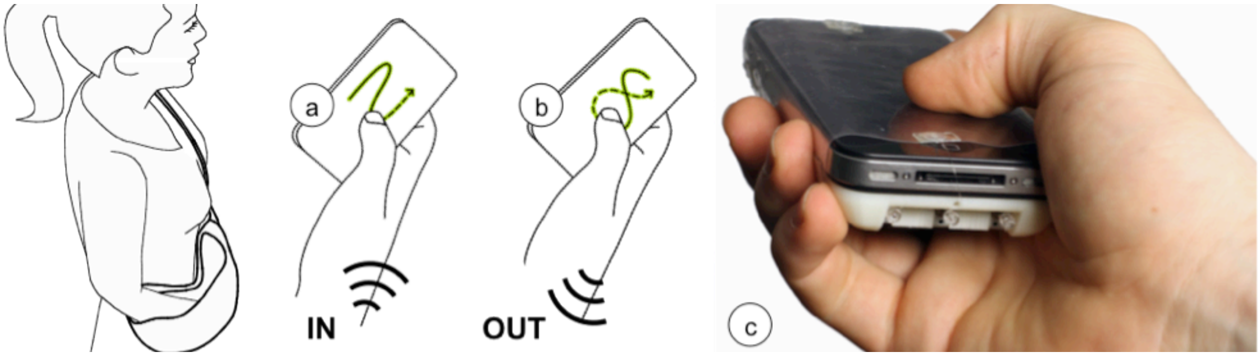

Figure 1: With our proposed gesture output, the device outputs messages to users using the same gesture language used for input. (a) Here, the user draws an N to check the house number of the upcoming meeting. (b) The device replies by translating the user's finger along the path of an 8. (c) The pocketOuija is one of the two force feedback touchscreen devices we built that support gesture output. It translates the user's finger by means of a transparent plastic foil overlaid onto the screen actuated using motors located on the back of the device.

ABSTRACT

We propose using spatial gestures not only for input but also for output. Analogous to gesture input, the proposed *gesture output* moves the user's finger in a gesture, which the *user* then recognizes. We use our concept in a mobile scenario where a motion path forming a "5" informs users about new emails, or a heart-shaped path serves as a message from a friend. We built two prototypes: (1) The *longRangeOuija* is a stationary prototype that offers a motion range of up to 4cm; (2) The *pocketOuija* is self-contained mobile device based on an iPhone with up to 1cm motion range. Both devices actuate the user's fingers by means of an actuated transparent foil overlaid onto a touchscreen.

We conducted 3 studies on the longRangeOuija. Participants recognized 2cm marks with 97% accuracy, Graffiti digits with 98.8%, pairs of Graffiti digits with 90.5%, and Graffiti letters with 93.4%. Participants previously unfamiliar with Graffiti identified 96.2% of digits and 76.4% of letters, suggesting that properly designed gesture output is *guessable*. After the experiment, the same participants were able to *enter* 100% of Graffiti digits by heart and 92.2% of letters. This suggests that participants learned gesture input as a side effect of using gesture output on our prototypes.

ACM Classification: H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces, Input devices and strategies, Haptic I/O.

Keywords: Gestures; Eyes Free; Force feedback; Touch.

INTRODUCTION

Gesture input allows users to interact eyes-free (non-visual, non-auditory) with their mobile touch devices, using an expressive and mnemonic set of commands [1]. Saponas et al. found that this is even possible while walking, based on users' sense of touch alone [22].

In order to have a dialog with the device, users need not only eyes-free input, but also *output.* Unfortunately, auditory output is not always possible, and vibrotactile output [3], which is the predominant eyes-free non-auditory type of output, was found to offer limited expressiveness [13], low bandwidth [17] and is hard to learn because it lacks mnemonic properties [12]. As a result, mobile users typically enter an expressive, mnemonic, easy-to-learn gesture as input (such as writing an M to request "messages"), but the system's response will be akin to Morse code [17]. This makes *output* the bottleneck of the system.

A way to alleviate this bottleneck is to use an array of vibrotactile cells, e.g. to render spatial strokes [10]. In this paper, however, we go one step further by enabling the system to actuate the finger in the form of a 2D gesture.

GESTURE OUTPUT CONCEPT

We propose the concept of *Gesture output*, a non-visual, non-auditory output technique that communicates 2D gestures to users by moving their finger along the path of a gesture. While this concept opens a new range of interaction possibilities, we think Gesture output is particularity interesting in a mobile scenario where visual and audio modalities are not always available. Fig.1 illustrates a scenario we envision. Without taking the device out of the bag,

CHI 2013, April 27–May 2, 2013, Paris, France.

the user touches the device and draws an N to ask the system for the house number of a meeting. The device replies by translating the user's finger along the path of an 8.

We created two prototypes capable of performing gesture output. Fig.2 shows one of them up-close. The pocketOuija uses a set of motors and a pulley system to actuate a flexible plastic foil on top of the screen of a touchscreen device, here an iPhone. We will present this device, as well as the desktop version using a PHANToM, in reproducible detail.

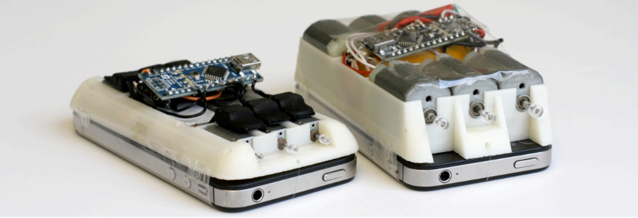

Figure 2: Our pocketOuija, here two versions of it, actuate the user's finger by moving a clear foil on top of the device's touchscreen (here an iPhone).

The language of gesture output

Gesture output can be used with any gesture language made of single stroke characters. Conceptually, this allows defining gesture output languages based on arbitrary strokes, which can be optimized for arbitrary objectives. Following the lead of the original unistroke input language, for example, we might pick strokes that are efficient to perform, so as to optimize for expert interaction performance (see *Study 1: recognizability of gesture output* for a study on performance with *marks* as an output language).

However, we argue that the main opportunity in gesture output is *learnability*: Vibrotactile patterns are hard to memorize because there are few existing associations between a vibrotactile pattern and the information it is encoding; thus users have to learn such associations to decode patterns before understanding their meaning for the system.

Gestures in the 2D plane, in contrast, associate readily with a wealth of existing mnemonic associations, including doodling, scribbling, and handwriting. We exploit these by adopting a gesture alphabet built on such associations. Such languages are readily available, including Graffiti and EdgeWrite [35]. For the purpose of this paper, we adopt Graffiti. Based on this, the shape 8 used in our introductory example, naturally communicates the digit "8", because users have spent years building up this association.

While gesture output is designed to simplify learning, interpreting a message requires cognitive focus. Although no visual focus is required, gesture output may require users to focus, making it difficult to perform other tasks in parallel.

Single-character messages

Single-character messages allow the system to notify the user or to answer a question.

Notify: The system uses a vibrotactile buzz to get the attention of the user. Then the user places the hand onto the device, and the system delivers the message, such as b for "low battery warning".

Question: a user enters the word "messages" using Graffiti (or just a single M gesture for short) and the system might respond with 5 for "5 unread messages". The system may respond ✓ or X to binary questions from the user, or → when asked what direction to go. A user can also enter ? to ask the system to repeat the message.

Compound messages

Compound messages, e.g. M2 for "two new messages" or ┐→ (T→) for "turn right" require to add a *delimiter* to our language. For gesture input, the delimiter is implemented by users lifting the finger or stylus off the screen. This clarifies when a character ends and the next one begins. We could try to port this concept to gesture output, but we want to maintain contact between user and device at all times to make sure the user is not missing anything. We therefore use the vibrotactile buzzer as delimiter. Using this model, we output "Turn right" as ┐•→ and the number thirty-six as the sequence 3 • 6 with "•" being the buzz delimiter.

We use the time span during which the delimiter is playing to move the finger to the beginning of the next gesture, e.g. for 3 • 6 the finger is translated diagonally between the end of the 3 and the start of the 6, the digits being superimposed in space. This keeps gestures in a consistent spatial reference and prevents longer gesture output sequences from driving the finger out of the bounds of the device.

Note that we can also extend this approach to more than two symbols. For instance, it can serve to spell out a contact name, words, and possibly even sentences. The ability of users to recognize gesture output composed of more than two symbols is, however, not addressed in this paper and requires further investigations.

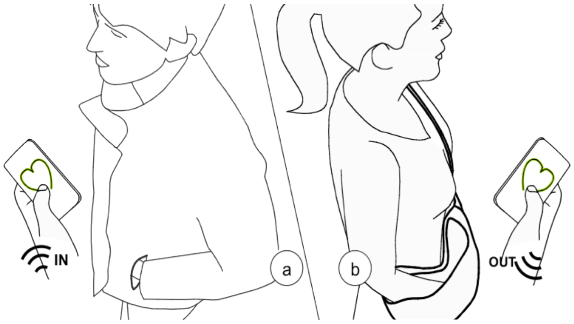

Figure 3: (a) A boyfriend sends a heart. (b) The girlfriend touches her device, the message is presented by moving her finger along the same heart.

Between people

We can use the same approach to enable the communication between users (Fig.3). Since no automatic recognition engine is involved here, any gesture both users have agreed upon can be used for communication.

STUDIES OVERVIEW

The main benefit of gesture output is its learnability because users are able to readily use a wealth of existing mnemonic associations. In this paper, we present three user studies that support this claim on the longRangeOuija.

Study 1: recognizability of gesture output

We wanted to verify the basic mechanics, i.e., if users were able to receive and recognize gestures. We therefore picked directional marks as a self-explaining gesture alphabet, and checked whether users were able to recognize their direction. Results show that using 1cm marks allows participants to recognize the eight compass directions with 86.8% accuracy, and marks longer than 2cm with 97% accuracy.

Study 2: learnability of single-character messages

The goal was to investigate if knowledge of input helps understand output ("transfer learning"). We picked the mnemonic alphanumeric Graffiti alphabet. We hypothesized that training in *input* would allow participants to successfully recognize *output* (and vice versa). Furthermore, due to the design of Graffiti for guessability, we hypothesized that Graffiti *output* would also be guessable, so that participants without training should be able to decode the gestures. Results show that users familiar with Graffiti *input* but with no training in Graffiti *output* recognized Graffiti *output* with 98.8% accuracy for digits and 93.4% for letters, thus showing that transfer learning had occurred. Participants unfamiliar with Graffiti altogether correctly *guessed* 96.2% of digits and 76.4% of letters, thus showing that the alphabet is self-explanatory. Finally, the same participants correctly *input* 100% of digits and 92.2% of letters after the experiment, thus showing that *reverse* transfer learning had occurred as well.

Study 3: learnability of bi-grams

In this study, we go further in our investigation of learnability and explored *compound* messages. We picked a highly mnemonic gesture alphabet made of pairs of Graffiti digits. We hypothesized that training in gesture *input* would allow users to successfully recognize compound gesture *output* by transfer learning and by aggregation of input knowledge. Results show that participants familiar with Graffiti *input* but with no training in Graffiti output recognized compound Graffiti *output* with 90.5% accuracy, thus showing that our design works for two-digits sequences. Additional studies are required for longer gesture sequence.

CONTRIBUTION

Our main contribution is the concept of gesture output that creates symmetry between non-visual, non-auditory input and output. We also present two prototypes, a desktop force feedback touchscreen (longRangeOuija) and a pocketsize version (pocketOuija). We contribute three user studies on the longRangeOuija that support that the blending of input and output in gestures is learnable even without training.

RELATED WORK

Vibrotactile output

Vibrotactile messages (Tactons [3]) allow communicating non-visual information using different rhythms and amplitude of vibration. For instance, Tan proposed associating vibration patterns with Morse code [26]. Another example is Shoogle that transforms the contents of the user's inbox into virtual "message balls" [33]. A user shaking Shoogle hears and feels the impacts of the balls bouncing around.

Implementing vibrotactile is comparably simple—it requires only an eccentric motor or voice coil—thus many of today's mobile devices offer it [21]. However, vibrotactile lacks expressiveness [13] and bandwidth [17]. In particular, a single vibrotactile unit allows conveying binary information, such as "target hit", but cannot directly encode locations. Vibrotactile also requires long learning phases as it is perceptively and cognitively demanding [12]. For instance Geldard [8] reported that users required 65 hours of training to recognize an encoding of the English alphabet.

Several works extend the expressiveness of vibrotactile messages using arrays of vibrotactile cells (e.g., [24, 37]). For instance, Poupyrev proposed augmenting mobile devices with tactile arrays in order to guide the user's finger and to create awareness interfaces [21]. In more recent work, Israr used a vibrotactile array mounted into a backrest to provide gamers with directional feedback [10].

Force feedback

Unlike vibrotactile, force feedback mechanisms allow creating a *directional* force. In their simplest form, force feedback devices offer a single degree of freedom. For instance, Enriquez [6] proposed using an actuated 1DOF rotary knob for output of brief computer-generated signals (haptic icons). More complex devices include articulated arms (e.g. PHANToM or Falcon) that allow 3D force feedbacks through a pen or an intermediary object. For instance, with the Palmtop display [18], a mobile device is attached to the articulated arm. It enables users to manipulate a remote object as if they were holding it in their hands. A limitation of articulated arms is that they only create force feedback at a single point. In contrast, the SPIDAR system [23] offers multi-point controls: it uses motors and a pulley system to actuate each finger of the user independently in order to create a sensation of manipulating 3D objects in the air.

Force feedback for communication between users

Force feedback has been used to allow users to communicate over a distance. Each InTouch device, for example, consists of three cylindrical rollers mounted on a base [2]. Each action done on one device is replicated on the other one creating the illusion of a single shared physical object. Telephonic Arm Wrestling [32] simulates the feeling of arm wresting over a telephone line. The Dents Dentata [9], device can squeeze a users hand while calling.

Force feedback in training systems

Much research has examined the use of force feedback to train users in performing tasks, such as surgery. Feygin [7] for instance introduced the term *haptic guidance* that consists in guiding users through an ideal motion, thus giving the user a kinesthetic understanding of what is required. Dang [5] also discusses a system that provides guidance to users performing surgery by restricting their movements from deviating from a path recorded previously by a real surgeon. Several researches have built on the same principle of replaying expert gestures to train motor skills: [25] for handwriting, [27] for writing Chinese characters, [38] for training medical operations or [15] teach an abstract motor skill that requires recalling a sequence of forces.

Actuated touchpads, tabletops, and touchscreens

On tabletop systems, actuating systems were initially used to actuate tangibles. Actuated workbench [19] and Pico [20] were the first systems of this kind; they actuated tangible pucks using an array of electromagnets mounted below the table. Madgets [30] extend this approach by moving tangible widgets consisting of multiple moving parts.

Similar approaches have been used to actuate fingers. FingerFlux, for example, combines the Madgets platform with finger-worn magnets to apply force feedback to that finger [31]. ShiverPad [4] combines a programmable friction device [34] with the *slip stick* effect, i.e., by alternating between low and high friction at the same frequency that the device is moving in the plane. At 60 milliNewtons, the device is not strong enough to move the finger, but it is able to actuate a little plastic ball.

Other devices combine motors and pulley mechanisms. Wang introduced the Haptic Overlay Device [28, 29]: the user touches an overlay material connected to drive rollers that can translate. In ActivePad [16], the same mechanism is combined with a programmable friction surface. Fing Viewer [36], a 2D version of the SPIDAR system [23], actuates a ring the user is touching using four motors mediated by cables. Our prototype pocketOuija uses this same string-motor mechanism, but allows for a mobile form factor by using a different arrangement of motors.

PROTOTYPE #1: THE LONG-RANGE-OUIJA

The longRangeOuija is our first prototype design and it is optimized for providing us all the control we need to run a wide range of user studies, such as how scale of gesture affects comprehension (see User Study 1).

As shown in Fig.4, the longRangeOuija transmits force to the user's finger via a rigid transparent foil overlaying the actual touch surface (an iPad). The foil is actuated using a PHANToM force feedback device, a device normally designed for moving a stylus in 3D space.

Figure 4: The longRangeOuija translates the user's finger via a clear foil actuated by a PHANToM arm.

During *gesture input* the motors in the PHANToM are turned off and the foil drags with the user's finger. Users can do so with reasonable resistance because the foil overlay is designed for minimum weight (100g). During gesture output, the foil is actuated by the PHANToM. The PHANToM delivers up to 3.3 Newtons.

Mechanics

Fig.5 illustrates the mechanics of the prototype. On the right, the foil is actuated by the PHANToM. On the left, the foil is guided by a groove that only permits left-right motion. This mechanical design causes the foil to pivot around its left extremity labeled *S* in Fig.6. This creates a non-linear relationship between the motion of the PHANToM arm and coordinate system of the iPad and the user's finger. The system translates between both systems as follows: Given F (finger start), A (arm start) and F' (finger final), we search A' (arm final):

$$(1)\quad Sx=-\sqrt{\|SA\|^2-Ay^2}+Ax \qquad (2)\quad Sx'=-\sqrt{\|SF\|^2-Fy'^2}+Fx'$$

$$(3)\quad F\hat{S}A=\arccos\left(\frac{SF\bullet SA}{\|SF\|\|SA\|}\right) \qquad (4)\quad F'\hat{S}'x=\arcsin\left(\frac{Fy'}{\|SF\|}\right)$$

$$(5)\quad \begin{aligned} Ax' &= \|SA\|\times\cos(F'\hat{S}'x+F\hat{S}A)+Sx' \\ Ay' &= \|SA\|\times\sin(F'\hat{S}'x+F\hat{S}A) \end{aligned}$$

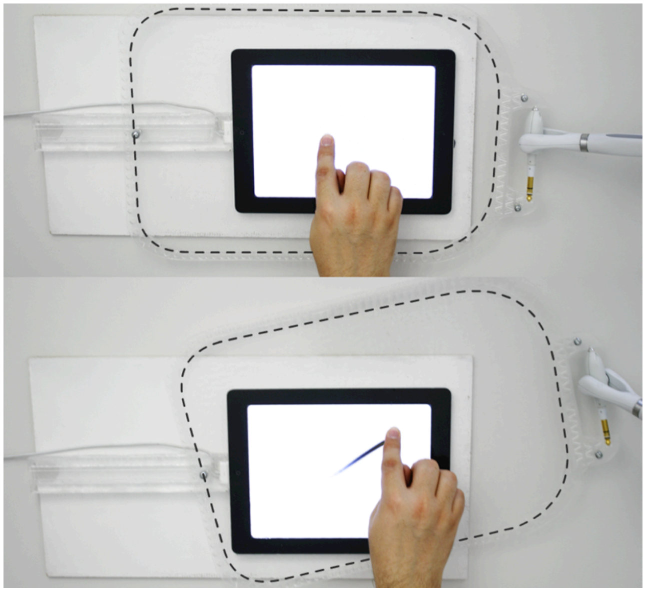

Figure 5: The longRangeOuija consists of a transparent foil actuated by a PHANToM. Here the arm pulls the foil to the top right and away from the user, displacing the finger accordingly.

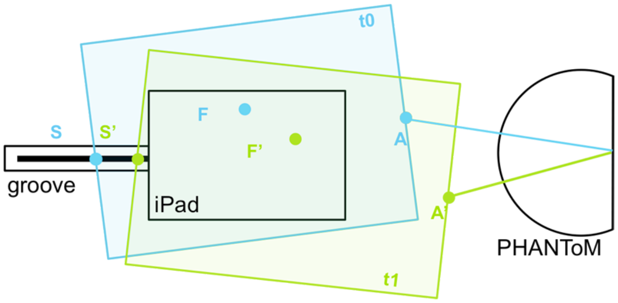

Figure 6: The longRangeOuija mechanics.

Software

The system senses the location of the finger via the iPad. The location of the foil is known via the PHANToM that is controlled using a computer running the OpenHaptics C++ library. The interface on the iPad is done with HTML5 and

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you’re on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm’s cloud-native docket research platform finds what other services can’t. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**