

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

roudaut@gmail.com, {andreas.rau, christoph.sterz, max.plauth}@student.hpi.uni-potsdam.de,
{pedro.lopes, patrick.baudisch}@hpi.uni-potsdam.de

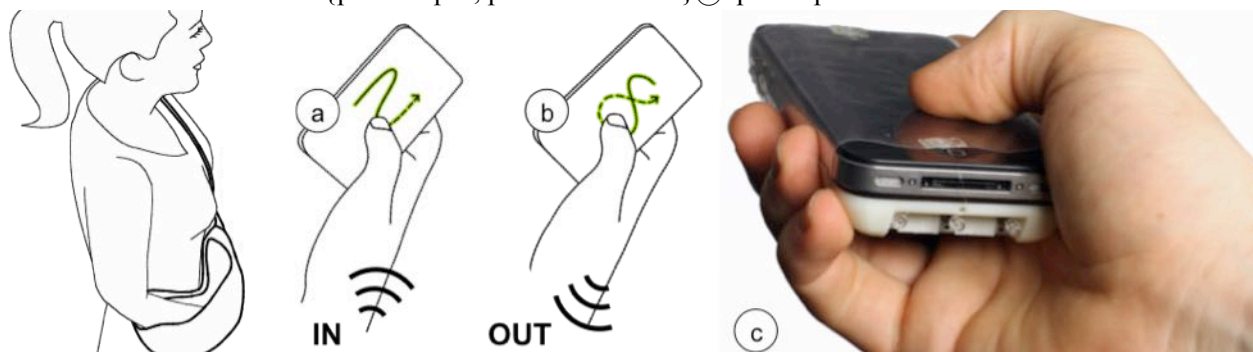


Figure 1: With our proposed gesture output, the device outputs messages to users using the same gesture language used for input. (a) Here, the user draws an **N** to check the house number of the upcoming meeting. (b) The device replies by translating the user's finger along the path of an **8**. (c) 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 *long-RangeOuija* 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 particularly 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,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

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

Copyright © 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

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 **✗** 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.

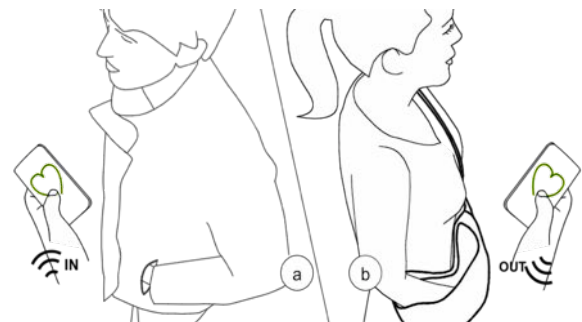


Figure 3: (a) A boyfriend sends a heart. (b) The girlfriend touches her device, the message is presented by moving her finger along the same heart.

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 pocket-size 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 wrestling 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, tablesps, 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):

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

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

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

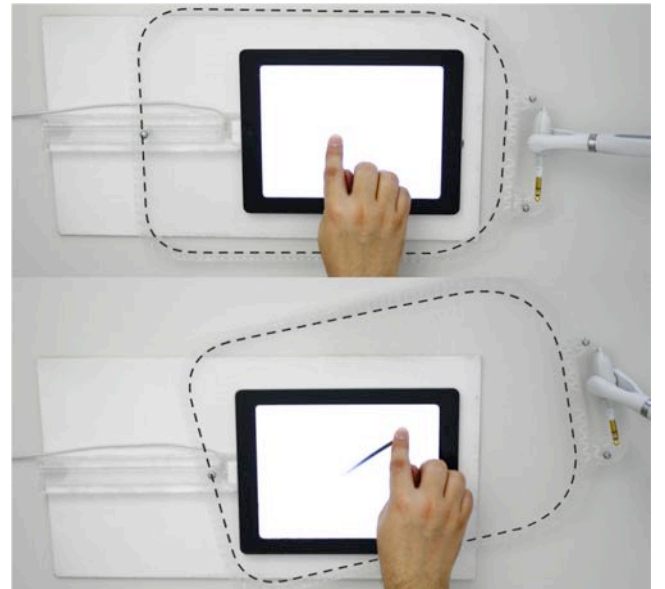


Figure 5: The longRangeOuija consists of a transparent foil actuated by a PHANToM. Here the arm pulls the foil to the top right and away from the user, displacing the finger accordingly.

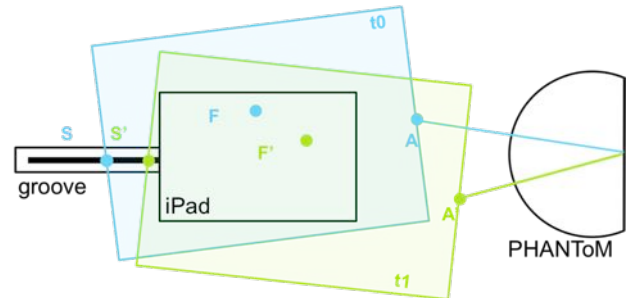


Figure 6: The longRangeOuija mechanics.

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

JavaScript in the Safari browser. The browser sends the coordinates of the touch events over a wireless network using XMLHttpRequest. The computation required for rendering gesture output on the PHANToM is processed on the computer side, thus maintaining a low latency between the iPad touch events and the haptic stimuli. In the studies, we used a speed between 3 and 4 cm/s.

PROTOTYPE #2: THE POCKET-OUIJA

The pocketOuija enables to experience gesture output in a mobile scenario. Our prototype (two versions are shown in Fig.2) is mobile, battery-operated, and self-contained.

The pocketOuija uses six motors to actuate a transparent sheet of plastic foil overlaid on the touchscreen of an iPhone. The prototype delivers 4 Newtons of force, which it transmits to the foil via a system of nylon strings. The device receives instructions from the iPhone via the iPhone's headphone-jack (we use frequency shift keying). The pocketOuija adds 30mm of thickness and 280g of weight to the iPhone. The smaller version we developed is limited in force and action radius but has a reduced weight of just 120g while measuring 17mm in depth.

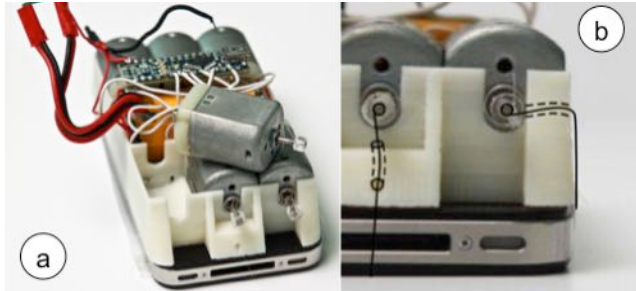


Figure 7: The pocketOuija mechanical design: (a) The casing contains 6 motors, 2 batteries and an Arduino Nano. (b) The casing implements a system of tubes that guide strings around the device.

Mechanics

Fig.7 illustrates the mechanical design of the pocketOuija, which consists of four parts:

The *clear foil* measures 9.5x8cm, making it 3cm wider and higher than the iPhone screen. The foil is 32 μ m thick, which is thin enough to allow touch input to be picked up by the capacitive touchscreen.

The *string system* is inspired by SPIDAR [23] and Fing Viewer [36]. In order to achieve the mobile form factor we modified the design as follows: (1) We built a system of tubes (Fig.7b) to guide the nylon strings around the device while minimizing friction. (2) We added two motors for an overall number of six. The additional motors, shown in the center of Fig.7b, provide additional motion range in that they allow pulling the foil all the way; at this point the strings of other two motors pull the foil into opposite directions and thus lose their power. (3) To obtain torque without a gearbox we used the thin motor axels directly as winches to roll up the nylon string. The strings are made from 0.18mm Nylon. In order to attach them to the foil, we sandwiched plastic washers between the film and its folded border ears and knotted the strings through both.

The *six motors* are MABUCHI FK-280 DC-Motors turning at 7.4Volts with an approximate torque of 2Nm each.

The *3D printed casing* holds the motors in place and provides the pipes which guide the pulley strings to the right location, while maximizing corner radii as shown in Fig.7b.

Electronics

Fig.8 illustrates the electronic design of the device. It consists in three components:

An *Arduino Nano* is programmed via a mini-USB port. Soldering battery pins and controller wires directly onto the board allowed up to fit the Arduino and the batteries in the available space between the motors.

A *Lithium-ion polymer battery* consists of 2 single cell S1 LiPo modules that are specified for 3.7Volts each. It provides the current of 1.3A required by the motors.

Six *transistors* control the current of up to 400mA each. We used BD243C FET enhanced with a suppressor diode protecting the transistor from inductive flyback.

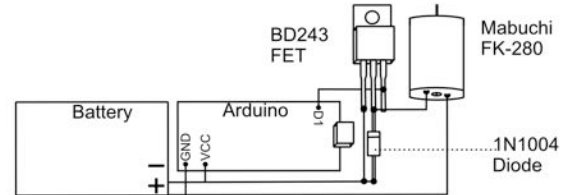


Figure 8: The pocketOuija electrical design consists of a lithium battery, an Arduino as well as one FET and Diode per motor (here only one is shown).

Software

The software running on the Arduino receives messages to be performed from a program on the iPhone. To actuate the foil, the iPhone app calculates the sequence of motor voltages required to produce the respective motion path and sends it to the Arduino via the iPhone's headphone jack (it encodes the information as a sequence of frequencies). As the device cannot detect the location of the foil itself, we can reset the sheet by pulling the foil alternately from all sides while decreasing force in each motor, which centers the foil in the middle of the screen. The motors are actuated at constant speed and allow translating the user's finger at a speed between 3 and 4 cm/s. (3 to 4cm/s).

Limitations and generalization of findings

The pocketOuija is an early prototype of a force-feedback touchscreen and therefore portraits limitations. Most importantly, it adds a plastic foil on top of the screen, affecting the user experience during regular interaction. It also adds 120g and 17mm of thickness to the mobile phone for motors and battery. Also, the motors generate noise equivalent to the vibrotactile motor already found on such devices. Our prototype also offer only 1cm range of motion: while this design had dedicated motors for North and South, it produced East motion by mixing force from the NE and SE motors. It is possible to improve this design by having a motor for all eight directions. Finally, we only explored the feasibility of gesture output on the longRangeOuija; other force feedback devices may produce different results.

STUDY 1: RECOGNIZABILITY OF GESTURES

While gesture output ultimately is about learnability (see Study 2 and 3), we first wanted to verify the basic mechanics, i.e., if users are able to receive and recognize gestures. We thus conducted a study to test only recognition without any learning part. To do this, we used a gesture alphabet that is even more self-explaining than Graffiti, namely simple directional strokes, also known as marks [11]. Participants' task was thus to recognize their *direction*.

We added stroke *length* as a second independent variable to determine the minimum distance required for reliable recognition. This will inform the design of future implementation and give us an idea of how small or large a device would have to be in order to offer reliable gesture output.

As a side effect, this would also give us a slightly enlarged 24 gestures repertoire of 8 directions \times 3 lengths, a language that presumably could support more elaborate applications than the 8 simply directional strokes. As discussed earlier, we think of gesture alphabets as a matter of learnability – an aspect that is investigated in Study 2 and 3.

Interface

Participants were seated in front of the longRangeOuija prototype described earlier. Participant's wrists were supported by the armrest shown in Fig.9.



Figure 9: After participants had received the output gesture, they entered the answer on the iPad.

Task

For each trial, participants started by tapping the screen. They then followed four verbally given instructions. On “close”, participants closed their eyes. On “touch”, participants pressed the screen and maintained contact with the screen. The system then performed one gesture from the repertoire of 24 different gestures, i.e., it actuated the participant's finger along the shape of the respective stroke. On “up”, participants moved their hand up in the air above the hand rest, and the system moved the device back into its initial position. On “open”, finally, participants opened their eyes and selected the gesture they felt they had received in the interface shown in Fig.9. They completed the trial by tapping the “next” button on screen.

Experimental design

The study used an $8 \times 3 \times 3$ within-subjects design, with two independent variables: *Direction* (the 8 compass directions) and *Length* (1, 2, and 4 cm). *Direction* and *Length* of the strokes were randomized for each of the 3 blocks. Each participant completed all conditions: 8 *Directions* \times

3 *Lengths* \times 3 blocks = 72 trials per participant. Participants performed 5min of training (30 trials) before the experiment. They had breaks (~1min) every 20 trials. All participants completed the study in 20min or less.

Participants

We recruited 12 right-handed participants (3 females) from our institution. They were between 21 and 25 years old. They received a small compensation for their time.

Hypothesis

The purpose of the study was to check if participants were able to recognize the direction of marks with different lengths. We thus picked a small length (1cm) as a baseline and hypothesized that strokes of this size would have a lower recognition rate than longer ones (2cm and 4cm).

Results and discussion

Fig.10 shows the results with a confusion matrix. Fig.11 provides summary data for the recognition of direction. ANOVA were performed on the average recognition rate across the three blocks. As expected, an ANOVA showed that participants recognized longer strokes more reliably ($F_{2,22}=10.73$, $p<0.001$). Post-hoc comparison tests (using a Tukey's HSD test) indicated that users performed significantly better with 2cm and 4cm marks than with 1cm marks. The 2cm and 4cm strokes resulted in recognition rates of 97.6% and 96.5%, respectively.

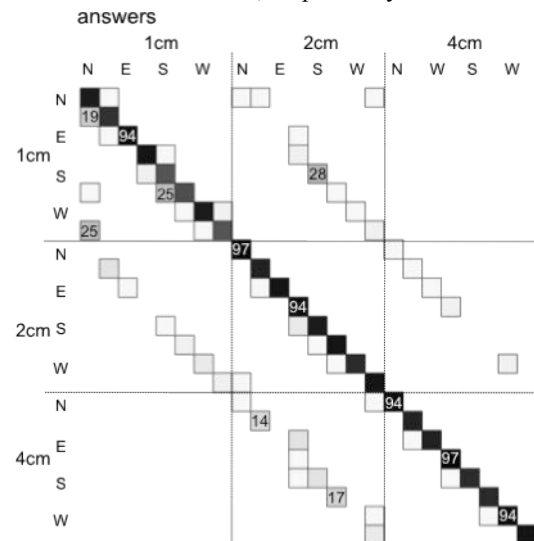


Figure 10: Confusion matrix for 8 directions & 3 lengths. The 2nd line reads as “19% of North-East strokes has been recognized as North strokes”.

The study thus showed that participants were able to recognize the direction of simple directional strokes. Results suggest using strokes of 2cm or more to reach a recognition rate of 97% for the eight compass directions strokes.

Preliminary results on the pocketOuija

We replicated the 1cm condition from this study on the pocketOuija with 6 new participants. This time, they were holding the device in their dominant hand and operated using the thumb, i.e., single-handed use, as shown in the last scene of the video. We found a recognition rate of

90.3% (compared to 86.8% reported with longRangeOuija). While these numbers are comparable to the numbers obtained with the longRangeOuija, additional testing is required to validate that the two prototypes are comparable.

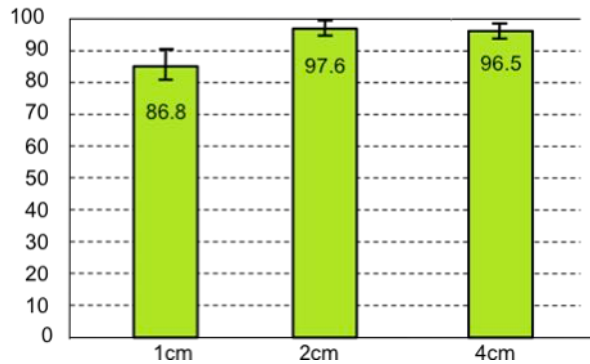


Figure 11: Recognition rates of direction for 1cm, 2cm, and 4cm strokes (Bars are $\pm 95\%$ confidence).

STUDY 2: LEARNABILITY OF SINGLE-CHARACTER

The purpose of this study was to validate our claims about the *learnability* of gesture output, in particular that users' knowledge of gesture *input* helps them understand gesture *output* ("transfer learning"). To tackle this, we picked a highly mnemonic gesture alphabet—the alphanumeric Graffiti alphabet. We hypothesized that training in gesture *input* would allow users to successfully recognize gesture *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 would be able to decode some of the gestures.

Task and Interface

The task and the interface were identical to the first study, except that the gesture alphabet was different. In the *letters* task, participants were presented with one of 26 output characters from the alphabetic portion of the Graffiti alphabet (Fig.12a). In the *digit* task, participants were presented with one of 10 output characters from the numeric portion of the Graffiti alphabet (Fig.12b).

Each trial took place the same way as in Study 1. We removed the need to touch the screen between each trial to reduce the time of the experimentation.

Second independent variable: Training vs. walk-up

Participants were assigned to one of the two groups: Participants in the *trained* condition received 20min of training in Graffiti *input* prior to the study by playing the training game shown in Fig.13. The game displayed a randomly chosen letter or digit at a time; participants responded by entering the respective Graffiti *input* gesture. For the first 5min of game play participants had access to a sheet with all Graffiti symbols; for the remaining 15min the sheet was taken away. The training period ended when the game score reached 100%. This score, displayed on the interface, was computed as the percentage of correct answers among the last 36 entries (all letters and all digits). Participants in the *walk-up condition* did not receive training with Graffiti, nor had they seen or used Graffiti before entering the study.

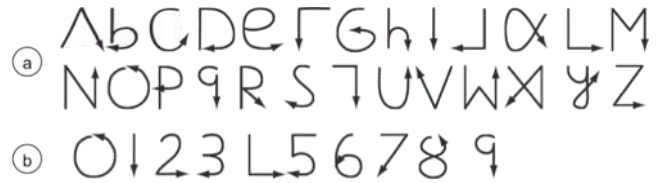


Figure 12: The Graffiti alphabet used in this study.

Producing the gesture database

To create the gestures, we drew them on the iPad and corrected them to create an ideal set (to scale, straight and round shapes). During the rendering, we also added short pauses on vertices, to ease the recognition of the gesture.

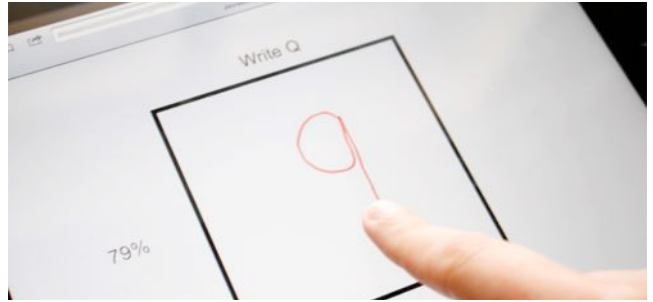


Figure 13: Participants in the trained condition were trained for 20min using this Graffiti game.

Experimental design

We used a $2 \times (26 + 10) \times 4$ between-participant subject design, with between-subjects variable *Training* (*trained* vs. *walk-up*) and within-subjects variable *Alphabet* (26 *letters* vs. 10 *digits*). Half of the participants started the experiment with *digits* while the other half started with *letters*. Within each category (*letters* or *digits*), characters were randomized. Each participant completed all conditions: 36 alphanumeric characters \times 4 trials = 144 trials. All participants received 5min (30 trials) of training before the experiment on a set of geometric figures (rectangle, triangle, and circle, *not* on Graffiti).

At the end of the study, we tested participants' understanding of the Graffiti *input* alphabet: An application on the iPad displayed a random letter or digit at a time; participants then entered the corresponding Graffiti *input*. We then checked *by hand* if the characters were correct or not. Given the unambiguous unistroke design of Graffiti, checking by hand was unambiguous and should come out the same when repeated by other experimenters.

Participants had breaks every 20 trials. They completed the study in 40min or less, plus 20min for the *trained* group.

Participants

12 new right-handed participants hired from our institution (1 female) between the ages of 21 and 32 participated in the study. They received a small compensation for their time.

Hypothesis

We hypothesized:

(H1) *Guessability*: even in the *walk-up* condition the recognition rate should be non-zero, because the mnemonic nature of the output gestures would allow participants to guess the gestures' meaning.

(H2) *Transfer Learning*: The *trained* group recognition rate would be higher than the *walk-up* group one because their previous exposure to gesture *input* would transfer to *output*.

(H3) *Reverse Transfer Learning*: assuming guessability, the experiment itself would function as training for gesture *output*, which we would expect to transfer back to gesture *input*. Participants in the *walk-up* condition should then be able to produce at least some *input* characters after the study.

We also expected to see differences between different characters recognition rates, such as the ambiguity of “QODGR” [17] and the limited guessability of “AFKT” [14].

Results: summary data

Fig.14 shows the overall recognition rate for the individual conditions. ANOVA were performed on the average recognition rate across the four trials.

An ANOVA found a main effect of *Training* on recognition rate ($F_{1,11}=20.77$, $p<0.001$), *Alphabet* ($F_{1,11}=43.96$, $p<0.001$) and the interaction *Alphabet* x *Training* ($F_{1,11}=14.42$, $p<0.003$). Post-hoc comparison tests (using a Tukey’s HSD test) indicated that *trained* participants performed significantly better than *walk-up* ones. We also found that *digits* were better recognized than *letters* in the *walk-up* group and that *letters* were better recognized in the *trained* group than in the *walk-up* group.

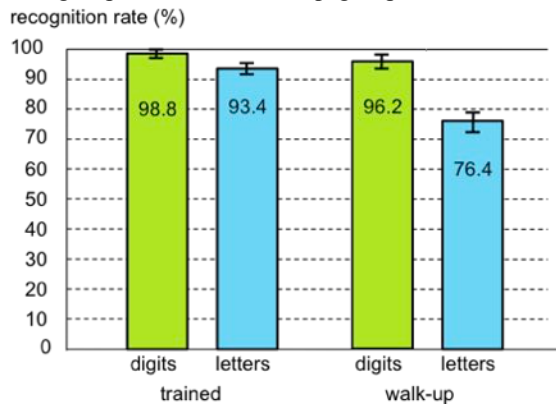




Figure 14: The recognition rate for *digits* and *letters*, *trained* and *walk-up* (Bars are $\pm 95\%$ confidence).

The substantial recognition rates in the walk-up condition, 96.2% for *digits* and 76.4% for *letters*, support our *guessability* hypotheses H1. The significantly higher performance of the trained over the walk-up group supports our hypothesis H2, i.e., that transfer learning had occurred.

Letters GHDEF harder to recognize

Participants from the *trained* group recognized 30 of all 36 characters with $>90\%$ accuracy. The confusion matrix in Fig.15 highlights the outliers.

An ANOVA found “QODGR” to be recognized significantly worse than the rest (88.8% against 95.8% recognition rate) ($F_{1,6}=9.36$, $p=0.02$). Ni [17] identified “QODGR” as being hard to enter eyes-free because they rely on *relative position features*. For instance, the Graffiti  is identical to the , except that the stroke ends slightly lower.

The digits recognition rate was 98.8%. An ANOVA found no significant differences between digits. Note that only “6” (87.5% recognition rate) was less well recognized than the others (100% recognition rate). It might again be caused by the relative position features between the “6” and the “0” since 12.5% of the “6” were recognized as a “0”.

These results suggest that the relative position features issue reported for *input* [17] holds for gesture *output*.

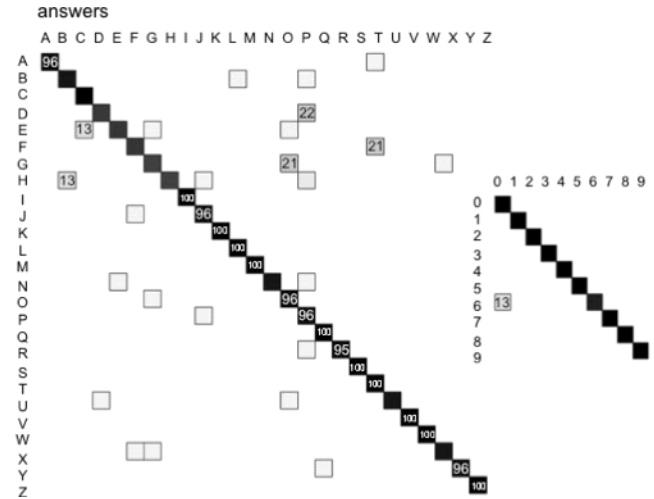


Figure 15: Trained group recognition of letters. 5th line reads as “13% of E has been recognized as a C”.

Letters KTFX harder to guess

Participants from the *walk-up* group recognized 24 of all 36 characters with $>84\%$ accuracy. The confusion matrix in Fig.16 highlights the outliers.

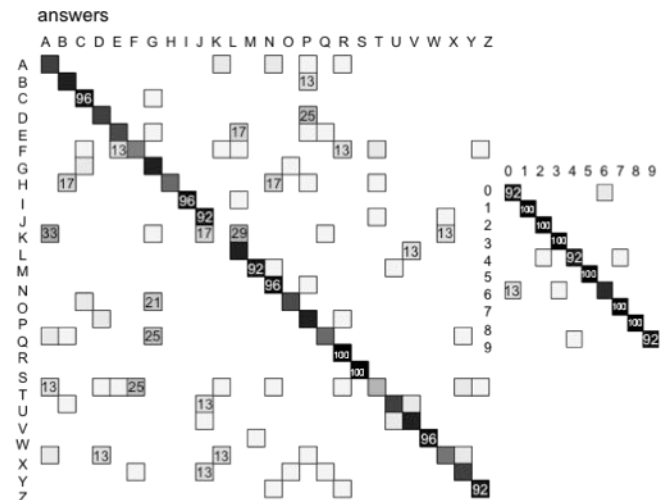




Figure 16: Walk-up group recognition of letters. 6th line reads as “13% of F has been recognized as a E”.

An ANOVA found “AKFT” (38.5% recognition rate) to be recognized significantly worse than the rest (87.3% recognition rate) ($F_{1,6}=31.73$, $p<0.002$). MacKenzie [14] identified “AKFT” as being hard to guess, because they do not match either an uppercase or lower case Roman letter. This causes these Graffiti characters to less resemble their counterparts, such as  for “A”,  for “K” etc.

“KTFX” had the least score (33.3% vs. 87.9% for the rest). An ANOVA showed significant differences between these groups ($F_{1,6}=201.99$, $p<0.001$). Using the confusion matrix Fig.16, we observed that: “K” was not recognized at all and misrecognized as “ALJX”, “T” was mainly recognized in “FA”, “F” in “ER”, “X” in “DK”. This suggests that these letters are not optimized for guessability. Contrary to results reported for input [14], “A” seems to be guessable with output.

The digits recognition rate was 96.2%. An ANOVA found no significant differences between digits. The “6” (83.3% recognition rate) was less well recognized than the rest (above 91.6% recognition rate). Using the confusion matrix on Fig.16, we observed that “6” was mainly recognized in “0”. As for *trained* group, it might be caused by the relative position features between the “6” and the “0”.

Results: walk-up group input

Fig.17 shows the *walk-up* recognition rate of letters at the end of the study. Results show 100% success for *digits*, 92.2% for *letters*, and 96.1% for letters without “KTFX” which were harder to guess during the study. This supports our hypothesis H3, i.e., that *reverse* transfer learning occurred. Unsurprisingly, the *trained* group performed well (100% recognition rate for *digits* and 99.7% for *letters*).

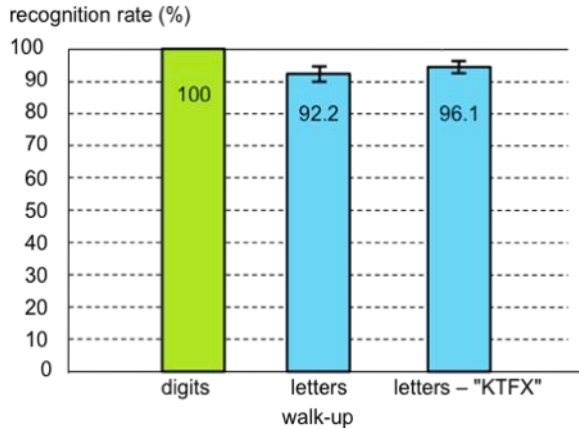


Figure 17: After the experiment, walk-up group participants entered 100% of digits and 92.2% of Graffiti letters correctly (Bars are +/-95% confidence).

Discussion

Study 2 finds support for our hypotheses, i.e., guessability, transfer learning, and reverse transfer learning. This validates our claim about the learnability of gesture output, in particular that there is a transfer of knowledge between I/O.

We also gained insights for the design of gesture alphabets: 1) the relative position features issue that stands for input as well as for output and that makes the letters “GHDEF” hard to guess; 2) the “F” was misinterpreted by its symmetrical opposite “T”; 3) the guessability issues that affect the letters “KTFX”; 4) some subsets of the language are very successful, in particular digits (98.8% recognition rate), even by walk-up users (96.2% recognition rate). This suggests that Graffiti digits form a particularly good alphabet to use when designing a gesture output language.

Finally, for the *walk-up* group, the post-experiment recognition rate for input *letters* (92.2%) is higher than the one of output *letters* (76.4%). We could have expected users to reproduce the same mistakes but results suggest that this is not happening. One explanation might be that, by forcing user to go through the entire alphabet, it reinforces the association between letters and gesture output.

STUDY 3: LEARNABILITY OF BI-GRAMS

The goal was to go investigate further the learnability of gesture output by testing *compound* messages made of two Graffiti digits. We hypothesized that training in input would allow users to recognize compound messages by transfer learning and by aggregation of input knowledge.

Task and Interface

The task and the interface were identical to Study 2, except for the tested alphabet consisting in pair of Graffiti digits. Participants received 5min of training in Graffiti digits *input* by playing the same training game.

Producing the gesture database

We merged each digits used in Study 2 and added short pauses on edges to ease the recognition of the gesture. The iPad having no vibrotactile unit, we played a vibration sound. We also created pauses at the end of the first digit and at the start of the second one to facilitate recognition.

Experimental design

We covered the 100 digits combinations using four groups of users. Each participant completed 25 trials and by aggregating users we had all numbers. We used a 4×25 between-participant subject design, with between-subjects variable *Group* (1 to 4) and within-subjects variable *Bi-grams* (25 pairs). Within each group, participants tested the same subset of *Bi-grams* (group number + $n*4$), but they were randomized. All participants trained 5min (30 trials) before the experiment on compound figures (rectangle-triangle, triangle-circle and circle-rectangle). Participants completed 5min of training and 15min or less of experimentation.

Participants

8 new right-handed participants hired from our institution (2 females) between 22 and 25 participated in the study. They received a small compensation for their time.

Hypothesis

The purpose of this study was to check whether participants were able to recognize bi-grams. Given the high recognition rate for single-character digits found the study 2, we expected that users would aggregate their *input* knowledge to learn compound digits as output, i.e. the recognition rate of compound digits to be non-zero.

Results and discussion

An ANOVA found no significant effects between *Group*. Results show that participants familiar with Graffiti *input* but with no training in *output* recognized compound *output* with 90.5%. This supports our hypothesis that participants are able to aggregate their *input* knowledge of single-character to recognize compound messages as *output*.

CONCLUSION

We have presented the concept of gesture output, a non-visual, non-auditory output technique that actuates the user's finger in a form of a 2D gesture. Gesture output allows leveraging people's existing mnemonic associations, such as from doodling, scribbling, and handwriting, making the language learnable by transfer and even guessable. In this paper we demonstrated their feasibility on the longRangeOuija and showed that recognition rates are high, especially for carefully selected gesture sets such as 2cm and 4cm strokes (97.6% and 96.5% recognition rates) or digits (98.8%), even in the context of walk-up use (96.2%).

Graffiti *input* has not caught on the market as of today, and there is some evidence that the up-front cost of learning is in part responsible for this. Our studies show, however, that gesture output does not only bypass this hurdle by being guessable, but also teaches Graffiti as a side effect. Future systems with a symmetric I/O system may thus even simplify the introduction of Graffiti input. Finally, we think that this technology can be used for other applications. As future work, we have started to investigate how to combine gesture output with spatial input and with visual output.

AKNOWLEDGMENTS

We thank J. K. Singer for lending us two PHANTOMs, and the University of Bristol for letting A. Roudaut finish this work during the time she was newly employed.

REFERENCES

1. Bragdon, A., Nelson, E. Li, Y., Hinckley, K. Experimental analysis of touch-screen gesture designs in mobile environments. *CHI'11*, 403-412.
2. Brave, S., Dahley, A. inTouch: a medium for haptic interpersonal communication. *CHI EA'97*, 363-364.
3. Brewster S., Brown, L.M. Tactons: structured tactile messages for non-visual information display. *Auic'04*, 15-23.
4. Chubb, E.C., Colgate, J. E., Peshkin, M. A. ShiverPaD: A Glass Haptic Surface That Produces Shear Force on a Bare Finger. *Haptic'10*, 189-198.
5. Dang, T., Annaswamy, T.M., Srinivasan, M.A. Development and evaluation of an epidural injection simulator with force feedback for medical training. *Medicine Meets VR*, 97-102.
6. Enriquez, M., MacLean, K. The haptic editor: a tool in support of haptic communication research. *Haptic'03*, 356-362.
7. Feygin, D., Keehner, M., Tendick, F. Haptic Guidance: Experimental Evaluation of a Haptic Training Method for a Perceptual Motor Skill. *Haptic'02*, 40-47.
8. Geldard, F.A. 1960. Some neglected Possibilities of Communication, *Science* 131, 1583-1588.
9. Goldberg, K., Wallace, I. Denta Dentata, SIGGRAPH '93.
10. Israr, A. and Poupyrev, I. Tactile Brush: Drawing on Skin with Tactile Grid Display. *CHI'11*, 2019-2028.
11. Kurtenbach, G., Buxton, W. Issues in Combining Marking and Direct Manipulation Techniques. *UIST'91*, 137-144.
12. Li, K, A. Designing easily learnable eyes-free interaction, *Ph.D. Thesis*. 2009.
13. Li, K.A., Baudisch, P., Griswold, W.G., Hollan, J.D. Tapping and rubbing: exploring new dimensions of tactile feedback with voice coil motors. *UIST '08*, 181-190.
14. MacKenzie S., Zhang, S.X. The immediate usability of graffiti. *GI'97*, 129-137.
15. Morris, D., Tan, H.Z., Barbagli, F., Chang, T., Salisbury, K., Haptic feedback enhances force skill learning. *WHC'07*, 21-26.
16. Mullenbach, J., D. Johnson, J. E. Colgate, and M. A. Peshkin, ActivePaD surface haptic device", *Haptic'02*. 414, 2012.
17. Ni, T., Baudisch, P. Disappearing mobile devices. *UIST'09*, 101.
18. Noma, H. Miyasato, T., Kishino, F. A palmtop display for dextrous manipulation with haptic sensation. *CHI'96*, 126-133.
19. Pangaro, G., Maynes-Aminzade, D., Ishii, H. The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. *UIST'02*, 181-190.
20. Patten, J., Ishii, H. Mechanical constraints as computational constraints in tabletop tangible interfaces. *CHI '07*, 809-818.
21. Poupyrev, I., Maruyama, S., Rekimoto, J. Ambient Touch: Designing tactile interfaces for handhelds. *UIST'02*. 51-60.
22. Saponas, T.S., Harrison, C., Benko, H. PocketTouch: through-fabric capacitive touch input. *UIST '11*, 303-308.
23. Sato M., Development of String-based Force Display: SPIDAR. *VSM'02*. 1034-1039.
24. Seo, J., Choi, S. Initial study for creating linearly moving vibrotactile sensation on mobile device. *Haptic'10*, 67-70.
25. Srimathveeravalli G., Thenkurussi, K. Motor Skill Training Assistance Using Haptic Attributes. *WHC '05*, 452-457.
26. Tan, H. Z., Durlach, N., Rabinowitz, W. Reed, C.M, Santos, J. Reception of Morse Code through Motional, Vibrotactile and Auditory Stimulation. *Perception & Psychophysics*, 59, 7, '97.
27. Teo, C., Burdet, E., Lim, H. A robotic teacher of Chinese handwriting. *HAPTIC'02*, 335-341.
28. Wang, D., Rossi, M., Tuer, K., Madill, D. Method and System for Providing Haptic Effects. *US Patent* 20060209037.
29. Wang, D., Tuer, K., Rossi, M., Shu, J. Haptic Overlay Device for Flat Panel Touch Displays. Demo at *Haptic'04*.
30. Weiss, M., Schwarz, F., Jakubowski, S., Borchers, J. Madgets: actuating widgets on interactive tabletops. *UIST '10*, 293-302.
31. Weiss, M., Wacharamanotham, C., Voelker, S., Borchers, J. FingerFlux: Near-Surface Haptic Feedback on Tabletops. *UIST '11*, 615-620.
32. White, N., Back D. Telephonic Arm Wrestling, *Strategic Arts Initiative Symposium* (Salerno, Italy, Spring 1986).
33. Williamson, J., Murray-Smith, R., Hughes, S. Shoogole: excitatory multimodal interaction on mobile devices. *CHI'07*, 121-124.
34. Winfield, L., Glassmire, J., Colgate, J. E., Peshkin, M. T-pad: Tactile pattern display through variable friction reduction. *WHC'07*, 421-426.
35. Wobbrock, J.O., Myers, B.A., Kembel, J. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. *UIST '03*, 61-70.
36. Yang, Y., Zhang, Y., Hou, Z., Lemaire-Semail, B. FingViewer: A new multi-touch force feedback touch screen. *ICMI'11*, 837-838.
37. Yatani, K., Truong, K.N. SemFeel: a user interface with semantic tactile feedback for mobile touch-screen devices. *UIST '09*. ACM, 111-120.
38. Yokokohji, Y., Hollis, R. L., Kanade, T., Henmi, K., Yoshikawa, T. Toward machine mediated training of motor skills-skill transfer from human to human via virtual environment. In *RO-MAN 1996*, 32-37.