SNOUT: One-Handed use of Capacitive Touch Devices

Adam Zarek, Daniel Wigdor, Karan Singh
Department of Computer Science, University of Toronto
zarek@cs.toronto.edu | dwigdor@dgp.toronto.edu | karan@dgp.toronto.edu

ABSTRACT
SNOUT is a novel interface overlay designed for occasional no-hand or one-handed use of handheld capacitive touch devices. Inspired by the desire to use these devices in scenarios where visually focused bimanual input is awkward, we performed a pair of studies intended to evaluate the potential of the nose to provide touch input. These studies influenced our design principles, resulting in the construction of a 'nose mode' which enables object selection, continuous parameter control, and speech-based text entry. Selection is accomplished via a nose tap, using a colour overlay and peripheral colour feedback to correct mistakes. The other two techniques are activated by a nose tap, but use the accelerometer to control parameters and speech-to-text for text entry. An evaluation of SNOUT shows it to effectively render handheld capacitive touch devices operational in scenarios where they are presently unusable.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design**,** Human Factors

Keywords: Touch screen, smartphone, tablet, mobile devices, nose input, UI overlay, accessibility

1. INTRODUCTION
Multi-touch smartphones and tablet devices offer rich interaction with a broad base of applications. Interacting with these devices can be cumbersome in scenarios where hands are either preoccupied or covered. Browsing a map while wearing gloves in cold weather, making or receiving a phone call with dirty hands at a construction site, checking an online recipe, and following assembly instructions with ingredients, a tool, or furniture part in one hand, all are impossible with current sensors. SNOUT addresses the operation of handheld capacitive touch devices in these situations.

A limitation of capacitive touch sensing is its inability to sense non-conductive objects. Wearing gloves for work or cold weather insulates the skin's capacitance. This has led to creative solutions, such as the use of frozen meat [35] (and more traditional styluses), modified gloves with conductive fingertips [24], and even a nasal prosthetic to facilitate nose-pointing [31]. While innovative, carrying such accessories can be a burden and may not address all scenarios.

Our solution, SNOUT, was designed through extensive iterative design. It addresses these limitations with a judicious mix of touch, haptics, speech, and peripheral vision. We observed that in most scenarios the nose is uncovered and available for use as an input mechanism. However, mobile user interfaces are not optimized for nose input, nor is it reasonable to expect developers to create nose-optimized versions of their applications. Our is an application-independent overlay, similar to an accessibility mode, enabling runtime optimization of applications for nose use. The design of such an overlay poses three significant challenges. First, nose input suffers from occlusion and accuracy issues that make it similar to but distinctly different from *fat-finger problem* [28]. Second, the ergonomics of interaction are reversed: rather than moving the touching object to the device, the device is typically moved to the touching object. Third, visual feedback is compromised with the nose touching the device, allowing only the use of peripheral vision.

On its face, nose-based interaction might appear too ridiculous to require further consideration. Our selection of the nose, however, is validated by an open-ended study intended to elicit users' suggestions for interacting with their phones in our target scenarios. 60% of respondents suggested the use of their nose for input. We conclude that there is a definite willingness of individuals to use their nose, yet current interfaces fail to support this mode of operation. Based on early testing of nose operation of traditional phone UI, SNOUT was designed based on 5 principles:

1. *Preserve application UI layouts.*
2. *Minimize the number of nose-taps required.*
3. *Minimize nose sliding on the screen of the device.*
4. *Eliminate the need for focused visual feedback while touching the screen.*
5. *Provide means of error mitigation.*

These led to the design and construction of a nose-friendly overlay which enables nose-based selection, touch-free continuous parameter specification, and nose-initiated speech-to-text. Not all features of SNOUT are applicable in every situation; however the final design incorporates, customizes, and enhances previous efforts intended to enable easier use of touch screen phones [8, 15, 32]. SNOUT increases the selectable area of an object by a factor 9, without altering its size. By leveraging accelerometers , SNOUT supports a touch-free method for specifying a continuous 1-D and 2-D parameters. Finally, SNOUT marries speech-recognition and nose-input to support text-entry in a rapid and fluid manner.

In this paper, we present the results of a survey which demonstrates that potential users think of the nose as a reasonable appendage for input in certain scenarios. We then review related work, including previous nose-focused efforts, and demonstrated limitations of nose input to touch devices. Next, we present the results of a study which quantifies the limitations of precision in nose-based selection, while collecting user feedback on nose input. We then present our design principles, derived from this study, and our design for SNOUT. Finally, we present a third study which demonstrates how SNOUT overcomes users' reluctance to perform nose input.

Figure 1. SNOUT enables one-handed use of capacitive smartphones while wearing gloves.

AVI '12, May 21-25, 2012, Capri Island, Italy

2. Medium Elicitation Survey

Before delving into the mechanics of developing a nose-specific UI, we wanted to know whether users would even entertain the notion of using their nose for input. We conducted a survey that provided 6 scenarios where conventional finger-based input was not possible. The scenarios mirror our motivating examples where the user is wearing gloves, is cooking with dirty hands, or his hands are occupied. Participants were then asked to describe how they would give input to their device in the given scenario.

We recruited 15 participants from a local University (13 male) to take our survey. 12 of the participants own a touch device. Participants were not compensated for their participation.

2.1 Survey Results

Each question in the survey walked through a situation where conventional touch-based input was impossible, and asked participants open-ended questions about how they would overcome this difficulty. Results were coded based on input method. The results for selection-based interactions are summarized in Table 1. Some respondents offered multiple answers so the totals do not sum to 15.

The results demonstrate that faced with situations where conventional inputs are not available, users are resourceful and are willing to interact with other readily-available body parts. 86% of participants answered the non-conventional appendages of nose, toe, or elbow at least once. Although in each scenario the nose was suggested by only 4 participants, these were rarely the same 4 participants. In fact, the nose provided the broadest coverage across participants, with 60% suggesting its use in at least one scenario. We thus focused our design efforts on facilitating nose-based input, and on ensuring broader coverage of input enabling our motivating scenarios. To facilitate this design, we first reviewed related work.

3. Related work

Three areas of related work were helpful in the design of SNOUT. These were works in nose input, in multi-modal mobile device input, and techniques which improve the precision of touch input.

3.1 Nose Appendage

Earlier projects have utilized nose tracking. Indeed, Gorodnichy demonstrated that the robustness of facial tracking algorithms can be significantly improved by tracking the nose [9]. Henry et al. demonstrated the use of nose tracking to extend the capabilities of virtual reality [13]. More recently, the Nouse system used nose tracking to control a mouse pointer, intended for users with disabilities [10]. While inspirational for their nasal focus, the goal of our project is to enable single-handed use of existing smart phones using their existing UI and sensing capabilities.

3.2 Touch-Free Mobile-Device Interaction

Several projects have examined methods by which content may be manipulated without relying on the touch screen. Modern smartphones are equipped with a host of sensors. Previous work has examined the use of accelerometers, proximity sensors, gyroscopes, cameras, and magnetometers for interaction with a mobile device.

Table 1. Survey results by state of hands within the scenario, summarized by input modality.

Hands are:	***Gloved***	***Dirty***	***Occupied***
User Action	Remove gloves (10) Nose(4) Accessory (3)	Nose (4) Elbow (4) Knuckle (1) Accessory (1) Do nothing (4)	Toe/foot (10) Nose (4) Don't know (4)

Smartphones use infrared proximity sensors to disable the screen when held to the user's ear. Earlier projects have used these devices to enable gestural input. Both Hoverflow, and SideSight used arrays of proximity sensors to enable the detection of gestures above a worn device [23]. Though interesting, their reliance on multiple proximity sensors would require modifying the phone's hardware. Our goal is a software-only solution. Magnetic tracking has been demonstrated by MagiTact and MagiWrite, however, their reliance on a megnetic marker requires an accessory – our goal is accessory-free input [19, 20].

Motion-sensing, via one more of the device's accelerometers, magnetometers, and camera, can enable touch-free interaction of a mobile device. These have been demonstrated for continuous input, such as scrolling, and for discrete gestures, such as tapping on the back of the phone (e.g. [2,12,14,17,27]). Recently, Hinckley and Song presented *sensor synaesthesia*, which combines motion sensing with use of the touch screen [15]. We draw from several of these works to inform our designs.

3.3 Precise Pointing

At first glance, pointing with the nose seems similar to pointing with the finger. Techniques such as OffsetCursor, FluidDTMouse, Shift, and those of Benko et al. each demonstrate improved precision by offsetting the touch location and cursor [4,7,26,29]. Each of these requires a closed feedback loop following the moment of touch, before the user lifts his finger from the device. In the case of nose input, however, the phone's proximity to the eyes dramatically reduces the bandwidth of this channel. Further, due to their reliance on take-off selection [26], they are not suitable for modern phones, which rely on 'touch and drag' primitives, such as modifying the volume or seeking within a video.

Thus, in designing SNOUT, we drew inspiration from these techniques in their use of feedback, but note that this feedback must be of a significantly different form.

Enhanced Area Cursors are also instructive. They include several two-step techniques: the user selects an area which may include multiple targets, at which point the UI is altered. In a second step, the user selects from among targets in the modified UI [8]. Escape is similar, but avoids the modification step using flags to preview the second step [34]. Like other precise pointing techniques, these too rely on a closed visual feedback loop with the user. From this work we draw the inspiration of modifying the spatial arrangement of targets following touch, with the caveat that this must be limited to changes which can be conveyed to the user given the limited feedback channel. Also of interest are techniques which improve precision by moving the feedback to *before* the moment of touch [30,33]. While informative, their reliance on hover sensing makes them unsuitable for modern touch phones.

While many of these projects have proven informative, it is clear that the unique use of a touch screen with the tip of the nose would require further investigation prior to design. To that end, we conducted a study intended, in part, to determine users' accuracy at selecting targets with their nose.

4. Initial Study

Buoyed that users would indeed interact with their touch devices in unconventional ways, we conducted a study to assist in developing design guidelines. The overarching question was: *are current smartphone UI's suited to nose-based interaction?* This was further divided into: *would users be willing to use them as is*, and: *how accurately can a user hit a target with the tip of his nose in each of the usage scenarios we identified?* These were answered in the forms of qualitative feedback, and quantitative measures.

4.1.1 Quantitative Measures

We picked gloved mobile phone use and one-handed tablet use to represent the spectrum of scenarios. To enhance external validity, we further divided the experiment by usage scenario, and had the user perform pointing tasks with each of these apparatuses both while stationary and while walking. On the basis of our study questions, we formed three hypotheses. The first of these aligns with traditional effects of pointing studies:

H1: Location *and* size *of the target would have an effect on* accuracy *on the phone.* Location *would not have an effect on* accuracy *for the tablet scenario.*

The effects anticipated by H1 are distinct from what might be predicted with a straight-forward application of Fitts' Law because the distance of the selection task was not controlled. While we could have elected to control the position of the user's head and hands before each selection, we instead chose to allow the users to find a comfortable posture, enhancing external validity of our results. Thus, the effects we anticipated relate to the difficulty of holding the device and what might be called the *fat-nose* problem [28]. This is reflected in our hypothesized asymmetry of the effect of *location* – we expected the effect of *location* to be due to ergonomic issues of reaching areas of the phone while holding it, not present for the larger tablet.

In addition to whether or not users hit the target, we also wished to measure two additional values: the distance from the user's land-on point to the target (*offset*), and the amount of *jitter* – unintended movement picked-up by the accelerometer. We hypothesized for each of these:

H2: *Target* size *will have an effect on* offset*.*

H3: Posture *will have an effect on* jitter.

Here, *posture* denotes whether the user was stationary or walking. With estimates for the effects of *error rate*, *offset*, and *jitter*, we are able to determine worst-case scenarios for these effects. In all cases, our designs are based on the *worst case* of these effects.

4.1.2 Qualitative Measures

The second part of our study was to answer the question of whether users would use existing UI techniques to engage with their noses, and if not, why? We solicited user preference and other qualitative feedback about the use of their nose to operate a mobile GUI. We expected users to have mixed reactions to the use of their nose to make selections, but beyond this, we wished to develop concrete guidelines that would allow SNOUT to be optimized not only with respect to performance, but with respect to user preference as well.

4.2 Experiment Design

Our experimental task consisted of tapping targets of 3 sizes at different locations on a screen using the tip of the nose. Participants were instructed to both land-on and take-off within the target bounds, but a target was "hit" if the user was within the square for either the touch-down or the touch-up event. This is because both are used as selection methods on popular platforms, and because our measures of *offset* and *jitter* would account for any disparity.

In keeping with our *scenarios*, the experiment was conducted with both a smartphone (held wearing winter gloves) and a tablet (with bare hands). Our small, medium and large targets were all squares with side lengths of 4mm, 8mm, and 12mm on the phone and 5mm, 11mm, and 15mm on the tablet (*target size*). Sizes were based on sizes of common UI elements. The screen was divided into a 3x3 grid of 9 different *target locations* and each target size was displayed in location. Finally, in order to gauge the impact of users' movements, participants alternated between remaining stationary and walking across a room (*posture*).

The ordering of the 4 combinations of *scenario* and *posture* were controlled with a Latin Square. We randomly ordered the 27 different combinations of target *size* and *location*, and repeated with another random order 3 times for each *scenario* x *posture* pairing. The design of the experiment can be summarized as:

8 participants
x 2 *scenarios* (gloved phone, tablet)
x 2 *postures* (stationary, walking)
x 3 *target sizes*
x 9 *positions*
x 3 repetitions
= 2592 total selections

Users began with a training set of 27 trials with both the phone and the tablet to familiarize themselves with the weight and positioning of the device. After each block, participants could take a break.

Following completion of the quantitative portion, participants were asked to describe the different tasks they could imagine performing with their nose, and for how long for a given task. Participants were also instructed throughout the experiment to share their thoughts and reactions to using their nose to interact with the device.

4.3 Apparatus

Two devices were used. The first was an HTC Desire HD phone. It has a screen size of 4.3 inches, a 480 x 800 screen resolution and weighs 164 grams. The second device was a Samsung Galaxy Tab tablet. It has a screen size of 7", resolution of 1024 x 600 and weighs 380 grams. The software was implemented using the Android Software Development Kit, and ran entirely on the devices.

4.4 Participants

We recruited 8 paid participants (7 male) ranging in age from 20-24 from the university community. All the users had extensive prior experience with mobile touch devices. Participants were paid $20 for their participation.

4.5 Quantitative Results

A GLM was used to measure the effects of the various factors (outliers removed). In cases of non-sphericity, Greenhouse-Geisser was used. Results are split by *scenario*.

Error rate was encoded as a Boolean (hit on first attempt, missed on first attempt) and counted across the three repetitions of a given set of conditions. For gloved phone use, both target *size* and *position* showed significant effects ($F_{1.358,9.503}$=83.406, $p < .001$, $F_{3.261,22.830}$=3.309, p =.035). There was not a significant interaction of *size* x *position*, indicating that difficult locations were harder to hit no matter the target size. There was no effect for *posture*, indicating that remaining stationary did not ease target selection when wearing gloves.

On the tablet, both *size* and *posture* were significant, while *location* was not ($F_{1.384,1.448}$=74.717, p<.001, $F_{1,7}$=16.155,p<.01). This indicates that the larger size of the tablet ensures that there are no particularly "tricky" target locations. There was no interaction of *posture* x *location*, indicating that walking makes targets uniformly harder to hit on a tablet, no matter their size. Results are shown in Figure 2.

As regards *jitter*, on the phone, only size had a significant effect ($F_{2,14}$=3.209, p = .033). There was no significant effect for *location* or *posture*, nor was there a significant effect for *size* x *posture*. This indicates that for smaller targets, users performed more corrective movements, but no more for any given target location. On the tablet, both *size* and *posture* had a significant effect on *jitter* ($F_{1.91,8.339}$ = 5.135, p =.048, $F_{1,7}$ = 22.411, p =.002). No interaction effects were present. In even the worst cases, the mean value for *jitter* was within the target size. Thus, once the user lands their nose on the device, they are able to keep it on the same location for the take-off.

In *offset*, we focused on accuracy with respect to the nose-down events. As expected, both target *size* and *location* had a significant effect on *offset* for the phone ($F_{1.106,7.744}$ = 128.896, p < .0001, $F_{1.125,7.873}$ = 59.397, p < .0001). The effect of location was not significant. Taken as a function of the target size, our results show that the mean distance from the nose land-on point to the centre of the target, under the worst conditions, is 0.43 STD Dev = .40) times the size of the target. This indicates that a selection primitive should require users to land only within 1.5 target widths of the center of the target. Furthermore, since *jitter* was within the target size, only land-on accuracy need-be corrected, while take-off location should be anywhere within the bounds of the target.

4.6 Qualitative Results and Observations

Participants were asked whether they would be willing to use their nose if they were wearing gloves or if their hands were otherwise unavailable. 7 of 8 indicated they would be comfortable using their nose for simple tasks requiring no more than a few interactions. One user commented that "in the beginning it was difficult and frustrating, but then I got the hang of it", while another 2 said that by the end it felt the same as using their fingers. However, all of the 7 tempered their enthusiasm by listing interactions they would not use. Two users were apprehensive about sliding their nose on their phone screen because it would get dirty. Five users would use their nose for large items that don't require high-precision. One user said he would use it for selection, but not for actions such as seeking within a video because he can't see the screen as he slides. Only one person said he would use his nose for text entry as currently designed, and even then he felt for no more than 10 characters.

Users devised their own ergonomic techniques that they claimed enhanced their accuracy. Two users found that they were more accurate when they tried to touch the phone with the underside of their nose as opposed to the tip. That aligns with the comments from a third participant that "felt like I was hitting above the target for most of the time." Participants were cognisant of the accuracy challenges of using their nose; they expressed annoyance with the smallest targets, especially when the targets were touching any of the screen edges, consistent with quantitative results

Interestingly, when pacing back-and-forth, two participants were so engrossed in their task they collided with the camera recording the study. Other participants commented that they are concerned about getting distracted while walking outside and putting themselves, or others, in danger. Whether this is a greater problem with nose input than with traditional phone input is an area for future study.

From this, we conclude that, while users are ready and willing to use their noses, the current design of smartphone software does not facilitate this, because it requires swiping, the targets are too small, and there is no nose-suitable feedback mechanism. We thus derived several principles which guided our design exercises.

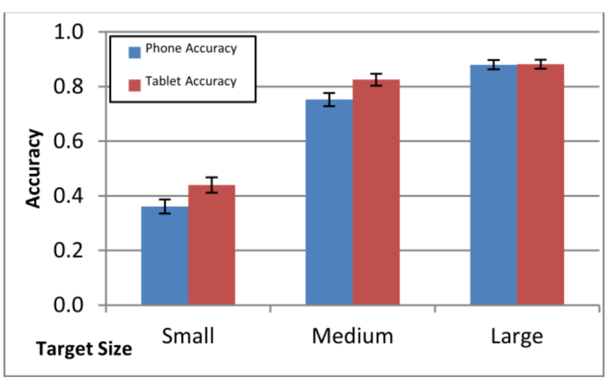

Figure 2. Accuracy as a function of target size for each scenario: *phone*, and *tablet*. Confidence intervals are within scenario.

5. Design Principles

Based on the results of our study we developed a set of principles for the design of nose-based interactions.

5.1 Minimize the Number of Nose Taps

In conventional finger-based touch input, a user can initiate a sequence of touch events in quick succession. More importantly, throughout the input sequence users can maintain focus on the screen and visualize any feedback offered by the interface. With nose-based touch input the user must disengage from the screen after each touch event to recognize and respond to visual feedback, or to acquire the subsequent target. Users also shared their unwillingness to engage in a large amount of nose-tapping. Of 8 participants, 5 indicated they would be willing to give fewer than 10 successive nose taps.

5.2 Minimize On-Screen Nose Sliding

For perceived hygienic reasons, 2 participants were reluctant to swipe their noses across their screens. When given a choice, users were more willing to use a tap than make a wiping gesture. Because modern smartphone UI's contain a great deal of 'dragging' gestures, such as for adjusting volume, seeking within a video, scrolling a list, and browsing a map, this principle gives rise to the need to design alternative methods for continuous parameter specification.

5.3 Provide Feedback Without Eye Fatigue

In the normal case with nose-based input, a user will generally focus on the tip of their nose to view what is the currently selected object on the screen. The result is a general impairment of visual focus, or even worse it can lead to eye fatigue and headaches after an extended period of time. Consequently, to avoid this case of visual impairment when operating with the nose we cannot rely on visual cues other than unfocused peripheral vision.

5.4 Error Mitigation

The results of our pilot study show that users are likely to make a mistake in their initial land-on location. Thus, our design principle includes three elements. First, make it possible to detect an erroneous land-on event. Second, provide the a means to correct the action before take-off without the need to make errors and then 'undo'. Third, provide a mechanism by-which the input can be cancelled entirely. There is a need for innovation in this area, since existing error-correction methods second the *drag* action as an error correction method, without providing an alternative method for continuous parameter specification. This makes them unsuitable for modern smartphones [4,7,26,29].

5.5 Preserve Application UI Layouts

While no study participants explicitly mentioned maintaining the pre-existing UI layout, we include it as a design principle because it would be unrealistic to expect application developers to design either a nose-optimized or a speech UI version of their applications [32]. Consequently, our goal is develop a design that can be overlaid on top of existing applications, without significant modification, or which could be integrated into a mobile operating system. The design of SNOUT applies these principles to create an overlay enabling nose-based input.

6. SNOUT Design

In this section we begin by identifying 3 of input primitives found in touch-enabled smartphones and tablets. We then present our multi-modal replacements which satisfy our design principles by relying on combinations of touch, motion, and speech input techniques. SNOUT builds on previous work in the design of each of these primitives. Our contribution lies in the customization of these prior techniques to enable nose input, and on their integration into a solution which enables *all* of the device's primitives.

1. *Selection* (button tapping, checkbox checking, etc.)
2. *Continuous parameter specification* (scroll, pan, etc.)
3. *Text entry*

Text entry could be enabled with selection [6], continuous specification [25], or a mix of the two [21], however it is included here as a primitive because it requires a lot of nose tapping, and requires selecting a lot of small targets.

We review our design for each of these 3 in turn, and then present applications controlled using SNOUT.

6.1 Selection

As we found in our experiment, small targets are particularly difficult to select with the nose. Our selection technique is intended to expand the area of a target, and to provide an opportunity for corrective adjustment of the selection before committing, employing take-off to enable precise selection with the aid of a feedback loop [4,7,26,29]. Nose selection is unique in that we must modify the form of feedback for viewing close to the eyes. To achieve increased precision, SNOUT first divides the screen into Voronoi regions, where every pixel on the screen is attached to the closest target, similar to previous work [3,11].

Figure 4 steps through an example of SNOUT selection. Initially, the user puts the phone into nose-mode, Figure 4 **Error! Reference source not found.**(a) where each object on the screen is coloured with a cyclic 4-color pattern that ensures no adjacent targets are like-coloured, using an algorithm derived from [18]. When the nose touches the screen as in Figure 4 (b), we determine the touched active region and the following 2 actions occur. (1) Adjacent regions expand to fill the entire screen, allowing rapid and coarse selections, similar to a marking menu [22], and (2) A band of colour around the perimeter of the screen conveys to the user what active region is currently selected. The user can correct the selection by sliding to the desired target, Figure 4 (c). The selection is then made by lifting the nose from the screen. There are many subtle details to the design of SNOUT selection and we explain each in turn.

Consistency. An important result of cyclic colouring is that regardless of where the user touches down, the selected object is always uniquely coloured within the adjacent colours. In a grid-layout, such as the soft keyboard and number-pads, every object of a given colour will have the same adjacent colourings. Thus, if a user touches down and sees a certain colour, the relative positions of the other colours are always consistent, enabling rapid corrections.

Figure 3. Text-entry uses the selection primitive to select tokens and to type replacements. Here, the user has their nose down on the "I", adjacent targets have expanded, and the colour is reflected along the perimeter of the screen.

Increased Accuracy. For a user to be able to select a target with SNOUT, the initial nose-down action must land within the target's Voronoi region or an adjacent region. For grid layouts, this results in an increase by a factor of 9 over which an objects can be normally selected (or 6 for objects at the edge of the screen). As shown from our initial pilot study, an increase the selectable area by a factor of 9 results in an error rate of less than 5% in the worst case.

Error Avoidance. One of our design principles required a cancellation mechanism that allows the user to terminate the current interaction prior to committing. This is achieved by sliding the nose off any side of the device.

6.2 Text Entry

Text entry using a soft keyboard can be viewed entirely as a sequence of single-touch selection gestures [6]. However, our study participants were reluctant to repeatedly hunt-and-peck at a keyboard. Consequently, SNOUT provides an alternative method to facilitate text entry.

Normally, touching a text box gives it focus and causes a soft keyboard to appear. With SNOUT, a speech recognition system is engaged. The user can immediately begin to speak, with automatic recognition of the end of the speech. The captured text is then placed in the text box. Individual words can then be selected for editing using our selection technique. To ensure easy selection, short words such as "I" and "a" are expanded to a minimum target size of 4mm. Cursor placement for correction is facilitated using our selection technique. Figure 4 illustrates.

6.3 Continuous Parameter Specification

Continuous parameter actions with either one degree-of-freedom, such as scrolling, controlling a slider, or zooming, or two degrees-of-freedom, such as panning an image, both require a user to have an unobstructed view of the screen in order to receive feedback. Further, traditional techniques for small target selection break down for this primitive – it is impossible to use Shift to select and then manipulate the thumb of a slider [29]. The obvious choice for continuous input is to use sliding as an input gesture. However, this is problematic for two reasons. First, our design principles included a minimization of nose sliding. Second, we wished to preserve sliding for corrective actions for target selection. Thus, we employed an alternative modality.

Similar to Sensor Synesthesia [15], SNOUT leverages tilt, provided by the phone's accelerometer and gyroscope, in combination with touch input to control continuous parameters with one or two degrees-of-freedom. We devised an input gesture to initiate this mode called touch + tap. The user holds their nose to the screen to the desired manipulation target and then taps the back of the device. The device responds to the user with a short vibrating pulse and enters manipulation mode.

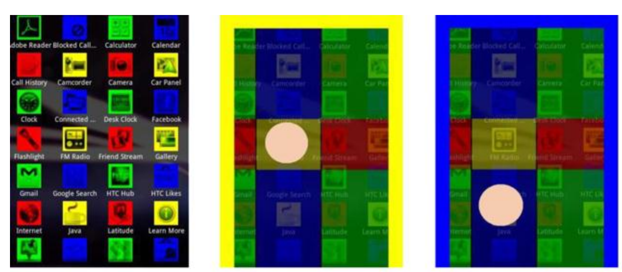

Figure 4. (a) SNOUT's cyclic colouring of a grid layout. (b) On nose-down, adjacent targets expand to fill the screen and the colour of the currently selection appears on the periphery. (c) Corrective sliding is reflected by perimeter colour.

Once in manipulation mode, the user can move the device away from his face, and tilt to control the direction and speed of manipulation. Two-degree of freedom manipulation, such as panning a map, is controlled by varying the device's pitch and roll. For objects that support multiple continuous manipulations, such as *pan* and *zoom*, the user scrolls through the available manipulation modes using the *volume down* button. Finally, a user presses *volume up* to exit manipulation mode. To avoid confusion about modes, SNOUT shows a state feedback icon in corner of the screen.

6.4 Summary

The 3 interaction techniques which make-up SNOUT represent coverage of interaction primitives used in popular mobile-phone operating systems. Between selection, text-entry, and continuous parameter specification, users are able to control the majority of applications written for these platforms. Though functionally complete, we wished to evaluate the usability of our techniques and to determine if, given an advanced nose-overlay such as SNOUT, users would be more inclined to use their nose for input in scenarios where one or both hands are not available.

7. Usability Study

In order to evaluate SNOUT we conducted a usability study with two objectives. First, to learn of users' impressions of the various novel elements of our system: the combination of touch-and-tilt input, the SNOUT selection method, the tradeoffs between speech-to-text versus hunt-and-peck character input, the different feedback mechanisms employed, and our interaction techniques overall. The second objective was to quantify how users' impressions of nose-based interactions changed after using our design.

Participants were first asked to complete a survey about their willingness to use different body parts to interact with their device. The participants were then given the opportunity to interact with three SNOUT-enabled applications. Participants were asked both open-ended and targeted questions to solicit impressions and reactions. They were also instructed to use a talk-aloud protocol. At the end, participants repeated the original survey.

7.1 Participants

We recruited 12 volunteer participants (9 male). Participants ranged in age from 22 to 35 and all have prior experience with mobile touch phones, most also with tablets.

7.2 Apparatus

We developed an implementation of SNOUT on an HTC Desire HD phone. We modified 3 applications from the Android Developer site and SDK samples to utilize SNOUT [1]. Each of these applications contains some elements of selection, continuous parameter specification, or text entry.

7.3 Procedure

Participants held the handset in their dominant hand. First, users were instructed to familiarize themselves with the *touch + tap* gesture. Then each application was loaded for the user in turn and the relevant interaction techniques were explained and demonstrated for the user. Within each application, participants were provided with training exercises before completing a set of tasks.

7.4 Quantitative Measures

The survey consisted of 5 Likert scale ratings of their willingness to operate their phone using different body parts. The questions sought to answer, *what body parts would you use* and *if so, for which tasks.*

7.4.1 Map Browser

To illustrate both 1D and 2D continuous parameter specification, we implemented a custom application using the Google Maps API. The application supports scrolling and zooming about the centre of the map, using SNOUT's continuous parameter technique.

The map control has 3 modes, *zoom, pan,* and *normal*. The application launches in normal-mode where the user can pan and zoom using drag and pinch gestures. By means of the *touch + tap* gesture, users can transition from normal mode into pan mode and then to zoom mode using the hardware volume buttons as described above.

The training exercises consisted of scrolling through the different modes and navigating the map. The tasks consisted of navigating between checkpoints in Cambridge, MA, Mexico City, Tokyo, and then returning to Cambridge.

7.4.2 Application Launcher

The application launcher is based on the HomeSample demo from the Android SDK [1]. The launcher is laid out as a grid of 4 columns of icons, as seen in Figure 4 **Error! Reference source not found.**(a). By construction, our grid does not fit on a single page and the user must scroll down to access all the applications. This application illustrates both nose-based selection and one-dimensional continuous parameter specification.

Application icons were coloured in a 4-column, 2-row GBGB RYRY pattern. The icons were approximately the same size as the medium-sized targets in the initial pilot study. When scrolling, a scrolling-bar icon appears in the bottom-left corner of the screen.

Training exercises consisted of selecting various applications, as well as scrolling to different positions in the list. The tasks consisted of launching 5 different applications. The applications were located in both the top, center, and bottom rows of the screen. As well, scrolling was necessary for some selections to be made.

7.4.3 NotePad

The NotePad application extends the NotePad sample code from the Android SDK [1]. This application demonstrates speech-initiated speech-to-text and colour-based selection. The fixed-size selection objects in this application, the keyboard keys, are rectangular in shape, but have the same approximate width as the small targets in our pilot study. The application was limited to portrait mode so the keyboard keys were of the smallest size possible.

The speech-to-text functionality is provided by Google's voice-recognition library available through Android APIs. The speech recognizer returns a list of candidate matches and SNOUT selects the best candidate without presenting the others to the user.

After familiarizing themselves with the keyboard and speech-to-text, participants were asked to compose different messages of at least 10 characters. They were then asked to compose short messages of less than 3 characters. Finally, there were asked to write text of

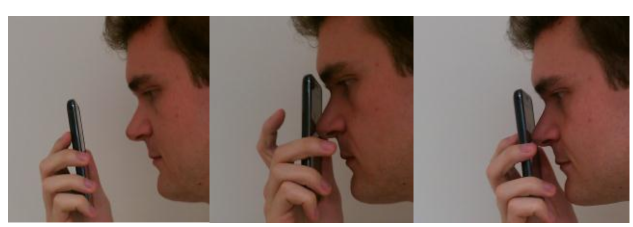

Figure 5. The touch + tap gesture activates manipulation mode. The user touches the object to be manipulated with their nose input and taps on the back of the device.

varying length using each input method.

7.5 Results and Discussion

Participants were encouraged to give feedback throughout the experiment. This feedback is presented here, grouped by method.

7.5.1 Selection

All the participants found the colour feedback very helpful for disambiguating the object touched by their nose. Similar to our first experiment, P9 noted that he consistently thought he was touching a different object than what the feedback was telling him, suggesting that nose-based selection may have a unique mental model for selection point, similar to but different from that known for fingers [16].P12 liked the selection technique so much he conjectured use-cases that would warrant such a technique and experimented with them, for example laying the phone flat on a table and selecting without using any hands. Another user said he would "definitely use this with my finger or thumb".

P2 and P12 had trouble remembering the relative layout of the colours, while P1 found the overlay obscured the application icon so much as to make it difficult to identify the application. Additionally, even with the colour-based selection, two users said their eyes hurt slightly from focusing their eyes on the tip of their nose. After the study they both suggested they might have fared better if they focused through the phone instead of trying to see the tip of their nose. P6 wears reading glasses and found the selection near-impossible without taking off her glasses first. Although the technique was usable and showed promise, it is clear that both the visual presentation and cases where near-vision is difficult require refinement.

7.5.2 Text Entry

All users were able to do text entry using the soft keyboard and SNOUT selection, although P6 removed her glasses first. When given a choice between either speech-to-text or hunt and peck selection using the soft keyboard, all of the users favoured using speech-recognition over the soft keyboard input. However, 7 people qualified their statements by saying that there are cases where they would be hesitant to use speech-to-text, for example if the content is private or their words are not found in the standard dictionary.

One user, P12, noticed the difficulty with using the keyboard to delete a whole word due to a lack of key-repeat. In SNOUT this is not possible, due to take-off selection. This suggests a need to enable word-level selections within text entry fields, or mechanism to enable key-repeat, such as a timeout.

P9 noticed that when the text box was positioned at the top of the screen, he had to move the phone away from his face to verify input. He suggested displaying the contents of the text box directly above the keyboard so he could keep his face in the same position as he typed, similar to many auto-complete keyboards.

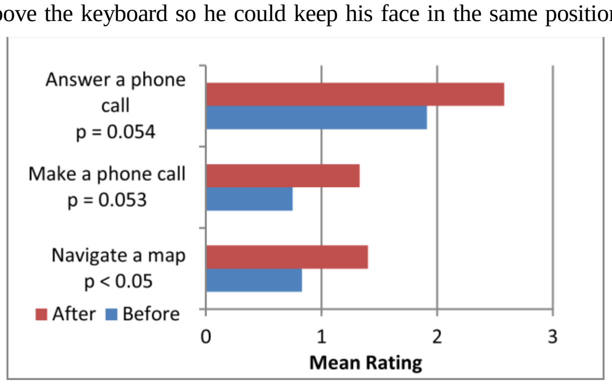

Figure 6. Participants' mean subjective ratings of whether they would use their nose to complete the action on a 4-point Likert scale with a 3 indicating higher preference. Statistical significance values are below the action.

7.5.3 Continuous Parameter Specification

Ten users particularly liked the *touch + tap* gesture to enter manipulation mode. They specifically liked the fluidity of the gesture and how there was no lag between tapping on the back of the phone and entering manipulation mode. One user liked how the gesture "is a fun action and makes me feel like a kid." After inadvertently tapping the back of the phone during a selection gesture, P9 commented that he felt the vibration, realized he was in the wrong mode and recovered gracefully.

Users took varying amounts of time navigating the 3 different tasks in the *Map Browser* application as they familiarized themselves with the sensitivity of the accelerometer in the phone. Ten users liked how they could control the *Map Browser* application just by tilting the phone. Three of them even asked if they could get such an application on their current smartphones. P6 commented that she had to concentrate on not gesturing with her hands as she spoke because it changed the position of the map. Two users also wanted the ability to scroll and zoom while in a single mode.

Some users fell back to familiar input gestures to scroll the application list and navigate the map. Within the *Application Launcher*, three users first tried to swipe the list with their nose, forgetting that SNOUT uses that gesture for selection.

Three users initially forgot about the hardware volume buttons to change modes and tried to transition from scroll mode to zoom mode with the *touch + tap* gesture. They immediately realized that (1) it doesn't work, and (2) the action of bringing the phone towards their nose caused the map to scroll to a new position. Three users didn't like the ergonomics of the holding the phone in a comfortable position for tilting, while still having easy access to the volume buttons. However, one participant liked the use of the volume buttons because it allowed him to keep the phone at a comfortable position the entire time he was manipulating the phone. The ergonomics of the volume button suggest that further testing is necessary with devices of varying sizes to evaluate the ergonomic effects.

7.5.4 Quantitative Results

We ran a Wilcoxon signed rank test comparing the users' impressions of nose-based input both before and after using SNOUT. As seen in Figure 6, participants were more willing to use their nose for simple tasks after using SNOUT. With respect to sending an SMS or interacting with either an elbow or chin, the differences between before and after were not statistically significant – this suggests that further refinement of our text-entry method may be in order.

7.5.5 Overall

Overall, our results show that users were able to control the handset using their noses, and that all of the techniques employed could be used to successfully complete the task. No user took more than 10 minutes to complete the training exercise for an individual application, validating our intention to design techniques which could be learned quickly. The feedback we received will be useful in refining the design of SNOUT prior to broader deployment.

8. Future Work

There are several avenues for potential future work. Our initial motivation for SNOUT came from discussion with friends and colleagues about their own experiences with mobile touch devices. Consequently we tailored our design focus and principles to address the case where the normal method-of-use is briefly restricted.

One extension to this project is for users who for accessibility reasons are not able to use touch-devices in the normal manner. A new design needs to support nose-based input for an extended period of time, and also for users that may have other physical limitation.

We implemented and demonstrate SNOUT as a proof-of-concept in several applications. We imagine deploying SNOUT as a UI interface layer between the user and the underlying application. SNOUT could then support a wider-range of applications without having to be specifically tailored for each application. Such an interface can rely on commonly found accessibility APIs included within the Android, iOS, and Windows Phone 7 SDKs.

One area that remains to be explored is how adept users are at associating screen objects to colours and relative position. This is not so prevalent for static layouts such as a keypad or number pad. However, it requires further study for tasks such as the application launcher where installing or uninstalling applications has the potential to modify the pre-existing layout and thus confuse users.

8.1 Alternative Appendages

Our initial motivations led to nose-input as an obvious solution, but the same principles and design is applicable to a much wider range of problems that fall within the intersection of the "fat-limb" problem, unusual ergonomics, and the unavailability of direct visual feedback.

8.2 Nose Mode Activation

One question that arose was how to initiate nose mode. We suggest that nose-mode can be enabled with an existing hardware button similar to Apple iOS`s method of accessing accessibility mode with a triple-click of the home button. However, more advanced techniques could leverage additional sensors in the handset such as proximity sensors, accelerometers, and gyroscopes.

9. Conclusion

In this paper we presented SNOUT, a novel UI overlay designed for nose-based input. SNOUT is intended for occasional use where one-handed use would otherwise be difficult or awkward. SNOUT is comprised of a trio of input methods: nose selection, continuous parameter specification, and nose-initiated speech-to-text.

Our pilot study suggested that, unaided by SNOUT, using one's nose can be difficult, error-prone, and lead to eye fatigue. We found, however, that by leveraging alternative channels such as peripheral vision, haptics, and accelerometers, nose input can become easier, more accurate, and fun.

10. References

1. Android SDK http://developer.android.com
2. Bartlett, J.F. Rock 'n' Scroll Is Here to Stay. IEEE Computer Graphics and Applications (May/June 2000), 40-45.
3. Baudisch, P. et al. Starburst: a target expansion algorithm for non-uniform target distributions. *AVI '08*, 129-137.
4. Benko, H. et al. Precise selection techniques for multi-touch screens. In *CHI'06*, 1263 -1272.
5. Butler, A. et al. SideSight: multi-touch interaction around small devices. *UIST '08*, 201-204.
6. Clarkson, E. et al. Revisiting and validating a model of two-thumb text entry. *CHI '07*, 163-166.
7. Esenther, A. et al. 2006. Fluid DTMouse: better mouse support for touch-based interactions. *AVI '06*.
8. Findlater, L. et al. Enhanced area cursors: Reducing fine-pointing demands for people with motor impairments. *UIST '10*, 153-162.
9. Gorodnichy, D. On importance of nose for face tracking. *Proc. Intern. Conf. on Automatic Face and Gesture Recognition (FG'2002)*, May 2002.
10. Gorodnichy,D. et al. (2002). "Nouse 'Use Your Nose as a Mouse' - a New Technology for Hands-free Games and Interfaces". *Vision Interface* (VI) 2002, 354-360
11. Grossman, T. et al. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *CHI 2005*, 281-290.
12. Harrison, B.L. et al. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. *CHI'98*.
13. Henry,T. et al. A nose gesture interface device: extending virtual realities *UIST '91*, 65-68.
14. Hinckley, K. et al. Sensing techniques for mobile interaction. *UIST 2000*.
15. Hinckley, K. et al. Sensor Synaesthesia: Touch in Motion, and Motion in Touch. *CHI 2011* (to appear)
16. Holz, C. et al. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints *CHI 2010*, 581-590.
17. Hudson, S. et al. Whack Gestures: Inexact and Inattentive Interaction with Mobile Devices. *TEI 2010*.
18. Jensen, T. et al. *Graph Coloring Problems Series in Discrete Mathematics and Optimization* (1995), John Wiley & Sons, New York USA.
19. Ketabdar, H. et al. MagiWrite: towards touchless digit entry using 3D space around mobile devices. *MobileHCI '10*, 443-446.
20. Ketabdar, H. et al. Towards using embedded magnetic field sensor for around mobile device 3-D interaction. *MobileHCI'10*, 153-156.
21. Kristensson, P. et al. SHARK 2: a large vocabulary shorthand writing system for pen-based computers. *UIST 2004*, 43-52.
22. Kurtenbach, G. (1993). *The Design and Evaluation of Marking Menus*, Ph.D**.** Thesis, Department of Computer Science, University of Toronto, May 1993.
23. Loclair, C. et al. PinchWatch: A Wearable Device for One-Handed Microinteractions. *MobileHCI'10*.
24. Malley, A. *Apple files for patent on winter-friendly iPhone gloves*. Posted Jan. 1, 2009. http://tinyurl.com/iglovepatent
25. Perlin, K. Quikwriting: continuous stylus-based text entry. *UIST '98*, 215-216.
26. Potter, R. et al. Improving the accuracy of touch screens: an experimental evaluation of three strategies. *CHI '88,* 27-32.
27. Roudaut, A. TimeTilt: Using Sensor-Based Gestures to Travel through Multiple Applications on a Mobile Device. *Interact '09*.
28. Siek, K.A. et al. (2005). Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. *INTERACT '05*. 267-280.
29. Vogel, D. et al. Shift: a technique for operating pen-based interfaces using touch. *CHI '07*, 657-666.
30. Wigdor, D. et al. LucidTouch: A See-Through Mobile Device. *UIST '07*, 269-278.
31. Wilcox, Dominic. Finger-Nose™ Stylus for Touch Screens, May 28, 2011: http://tinyurl.com/nosepoint
32. Yankelovich, N. et al. Designing SpeechActs: Issues in Speech User Interfaces. CHI'95, pp. 369-376.
33. Yang, X-D et al. TouchCuts and TouchZoom: Enhanced Target Selection for Touch Displays using Finger Proximity Sensing. *CHI '11, 2585-2594.*
34. Koji Yatani, et al. "Escape: A Target Selection Technique Using Visually-cued Gestures" *CHI'08*, 285 - 294.
35. Using a Sausage On Your iPhone4, Posted Feb. 15, 2010: http://tinyurl.com/sausagephone