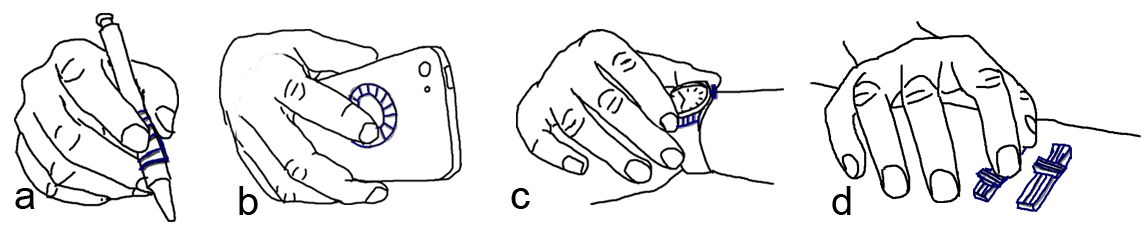
VIBGETs: Swipe Surfaces for Small Physical Objects

| 1st Author Name  Affiliation  City, Country  e-mail address | 2nd Author Name  Affiliation  City, Country  e-mail address | 3rd Author Name  Affiliation  City, Country  e-mail address |
| --- | --- | --- |

Figure . Scenarios of using on hand motion sensors (ring, watch, wrist band) to enhance the use of everyday things. (a) swiping on a textured pad while holding a pen. (b) swiping on a dial pad on the back of a mobile device. (c) pushing a button or swiping on a textured pad on a smart watch. (d) sliding on sliders on a table.

# ABSTRACT

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:

* ACM copyright: ACM holds the copyright on the work. This is the historical approach.
* License: The author(s) retain copyright, but ACM receives an exclusive publication license.
* Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Times New Roman 8-point font. Please do not change or modify the size of this text box.

Each submission will be assigned a DOI string to be included here.

Motion sensors embedded in wearable devices, such as sport bracelets, watches and rings, can add new dimensions to the way we use everyday objects. These sensors have the ability to digitalize our finger movements and transmit the information to computing devices for ..... We demonstrate that an accelerometer worn on the wrist or on a finger can accurately detect the distinctive vibrations that occur when a finger swipes over different surface profiles and present VIBGETs – VIBration widGETs – differently surfaced attachable stripes that each generate a different frequency patterns when swiped across. We explore the VIBGETs design space in three user experiments where we investigate elemental design factors such as swipe posture, sensor position, swipe direction, bump height, bump density, softness of stripes. From our results we extract initial guidelines for VIBGET design and then demonstrate several practical usage scenarios where VIBGETs are attached to physical objects to enrich/enable/extend interaction possibilities.

## Author Keywords

Wearable devices; Gestural input; 3D printed widgets; Everyday objects.

## ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous; See<http://acm.org/about/class/1998> for the full list of ACM classifiers. This section is required.

# INTRODUCTION

Recent research in always-available input has shown the promise of empowering daily objects with touch input capability, where users can interact with their computing devices using touch gestures on any object that is available to them, including walls [ref.] or tables [ref.]. A common approach to implement always-available input is to instrument the environment where the user is in with electronics and sensors [ref.]. This way, touch events can be detected by the sensors embedded in the touched object. Another popular approach requires a specially-designed sensor to be worn on user’s body [ref.]. This allows the user to carry out touch input on the surface of an object, where the movement of the hand or finger is tracked by the body-worn sensor.

Despite the pros and cons of the existing approaches, a common problem of today’s always-available input technology is that the technical burden of using them are too high for ordinary users, who are often lack of necessary skills to use them. For example, users will need to be familiar with capacitive [ref.], acoustic [ref.], or time domain reflectometry [ref.] sensors and need basic electronic skills to ensure the electronics and sensors to be properly installed at home or in the office. Additionally, the structures of the current environment or daily objects need to be well-examined in order for the sensors and wires to be seamlessly embedded. This in most cases can be rather tricky [ref.] and can take substantial efforts from the user.

On the other hand, the state-of-the-art body-worn sensors are usually made ad-hoc for research purpose to demonstrate concept ideas. To date, most of them still remain as research prototypes thus lack of ergonomic design, making them too heavy and bulky to wear in everyday life [ref.]. Finally, none of the existing technique provides an easy-to-use software toolkit to guide the user through the entire design, fabrication, and instrumentation process.

We propose to reduce the burden of enabling touch input on everyday objects through passive clip-on widgets (Figure 1). With passive widgets, ordinary users can easily instrument their environment by attaching the widget to the surface of any object. Users do not need to wear ad-hoc sensors to sense touch events. Instead, existing smart devices worn on user’s wrist or finger (e.g. watch or ring) can be used. We exemplify the capability of the proposed widget through a series of design iterations, and introduce Vidgets, a 3D printed swipe widget consisting a number of carefully-designed bumps on its surface that are used to generate unique patterns of mechanical vibrations on user’s finger during a swipe motion. The vibrational energy patterns can be sensed by the accelerometer in the smartwatch or ring, and can be recognized using a machine learning classifier for contextual actions.

In comparison to the existing methods [ref.], Vidgets can be easily deployed without the need of using special sensors in the environment or on user’s body. Our studies show that using accelerometer alone, Vidgets can reliably identify N different vibrational patterns, making it comparable, in terms of recognition capability, to the existing vision or acoustics based approaches [ref.], which rely on ad-hoc hardware or configuration that are not available in off-the-shelf smartwatches or rings. Equally important, we developed a software toolkit to help the user design Vidgets for objects with surfaces of different shapes. The user can finish the design using our tool, and fabricate the Vidgets using a 3D printer. Our toolkit also provide an easy-to-use interface to allow the user to map different actions to widgets with different vibrational patterns. A final user study shows that given a relatively short period of time, users can use our toolkit to successfully design, fabricate, and deploy Vidgets on a variety of daily objects.

Our contributions of this work are: (1) a demonstration of the accelerometer, that it works and is accurate enough (2) design guidelines, show limit/ranges of fundamental design parameters (material, finger posture, height, density...), and (3) a demonstration of practical use cases for VIBGETs.

# Related Work

In this section, we review the related work in environment and user instrumentation and in the area of context-awareness interactions.

## Instrumenting Physical Objects

Computer vision based techniques have been widely used in supporting touch on existing physical objects. Portico [ref.] and Bonfire [ref.] use RBG cameras installed on a laptop to capture surrounding touch events on a regular tabletop. Light Widgets [ref.] use cameras on the ceiling to detect touch input on walls or tables in a smart home. Recently, more capable depth-sensing cameras have started to replace their 2D counterparts in many interactive applications. For example, Wilson [ref.] showed a depth-sensing camera can be used to detect a wide variety of touch input on a flat surface. LightSpace [ref.] and WorldKit [ref.] use multiple depth-sensing cameras on the ceiling to enable even richer interactions on and between the surfaces of tables and walls.

Other approaches utilize custom-made sensors attached to the touched objects. For example, Pinstripe [ref.] enable touch input on clothing via capacitive sensing threads sewn into the fabric. Similarly PocketTouch [ref.] allows touch to be sensed through fabric but requires the capacitive sensor to be twisted in order to achieve a larger sensing range. Touché [ref.] enables touch input on daily objects, where each object needs to be instrumented with a swept frequency capacitive sensor. Wimmer and Baudisch further extend the concept to allow deformable object to be touch-sensible by using a time domain reflectometry sensor [ref.] that is not easily accessible by novice users.

Acoustic sensors are also widely used in sensing touch. Scratch input [ref.] uses an embedded microphone to detect finger scratches on a table or wall. Toffee [ref.] uses four back-mounted piezo on a mobile device to sense the location of finger taps on the table, on which the device resides. Touch & Activate [ref.] instruments daily objects with two piezo, which sense the change of resonant spectra upon a touch event. Stane [ref.] and Texture pad [ref.] use a piezo microphone attached to a textured surface to sense finger scratches. Acoustic barcode [ref.] takes a similar approach but encodes 1D barcode into the textured surface. Finally, Lamello [ref.] extends Acoustic barcode and apply the technique on passive tangible widgets. Unlike the previous approaches, Vidgets does not use acoustic sensing as microphones are not commonly seen in rings.

Instrumenting the existing electronic devices can also be challenging. Back-of-the-Device input [ref.], for example, utilizes the unused surface on the back of a smartphone for touch input through an “aftermarket” capacitive sensor. Similarly, SideSight [ref.] uses a custom-made proximity sensor array on the sides of a mobile phone to enable touch input on the desk, where the phone resides. FlexAura [ref.] and Multi-touch pen [ref.] enables touch on the barrel of a pen using specially designed touch sensors.

## Instrumenting the User

Custom-made body-worn sensors are also widely used in always-available input [on-body interaction]. Omnitouch [ref.] and Imaginary Phone [ref.] use shoulder-worn depth-sensing cameras to turn a user’s palm into a touch-sensitive surface. Saponas, et al. [ref.] detects different pinch gestures through 10 electromyography sensors worn on a user’s forearm. Skinput [ref.] is a bio-acoustic sensor array also worn on the forearm that can detect taps on the different locations of the arm. iSkin [ref.] is a skin overlay made of thin and flexible biocompatible materials that can sense both capacitive and resistive touch input on the user’s body. Magic Finger [ref.] requires a user to wear a pair of optical sensors (e.g. a low resolution optical flow sensor and a high resolution miniature RGB camera) on the fingertip to track the finger motions on a flat surface of a daily object. LightRing [ref.] achieves the same goal by using a ring-shaped sensing unit, consisting of a gyroscope, an infrared emitter and detector, worn on the base of the finger to avoid occluding the fingertip when performing normal hand functions. Cohn, et al. [ref.] demonstrate that using a neck-worn voltage sensor, the user’s body posture or the location where the user touches on the wall can be detected through the electromagnetic signals radiating from the power lines and walls at home [ref. ref.]. Implanted User Interfaces [ref.] are touch sensors implanted under the user’s skin, a step towards the future where sensors may become disappear from the user’s perspective [ref.].

|  |  |  |
| --- | --- | --- |
|  | Property/Factor | Description and Examples |
| *Hardware & Software* | **Sensor position** | The sensor can be positioned on any part of the arm performing the swiping motion. From close to the swiping surface to the shoulder. *Mounted on fingernail,* ***ring****,* ***wrist****, or elbow.* Swiping with phone in hand and using phone sensor. |
| **Sensor quality** | Performance, reliability... Frame rate... |
| **Classifier performance** | Training period and training approach of the classifier. How many “features” are used in the classifier; how clever is the algorithm... |
| *Surface* | **Profile** | Uniformity between bumps. The same uniformity could be “reversed” and we could have a series of wells to swipe over. |
| **Material** | The surface, and or the bumps, can be of different material, hard, soft, mixed. |
| **Bump dimension** | Unknown number of bump combinations for height, width, length (smaller than finger width), ... |
| **Bump density** | Bumps on the surface could be densely or sparsely packed together. |
| **Bump shape** | Bumps can be sharp, flat, wide, round... |
| **Curvature** | The surface could be flat or curved (convex or concave, ...) this may influence the vibration pattern |
| **Stability** | The surface could be stable or unstable (or somewhere between) and it may be moved when swiped across, accordingly these movements ought to influence the resulting acceleration pattern. |
| *User-determined* | **Swipe length** | A swipe that starts close to the bump (or the first bump on a multi-bump stripe) may yield a different swipe pattern than a swipe that starts further away from the bump. Likewise, the distance swept after the last bump may influence the swipe pattern. |
| **Swipe object** | A user can swipe with a finger or any other object, e.g., the tip of a pen or a corner of a credit card. Properties such as size, flexibility, sharpness, and the material of the swipe object may influence the swipe pattern. |
| **Swipe pressure** | A user can regulate the pressure applied to the swipe surface. The pressure can be constant throughout a swipe; the pressure can vary during a swipe. |
| **Swipe direction** | A user can swipe across a bump along the horizontal direction (left-to-right, right-to-left) or along the vertical direction (up-to-down, down-to-up). Diagonal. On a larger surface with a “shape” the direction/motion might be circular, or even swiping along the sides of a square (combination/sequence of “swipes”, rapidly after one another |
| **Swipe speed** | A user can regulate the speed while swiping. The speed can be constant throughout a swipe. The speed can vary during a swipe. |
| **Swipe angle** | A user can swipe with different finger postures or swipe while holding the swipe object at any angle to the stripe. A user can have the finger/object in an upright position (i.e., close to a 90° angle to the stripe) or in a more horizontal position (i.e., close to a 0° angle to the stripe) while swiping. |
| **Cognition** | Several cognitive factors influence the utility of the system, how many stripes can a user “remember”, anticipate the correct acceleration pattern from various bumps/surfaces etc. |
| *External* | **Temperature** | The temperature might influence the surface and its properties. Accordingly, a bump might get stiffer or softer and so the pattern may vary from time to time according to the temperature. |
| **Moisture** | As above, swiping across a wet surface may be different to a dry surface/material/bump. |
| **Stability** | The level of ambient noise. Swiping in a bus/car/train etc. |

Table . The design space for VIBGETs consists four main categories with factors and properties.

## Contextual Interactions

Contextual interactions have been widely explored in the ubiquitous computing community. A comprehensive survey can be found in [ref.]. In the context of touch interactions, a variety of interaction techniques have been proposed to allow tabletops to respond differently upon what cause a touch event [ref.] or which user touches it [ref.]. In contrast, Magic Finger [ref.] triggers contextual actions based on the object the user touches. The device uses a miniature RGB camera to distinguish different objects based on the texture of the objects. Other approaches use material’s optical properties to distinguish the objects the sensor touches [SpecTrans, Harrison]. Similar to Magic Finger, these sensors in their current form are too big to be worn in everyday life. Wrist-worn IMU sensors have also been used in context-awareness applications. For example, Object Hallmarks [ref.] uses the IMU sensor in a wrist band to identify the user who is using a home appliance.

To summarize, although most of the existing techniques have shown great sophistication and versatility, they are still not quite accessible by novice users. Vidgets allows the users to easily design and fabricate their own clip-on widgets to instrument small items, on which complex 2D gestures are not preferred. Already popular wrist or finger worn smart devices can be used to detect the touch events and trigger contextual actions.

# Design space for Vibgets

Several factors are likely to influence the design and usage of VIBGET interfaces and interactions. We first identify, discuss and structure the most prominent factors. After that we start our exploration of this new design space with three experiments where we investigating how six of the identified factors determine detection performance.

Obviously, the expressiveness (in terms of how many different acceleration patterns that can be distinguished by the system) and the utility for a user depend are highly dependent on how well the used accelerometer registers the vibrations resulting from a swipe. Next, the associated software needs to be able to interpret the sensor data and to distinguish between intentional vibration patterns and background noise. The vibration patterns resulting from a swiping motion are themselves likely to depend on properties related to the surface that is swiped (such as its softness and profile). Furthermore, the user can also strongly influence the vibration patterns by, for example, swiping at different speeds or in different directions. Finally, external factors, such as the amount of ambient vibrations, moisture and temperature, may influence the resulting vibration pattern. Accordingly, we structure the design space into four factor categories: 1) *Hardware & Software properties*,2) *Surface properties*, 3) *User-determined factors*, and 4) *External factors*. Table 1 lists the categories and the identified factors and properties.

This list of design-relevant factors is not intended to be exhaustive. Instead, we anticipate that we will encounter additional factors along our exploration. More importantly, we also anticipate that many of these factors and properties have strong interplays. It is likely that certain property combinations on two or more factors contribute to influencing one and the other. For example,

We start exploring the most foundational factors. We pick xxxx, yyyyyy, zzzzzz, zxzxzxzxzxzx. [explain factors and motivate why we pick these and why we regard these as foundational/most important].

# PILOT: VIBGET Hardware and software

To explore the feasibility of VIBGETs and detecting vibration patterns with accelerometers, we first built an experimental prototype that includes two IMUs (inertial measurement units) (IMU spec comes here xxxx yyyy zzzz). Figure XYa shows the prototype. One IMU is positioned on the wrist and one on the index finger. With two sensors, we can simultaneously detect vibrations on the finger and on the wrist (we opted not to use the on-board accelerometer in a smartwatch, since we wanted to have the same sensor on the finger for an accurate comparison without device-dependent influences [better formulation needed]). We used a xxyyzzaabb board with a yyzzxxyyy microcontroller and a ZigBee wireless transmitter to continuously send triple-axis acceleration values and absolute orientation values from the two IMUs at 55Hz to a desktop computer, where signal analysis takes place.

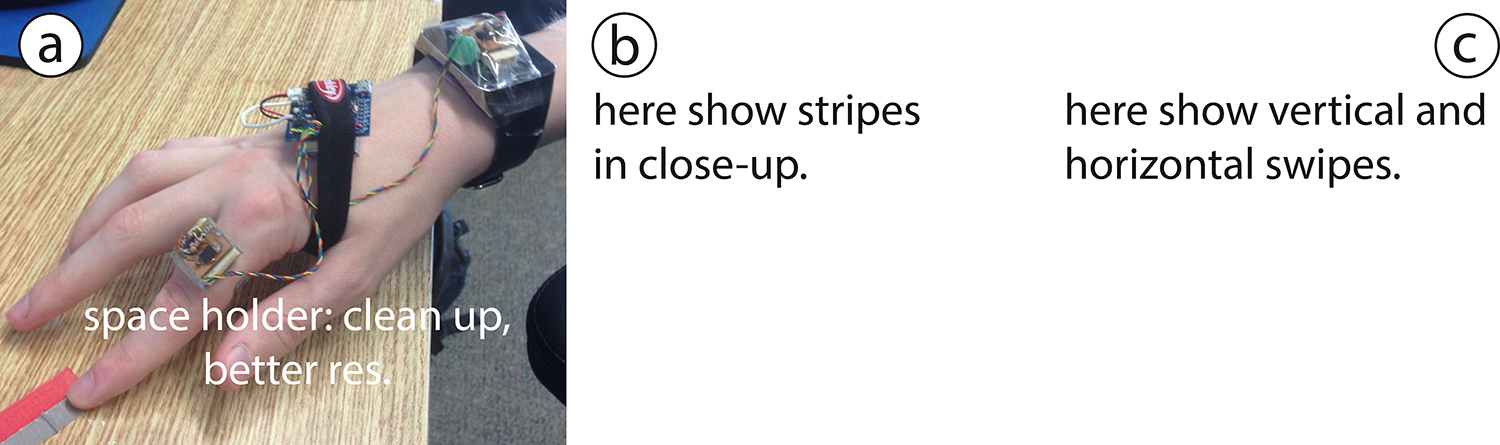


Figure . a) Prototype. b) Swipe stripes. c) Horizontal stripes (or something like this).

To better understand finger and hand motion when swiping over a surface, we asked X persons to alternatively swipe with their index finger along two plastic stripes on a table while wearing the prototype. The stripes measured 1×6 centimeters, one was flat; one had a 1mm high bump in the middle, as shown in Figure XYb. We instructed participants to explore different swipe speeds and finger postures that felt comfortable and natural. Participants performed horizontal swipes (Figure XYc), going from left to right across the stripes, and vertical swipes (Figure XYc), direction, from the top to the bottom. We recorded their swipes in slow motion video mode and collected the accelerometer data from the two IMUs.

The video recordings revealed two main swiping styles. Most swipes were arm-motored with the arm pulling the finger across the surface while the finger posture barely changed. The finger was held in either an upright posture or in a more horizontal and flat posture. With the first style, as depicted in Figure Ya, the fingertip and/or the nail slides across the surface. With the flat posture (Figure Yb), the finger pulp slides across the surface. Next to these two main styles, we also observed joint-motored swipes (Figure Yc) where the surface was traversed by bending the Distal and Proximal interphalangeal joints, which caused a slight raise of the hand and the fingernail to scratch the surface for most parts of the swipe.

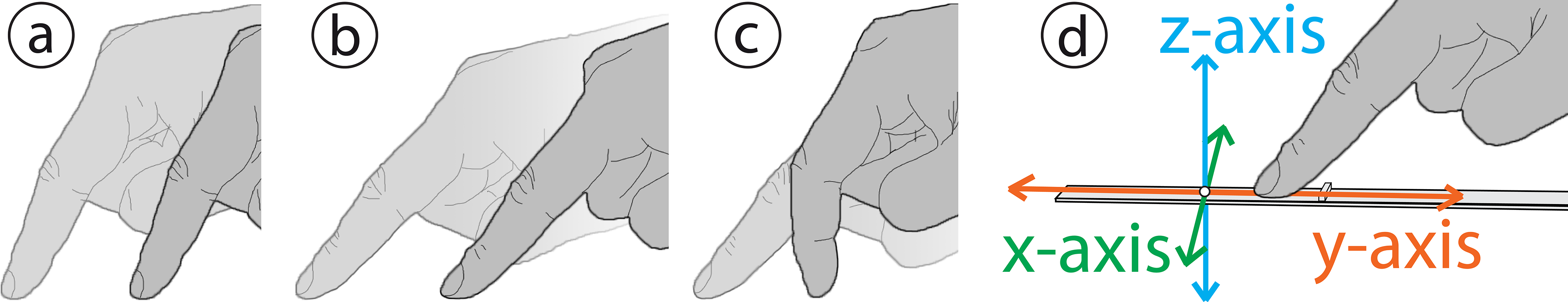


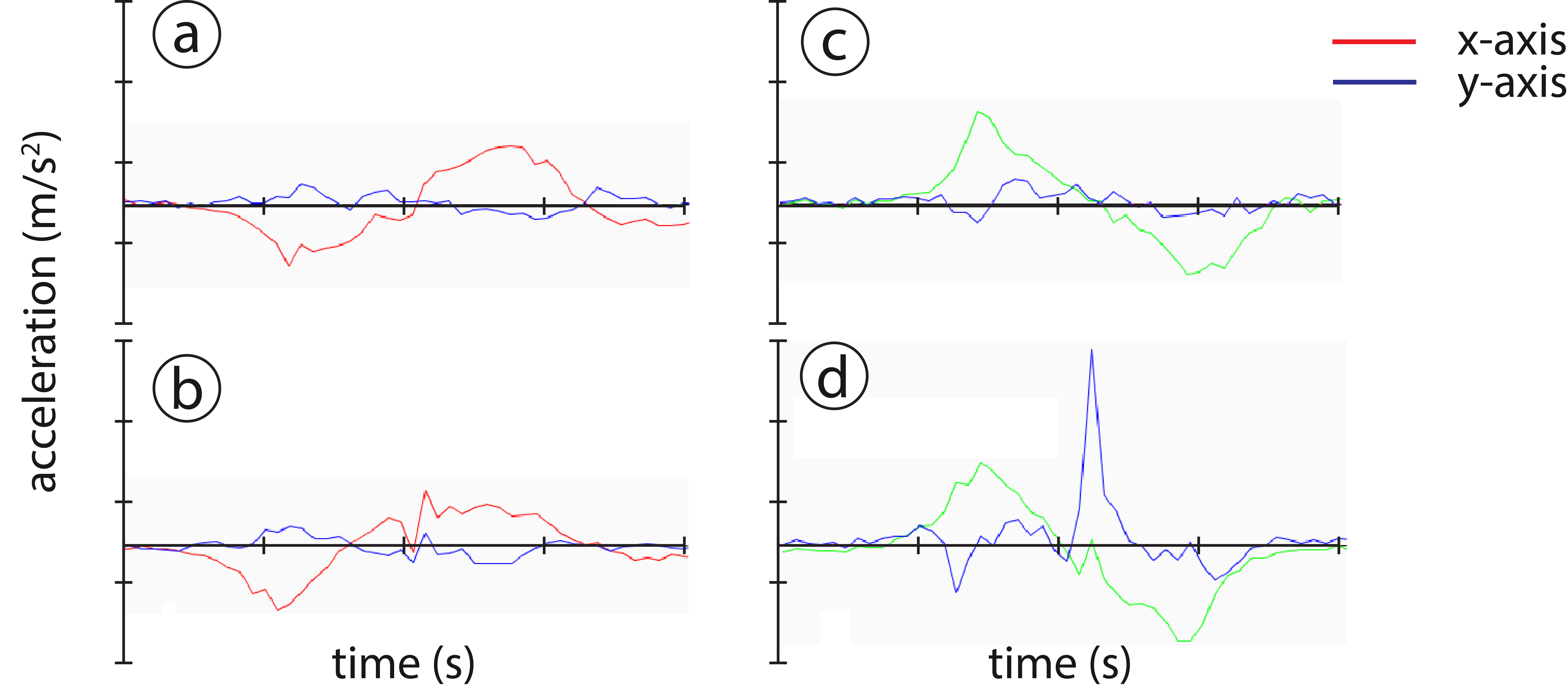
Figure . Swipe styles: arm-motored (a) upright and (b) flat swipe styles. (c) Joint-motored swipe style. (d) Vertical swipe in sensor direction... , or something like that...

In the video recordings we could also clearly observe how the finger was quickly raised and dropped (along the z-axis, see Figure 3d) when it passed over the bump. We did not observe any such up-down movements of the hand: presumably, the up-down movements were dampened by the swiping finger and were not propagated to the hand. We also did not see any marked movements along the x-axis (cf. Figure 3d) when the finger crossed the bump. Accordingly, when analyzing the collected sensor data we focused on accelerations along the z-axis and y-axis (the movement direction).

[original text: 1) along the motion direction, the finger’s movement was hindered by the blimp, and (2) orthogonal to the motion direction, the finger was raised by the blimp a little bit and then dropped back to the original level quickly and (3) orthogonal shift on wrist can be barely observed due to hand’s damping effect, but the motion along the moving direction was hindered in the same way (Figure xd). Thus we are interested in studying the motions along and orthogonal to the hand’s moving direction.]

The accelerations in these two directions were calculated by projecting the sensors’ linear acceleration (that is applied due to movement without the gravity force) to the hand moving direction and its orthogonal direction, and adding those sub vectors. For instance, for the sensor attached to the finger parallelly, it mainly rotates along the x-axis. As mentioned, the finger wipes vertically on the stripe, thus the hand moving direction was known. We calculated the vector components for z-acc and y-acc along the moving direction and its orthogonal direction, as shown in Figure X. Notice that x-acc was not used in this case. For horizontal swipes, it works in the same way.

Figure X plots calculated acceleration data along the z-axis and y-axis for two vertical example swipes, one across a bump and one across a stripe with no bump. The left part of the figure shows data from the finger sensor, the right part from the wrist sensor. Along the moving direction (y-axis, orange curve), both sensors respond to the bump (here guide the reader through the figure). Along the z-axis, however, we see marked differences between the two sensor positions with the finger sensor showing greater changes in the acceleration (here guide the reader through the figure).

Figure . Space holder. We need something like this, with axes labels. Calculated accelerations from the finger and wrist sensors.

With these initial insights regarding swipe styles and acceleration data, we followed the methods McGrath et al.’s [17] and Zhang et al.’s [33] to select features to represent the accelerations. Our selected features are xxxxxxxxx, yyyyyyyy, zzzzzzzzz, xyxyxxyxyyxyxyxy, and fkkkkkkkkk. Here please explain/motivate the selected features here.

Both IMUs collects ~55 frames per second, and a swipe gestures usually takes about 0.5 seconds. We extract 32 frames containing the largest acceleration value to calculate features. We make it to 32 frames for convenience of translating the signals into frequency domain. Besides adding the common measurements for brutal force signals as our features, such as standard deviation, skewness and kurtosis of both accelerations (3 \* 2), we calculate the largest absolute value difference between two consecutive frames of the signal (1 \* 2). In theory, compared to a swipe over a flat surface, a swipe over a bump should cause a sudden shift of the accelerations. Accordingly, the difference between two consecutive frames should increase. Notice, because the sensor output noise often leads to DC shifts – which are unpredictable – we do not use mean values of the acceleration series. Then we did fast frequency transform (FFT) to the signals and recorded the frequency amplitudes as our features. Since we use 32 frames of signals, the FFT function calculates 32 bins over 55Hz, and we were only interested in the first 16 bins based on the Nyquist frequency theory. We converted the result values with a decibel (dB) scale. As a result, we got 16 features on each acceleration. The 1st bin was abandoned as the value was extremely small. In total, there are 38 (8 + 30) features. Notice that these are the features we used for classifying swipes with and without bumps, or with different types of bumps, being aware of swiping direction (vertical or horizontal). To classify swipes with other hand motions like random movements, we added extra features such as displacement, which calculates the sensor’s travelling distance during the 32 frames (not sure yet, need some tests).

As indicated by our initial analysis, the horizontal and vertical swipe directions, swipe styles, and sensor positions are likely to produce different signal patterns. Accordingly, we were interested in knowing how well our classification algorithm performs in the various cases. We explore these factors in our first study, presented in next section.

# STUDY 1: SENSOR POSITION, SWipe direction, and finger posture

The main objective with our first study was to investigate how accurately our algorithm can distinguish between acceleration patterns produced from a finger swipe across a stripe with a bump and acceleration data that arises from ‘background noise’, such as random hand or body movements. Since we cannot a priori identify all possible acceleration patterns that may be picked up by the accelerometer, we use swipes across a stripe without a bump as our “closest-to-positive” negative pattern example to represent any random background noise that should be rejected by our swipe detection algorithm. The second objective with our first study was to start exploring some fundamental factors that are likely to influence the detection accuracy of our algorithm. At this early stage, we were interested in how the position of the accelerometer – in relation to the swiping finger – influences the detection accuracy and whether there are any differences in detection accuracy between horizontal and vertical swipes (i.e., along the sensors x-axis or y-axis) and how the swipe style effects detection accuracy. Accordingly, we invited 12 right-handed persons (9 male) aged xx to yy years (mean xx, s.d. yy) to swipe across horizontally and vertically oriented stripes using either the upright or the flat swipe style. We used our prototype (Figure X) to simultaneously collect acceleration data from the IMU positioned on the swiping finger and the IMU on the wrist.

## Data Collection

We used the same data collection procedure for all participants. The procedure included two sets of four series of 65 swipes. Set 1 included swipes across vertically oriented stripes. Set 2 included swipes across horizontally oriented stripes. The first two series of swipes in both sets included swipes using the flat swipe style, first 65 swipes over a stripe with a 1mm high bump (Figure X) and then 65 swipes over a stripe without a bump. The two last series in both sets included swipes with the upright swipe style. Again, first 65 swipes over a stripe with a bump and then 65 swipes over a stripe without a bump. When the four series of 65 swipes in Set 1 were completed, the procedure was repeated in Set 2 on horizontally oriented stripes, one stripe with a bump and one stripe without a bump. Accordingly, each participant performed a total of 8 × 65 = 520 swipes. With 12 participants, we collected acceleration data from 12 × 520 = 6240 swipes using two IMUs simultaneously (one on the finger and one on the wrist) and thus, ended up with 6240 × 2 = 12,480 data samples.

To ease and reduce pre-processing of accelerometer data we used a timed counter, displayed on a desktop monitor and shown in Figure X, to pace the participant through each series of 65 swipes. For 1.5 seconds, the counter counted down from ‘3’ to ‘1’ and an ‘S’ was shown to prompt the participant to start the next swipe. This also started the accelerometer recordings, which lasted for 1.5 seconds, giving the participant enough time to perform the swipe. After 1.5 seconds, the counter started to count down from ‘3’ again to prepare the participant for the next swipe. The experimenter observed the participant’s finger posture throughout the sessions and corrected the participant when necessary.

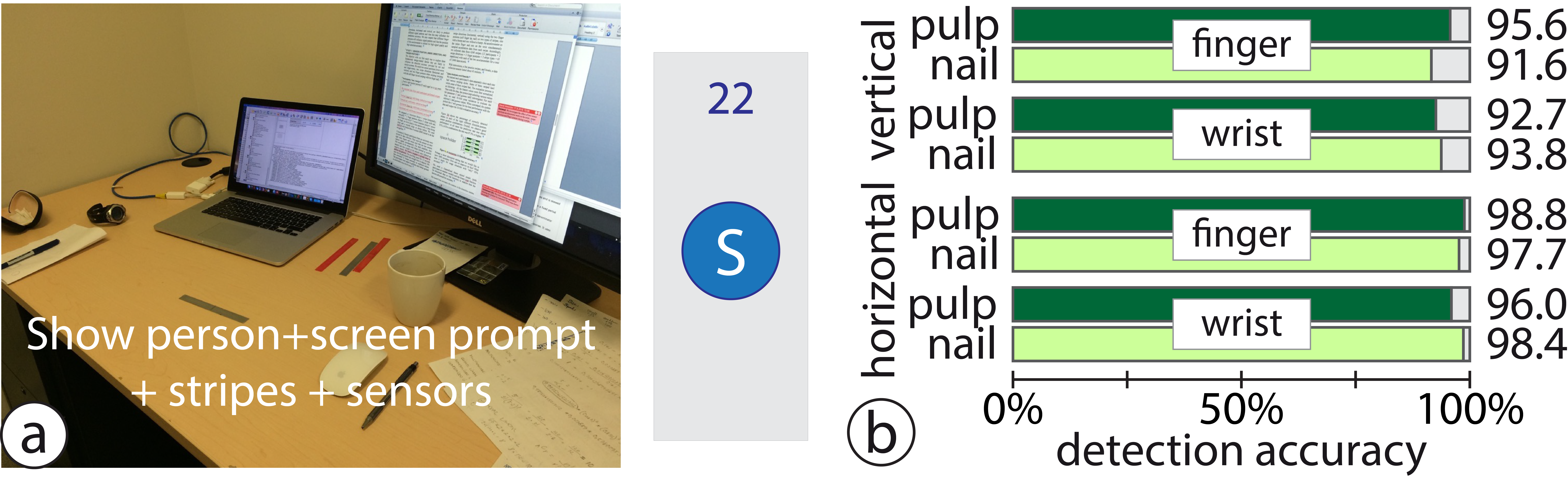


Figure . a) Study setup and screen prompt.

With introducing explanations and instructions, 20 practice swipes, and breaks between series of swipes, a session lasted approximately 50 minutes.

## Swipe Classification, Training, and Results

Here comes an explanation of the extraction, training etc. Include explaining that we ended up with different number of swipes in the different conditions for different persons: some swipes started too early and/or too late, and therefore we could not extract the necessary number of frames... From our 12,480 data samples we could use 11,826 samples (94.8%): in 645 samples participants swiped out of the required pace, and the recordings were not kjdfljd flej rejrel jrjlkjdlfkj adfl. Thus we discarded these . Also include something like this:

We discarded at most 15 of the 65 swipes in a series, still having a sufficient number of swipes (at least 50 in each of all participants’ 8 swipe series) for classification training and classification. We trained each participant’s data separately since each one had various swiping styles. Some of them swiped hard while some of them swiped fast. The evaluation process is as following: All the feature values were first normalized. Then with the data, the system kept selecting various values for parameter *C* and *gamma* until convergence. For each loop, we used 10-fold cross validation to evaluate the selected *C* and *gamma* values. When converged, the system picked the *C* and *gamma* for the best performance with the data and output the 10-fold evaluation accuracy.

We used a Generalized linear mixed model (with participant as random variable and the binary logit function for binominal data as link function) to analyze the classification accuracy for the different factor levels.

The classifier correctly classifies swipes as either a swipe over a bump or as a swipe over a stripe without a bump in 11,305 of 11,826 cases with an overall accuracy rate of 95.6%. The classification accuracy for vertical swipes is 93.42% and 97.77% for horizontal swipes. The difference between the two directions is statistically significant (F1,176 = 89.50, p < .001). The classification accuracy for the wrist sensor is 95.26%, the accuracy for the finger sensor is 95.93%. The difference is small, but statistically significant (F1,176 = 7.89, p < .01). Likewise, the difference in classification accuracy between swipes over a bump and swipes over the flat stripe is small, 95.40% vs. 95.78%, yet statistically significant (F1,176 = 7.73, p < .01). The difference between the two finger postures is not statistically significant (upright: 95.38%, flat: 95.81%).

These results should, however, be seen in the light of three significant interaction effects. First, in Figure Xa we see that there is no difference between the stripe with a bump and the flat stripe when swiped across in the vertical direction. When changing to the horizontal direction, the accuracy increases for both stripes, but to a somewhat greater extent for swipes over the flat stripe; this effect is reflected in a significant *direction* × *stripe type* interaction (F1,176 = 8.30, p < .01). The second interaction, *sensor position* × *finger posture* (F1,176 = 14.68, p < .001), is visible in Figure Xb. Here we see that the classification accuracy with the finger sensor is higher for swipes performed with the flat finger posture then than for swipes performed with the upright posture, in both the horizontal and the vertical direction. With the wrist sensor we find the opposite: classification accuracy is higher with the upright posture than with the flat finger posture (for both directions). Finally, after close inspection of Figure Xb, we also see the threefold *sensor position* × *finger posture* × *direction* interaction (F1,176 = 5.17, p < .05). This interaction demonstrates that the difference between the two finger postures are greater in the vertical than in the horizontal direction with the finger sensor but that the greatest difference between the two finger postures occurs in the opposite, horizontal, direction with the wrist sensor.

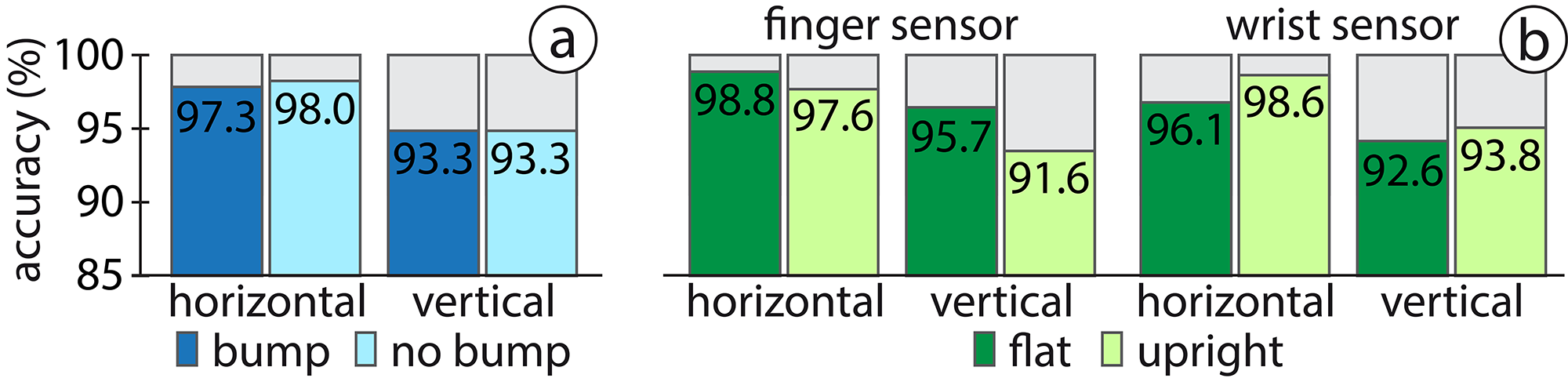


Figure . Interaction effects (note: the x-axis starts at 85%).

From these results, the following considerations and implications for designers and users emerge:

The positive news is that no matter sensor, posture or direction, the system is more likely to make an error by rejecting an intended swipe than by accepting random noise as a swipe (main effect stripe effect).

Expect better detection accuracy with a sensor on the finger than on the wrist (main effect sensor). Furthermore, no matter whether the finger position or the wrist position is used for the sensor, or what finger posture is used to perform the swipe, expect higher accuracy for horizontal swipes than for vertical swipes (main effect direction). Expect similar detection accuracy for intended swipes and noise in the vertical direction and be aware of a slightly better detection accuracy for noise than for swipes in the horizontal direction (*direction* × *stripe type* interaction). Accordingly, (*DG1*) in a design, if there is a choice, *make use of the horizontal direction*.

(*UG1*) *With a sensor worn on the wrist, swipe with an upright finger posture*. (*UG2*) *If the sensor is worn on the finger, swipe with a flat finger posture* (*sensor position* × *finger posture* interaction). With a finger sensor, swiping with the best finger posture (flat) is particularly important on vertical stripes: the sub-optimal upright posture will result in larger losses in detection accuracy in the vertical than in the horizontal direction. With a wrist sensor, swiping with the best finger posture (upright) is particular important on horizontal stripes (*sensor position* × *finger posture* × *direction* interaction).

## Summary Study 1

Having explored swipe direction, swipe posture, and sensor position we now precede with Study 2 where we explore three other design factors: bump height, stripe stiffness and stability.

# Study 2: Bump height, surface Stability, and Swipe device

In our second study we focus on the bump height, surface stability, and swipe device. As our first study show, our system is capable to distinguish between swipes across a 1mm high bump and swipes across a flat surface. Now we are interested in exploring the preciseness Also study performance on a soft stripe and a hard stripe, and on a stable and less stable object (pen and table). What posture should we use? Motivate this choice. What direction should we use? Motivate this choice. We only use finger sensor position. Motivate this choice. Explain why we do not use flat stripes as negative example, as we did in Study 1.

Twelve right-handed persons (9 male) aged xx to yy years (mean xx, s.d. yy) participated. None had participated in Study 1. We used the same prototype as in Study 1 and collected acceleration data from the IMU positioned on the swiping finger and the IMU on the wrist.

## Data Collection

The data collection procedure consisted of four sets of nine series of 65 swipes. In two sets the stripe was mounted on a pen, in two on a table (in vertical direction). Six participants started with the two sets with the stripe on the table, one set swiping with their bare finger and one set swiping when wearing a glove. The other six participants started with two sets swiping on the pen, one set with the bare finger and one set with the glove (three participants started with the glove, three started with the bare finger). Within each set, nine series of 65 swipes were performed: one series for each of nine different bump heights 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, and 1.6 millimeters. The bump heights were used in a random order. Accordingly, each participant performed a total of 4 (sets: pen+glove, pen+finger, table+glove, table+finger) × 9 (bump heights) × 65 = 2,340 swipes. With 12 participants, we collected acceleration data from 12 × 2340 = 28,080 swipes [using two IMUs simultaneously (one on the finger and one on the wrist) and thus, ended up with 28080 × 2 = 56,160 data samples].

We used the same timed counter as in Study 1 to pace the interval between participants’ swipes in order to ease and reduce the necessary pre-processing of accelerometer data. Participants were allowed to use (and change) the swipe posture as they desired. With introducing explanations and instructions, 20 practice swipes, and extended breaks between series of swipes, a session lasted about two hours.

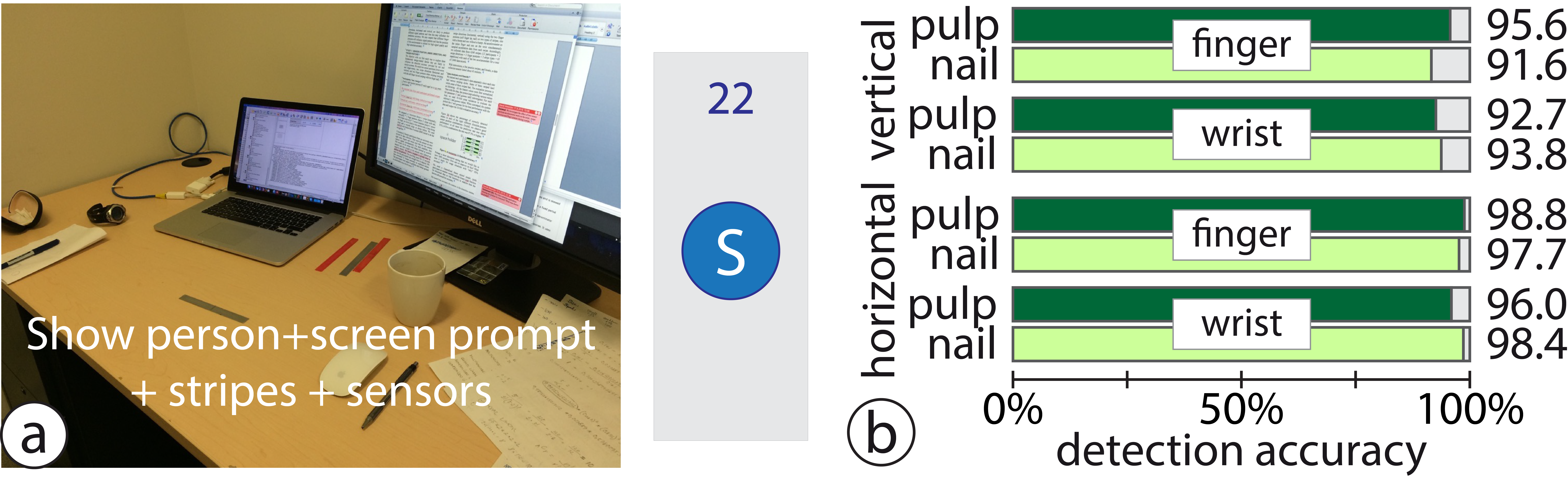


Figure . Study setup. Show the pen, the glove, and the stripes.

## Swipe Classification and Results

As with the data from Study 1, we could not use all 28,080 data samples for training and classification purposes. In X samples (xx%), participants swiped in the wrong “pace”. Accordingly, we discarded these samples (at most x from 65 swipes in one series). On the remaining XY swipes we used a Generalized linear mixed model (with participant as random variable and the binary logit function for binominal data as link function) to analyze the classification accuracy for the different factor levels.

Show height matrix. One for each of the four sets.

## Summary Study 2

# Study 3: Bump height, Stripe stiffness and Stability

If time and space allows, here comes a third study with three other design factors. Which do we pick?

## Swipe Classification and Results

## Summary Study 3

In the next section we apply our findings and design a number of tools/applications and demonstrate how the VIBGETs can be used in practical examples.

# Discussion, Limitations and future work

[Teng: we need to include sub-sections here for what we plan to discuss].

The discussion should include the factors we did not study, but are likely to influence either detection performance or the use of VIBGETs. The first category may include:

* How much can detection accuracy be improved with better sensors?
* Does the length of swipes (swiped distance before the bump and swiped distance after the bump) influence detection accuracy? How short can we go?
* Detection accuracy when swiping on other surfaces (without bumps) than our 3D stripes
* In vertical direction we only studied swipes “toward the body”, not away from the body (when swiping away from the body, the finger is more likely to “get stuck” and pause before going over the bump, i.e., similar to stick-slips on touchscreens)
* The time period between swipes. Can the system detect each swipe in a rapidly performed series of consecutive swipes? Using consecutive swipes could be “combined” into commands...
* Shaky environments, will VIBGETs work?
* We only have “one-dimensional” bumps, what if we swipe across a 2D-array of pimples.
* more ideas?

The second category (user related issues) could include the following:

* How easy is it for a user to remember and learn what functionality is triggered by swipes?
* more ideas?

## Future Work

Toolkit.

# CONCLUSION

Notes:

We need to make sure (in the motivation) to point out the advantages of an “unintelligent” plastic stripe. Reader may say “why not put a hardware button on some objects (e.g., the ones depicted in the first figure”. Why bother with printing bump-stripes, reading accelerations etc?”

# REFERENCES

1. Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2043-2046.
2. Daniel Avrahami, Jacob O. Wobbrock, and Shahram Izadi. 2011. Portico: tangible interaction on and around a tablet. In Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11). ACM, New York, NY, USA, 347-356.
3. Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: multi-"touch" interaction around small devices. In Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08). ACM, New York, NY, USA, 201-204..
4. Liwei Chan, Stefanie Müller, Anne Roudaut, and Patrick Baudisch. 2012. CapStones and ZebraWidgets: sensing stacks of building blocks, dials and sliders on capacitive touch screens. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 2189-2192.
5. Xiang 'Anthony' Chen, Tovi Grossman, Daniel J. Wigdor, and George Fitzmaurice. 2014. Duet: exploring joint interactions on a smart phone and a smart watch. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 159-168.
6. Lung-Pan Cheng, Hsiang-Sheng Liang, Che-Yang Wu, and Mike Y. Chen. 2013. iGrasp: grasp-based adaptive keyboard for mobile devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 3037-3046.
7. Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11). ACM, New York, NY, USA, 441-450.
8. Chris Harrison and Scott E. Hudson. 2008. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. In Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08). ACM, New York, NY, USA, 205-208.
9. Chris Harrison, Robert Xiao, and Scott Hudson. 2012. Acoustic barcodes: passive, durable and inexpensive notched identification tags. In Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12). ACM, New York, NY, USA, 563-568.
10. Sungjae Hwang, Myungwook Ahn, and Kwang-yun Wohn. 2013. MagGetz: customizable passive tangible controllers on and around conventional mobile devices. In Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13). ACM, New York, NY, USA, 411-416.
11. Shaun K. Kane, Daniel Avrahami, Jacob O. Wobbrock, Beverly Harrison, Adam D. Rea, Matthai Philipose, and Anthony LaMarca. 2009. Bonfire: a nomadic system for hybrid laptop-tabletop interaction. In Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST '09). ACM, New York, NY, USA, 129-138.
12. Wolf Kienzle and Ken Hinckley. 2014. LightRing: always-available 2D input on any surface. In Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14). ACM, New York, NY, USA, 157-160.
13. Ji-Eun Kim, John Sunwoo, Yong-Ki Son, Dong-Woo Lee, and Il-Yeon Cho. 2007. A gestural input through finger writing on a textured pad. In CHI '07 Extended Abstracts on Human Factors in Computing Systems (CHI EA '07). ACM, New York, NY, USA, 2495-2500.
14. Gierad Laput, Eric Brockmeyer, Scott E. Hudson, and Chris Harrison. 2015. Acoustruments: Passive, Acoustically-Driven, Interactive Controls for Handheld Devices. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 2161-2170.
15. Rong-Hao Liang, Liwei Chan, Hung-Yu Tseng, Han-Chih Kuo, Da-Yuan Huang, De-Nian Yang, and Bing-Yu Chen. 2014. GaussBricks: magnetic building blocks for constructive tangible interactions on portable displays. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems (CHI '14). ACM, New York, NY, USA, 3153-3162.
16. Rong-Hao Liang, Chao Shen, Yu-Chien Chan, Guan-Ting Chou, Liwei Chan, De-Nian Yang, Mike Y. Chen, and Bing-Yu Chen. 2015. WonderLens: Optical Lenses and Mirrors for Tangible Interactions on Printed Paper. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 1281-1284.
17. William McGrath and Yang Li. 2014. Detecting tapping motion on the side of mobile devices by probabilistically combining hand postures. In Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14). ACM, New York, NY, USA, 215-219.
18. Marios Papas, Thomas Houit, Derek Nowrouzezahrai, Markus Gross, and Wojciech Jarosz. 2012. The magic lens: refractive steganography. ACM Trans. Graph. 31, 6, Article 186 (November 2012), 10 pages.
19. Young-Woo Park, Joohee Park, and Tek-Jin Nam. 2015. The Trial of Bendi in a Coffeehouse: Use of a Shape-Changing Device for a Tactile-Visual Phone Conversation. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 2181-2190.
20. Ring Zero, Logbar Inc. http://logbar.jp/ring/en
21. Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 483-492.
22. Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B. Goldman, Gautham Mysore, and Wilmot Li. 2015. Lamello: Passive Acoustic Sensing for Tangible Input Components. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 1277-1280.
23. Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. 2011. Grips and gestures on a multi-touch pen. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 1323-1332.
24. Brandon T. Taylor and V. Michael Bove, Jr.. 2009. Graspables: grasp-recognition as a user interface. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09). ACM, New York, NY, USA, 917-926.
25. Stuart Taylor, Cem Keskin, Otmar Hilliges, Shahram Izadi, and John Helmes. 2014. Type-hover-swipe in 96 bytes: a motion sensing mechanical keyboard. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 1695-1704.
26. Cary Williams, Xing Dong Yang, Grant Partridge, Joshua Millar-Usiskin, Arkady Major, and Pourang Irani. 2011. TZee: exploiting the lighting properties of multi-touch tabletops for tangible 3d interactions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 1363-1372.
27. Katrin Wolf, Stefan Schneegass, Niels Henze, Dominik Weber, Valentin Schwind, Pascal Knierim, Sven Mayer, Tilman Dingler, Yomna Abdelrahman, Thomas Kubitza, Markus Funk, Anja Mebus, and Albrecht Schmidt. 2015. TUIs in the Large: Using Paper Tangibles with Mobile Devices. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15). ACM, New York, NY, USA, 1579-1584.
28. Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic finger: always-available input through finger instrumentation. In Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12). ACM, New York, NY, USA, 147-156.
29. Xing-Dong Yang, Edward Mak, David McCallum, Pourang Irani, Xiang Cao, and Shahram Izadi. 2010. LensMouse: augmenting the mouse with an interactive touch display. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 2431-2440.
30. Lining Yao, Jifei Ou, Chin-Yi Cheng, Helene Steiner, Wen Wang, Guanyun Wang, and Hiroshi Ishii. 2015. bioLogic: Natto Cells as Nanoactuators for Shape Changing Interfaces. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 1-10.
31. Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. PneUI: pneumatically actuated soft composite materials for shape changing interfaces. In Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13). ACM, New York, NY, USA, 13-22.
32. Neng-Hao Yu, Li-Wei Chan, Seng Yong Lau, Sung-Sheng Tsai, I-Chun Hsiao, Dian-Je Tsai, Fang-I Hsiao, Lung-Pan Cheng, Mike Chen, Polly Huang, and Yi-Ping Hung. 2011. TUIC: enabling tangible interaction on capacitive multi-touch displays. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2995-3004.
33. Cheng Zhang, Anhong Guo, Dingtian Zhang, Caleb Southern, Rosa Arriaga, and Gregory Abowd. 2015. BeyondTouch: Extending the Input Language with Built-in Sensors on Commodity Smartphones. In Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI '15). ACM, New York, NY, USA, 67-77.
34. Haimo Zhang and Yang Li. 2014. GestKeyboard: enabling gesture-based interaction on ordinary physical keyboard. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems (CHI '14). ACM, New York, NY, USA, 1675-1684.

According to new format, we need to put DOIs to the references.

## Study 1A: Blimp height levels

We evaluate 4 different blimp height levels for the two material.

Study factors: heights (4, 0.6mm, 0.8mm, 1.0mm, 1.2mm) × materials (2) giving a 4×2 factorial design. The task consisted of asking the user to flick over a blimp. The system recorded matching accuracy for each blimp.

Task: flick over blimps under the conditions. Record classification accuracy.

Procedure: We printed 8 stripes (x width by x length, as shown in Figure x) with the 8 conditions. Participants are first required to swipe through the stripes with bare index fingers and asked to point out whether they can tell the difference height levels. The result is used as baseline. Then they wear our prototype device with two motion sensors on index finger and wrist, one each. Similar to the procedure described in the previous section, the participants are given prompt on a monitor and perform swiping gestures on each blimp. Each blimp takes 65 swipes, all without nail, vertically. Swipes with each condition is labeled with a unique number.

Participants: we recruited N=10.

Results: we find that for material A and B, we can identify 3 and 2 levels respectively.

## Study 1B: Inter-blimp density levels

We next evaluate 4 density levels for recognizing inter-blimp density. We pick X height as it is highly recognizable with both materials and then follow the same procedure.

Task: flick over blimps under the conditions. Record classification accuracy.

Study factors: densities (4, x, x, x, x) × materials (2) giving a 4×2 factorial design.

Procedure: We printed 8 stripes (Figure x) following the study factors. Participants are first required to swipe on each of them and asked to tell whether they can feel the difference. Then they wear our prototype device, as in study 1a and swipe on each stripe for 65 times. We index the data for each stripe.

## Study 1C: Uniformity levels

We pick X height and X density and then evaluate 4 levels of uniformities on both material.

Task: flick over blimps under the conditions. Record classification accuracy.

Participants: we recruited N=10. (same group of people?)

Study factors: uniformities (4, x, x, x, x) × materials (2) giving a 4×2 factorial design.

Procedure: We printed 8 stripes (Figure x) following the study factors. Participants are first required to swipe on each of them and asked to tell whether they can feel the difference. Then they wear our prototype device, as in study 1a and swipe on each stripe for 65 times. We index the data for each stripe.

## Study 2: Parameter Manipulation

The prior study allows us to define certain parameters for our blimps for good enough accuracy. We now turn our attention to tasks in which a finger could introduce some effects in an application. We choose parameter manipulation as a canonical task, as it matches the affordance of the gesture and also is universally used for flicking, panning, zooming, menu navigation, selection of values from a multi-item control. Given that such tasks are designed for action in which the user is grasping an object, we investigate the effect of grasp style (tripod, i.e. pen or palm, i.e. smartphone) on performance accuracy. We also investigate the accuracy levels for this task.

Task: the task consisted of moving a virtual cursor onto a given target. We record time to complete, number of overshoot/undershoots. Selection is achieved after the cursor dwells on the target for 200 ms, just enough time to disambiguate from incidental activations during cursor movement.

# VIBE-DGET DESIGN GUIDELINES

Based on our results, we propose the following guidelines to apply in the design of VIBE-DGETS. [Teng: we need to think carefully about how to fill this table and I would like you to start filling it, even before running the studies, so that you know what your studies should provide you as output. You need to think clearly about what will be the most crucial guidelines and include those first, then include the less important ones, then check if the studies will give these or not.]

# VIBE-dgets

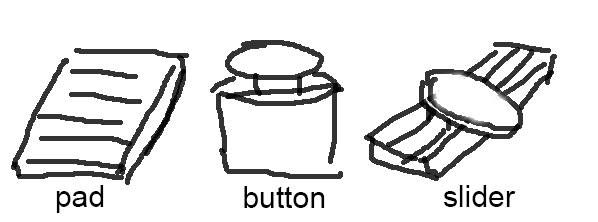
Based on the above two studies and our derived guidelines, we design a suite of VIBE-dgets each with their unique affordances.

## Design Example: Textures

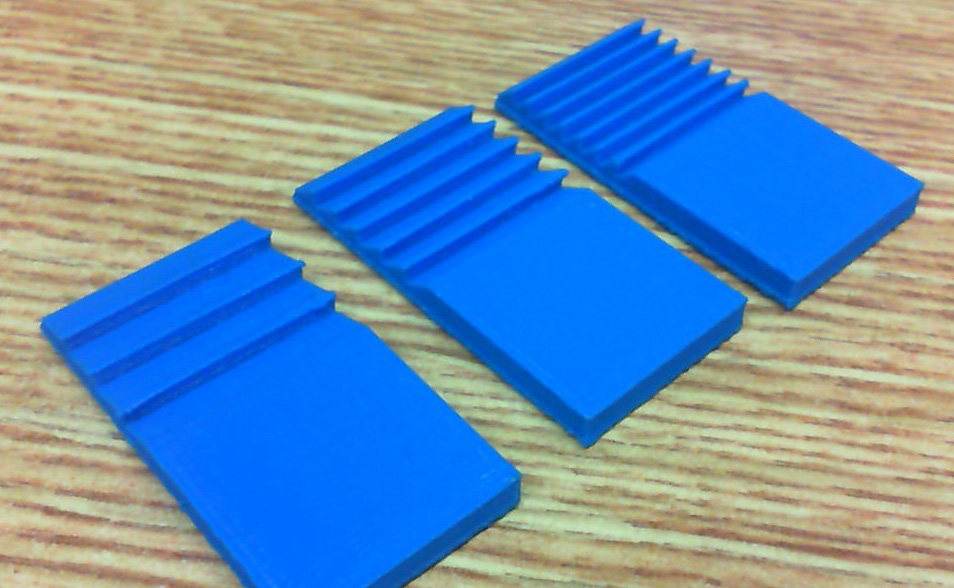
M

## Design Example: Form Factors

M



**Figure x. Form factors and their working mechanisms**



**Figure x. Printed samples**

# USER FEEDBACK

W

## Let Users Design Their Own Widgets

With

## Subjective Feedbacks

With