TWEND: Twisting and Bending as new Interaction Gesture in Mobile Devices

Gero Herkenrath
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
gero@cs.rwth-aachen.de

Thorsten Karrer
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
karrer@cs.rwth-aachen.de

Jan Borchers
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
borchers@cs.rwth-aachen.de

CHI 2008, April 5 – April 10, 2008, Florence, Italy
ACM 978-1-60558-012-8/08/04.

Abstract
In this work we present a hardware prototype that uses bending gestures as input for a mobile device and experimental setups that compare possible gestures with other, more traditional input methods in mobile computing. These will eventually result in guidelines for researchers and designers how to build bendable devices and show new interaction metaphors for computer user interfaces.

ACM Classification Keywords
H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces - Evaluation/methodology, Haptic I/O, Interaction Styles, Prototyping; H.1.2 [Models and principles]: User/Machine Systems; D.2.2 [Software engineering]: Design Tools and Techniques - User Interfaces.

Keywords
Deformable user interfaces, gesture input, bending, mobile computing, design, experimentation, human factors

Introduction
Technological development of materials has reached a stage where computers do not need to be made of rigid materials exclusively. Deformable batteries, bendable

circuit boards and flexible displays exist. These technological advancements are used, for example, in wearable computing to make mobile computers more robust or convenient. The ongoing work described here focuses on a different approach and looks at the possibilities of using deformation as a novel input methodology. Although there have been several other ideas on how to use deformation in human-computer interaction (HCI), e.g. [4, 5, 6, 7, 8], there is little knowledge on how users actually perceive deformation as an input method, i.e. what natural mappings exist between deformation and computer operations other than the obvious ones in 3D modelling, like [5, 8]. Our goal is to investigate the effects of using deformation as input and to provide design guidelines for a deformable user interface.

Because the different possible kinds of deformation of physical objects is quite large, we limit our research to the field of bending a device in the context of mobile computing, i.e. tasks like map navigation or eBook reading. This will extend the research done by [6, 7] whose authors explicitly state that currently there is an "[…] absence of established paradigms for interaction by deformation […]" [7].

The hardware prototypes

To explore the possibilities of using bending gestures as input we are currently building a hardware prototype that allows users to perform common computer related tasks by bending it. Due to cost limitations and the early stage of our research this prototype will not be completely mobile but connected to a desktop computer that does the calculations necessary for recognizing any bending gesture. For now it will not have its own display either, but this may change depending on availability and cost of flexible displays in the near future. Since we envision it being used for tasks usual in mobile computing its size will be about 25 cm x 15 cm, which we think is a good compromise between a larger personal digital assistant (PDA) and the ease of building the hardware prototype.

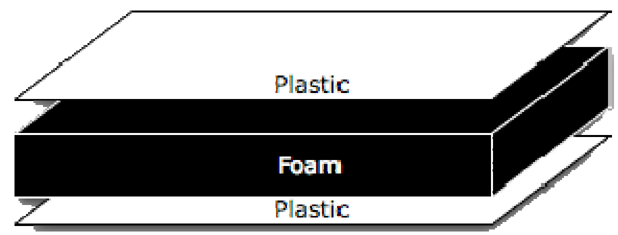

Figure 1. Two layers of thin plastic make the device bendable, the foam makes the device thicker, thus easier to grasp, and protects the embedded sensors.

Internally it will measure the bending state with eight optical bending sensors [3]. We use an Arduino[1] board to sample sensor data over the standard USB bus.

We already have a smaller prototype with a single sensor for testing our software implementation, which functions in the same manner like the larger one we envision. Their general structure is the same and illustrated in figure 1.

1 http://www.arduino.cc

Figure 2. The smaller hardware prototype to measure horizontal bending. The dimensions are about 13 cm x 5 cm.

Both prototypes, the small one shown in figure 2 and the planned large one, consist respectively will consist of two pieces of thin, flexible plastic with a layer of foam in between, in which the sensors are embedded. Each sensor consists of a single LED, a photocell and a partially abraded glass fiber cable plus resistors and power cables [3].

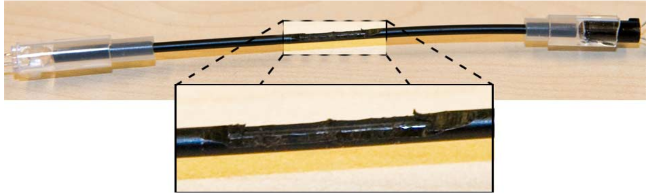

Figure 3. The optical bend sensor inside the prototype. The enlarged area shows the abraded part of the optical fibre cable necessary for bend sensitivity.

Implementation

The software generating the sensor data on the Arduino board is a simple C program that reads the Arduino's analog input pins and then sends the sampled data over the USB port to a desktop computer. It does not differ for either hardware prototype, although only one measured value is valid if the small prototype is connected to the Arduino.

We use an Apple Macintosh to translate the raw sensor data into bending gestures. Our simple gesture recognizer is still in development, but it already converts the data to a more abstract scale, reading how much each sensor is bent in which direction. The important part of the software however then recognizes the performed gestures, currently planned are eighteen possible ones, and maps them to preconfigured standard Mac OS X system events. Which exact events are generated can be specified in a configuration file. This basically enables the prototype to act like a standard computer input device, such as a keyboard or a mouse, and to control any application on the Mac. As a result we are able to very quickly set up experiments without always being compelled to write new software for it. As long as an existing application can be controlled by the keyboard or the scrolling of the mouse, it can be controlled by our hardware prototypes. For example, we plan to use Google Earth [1] as a map navigation substitute in our upcoming experiments. Even new applications can be coded very quickly using the Apple standard SDK. This offers a high reusability of our prototype for future experiments.

Bending gestures

Since we are not yet fully aware of which bending gestures are appropriate for input we attempt to make our prototype measure as many gestures as possible without making the device too complex. Due to the characteristics of the used material we were not able to make it fully twistable, because this would cause a deformation into a form that is not topologically equivalent to its original planar state, i.e. the area would change. Nevertheless the layout of sensors will hopefully allow us to distinguish between the following eighteen gestures on the larger prototype:

- Bending each of the prototype's four edges back and forth (8 gestures)
- Bending the prototype back and forth along the vertical, the horizontal and each diagonal axis (8 gestures, figure 5 shows one of these)
- Bending the prototype into a horizontal waveform with the wave valley either on the left or the right side of the device (2 gestures, figure 4 shows one of these)

The last two gestures appear to be especially interesting to us in the context of scrolling, for example, through the pages of an eBook. Imagine persons flipping through the pages of a softcover book. What they usually do is to hold the back with one hand and to grasp the side of the book with the other, then letting the pages slip from under their thumb. This effectively deforms most books into something vaguely like a wave, just like the gestures that we imagine. Of course the experimental evaluation is still to be done, but informal feedback from seven colleagues as well as from ten people not related to HCI was promising.

Figure 4. A waveform gesture for page flipping. The other one would look like a vertical mirror image to this.

Other possible tasks for certain gestures could be single page flipping by bending an edge of the device, the equivalent location of where a person would grasp a book’s page to flip it (see [2]), or bending it completely along its horizontal axis for zooming in and out or enter and cancel actions like in [6, 7].

Figure 5. Bending along the horizontal axis could be used for zooming or as an equivalent to an enter button.

Each gesture can be configured to work either as discrete or as continuous input. Discrete input could generate a single keystroke. For example when the user bends an edge to a certain degree the software would simulate a key press. Continuous input would consider the angle at which the edge is bent and, for example, influence the scroll speed or the keystroke repeat rate.

The experimental setup

As previously explained we plan to simulate common tasks in our experiments that may be performed either by bending our prototype system or by traditional input methods. The latter would be typical input gestures on common mobile devices such as PDAs or mobile phones, i.e. button presses or gestures on a touch sensitive area with either a finger or a pen. They will not include standard desktop computer input devices, because we do not believe that bending will be a superior input mechanism to those in most cases.

We currently plan several experiments to test and compare the bending gestures that can be performed with our larger prototype. We expect that to be eighteen gestures. The exact layout of the experiments is still to be planned, as it is not possible to do this without having a working prototype, however several points are already clear. First, the other devices compared with the prototype should be slightly modified in such a way that their output is displayed on the same display as the prototype’s output. Second, it seems reasonable not only to compare efficiency between the different input methods but also which gestures are natural and easy to learn for inexperienced users. However, this will not be easy to measure, and may be best accomplished by allowing the users to find out on their own, how each device works or by giving only little instruction. The last clear point regarding the experiments is to do a normal efficiency comparison between the different input gestures, i.e. to give users a certain task and to measure how exactly or how quickly they can accomplish this task with each gesture.

Considering the last two points it becomes apparent that we are more interested in users without any prior experience of any of the devices we plan to use, at least in the context of the given task. Although it may be of interest to determine whether bending gestures offer an efficiency gain over other devices if their users are familiar with the latter, e.g. a PDA, we want to focus on finding design guidelines for bending gestures. This means we have to begin our experiments with users new to both kinds of input before we can investigate the impact bending as input could have for people who already are professionals with another input method.

For example we plan the following tasks to measure the efficiency of using bending gestures as input:

- Map navigation: The user will navigate in Google Earth [1] from a starting point to a given location using only different gestures. This task will probably also include zooming using a continuous bending gesture.
- Scrolling: The user will be asked to find a certain page of an eBook using the scrolling mechanisms of the different devices.

Our hypotheses so far are as follows:

- Zooming with bending gestures is more efficient than zooming with mobile phone buttons or PDA-typical input methods.
- New users consider the page flipping and page scrolling gestures as convenient and easy to learn.
- The page flipping and page scrolling gestures are more efficient than those found in PDAs and mobile phones.

Whether these hypotheses are true or not has to be shown in future work but in any case we hope they will provide new insights into using bending as a new input method.

Ongoing and Future Work

In this paper we outlined the current status and plans of our project. Since it is in an early stage several things may still change. We plan the software to be as reusable as possible, allowing us to design new experiments for further investigation, such as which bending gestures are suitable for user input or even to determine if reducing the prototype in size has any effect and if so how much influence this has on the efficiency and convenience of gestures. A rather exciting and important question is also what influence a prototype mounted flexible display has on the user's experience with a bendable device. Finally, it may be interesting to investigate the benefits of a bendable device for desktop computing.

Acknowledgements

We thank Silke Wettich from Reisgies Schaumstoffe GmbH for providing us with foam probes for testing and Philipp Hetzelt for giving advice on how to wire the bending sensors. Also Peter Nequest, Daniel Spelmezan and Malte Weiss for proofreading the paper.

References

[1] Google Earth. http://earth.google.com/

[2] Harrison, B. L., Fishkin, K. P., Gujar, A., Mochon, C., Want, R. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. *Proc. CHI 1998,* ACM Press (1998), 17-24.

[3] Kuang, K. S. C., Cantwell, W. J., Scully, P. J. An evaluation of a novel plastic optical fibre sensor for axial strain and bend measurements. *Measurement Science and Technology,* 13 (2002), 1523-1534.

[4] Michelitsch, G., Williams, J., Osen, M., Jimenez, B., Rapp, S. Haptic chameleon: a new concept of shape-changing user interface controls with force feedback. *Ext. Abstracts CHI 2004,* ACM Press (2004), 1305-1308.

[5] Murakami, T., Nakajima, N. Direct and intuitive input device for 3-d shape deformation. *Proc. CHI 1994,* ACM Press (1994), 465–470.

[6] Schwesig, C., Poupyrev, I., Mori, E. Gummi: user interface for deformable computers. *Ext. Abstracts CHI 2003,* ACM Press (2003), 954-955.

[7] Schwesig, C., Poupyrev, I., Mori, E. Gummi: a bendable computer. *Proc. CHI 2004,* ACM Press (2004), 263-270.

[8] Sheng, J., Balakrishnan, R., Singh, K. An interface for virtual 3d sculpting via physical proxy. *Proc. GRAPHITE 2006,* ACM Press (2006), 213–220.