PaperPhone: Understanding the Use of Bend Gestures in Mobile Devices with Flexible Electronic Paper Displays

Byron Lahey[1,2], Audrey Girouard[1], Winslow Burleson[2] and Roel Vertegaal [1]

[1] Human Media Lab
School of Computing, Queen's University
Kingston, Ontario, K7L 3N6 Canada

[2] Motivational Environments Research Group
Arizona State University
Tempe, Arizona, AZ 85281, USA

{lahey,audrey,roel}@cs.queensu.ca, winslow.burleson@asu.edu

ABSTRACT

Flexible displays potentially allow for interaction styles that resemble those used in paper documents. Bending the display, e.g., to page forward, shows particular promise as an interaction technique. In this paper, we present an evaluation of the effectiveness of various bend gestures in executing a set of tasks with a flexible display. We discuss a study in which users designed bend gestures for common computing actions deployed on a smartphone-inspired flexible E Ink prototype called PaperPhone. We collected a total of 87 bend gesture pairs from ten participants and their appropriateness over twenty actions in five applications. We identified six most frequently used bend gesture pairs out of 24 unique pairs. Results show users preferred bend gestures and bend gesture pairs that were conceptually simpler, e.g., along one axis, and less physically demanding. There was a strong agreement among participants to use the same three pairs in applications: (1) side of display, up/down (2) top corner, up/down (3) bottom corner, up/down. For actions with a strong directional cue, we found strong consensus on the polarity of the bend gestures (e.g., navigating left is performed with an upwards bend gesture, navigating right, downwards). This implies that bend gestures that take directional cues into account are likely more natural to users.

Author Keywords

Flexible Displays, E Ink, Bend Gestures, Organic User Interfaces

ACM Classification Keywords

H.5.2. Information interfaces and presentation: User Interfaces – *Interaction styles, evaluation/methodology, user-centered design*.

INTRODUCTION

While research in the domain of flexible display interfaces has been ongoing for the better part a decade, there is, to our knowledge, little to no user interface research where actual flexible displays were deployed. Most of the display technologies used in prior studies were either based on simulations using projection on paper [10], rigid LCD displays on a flexible substrate [20] or paper mockups [13]. These methods of simulating real flexible displays potentially introduce biases for the evaluation of interactions. By using real flexible displays and integrated bend sensing we achieve interactions that align with the performance characteristics of devices that could be commercially available in the immediate future. While there may be suggestions that bending of a flexible display can be as effective and efficient an input technique as button controls in rigid displays for tasks like paging, the case for the use of flexible over rigid screens is not necessarily based on the superior efficiency of interactions. Indeed, much work is required for flexible touch screens to become as effective as rigid ones. However, while rigid screens may continue to have the edge in terms of interaction efficiency for some time, we believe there are sufficient practical and interactional reasons for flexible displays to achieve mass adoption. The likely reason for adoption of flexible displays is that they may closely approximate the look and feel of paper docu-

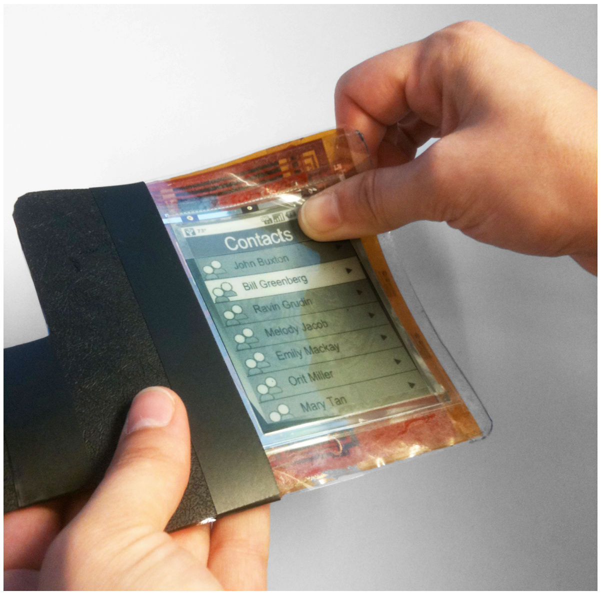


Figure 1. The PaperPhone prototype with flexible E Ink display features bend gesture input recognition.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
Copyright 2011 ACM 978-1-4503-0267-8/11/05...$5.00.

ments. Sellen and Harper [21] describe some characteristics of paper documents that may explain their continued popularity. Rigid Graphical User Interfaces (GUIs) often feature input that is indirect, one-handed, and dependent on visual cues. By contrast, paper documents, and presumably flexible displays, may:

1. Be very thin, low-weight, yet rugged, allowing superior portability over any current mobile computing form factor.
2. Have many form factors. This allows for distinct physical affordances that relate to specific functionalities: reading a newspaper serves a different purpose than reading a product label, and implies a different form factor.
3. Provide variable screen real estate that fits the current context of use.
4. Have many physical pages, each page pertaining only to a specific and physically delineated task context.
5. Use physical bend gestures with strong tactile and kinesthetic feedback for efficient navigation.

Prior simulations of flexible displays [8, 9, 10, 13, 20] have already produced a library of paper-like interaction styles, most of which focus on the use of bend gestures. A bend gesture is the physical, manual deformation of a display to form a curvature for the purpose of triggering a software action. In this paper, we present an evaluation of user preferences for bend gestures in executing a real set of tasks, using an actual flexible display. We designed a study in which users were asked to design their own bend gestures using a thin film E Ink display with integrated bend sensors. This approach has two distinct advantages over prior work: (1) visual feedback is provided directly on the display itself, and (2) dynamic material characteristics of bending layers of sandwiched flexible electronics were included.

In the first part of our study, we asked participants to define 8 bend gesture pairs. In the second part, we asked them to evaluate the appropriateness of their bend gestures for use with multiple actions. Finally, users were asked to use and evaluate bend gestures in the context of complete tasks (e.g., operating a music player). Results showed that users selected individual bend gestures and bend gesture pairs that were conceptually simpler and less physically demanding. There was a strong agreement among participants to use 3 bend gesture pairs in applications: (1) side of display, up/down (2) top corner, up/down (3) bottom corner, up/down. There was also strong consensus on the *polarity* (physical bend direction: up or down) of bend gesture pairs for actions with clear *directionality* (e.g., navigating left and right to select an icon).

RELATED WORK

We will first discuss work related to the development of flexible display interfaces, after which we will address empirical work on the design of bend gesture sets for multi-touch and flexible display user interfaces.

Understanding Interactions with Flexible Displays

Balakrishnan et al. [2] explored the use of ShapeTape, an input device that senses bend and twists, as a tool for 3D modeling. They emphasized the significance of the sensor affordances and the abilities of the user. They classified this input device as a high dimensional device, with more than three simultaneous degrees of freedom. We believe that flexible displays using deformation as an input modality will typically fall into this class of device, and are subject to user challenges arising from the associated complexity.

Schwesig et al. discuss Gummi, a bendable computer prototype. They demonstrated the feasibility and potential benefits of compact, flexible mobile computing form factors [20]. Gummi was designed with flexibility as an affordance, allowing both discreet events, considered at a maximum bending threshold, and analogue events, by measuring continuous transition states between thresholds. Navigation was achieved through bending the display. The interface was implemented using a rigid form factor display and a flexible sheet of acrylic augmented with resistive bend sensors. They proposed that such devices should have different interaction styles than traditional GUIs.

In PaperWindows, Holman et al. [10] created a projection based windowing environment that simulated fully wireless, full-color digital paper. Holman merged the properties of digital media with those of physical paper, allowing for input and output directly on the flexible display. They demonstrated use of gestural inputs such as hold, collate, flip, bend, point and rub [8, 10]. Augmenting this work, Gallant et al. [8] designed Foldable User Interfaces, a prototyping tool for flexible displays that uses Foldable Interaction Devices, sheets of paper augmented by infrared retroreflectors. They argued that physical page bends are effective metaphors for document navigation, an argument congruent with findings by Lee and Herkenrath [9, 13].

Twend was a hardware prototype developed by Herkenrath et al. that allowed complex navigations using twisting and bending [9]. Twend was constructed out of 8 optical bend sensors to recognize a wide variety of contortions. Similar in nature, Watanabe et al. [17] discussed Bookisheet, a set of flexible input devices made out of sheets of thin acrylic augmented with bend sensors. Bookisheet could simulate the turning of pages through bends. The interface changed between discrete jumping and continuous scrolling modes based upon the degree of bend between two sheets of cardboard. Similarly, Lee et al. [12] used image projection on foldable materials to simulate flexible displays with variable form factors and dimensions. They did not conduct an evaluation of this system, but suggested that devices of this nature may have advantages in mobile contexts and will afford new interaction styles.

Designing Gestures

Wobbrock et al. [19] investigated user-defined gestures for tabletop computing using the Microsoft Surface through a participatory design and a *guessability* session [18]. They asked non-technical users to perform gestures for 27 typical

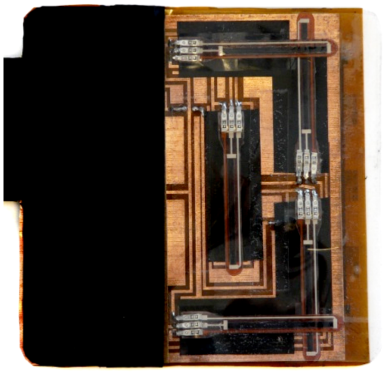


Figure 2. The back of PaperPhone, showing a Flexible Printed Circuit featuring an array of bend sensors.

computing actions. They used a measure of agreement between users to define a gesture set for each action. In a follow up study, Morris et al. [15] compared gestures for the Surface defined by users to those defined by interaction designers. They concluded that users preferred gestures that were generated by larger groups and generally favored the gestures created by users, as these tended to be conceptually simpler and less physically demanding. For our evaluation, we borrowed heavily from the basic methodology used in these papers, allowing users to generate, test and rank gestures for mobile computing tasks.

Lee et al. [13] conducted a study to generate a set of interaction gestures for mockup deformable displays as input devices. In this study, participants were given A4-sized paper, plastic and elastic cloth as imaginary displays. The participants were given 11 specific interaction tasks, such as zooming or navigating to the next page, and were instructed to deform the displays in ways that would execute these tasks. They found that users preferred pairings of closely related but opposite actions and gestures. This observation informed the design of our study.

PAPERPHONE: A FLEXIBLE SMARTPHONE

We anticipate that one of the first major commercial applications of flexible displays will be in handheld mobile devices [16]. There are several reasons for this. First, the flexible displays that arrive on the market will be limited in size for technical reasons. Second, many of the benefits of flexible displays, such as portability, are ideally suited for mobile form factors. Third, mobile devices benefit most from the power efficiency of electrophoretic displays. For these reasons, we developed PaperPhone, a smartphone prototype designed around a 3.7" electrophoretic display. PaperPhone features an array of thin film bend sensors on the back of the display (see Figure 2) that allows triggering of software actions on the device. Our prototype was designed to allow users to build their own bend gesture vocabulary, allowing us to study their preferences for mapping specific bend gestures to specific actions on the flexible display.

Apparatus

PaperPhone consists of an Arizona State University Flexible Display Center 3.7" Bloodhound flexible electrophoretic display, augmented with a layer of 5 Flexpoint 2" bidirectional bend sensors [6]. The prototype is driven by an E Ink [5] Broadsheet AM 300 Kit featuring a Gumstix [7] processor. The prototype has a refresh rate of 780 ms for a typical full screen gray scale image.

An Arduino [1] microcontroller obtains data from the Flexpoint bend sensors at a frequency of 20 Hz. Figure 2 shows the back of the display, with the bend sensor configuration mounted on a flexible printed circuit (FPC) of our own design. We built the FPC by printing its design on DuPont Pyralux flexible circuit material with a solid ink printer, then etching the result to obtain a fully functional flexible circuit substrate. PaperPhone is not fully wireless. This is because of the supporting rigid electronics that are required to drive the display. A single, thin cable bundle connects the AM300 and Arduino hardware to the display and sensors. This design maximizes the flexibility and mobility of the display, while keeping its weight to a minimum. The AM300 and Arduino are connected to a laptop running a Max 5 [14] patch that processes sensor data, performs bend gesture recognition and sends images to the display.

Recognizing Bend Gestures

PaperPhone has a training mode during which the user designs and records bend gestures, and an operating mode in which the system uses currently defined bend gestures to trigger software actions. In the training mode the bend sensor data is recorded and used to train a k-Nearest-Neighbor (kNN) algorithm with k=1. kNN assigns the label of the most similar examples (the closest neighbor) to the example to classify. In our case, the examples are vectors from the live values of the 5 bend sensors. We trained the system to recognize the flat shape as the baseline, or a neutral state. In the operating mode, in which trained bend gestures trigger software actions, a bend gesture is recognized when the display is bent to a curvature that is closer to a recorded shape than a flat shape. This recognition algorithm requires only a single training input for each gesture, making it ideal for rapid programming of user defined bend gestures.

To minimize the unintended triggering of actions by false positives, an additional stage of filtering was implemented immediately after the raw kNN classification output. The software takes a sample of the recognized bend gesture alternatives, reporting the mode value from this set as the recognized bend gesture. The window size of the sample ranged from 5 to 40 samples depending on the number of candidate bend gestures and on the similarity of these bend gestures to one another. This window size was manually defined based on observations of system performance. The final stage of the Max program maps the recognized bend gestures to a set of actions on the flexible display. For this purpose, we designed a state machine in Max that takes recognized bend gestures as inputs and produces states as output. The state data includes the specific action to be executed (such as placing a phone call), and the next state the state machine should be in on the next cycle (such as a menu for icon navigation). This information is transmitted to the Gumstix computer, which renders the appropriate images on the flexible display of PaperPhone. The state machine allows bend gesture pairs to be used in isolation and applied to all the individual actions, or used in concert to perform compound tasks. Although PaperPhone is fully flexible, the current design contains a number of fragile connectors on the left side of the display that may be damaged while bending. We protected these connectors by affixing a less pliable plastic board to this side. The right side of the PaperPhone display allows bends up to 45 degrees. Our bend gesture recognition system requires a minimum bend of 10 degrees for proper detection of bend gestures.

DEFINING BEND GESTURES

We defined a bend gesture as the physical, manual deformation of a display surface to form a curvature for the purpose of triggering an action on a computer display. To aid in the design of our study, we developed a simple classification scheme for bend gestures based on the physical affordances of the display, the sensing data available from the bend sensor array, and the PaperPhone bend gesture recognition engine. We classify the bend gestures our users could perform according to two main characteristics: the *location of the force* exerted on the display, and the *polarity* of that force. The rigid bezel allowed three fundamental locations of the force that can be exerted on the display: Bend gestures could be located on either right corners, or along the side of the display. Individual bend gestures could be of two sorts: a single bend or a compound bend. A single bend gesture contains only one fold, and is generated by applying a force to a single location. A compound bend consists of more than one fold, and is generated by applying forces to multiple locations simultaneously, e.g., bending both corners of the display. For each bend location, the *polarity* of a bend gesture could be either up (towards the user) or down (away from the user). Note that we recognize alternative criteria, such as the amount of force exerted on the display, the number of repetitions of bends, the velocity of movements, continuous vs. discrete use of bends and the orientation of the screen (portrait or landscape). However, given the constraints of our hardware, and in order to limit the overall time spent by participants designing bend gestures, we decided against investigating these in the present study.

BUILDING A FLEXIBLE INTERACTION LANGUAGE

We wanted users to build a simple interaction language for bend gestures: one that is both sufficiently general to be used universally, yet at the same time personalized and easy to reconfigure. In this language, bend gestures trigger individual actions on the PaperPhone system. We defined actions as the lowest verbalizable activities in the PaperPhone user interface [4]. Examples included selecting, navigating menu items, and ending a phone call. One of the goals of our study was to evaluate whether users would associate bend gestures with actions in a way that would approach the formation of a general interaction language, satisfying criteria of orthogonality, consistency, polymorphism and directionality [3, 22].

Orthogonality

Orthogonality, at a basic level, means that one bend gesture can be recognized as independent from another bend gesture, thus allowing each to map to a single action in a way that is combinatory [3]. We were particularly interested to see whether, at a semantic level, users associate orthogonal bending gestures to orthogonal actions of similar meaning. A design implication of this criterion is that orthogonal bend gestures can be conducted concurrently leading to predictable actions, e.g., the combination of two orthogonal bend gestures will result in a predictable outcome that is the direct combination of the two actions. When a right top corner up bend gesture moves the cursor to the left and a right bottom corner up bend gesture moves the selection point up, will users define a combination gesture that moves the selection point diagonally to the upper left corner?

Consistency

Orthogonality also leads to the question of consistency: a consistent design uses the same, or similar, bend gestures to trigger the same, or similar, actions across different applications. We were interested in whether users would, e.g., use the same bend gesture for moving down through a list of menu items as they would use to move down a selection of application icons.

Polymorphism

We were interested to see whether the same bend gesture would be used in a consistent manner, to trigger *different* actions that were related *semantically*. For example, if one chooses to bend the right side of the display down for a page forward action, would they choose it again to go to the next song? We examined whether polymorphism would reduce the diversity of gestures to a smaller set of favorites.

Directionality

Directionality refers to the spatial relationships defined or implied by the application (e.g., navigating *up* or *down* to select an icon). Directionality may be explicit, as is the case when icons are spatially distributed on a screen, or implicit, such as when navigating between pages of a document. Transitional animation effects can make implicit directionality explicit. We wondered whether users would, for example, associate an *up* action with bend gestures performed at the top of the display, and a *down* action with bends at bottom of the display, or if they would instead associate this with the polarity of the bend gesture, such as performing an upwards bend of the top corner for the *up* action, and a downwards bend of the same corner for the *down* action.

Polarity and directionality are distinguished by the item they relate to: polarity always refers to the physical deformation of display, bent towards (up) or away from (down) the user's body, while directionality always refers to either a spatial relationship on a screen (e.g., between icons) or mental model (e.g., previous/next page could be up/down, left/right).

Applications	Action Pairs	# Pairs	Session
Icon Navigation	Left – Right	3	2 3
	Up – Down		2 3
	Open – Close		2 3
Contacts	Up – Down	3	- 3
	Open – Close		2 3
	Call – Drop		2 3
Music Player	Play – Pause	2	2 3
	Next – Previous Song		2 3
Book Reader	Next – Previous Page	1	- 3
Map Navigation	Zoom In – Out	1	- 3

Table 1. Mobile applications and associated 10 action pairs, to which bend gesture pairs were mapped by participants.

USER-DEFINED BEND GESTURES STUDY

To determine what sets of bend gestures users would find appropriate as inputs for various actions in PaperPhone, we asked participants to define, design and evaluate bend gestures for specific functions in the context of a number of mobile applications. Our methodology was based on studies by Wobbrock et al. [19] and Lee et al. [13] on the participatory design of gestures for multi-touch tabletops as well as flexible display mockups.

Our study consisted of three sessions. In the first, we asked users to define a set of 8 bend gesture pairs. In the second session, we asked users to evaluate the appropriateness of each of these bend gesture pairs for each one of seven action pairs pertaining to three applications. They then selected their favorite bend gesture pair for each action pair. In the third session, users were asked to perform all available actions in each application.

Participants

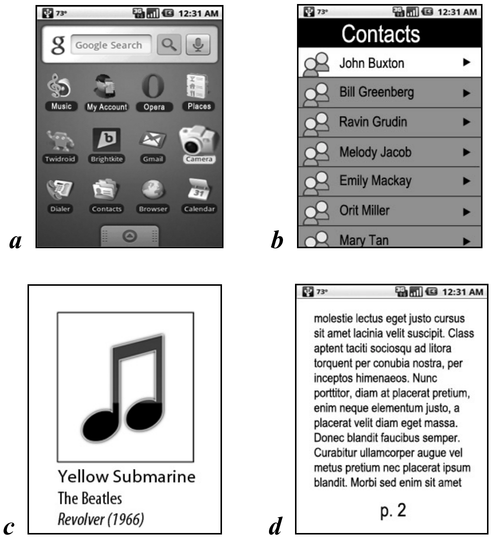
10 participants volunteered for this study (3 females). The participants were between the age of 19 and 36 (average of 23.7 years old). All participants were university students, and received $20 for their 2 hours of participation.

Applications and Action Pair Design

We selected five typical applications that are commonly performed on a mobile phone: navigating through icons, selecting contacts and making phone calls, playing music, reading a book, and navigating a map (see Table 1). Figure 3 shows four of the screen layouts on our PaperPhone prototype. Many user actions have a symmetrical correlate. We call such symmetrical actions *action pairs.* We identified 20 actions (10 action pairs) for the five applications.

(a) Icon Navigation

The user was required to navigate a series of twelve application icons distributed in a 3*x*4 grid pattern (see Figure 3a). They were asked to perform these actions by going *left*, *right*, *up* and *down*. *Opening an application* led to a splash screen. The user could *close the application* which returned the interface to the set of application icons.



molestie lectus eget justo cursus sit amet lacinia velit suscipit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nunc porttitor, diam at placerat pretium, enim neque elementum justo, a placerat velit diam eget massa. Donec blandit faucibus semper. Curabitur ullamcorper augue vel metus pretium nec placerat ipsum blandit. Morbi sed enim sit amet

Figure 3. Screenshots of 4 of the applications: Icon Navigation, Contacts, Music Player, and Book Reader.

(b) Contacts

The user was asked to navigate *up* and *down* a list of contacts (see Figure 3b). Once the user had chosen a contact, she could *select* it to view the contact details. The user could *close the contact details* and return to the main list, or *call the contact.* When calling, the user could *drop the call*.

(c) Music Player

The user was asked to *play* and *pause* a song, and select the *previous* or *next song* (see Figure 3c). To minimize bias, we provided no visual or verbal cues of the directionality of these actions. When the *play* or *pause* action was performed, the state of the action was displayed on the screen. When new songs were selected, the name of the song and performer was also visible.

(d) Book Reader

The user was asked to navigate to the *previous* or *next page* (see Figure 3d). We again avoided introducing directional bias by not asking users to *page up, down, left* or *right*. We limited actions for this application to a single action pair to allow us to observe the user's orthogonality considerations in applying this mapping.

(e) Map Navigation

The user was asked to *zoom in* or *zoom out* (not shown in Figure 3). Because of the limited refresh rate of the display, zooming was implemented as a discrete action. We again limited this application to a single action pair.

Procedure

Before starting the experiment, users were provided with minimal instructions to prevent damage to PaperPhone. We physically demonstrated a single bend gesture (Figure 4A), emphasizing the degree to which the display could be bent without damaging the device. We instructed the users to avoid bending directly on the left edge of the device, where the electrical contacts were located. We guided the participants to hold the display as if it were wireless, and to ignore

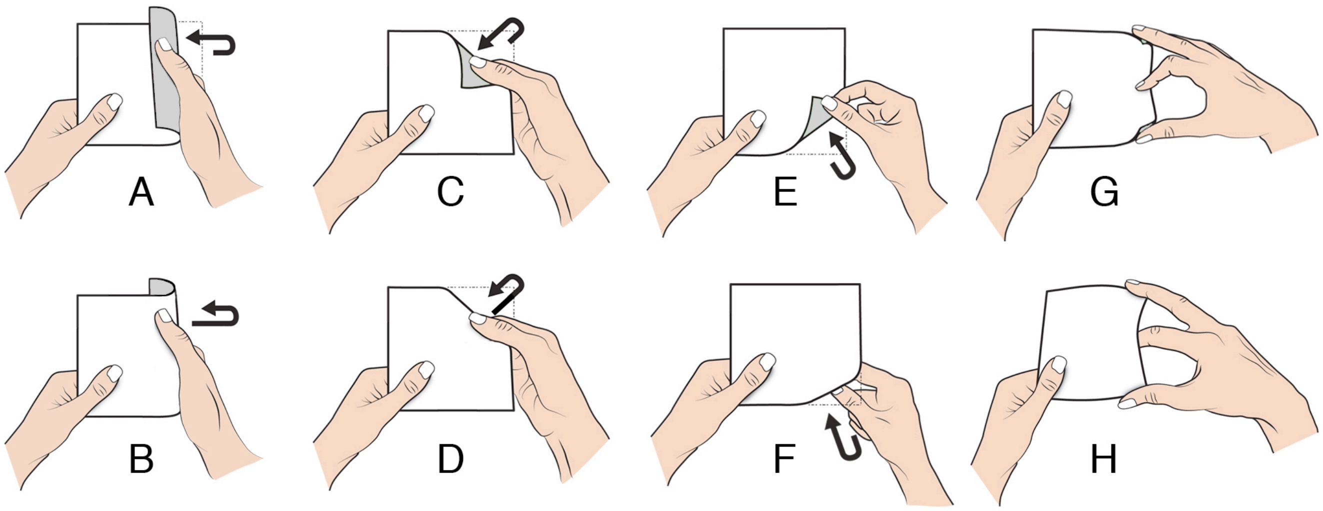


Figure 4. The eight participant defined individual bend gestures used in bend gesture pairs.

and not hold the connecting ribbon cables. Participants were informed that the system would only recognize discrete bend gestures. Aside from this, we did not instruct participants on bend gestures. Throughout the experiment, participants were encouraged to think aloud, so as to verbalize their thought processes.

Session 1: Defining Bend Gestures

To encourage users to consider a wide variety of bend gestures, their first assignment was to design 8 unique pairs of bend gestures. We derived 8 as the number of bend gesture pairs empirically from a pilot study: a high enough number to challenge beyond obvious choices, while allowing completion within 2 hours. Participants were allowed to reuse individual bend gestures in different pairs, as long as the resulting pairs were not identical. First, the user executed each bend pair once to train PaperPhone's bend recognition system. After the system was trained, it executed an action whenever the bend gesture was performed. To emphasize that each bend gesture was going to be associated with an individual action, and to encourage participants to create comfortable bend gestures, we gave the users the opportunity to try out their bend gesture with an abstract action. Here, the display turned either to black or white when the user performed a bend gesture pair successfully. This continued until they had defined all 8 pairs.

Session 2: Assigning Bend Gestures to actions

The second part of the experiment let users test out each bend gesture pair with each individual action pair. We selected 7 unique action pairs from the list of 10 (Table 1). The *up/down* action pair from the Contacts application was not repeated, as it is a duplicate of the *up/down* action pair in the Icon Navigation application. To examine orthogonality, we reserved the Book Reader and Map Navigation applications for evaluation in session 3. In the Icon Navigation application, users moved *left, right, up, down through icons, opened* and *closed the application.* In the Contacts application, users *opened* and *closed a contact, called the contact* and *dropped the call.* In the Music Player, users *played* or *paused*, and selected the *next or previous song.*

Users first assigned the mapping of each bend gesture to an action, meaning that they selected which bend gesture components of the previously designed bend gesture pair would trigger the individual actions in the action pair. The user was then able to try out each bend gesture pair/action pair mapping, after which they rated the appropriateness of the bend gesture pair for this action pair using a 5-point Likert scale of agreement (1 Strongly Disagree-5 Strongly Agree). This was repeated for all 8 bend gesture pairs. The participants were then asked to determine their favorite bend gesture pair for the action pair. When a user suggested an alternative bend gesture pair, we would record this pair and add it to our total count of bend gesture pairs. Users each tested 56 mappings of bend gesture pair to action pairs (8 bend gesture pairs x 7 actions pairs). The presentation of bend gesture pairs for each action pair, as well as of action pairs, was counterbalanced using a Latin-square design.

Session 3: Using Bend Gestures across Applications

For the final part of the study, the users were instructed to try out the full suite of top ranked bend gesture pair/action pair mappings, in each of the five applications. In the previous part, each action pair was performed individually. In this session all of the action pairs for the active app were available at once, allowing users to perform them in any order, independently of the pairs. Users were free to assign any bend gesture pair to any action pair, with any polarity, whether previously used or not. Users were reminded of their favorite bend gesture/action mappings for each application and were instructed to determine whether there were any conflicts between these bend gestures. In the case of orthogonality conflicts, the user was invited to revise their choice of bend gestures to eliminate any conflicts. For each app, the system was trained with the selected bend gestures and the user was allowed to freely test and evaluate the interaction experience. Before ending the experiment, users

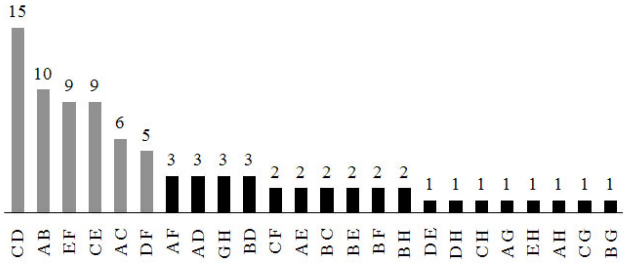


Figure 5. The 24 unique bend gesture pairs generated and frequency (Letters refer to individual bend gestures from Fig. 4. Gray indicates six most frequently used pairs).

Applications	Action Pairs	CD	AB	EF	CE	AC	DF
Contacts	Call - Drop	3.5	4.4	3.1	3.3	3.2	3.0
	Open - Close	3.8	4.9	4.0	3.5	4.2	3.6
Music Player	Next - Previous	4.1	4.3	4.2	3.6	2.8	3.6
	Play - Pause	4.1	4.0	3.7	3.1	3.2	4.2
Icon Navigation	Left - Right	3.6	4.1	3.8	1.9	2.2	2.4
	Open - Close	3.7	4.1	3.7	3.1	3.8	3.2
	Up - Down	4.0	3.6	4.3	4.4	2.0	4.6

Table 2. Appropriateness scores per action pair (1-5 scale, 5 being most appropriate). Gray cells highlight the action pairs with the highest appropriateness value.

were asked to identify situations where they would prefer to use bend gestures over other input techniques.

RESULTS

The first session in the experiment generated 8 bend gesture pairs per participant, for a total of 80 bend gesture pairs. A few participants created bend gesture pairs in the 2nd session (7 additional bend gesture pairs), for a total of 87 pairs (174 individual bend gestures). We first identified high-frequency individual bend gestures. Four HCI researchers grouped each bend gesture, according to the location and polarity of the force exerted on the display, such that each group only contained identical bend gesture. The same procedure was repeated for bend gesture pairs. We did not consider the order of the bend gestures in the pair.

A total of eight individual bend gestures were identified out of a possible set of ten: six single bends and two compound bends, illustrated in Figure 4. Individual letters identify individual bends (e.g. *A*). Bend gesture *C* was the most frequent used at 20.9% (36 out of 172 individual bends). The other five single bend gestures obtained an average frequency of 14.1% (24/172). Two compound bends constituted 8.7% of the total individual bend gestures (15/172).

A total of 24 unique pairs were identified, from a possible set of 45. Pairs of letters indicate which individual bend gestures constitute each bend gesture pair (e.g. *AD*). Their composition and frequency is shown in Figure 5. Six bend gesture pairs obtained a frequency of five or more, meaning they were performed by at least half of the participants: (*CD, AB, EF, CE, AC* and *DF)*. We consider those six bend gesture pairs to be our *most frequent used bend gesture pairs*. The six bend gesture pairs are identified with a gray color in Figure 5. We focused our analysis of session 2 on these six bend gesture pairs.

Applications	Action Pair	Agreement	Unique Bend Gesture Pairs
Contacts	Open - Close	0.52	5
	Call - Drop	0.28	4
Music Player	Next - Previous	0.38	3
	Play - Pause	0.26	5
Icon Navigation	Left - Right	0.44	3
	Open - Close	0.24	6
	Up - Down	0.16	7

Table 3. Agreement for each action pair from the user's favorite bend gesture pairs.

Session 2: Bend Gesture Pairs for Action Pairs

For each bend gesture pair defined in the first part of the experiment, users rated their appropriateness for a series of action pairs. Table 2 shows the frequency distributions of these appropriateness ratings per action pair.

Agreement on Favorite Bend Gesture

To identify the best bend gesture for each action, we looked at the bend gesture pairs identified by each participant as their favorite for that action pair. For each action pair, we calculated a measure of agreement, as defined in Wobbrock et al. [18, 19]. The agreement score reflects the degree of consensus among participants. An agreement score of 1 indicates that all the participants selected the same bend gesture pair as their favorite, while an agreement of 0 indicates that every participant selected a different bend pair. Table 3 shows this agreement score for every action pair. Agreement was highest for *open-close* in Contacts (A_{OC}=.52) and *left-right* in Icon Navigation (A_{LR}=.44).

Polarity of Favorite Bend Gestures

We observed the polarity of the individual bend gesture in each pair as it related to each individual action. In the case of two identical polarities (e.g. two upwards bend gestures), we define polarity as the location of the bend (top, side or bottom of the display). The *left/right* action pair in the Icon Navigation application had a 100% polarity agreement, with all users performing an upward bend gesture for *left*, and a downward bend gesture for *right*. Nine out of ten participants associated the *open* action in Icon Navigation with an upward bend gesture, and the *close* action with a downward bend gesture. We observed that the *up* action corresponded to either an upward bend gesture (6 participants), or a top (location) bend gesture (3 participants), while a *down* action corresponded with either a downward bend gesture (5 participants) or a bottom bend gesture (3 participants). For the remainder of the applications, the actions were approximately equally distributed between two polarities.

Applications	Action Pair	Agreement	Unique Bend Gesture Pairs
Book Reader	Next - Previous	0.42	3
Contacts	Call - Drop	0.26	5
	Open - Close	0.66	3
	Up - Down	0.30	4
Map Navigation	Zoom in - out	0.42	4
Music Player	Next - Previous	0.38	3
	Play - Pause	0.40	5
Icon Navigation	Left - Right	0.38	3
	Open - Close	0.30	4
	Up - Down	0.28	5

Table 4. Agreement for each action pair from the user's application bend gesture pairs.

Session 3: Bend Gestures for Applications

Agreement on Bend Gestures in Applications

We calculated the agreement among participants for each of the 10 action pairs. Table 4 shows the agreement score for every action pair.

Orthogonality in Applications

We extracted the bend gesture pairs used in applications by each participant, creating either sets of 2 pairs (for the Music Player), or 3 pairs (for the Contact and the Icon Navigation application). We counted the frequency of those pairs, and calculated the agreement score. We observed a higher consensus in applications with three action pairs: the majority of participants selected the trio of bend gesture pairs *AB, CD,* and *EF* in the Icon Navigation applications (A_{IN}=0.66, 8 participants), and the Contacts applications (A_C=0.40, 6 participants). The Music Player obtained an agreement score of A_{MP}=0.32, as participants selected either the set of bend gesture pairs *AB* and *CD* (5 participants), or *CD* and *EF* (2 participants).

Polarity of Bends in Applications

The majority of the bend gesture pair/action pair mappings were consistent in terms of their polarity. 10 (out of 10) participants selected downward bend gestures for the *right* action and upward bend gestures for the *left* action in the Icon Navigation application. 8 participants selected a downward bend gesture for *zooming in* and an upward bend gesture for *zooming out*. 8 participants selected an upward bend gesture for *calling*, and 7 participants selected a downward bend gesture for *dropping* a call.

DISCUSSION

The results show that participants express strong agreement when designing individual bend gestures as well as bend gesture pairs. However, they agreed less on the assignment of bend gesture pairs to action pairs. Specifically, we found a cohesive set of bend gesture pairs with high frequency, and a cohesive set of individual bend gestures, indicating agreement. However, the consensus on the mapping of those bend gestures to actions was overall low, showing that each participant had his or her own preference. This has strong implications for the design of flexible display user interfaces that use bend gestures as a source of input.

Cohesive Set of Bend Gestures and Bend Gesture Pairs

When examining the bend gestures and bend gesture pairs in isolation, without their action mappings, the set of six most frequent gesture pairs are all composed of simple individual bend gestures. From the six identified bend gesture pairs, we can identify a subset of three that were both the most frequently designed (in session 1), and the most frequently assigned in applications (in session 3), with high agreement. We believe that those three bend gesture pairs (*AB*: side of display up/down, *CD*: top corner up/down and *EF*: bottom corner up/down) likely form a good foundation for a simple bend gestural interaction language. The three bend gesture pairs both consisted of the simplest individual bend gestures, and were also orthogonal to one another. We also observed their repeated and consistent use amongst different applications in session 3. We believe that individual bend gestures and bend gesture pairs that are conceptually simpler, and less physically demanding, were purposefully selected by users with higher frequency and appropriateness, an observation similar to that of Wobbrock et al. [19] with their set of gestures for multitouch input.

Overall Favorite Bend Gesture Pair

When assigning appropriateness scores for bend gesture pair/action pair mappings, we found that the bend gesture pair *AB* was rated the highest for the majority of action pairs (5 out of 7). The appropriateness of bend gesture pair *AB* was even higher for the contacts application. This indicates that *AB* was the favorite bend gesture pair amongst participants in this study. Note, however, that it was also considered the least appropriate for the *up/down* action pair in the Icon Navigation application. One likely reason for this is that the *AB* bend gesture pair was the least spatially ambiguous, as the bend gesture was on a vertical axis. Additionally, we observed that the Contacts *open* and *close* action pair had the highest agreement score both in the second and third session. In both cases, the large majority of participants mapped this action pair to the *AB* bend gesture pair (70% and 80% of participants, respectively).

Building a Bend Gesture Interaction Language

We proposed four criteria required when creating a bend gesture interaction language. We were interested in determining whether participants would naturally integrate each criterion in their assignment of bend gestures to actions in applications when more than one mapping was required.

Orthogonality

In terms of *orthogonality*, users did understand and respect the need to associate a unique bend gesture to each action. If their mapping of bend gesture pairs to action pairs in the second session was not orthogonal when applying them to applications in the third session, they updated those mapping to find a set that was orthogonal. Approximately 42% of all mappings changed for this reason.

Consistency

We found no strong evidence of *consistency* among the action pairs present in more than one application (i.e. *open* and *close* action pair in the Contacts and Icon Navigation applications). Only with the action pair of moving *up* or *down* did a majority (6 participants) choose the same bend gesture pair in both Contacts and the Icon Navigation applications. This is partly due to the fact that orthogonality plays a large role in assigning a bend gesture pair to an action pair. We believe this has implications for the design of flexible user interfaces in that designers may be better able to preserve consistency amongst applications than users.

Polymorphism

Polymorphism, which dictates the use of a bend gesture across different actions, did not reveal any consensus. Two action pairs with similar meaning, the page forward and backward, and skipping to the next or previous song, obtained little to no agreement in the bend gestures associated with them. Because the design of the study dictated the use of action pairs, we did not include a *symmetry* criterion, which would require symmetrical bend gestures to be used with symmetrical actions. However, participants still considered the relative symmetry of actions and the bend gestures used to trigger these actions. One user in particular described not liking using, what he considered, symmetrical bend gestures for actions he did not consider symmetrical. He observed that when bend gestures were symmetrical, it was more difficult to recall the polarity of his mappings.

Redundancy

Redundancy is a criterion where multiple bend gestures may be programmed to activate the same action. Our experiment was not designed to test for redundancy. However, because users evaluated many bend gestures for a single action in the second part of the study, we can extrapolate that it would be possible, and suitable, to provide the user with redundant bend gestures. For instance, the appropriateness scores were very close for three bend gesture pairs in action pairs in the music player application. Selecting the *previous* or *next* song can be accomplished with bend gesture pairs *AB*, *CD* or *EF* with similar appropriateness results. *Playing* or *pausing* the music yielded comparable scores whether mapped to *AB*, *CD* or *DF*. All appropriate bend gestures could be redundantly assigned to these actions, when available.

Directionality

Spatial and *directional cues* did play an important role in the mapping of bend gestures to actions. The Icon Navigation application included actions with a clear spatial relationship *(up/down/left/right*). For other actions, such as *opening* and *closing* applications, spatial relationships appeared based on mental models constructed by the participants. In particular, participants described the action of opening an item as pulling the information towards them, or opening a door. As actions with strong directional cues showed consensus on the polarity of the associated bend gestures, we believe bend gestures that take directionality into account will likely seem more natural to users [11]. Bend gestures were mapped to directionally signified actions in a variety of ways. However, the *directionality* of actions was not clearly defined from our data. We did observe a similar scenario with the polarity of bend gestures in application actions as we did with the individual actions. The action of selecting an icon *left* was strongly coupled with an upwards bend gesture, and the *right* action with a downwards bend gesture. Bend gestures performed on either corner (a diagonal axis) were logically mapped to both *up* and *down* actions. Had the entire display been flexible, with equivalent bend gestures on all opposite corners and sides, we would expect to see more opportunities for the criteria to be addressed.

Physical Affordances of PaperPhone

Users consistently reported that bending the corners of the display was easier than bending the whole side of the display. Three users reported bending the lower right corner down to be a more comfortable gesture than bending the same corner up, as the result of the angle of their wrist when holding this corner. They had more range of motion in one direction than the other, and needed to change their grip to compensate. Gestures such as bending two corners at once were also described as requiring more physical effort. Few participants generated, preferred, or used compound bend gestures in complex applications (below 9% overall). In addition, while the recognition engine supported it, no user defined compound bend gestures with opposite polarities as they were physically challenging. One user specifically commented about how it seemed natural to use bend gestures on PaperPhone to navigate *left* and *right* but found it challenging to find bend gestures that seemed appropriate to navigate *up* and *down*. This user preferred bending the entire vertical side of the display up and down to navigate *left* and *right*. Because it was not possible to bend the top or the bottom side of the display in the same way, this user could not chose an equivalent bend gesture to navigate upwards and downwards. Several users spoke about how much it would help to have the entire display be flexible and could clearly see how this would afford more input options. One user said that they would have preferred to use the device in a landscape orientation if one edge had to be kept rigid so that they could make bend gestures with both hands on left and right corners.

Mental Models

Users described how mental models of the actions and of the display affected the bend gesture pairing and polarity choices. These mental models were influenced by metaphors, such as: Viewing the display as a book; Prior experiences with GUI layouts; Physics models, such as inclined planes on which icons slide; and iconic representation of actions such as the right pointing arrow used for play on music players. Several users specifically described liking bend gestures for navigating the pages of a book because of the physical similarity to flipping pages in real books. The *zoom-in* action was commonly defined with bending the display in a convex shape in relation to the user. Users explained this by observing that with this bend gesture, the middle of the display was moving towards them. Several users described mental models in which more complex or physically challenging bend gestures would be reserved for applications that were performed less frequently or had a higher psychological significance, such as dropping a phone call. One user who used a compound bend gesture to drop a call described the bend gesture as “crushing the call”.

Making the Case for Bend Gestures

Users saw potential for the use of bend gestures when wearing gloves, which inhibit touch screen interactions. They also imagined usage by people with motor skill limitations that prevented the use of other input systems. Bend gestures were recognized as potentially usable without visual engagement with the device and when one was interacting directly with the display but needed to avoid occluding areas of the display. Users reported bend gestures as appropriate for navigating pages in a book reader, which could take advantage of the analog properties of the bend gesture to allow for variable speed scrolling based on the degree of bend. Zooming in and out of a map was also noted, but several participants specifically described wanting this function to be implemented as a continuous analog control.

Limitations and Future Directions

The main limitation of this work resides in the physical engineering of the prototype display, which restricted bending to one side of the display. This reduced the number of bend gestures available for consideration. We believe this limitation did not outweigh benefits of being able to evaluate a functional flexible display, with results representing a significant subset of findings for a full flex display. While it was possible for us to detect continuous (analog) bend gestures, the slow refresh rate of flexible E Ink delayed visual feedback, making real-time animation impossible. Effects of display size on the use of bend gestures may be answered through future studies: We believe that with appropriate material qualities, bends could apply from small to large form factors. We expect touch input to complement bends and recognize the challenges this presents: current flex touch input options are limited. In addition, our study proposed a maximum of six actions per application, which was the max number of single bend gestures available given our constraints. An important step to validate our bend gesture set would be to test compound applications with four action pairs or more. Finally, it would be interesting to perform a follow-up study that compares user generated bend gestures mappings with those produced by designers [15].

CONCLUSION

In this paper, we presented PaperPhone, a smartphone with a functional flexible electrophoretic display and 5 integrated bend sensors. We studied the use of user-defined bend gestures for triggering actions with this flexible smartphone. Results suggest a strong preference for 6 out of 24 bend gesture pairs. In general, users selected individual bend gestures and bend gesture pairs that were conceptually simple and less physically demanding. There was a strong agreement among participants to use 3 particular bend gesture pairs in applications, bending the: (1) side of display, up/down (2) top corner, up/down (3) bottom corner, up/down. For actions with a strong directional cue, there was strong consensus on the polarity of the bend gestures. Results imply that gestures with directional cues are preferred. Results suggest bend gestures form a useful addition to interaction modalities of future flexible computers.

ACKNOWLEDGMENTS

We thank Jann Kaminski & Nicholas Colaneri of the ASU Flexible Display Center, Seth Bishop & Michael McCreary of E Ink Corporation, fuseproject and Autodesk Research for their support. This project was funded by the Ontario Research Fund and by an NSERC Strategic Project Grant.

REFERENCES

1. Arduino. http://www.arduino.cc.
2. Balakrishnan, R., Fitzmaurice, G., Kurtenbach, G., and Singh, K. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In Proc. Interactive 3D Graphics. I3D '99 (1999), 111-118.
3. Bowdon, E., Douglas, S. and Stanford, C. Testing the Principle of Orthogonality in Language Design. Human-Computer Interaction (4) 2 (1989), 95 – 120.
4. Card, S.K.; Thomas, T.P.; Newell, A. The Psychology of Human-Computer Interaction London: Lawrence Erbaum (1983).
5. E Ink Inc. http://www.eink.com
6. Flexpoint Inc. http://www.flexpoint.com
7. Gumstix Inc. http://www.gumstix.com
8. Gallant, D.T., Seniuk, A.G., and Vertegaal, R. Towards more paper-like input: flexible input devices for foldable interaction styles. In Proc. UIST '08 (2008).
9. Herkenrath, G., Karrer, T., and Borchers, J. Twend: twisting and bending as new interaction gesture in mobile devices. CHI 2008 Extended Abstracts (2008).
10. Holman, D., Vertegaal, R., Altosaar, M., Troje, N., and Johns, D. Paper windows: interaction techniques for digital paper. In Proc. CHI '05 (2005), 591-599.
11. Jacob, R.J., Girouard, A., Hirshfield, L.M., et al. Reality-based interaction: a framework for post-WIMP interfaces. Proc. CHI’08 (2008), 201–210.
12. Lee, J.C., Hudson, S.E., and Tse, E. Foldable interactive displays. In Proc. UIST '08 (2008), 287-290.
13. Lee, S. et al. How users manipulate deformable displays as input devices. In Proc. CHI '10 (2010), 1647-1656.
14. Max 5. http://www.cycling74.com
15. Morris, M.R., Wobbrock, J.O., Wilson, A.D. Understanding Users’ Preferences for Surface Gestures. Proc. of Graphics Interface (GI '10) (2010), 261-268.
16. Readius. http://www.readius.com
17. Watanabe, J., Mochizuki, A., and Horry, Y. Bookisheet: bendable device for browsing content using the metaphor of leafing through the pages, Proc. UbiComp (2008), 360 - 369.
18. Wobbrock, J.O., Aung, H.H., Rothrock, B. and Myers, B.A. Maximizing the guessability of symbolic input. Ext. Abstracts CHI '05 (2005), 1869-1872.
19. Wobbrock, J.O., Morris, M.R., and Wilson, A.D. User-defined gestures for surface computing. In Proc. CHI '09 (2009), 1083-1092.
20. Schwesig, C., Poupyrev, I., and Mori, E. Gummi: a bendable computer. In Proc. CHI '04 (2004), 263-270.
21. Sellen, A. J. and Harper, R. H. The Myth of the Paperless Office MIT Press (2003).
22. Stroustrup, B. The C++ Programming Language Special Edition. O'Reilly Media, Inc. (2000).