VersaPen: An Adaptable, Modular and Multimodal I/O Pen

Marc Teyssier
LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
Paris, France
marc.teyssier@telecom-paristech.fr

Gilles Bailly
Sorbonne Universités, UPMC
Univ Paris 06, CNRS, ISIR
Paris, France
gilles.bailly@upmc.fr

Éric Lecolinet
LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
Paris, France
eric.lecolinet@telecom-paristech.fr

CHI'17 Extended Abstracts, May 06-11, 2017, Denver, CO, USA
ACM 978-1-4503-4656-6/17/05.
http://dx.doi.org/10.1145/3027063.3053159

Abstract
While software often allows user customization, most physical devices remain mainly static. We introduce VersaPen, an adaptable, multimodal, hot-pluggable pen for expanding input capabilities. Users can create their own pens by stacking different input/output modules that define both the look and feel of the customized device. VersaPen offers multiple advantages. Allowing in-place interaction, it reduces hand movements and avoids cluttering the interface with menus and palettes. It also enriches interaction by providing multimodal capabilities, as well as a mean to encapsulate virtual data into physical modules which can be shared by users to foster collaboration. We present various applications to demonstrate how VersaPen enables new interaction techniques.

Author Keywords
Pen-based interaction, Modular interaction, Adaptable device

ACM Classification Keywords
H.5.2. [Information Interfaces and Presentation]: User Interfaces

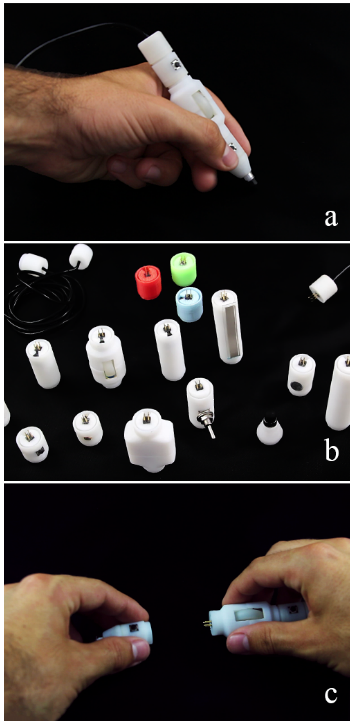

Figure 1: a) VersaPen is a device with the same form factor as traditional pens. b) It is composed by a variety of sensor and display modules that c) can be stacked together to create customized digital pens.

Introduction

User customization is broadly used in off-the-shelf software. Users can choose the position and the content of palettes or change the correspondence between hotkeys and commands. In contrast, physical devices such as keyboards, mice or stylus remain mostly static. They have few input and output capabilities that users can seldom customize for adapting them to their tasks and applications.

In this paper, we investigate the benefits of *adaptable devices*, which are highly customizable in terms of input and output capabilities. We instantiate our approach in the context of pen-based interaction. While stylus and digital pens are appreciated for their precision and their form factor, which leverage the dexterity of the human hands [10], their input/output (I/O) capabilities remain limited.

We propose VersaPen (Figure 1), an adaptable device which leverages pen-based interaction. VersaPen relies on a set of versatile input/output modules including sensors and displays that can be easily combined by end users by stacking them together while preserving the form factor of a standard pen.

Our primary contributions are (1) the presentation of an open-source [18] adaptable and modular interaction device; (2) its implementation offering expressiveness while preserving the affordance of a pen (3) a set of interaction techniques, samples of opportunities raised by VersaPen.

Related work

Pen-based interaction. Pen augmentation has been proposed in several studies to alleviate the problem of mode switching, for instance for switching between drawing, annotating and issuing commands, or more generally for selecting modes [15, 26]. To achieve this, various input modalities have been considered such as tilting, rolling [26], shaking [24] the device, as well as pressure [19], grasp [21], near-surface interaction [17], speech recognition (LiveScribe) or just using an additional scroll wheel (Wacom Air Brush). Previous studies also considered different output modalities, such as visual feedback (using LEDs [16], embedded screens or projectors [22]), audio feedback or haptic feedback [14]. These technologies propose new pen-based interactions, but each one requires a specific device and works with prototype applications. VersaPen embeds many of these I/O modalities and allows users to rapidly prototype and customize devices by combining them. It enables exploring different form factors by adding or removing modules or by assembling them differently.

Several studies recently combined pen and touch interaction [1, 7] to leverage the benefits of two-handed interaction. VersaPen also supports two-handed interaction. Not only users can interact with the non-dominant hand on a multi-touch surface, but the pen can be extended so that modules can serve as remote auxiliary controls with the other hand.

Adaptable interaction. Applications, video games, graphical toolkits provide various mechanisms such as themes, skins, or styles for customizing user experience (interface, inputs). Other systems have also been proposed for creating more adaptable interfaces in research [23, 25, 3], but these studies focus on software rather than device customization. The Icon toolkit [2] provides a way of adapting input interaction for multimodal interactions, but does not allow customizing each device individually.

Modular tangible interfaces. Some modular devices are now commercially available such as modular keyboards (e.g *Azio Levetron Keyboard* [11]) or the Google Ara smartphone. While users can modify their shape, the interaction techniques remain essentially the same. Tangible User Interfaces (TUIs) also aim at the customization of physical

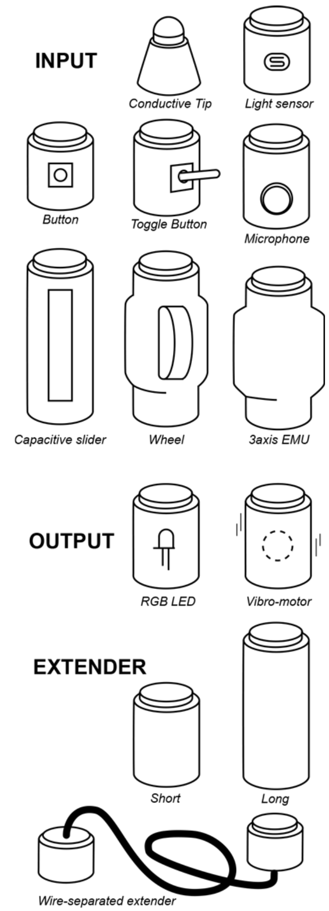

Figure 2: List of available modules. It includes input and output modules and extender modules.

interfaces. For instance, Phidgets [4], d.tools [6] or Calder [13] provide toolkits including a set of controls, sensors and actuators allowing quick prototyping of physical interfaces. However, the target users of these systems are developers or product designers rather than the end users who will manipulate these devices. In contrast, MagGetz [8] or WoodooIO [28] propose a set of user-customizable physical controls but their form factor is not appropriate for pen-based interaction. The tangible device proposed in [9] shares some characteristics with VersaPen such as modularity, interchangeability and the capacity to plug-in parts. However this is a tool for one specific task and doesn't share versatility features and handleable form factor.

VersaPen

VersaPen enhances pen-based interaction by alleviating the problem of mode switching between for instance drawing tools or controls such annotations. VersaPen is a modular, adaptable, hot-pluggable pen allowing users to customize their own pen on-the-go while preserving the shape of a regular pen. The users can map each input/output module of the pen to a control or mode from their favorite applications providing multiple shortcuts at the finger tip.

VersaPen offers multiple advantages. First, users can create their own instruments adapted to their tasks and applications by physically customizing a unique device. VersaPen also reduces hand movements and cursor round trips between menus, toolbars, palettes and the objects of interest because the controls are attached to the physical device. Placing several sensors close to each other makes it easier to use them in parallel and favors multimodal interaction. Finally, VersaPen provides a mean to encapsulate data, software settings or behaviors in physical modules. These physical containers can be shared among users and foster collaboration.

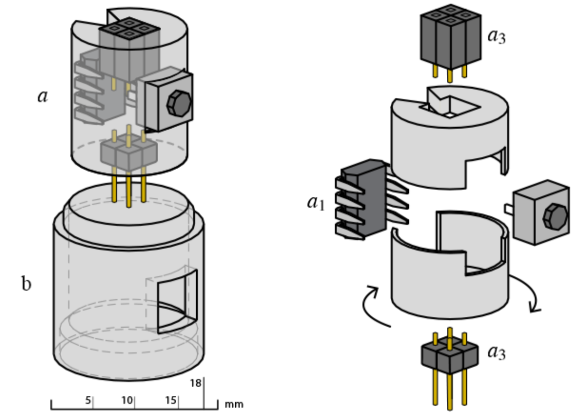

Figure 3: VersaPen module Design. A module is composed of a) an inner cylinder with embedded electronics which can be rotated, a_1) a microcontroller, a_2) a sensor, a_3) male-female connectors. This element fits in a b) outer shell.

The core components of VersaPen are its modules. A module (Figure 3) is a cylinder containing a microcontroller and a sensor or a display, that can be added and removed as needed.

Implementation

Form Factor. A module contains two 3D-printed cylinders: an inner tube (Figure 3 *a*) with a diameter of 14mm and an outer shell (Figure 3 *b*) of 18mm. The inner cylinder contains the electronic parts and fits into the outer shell to form the final module. This design allows the user to adjust the orientation of a sensor or a display by rotating the module up to $\pm 60°$. The height of each module depends on the embedded electronics, with a minimum of 22mm. The form factor results from a compromise to achieve sufficient rigidity avoiding deformations while being small enough to enable a comfortable grasp, as with regular pens (available digital pens diameter varies between 19mm to 12mm).

Modules Capabilities. We built a set of 15 modules (Figure 2), which could easily be extended. This set includes various input modules such as a button, a mouse wheel, an accelerometer, a touch sensor and a microphone. The tip of the pen is a conductive module compatible with a capacitive screen and is sensitive to pressure. Output modules provide visual feedback, through RGB LED modules, or haptic feedback, via a module including a vibro-motor. These modules can provide information about the current modes or settings, or for issuing various notifications. VersaPen also provides modules without I/O capabilities to encapsulate and share data (software behaviors, predefined settings, etc.). These modules do not include a memory chip (size constraint) but have a static unique ID (stored in 2 bytes) to make a link with data stored on a remote server.

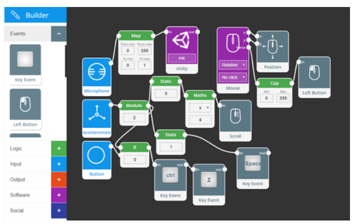

Figure 4: VersaPen flow-based programming interface.

In order to improve ergonomics, empty modules with different heights (which only contains wires) serve to precisely position the other modules when assembling the pen. One of them is divided in two parts connected with a 40cm long flexible wire. This module allows two-handed interaction: the mobile part controls the behavior of the pen with the non-dominant hand.

Embedded Electronics. Each module encloses an *Atmel Attiny85* 8 bit microcontroller (Figure 3 a_1) which manages the embedded sensor/actuator (Figure 3 a_2) and handles the communication between modules. This controller is small enough (< 9.5mm) to fit in the pen-sized container. Modules are cost-effective as their total components cost about $4.

Communication and Data Processing. Messages from the entire stack are passed via a shared bus to an Arduino controller connected to the computer. Each module acquires sensor data locally. The data is transmitted to the top neighbor module and so on until it reaches the last module. Modules are connected by male-female connectors (Figure 3 a_3) with four pins (two serving for power and two for data transmission) that also reinforce the rigidity of the pen. Data are differentiated by the ID of the module involved and the type of sensor attached. These IDs also makes possible to send specific instructions to a given output module. The computer receives the data at a rate of 10ms, which allows fluid interaction with the main application. A *Nodejs* server translates incoming messages and streams them via TCP to the software managing events.

Software implementation. A flow-based interface allows end users to connect the various input and output sources to computer applications through visual programming (Figure 4). This software, built in JavaScript, simulates system mouse and keyboard events to trigger commands in existing applications. For more advanced interaction with applications we implemented an OSC bridge, a standard which is used for add-ons in many off-the-shelve programs. When the software recognizes a specific pen configuration that has already been programmed, parameters are restored and the user can resume his work immediately.

Interaction techniques and Applications

To showcase the possibilities of VersaPen, we highlight its main capabilities and illustrate how it allows creating new tools adapted for specific needs and applications.

In-place interaction

Selecting items on menus, toolbar or palettes requires cursor round-trips between the objects of interest located on the center of the screen and graphical widgets, usually in the periphery of the screen. In contrast, VersaPen favors "in-place" tool switching [5]. Users can rapidly change tools without moving the pen from the object of interest. We used this ability to enhance a 3D world sculpting application (Fig-

Figure 5: VersaPen in use with a 3D sculpting tool (*Unity3D*). Different input methods are combined to improve workflow and adapt input methods for specific tasks.

ure 5). The pressure and slider modules were respectively used to control the topology of the 3D object and the orientation of the camera while the wheel module provided access to the texturing functions.

Multimodal & parallel interaction
VersaPen allows users to control multiple *continuous* sources of input at the same time. For instance, users can simultaneously control the size (using the pressure module) and the color (touch module) of a digital brush while drawing (moving the pen) with a single hand (Figure 5). One flexible extender module allows using both hands to control interactions. We used this capability to manipulate auxiliary controls with the non-dominant hand while the dominant hand was controlling the brush position and pressure. Adding a vibro-motor and a joystick module transforms VersaPen into a game controller (Figure 6).

Figure 6: VersaPen fosters multimodal interaction and can be used as game controller.

Space-efficient interaction
Graphical applications (e.g. Adobe Photoshop) provide many commands gathered in palettes and toolbars. Graphical widgets mask a part of the screen and round trips between them and the objects of interest slows down interaction [12]. Users can assign a selection of functionalities to VersaPen modules to save screen real-estate. This customization benefits from existing hotkeys shortcuts. We used this capability in a drawing application which provides a novice and an expert mode. In novice mode, turning the wheel to select a command displays a palette on the screen helping the user to choose an action. In expert mode, the user turns the wheel without waiting for the palette to be displayed on the screen. The haptic module or a LED can act as feedback to display the current tool.

Shortcuts and automatic configuration
VersaPen buttons can be used to create shortcuts for activating frequent commands. For instance, we used this capability to change the stacking order of overlapping windows on the screen. When the user toggles a dedicated button, the window under the frontmost window comes to the front (and so on by clicking again if more windows are overlapped). This makes it possible to perform drag and drop operations between windows without releasing the pen. Letting the user set up his own shortcuts helps him to be more efficient when interacting with the system.

Tangible interaction
The modules have different colors (Figure 7), making it easy to distinguish them. When acting as data containers, these modules can encapsulate software settings, predefined behaviors or virtual data. A module can "contain" personalized properties for a given application such as a given font, theme, color, etc. A module without I/O can also serve to configure the pen, by changing the mapping between

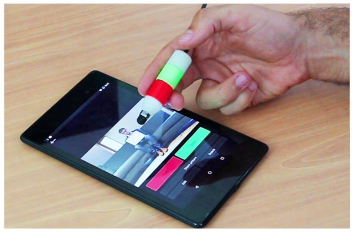

Figure 7: Modules can be used as data containers to share data or settings.

Figure 8: Additional information of a visualization are displayed on a LED module.

Figure 9: VersaPen expanding input capabilities of an existing device.

its buttons, sensors and actuators. The user can then restore these parameters when she uses another computer or changes the configuration of the pen.

Collaboration
Tangible interaction favors collaboration [20, 27]. We took advantage of the capability of *encapsulating* data for exchanging images and videos between devices and users (Figure 7). In a photo gallery application connected to VersaPen, a user can share images by transferring it to the data container he wants. Once exchanged to someone else, the software can display the content stored in the module currently mounted on the pen. The user retrieves the image from the pen to his computing device. A remote server is linking the static unique ID to the files, which are retrieved when needed. Several persons can thus exchange data through physical containers. In addition, a single user can also use the same technique for transferring data from a computing device to another.

Device augmentation
Adding a novel input to a device can increase its performance. VersaPen can serve as an auxiliary input for existing devices and provide additional feedback. For instance, we created a 2D data visualization application taking advantage of the LED and vibrator modules to convey additional information (Figure 8). Conversely, VersaPen can serve as an input device extending the capabilities of another device. For instance, we extended a keyboard by attaching the pen to the keyboard. The pen slider module thus allows scrolling content without moving the hands from the keyboard (Figure 9).

Discussion & Future work
We conducted a preliminary study with seven professionals (illustrators, designers) who use digital pens in their work. They used VersaPen for 20min sessions and were allowed to switch modules and create custom interactions. Our participants foresaw how this device could be used with their own applications and appreciated the fact that "*programming [the device] is simple and compatible with existing keyboard shortcuts*". The modules for file sharing were found particularly relevant: using a physical object while working with co-located colleagues is "*better to track the location of a file when working on the same file*". We seek to further evaluate the usability of our system through a qualitative user study. Although our device inherits its final shape from the modules, various ergonomic aspects can be tested to improve the affordance of the pen while miniaturizing the technology.

VersaPen is a step towards modular and multimodal devices. While our work focuses on pen-based interaction, our device raises the challenge of extending this modular approach to other input/output devices. Rather than developing specific devices to achieve a precise task, VersaPen focuses on the interchangeability and the versatility of its modules. It enables the exploration of different pen interactions by adding or removing modules or by assembling them differently. We expect this capability to inspire novel usages of pen-based and multi-modal interaction.

ACKNOWLEDGMENTS
This research work was supported by the Agence Nationale de la Recherche (ANR 13 CORD 008) and by the EQUIPEX Digiscope ANR-10-EQPX-0026.

REFERENCES

1. Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the working conference on Advanced visual interfaces*. ACM, 154–161. DOI: http://dx.doi.org/10.1145/1385569.1385595

2. Pierre Dragicevic and Jean-Daniel Fekete. 2004. The Input Configurator Toolkit: Towards High Input Adaptability in Interactive Applications. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, 244–247. DOI: http://dx.doi.org/10.1145/989863.989904

3. W Keith Edwards, Scott E Hudson, Joshua Marinacci, Roy Rodenstein, Thomas Rodriguez, and Ian Smith. 1997. Systematic output modification in a 2D user interface toolkit. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. ACM, 151–158. DOI: http://dx.doi.org/10.1145/263407.263537

4. Saul Greenberg and Chester Fitchett. 2001. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, 209–218. DOI: http://dx.doi.org/10.1145/502348.502388

5. Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 861–870. DOI: http://dx.doi.org/10.1145/1124772.1124898

6. Björn Hartmann, Scott R. Klemmer, Michael Bernstein, and Nirav Mehta. 2005. d.tools: Visually Prototyping Physical UIs through Statecharts. In *in Extended Abstracts of UIST 2005*. ACM Press.

7. Ken Hinckley, Michel Pahud, Hrvoje Benko, Pourang Irani, François Guimbretière, Marcel Gavriliu, Xiang 'Anthony' Chen, Fabrice Matulic, William Buxton, and Andrew Wilson. 2014. Sensing Techniques for Tablet+Stylus Interaction. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 605–614. DOI: http://dx.doi.org/10.1145/2642918.2647379

8. 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. DOI: http://dx.doi.org/10.1145/2501988.2501991

9. Alec Jacobson, Daniele Panozzo, Oliver Glauser, Cédric Pradalier, Otmar Hilliges, and Olga Sorkine-Hornung. 2014. Tangible and modular input device for character articulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 82. DOI: http://dx.doi.org/10.1145/2614066.2614072

10. Lynette A Jones and Susan J Lederman. 2006. *Human hand function*. Oxford University Press.

11. Levetron Mech5 Modular Gaming Keyboard. 2011. http://tinyurl.com/leveronMech5. (2011). (Accessed on 06/09/2016).

12. Gordon Kurtenbach and William Buxton. 1991. Issues in Combining Marking and Direct Manipulation Techniques. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology (UIST '91)*. ACM, New York, NY, USA, 137–144. DOI: http://dx.doi.org/10.1145/120782.120797

13. Johnny C Lee, Daniel Avrahami, Scott E Hudson, Jodi Forlizzi, Paul H Dietz, and Darren Leigh. 2004a. The calder toolkit: wired and wireless components for rapidly prototyping interactive devices. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*. ACM, 167–175. DOI: http://dx.doi.org/10.1145/1013115.1013139

14. Johnny C Lee, Paul H Dietz, Darren Leigh, William S Yerazunis, and Scott E Hudson. 2004b. Haptic pen: a tactile feedback stylus for touch screens. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. ACM, 291–294. DOI: http://dx.doi.org/10.1145/1029632.1029682

15. Yang Li, Ken Hinckley, Zhiwei Guan, and James A Landay. 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 461–470. DOI: http://dx.doi.org/10.1145/1054972.1055036

16. Chunyuan Liao, François Guimbretière, and Corinna E Loeckenhoff. 2006. Pen-top feedback for paper-based interfaces. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, 201–210. DOI: http://dx.doi.org/10.1145/1166253.1166285

17. Shenwei Liu and François Guimbretière. 2012. FlexAura: a flexible near-surface range sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 327–330. DOI: http://dx.doi.org/10.1145/2380116.2380158

18. VersaPen open source and open-hardware files. 2017. https://github.com/marcteys/Versapen. (2017).

19. Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. 2004. Pressure widgets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 487–494. DOI: http://dx.doi.org/10.1145/985692.985754

20. Bertrand Schneider, Patrick Jermann, Guillaume Zufferey, and Pierre Dillenbourg. 2011. Benefits of a tangible interface for collaborative learning and interaction. *IEEE Transactions on Learning Technologies* 4, 3 (2011), 222–232. DOI: http://dx.doi.org/10.1109/TLT.2010.36

21. 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*. ACM, 1323–1332. DOI: http://dx.doi.org/10.1145/1978942.1979138

22. Hyunyoung Song, Tovi Grossman, George Fitzmaurice, François Guimbretiere, Azam Khan, Ramtin Attar, and Gordon Kurtenbach. 2009. PenLight: Combining a Mobile Projector and a Digital Pen for Dynamic Visual Overlay. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 143–152. DOI: http://dx.doi.org/10.1145/1518701.1518726

23. Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. 2006. User interface façades: towards fully adaptable user interfaces. ACM, 309–318. DOI:http://dx.doi.org/10.1145/1166253.1166301

24. Yu Suzuki, Kazuo Misue, and Jiro Tanaka. 2007. Stylus Enhancement to Enrich Interaction with Computers. In *Proceedings of the 12th International Conference on Human-computer Interaction: Interaction Platforms and Techniques (HCI'07)*. Springer-Verlag, 133–142. http://dl.acm.org/citation.cfm?id=1757268.1757284

25. Desney S Tan, Brian Meyers, and Mary Czerwinski. 2004. WinCuts: manipulating arbitrary window regions for more effective use of screen space. In *CHI'04 extended abstracts on Human factors in computing systems*. ACM, 1525–1528. DOI: http://dx.doi.org/10.1145/985921.986106

26. Feng Tian, Xiang Ao, Hongan Wang, Vidya Setlur, and Guozhong Dai. 2007. The Tilt Cursor: Enhancing Stimulus-response Compatibility by Providing 3D Orientation Cue of Pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 303–306. DOI: http://dx.doi.org/10.1145/1240624.1240675

27. Brygg Ullmer, Hiroshi Ishii, and Dylan Glas. 1998. mediaBlocks: Physical Containers, Transports, and Controls for Online Media. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, New York, NY, USA, 379–386. DOI: http://dx.doi.org/10.1145/280814.280940

28. Nicolas Villar and Hans Gellersen. 2007. A malleable control structure for softwired user interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*. ACM, 49–56. DOI: http://dx.doi.org/10.1145/1226969.1226980