

# An Interaction System for Watch Computers Using Tactile Guidance and Bidirectional Segmented Strokes

Gábor Blaskó      Steven Feiner

Department of Computer Science, Columbia University  
New York, NY 10027

[gblasko, feiner]@cs.columbia.edu

## Abstract

We introduce an input system that is based on bidirectional strokes that are segmented by tactile landmarks. By giving the user tactile feedback about the length of a stroke during input, we decrease the dependence of the GUI on the visual display. By concatenating separate strokes into multi-strokes, complex commands may be entered, which may encode commands, data content, or both simultaneously. To demonstrate their power, we show how multi-strokes can be used to traverse a menu hierarchy quickly. In addition, we show how inter-landmark segments of the sensor may be used for continuous and discrete parameter entry, resulting in a multifunctional interaction paradigm. We also introduce multi-widgets, which allow the direct control of multiple virtual widgets without the need to change the state of the device or use modifier buttons. This approach to input does not depend on material displayed visually to the user, and, thanks to tactile guidance, may be used by expert users as an eyes-free user interface. We believe that these benefits make this interaction system especially suitable for wearable computer systems that use a head-worn display and wrist-worn watch-style devices.

## Introduction

Wrist-worn devices are one of the most socially acceptable forms for wearable computing. Their main benefits of portability and quick accessibility are a result of their small size. However, their constraints and disadvantages are also due to their small size. Their physical form limits the number of mechanical input devices with which they can be equipped, while their small screen size limits the amount of textual and graphical information they can display. Desktop user interfaces cannot be easily adapted to this computing domain. Alphanumeric user interfaces using typed commands are inappropriate, since there is not enough space on the device to implement a keyboard—not even a chording keyboard—and other character entry methods (such as the stylus-based gesture systems used on PDAs) are quite time consuming and tedious for extended use.

Graphical user interfaces that are dependent on manipulating an on-screen cursor are very versatile for both desktop and PDA platforms. By using the cursor with a multitude of on-screen widgets for application

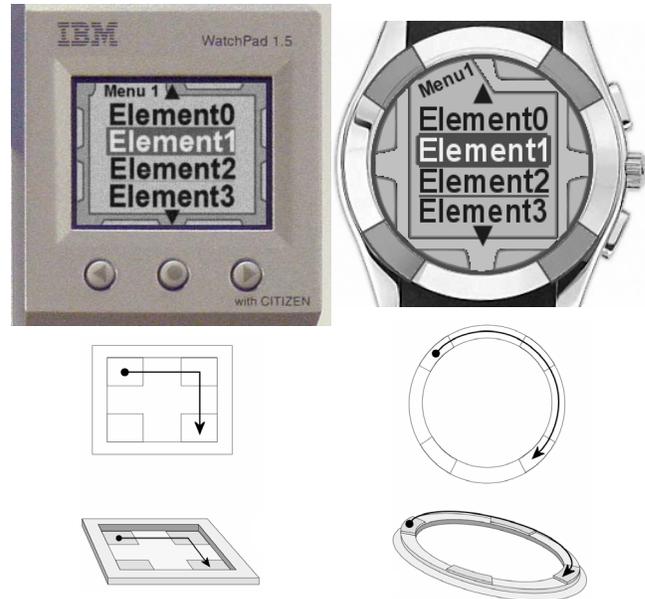


Figure 1. (a) Proposed menu interface design for two different watch faces (shown actual size). (b) Tactile landmarks: inside corners of touchscreen frame, extruded bezel segments.

control and parameter adjustment, a wide range of user interfaces can be built. However, due to the limited screen size of wrist-worn devices, user interfaces that require the navigation of an on-screen cursor, or that are highly dependent on visual feedback, are unsuitable. Furthermore, any user interface that requires a user's visual attention can be problematic in a mobile setting in which the user must attend to the surrounding environment.

To address this problem, we present an input system for display-size-constrained compact wearable devices (proposed designs for which are shown in Figure 1a) that use tactile landmarks and bidirectional segmented strokes (Figure 1b). Then, we show how this system can be used to create feature-rich multifunctional user interfaces.

## Related Work

Many approaches have been presented for system control on wearable platforms. We restrict our brief review to non-speech-based user interfaces. One of the

earliest input systems designed for wearable computing was the waist-worn dial of the VuMan system [1], which was used to highlight and select menu items shown on the screen of a head-worn display. As users became experts, visual search time decreased; however, the mechanical operation of the dial interface still remained time-consuming. This system relied heavily on the visual feedback provided by the display, and no manual shortcuts were provided for experienced users. In contrast, our system decreases the need for visual feedback by moving interface complexity off the screen and onto the device, and by providing a method for experts to access functionality using shortcuts (as described later).

By mounting an input device (or the entire wearable computer) on the wrist, access to the input device and the system's functionality becomes very fast. Glove-based input systems (e.g., [7]) also share this benefit; however, they require the user to wear a glove that covers the fingers with material. Lyons and colleagues [3] show that the Twiddler wearable chording keyboard is also a viable and quick method of menu control. Their KeyMenu complements the text entry capabilities of the Twiddler—which, unfortunately, needs to be enclosed by the hand during use—with menu selection, and they claim that expert users can traverse the menu hierarchy without looking at the head-worn display. The Twiddler has many buttons that are well suited to alphanumeric character entry with chording; however, to adjust application parameters, multiple button presses are needed to change a value or, on one version of the device, a built-in TrackPoint can be used to control an on-screen cursor and on-screen widgets.

The Gesture Pendant [6] is a truly eyes-free input device for executing commands and adjusting parameters; however, it requires the user to remember hand signs and gestures. As presented later, our system may be used by novices who rely on the display. However, after they have committed the menus to memory, expert users can execute shortcut gestures to traverse the menu hierarchy quickly, and adjust application parameters without relying on the display.

## Tactile Landmarks

The physical design of watches has not changed much over the past few decades, even though the range of features they provide has expanded. [4] discusses the similarities in the historical evolution of wristwatches and wearable computers. Traditional mechanical watches, as well as modern computer watches, share common traits that can be exploited in the design of a watch computer interface: they have a face/display and, around it, a bezel/display frame.

The IBM/Citizen WatchPad—the computer watch which we had in mind during the design—has a touchscreen surrounded by a plastic frame and its prototype user interface monitors finger tapping in the four quadrants of the touchscreen [5], simulating

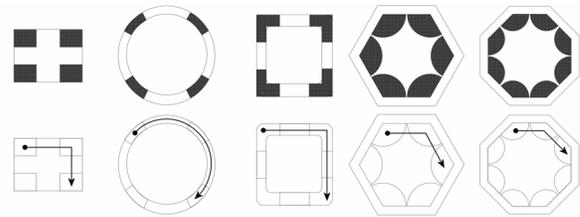


Figure 2. Sensor arrangements for touch-sensitive surfaces with frames and watch bezels with tactile landmarks.



Figure 3. Touchpad based prototypes: (a) Rectangular overlay simulating the touchscreen and frame of the IBM WatchPad. (b) Circular overlay simulating circular watch bezel with extruded landmarks.

buttons. These quadrants are significantly tangible, since the corners of the frame can be easily felt by the finger; therefore, we refer to these corners as *tactile landmarks*. It is easy to envision watches—without touchscreens—of different shapes (e.g., circular, square, hexagonal, or octagonal), which have tangible landmarks on their bezels (e.g., bumps, extrusions, or hollow sections), as illustrated in Figure 2. Because of the small size of the watch, and its location on the wrist, it is easy to home the hand to the device, and the index finger to a given landmark, without looking at the device. For example, the index finger might be positioned quickly by holding the watch between the thumb and the middle finger. Furthermore, since the device is very small, it is easy to execute a gesture by moving the fingertip from one tactile landmark to another, as illustrated in Figure 1(b); for example, from corner to corner along the frame of a touchscreen, or from extrusion to extrusion on the bezel around the watch face.

## Bidirectional Segmented Strokes

Without looking at the device, the user can determine, through their sense of touch alone, the length of a given stroke as measured in landmark-to-landmark length. The tactile landmarks serve as starting, stopping, and intermediate points, as the fingertip of the user moves in a circular gesture on the edge, along the frame of a touchscreen, or on the bezel of a watch.

A circular gesture may begin in either a clockwise (CW) or a counter-clockwise (CCW) direction, and this direction may change upon reaching a certain landmark. For example if there are four corners, two directions (CW/CCW), and strokes may be from one to three landmarks in length, the number of possible strokes that may be executed is 24. This already offers a large number of command-to-stroke mapping possibilities. However, we may further allow the user to execute a stroke in one direction, reach a landmark, then continue the stroke in the other direction—without lifting the finger off the sensor—then after a given length switch directions again, and so on. If we allow such *concatenated multi-strokes* to include two direction

switches, but still restrict the length of the substrokes to three, the number of possible quickly executable stroke possibilities increases to 216 ( $4 \times 2 \times 3 \times 3 \times 3$ ). In addition to mapping all these different multi-strokes to different functions, it is also possible to envision a system in which concatenated substrokes represent not only control/command functions, but also encode preset parameter data values.

This bidirectional segmented gesture system can be implemented on any device that can sense motion/rotation along one dimension that loops around, where this motion is segmented by landmarks.

## Prototyping

For testing these interaction methods, we created simple applications on a desktop computer with an external touchpad, equipped with stencil overlays as shown in Figure 3. The tactile sensation of moving the finger on the pad with the template is a very adequate approximation of the real WatchPad device. For the circular template, the template has thicker portions positioned where the extruding landmarks on the bezel of a circular watch would be. With this plastic overlay in place, the capacitance-based sensor is still sensitive to touch, even though the finger is separated from the touchpad's surface by the stencil's thickness.

## Parameter Adjustment

If the sensor hardware is not only able to differentiate between landmark and non-landmark contact, but can do so with subsegment accuracy—as in our touchpad-based prototype—multiple methods of discrete and continuous parameter adjustment are possible. In their current arrangement, along the inner frame of the touchscreen (or on the circular bezel), the regions between the four landmarks create two horizontal and two vertical linear segments, as shown in Figure 4(a). These *inter-landmark linear segments* can be used to simulate three interaction devices (slider, spinner wheel and spring-loaded wheel) [2]. Additionally, since the landmarks and the segments between them are arranged in a ring, it is also possible to implement a virtual dial by dragging the finger over multiple landmark and non-landmark segments of the sensor in a circular stroke.

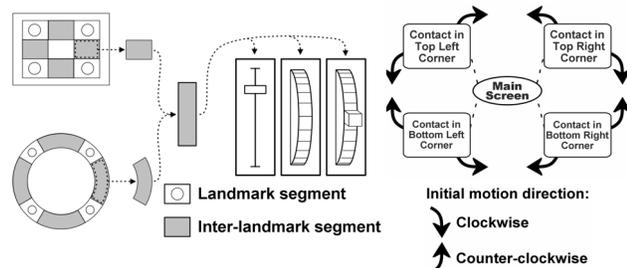


Figure 4. (a) Inter-landmark linear segments may be used to simulate widgets: slider, spinner wheel, and spring-loaded wheel. (b) The widget used may depend on the starting landmark and direction with which the stroke motion begins.

## Multi-Widgets

By introducing tactile landmarks, not only can we segment bilinear gestures and subdivide the circular touch sensor into multiple widgets in the inter-landmark segments, but we can also create a *multi-widget*. In other words, it is possible to implement a multitude of widgets, all of which may be directly accessed, without needing to switch the state of the input device. With a simple extension, we can have as many virtual widgets as there are landmarks, by monitoring the landmark on which the stroke was started. In the case of a rectangular sensor, or a circular sensor with four landmarks, we can use this method to implement four directly accessible independent widgets (e.g., four independently controllable dials).

We may further extend the number of implementable widgets, by not only monitoring the landmark in which the gesture began, but also by differentiating based on the direction in which the gesture was started, as illustrated in Figure 4(b). For example, with four landmarks, it is possible to implement eight directly controllable widgets and adjust eight parameters without needing to change the state of the input device, and without using modifier buttons.

## Menu Control

Menu control on small mobile devices is very often handled with multiple buttons to highlight elements on the screen and a button to activate that selection. Not only is this commonly used method highly dependent on the user looking at the display during the selection process, but with such navigate-and-select systems, even after the user has committed the menu hierarchy to memory, it is not possible to traverse the hierarchy quickly with shortcuts. Similar to the KeyMenu [3], some mobile phones make quick traversal of the hierarchy possible by pressing buttons in a sequence on the numeric keypad. However watch-style devices are too small to equip them with enough adequately sized buttons to make such an approach practical.

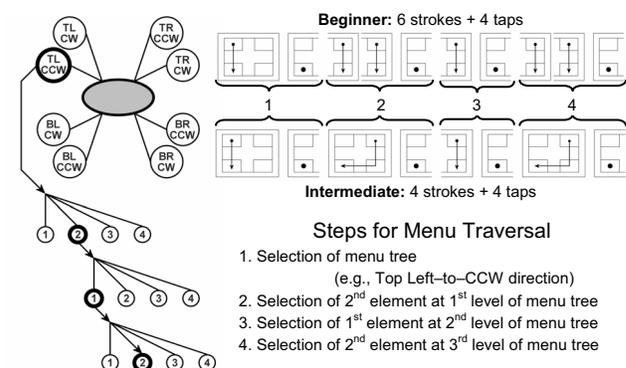


Figure 5. Menu traversal with segmented strokes. First stroke selects menu tree; stroke + tap in lower left. Subsequent strokes traverse hierarchical tree; multiple single-length strokes (beginner) or longer continuous strokes (intermediate) along right edge + tap in lower left corner. Tap in upper left steps back.

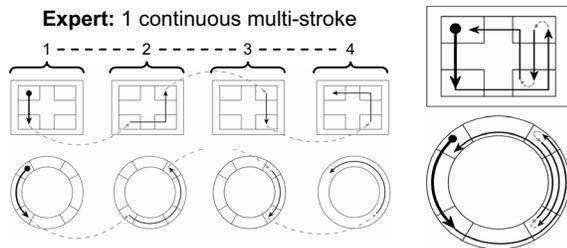


Figure 6. Menu hierarchy traversing shortcut with concatenated strokes.

With our compact input system, based on landmarks and strokes, a menu navigation system can be implemented that accommodates novice, intermediate and expert users as explained below and illustrated in Figures 5 and 6. We differentiate users based on their knowledge of the menu hierarchy and the amount of visual feedback they require during menu traversal.

Novices, who are new to the overall system (including its input mechanism, user interface, menu layout, and system capabilities) may use a slower but more “traditional” traversal method. In the touchscreen implementation, during the execution of strokes and taps the screen is obscured by the finger; therefore, it is necessary to allow the user to view the small screen’s contents and keep track of selections during interaction. In a four-landmark system it may be possible to access up to eight menu trees with a single-length stroke, depending on the starting landmark and starting drag direction, as shown in Figure 5. After executing the stroke, the user confirms the choice of the menu tree by tapping in the lower left landmark. Up/down navigation among the listed menu elements is done with single length up/down strokes between the rightmost two landmarks. Taking a step deeper in the hierarchy is done by tapping on the lower-left landmark, stepping back by tapping on the upper-left landmark.

Novice users, who are familiar with the menu elements—amongst which a choice can be made—at a given level of the menu tree, may use longer strokes extending over multiple landmarks (similarly to setting a numeric parameter with the aforementioned circular dial widget) to highlight a menu item. Selection of the highlighted element is done with a tap on the lower-left landmark.

Expert users, who know the full layout of the menus and are confident in traversing the menu hierarchy without needing to look at the display, may concatenate multiple strokes together into a long, but swiftly executable bidirectional segmented multi-stroke. Menu tree selection as well as tree traversal may be accomplished at once as illustrated in Figure 6, showing the traversal shortcut to the same menu element that is illustrated in Figure 5. The initial landmark and motion direction part of the stroke selects the menu tree, then the selectable menu item’s position in the list is encoded in the length of the subsequent part of the stroke. Selection of an item is done by changing the direction from CW to CCW. If the user is confident in his knowledge of the menu layout, this navigation stroke

may all be executed eyes-free thanks to the fact that the tactile landmarks are felt by the user’s finger during the stroke execution. To assist the expert user with the eyes-free interaction, feedback may be given to the user in the form of haptic vibrations, audible beep signals, or speech synthesis.

## Conclusions and Future Work

We have shown how introducing tactile landmarks and using bidirectional segmented strokes make versatile interaction possible for very small wrist-worn watch-style devices. Since neither the parameter adjustment methods, nor the menu navigation system requires the navigation of an on-screen cursor nor the use of on-screen widgets, the input system is highly suitable for devices that are equipped with small displays and wish to impose the least amount of user distraction during use.

In this work we did not address text entry; however, we note that a modified version of the EdgeWrite system [8], where users are restricted to gestures that are executed only along the edge of the sensor, may be used to extend our input system for character entry.

We are in the process of porting our interaction framework from our desktop computer prototype to the IBM/Citizen WatchPad device and plan to use it as a toolkit to build applications and perform quantitative user studies on interaction performance. We also plan to transfer our approach onto other platforms (augmented- and virtual reality, and in-car computing) where cursor-navigation-based user interfaces are inappropriate due to their high dependence on visual feedback.

## Acknowledgements

Initial conceptual and prototyping work on the presented work was done at IBM T.J. Watson Research Center in Hawthorne, NY. The authors wish to thank the technical assistance and helpful comments of Chandrasekhar Narayanaswami, Mandayam Raghunath, and Stefan Berger. This research is funded in part by the NYS Science and Technology Foundation Center for Advanced Technology, ONR Contract N00014-04-1-0005, and gifts from IBM, Microsoft, and Synaptics.

## References

- [1] L. Bass, C. Kasabach, R. Martin, D. Siewiorek, A. Smailagic, J. Stivoric. The Design of a Wearable Computer. *Proc. CHI '97*.
- [2] G. Blaskó, and S. Feiner, Single-Handed Interaction Techniques for Multiple Pressure-Sensitive Strips, *Ext. Abstr. Proc. CHI '04*.
- [3] K. Lyons, N.J. Patel, T. Starner, KeyMenu: A Keyboard Based Hierarchical Menu, *Proc. ISWC '03*
- [4] Martin, T.L.: Time and Time Again: Parallels in the Development of the Watch and the Wearable Computer, *Proc. ISWC '02*
- [5] C. Narayanaswami, M. Raghunath, Designing a New Form Factor for Wearable Computing, *Proc. Pervasive Computing 2002*.
- [6] T. Starner, J. Auxier, D. Ashbrook, M. Gandy, The Gesture Pendant: A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring, *Proc. ISWC 2000*.
- [7] B.H. Thomas, and W. Piekarski, Glove Based User Interaction Techniques for Augmented Reality in an Outdoor Environment, *Virtual Reality: Research, Development, and Applications*, Vol. 6, No. 3, 2002.
- [8] J.O. Wobbrock, B.A. Myers, J.A. Kembel, EdgeWrite: A Stylus-Based Text Entry Method Designed for High Accuracy and Stability of Motion. *Proc. UIST '03*