# Interacting with mobile and pervasive computer systems

Vassilis Kostakos and Eamonn O'Neill

HCI Group, Department of Computer Science

University of Bath

BA2 7AY, Bath, UK

{vk, eamonn}@cs.bath.ac.uk

## 1. Introduction

One of the most exciting developments in current Human-Computer Interaction research is the shift in focus from computing on the desktop to computing in the wider world. Computational power and the interfaces to that power are moving rapidly into our streets, our vehicles, our buildings and our pockets. The combination of mobile/wearable computing and pervasive/ubiquitous computing is generating great expectations.

We face, however, many challenges in designing human interaction with mobile and pervasive technologies. In particular, the input and output devices and methods of using them that work (at least some of the time!) with deskbound computers are often inappropriate for interaction on the street.

Physically shrinking everything including the input and output devices does not create a usable mobile computer. Instead, we need radical changes in our interaction techniques, comparable to the sea-change in the 1980s from command line to graphical user interfaces. As with that development, the breakthrough we need in interaction techniques will most likely come not from relatively minor adjustments to existing interface hardware and software but from a less predictable mixture of

inspiration and experimentation. For example, Brewster and colleagues have investigated overcoming the limitations of tiny screens on mobile devices by utilising sound and gesture to augment or to replace conventional mobile device interfaces [Brewster 2003; Brewster et al., 2003].

In this paper we present existing and ongoing research within the Human-Computer Interaction group at the University of Bath into the development of novel interaction techniques. With our research we aim to improve the way in which users interact with mobile and pervasive systems. More specifically, we present work in three broad categories of interaction:

- Stroke interaction
- Kinaesthetic interaction
- Text entry

Finally, we describe some of our currently ongoing work as well as planned future work. Before we discuss our research we present some existing work in the areas mentioned above.

## 2. Related work

One of the first applications to implement stroke recognition was Sutherland's sketchpad [Sutherland, 1963]. Strokes-based interaction involves the recognition of pre-defined movement patterns of an input device (typically mouse or touch screen). The idea of mouse strokes as gestures dates back to the 1970s and pie menus [Calahan et al. 1998]. Since then, numerous applications have used similar techniques for allowing users to perform complex actions using an input device. For instance, design programs like [Zhao, 1993] allow users to perform actions on objects by performing mouse or pen strokes on the object. Recently, Web browsing

applications, like Opera[1] and Mozilla Firefox,[2] have incorporated similar capabilities. There are numerous open source projects which involve the development of stroke recognition, including Mozilla, Libstroke,[3] X Scribble,[4] and WayV.[5]

Furthermore, a number of pervasive systems has been developed to date, and most have been designed for, and deployed in, specific physical locations and social situations [Harrison & Dourish, 1996] such as smart homes and living rooms, cars, labs, and offices. As each project was faced with the challenges of its own particular situation, new technologies and interaction techniques were developed, or new ways of combining existing ones. This has led to a number of technological developments, such as tracking via sensing equipment and ultra sound [Hightower & Borriello, 2001], or even motion and object tracking using cameras [Brumitt & Shafer, 2001]. Furthermore, various input and output technologies have been developed including speech, gesture, tactile feedback, and kinaesthetic input [Rekimoto, 2001]. Additionally, environmental parameters have been used with the help of environmental sensors, and toolkits have been developed towards this end [Dey et al., 2001]. Another strand of research has focused on historical data analysis, which is not directly related to pervasive systems but has found practical applications in this area. Finally, many attempts have been made to provide an interface to these systems using tangible interfaces [Rekimoto et al., 2001], or a metaphoric relationship between atoms and bits [Ishii & Ullmer, 1997].

---

[1] See http://www.opera.com
[2] See http://www.mozilla.org/
[3] See http://www.etla.net/libstroke/libstroke.pdf
[4] See http://www.handhelds.org/projects/xscribble.html
[5] http://www.stressbunny.com/wayv/

Some projects have incorporated a wide range of such technologies into one system. For instance, Microsoft's EasyLiving project [Brumitt et al., 2000; Brumitt & Shafer, 2001] utilized smart card readers, video camera tracking, and voice input/output in order to set up a home with a pervasive computing environment. In this environment, users would be able to interact with each other, as well as have casual access to digital devices and resources.

Additionally, text entry on small devices has taken a number of different approaches. One approach is to recognise normal handwriting on the device screen, which will allow users to enter text naturally. The Microsoft PocketPC[6] operating system, for instance, supports this feature. Another approach aiming to minimise the required screen space is the Graffitti7 system used by Palm PDAs, which allows users to enter text one character at a time. Text entry happens on a specific part of the screen, therefore only a small area is required for text entry. An extension of this approach is provided by Boukreev,[8] who has implemented stroke recognition using neural networks. This approach allows for a system that learns from user input, thus becoming more accurate. A third approach is to display a virtual keyboard on-screen, and allow the users to enter text using a stylus.

The work we report in Section 3 presents a technique for recognising input strokes which can be used successfully on devices with very low processing capabilities and very limited space for the input area (i.e. small touch-screens). The technique is based on the user's denoting a direction rather than an actual shape and has the twin benefits of computational efficiency and a very small input area requirement. We

[6] See http://www.pocketpc.com
[7] See http://www.palm.com
[8] See http://www.generation5.org/aisolutions/gestureapp.shtml

have demonstrated the technique with mouse input on a desktop computer, stylus and touch-screen input on a wearable computer and hand movement input using real-time video capture.

Furthermore, the work on kinaesthetic user input we present in Section 4 provides valuable insight into different application domains. The first prototype we present gives real-time feedback to athletes performing weight lifting exercises. Although a number of commercial software packages are available to help athletes with their training programme, most of them are designed to be used after the exercises have been carried out and the data collected. Our system, on the other hand provides instant feedback, both visual and audio, in order to improve the accuracy and timing of the athletes. The second prototype we present is a mixed reality game. We present a pilot study we carried out with three different version of our game, effectively comparing traditional mouse input with abstract, token-based kinaesthetic input and mixed-reality kinaesthetic input.

Finally, the text-entry prototypes we present in Section 5 provide novel ways of entering text in small and embedded devices. An additional design constraint has been the assumption that the users will be attending to other tasks simultaneously (such as driving a car) and that they will only be able to use one hand to carry out text entry. The two prototypes we present address this issue in two distinct ways. The first prototype utilises only 3 hardware buttons, similar to the traditional buttons used in car stereos. Our second prototype makes the best use of a small touch screen and utilises the users' peripheral vision and awareness in order to enhance users' performance. By maximising the size of buttons on the screen, users are given a larger target to aim for, as well as a larger target to notice with their peripheral vision.

## 3. Stroke interaction

In our recent work [Kostakos & O'Neill, 2003] we have developed a technique for recognising input strokes. This technique can be used successfully on a wide range of devices right across this scale. Previously, we have demonstrated the technique with mouse input on a desktop computer, stylus and touch screen input on a wearable computer and hand movement input using real-time video capture. We have termed our technique Directional Stroke Recognition (DSR). As its name implies, it uses strokes as a means of accepting input and commands from the user. In this section we give a brief synopsis of how our technique works and in which situations it can be utilised. A fuller description of the technique is available in [Kostakos & O'Neill, 2003].

The technique is based exclusively on the direction of strokes and discards other characteristics such as the position of a stroke or the relative positions of many strokes. The algorithm is given an ordered set of coordinates (x, y) that describes the path of the performed stroke. These coordinates may be generated in a number of different ways, including conventional pointing devices such as mice and touch screens, but also smart cards, smart rings, and visual object tracking. The coordinates are then translated into a "signature" which is a symbolic representation of the stroke. For instance, an L-shaped stroke could have a signature of "South, East". This signature can then be looked up against a table of pre-defined commands, much as a mouse button double-click has a different result in different contexts. An advantage of using only the direction of the strokes is that a complex stroke may be broken down into a series of simpler strokes that can be performed in situations with very limited input space (Figure 1).

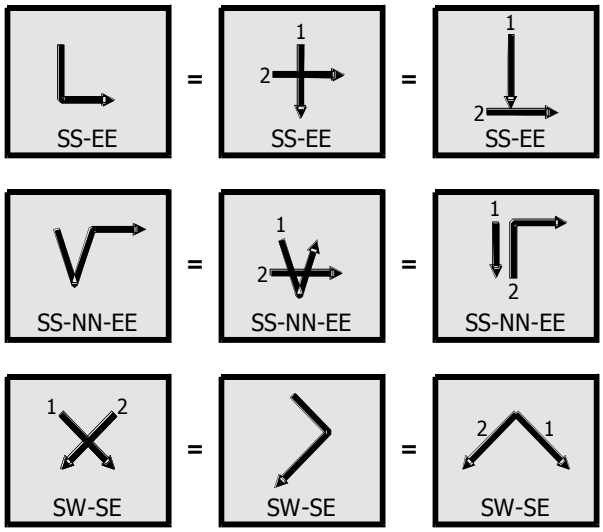


**Figure 1.** The recognition algorithm allows a signature to be accessed via different strokes.

The flexibility of our method allows switching between input devices and methods with no need to learn a new interaction technique. For example someone may at one moment wish to interact with her PDA using a common set of gestures and in the next moment move seamlessly to interacting with a wall display using the same set of gestures. At one moment the PDA provides the interaction area on which the gestures are made using a stylus; in the next moment the PDA itself becomes the "stylus" as is it waved in the air during the interaction with the wall display. Any object or device that can provide a meaningful way of generating coordinates and directions can provide input to the gesture recognition algorithm (Figure 2).

Some important characteristics of this technique include the ability for users to choose the scale and nature of the interaction space they create [Kostakos, 2005; Kostakos & O'Neill, 2005], thus influencing the privacy of their interaction and others' awareness of it. In addition, the physical manifestation of our interaction technique can be tailored according to the situation's requirements. As a result, the technique also allows for easy access, literally just walking up to a system and using it, with no need

for special equipment on the part of the users. This makes the technique very suitable for use in domains such as the hospital A&E department's waiting area.

The Directional Stroke Recognition technique is flexible enough to accommodate a range of technologies (and their physical forms) yet provide the same functionality wherever used. Thus, issues concerning physical form may be addressed independently. In contrast, standard GUI based interaction techniques are closely tied to physical form: mouse, keyboard and monitor. The technique we have described goes a long way towards the separation of the physical form and interaction technique.

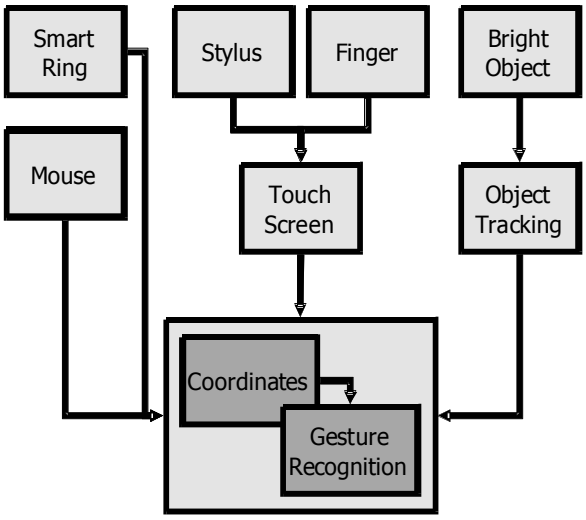


**Figure 2.** Using various techniques with the stroke recognition engine.

As a proof of principle, we implemented a real-time object tracking technique that we then used along with our stroke recognition algorithm as an input technique. For our prototype we implemented an algorithm that performs real-time object tracking on live input from a web camera. The user can select a specific object by sampling its colour, and the algorithm tracks this object in order to generate a series of coordinates that describe the position of the object on the screen, or to be precise, the position of the object relative to the camera's view. We then pass these generated coordinates to

our stroke recognition algorithm, which proceeds with the recognition of the strokes. Due to the characteristics of our stroke recognition method, the coordinates may be supplied at any rate. So long as this rate is kept steady, the stroke recognition is very successful. Thus, despite the fact that our object tracking algorithm is not optimal, it still provides us with a useful prototype.

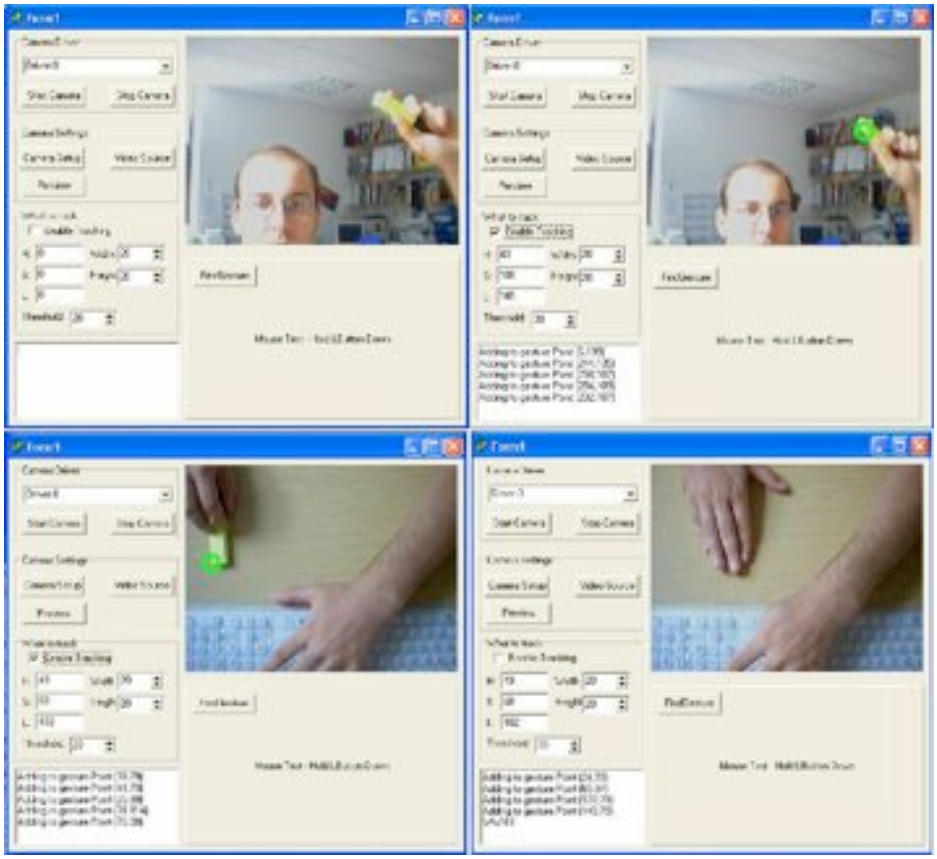

**Figure 3.** Our prototype system for object tracking used with DSR. An control object is identified by clicking on it (top left), and then this object is tracked across the image to generate coordinates (top right). The same object can be tracked in different setups (bottom left). By obscuring the object (bottom right) the stroke recognition algorithm is initiated.

### *3.1 Experimental evaluation*

Our concerns to test the usability of interaction techniques in the absence of visual displays led us to develop a prototype system for providing information to A&E patients through a combination of gesture input and audio output. We used our DSR

technique for the gesture input and speech synthesis for the audio output. We ran an experimental evaluation of this prototype system. The main question addressed by the evaluation was: if we move away from the standard desktop GUI paradigm and its focus on the visual display, do we decrease usability by losing the major benefit that the GUI brought, i.e. being able to see the currently available functionality and how to invoke it?

The experiment itself (screenshots shown in Figure 3) is extensively reported in [O'Neill et al., to be published]. The results of our evaluation may be interpreted as good news for those developers of multimodal interaction who want to mitigate our reliance on the increasingly unsuitable visual displays of small mobile and wearable devices and ubiquitous systems. We found no significant evidence that usability suffered in the absence of one of the major benefits of the GUI paradigm: a visual display of available services and how to access them. However, we must sound a note of caution. Our study suggests that *with particular constraints*, the effects of losing the cognitive support provided by a standard GUI visual display are mitigated. These constraints include having a small set of available functions, a small set of simple input gestures in a memorable pattern (e.g. the points of the compass), a tightly constrained user context and semantically very distinct functions.

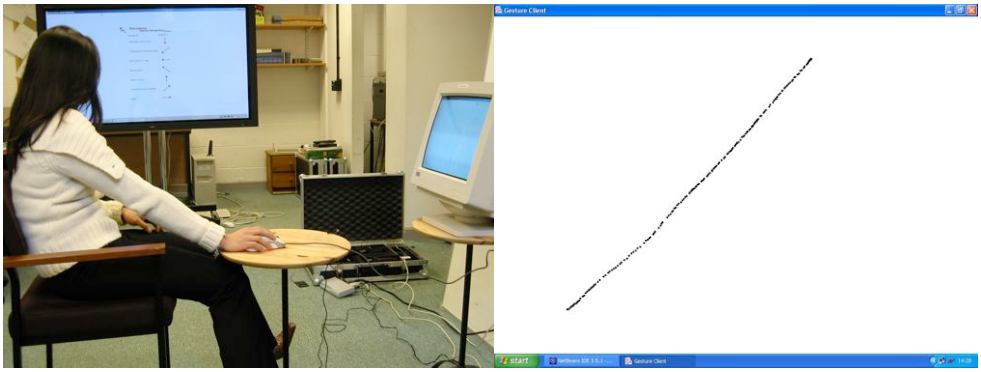

**Figure 4.** Our experimental setup shown on the left, and a sample stroke as entered by a user shown on the right.

Our initial concern remains for the development of non-visual interaction techniques for *general* use in a mobile and pervasive computing world. Our DSR technique for gestural input can handle arbitrarily complex gestures comprised of multiple strokes. There is no requirement for it to be confined to simple single strokes to compass points. Its potential for much richer syntax (similar to a type of alphabet) coincides with the requirement for much richer semantics in general purpose mobile devices.

## 4. Kinaesthetic interaction

Another focus of our research is on developing interaction techniques that utilise implicit user input. More specifically, the prototypes we describe here utilise kinaesthetic user input as a means of interaction. The two prototypes were developed by undergraduate students at the University of Bath and utilise motion tracking technology (XSens MT9 XBus system[9] with Bluetooth) to sense user movements. The first prototype we describe is a training assistant for weight lifting and provides real-time feedback to athletes about their posture and timing. The second prototype described here is a game application which turns a Tablet PC into a mixed-reality maze game in which players must navigate a virtual ball through a trapped maze by means of tilting the Tablet PC.

### *4.1 Weight lifting trainer*

For our first prototype we utilised our motion sensors to build an interactive weight lifting trainer application. Our system is designed to be used by athletes whilst they are actually performing an exercise. The system gives feedback as to how well the

[9] See http://www.xsens.com

exercise is being performed (i.e. if the user has the correct posture and timing). The prototype system is shown in Figure 5.

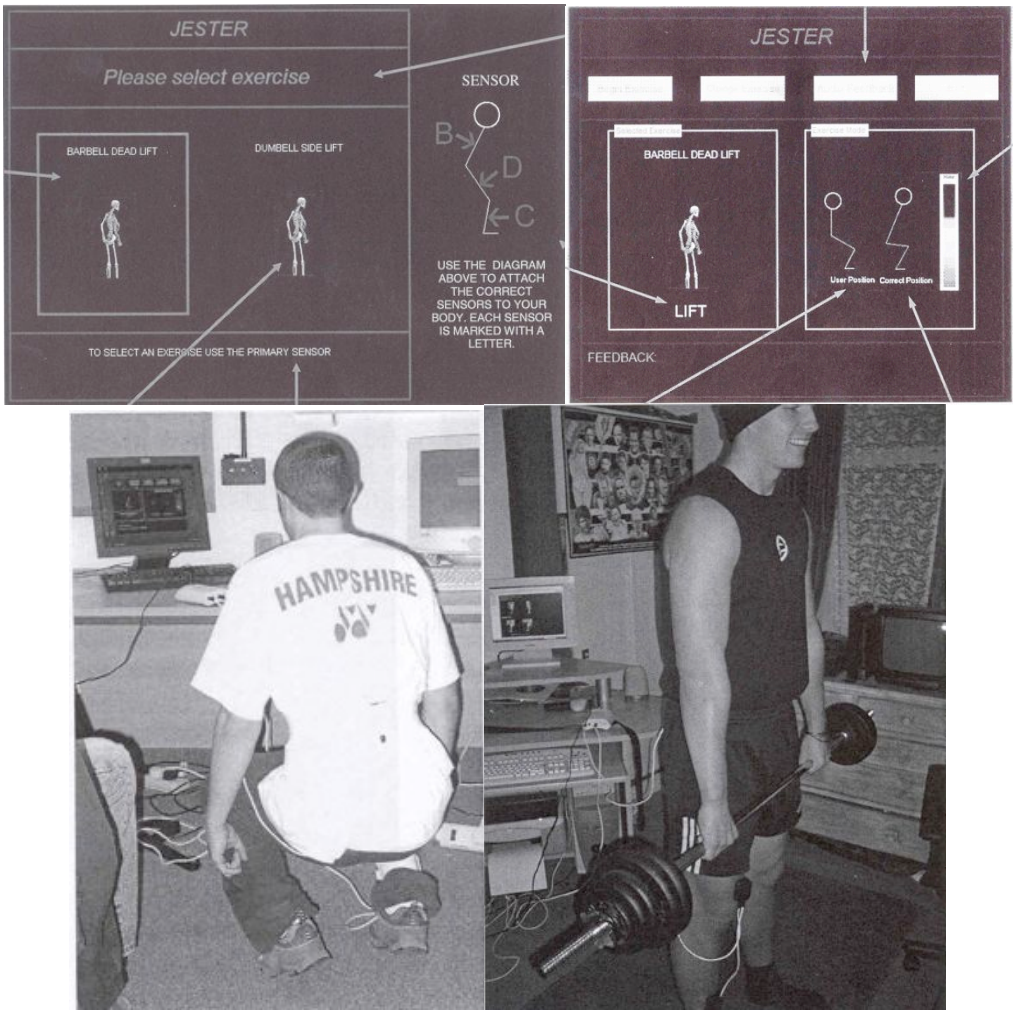


**Figure 5.** The weight lifting trainer prototype. The two images at the top show screenshots of the system. The two images below were taken during our evaluation session.

To use the system, users need to attach the motion sensors to specific parts of the body. The system itself provided guidance on how to do this (top left image in Figure 5). The sensors we used are self-powered and communicate via Bluetooth with a laptop or desktop computer. Therefore, the athlete only has some wiring from each individual sensor to a hub. The hub is placed on the athlete's lower back. This allows users for complete freedom of movement in relation to the computer.

Once the user selects an exercise to be performed, the system loads the hard-coded set of data for the "correct" way of carrying out the exercise. This data was produced by recording a professional athlete carrying out the exercise. The skeleton image on the left provided indications for the main stages of an exercise (such as "Lift", "Hold", "Drop"). The right stick-man diagram (top right image in Figure 5) demonstrates the correct posture and timing for performing the exercise, whilst the stick-man to its left represents the user's actual position. There is also a bar meter on the right which describes the degree of match between optimal and actual position and timing. All these diagrams were updated in real-time and in reaction to user movement. Furthermore, the system provided speech feedback with predetermined cues in order to help the users with the exercise.

To evaluate this prototype we carried out an initial cooperative evaluation [Wright & Monk, 1991] with 5 participants (bottom left and bottom right in Figure 5). Our evaluation revealed that users found it difficult to strap on the sensors, due to the ineffective strapping mechanism we provided. Additionally, we discovered that the sensors didn't always stay in exactly the same positions. Both of these problems can be addressed by providing a more secure strapping mechanism and smaller motion sensors. These problems, however, caused some users to believe that the system was not functioning properly. The users thought that the bar meter feedback was useful and easy to understand. Some of the users found that the skeleton didn't help them. Finally, some users found the voice annoying, while others found that the voice helped them to keep up with the exercise. Most users, however, agreed that more motivational comments (such as the comments that a real life trainer makes) would have been appropriate.

### *4.2 Tilt the maze*

With this prototype we explored the use of motion sensors in a mixed-reality game of tilt the maze. Utilising motion sensors we build three different versions the game. The objective was to navigate a ball through a maze by tilting the maze in different directions. This tilt was achieved though the use of:

- A mouse connected to a typical desktop PC. The maze was displayed on a typical desktop monitor.
- A lightweight board fitted with motion sensors. The maze was displayed on a large plasma screen.
- A Tablet PC fitted with motion sensors. The maze was displayed on the Tablet PC itself so that tilting the tablet would appear to be tilting the virtual maze itself.

We carried out a pilot study to compare performance and user preference for all three conditions. During this study we collected qualitative data in the form of questionnaires, as well as quantitative date by recording the number of aborts, errors and time to completion. The three experimental conditions are shown in Figure 6.

**Figure 6.** At the top we see the system being used by means of a paper cardboard acting as a control token. At the bottom left we see the condition with the PC and mouse, and at the bottom right we see the condition with a Tablet PC acting both as a screen and a control token.

Each participant was given the chance to try all systems. The order in which each participant tried each of the systems was determined at random. The interaction technique of using motion sensors to move the board was well received by the participants. This was not only shown in the high numbers of participants which 'preferred' the Tablet PC (78%) but also in the very low number of participants who 'preferred' the standard and most commonly used interaction technique of a mouse (3%). This was also comparatively low to the percentage of people who preferred the Plasma Screen (19%), which also used the motion sensors to tilt the board.

The questionnaires showed that participants found the Tablet PC the least difficult, then the Plasma Screen and found the mouse the hardest way of interacting with the system. Using Tablet PC participants took on average 79 seconds using than the Plasma Screen 91 seconds, and with the Mouse 154 seconds. The mouse on average took almost twice as long as the Tablet PC to complete. The number of aborted games was also least on the Tablet PC (1) and most by the Mouse (9), while the Plasma screen had 4 aborts. It should be noted however that the average number of errors made was greatest on the Mouse (160), but the Plasma screen seemed to produce on average less errors (94) than the Tablet PC (104), although the difference was relatively small.

These results show that on average the participants liked using the Tablet PC the most, made slightly more errors on it than on the Plasma screen but finished in a faster time. The lab experiment has given some confirmation that the novel interaction technique of using motion detectors to manipulate a maze (and hopefully an indication that similar tasks will behave in a similar manner) was received well and that it outclassed the most common interaction technique of using a mouse.

## 5. Text entry

In our earlier work on gestural interaction we noted that the DSR may be utilised to communicate complex strokes, essentially acting as a kind of alphabet with 8 distinct tokens. Although this allows for complex interactions, it does not address the perennial issue of text entry in mobile and pervasive systems. In this section we describe two prototype systems for text entry in embedded devices. These prototypes were developed by undergraduate students at the University of Bath. The first prototype makes use of two keys and a dial to enter text. The second prototype allows for text entry on a small size touch screen. Both prototypes address the entry of text

on embedded devices. The application domain for both prototypes were designed is embedded digital music players. We designed these systems so that users can interact with them using only one had and situations were the users have to attand to other tasks simultaneously (such as driving a car).

### *5.1 Key and dial text entry*

The first prototype we present allows for text entry on an embedded digital music player. We envision this system to be used in cars, an application domain in which traditionally all interaction takes place via a minimum number of hardware keys. One of the main purposes of this approach is to minimise the cognitive load on drivers who are concurrently interacting with the steering controls as well as the music player.

In Figure 7 we can see our first prototype. The top of the figure is a mock-up of the actual hardware façade that would be visible in a car. The main aspects of this façade we focus on is the circular dial on the left, the left/right arrows below it, as well as the grey area which denotes a simple LCD screen. At the bottom of Figure 7 we see the screenshots of our functional prototype's screen. Bottom left depicts normal operation, while bottom right depicts edit mode.

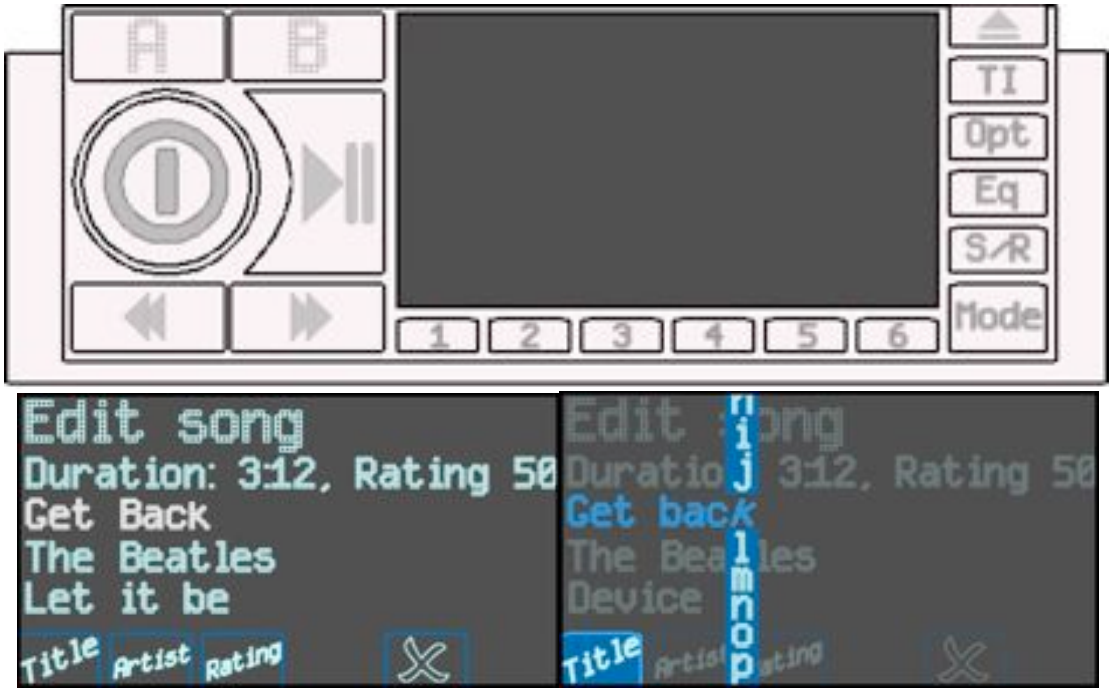


**Figure 7.** Our mock-up prototype shown on top. The circular dial on the left is used to select a letter from the alphabet. The left/right arrows below the dial are used to shift the edited character in the word.

When the user enters text edit mode, the system greys everything on the screen except the current line of text being edited. In Figure 7, the text being edited is the title of a song called "Get back". Text entry with this system takes place as follows. The user uses the left/right buttons to select the character they wish to change. The character to be changed is placed in the middle of a column of characters making up the alphabet. For example, in the bottom right part of Figure 7 we can see that character 'k' is about to be changed. To actually change the character, the user turns the dial clockwise or anti-clockwise, which has the effect of scrolling up and down the column of characters. When the user has selected the desired character, they can move on to the next character in the word using the left/right buttons.

We have carried out an initial set of cooperative evaluation sessions with 10 participants. The evaluation itself was carried out on the whole spectrum of the prototype's functionality, which included playing music tracks from a database,

adding/deleting tracks and tuning to radio stations. We received very positive feedback in relation to text entry interaction. Some users were able to pick up the interaction technique without any prompt or instructions from us. A few users, on the other hand asked for instruction on how text entry worked. Generally, however, towards the end of the evaluation sessions all users felt happy and comfortable with entering text using the dial and keys.

### *5.2 Text entry on small touch screens*

The second prototype we have developed and evaluated utilises small-sized touch screens for text entry. Once more, this prototype was developed for text entry in environments were the users are distracted or must be focused on various tasks. For this prototype we wished to take advantage of user's peripheral vision and awareness. For this reason, the prototype utilises the whole of the touch screen for text entry. This enables users to aim for bigger targets on the screen while entering text. Furthermore, this prototype was designed to allow for single-handed interaction. The prototype is shown in Figure 8.

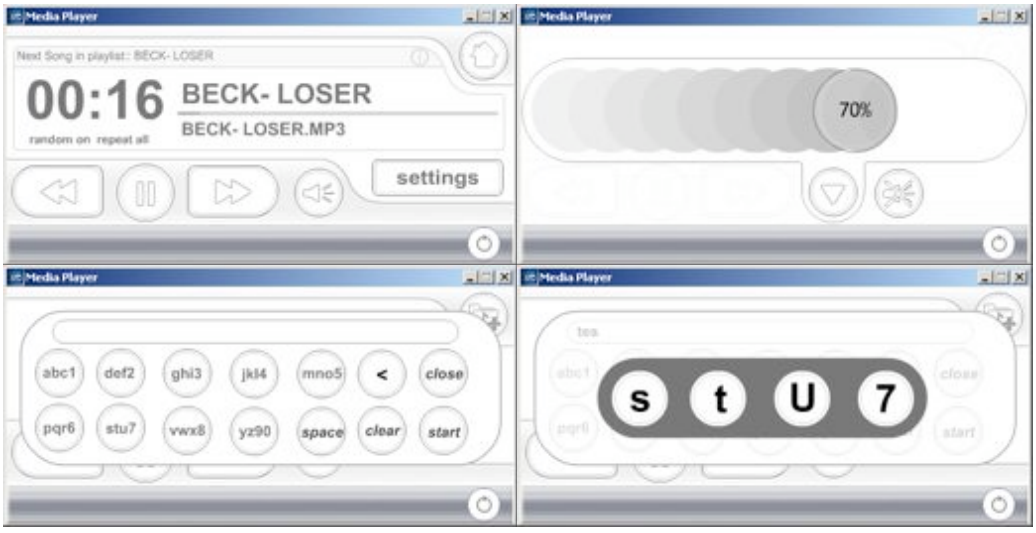

**Figure 8.** The prototype's main playing screen is shown in the top left. The volume control screen is shown in the top right. The keyboard screen is shown in the bottom left. Once a key is pressed, the four options come up, as shown in the bottom right.

To enable text entry, the system brings up a keyboard screen, shown in the bottom left in Figure 8. This design closely resembles the layout of text used in traditional phones and mobile phones. At this stage, the background functionality of the system has been disabled. When a user presses a button, a new screen is displayed with 4 options from which the user may choose (bottom right in Figure 8). Notice that the user can only enter text, and no other functionality is accessible. This decision was made in order to accommodate for clumsy targeting resulting in the use of a finger, instead of a stylus, to touch the screen.

We evaluated this prototype by carrying out 6 cooperative evaluation sessions. The initial phase of our evaluation was used to gauge the skill level of the user. The co-operative evaluation was then carried out following a brief introduction to the system. During the evaluation breakdowns and critical incidents were noted either via user prompting or by the evaluator noticing user problems. After the evaluation was complete the user was queried on these breakdowns and instances. A brief qualitative questionnaire was given followed by a longer quantitative questionnaire. These gave us both feedback on user opinions, and suggestions about the overall system.

According to our questionnaire data, users found the text entry functionality quite intuitive. Specifically, on a scale of 0 (very difficult) to 9 (very easy), the text entry functionality was rated 8 on average. Based on the qualitative data collected, we believe that the design employed, that of the simulation of a mobile phone keyboard, worked well and was highly intuitive.

## 6. Ongoing and future work

In our research we are currently exploring new ways of interacting with big and small displays. One of the systems we are currently developing is used for exploring high-resolution images on small displays. This system, shown in Figure 7, provides an overview of the image, and then proceeds to zoom into hot-spots, or areas of interest within the image. The feedback area at the top provides information about the progress of the task (progress bar), the current zoom level (circle), and the location of the next hot-spot to be shown (arrow).

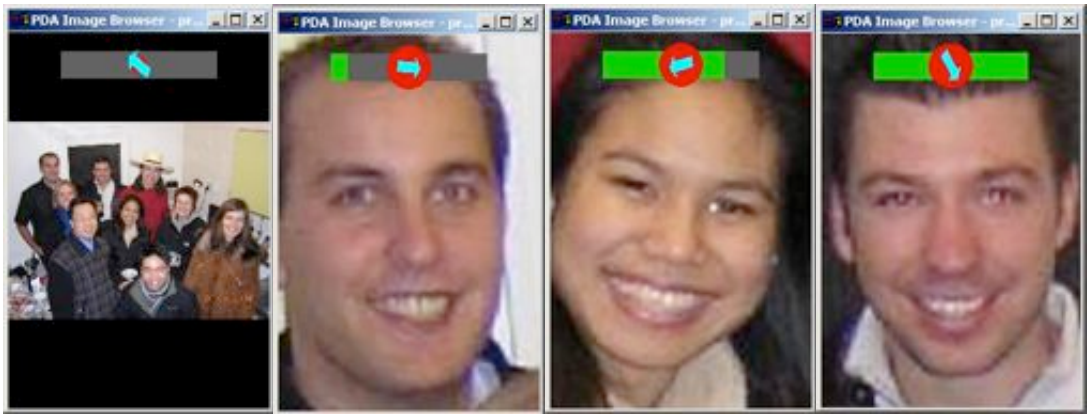


**Figure 9.** Our image explorer provides an overview of the image to be explored, and the proceeds to zoom into specific areas of interest within the image.

Another research strand we are currently exploring is the use of both large screen and small screen devices in situations were public and private information is to be shared between groups of people. We are exploring the use of small-screen devices as a private portal, and are developing interaction techniques for controlling where and how public and private information is displayed. Our overall aim is to develop interaction techniques that match our theoretical work on the design of pervasive systems [Kostakos, 2005], the presentation and delivery of public and private information [O'Neill et al., 2004], and making use of physical and interaction spaces for delivering such information [Kostakos & O'Neill, 2005].

## 7. Acknowledgements

We wish to thank Andy Warr and Manatsawee (Jay) Kaenampornpan for their contribution and assistance. We are also very grateful to Adrian Merville-Tugg, Avri Bilovich, Christos Bechlivanidis, Colin Paxton, David Taylor, Gareth Roberts, Hemal Patel, Ian Saunders, Ieuan Pearn, James Wynn, Jason Lovell, John Quesnell, Jon Bailyes, Jonathan Mason, Ka Tang, Mark Bryant, Mary Estall, Nick Brunwin, Nick Wells, Richard Pearcy and Simon Jones for developing of the prototypes presented in Sections 4 and 5. Special thanks to John Collomosse for his assistance in the development of the image explorer application.

## 8. References

Brewster, S. A. (2002). Overcoming the Lack of Screen Space on Mobile Computers, Personal and Ubiquitous Computing 6(3), p. 188–205.

Brewster, S. A., Lumsden, J., Bell, M., Hall, M. & Tasker, S. (2003). Multi-modal "Eyes Free" Interaction Techniques for Wearable Devices, in G. Cockton & P. Korhonen (eds.), Proceedings of CHI'03 Conference on Human Factors in Computing Systems, CHI Letters 5(1), ACM Press, p. 473–80.

Brumitt, B. and Shafer, S. (2001). Better Living Through Geometry. Personal and Ubiquitous Computing, 2001. 5(1): p. 42-45.

Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. (2000). EasyLiving: Technologies for Intelligent Environments. Lecture Notes in Computer Science, 2000(1927): p. 12-29.

Callahan, J., Hopkins, D., Weiser, M. & Shneiderman, B. (1998). An Empirical Comparison of Pie vs. Linear Menus, in M. E. Atwood, C.-M. Karat, A. Lund, J.

Coutaz & J. Karat (eds.), Proceedings of the CHI'98 Conference on Human Factors in Computing Systems, ACM Press, p. 95–100.

Dey, A.K., Abowd, G.D., and Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human Computer Interaction, 2001. 16(2/4): p. 97-166.

Harrison, S. and Dourish, P. (1996). Re-place-ing space: the roles of place and space in collaborative systems. In Proceedings of the 1996 ACM conference on Computer supported cooperative work. 1996: ACM Press. p. 67-76.

Hightower, J. and Borriello, G. (2001). Location Systems for Ubiquitous Computing. Computer, 2001. 34(8): p. 57-66.

Ishii, H. and Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In Human factors in computing systems. 1997.

Kostakos, V. (2005). Ph.D. Dissertation: A Design Framework for Pervasive Computing. Department of Computer Science, University of Bath. Technical Report CSBU-2005-02. Technical Report Series ISSN 1740-9497.

Kostakos, V. and O'Neill, E. (2003). A Directional Stroke Recognition Technique for Mobile Interaction in a Pervasive Computing World, People and Computers XVII, Proceedings of HCI 2003: Designing for Society, Bath, Sept 2003, p. 197-206.

Kostakos, V. and O'Neill, E. (2005). A space oriented approach to designing pervasive systems. In proceedings of the 3rd Uk-UbiNet Workshop, 9-11 February 2005, University of Bath, UK.

O'Neill, E., Kaenampornpan, M., Kostakos, V., Warr, A. and Woodgate, D. “Can we do without GUIs? Gesture and speech interaction with a patient information system.” *Personal and Ubiquitous Computing*, to be published.

O'Neill, E., Woodgate, D., and Kostakos, V. (2004) Easing the wait in the Emergency Room: building a theory of public information systems. Proceedings of the ACM Desiging Interactive Systems 2004 , Boston, MA., August 2004, p. 17-25.

Rekimoto, J. (2001). GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. In Wearable computers. 2001. Zurich, Switzerland: IEEE. p. 21-30.

Rekimoto, J., Ullmer, B., and Oba, H. (2001). DataTiles: a modular platform for mixed physical and graphical interactions. In Proceedings of the SIGCHI conference on Human factors in computing systems. 2001: ACM Press. p. 269-276.

Sutherland, I. (1963). Sketchpad: A Man–Machine Graphical Communication System, in proceedings of the Spring Joint Computer Conference, IFIP, p 329–46.

Wright, P.C. and Monk A.F. (1991). A cost-effective evaluation method for use by designers, International Journal of Man-machine Studies, 35, p. 891-912.

Zhao, R. (1993). Incremental Recognition in Gesture-based and Syntax-directed Diagram Editors, in S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel & T. White (eds.), Proceedings of INTERCHI’93, ACM Press/IOS Press, p. 95–100.

## 9. Key terms

Kinaestetic interaction: interacting with a computer via body movement – i.e. hand, arm, leg movement.

Strokes: straight lines of movement.

Stroke interaction: interacting with a computer using strokes. To perform the strokes a user needs a token object, such as the mouse, their hand, or a tennis ball.

Gesture interaction: interacting with a computer using movements (not restricted to strokes) performed by a token object.

Text entry: entering alphanumeric characters into a computer system.

Pilot study: an initial, small-scale evaluation of a system.

Cooperative evaluation: the process by which a computer system developer observes users using the system. The purpose of this process is for the developer to identify problems with the system.