Mobile Phones as Pointing Devices

Rafael Ballagas
Media Computing Group
RWTH Aachen University
Aachen, Germany
ballagas@cs.rwth-aachen.de

Michael Rohs
Institute for Pervasive Computing
ETH Zurich
Zurich, Switzerland
rohs@inf.ethz.ch

Jennifer G. Sheridan
Ubiquitous Computing Group
Lancaster University
Lancaster, England
sheridaj@comp.lancs.ac.uk

ABSTRACT

This article outlines two techniques that allow the mobile phone to be used as a pointing device for public terminals and large public displays. Our research has produced two complimentary camera-based input techniques. We outline the details of the interaction techniques and identify further areas of exploration.

1. INTRODUCTION

Mobile phones are the first truly pervasive computer. They help us keep in touch with our loved ones, and help manage everyday lives with address book and calendar functionality; consequently, the mobile phone is always carried with us. Technological trends are packing more and more computational, communication, and sensing resources into the small, convenient form factor of today's smart phone. We leverage these trends to provide a ubiquitous pointing device using the mobile phone.

We have developed two complementary camera-based interaction techniques called *point & shoot* based on optical sensing and *sweep* based on optical-flow detection. Both techniques can be used for pointing, for example, to control a cursor on a large display. *Point & shoot* is used for absolute cursor positioning, while *sweep* realizes relative cursor movement. The techniques could be combined in the same task, depending for example on the distance to be covered or the pointing precision required.

2. RELATED WORK

Other systems have used personal devices for direct manipulation interactions with large displays and information terminals. The Remote Commander enables individuals to use a PDA to control the mouse and keyboard on a remote display using the PDA's touch sensitive display for mouse input and graffiti for text entry [6]. The C-Blink [5] system allows users to control a cursor on a large display using a mobile phone with a colored screen. The user runs a program on the phone that rapidly changes the hue of the phone screen and waves the phone screen in front of a camera mounted above a large display. The camera tracks the relative position of this signal to control the cursor on the display. Slifverberg et al. [8] have studied the use of the handheld joystick, increasingly more common on today's mobile phones, as a pointing device for information terminals. Madhavapeddy et al. [4] introduce techniques that use visual tags known as SpotCodes. Interaction involves using a phonecam to scan tags or to manipulate tagged GUI widgets. The main distinction of our design is that it can be used to select any arbitrary pixel, where Madhavapeddy's work only allows the user to select or manipulate tagged objects.

3. INTERACTION TECHNIQUES

3.1 Point & Shoot

The *point & shoot* interaction technique is illustrated in Figure 2. The user aims the mobile phone to the target on the large display. The contents of the large display appear on the phone screen, which acts as a view finder and is continously updated as the device moves. Aiming is facilitated by a cross-hair cursor in the center of the phone screen including a magnification of the area around the center. The magnified part is shown in the upper right corner of the phone screen. In *point & shoot* interaction, the user's locus of attention is on the phone screen. The cursor on the phone screen is active and the large display cursor remains inactive. *Point & shoot* is triggered by horizontally pushing and releasing the joystick button. As soon as the user "shoots," a grid of tags is shortly superimposed (approx. 0.5 seconds) over the large display contents, as can be seen in the middle part of Figure 2. The coordinate systems of the recognized elements are then used to compute the precise point on the large display that was targeted. Finally, a selection is issued on the large display at the target point.

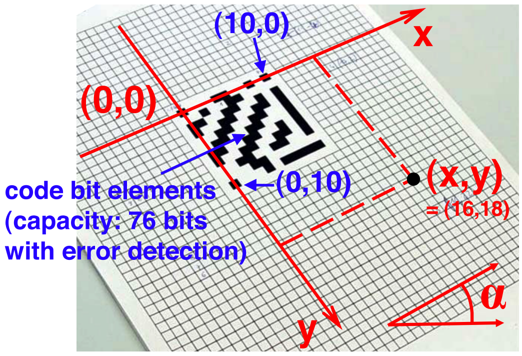

Figure 1: Each Visual Code has its own local coordinate system that is invariant to perspective distortion.

The *point & shoot* technique utilizes Visual Codes [7] to determine absolute positioning information. Phones can detect codes from any arbitrary orientation. The codes themselves can store up to 76 bits of information, which can serve as a digital handle for a physical object, an operation on the phone, a specific service, or an item on the large display. In order to provide richer interaction possibilities, the rotation and tilting values can serve as additional input parameters. Depending on the posture of the phone relative to the code, different information aspects or services can thus be invoked. Each

Figure 2: ***Point & shoot*** **interaction: The left screenshot shows the phone screen as the user is aiming at the highlighted target. The cursor on the phone is active and the locus of attention and the cursor on the display is inactive. Pushing the joystick to the left indicates a selection at the location of the cursor on the phone display. Then the visual code grid is briefly displayed for computing the target coordinates as shown in the middle screenshot. The grid disappears again and a mouse selection occurs in the target region.**

code defines its own local coordinate system (shown in Figure 1), which is invariant to perspective distortion arising from the inherent mobility of the phone camera. This enables the mapping of arbitrary coordinates in the camera image to corresponding coordinates in the code plane (in our case, the large display). In the *point & shoot* technique, we use this feature to determine the precise absolute pixel coordinates of the point on the large display that corresponds to the user's cursor on the local phonecam display.

3.2 Sweep

The *sweep* technique utilizes optical-flow image processing, which involves rapidly sampling successive images from a camera phone and sequentially comparing them to determine relative motion in the (x, y, θ) dimensions. This enables the camera to be used as a three degrees of freedom (DOF) input device. No Visual Code has to be present in the view of the camera, since the relative movement detection solely relies on the comparison of camera images. In our implementation, optical-flow processing is performed directly on the phone rather than on the computer driving the display. One advantage of this strategy is user scalability; the interaction technique easily scales to a high number of users. A disadvantage however, is the high latency with current hardware (about 200 ms) that occurs when calculating the (x, y, θ) changes from successive images. Studies have shown that system lag has a multiplicative effect on Fitts' index of difficulty [3]. Yet mobile computing trends indicate that in the not too distant future mobile phones will have the processing power necessary to create a fluid interaction experience.

To invoke the *sweep* function, users vertically push and hold the joystick button, which acts as a clutch, to indicate to the system that they are actively controlling the cursor, and then they wave the phone in the air to control the (x, y, θ) input. Users can release the clutch button to reposition their arm, which is similar to the way a mouse can be lifted to be repositioned on a desktop surface. This means that the camera need not be pointed directly at the display but can be pointed at the floor to allow users a more comfortable arm posture. In the *sweep* mode, the user can ignore the display on the phone and focus attention on the large display to observe the cursor movement.

3.3 Implementation

To enable selection, dragging, and rotation, the *point & shoot* and *sweep* techniques are mapped to the phone's joystick button as shown in Figure 4. Absolute movement (*point & shoot*) is invoked

Figure 3: The sweep technique can be used to control a cursor by waving the phone in the air.

by pushing the joystick in a horizontal direction. Pressing it to the left and releasing it again triggers absolute movement of the cursor only, whereas pressing and releasing it to the right also drags the object currently located beneath the cursor to the new cursor position. Relative movement (*sweep*) is invoked by pushing the joystick in a vertical direction. Holding it upwards invokes relative cursor movement only, whereas holding it downwards additionally drags the current object. Relative dragging includes rotation of the objects, which is done by rotating the phone around the z-axis. Absolute dragging includes rotation as well. Pressing the joystick key inwards (along the z-axis into the phone) is used for explicit selection. Mapping the interaction techniques to the joystick button in this way preserves simple one-handed operation and does not impinge on dexterity as the user is not required to reposition his or her finger to different buttons.

As shown in the classification in Figure 5, the interaction properties of the device become richer by adding the camera as a relative and absolute movement sensor. Given a camera resolution that is fine enough, in principle arbitrarily small relative motion updates can be sensed. The three horizontally connected circles labeled *sweep* correspond to the 3 DOF and map to the (x, y, θ) dimensions. Although it is possible to also detect relative Z movement and relati-

Figure 4: Phone input interaction: ***point & shoot*** **is mapped to horizontal joystick push-and-release,** ***sweep*** **is mapped to vertical push-and-hold.**

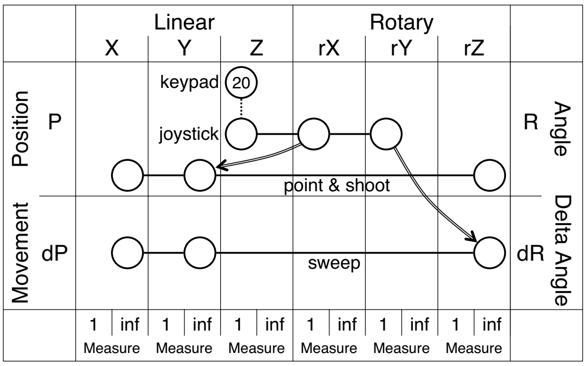

Figure 5: Card-style design space classification [2] of our mobile phone extended by the camera-based interaction techniques.

ve X and Y rotation, we exluded it here in order to focus on the most important data points. In our implementation, relative rotation around the X axis (dR:rX) is equivalent to linear Y motion and relative rotation around the Y axis (dR:rY) is equivalent to linear X motion. This means that for the *sweep* technique, bending the wrist is equivalent to moving the whole arm. In addition, relative Z movement (dP:Z) could be mapped to a further input dimension. The three horizontally connected circles labeled *point & shoot* represent absolute position sensing. It provides the X and Y position and the state of rotation around the Z axis.

When multiple users interact with a large display simultaneously, multiple cursors are required. This can be achieved by shaping or coloring the cursors differently as done in PebblesDraw [6]. The cursor color could match the shape and color of the cursor on the mobile phone to help users identify which large display cursor they are controlling. Additionally, to help users locate their respective cursor on the large screen, a press on a special phone button could shortly flash or highlight their cursor.

The current implementation of the *point & shoot* interaction clearly has disadvantages for multi-user environments, in that flashing the code grid over the display can disrupt the activity of other users. This problem can be addressed in several ways. First of all, the visual codes can be integrated into the application layout, although this may lower its overall aesthetics. Alternatively, infrared display technology could be used so that they are invisible to the human eye, but still detectable with the camera interface.

3.4 Designing for serendipity

In addition to establishing a coordinate plane, we use Visual Codes (see Figure 1) to encode the public display's Bluetooth address information thus enabling a communications channel to be rapidly established between the mobile phone and the large display. Users merely take a picture of a Visual Code associated with the display and the phone will automatically connect to send (x, y, θ, text) information via Bluetooth. The latency to establish the channel is fairly low and the amount of jitter (variance of delay) during interaction is negligible. The same connection can be used to authenticate the user, to send user profiles for adapting the content shown on the large display to personal preferences, to transfer sensitive information to the personal display, and to copy and store information and the current state of interaction on the phone. This creates a very low threshold of use and allows for highly serendipitous interactions. In order to do Visual Code recognition and optical-flow processing, our proposed device interactions require that users install special software on their mobile phone. However, this software could potentially be installed during manufacturing, via the mobile phone network using over-the-air provisioning, or users could retrieve it directly from the computer driving the display via Bluetooth. Fortunately, this software only needs to be installed once and therefore only slightly increases the threshold of use for first time users.

4. IMPROVING PERFORMANCE AND RELIABILITY

We have performed a detailed analysis of these interaction techniques in our previous work [1]. The *sweep* technique performed worse than the *point & shoot* technique and a standard 2D phone joystick for task completion times. One of the limiting factors was that the optical-flow processing has extremely limited capabilities in determining the velocity of arm movement, preventing the use of effective cursor acceleration. This is primarily a problem of perspective, in that the distance from the camera lens to the objects in the image content influences how fast they move in the image picture. Thus, even with perfect optical-flow processing, it is difficult to measure arm velocity. We are currently examining if optical-flow processing can be combined with accelerometers to produce multi-resolution movement detection where the phone can easily distinguish between fast and slow movements to enable better cursor acceleration. Our hypothesis is that combining sensors will also improve the overall reliability of the movement detection system.

Another problem that we discovered in our user tests was that users made many errors in selecting the appropriate technique. This revealed that our mapping of input techniques to the joystick was hard to remember. The research question emerges of how to map multiple techniques with similar interaction semantics onto the limited interface of a mobile phone.

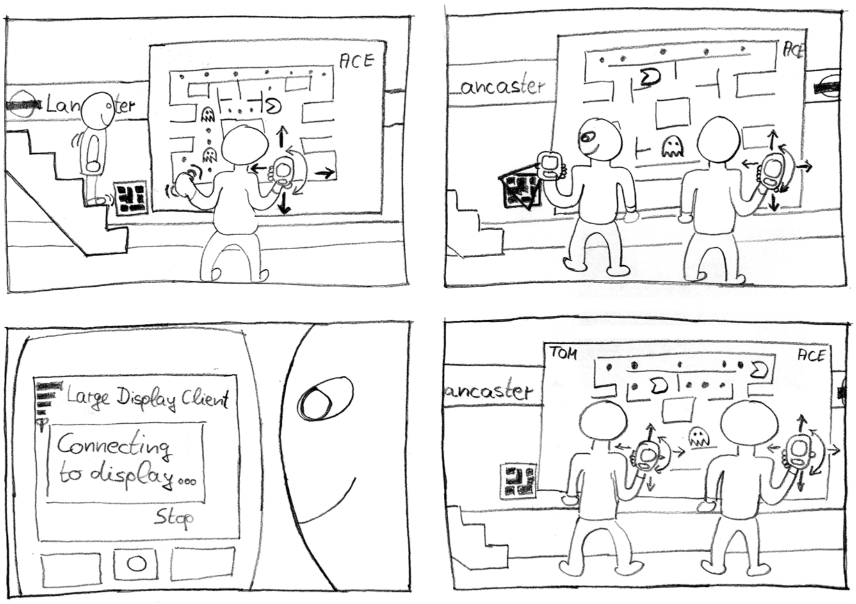

Figure 6: A storyboard illustrating envisioned interactions between mobile phones and large public displays.

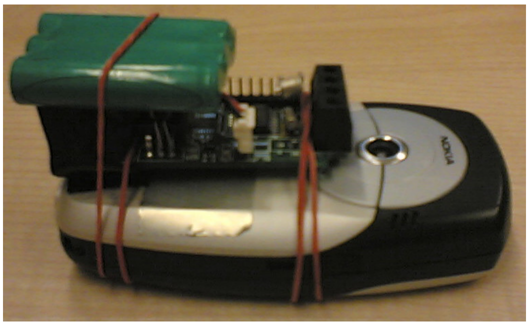

Figure 7: A mobile phone augmented with a 2D accelerometer.

The *Point & Shoot* technique was found to be very sensitive to distance. As users moved away from the display, the targets were perspectively smaller, and thus harder to select. More research is required to determine if zooming lenses or active image stabilization can alleviate these issues. Another approach would be to use hierarchically nested visual codes of several sizes that can easily be recognized from a wide range of distances.

5. CONCLUSIONS

Sweep and *Point & Shoot* enable a new class of highly interactive applications with information terminals or large public displays including interactive art, public games, digital bulletin boards, and advertising. The techniques are functional now, but research is still needed to refine performance and reliability to provide a more pleasant and fluid interaction experience.

6. REFERENCES

[1] Ballagas, R., Rohs, M., Sheridan, J. G., and Borchers, J. Sweep and point & shoot: Phonecam-based interactions for large public displays. *CHI '05: Extended abstracts of the 2005 conference on Human factors and computing systems*. ACM Press (2005).

[2] Card, S. K., Mackinlay, J. D., and Robertson, G. G. A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst. 9*, 2 (1991), 99–122.

[3] MacKenzie, I. S., and Ware, C. Lag as a determinant of human performance in interactive systems. *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press (1993), 488–493.

[4] Madhavapeddy, A., Scott, D., Sharp, R., and Upton, E. Using camera-phones to enhance human-computer interaction. *Adjunct Proc. of Ubicomp*. Springer-Verlag (2004).

[5] Miyaoku, K., Higashino, S., and Tonomura, Y. C-blink: A hue-difference-based light signal marker for large screen interaction via any mobile terminal. *Proc. UIST*. ACM (2004), 147–156.

[6] Myers, B. A., Stiel, H., and Gargiulo, R. Collaboration using multiple PDAs connected to a PC. *Proc. CSCW*. ACM Press (1998), 285–294.

[7] Rohs, M. Real-world interaction with camera-phones. *2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*. 39–48.

[8] Silfverberg, M., MacKenzie, I. S., and Kauppinen, T. An isometric joystick as a pointing device for handheld information terminals. *Proceedings of Graphics Interface 2001*. Canadian Information Processing Society (2001), B. Watson and J. W. Buchanan, Eds., 119–126.