Use Your Head – Exploring Face Tracking for Mobile Interaction

Thomas Riisgaard Hansen
Pervasive Healthcare,
Department of Computer Science,
University of Aarhus
Aabogade 34, DK8200,
Aarhus N, Denmark
thomasr@daimi.au.dk

Eva Eriksson
IDC | Interaction Design Collegium
Department of Computer Science
and Engineering,
Chalmers University of Technology
SE-41296, Göteborg, Sweden
eva.eriksson@chalmers.se

Andreas Lykke-Olesen
Department of Design
Aarhus School of Architecture,
Nørreport 20, DK8000
Aarhus C, Denmark
alo@interactivespaces.net

CHI 2006, April 22–27, 2006, Montreal, Québec, Canada.
ACM 1-59593-298-4/06/0004.

Abstract
In this paper we present how face tracking can be implemented on mobile devices. Our main contribution is to present how face tracking on mobile systems can be used as a multi-dimensional input technique and to demonstrate how this can be used in different mobile applications. We present at set of different applications based on the tracking, and discuss current and future advantages, challenges and problems with face tracking as input device for mobile systems.

Author Keywords
Mixed Interaction Spaces, Mobile Interaction Technique, Computer Vision, Face Tracking, Mobile Devices

ACM Classification Keywords
H5.2. User Interfaces (Interaction styles, Haptic I/O, GUI).

Introduction
Today, many of the applications that previously belonged to the traditional computer are now able to run on a mobile device. As the number of possible tasks to perform increases along with an increased complexity of the task, the limited interaction on mobile devices becomes problematic. Because the devices have to be mobile, the sizes of the screens, the

keyboards, and the keypads are limited. The most common interaction today is through buttons, thumbwheels or pens - through something that can be characterized as a downscaling of the classic WIMP interface.

However, instead of seeing mobility as a problem it opens up for novel interaction techniques that actually use the fact that the device is mobile. One new approach is to equip the mobile devices with accelerometers and use these sensors to register the movement of the device in different applications [8]. Samsung has e.g. introduced one mobile phone (SCH-S310) with accelerometers, but so far the use of accelerometers in mobile devices has been limited.

With Mixed Interaction Spaces (MIXIS) we have taken another approach, where the cameras on mobile devices track the location of the device in relation to a tracked feature. In our first system using MIXIS [3, 4] we determine the location of the mobile device by tracking a printed or hand drawn circle.

Other researchers have worked with cameras as interaction techniques for mobile devices [9, 7]. However, these techniques all use some version of barcode as their feature point. The strength of these approaches are that a unique ID can be encoded in the tags, but the limitation is that these interaction techniques only can be used in front of a computer generated 2D barcode.

In the new face tracking version of MIXIS, presented in this paper, we use a mobile device with a camera pointing towards the user to track the location and rotation of the device by using the user's face as a feature point. Face tracking has traditionally been used to interact with PCs and web-cameras, where the user moves the head to interact. The fundamental difference with face-tracking on mobile devices is that it is not the face that is moved in front of a mobile device, but it is the device that is moved or tilted instead. This greatly minimizes the exhaustion that follows by having to move the head.

Face Tracking and Mixed Interaction Spaces

The main idea of "Mixed Interaction Space" (MIXIS) is to use the space around the mobile device as input, instead of limiting the interaction with mobile devices to the device. With this approach, we are able to create mixed reality applications where the physical space is used to interact with programs running on the mobile device or on nearby computers.

The MIXIS interaction technique uses the mobile device's camera to track feature points and use the distance and rotation of the device from this or these feature points as input vectors to a set of different applications. Depending on how the different movements of the device are mapped to the applications the device can be a 1-4 dimensional input device (x, y, z and rotation).

The tracking works only when the feature point is within view of the camera, resulting in a spanned pyramid-shaped space in which the device can be tracked. The space is narrower close to the feature point and wider further away and ends when the camera loses track of the feature point (see Figure 1).

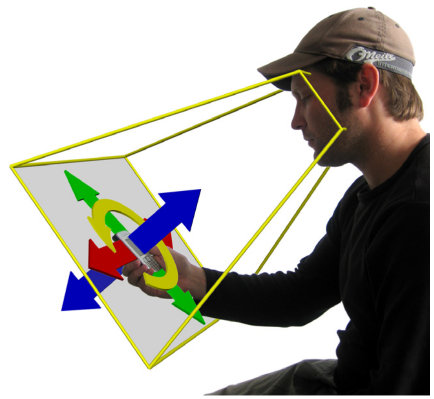

Figure 1: The Mixed Interaction Space with face tracking

The MIXIS concept and some applications are further described in [3, 4], but basically, with the first version of MIXIS a circle was needed for tracking. Even though this assumption was acceptable in a number of different applications it still limited the number of different applications that can use the technique.

With the face tracking version we use the front camera of a 3G mobile device (for testing we used the Nokia 6680) to track the position in 3D and rotation of a person face. The clear advantage of this approach is that there is no need for external symbols e.g. circles or barcodes for the tracking. The face will always be available, ready to be tracked, but even though the face is "just" another type of feature point several new issues arise with the switch to this tracking type.

Tracking Algorithm

The tracking algorithm used for face tracking is strongly inspired by the tracking algorithm described in [1, 2, 5], but it has been modified to run on mobile phones. The basic steps in the algorithm are:

1. Calculate a histogram for the tracked face (object).
2. Repeatedly track the location of the face (or object)

a. Find the position of the center of the tracked object.
b. Move the search window to this position, enlarge it
c. Repeat step a and b until the algorithm reaches convergence or reaches a threshold and determine the location and rotation of the face from the convergent search window.

The first step of the algorithm is to find a probabilistic color histogram of the object to track. In our implementation, the user can choose the face to track by placing the face in the center of the camera view and press a key. The color histogram of a default face can also be hard coded, but the first approach is more flexible and is able to track any object that has a color distribution that is different from the background (From the software it is possible to determine if we want to use the front or the rear camera of the phone).

To create a good probabilistic color histogram we use some of the techniques from [1] e.g. focus only on the hue component of a color, and ignore too bright and too dark pixels. The histogram is used to calculate the moment within a search window as described in [5, 2, 1] and the search window is moved to the center of the mass and expanded. This is repeated until the algorithm converges or reaches a threshold and information from the final search window is used to extract the location and rotation of the tracked object .

The outcome returned from the algorithm is the position of the tracked object, its size, its rotation and a parameter describing the uncertainty in the result.

By applying fixed point arithmetic and other speed performance optimizations we have been able to speed the algorithm significantly and it runs fluently on both Nokia 6680 and Nokia 7610. It takes 75-90 ms to calculate the position based on the current version of the algorithm running on the Nokia 6680 and we are able to run simple graphic applications using face tracking with a frame rate between 11-13 pr. second.

The application runs in many different light condition, but the current algorithm has a problem with low lightning due to the quality of the camera and skin colored backgrounds. However, because the algorithm “stick” to the last found feature it is pretty robust to people walking by or partly occluded faces. Details on how the algorithm reacts to noise can be found in [1].

Applications

To prove the concept we have currently developed four applications using face tracking as input on mobile phones, and several more running on a PC controlled through face tracking on the mobile phone.

Face tracking application

This is a proof-of-concept application, where the face tracking algorithm visualizes the face tracking concept by placing a circle around a persons face. The purpose is mainly for testing, even though it is fun to see if you are able to get away from the pretty sticky tracking.

PongApp

This application is a pong game running on the mobile phone that is controlled by face tracking (see Figure 2, left). The pad can be controlled by moving the phone along the x-axis inside the space or tilting the phone.

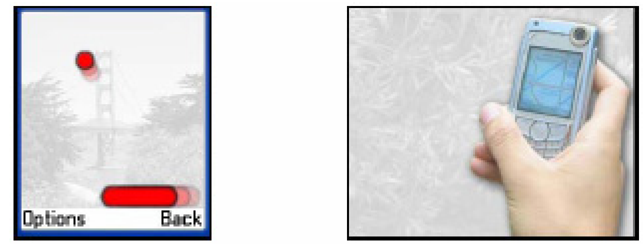

Figure 2: The Pong and the ImageZoomViewer application using face tracking as the only input

ImageZoomViewer

This application was first implemented with the previous version of MIXIS [3, 4]. The idea is that the user is able to simultaneously pan and zoom a map or an image by moving the mobile device in the Mixed Interaction Space (see Figure 2, right). Moving the mobile device up or down, left or right pan the map in the direction the phone is moved. Moving the phone further away or closer to the face results in a zoom out or zoom in action. To overcome the problem about the precision on the depth axis when the face is not fully visible, the user is only allowed zooming when the mobile device is centered on the user’s face.

BlueMix

BlueMix is an application on the phone that forwards the input vector to a Bluetooth connection. A Bluetooth server application on a nearby PC receive the input from up to seven mobile devices running BlueMix and forwards the input to a socket. Programs written in C#, Java or Flash can then connect to this socket and use the input from up to seven devices in their applications. We have currently explored different multi-user applications with the use of the BlueMix program [6].

Preliminary findings with face tracking

By working with face tracking as input we have encountered several characteristics and challenges with this type of input mechanism.

Tilting versus moving

A new way of using MIXIS has been observed with face tracking. Instead of moving the mobile device within the interaction space; it can sometimes be easier to tilt the phone, and thereby move the entire space. Sometimes it is difficult to read the display if it is moved far away from the face in the interaction space and it is therefore easier to tilt the phone. In other circumstances it is more natural to actually move the phone, but we still need to explore this issue further.

Uncertainty when tracking several dimensions

Another relevant issue with face tracking is that the face has the shape of an ellipse. Normally the camera captures in a 4 to 3-format, meaning that there is a much higher tracking resolution along the x-axis than along the y-axis – especially since the face has an oval shape (see Figure 3a). There is a similar issue with the z-axis and rotation. The algorithm is relatively good to determine the x, y location of the face with a large amount of noise, and also if the face is only partly visible. This is however, not true for the z-axis and rotation (see Figure 3b). To calculate the rotation and depth, the entire face has to be within sight, meaning that the depth and rotation information is much more accurate when the face is in the center of the image.

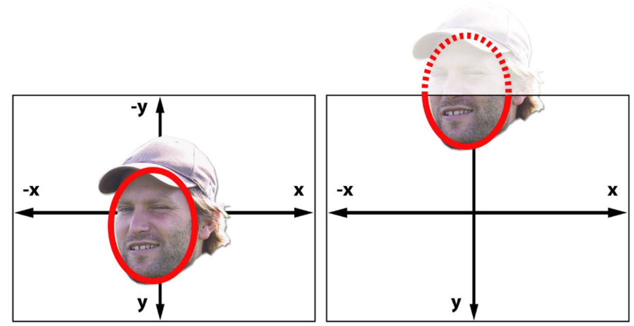

Figure 3: a) The limited resolution on the y-axis, b) the problem of determine the depth and rotation of only partly visible faces

Visualizing the Mixed Interaction Space

The mixed interaction space demonstrated in Figure 1 is not a physical bounded space, but bounded by the vision of the camera. Visualizing what the camera sees and how well it sees is a big challenge. Showing what the camera sees on the screen is one solution, but it becomes more challenging when the limited screen space is being used for other purposes e.g. displaying game graphics or a map. Also, finding a good cursor that works for MIXIS is important for fluent interaction.

Supporting mobility

One of the clear advantages and differences between MIXIS with face tracking and previous and related work is that there is no external feature point. In the previous version of Mixis [3, 4] a circle is needed and in most related projects [9, 7] a 2D barcode is needed for tracking. With face tracking the feature point, the face, is always with the user and if the camera is able to produce a good enough image the tracking system works anywhere.

We have just begun exploring how we can take advantage of a highly mobile interaction system and its consequences. Initial experience with e.g. ImageZoomViewer suggests that it is much more natural to move or tilt the phone in the direction you want the map to move instead of having to press a button several times. Another interesting aspect is the difference between tilting the interaction space and moving the phone. Tilting the phone can be done with really small movements whereas moving the phone in the interaction space require larger movement that can be seen as strange in social settings.

Future work

The main contribution of this paper is both to demonstrate that face tracking is feasible on mobile phones and that it clearly differ from face tracking on PCs using web cams.

Currently we are working on improving the tracking algorithms by looking at both features and color histograms to overcome some of the limitations with the tracking technology. The next step is to develop new applications taking advantage of this type of interaction mechanism and also the usability issues of switching between buttons and face tracking.

Acknowledgement

We would like to thank ISIS Katrinebjerg, Center for Interactive Spaces, Center for Pervasive Healthcare and Chalmers University of Technology for support.

References

[1] Bradski, G.R., Computer Vision Face Tracking For Use in a Perceptual User Interface, Intel Technology Journal Q2, 1998.

[2] Freeman, W.T., K. Tanaka, J.Ohta, and K. Kyuma, "Computer Vision for Computer Games," Int. Conf. On Automatic Face and Gesture Recognition, pp. 100-105, 1996.

[3] Hansen, T. R., Eriksson, E., Lykke-Olesen, A. 2005. Mixed Interaction Spaces – Expanding the Interaction Space with Mobile Devices. In proceedings of HCI2005 on The Bigger Picture (Edinburgh, UK, September 05 - 09, 2005). Springer-Verlag London, 365-380. ISBN 1-84628-192-X.

[4] Hansen, T. R., Eriksson, E., and Lykke-Olesen, A. 2005. Mixed interaction space: designing for camera based interaction with mobile devices. In CHI '05 Extended Abstracts on Human Factors in Computing Systems (Portland, OR, USA, 2005). CHI '05. ACM Press, New York, NY, 1933-1936. ISBN:1-59593-002-7.

[5] Horn, B. K. P. Robot vision. MIT Press, 1986.

[6] Multi-user Mixis home page (last accessed: 10-02-2006) http://www.pervasive-interaction.org/Mixis

[7] OP3.com, ShotCode, (last accessed: 22-09-2005) http://www.op3.com/en/technology/shotcodes

[8] Patridge, K., Chatterjee, S., Sazawal, V., Borriello, G., Want, R., TiltType: Accelerometer-Supported Text Entry for Very Small Devices, UIST'02, October 27-30, 2002, Paris, France, 2002.

[9] Rohs, M., Real-world Interaction with Camera Phones, in H. Tokuda & H. Murakami (eds.), Proceedings of the 2nd International Symposium on Ubiquitous Computing Systems (UCS 2004), 2004.