Supporting Hand Gesture Manipulation of Projected Content with Mobile Phones

Matthias Baldauf
Telecommunications Research Center Vienna (ftw.)
Donau-City-Strasse 1
Vienna, Austria
+43 1 5052830-47
baldauf@ftw.at

Peter Fröhlich
Telecommunications Research Center Vienna (ftw.)
Donau-City-Strasse 1
Vienna, Austria
+43 1 5052830-85
froehlich@ftw.at

ABSTRACT

The detection of a user's hand gestures enables a natural interaction with digital content. Recently, wearable gesture detection systems have been presented which use a camera to visually detect the gestures and tiny projectors to augment nearby surfaces and real-world objects with digital information. Still, current approaches rely on laptop computers restricting the systems' mobility and usability. In this paper, we present a framework for spotting hand gestures that is based on a mobile phone, its built-in camera and an attached mobile projector as medium for visual feedback. Other existing mobile applications can simply connect to our framework and thus, become gesture-aware. The proposed framework will allow us to easily and fast create gesture-enabled research prototypes shifting the user's attention from the device to the content.

1. INTRODUCTION

Mobile phones are the most widespread ubiquitous devices. Due to their inherent context-awareness [2], they are increasingly used to interact with the user's current surroundings and nearby real-world objects. Sensors integrated in today's mobile phones such as GPS receivers, compasses, accelerometers, NFC modules and cameras not only allow to view digital information about such objects but also to manipulate it. Previous research has proven the feasibility and usability of several interaction techniques. Examples include short-range techniques such as touching an NFC-enabled object with a mobile phone [16] and medium- and wide-range techniques such as pointing the device at buildings [19].

More recently, demonstrations of MIT's project "6th sense" [11], a mobile gestural-controlled system, have attracted considerable interest. The portable combination of a common webcam, a laptop computer and a tiny projector (see Figure 1) allows the augmentation of arbitrary surfaces and objects by projected information while triggering actions through natural hand gestures. The visual detection of a user's gestures causes the involved computer to vanish into the background. Still, the system relies on the laptop computer which has to be carried in a backpack when used on the move.

Inspired by this work and with emerging projector phones in mind, we develop a framework supporting hand gesture manipulation of projected content through a mobile phone. Our aim is to make the mobile phone a wearable, truly unnoticeable mediator between the real and the virtual world changing the human interaction style from a device-centric over to a content-centric one. Existing mobile applications can simply connect to the proposed framework and thus, made gesture-aware.

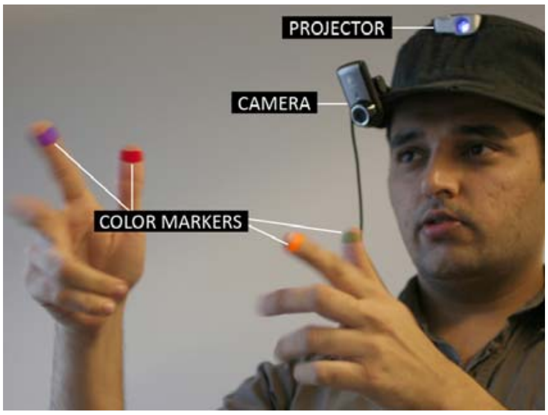

Figure 1: Setup of MIT's project "6th sense" [11]

In this paper, we present our current work on this highly mobile and autonomous gesture detection framework. After giving an overview of related work in Section 2, we describe our hardware setup in Section 3. In Section 4 we describe the chosen event and gesture model. Section 5 explains the proposed system's architecture and provides implementation details. We conclude with an outlook in Section 6.

2. RELATED WORK

In the past few years, gesture-based interfaces made their way to market-ready or even mainstream products. Examples include Microsoft's Surface [10], a table with a touch-sensitive top responding to hand gestures and real-world objects, and Apple's iPhone [1], a multi-touch-enabled smartphone. Besides such devices that have to be physically touched for interaction, impressive touchless appliances emerge. E.g. Ubiq'window [8] enables gestural interaction with a screen behind glass through optical motion detection. g-speak [14] uses special sensor gloves for detecting spatial hand gestures. However, these applications rely on expensive custom hardware and are not mobile.

Research in the field of mobile computing also investigated the usage of acceleration sensors to detect manual gestures [9]. One of the favored use cases is the interaction with connected real-world objects such as distant displays through a gesture-aware mobile phone [4]. Another possibility to detect a phone gesture is to analyze the built-in camera's video stream [7]. With the enhancements in mobile hardware, more complex computer vision algorithms can be realized on smart phones leading to handheld augmented reality applications [17][15].

As a visual feedback medium, mobile projectors are increasingly applied. Examples include work augmenting real-world items such as maps with overlaid digital data [18] and studies evaluating the usability of such extended displays [6].

Recently, the aforementioned project "6th sense" [11] combined visual gesture detection methods performed on a laptop computer with projector feedback. The resulting wearable device built from off-the-shelf components visually augments objects the user is interacting with. An example for a similar but custom hand gesture interaction device is "Brainy Hand" [5], a small gadget that comprises an earphone, a color camera, and a mini-projector and is attached to one ear.

3. HARDWARE SETUP

For our purely mobile-based setup, we attached a tiny projector to a smartphone simulating upcoming mainstream projector phones. Figure 2 depicts the hardware components of our current setup.

We use a Nokia N95 mobile phone, a smartphone running Symbian OS with the S60 platform. Due to its multitasking capability and its built-in camera, we are able to execute both the gesture tracking engine and the gesture-enabled application on one device. Lots of mobile research prototypes – applications we want to gesture-enable - are implemented using the Java 2 Micro Edition which is featured by Symbian OS. Alternative smartphone operating systems are not suitable for our approach: e.g. the iPhone lacks multitasking support and is only scarcely used in research projects, and so far, none of the available Android-powered phones provides a video output.

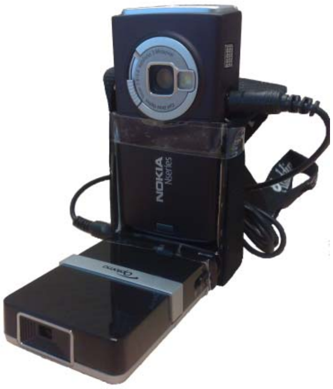

Figure 2: A N95 mobile phone and a PK101 projector assembled to a wearable gadget

In order to augment nearby surfaces or objects we apply the pocket projector PK101 from Optoma. This LED-based projector with similar dimensions as the N95 is perfectly suited for mobile usage and is connected to the phone through a short video cable. Such an assembled gadget can be worn like pendant, i.e. both devices are arranged along a lanyard worn around the neck. In contrast to the setup in the "6h sense" project [11], the lanyard contains the complete equipment, there is no additional backpack needed. For alternative perspectives, only the camera phone can be attached to the lanyard and the projector is integrated in a hat or the user's clothing.

Thus, the presented equipment is highly mobile and easy to use: No backpack for a laptop computer is necessary, no annoying cables are involved. The equipment consists only of off-the-shelf components and is completely autonomous.

4. EVENTS AND GESTURES

In order to ease the detection of gestures and identify single fingers, we attach colored markers to the user's fingers. During the detection process, we distinguish between low-level events and gestures as a combination of such events. Our current prototype is capable to recognize the following three low-level events.

- *Marker detected*. This event occurs when a marker is detected in a video frame but the same marker was not present in the previous frame, i.e. this marker just appeared. The event's parameters include the color of the detected marker as well as the position in pixels where the marker has appeared.
- *Marker moved*. When a marker was present in the previous frame and is detected at another location in the current frame, this event is triggered. Again, the event's parameters contain the spotted marker's color and its current location.
- *Marker lost*. A formerly detected marker can not be recognized in the current video frame, i.e. the marker disappeared. This event only owns one parameter, namely the color of the lost marker.

Based on these three fundamental events we define several gestures. Such gestures combine at least two low-level events and abstract from pixel-sensitive positions forming more meaningful high-level actions. Gestures can be either absolute or relative ones. Absolute gestures directly operate on the displayed information, e.g. by pointing at a shown photo to select it. Thus, some kind of calibration is necessary for absolute gestures in order to map camera-detected positions to display-coordinates. At the moment, our current prototype features relative gestures, i.e. gestures derived from the motion and geometric relation of the involved markers.

We named the implemented gestures according to their most likely uses.

- *Panning*. This gesture involves only one marker. The panning occurs when a formerly detected marker moves. Besides the type of marker, this gestural event informs about the relative movements on the horizontal and vertical axis.
- *Scaling*. This well-known gesture measures the distance between two formerly spotted markers. By moving the markers closer or farther away from each other, scaling or zooming actions might be triggered. The appropriate parameters include the types of the two involved

markers as well as the relative distance between them with regard to the first measured distance.

- *Rotating*. Similarly to scaling, this gesture is based on the interaction of two spotted markers. Instead of the distance, the slope of the line defined by the markers is measured. The parameters again consist of the two involved markers and the change of the slope in degrees. This value is relative to the initially detected slope. Obviously, the scaling and rotating gestures can be combined by changing both the marker's distance and orientation at the same time.

So far, we implemented gestures making use of a maximum of two markers. The tracking engine can be extended to detect more markers and thus, support more complex gestures.

5. SYSTEM ARCHITECTURE

Figure 3 gives an overview of our approach's software architecture. It consists of two main components where one element is responsible for the actual tracking of gestures and the other one triggers the according actions as part of the application to be gesture-enabled.

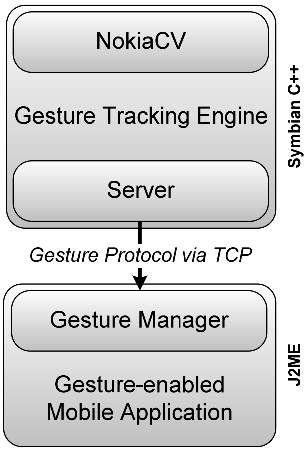

Figure 3: Architecture consisting of the Gesture Tracking Engine and the gesture-enabled application communicating via a local socket connection

In order to run computational intensive operations such as image recognition algorithms at interactive frame rates on resource-limited devices like mobile phones, they have to be implemented in native code. We based our gesture tracking engine on NokiaCV, a computer vision library written in Symbian C++. NokiaCV comes with source code and provides standard image operations and basic image recognition methods such as implementations of corner or edge detection algorithms. We adapted some of the algorithms to analyze the video stream provided by the built-in camera. The presented gesture tracking engine uses a color-sensitive detection technique and therefore, is able to recognize and track differently colored markers in the viewfinder's image.

Once, a video frame has been completely analyzed, the tracking engine determines the occurred events described in Section 4. The according gestures are then derived by comparing the events to the ones detected in the former frame.

To notify another local application about spotted actions, the gesture tracking engine contains a small server component. As we only allow one client application to connect to this server, complexity of client management is reduced. For conveyance, the events and gestures are wrapped in a simple "gesture protocol", i.e. a short textual description of the events and gestures together with their particular parameters. An application might not only be interested in gestures but also in low-level events. E.g. an application might provide an acoustic signal to indicate a "marker detected" event – which usually marks the starting point of a gesture.

Obviously, any local application may connect to the gesture tracking engine, independent of the language it is written in. For a J2ME application to become gesture-aware, we provide the so-called "Gesture Manager". On initialization, this J2ME component connects to the gesture tracker engine and waits for incoming notifications to unwrap. Following the well-known observer pattern, a gesture listener has to be provided to the Gesture Manager. This gesture listener describes the operations to be triggered when a certain event or gesture is detected. In case no connection to the tracking engine could be established, the listener is ignored and the application's behavior is unmodified.

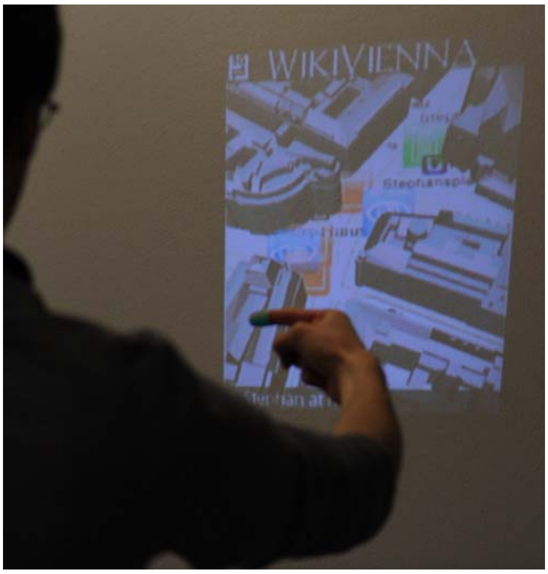

Figure 4: Controlling a mobile 3D urban model through a panning gesture

As an example, Figure 4 shows a mobile 3D urban exploration tool developed in our project 'WikiVienna' [3]. The application was made gesture-aware through the presented framework. We use the three supported gestures to move the point of view, to zoom in and out, and to change the viewing angle.

6. CONCLUSION AND OUTLOOK

In this paper we presented our ongoing work on a gesture detection framework for mobile phones. The framework aims at easily adding gestural interaction support to existing mobile application and, respectively, enables the rapid development of gesture-aware research prototypes. Our current framework prototype is deliberately designed for experimentation.

Future work will include the implementation of absolute gestures to directly operate on the projected content. Therefore, appropriate calibration and mapping techniques have to be developed.

A general problem when using a visual detection method and a projector as feedback medium, are the light conditions. Whereas the projected image can be recognized best in a rather dark setting, the visual detection works best in a well-illuminated ambience. Therefore, we will try to improve the robustness of our detection approach making it as illumination-invariant as possible in future work. The implementation of more sophisticated computer vision algorithms might even allow marker-less gestural interactions.

7. ACKNOWLEDGMENTS

This work has been carried out within the projects WikiVienna and U0, which are financed in parts by Vienna's WWTF funding program, by the Austrian Government and by the City of Vienna within the competence center program COMET.

8. REFERENCES

[1] Apple iPhone. http://www.apple.com/iphone/

[2] Baldauf, M., Dustdar, S., and Rosenberg, F. 2007. A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. 2, 4 (Jun. 2007), 263-277.

[3] Baldauf, M., Fröhlich, P., and Musialski, P. 2008. WikiVienna: Community-Based City Reconstruction. IEEE Pervasive Computing Magazine, Vol. 7, No. 4.

[4] Dachselt, R., and Buchholz, R. 2009. Natural throw and tilt interaction between mobile phones and distant displays. In Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems (Boston, MA, USA, April 04 - 09, 2009). CHI EA '09. ACM, New York, NY, 3253-3258.

[5] Emi, T., Takashi, M., and Jun, R. 2009. Brainy hand: an ear-worn hand gesture interaction device. In Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems (Boston, MA, USA, April 04 - 09, 2009). CHI EA '09. ACM, New York, NY, 4255-4260.

[6] Greaves, A., Hang, A., and Rukzio, E. 2008. Picture browsing and map interaction using a projector phone. In Proceedings of the 10th international Conference on Human Computer interaction with Mobile Devices and Services (Amsterdam, The Netherlands, September 02 - 05, 2008). MobileHCI '08. ACM, New York, NY, 527-530.

[7] Hannuksela, J., Sangi, P., and Heikkilä, J. 2007. Vision-based motion estimation for interaction with mobile devices. Comput. Vis. Image Underst. 108, 1-2 (Oct. 2007), 188-195.

[8] LM3LABS Ubiq'window. http://www.ubiqwindow.jp

[9] Mäntyjärvi, J., Kela, J., Korpipää, P., and Kallio, S. 2004. Enabling fast and effortless customisation in accelerometer based gesture interaction. In Proceedings of the 3rd international Conference on Mobile and Ubiquitous Multimedia (College Park, Maryland, October 27 - 29, 2004). MUM '04, vol. 83. ACM, New York, NY, 25-31.

[10] Microsoft Surface. http://www.microsoft.com/surface/

[11] Mistry, P., Maes, P., and Chang, L. 2009. WUW - wear Ur world: a wearable gestural interface. In Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems (Boston, MA, USA, April 04 - 09, 2009). CHI EA '09. ACM, New York, NY, 4111-4116.

[12] Mobile Spatial Interaction Initiative. http://msi.ftw.at

[13] Nokia Computer Vision Library. http://research.nokia.com/research/projects/nokiacv/

[14] Oblong g-speak. http://oblong.com/

[15] Rohs, M., Schöning, J., Krüger, A., and Hecht, B. 2007. Towards Real-time Markerless Tracking of Magic Lenses on Paper Maps. In Adjunct Proceedings of the Pervasive 2007.

[16] Rukzio, E., Broll, G., Leichtenstern, K., Schmidt, A. 2007. Mobile Interaction with the Real World: An Evaluation and Comparison of Physical Mobile Interaction Techniques. European Conference on Ambient Intelligence (AmI-07). Darmstadt, Germany.

[17] Schmalstieg, D., and Wagner, D. 2008. Mobile Phones as a Platform for Augmented RealityProceedings of the IEEE VR 2008 Workshop on Software Engineering and Architectures for Realtime Interactive Systems (Reno, NV, USA), pp. 43-44, IEEE, Shaker Publishing, 2008-March

[18] Schöning, J., Rohs, M., Kratz, S., Löchtefeld, M., and Krüger, A. 2009. Map torchlight: a mobile augmented reality camera projector unit. In Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems (Boston, MA, USA, April 04 - 09, 2009). CHI EA '09. ACM, New York, NY, 3841-3846.

[19] Simon, R. and Fröhlich, P. 2007. A mobile application framework for the geospatial web. In Proceedings of the 16th international Conference on World Wide Web (Banff, Alberta, Canada, May 08 - 12, 2007). WWW '07. ACM, New York, NY, 381-390.

[20] Wang, J., Zhai, S., and Canny, J. 2006. Camera phone based motion sensing: interaction techniques, applications and performance study. In Proceedings of the 19th Annual ACM Symposium on User interface Software and Technology (Montreux, Switzerland, October 15 - 18, 2006). UIST '06. ACM, New York, NY, 101-110.