Gesture interaction techniques on cell phones: overview and taxonomy*

Dani Rotzetter
Department of Informatics
University of Fribourg
Bd. de Perolles 90
1700 Fribourg, Switzerland
dani.rotzetter@unifr.ch

ABSTRACT

Cell phones serve more and more as a replacement for a variety of devices that are used to accomplish everyday tasks. In parallel, the research on gesture interaction is evolving. When bringing those two aspects together, we enter a relatively new research topic of gesture interaction on mobiles. In this article, a coarse taxonomy on current gesture interaction approaches on cell phones is presented. Besides revealing possibilities and a short description, some examples of the different methods are discussed, along with advantages and limitations and, if available, possible solutions.

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information interfaces and presentations—*Input devices and strategies*

General Terms

Theory

Keywords

Mobiles, gesture interaction, taxonomy

1. INTRODUCTION

Smart phones are increasingly used by all strata of the population. Along with this, one can observe the development of human-computer-interactions that are based on techniques other than pure key strokes. Today's cell phones often feature a touch screen, sophisticated sensors and cameras, which can capture and process gesture information. This may result in a convenient, precise and smooth usage of applications.

In this paper, the different gesture interaction methods and their advantages and possibilities are depicted.
Table 1 summarizes the techniques briefly.
Finally, we will conclude with an estimation of the further development in mobile gesture interaction.

2. GESTURE INTERACTION TECHNIQUES

2.1 Touch interfaces

Almost every newly produced smart phone comes with a touch screen. The advantage of this widespread technology is its relatively cheap implementation and straightforward use. People do not have to learn how a program can be opened or a button is to be activated; they just simply touch it. This is the most frequent touch interaction, called *direct touch*.
Touch as a input method is the only technique discussed in this paper that is limited to two-dimensional gestures. It is more cumbersome to type in longer texts and execute 'rich' tasks using a touch screen compared to a desktop PC. Furthermore, precise and accurate selections can become difficult on small touch screens. In addition, most users navigate on the touch screen with their thumbs. By doing so they not only hide a big part of the screen, but also have troubles reaching its corners and select small icons as the thumb's surface is relatively big.

To face these problems, an offset cursor could be used, where the selection area is slightly shifted from the actual finger-screen contact position. There, the selection area is not hidden by the finger and more precision is possible[6]. Another solution is to enlarge the area around the cursor, thus providing a 'magnifier'[2]. Or a magnetic-like function could automatically select targets close to the finger's touch contact point[1], making it easier to click on small icons. Furthermore, a hybrid technique was developed, where a copy of the selected area is shown in a more visible place of the screen to allow more accurate cursor positioning[7].

Roudaut et al.[8] developed two further techniques to overcome the above stated problems. The first one is *TapTap*, in which a zoomed window is displayed after tapping on a target to allow a precise selection. *MagStick* is another solution that allows the user to control a 'stick pointer' by dragging its end like a telescope with a fixed centre. This approach is combined with a magnetic function that automatically selects close targets. Both techniques - TapTap and Mag-

*The Paper is a delivrabled of the MSc Research Seminar *Multimodal Interaction on Mobiles Devices* in 2012, DIVA Group University of Fribourg, supervised by Dr. Denis Lalanne (`https://diuf.unifr.ch/main/diva/teaching/seminars/seminar-multimodal-interaction-mobiles-devices`)

Stick - proved to be more accurate and more appreciated by the users than any alternatives mentioned earlier. However, direct touch still remains the fastest technique.

Even though touch interaction may suffer from accuracy issues, it will remain the most frequent interaction technique thanks to its wide acceptance and straightforward use.

2.2 Position sensors and accelerometers

Tilt sensors or accelerometers can regularly be queried by tools to trigger actions according to the cell phone's orientation. Instead of the user's movements, the way he is moving and holding the mobile is analyzed.

An experienced operator is familiar with the gesture of rotating the mobile to switch its display mode from horizontal to vertical and vice-versa. A further natural device gesture is to turn the phone in order to reject a call. But besides those straightforward movements, there exist few universally accepted set of moves. Ruiz et al.[10] conducted a guessability study to find out that there are four principles that should be considered when defining application gestures: *Real world metaphors*: people tend to transform mental visions into appropriate, equivalent physical actions. When defining a gesture to return to a 'home' place, for instance, the mobile could be shaken to emulate the idea of 'shaking off' the current situation. *Mimic normal use*: users frequently attempt to perform movements like 'in real life'. For example, in order to accept a call, the cell phone is held to the ear. *Natural consistent mapping*: opposed actions should be achieved by reverse movements. If a 'tip' to the left evokes opening the preceeding page in a document viewer, then the right-tip is applied to switch to the next page. *Need for feedback*: from investigation, it has been learned that it is essential for users to get feedback from the application.

A challenge is that the device must somehow be able to distinguish between intended and regular motion[1] (e.g. not interpreting the device rotation and position when the user sits down on a chair). For this purpose, Ruiz and Li[9] have proposed a gesture called *Double-Flip*, which consists of rotating the device along its longitudinal axis until the display is invisible, and then back. This pattern has proven itself and is already used in a Google application called Gesture-Search[2].

Designers should also note that in a crowded environment, people hesitate to perform conspicuous actions unless the gesture is obviously and unambiguously interpreted by surrounding folks as a device command.

Despite those restrictions, additional capabilities and sometimes more accurate and natural movements come up with motion gestures. This might boost the technique in the future - mainly if it has been agreed on a consistent and universal gesture set. However, the discussed situation-dependent repulsion is likely to limit the field of application.

[1] In fact, a hypothetical perfect system would not need an explicit distinction but automatically recognize commands. However, research has shown that users still prefer to explicitly denote an action[9]. Users want to be in full control of the device and feel unsafe when all gestures are interpreted.

[2] `http://www.google.com/mobile/gesture-search/`

2.3 Camera-based techniques

Gestures can also be captured by relying on the in-built camera of mobiles. The obvious advantage hereby is the sensor-independence. As sensors are additional built-in gadgets, they lead to a more pricey device. Cameras, however, are nowadays also present in cheap mobiles.

2.3.1 Device motion detection

An application can detect horizontal, vertical and rotational tilt movements from the pictures of the camera. In contrast to the accelerometers, where the change in motion speed is detected, the camera spots background shifts.
Many implementations must be carefully calibrated and require a static background with high contrasts, which obviously limits the application scope. To face this problem, Wang et al.[11] have developed a background-independent technique called *TinyMotion*. The strength of TinyMotion lays in its algorithm: at first, the detected image is converted to a gray scale image, which can be processed easier and faster. Secondly, the software applies grid sampling in order to reduce CPU and memory usage. By doing so, a very high recognition rate is achieved, and almost uniform backgrounds, like a blue sky, are supported, too.
Camera-based motions can be used by any software for general tasks such as panning, zooming or even authentication by forcing the user to execute a secret movement pattern.

2.3.2 User motion detection

Another option is to capture the movement of the *user*. Novel interactive commands are supported thereby, such as air-waving to navigate between pages or music tracks. Or - if the mobile device is positioned upright - spreading and contracting the hands to zoom in and out. It is however difficult to design such a variety of gesture commands that a smooth and straightforward interaction with an application is guaranteed.

A more promising utilization are virtual keyboards. Since mobile devices are small, the displayed touch-screen keyboard or an attached mini hardware keyboard are too compact for handy typing, especially for longer texts. To cope with this hurdle, the mobile camera can be used to capture finger movements performed on a virtual keyboard. The typing area must lie on a flat surface. The keys could be projected by an additional mini laser projector and thereby facilitating typing. A supplementary second camera can increase the hit ratio, with the drawback of an additional device which has to be carried.

Murase et al.[5] have built a prototype that works with just one camera. The cost of this simpler solution is the increased error rate and issues with robustness, accuracy, light variations and 'hidden fingers'.
The virtual keyboards so far lead to fatigue and cramped hands, as the whole palm must hover and only the 'typing finger' hits the surface.

The novel interaction technique is interesting for some applications. Users, however, might not only feel uncomfortable

Technique	Description	Advantages	Disadvantages	Application examples
Touch	Using fingers or palm on touch-sensitive display (resistive: pressure needed, vs capacitive: using conductor)	Mature technology, widespread, cheap	Sometimes unnatural gestures required, hidden screen, reach screen corners	Touch operating systems (Android OS, iOS, Windows Mobile, Windows 8...), map navigation
Position sensors/ accelerometers	Integrated sensors detect the phone's alignment and movement	Environment independent	Unintentional triggering (false positives): needs a delimiter, limited scope of application: cannot replace all commands, social acceptability, no universally defined set of gestures	Accept and reject calls by moving mobile to ear or turning screen, automatic horizontal/ vertical screen alignment
Camera-based device motion detection	Cameras detect background alterations and therefrom deduce the device's movements	Most mobiles feature a camera	Low recognition rate (false negatives), sometimes slow, background requirements, may require calibration	Games, authentication gesture
Camera-based user motion detection	Cameras detect user's gestures	Most mobiles feature a camera	Illumination requirements, camera opposed to the screen, partially low recognition rates	Panning, zooming, authentication, games, virtual keyboard
Proximity sensors	Sensors detect, for example by analyzing infrared light reflection, a presence or a proximity of an object	Device orientation and size independence, simpler analysis compared to picture data	Requires additional sensors (integrated or adapter), limited set of gestures possible	Page and music track navigation, accept call, mute phone
Magnetic field based interaction	Digital compass sensors are queried to detect the position of a magnet	Allows smooth and continuous movements, alignment and illumination independence	Sophisticated algorithms and calculations needed, requires additional magnetic gadget	Zooming, list navigation, volume control, instruments application

Table 1: Overview of gesture interaction techniques for mobiles

in performing larger gesture actions and find the virtual keyboard too tiring, but could shy away from this technique due to the elevated error rates.

2.4 Proximity sensors

Infrared sensors can detect the proximity of an object. This feature can also be used for interaction. The advantage of this technique is the screen size and - if the sensors are placed accordingly - the orientation independence. Hence it would not be a problem to interact with small mobiles. Furthermore, there is no more need of having the device's screen insight. For example, when discussing with a partner, it is not polite to take out the phone and manipulate on the screen. Proximity-based devices can, in this case, be controlled unobtrusively.
Provided that the sensors work with high precision, even 3D controls are possible. A sophisticated framework could combine the three-dimensional user interface on the screen with the exact finger position, enabling mobile interaction in space.
Developers can tackle the problem of inaccuracy in gesture detection by either including more sensors, by adapting the recognition algorithm to the environment [4] or by relying on more precise but more expensive IR detectors. It is possible that, as technique evolves and prices decrease, the more precise *Time-of-flight* cameras will find their way into smart phones.

Kratz and Rohs[4] have attached distance sensors to an I-Phone and developed a sample color-choosing tool to show the possibilities of proximity-based gesture interaction. Using wipe movements, the user can then browse through colors. When his hand approaches the screen, the currently active color is selected, whereas deselection is triggered when pulling back the palm from the screen.

Besides a lower recognition rate compared to other methods, there are also multiple built-in sensors needed, which is not standard for today's mobiles. Unless those points are resolved, the field of application with proximity sensors seems rather limited.

2.5 Magnetic field controlled gesture

A rather innovative technique relies on the magnetic field around a device. For the purpose of more precise map navigation, some mobiles combine the GPS receiver with the magnetic field sensor, the latter serving as a digital compass.
Several research approaches take advantage of this hardware to introduce a novel interaction method: the user either fixes a magnetic ring or simply picks up a suitable rod or pen and then performs an action using his hand or finger. The compass sensors examine the strengths of magnetic waves on the three axis and therefrom deduce the magnet's current position, allowing spacial gestures to be recognized.

The technique can be used in existing programs for continuous zooming or list navigation. A smooth track selection and volume control, for instance, can be implemented in a music player. Besides, a big variety of other control patterns allow for new tools and games that could not be implemented before. Music instruments, for example a guitar, can be imitated in a more natural way than ever before[3].

The alignment independence is a resulting advantage: the technique also works, if the screen is turned away or if the mobile is worn upside-down in the trouser's pocket. Besides, in contrast to vision-based techniques, there are no issues with illumination variation.
The additional magnetic gadget that is required beyond the mobile device itself is very simple, light and small. Nonetheless, people would need to carry an additional object with them, which slightly reduces portability and could lower the acceptance of the new technique.

3. CONCLUSION

We have seen different techniques that are applied to interact with the cell phone by using gestures. The touch approach is very widespread and mature in both technology and implementation. It has become a standard for smart phones. Regularly, mobile devices also contain position sensors and accelerometers that allow to detect a user's 'natural' gestures. But depending on their environment, people reject this technique because of shyness and spacial limitations. A big majority of new mobiles has an in-built camera. Applications can make use of it to implement motion detection of both the device and the user's gestures. The sensor independence is paid by some loss of precision and reliability. If a mobile provides proximity sensors, around-device interaction is possible. As a last technique, we have discussed the interpretation of the magnetic field to detect the position of small metal objects like rings. Whilst this method provides novel visibility independent controls, its downside is the requirement of an additional magnetic gadget.

Overall, we can conclude that from a user's point of view, touch is still the simplest and most reliable input. Besides, with the upcoming integrated sensors and cameras, new methods can enrich the interaction with mobiles and complement today's touch screen usage. Whilst relying on one of the novel techniques is a challenge for both developers and users, the combination of them may yield interesting applications and funny games that are controlled through more natural and intuitive gestures.

References

[1] Yves Guiard, Renaud Blanch, and Michael Beaudouin-Lafon. Object pointing: a complement to bitmap pointing in guis. *GI '04: Proceedings of Graphics Interface 2004*, May 2004.

[2] Amy K. Karlson and Benjamin B. Bederson. Thumbspace: Generalized onehanded input for touchscreen-based mobile devices. *Proc. INTERACT 2007*, 2008.

[3] Hamed Ketabdar, Amirhossein Jahnbekam, and Kamer Ali Yüksel. Magimusic: Using embedded compass (magnetic) sensor for touch-less gesture based interaction with digital music instruments in mobile devices. *TEI '11: Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 241–244, January 2011.

[4] Sven Kratz and Michael Rohs. Hoverflow: Expanding the design space of around-device interaction. *MobileHCI '09 Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, September 2009.

[5] Taichi Murase, Atsunori Moteki, Noriaki Ozawa, Nobuyuki Hara, Takehiro Nakai, and Katsuhito Fujimoto. Gesture keyboard requiring only one camera. *UIST '11 Adjunct: Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, pages 9–10, October 2011.

[6] Richard L. Potter, Linda J. Weldon, and Ben Schneiderman. Improving the accuracy of touch screens: an experimental evaluation of three strategies. *CHI '88 Proceedings of the SIGCHI conference on Human factors in computing systems*, 1988.

[7] Xiangshi Ren and Shinju Moriya. High precision touchscreens: design strategies and comparisons with a mouse. *Transactions on Computer-Human Interaction (TOCHI)*, September 2000.

[8] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. Taptap and magstick: improving one-handed target acquisition on small touch-screens. *AVI '08: Working conference on Advanced visual interfaces*, May 2008.

[9] Jaime Ruiz and Yang Li. Doubleflip: A motion gesture delimiter for mobile interaction. *CHI '11: Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 2717–2720, May 2011.

[10] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. *CHI '11: Proceedings of the 2011 annual conference on Human factors in computing system*, May 2011.

[11] Jingtao Wang, Shumin Zhai, and John Canny. Camera phone based motion sensing: Interaction techniques, applications and performance study. *UIST '06: Proceedings of the 19th annual ACM Symposium on User Interface Software and Technology*, October 2006.