

Design and Development of an Everyday Hand Gesture Interface

Zoltán Prekopcsák
Kitchen Budapest
30 Ráday street
Budapest, Hungary
preko@kitchenbudapest.hu

Péter Halácsy
Kitchen Budapest
30 Ráday street
Budapest, Hungary
peter@halacsy.com

Csaba Gáspár-Papanek
Budapest University of
Technology and Economics
2 Magyar Tudósok krt.
Budapest, Hungary
gaspar@tmit.bme.hu

ABSTRACT

Hand gesture interfaces gained great popularity due to game consoles in the past few years and they have the potential to appear in our everyday lives too. In this paper, we introduce four design principles for an everyday hand gesture interface. We evaluate related works and show a prototype that has been designed and developed with these principles in mind. This prototype satisfies most of the principles, and can be easily extended to satisfy all of them when previously announced hardware and software tools hit the market.

Categories and Subject Descriptors

H.5.2 [Information interfaces and presentation]: User Interfaces—*prototyping, user-centered design*; I.5.m [Pattern recognition]: Miscellaneous

General Terms

Algorithms, Design, Human Factors

Keywords

Gestural user interfaces, design principles, machine learning

1. INTRODUCTION

Hand gesture recognition is a discipline that has been around for decades in the CHI research community. Its prototypes range from special applications like sign language interpretation [5] to general gesture recognizers, and they provide various hardware and software solutions for capturing and processing human hand motion. Lately, the Nintendo Wii game console popularized the concept with its gesture-based controller. Using Wii Remote and Nunchuk, games can be controlled by natural gestures which made the console very popular, especially for sports games.

Although gesture-based games spread fast all around the world, there has been very few comprehensive studies about

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobileHCI 2008, September 2-5, 2008, Amsterdam, the Netherlands.
Copyright is held by authors/owners.

everyday use of gesture interfaces. Korpipää et al. have created a context-aware interface prototype that included hand gesture recognition for smart phones, but the rule-based setup concept was quite complicated for first-time users. Their earlier works included more prototypes for specific applications [3].

In our work, we intend to create a customizable gesture interface prototype as accessible as the Nintendo Wii. In the following sections, we introduce our general design principles and show the construction steps of the prototype that has been realized for everyday use.

2. DESIGN PRINCIPLES

The design of the prototype was an iterative process, because new ideas and perspectives arose after every finished development step. The system was tested and discussed with a diverse group of professionals from artistic and engineering fields who gave extensive feedback on the system at every stage. The below mentioned design principles cover these ideas and they are listed in the order they appeared.

2.1 Ubiquity

The gesture interface should be available everywhere, not restricted to time and place. Most gesture recognizer systems are based on video cameras [6], which usually makes them immovable. It works very well in controlled environments, but it is not suitable for everyday use. One alternative is to use accelerometer sensors, which is small enough to build into mobile devices. We have used a Sony-Ericsson W910i mobile phone, which has in-built accelerometer.

For perfect ubiquity, the signal processing should be on the mobile device too. We have decided not to take this step now, as current mobile phones have strong restrictions on user application rights. As long as these restrictions apply, we have created the processing logic on a personal computer as it gives more freedom in controlling different devices. It even has more freedom to control the mobile phone itself by simulating keystrokes via Bluetooth serial AT commands. To help porting the PC application to future mobile devices, we have created the algorithm to use minimal resources. Although we couldn't provide perfect ubiquity, the prototype works perfectly at the Bluetooth range of a PC, which is sufficient in many cases.

2.2 Unobtrusiveness

Unobtrusive (or inconspicuous) means that we barely notice that we are using an interface. The gesture system

shouldn't use special controllers or gloves for processing and sensor placement, just our everyday clothes and devices.

According to user feedbacks, one of the most important features is to cover interface details from the user. Thus, we need to avoid special rules for the use of the system. Most accelerometer-based prototypes have a button with which the user signs the start and the end of the gesture. It is extremely reliable but can be disturbing. Our approach is based on the work of Hofmann et al. [2] with some refinements. There is no need for a button as the gestures are automatically extracted from the continuous sensor data stream which is called segmentation. The only rule is to keep small pauses between gestures. Currently the mobile phone should be kept in hand to perform the gestures, but there is no need to press buttons.

Another very important feature when speaking about interfaces, is the response time. Usability engineering books suggest that response time should be as low as 100ms [4]. On an average PC, the startup time of our system was around 300ms and the recognition time between 1-5ms. For a mobile phone, these values are expected to rise with an order of magnitude which results a few seconds of startup time, but the recognition and response time stays far below 100ms.

2.3 Adaptivity

There are two possibilities when designing a gesture interface: either create a universal gesture set or let users define personalized gestures. In the first case, it is quite difficult to decide about the gesture set as gestures greatly depend on culture and personal preferences. The most obvious personal preference is being right- or left-handed, but there are many more that affect our natural gestures. Hence, the gesture interface should adapt to user preferences, either by altering universal gestures, or allowing the definition of personalized gesture sets.

Observation-based design have been proposed to reveal the optimal gestures for a given task [1], but it is a long design process which is not suitable for everyday use. To be able to cover all everyday tasks, we decided to create a completely adaptive system which lets users define gesture sets with examples. We involved machine learning techniques like Hidden Markov Models and Support Vector Machines to ensure the general applicability of the system. These techniques allow to continuously refine gesture models, so subconscious meanings can be handled by altering the models based on use history.

Automatic segmentation raises a problem for the recognizer algorithm as it captures movements that are not intended to be gestures. These movements are considered noise, and the machine learning models have to be aware of them. We have slightly modified the models and set a dynamic noise limit which is able to filter most of the noise.

Adaptive machine learning algorithms have the great property of getting more and more accurate as people use it. The accuracy rate has been better than 95% when at least five examples were available for each gestures.

2.4 Simplicity

The interface should be easy to set up and use in a few minutes. The system shouldn't have high expectations about the user understanding it, so it should identify and disregard misuse, and provide feedback about successful gesture recognition. For advanced users, there should be an option for

signing bad recognition and fine-tuning the gesture model.

To be able to define personalized gestures, there is a need for a simple meta-interface that helps defining the gesture sets. With the use of the adaptive machine learning techniques, it is possible to build models from only one example and iteratively refine the model when a successful recognition is made. Thus, the setup process can be finished in a minute and we can start using the interface. In the beginning, we will face low recognition rate, but the system gets better and better as we use it.

As gesture recognition will never reach 100%, it is important to inform the user about what is happening with the system. Successful recognition can be signed with auditory or vibratory feedback, so the user instantly realizes if it is needed to repeat the gesture. Advanced users can also refine the model by signing recognition mistakes with strong shaking of the sensor unit. Shaking can be easily recognized, so the system can correct its previous mistake.

3. CURRENT AND FUTURE WORK

The prototype is still in active development, but it already includes most of the functionality mentioned above. We can easily define free-form gestures by one example and use it to control the mobile phone with simulated keystrokes or to control PC applications like media players and presentation tools. It has been presented at local demo sessions and it received great interest and lots of user feedback that we could use to enhance it.

Future hardware and software tools open up new ways to solve current shortcomings of our prototype. First, the Google Android platform promises to erase strict restrictions for mobile applications, making mobiles as free as PCs nowadays, so we could reach perfect ubiquity. Second, current wireless technologies consume a lot of power which makes it hard to create small devices with wireless communication. The next version of Bluetooth, codenamed "Seattle", will support a low-energy profile for small devices. Placing the sensors to jewellery or wristwatches will allow us to use the prototype in a completely unobtrusive, hands-free way.

4. REFERENCES

- [1] D. L. Akers. Observation-based design methods for gestural user interfaces. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 1625–1628, New York, NY, USA, 2007. ACM.
- [2] F. Hofmann, P. Heyer, and G. Hommel. Velocity profile based recognition of dynamic gestures with discrete hidden Markov models. *Gesture and Sign Language in Human-Computer Interaction*, pages 81–95, 1997.
- [3] P. Korpipaa, E. Malm, T. Rantakokko, V. Kyllonen, J. Kela, J. Mantyjarvi, J. Hakkila, and I. Kansala. Customizing User Interaction in Smart Phones. *Pervasive Computing, IEEE*, 5(3):82–90, 2006.
- [4] J. Nielsen et al. *Usability Engineering*. Morgan Kaufmann, 1994.
- [5] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of american sign language. *Computer Vision and Image Understanding*, 81(3):358–384, 2001.
- [6] Y. Wu and T. S. Huang. Vision-based gesture recognition: A review. *Lecture Notes in Computer Science*, 1(1739):103–126, 1999.