Developing Gestural Input

Matthias Kranz, Stefan Freund, Paul Holleis, Albrecht Schmidt
Research Group Embedded Interaction
University of Munich
Amalienstrasse 17, 80333 Munich, Germany
{matthias, stefan, paul, albrecht}@hcilab.org

Holger Arndt
University of Technology, Munich
Theoretical Computer Science and Foundations of Artificial Intelligence
Boltzmannstrasse 3, 85748 Garching, Germany
arndt@cs.tum.edu

Abstract

In this paper, we present the Gesture Cube, a digitally augmented cube for human-computer interaction. The Gesture Cube is designed to be an unobtrusive and playful interaction device for controlling media appliances.

In an explorative user study, we analyzed user requirements; in particular we were interested in the set of meaningful gestures that users would think of. Based on this, a digitally augmented cube for controlling home appliances and computer applications has been designed. Gestures are used as the main form of input for interaction with the system.

To illustrate the feasibility of our user interface, we present two applications, a 3D visualization of the cube's state and an MP3 player. These applications were evaluated and showed that gesture interfaces can provide a playful and yet efficient means for interaction with everyday devices.

1. Introduction

Gestural input is becoming more and more important, especially for remote controls and mobile devices of any kind. The simplicity and intuitiveness of performing gestures with mobile devices makes them a suitable mode of input for a wide range of applications and appliances. This is particularly relevant for devices that can easily be manipulated with one hand. For gestural input, the form factor, size, weight, texture, and the affordances of the input device in general play a central role.

Tilting the device and other simple gestures of that type are relatively easy to analyze, when the device is only moved from a starting point in one direction and back. Examples are given in TiltText [18] or TiltType [13] and Rekimoto's work on tilt operations for small screen interfaces in [14].

In this case, identification of the gestures can be realized quite easily by simple algorithms developed from 'just looking' at the movements of the device. More complex gestures, like shaking or moving in circles, are much harder to classify. Different people perform the same gesture differently which makes recognition more difficult. The reason is the dynamic shape of these gestures, so that even a single user is not able to perform the same gesture twice in exactly the same way. When dealing with composite or other complex gestures, hard-coding the identification and recognition algorithm is not an option.

In our research, we are exploring the usage of machine learning methods, especially neural networks, to resolve this problem. In particular, we are interested in mechanisms where gestures can be *learned* during development time. In the current implementation, we restrict ourselves to basic gestures which can easily be recognized while data processing is done completely on a host PC. However, we are investigating the possibilities of pre-processing and training of a neural network directly on the embedded hardware platform.

We outline results form an explorative user study describing what gestures people would come up with using a wooden cube with 6 cm edge length. Based on these findings, a prototype system has been developed. We describe the details of the hardware and the software, as well as the tools developed. In Sec. 5, we present an MP3 player application that can be controlled via gesture input, substantiating the feasibility of our approach. We conclude by presenting a short outlook on the next steps where we plan to develop a completely self-contained device with the capabilities of learning a fixed number of arbitrary gestures for controlling media appliances and computer applications online without external support.

2. Related work

Tangible User Interfaces (TUIs) have been subject to scientific research for a long time. The initial works by Ullmer and Ishii [7] and others show the high potential of intuitive, physical interfaces for human-computer interaction. An extensive overview of tangible user interfaces developed recently can be found in Holmquist et al. [6].

Especially cubes as basic three-dimensional shapes have attracted a lot of research interest. The affordances of the cube, as described by Gibson in [5], in particular its form, size and textures, affect the way people handle the physical object. In [16], Sheridan investigates the potential of physical cubic interfaces, concentrating on handling and designing those interfaces. The potentials for using physical cubes as navigational input devices are explored by van Laerhoven et al. in [10], with the focus on the hardware itself, omitting application potentials.

Other recent works analyze the use of cubes for intuitive navigation through media content. Block et al. [2] use a cube with unobtrusively integrated sensors for visualizing the cube's state on a computer screen. The 3D visualization is overlaid with media streams, one on each side of the cube. By changing the orientation, previews of the different media streams are displayed on the screen. Placing the cube on a flat surface selects the stream associated with this face.

In [12] Nicols et al. analyze the requirements of remote controls and similar devices in their work on the 'personal universal controller'.

An interesting example of gestural interaction is the project by Koner [8]. He developed loopqoob, a set of cubes allowing adults and children whose motor skills are unsuited for traditional instruments to make music depending on their orientation. This simple and yet playful design only using static states aims to create gestural interaction to stimulate and develop intuitive and creative skills.

Kranz et al. [9] explore the potentials of a cube not only augmented with input capabilities, but with additional outputs for visual and auditive feedback in the form of displays and speakers integrated in the 'Display Cube'. The TUI presented is used as intuitive and playful generic learning platform for quizzes and tests for children. The playfulness of cubes has also been exploited in [17] for building an interface for learning appliances.

The feasibility and advantages of gestural input over current input technologies, with a focus on mobile devices, has been intensively studied by MacKenzie in [11] and compared to other forms of input, especially regarding text input on PDAs and mobile phones.

Nintendo's upcoming new game console 'Revolution' uses gestures as main form of input. Gamers have to perform gestures like aiming and throwing to steer the virtual game characters. The form factor of the input device is that of a remote control.

In contrast to the works cited above, we do not use gestures for text input only, but as a more general and generic form of input. We do not only restrict ourselves to a specific application domain, e.g. remote control of applications, but present a more general way to address the problem of gestural input for interaction. Additionally, we exploit the playfulness of the physical interaction device used for input. To show the feasibility of our approach, we performed two user studies to support our arguments. Our approach shows the importance of integrating different methods of computer science, especially machine learning, in ubiquitous computing.

3 Gestures and cubes — an explorative user study

Cubes are especially interesting as tangible user interfaces as they constitute very basic three-dimensional objects, but nevertheless offer a wide range of possible manipulations. From a social point of view, people know that there are potentially different actions or meanings associated with each of the six faces of a cube. Playing dice games has been part of many cultures for several hundred years and people are familiar with handling cubes from childhood on. We therefore conducted a user study to find out which gestures the cube could be used for, in addition to the apparent manipulations of translation and rotation.

To address the question of how people would use cubes as interfaces, we conducted semi-structured open-ended interviews with 10 people, 5 men and 5 women, aged between 20 and 50 years. All participants had at least basic knowledge of using computers and traditional input devices.

The first task for the participants was to describe a set of intuitive gestures or natural meaningful movements using the cube. They had to rank how important the gestures are to them. The gestures that were named most often were 'turning' (86%), 'rolling' (58%), 'shaking' (49%), 'throwing' (35%) and 'shifting' (30%).

The second task was to reflect on gestures that could be used to interact with a computer system. The users were handed a wooden cube of 6 cm edge length, see Fig. 1, and asked to imagine controlling home appliances, computer applications or games with gestures performed with it and to rate the importance of these gestures. The gestures that were now named most often were 'turning' (87%), 'shifting' (54%), 'positioning' (36%), 'orientation' (32%) and 'shaking' (28%).

Finally, the participants were asked for which appliance or computer application the cube would be an appropriate input device. Common answers included '3D positioning for graphic applications', '3D control for flight simulators' and simple computer games such as 'jump and run games' and '(3D) Tetris'. But numerous other examples for using the cube as input device were given, e.g. 'text adventures', 'ICQ status control' and 'vocabulary trainer', or more generally 'edutainment and e-learning' appliances.

We were surprised how many different gestures and applications the participants came up with. They stated that they had a playful experience performing the gestures and actually would really like to use it. The difference between general gestures and gestures for controlling a specific application is interesting and shows the importance of gestural human-computer interaction.

Summarizing the results of our study, we found out that a device for gestural input should be easy to handle as far as size and weight are concerned, and flexible enough to deal with a wide range of different gestures. The device must not only be able to recognize its orientation but should also react on dynamical movements. In the next section we present our prototype which we think is universal enough to serve as a basic human-computer interaction device for gestural input.

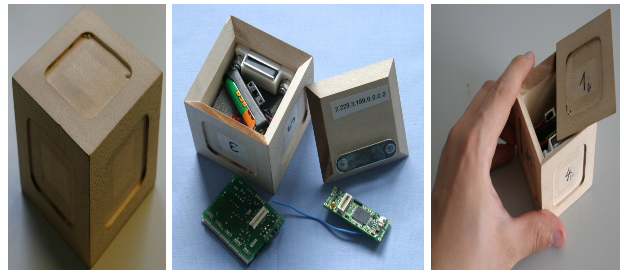
4 Prototype

The prototype consists of the hardware included in the cube, the communication architecture, the software running on the micro controller in the cube and the software on a host PC used for data analysis and gesture recognition.

The cube itself is a hollow wood block with 6 cm edge length. The size was a deliberate choice to have an object that is small enough to smoothly integrate into the environment, but also large enough to be easy to grasp, hold and manipulate. A small rectangular portion of each side was slightly carved into the surface such that images or different surfaces could be mounted without changing the form factor. A picture of the opened Gesture Cube, showing its internals, is depicted in Fig. 1

4.1 Hardware and communication

To equip the cube with a microcontroller system we used a Particle computer [3], which will serve as the basis for our implementation [4]Ṫhe hardware comprises a communication board integrating a microcontroller, a radio transceiver, and 512 kB of additional Flash memory. It can run with a single 1.2V battery and consumes 40mA



(a) Wooden Cube (b) Gesture Cube Hardware (c) Cube Handling

Figure 1. The Gesture Cube is a wooden cube of 6 cm edge length. The Particle computer platform is embedded unobtrusively in the cube. The device can be opened by removing one side which is held closed using a strong magnet. The faces can be labeled to ease visually recognizing static states, e.g. which side is on top.

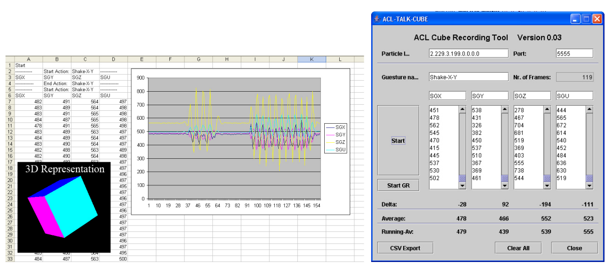
on average. The Particle computer in the cube is extended with a sensor board. The sensors used are two two-axes accelerometers (ADXL311JE from Analog Devices) which are orthogonally mounted inside the Gesture Cube. The acceleration sensors detect acceleration as well as gravity which can be exploited for static state detection, i.e. which side is on top. Thus, they can be used for both determining the cube's current orientation and the gestures performed by the user. Gestures can be performed at any time with any side on top. The sensor board hardware is described in detail at [15]. The Particle computer including the sensor board measures 35 mm x 48 mm x 10 mm.

Radio-frequency (RF) communication of the Particle is based on AwareCon, a wireless ad-hoc protocol, which is described in detail in [1]. Because the data format of Particle packets (ACL packets) is not suitable for IP based communication, special devices are used to convert the format from AwareCon to IP and back. On the IP based devices, the network runs on top of UDP/IP. On the PC side, we use a library written in C to access the transferred packets, as well as to transmit information to the Particle. Since Particles are also able to communicate ad-hoc, applications for which no PC-connection is necessary, could be realized without additional infrastructure.

4.2 Software

The program on the Particle is responsible for reading sensor values and storing the last n of them. If the number of data gathered is large enough to be analyzed (n is equal to a predefined frame size), the current orientation of the cube in 3D space is calculated. If the cube has been in a stable state for the whole time for which the last frame size samples have been measured, this information is immediately sent. If, on the other hand, the cube has been moved, the data is pre-processed and analyzed. If one of the trained gestures has been recognized this is sent to, e.g., an MP3 player (see Sec. 5). A visualization of the program flow running on the Particle in the Gesture Cube shown in Fig. 1 is shown in Fig. 3.

We implemented a 3D visualization application as a basic tool to get a better understanding of the sensor values. To assure that the mapping between the state of the physical device and our digital model of the device is congruent, we implemented a virtual view of the cube. The software is built on top of Java 3D. A screen shot of the application can be seen in Fig. 2. Having visual feedback to inspect and debug algorithms for detecting gestures was of great importance in the developing process. For early development and logging, we have developed a JAVA tool to record data from all four sensors in real-time.



(a) Sensor Recording Tool (b) Sensor Analyzer Tool

Figure 2. (a) Screen dump of the acceleration sensor data recording tool. Four axes are monitored (SGX, SGY, SGZ and SGU). Simultaneously a delta to the last sensor value, an overall average and a running average is calculated. (b) Screen shot of the acceleration sensor data used for quick analysis in Microsoft Excel during the development. In the foreground runs the 3D visualization application to show the current orientation of the Gesture Cube. The pattern currently analyzed is a shaking followed by a second, longer shaking.

For calibration purposes and gesture detection, overall and running averages over the last 10,000 values are calculated for each sensor. Data can be exported in CSV format for analysis and further processing in the early stage of development.

This tool has been used in an explorative user study to acquire data for further experiments. Initially, 10 different gestures with 10 examples each have been recorded. These gestures included shifting, shaking, turning and spinning the cube. The recorded results of a 'shaking' gesture can be seen in Fig. 2. Beginning and end can easily be recognized since the noise level of an unexcited sensor is very low. The data logger application and a three dimensional virtual representation of the cube is depicted in Fig. 2.

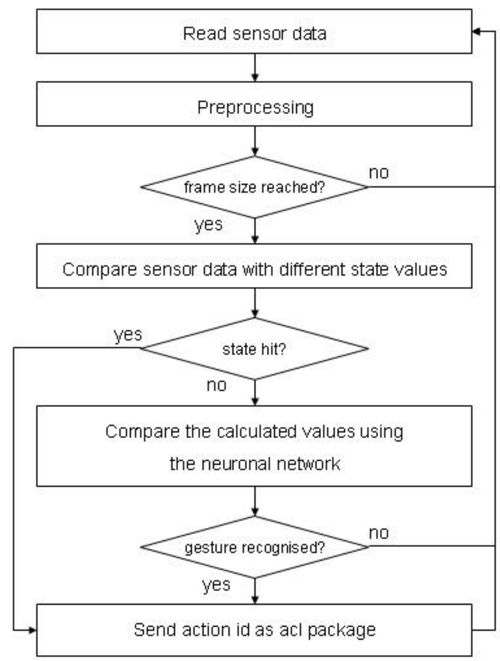


Figure 3. Program flow of the Gesture Cube program running on the Particle Computer platform. The raw sensor data is read and pre-processed. Then a static state detection takes place to determine the orientation of the cube (see Tab. 1). After that, gesture recognition is performed. If a gesture has been successfully identified, it is transmitted via RF to a potential listener application using the standard package format of Particles (ACL).

5 MP3 player control with the gesture cube

We implemented an MP3 player control on the Gesture Cube. For our prototypical implementation, it currently relies on a computer running a winamp-like MP3 player realized in Java using the Java Media Framework (JMF). We envision to use the device later directly with the digital home equipment. This would only required including a small corresponding receiver control into the home appliance. The MP3 player application allows the playful use of the Gesture Cube interface.

We conducted another user study to find out which gestures people would want to use for controlling an MP3 player. The participants of this study were completely disjunct from the explorative study presented earlier in this paper. We interviewed 15 participant, 10 men and 5 women with at least basic knowledge of computers and traditional input devices. The first task was to *invent* gestures for the most important functions of a MP3 player. They participants were then asked to name one gesture for each basic function they could think of.

For actions like 'PLAY' and 'STOP' or 'NEXT' and 'PREVIOUS', most participants thought of static positions or short, rather abrupt actions like shifting. Continuous gestures, like spinning the cube around, were preferred for varying actions like changing the volume. After evaluating the participants answers, the following states and gestures were chosen to be controls for the MP3 player application. The gestures are summarized together with their associated functions in table 1.

Based on these findings, the system has been implemented, including a graphical user interface. In Fig. 4, the GUI is depicted and you can see a user controlling the volume of his MP3 player application while reading the booklet of the currently playing album. His attention is not drawn away from his reading as the Gesture Cube does not require high mental load for controlling.

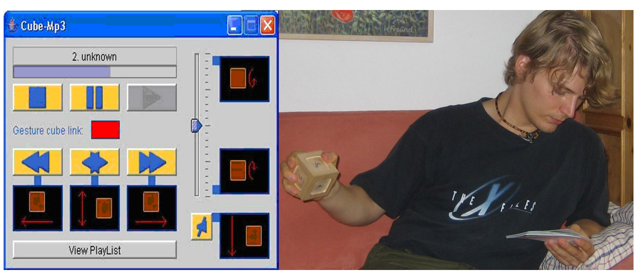


Figure 4. The Gesture Cube is used for changing the current song. No explicit attention is required as the gestures can be performed with minimal attention and cognitive load towards the input device.

6 Conclusion and future work

We showed that cubes can be used as meaningful input devices to control different types of applications and showed that gestures are an important means of input for mobile devices. These findings were supported by two user studies.

Play	State: side 1,3,6 or 4 facing upwards
Pause	State: side 5 facing upwards
Stop	State: side 2 facing upwards
Next	Gesture: shift to the left
Previous	Gesture: shift to the right
Random	Gesture: shaking up/down
Volume up	Gesture: spinning forwards
Volume down	Gesture: spinning backwards
Mute	Gesture: shift down

Table 1. Gestures for the MP3 player. The gestures and states used were derived from the user study presented in section 5

Currently we are working into a completely self-contained device capable of learning a fixed number of arbitrary gestures, with no need for additional infrastructure or off-line training, which can be used for controlling media appliances and computer applications.

As a next step, we would like to extend the number of recognizable gestures and include more complex ones like drawing figures into the air. How we will tackle the recognition problem is still an open issue. We might consider Hidden-Markov chains or recurrent neural networks to achieve this goal. Currently, an open-source toolkit for capturing, training and analyzing structured data is under development for exactly this purpose.

Additional user studies are planned to investigate potential gestures for controlling a broader range of applications and appliances.

Our objective is to derive an intelligent device capable of learning and classifying gestures suitable for a wide range of applications, not restricting our work to cubic interfaces, but to tangible user interfaces in general.

7. Acknowledgements

The work has been conducted in the context of the research project Embedded Interaction ('Eingebettete Interaktion') and was funded by the DFG ('Deutsche Forschungsgemeinschaft').

References

[1] M. Beigl, A. Krohn, T. Zimmer, C. Decker, and P. Robinson. AwareCon: Situation Aware Context Communication. In *Proceedings of Ubicomp 2003*, Seattle, USA, October 2003.

[2] F. Block, A. Schmidt, N. Villar, and H.-W. Gellersen. Towards a playful user interface for home entertainment systems. In *European Symposium on Ambient Intelligence*

(EUSAI 2004), Springer LNCS 3295, pages pp207–217. Springer, 2004.

[3] C. Decker, A. Krohn, M. Beigl, and T. Zimmer. The particle computer system. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 443–448, April 2005.

[4] H.-W. Gellersen, G. Kortuem, M. Beigl, and A. Schmidt. Physical Prototyping with Smart-Its. *IEEE Pervasive Computing Magazine*, 3(3):74–82, July–September 2004.

[5] J. Gibson. The theory of affordances. In *Perceiving, Acting, and Knowing*. Lawrence Erlbaum Associates, 1977.

[6] L. E. Holmquist, A. Schmidt, and B. Ullmer. Tangible interfaces in perspective: Guest editors' introduction. *Personal Ubiquitous Computing*, 8(5):291–293, 2004.

[7] H. Ishii and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241. ACM Press, 1997.

[8] M. Konar. loopqoob. `http://www.muratnkonar.com/id/loopqoob/`, 2003.

[9] M. Kranz, D. Schmidt, P. Holleis, and A. Schmidt. A display cube as tangible user interface. In *In Adjunct Proceedings of the Seventh Internation Conference on Ubiquitous Computing (Demo 22)*, September 2005.

[10] K. V. Laerhoven, N. Villar, A. Schmidt, G. Kortuem, and H. Gellersen. Using an autonomous cube for basic navigation and input. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 203–210, New York, NY, USA, 2003. ACM Press.

[11] S. MacKenzie and W. Soukoreff. *Text entry for mobile computing: Models and methods, theory and practice*. 2002.

[12] J. Nichols, B. A. Myers, M. Higgins, J. Hughes, T. K. Harris, R. Rosenfeld, and M. Pignol. Generating remote control interfaces for complex appliances. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 161–170, New York, NY, USA, 2002. ACM Press.

[13] K. Partridge, S. Chatterjee, V. Sazawal, G. Borriello, and R. Want. Tilttype: accelerometer-supported text entry for very small devices. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 201–204, New York, NY, USA, 2002. ACM Press.

[14] J. Rekimoto. Tilting operations for small screen interfaces. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 167–168, New York, NY, USA, 1996. ACM Press.

[15] Research Group Embedded Interaction. Particle Display Add-On Board Website. `http://www.hcilab.org/projects/particles/particles-display-add-on.htm`, 2005.

[16] J. Sheridan, B. Short, K. V. Laerhoven, N. Villar, and G. Kortuem. Exploring cube affordance: Towards a classification of non-verbal dynamics of physical interfaces for wearable computing. In *In Proceedings of the IEE Eurowearable 2003*, pages pp 113–118. IEE Press, 2003.

[17] L. Terrenghi, M. Kranz, P. Holleis, and A. Schmidt. A cube to learn: a tangible user interface for the design of a learning appliance. In *3AD: Third International Conference on Appliance Design*, pages pp. 82–86, June 2005.

[18] D. Wigdor and R. Balakrishnan. Tilttext: using tilt for text input to mobile phones. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 81–90. ACM Press, 2003.