Technology Support for Analyzing User Interactions to Create User-Centered Interactions

Dirk Burkhardt, Kawa Nazemi, Nadeem Bhatti, and Christoph Hornung

Fraunhofer Institute for Computer Graphics Research (IGD), Fraunhoferstrasse 5, 64283 Darmstadt, Germany
{Dirk.Burkhardt,Kawa.Nazemi,Nadeem.Bhatti, Christoph.Hornung}@igd.fraunhofer.de

Abstract. Alternative interaction devices become more important in the communication between users and computers. Parallel graphical User Interfaces underlay a continuous development and research. But today does no adequate connection exist between these both aspects. So if a developer wants to provide an alternative access over more intuitive interaction devices, he has to implement this interaction-possibility on his own by regarding the users perception. A better way to avoid this time-consuming development-process is presented in this paper. This method can easy implement by a developer and users get the possibility to interact on intuitive way.

Keywords: User-Centered Interactions, Human-Computer-Interaction, gesture recognition.

1 Introduction

Alternative interaction devices get nowadays more importance in the communication between human and computer. Different game pad developer established such interaction devices with a more intuitive way of usage. One of the most common devices is the WiiMote from Nintendo. The benefit of such devices is the support and orientation on humans' behavior and acting. Nintendo Wii's success is an indicator for the trend and the necessity of such devices. The low cost controller play a key role for its mass circulation.

Another trend is the research and development of Graphical User Interfaces (GUI). Innovative GUIs visualize information, for example by using semantic technologies on the web and, herewith, reduce the cognitive overload of the processes for users, which require gathering information.

In the last years, both processes, the research on graphical visualizations as well as the research in alternative interaction techniques, have underlain a rapidly development. But, nevertheless, up to now there exists no adequate connection between them. One of the goals of this paper is to develop a method for using alternative and more intuitive interaction-devices e.g. the WiiMote in web-based graphical visualization. Within the range of this development, there are further aspects, which have to be regarded. Intuitive use is a subjective perception of a person [1], so the user's perceptions must flow into the development-process. But the considering of the users individual perception is time-consuming, which the most developers cannot or does not want to offer.

C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2009, LNCS 5614, pp. 3–12, 2009.

In this paper a technique will be presented, how these problems can easily be solved. The technique allows developers to use alternative controllers on an easy way and provide them an easy to use and easy to implement programming interface, so that they can pre-configure it for the use in web-applications. To realize a user-centered design, the concept will orientate on the Process Model of User-Centered Software Design Process [1]. Later a user is able to extend the pre-configured interface by his personal perceptive actions, if it is necessary. For evaluation issues the system is developed for Nintendo's Wii.

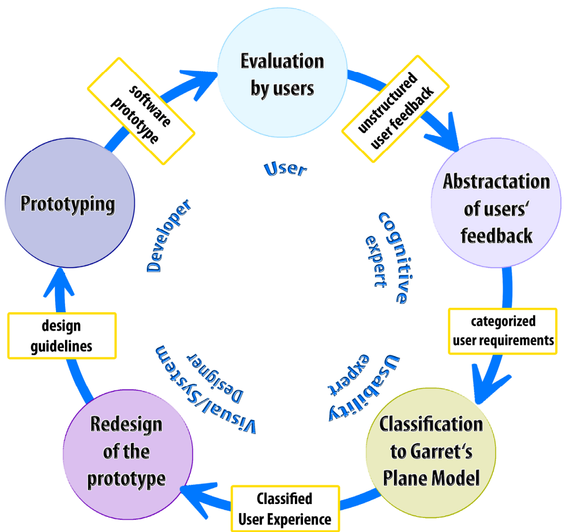

Fig. 1. Process Model of User-Centered Software Design [1]

2 Related Work

Concerning adaptive user interfaces there exist several approaches. The most common research there is focused on gesture recognition techniques or the specific implementation of such techniques on new input device. Less weighted are surrounding aspects like culture specific interactions or the support from developers of user-centered applications. The relation between a cultural background and how it influences the execution of a gesture is described in [2]. Rehm et al. concentrate on the districting of the properties power, speed and spatial extend with which they classify the different cultures. During a recognition-process these properties will help to identify a gesture. By the use of low-level gestures e.g. a movement to the right sight such influences can neglect, because the shape and at the end a gesture for specific common instructions within applications is similar.

2.1 WiiMote-Based Recognition Systems

In our prototypical implementation Wii-Controller is used. Two most prominent libraries Wiinput [3] and WiiGee [4] exist for Wii-Controller to realize a gesture-recognition. Both of them are not able to be used within web-applications and cannot be configured individually for or by the user and they do not support a developer by implementing an alternative interaction controller. But, from a recognition point of view, they provide a similar functionality.

The first is Wiinput, which is developed by Polak M. [3] and implements a type of instance-base learning. Wiinput is primary designed for games and is trained only once by the developers. Users have no possibility to commit their perception. Another problem is the recognition-process and when the gesture recognition should be started. In the existing version a button on the controller must be released to start the recognition.

The second recognition program is WiiGee, which is developed by Popinga B. [4] and is one of the most powerful programs for recognizing gestures with the WiiMote. It uses an implementation of a Hidden-Markov-Model and achieves well recognition results. The recognition-process also has to be initialized over the release of a button. WiiGee is designed for small programs but not for usage on web-applications. There is a way to teach the system, but it is not possible to let a user commit his perception of intuitive gesture. Only a developer can train the system by his own understanding of rational gestures. In later times it is hard to extend the system with additional gesture-executions by a developer, too.

2.2 Methods and Techniques for Recognizing Gestures

For gesture-recognition at all, a well-working method for gesture-recognition is necessary to provide a useful interface to a user to interact in an intuitive way. Many approaches were developed in the last years, to realize a accurate gesture recognition on the one hand and fast recognition on the other hand, so that an interface will be intuitively usable. The most common methods, which are described in LaViola et al. [5], can be divided into 3 classes:

1. Feature Extraction, Statistics and Models
2. Learning Algorithms
3. Miscellaneous Techniques

To the first class belong all methods, which are working on the base of mathematical concepts like models, statistics or indirectly by the extractions of special mathematical features. The most used methods of this category are Template Matching, Dynamic Time Warping, Feature Extraction and Active Shape Model.

Template Matching and Active Shape are working on a similar way. To identify a gesture with Template Matching, the data will be compared with a template that borders the specific gesture properties. When using the Active Shape Model, a shape is given, which will scale down up to the gesture will fits best and, if the difference is in a certain range, the gesture is recognized.

Dynamic Time Warping has a pool of known gesture-graphs. To identify a performed gesture the data-graph will be compared with the known gesture-graphs. The searched gesture is that with the smallest Euclidean distance between the gesture and data-graph.

Feature Extraction works by finding significant information within the gestures, like speed, duration or acceleration. A performed gesture will be indicated, if the extracted information is similar to one of known gestures.

Table 1. List of common methods for Feature Extraction, Statistics and Models

Method	Vision	Glove	Accuracy	Previous Work	Implementation complexity
Template Matching	Yes	Yes	High	Extensive	Simple
Active Shape Model	Yes	No	Low	Minimal	Simple
Dynamic Time Warping	No	Yes	High	Moderate	Moderate
Feature Extraction	No	Yes	High	Moderate	Moderate

The second class Learning Algorithms contains methods that are getting better accuracies after they have learned more interaction information. So it is common that gestures will be taught as many times as possible to the system, to attain well recognition results. Common methods are Neuronal Networks, Markov Models and Instance-Based Learning. Often they are computational intensive during their recognition or learning phase, so that high performance computers are required.

Neural Networks emulate the functionality of a human brain with its neurons and the links between them. In comparison to Markov Models, a re-training is necessary if a new gesture is learned. Markov Models are working with probabilistic models, often a kind of Hidden Markov Models is used where the probabilistic model is encapsulated as a separate hidden part.

Instance-Based Learning is different from neutral networks and Markov models. While the latter ones store the information after they are assimilated, in instance-based learning the information is saved uncalculatedly, so that the uncompressed information has to be compared with the data of a performed gesture, which has to recognized.

Table 2. List of common methods for Learning Algorithms

Method	Vision	Glove	Accuracy	Previous Work	Implementation complexity
Neuronal Networks	Yes	Yes	High	Extensive	Extensive
Markov Models	Yes	Yes	High	Extensive	Moderate
Instance-Based Learning	Yes	Yes	High	Minimal	Simple

Some special kinds for gesture recognition are summarized in the third class, Miscellaneous Techniques. These techniques have very different rudiments in comparison to the previously called methods. Common examples are The Linguistic Approach and Appearance-Based Motion Analysis.

By the Linguistic Approach, gestures are mapped to a grammar, which specifies a series of token and production rules. Humans are able to recognize actions from images with very low resolution and with little or almost no information about the three-dimensional structure of a scene. The Appearance-Based Motion Analysis is based on these facts.

Table 3. List of common methods for Miscellaneous Techniques

Method	Vision	Glove	Accuracy	Previous Work	Implementation complexity
Linguistic Approach	Yes	Yes	Moderate	Minimal	Simple
Appearance-Based Motion Analysis	Yes	Yes	High	Minimal	Simple

3 WiiMote as a Representative Intuitive Interaction Device

The Nintendo Wii game-console came on the market in 2006. Within a short time it had an amazing success which can be read from the selling chart (Fig. 2). In the year 2008 the Wii console broke as first game console the border of 10 million sales within one year [6].

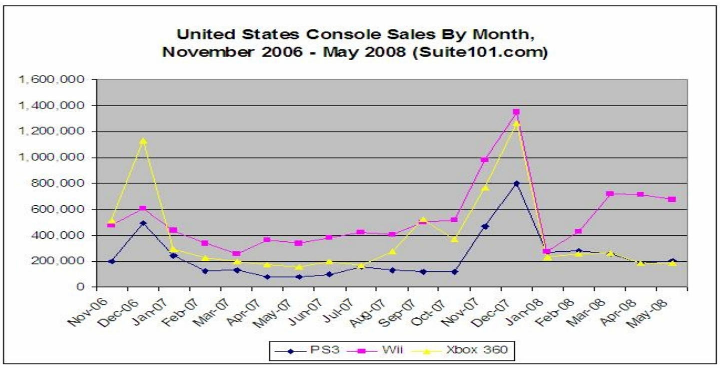

Fig. 2. Console Sales of the 3 most common Game Consoles, by Month, since the Wii's release in November 2006 [7]

The Wii profits at most by his new innovative input-controller, the WiiMote. This controller is a good example for the above-mentioned development process of alternative interaction-devices.

The WiiMote is designed from the shape, use and feel like a TV-Remote [8]. So the first of 3 ways of using this controller is like a Remote over the Buttons. The second way is by using the Infrared-Sensors, so that it can be used as pointing-device – equal to a classical pointer. But the most popular way especially in Wii-Games is the functionality of the accelerators. With these accelerators the WiiMote is able to know its location and the movements within a 3D-room. This way it is suitable for an interaction with gestures. The accelerators generally have sensors for all 3 directions (X, Y, Z-Axis) and, additionally, can calculate the rotation around these 3 axis. With this possibility this controller can defined as an input device with a six degree of freedom.

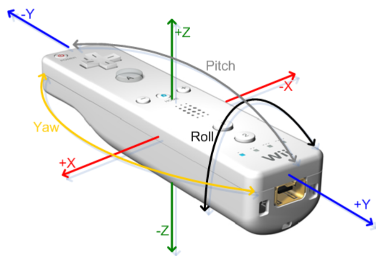

Fig. 3. The Wii Remote as three dimensional input device and the 6 degree of freedom realized by the integrated accelerators. With these accelerators this controller is suitable as controller to perform gestures.

4 Concept of a Technique to Realize User-Centred Interaction

To create a User-Centred interface, the user´s needs have to be regarded, for that fact the key-concept during the implementation was the Process Model of User-Centred Software Design (Fig. 1).

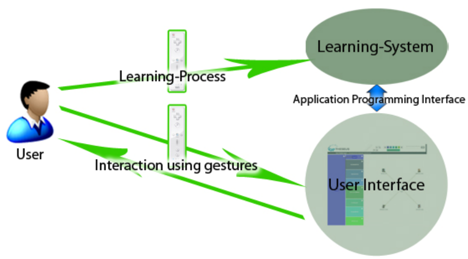

Fig. 4. Process of learning interactions: The Learning Process will realized within the Learning-Tool and interaction works with an integrated API, that uses the learned functions for recognizing the gestures

The goal of our technique is to realize gesture-recognition witch works on an abstract level, so that it can later also be used with other controllers than the WiiMote. Another aspect is the configuration of the interface. So of course the developer must define the required commands that, later on, will be linked with the several gestures. As a consequence thereof the developer must be able to define default-gestures. But later on, a user must also get the possibility to permit the own perceptions of intuitive gestures for the several commands.

These facts make it essential to divide the problem in 2 elements. The first is the Learning-Process with which a developer and a user can set gestures on an easy way. The second element is the recognition application programming interface (API) that provides an easy to implement interface to an existing application. This API uses the learned function from the Learning-Process.

4.1 The Learning-Process

To support developers by using alternative intuitive devices the focus does not lie on the recognition, even although it is relevant. So we decided to use an implementation of Template Matching method, which is useful for a high accuracy during the gesture recognition in real-time on the one hand and because of the missing of a big probabilistic model like by Markov Models it can be implemented without difficulties on the other hand.

Because of the autonomous handling in storing and analysing of defined gestures in Template Matching, gestures can generally be added, dropped and edited without risking a bad accuracy. Typical use-cases for the adding of gestures are the learning of new commands or the perception of new gestures for already existing commands.

Another advantage is, that it can easily be adapted to other interaction devices which work with accelerators like a modern mobile phone (e.g. Apple's iPhone) etc. Alternative devices need only an equal interface to the sensors-information.

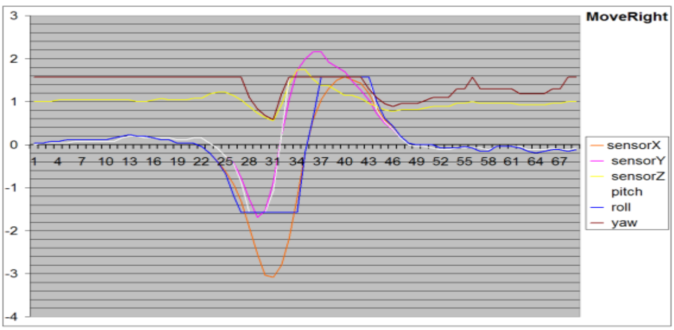

Fig. 5. Example graph of the accelerators of the WiiMote during a performed gesture (a side-move to the right)

To provide an easy way of teaching new gestures, the "WiiGesture Learning" tool (see below) generates the templates autonomously. This happened after the demonstration of a gesture to the system with the WiiMote, after that the calculation-process starts and generates a template. The template consists of equal properties that can find within the repetitions. As property the turning points of the different functions of the sensors are defined. After generating the template-functions, they will be saved in a configuration file, so that they can be published with API-using application.

The screenshot (Fig. 6) shows the elements of the Learning-Tool "WiiGesture-Learning". On the left-top side the status of the WiiMote will be shown. On the top a graph by three physically accelerated and the functions by representing the rotation around these axis. The bar-chart and the list on the right hand side represent the extracted properties of a performed gesture. The text-field on the left side is for saving new gestures and list underneath shows the actual known gestures.

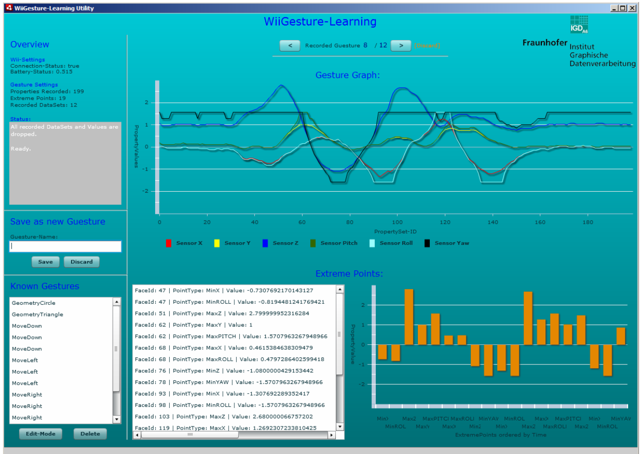

Fig. 6. Screenshot of the Learning-Tool “WiiGesture-Learning”

4.2 The Gesture-Interaction-Process

By the use of an alternative interaction-device an application has to integrate the “Gesture-API”. The API interprets performed gestures to commands and provides them to the application. Herewith, the developer has to specify the needed commands only. The rest of the work is overtaken by the API, which will load the configuration file on startup and read the previously learned template-functions. After that all performed gestures over the WiiMote will be compared with the templates to recognize gestures.

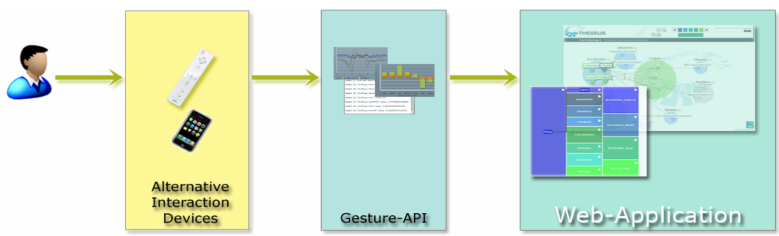

Fig. 7. Concept of the Gesture-Interaction-Process by using Gesture-API within a Web-Application

5 Evaluation

For the evaluation purpose, the participants used a preconfigured system by a developer. So the commands are defined and also a small set of possible gestures for the commands. After that the users have to activate 16 commands in a special to observed order. The average results of the users are shown in Fig. 8 in the category of “Predefined System”.

After this step, the users train the system with gestures, which fit with the individual personal perception of intuitive gestures for existing commands. Then the users have to activate the same 16 commands in the same observed order. The average results are presented in Fig 8 in the category "User-customized System".

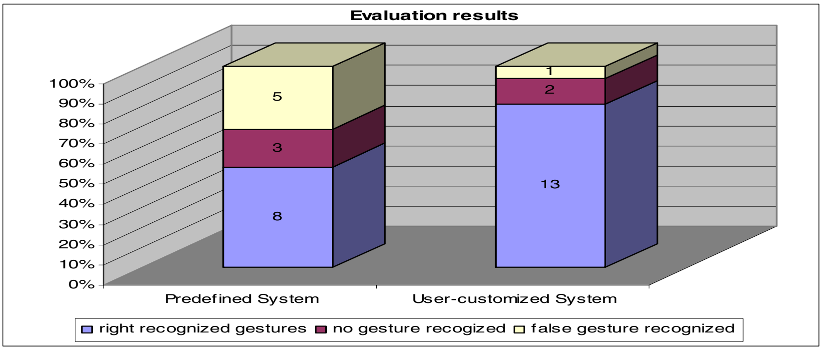

Fig. 8. Evaluation results of users before and after customizing the system with own intuitive perceptions

The recognition-results become better, after the system was extended by the personal perception of the users. The error rate of not or false detected gestures fell down from nearly 50% to less than 20%.

To evaluate the easy integration in existing web-application to support developers, the API was included in some existing applications to provide the possibility to interact via gestures. Only a small number of additional source-code was necessary, so it was no big deal for the developers. One of the applications is the seMap-Visualization [9] that provides a new form for presenting semantic information.

6 Conclusion and Discussion

In this paper a technique was shown, how developers can be supported by providing programs that allow the easier use of alternative interaction-devices and how personal perception of intuitive metaphors of users can be used. The Evaluation proved the described technique and shows that also better recognition-results can be achieved, if users get the possibility to commit there perception of intuitive gestures. In later works the developed tools can be extended, so that both elements are combined in only one API. A further learning after that is possible during the normal interaction.

The innovative point of this paper is the support of developers, where they are now able to provide an intuitive interface to the users which can also customized by users to permit their own perception for intuitive gestures. Furthermore the technique is encapsulated in an Application Programming Interface, so that the integration in existing applications takes only less time.

References

1. Nazemi, K., Hornung, C.: Intuitive Authoring on Web: a User-Centered Software Design Approach. In: Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008, pp. 1440–1448. AACE, Vienna (2008)
2. Poppinga, B.: Beschleunigungsbasierte 3D-Gestenerkennung mit dem Wii-Controller, Universität Oldenburg (2007)
3. Polak, M.: Entwicklung einer Gestenerkennung auf Basis des Beschleunigungssensors in der Wii Remote, Diplom-Thesis, Fachhochschule Köln – Campus Gummersbach: Fakultät für Informatik und Ingenieurwissenschaften (2008)
4. Poppinga, B.: Beschleunigungsbasierte 3D-Gestenerkennung mit dem Wii-Controller, Universität Oldenburg (2007)
5. LaViola Jr., Joseph J.: A Survey of Hand Posture and Gesture Recognition Techniques and Technology, Brown University: Department of Computer Science, Providence (1999)
6. Nintendo: Wii & Nintendo DS Set New Industry Records (2009), http://www.nintendo.com/whatsnew/detail/3vrEIL4gq33i0RmNbYJ9i97j65EwSpVI
7. Jonathan Coley: Video Game Console Sales and Top-Selling Video Games (2008), http://videoonlinegames.suite101.com/article.cfm/may_2008_video_game_sales_charts
8. Iwata, Satoru: Iwata Asks: Wii Remote, Part 1-3 (2006), http://wii.nintendo-europe.com/164.html
9. Nazemi, K., Breyer, M., Hornung, C.: SeMap: A Concept for the Visualization of Semantics as Maps In: HCI-International (2009)
10. Keogh, E.J., Pazzani, M.J.: Derivative Dynamic Time Warping. In: Proc. of the 1st SIAM Int. Conf. on Data Mining (2001)
11. Ekman, P., Friesen, W.V.: The repertoire of nonverbal behavior: categories, origins, usage and coding. Semiotica 1 (1969)
12. Grudin, J.: Utility and Usability: Research Issues and Development Contexts. Interacting with Computers 4(2), 209–217 (1992)
13. Hassenzahl, M., Sandweg, N.: From mental effort to perceived usability: transforming experiences into summary assessments. In: CHI 2004: CHI 2004 extended abstracts on Human factors in computing systems. ACM, New York (2004)
14. Kendon, A.: Current Issues in the Study of Gesture. In: Nespolous, J.L., Perron, P., Lecours, A.R. (Hrsg.) Laurence Erlbaum Associates (Veranst.): The Biological Foundations of Gestures: Motor and Semiotic Aspects, Hillsdale, London, pp. 23–48 (1986)
15. Myers, B.A., Rosson, M.B.: Survey on user interface programming. In: CHI 1992: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 195–202. ACM, New York (1992)
16. Pavlovi’c, V.I., Sharma, R., Huang, T.S.: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 677–695 (1997)
17. Watanabe, T., Lee, C.-W., Tsukamoto, A., Yachida, M.: Real-Time Gesture Recognition Using Maskable Template Model. In: ICMCS, pp. 341–348 (1996)

Citation

Burkhardt, Dirk; Nazemi, Kawa; Bhatti, Nadeem; Hornung, Christoph:
Technology Support for Analyzing User Interactions to Create User-Centered Interactions.
In: HCI International 2009. Proceedings and Posters [DVD-ROM] : With 10 further Associated Conferences.
Berlin, Heidelberg, New York : Springer, 2009, LNCS 5614, pp. 3-12
(Lecture Notes in Computer Science (LNCS)).