ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS

Volume LXI 297 Number 7, 2013
http://dx.doi.org/10.11118/actaun201361072655

MAINSTREAMING GESTURE BASED INTERFACES

David Procházka, Jaromír Landa, Tomáš Koubek, Vít Ondroušek

Received: April 11, 2013

Abstract

PROCHÁZKA DAVID, LANDA JAROMÍR, KOUBEK TOMÁŠ, ONDROUŠEK VÍT: *Mainstreaming gesture based interfaces.* Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, 2013, LXI, No. 7, pp. 2655–2660

Gestures are a common way of interaction with mobile devices. They emerged especially with the iPhone production. Gestures in currently used devices are usually based on the original gestures presented by Apple in its iOS (iPhone Operating System). Therefore, there is a wide agreement on the mobile gesture design. In last years, it is possible to see experiments with gesture usage also in the other areas of consumer electronics and computers. The examples can include televisions, large projections etc. These gestures can be marked as spatial or 3D gestures. They are connected with a natural 3D environment rather than with a flat 2D screen. Nevertheless, it is hard to find a comparable design agreement within the spatial gestures. Various projects are based on completely different gesture sets. This situation is confusing for their users and slows down spatial gesture adoption.
This paper is focused on the standardization of spatial gestures. The review of projects focused on spatial gesture usage is provided in the first part. The main emphasis is placed on the usability point-of-view. On the basis of our analysis, we argue that the usability is the key issue enabling the wide adoption. The mobile gesture emergence was possible easily because the iPhone gestures were natural. Therefore, it was not necessary to learn them.
The design and implementation of our presentation software, which is controlled by gestures, is outlined in the second part of the paper. Furthermore, the usability testing results are provided as well. We have tested our application on a group of users not instructed in the implemented gestures design. These results were compared with the other ones, obtained with our original implementation. The evaluation can be used as the basis for implementation of similar projects.

spatial augmented reality, Kinect, usability, gesture

1 INTRODUCTION

Without hesitation, graphical user interfaces (GUI) for mobile devices can be divided in two halves: before the first iPhone and after it. Although the first iPhone had many flaws, from the usability point-of-view, it started a true revolution. Virtually all user interfaces before the iPhone were based on the idea of a simplified desktop environment. This tendency to transfer well known desktop graphical features to a mobile screen was obvious especially in case of Windows Mobile devices (see e.g. file manager in Fig. 1, left). Even the mobile version of Windows comprised a huge amount of small widgets (checkboxes, scrollbars etc.). Hence, some kind of stylus was necessary to operate the system effectively. Nonetheless, some companies were able to present a GUI adjusted clearly for small screens of mobile devices. A well-known example is the PalmOS (Fig. 1, center). The PalmOS (for many years a leader in the smart devices market) used a simple and very effective GUI. Most of common operations were possible even without the stylus. However, especially its character input was based on stylus usage. Apple obviously based their first iOS design on PalmOS (compare e.g. application launchers on center and right part of Fig. 1), nonetheless they (at that time controversially) decided to remove the stylus and focus on a more natural way of communication – gestures.

Any substantial redesign of a common application with many users or a standard input method is always a very complex task. For instance, even the

1: *Mobile operating systems. From left: file manager in Windows Mobile, PalmOS, iOS.*

incomparably smaller Microsoft Office or Windows GUI redesign brings always resistance from many users that are used to the older design. If we simplify the problem substantially, we can say that the gain from the new GUI/input method must be really significant (not just some incremental update), because it must outweigh the necessity to learn the new environment. Therefore, the decision to present a new platform with an up to that time unknown input mechanism was a bold step from a management point-of-view. Now, we can say a very successful one. The gestures were so simple and natural that the users adopted them literally from day to day. Moreover, other companies (e.g. Nokia, Google and later even Microsoft) based their new mobile operating systems on this idea of stylus-less control. Gestures can be found also in many other applications – touch walls, large projections, cars etc.

This article is focused on the problem of gesture-controlled large screen projections. Particularly, how to design and implement a user-friendly control mechanism for presentation software in case when the user is not able to touch the projection surface directly. Our experimental implementation of such a projection is outlined as well as its comparison with some selected frequently used approaches. Further, the analysis of user behavior is summarized.

2 COMPUTER VISION-BASED METHODS USED FOR PROJECTION CONTROL

On the beginning, it is necessary to emphasize that we are dealing with a kind of projection where there is a substantial gap between the user and the projection screen. Therefore, the user is not able to touch the surface of the projection directly. Hence, the projection is not a large touch screen from the user point-of-view.

Generally, there are three main approaches. It is possible to designate them as a widget-based solution, gesture-based solution and device-based solution. The widget-based solution is a very common principle transferred from different information kiosks, mobile devices etc. Projection is taken as a large touch screen with well-known widgets such as button or checkbox. This approach obvious advantage of this concept is familiar to majority of users. They are used to click on graphical widgets; therefore, they try to use their hands as mouse pointers, and click on the requested widgets. Feasibility of this task strongly depends on the implementation itself. In case of the well-tested implementation, the user is not forced to make large movements to move the pointer on the projection and on the other hand the motion detection is not too sensitive to move the pointer accidentally or too hastily. Generally, it is hard to find such equilibrium for all possible users. Moreover, this widget-based control approach was designed for communication with screens that can be directly touched (tablet, PC screen, embedded device). Touching a widget (or generally an object) is natural behavior; nonetheless, pointing in the air to move with some virtual pointer on a distant screen is not connected with any natural behavioral pattern.

The second approach is based on gestures. This method was adopted from mobile devices such as Apple iPad or iPhone. It is possible to find a substantial number of projects based on this principle. Different projects even propose very innovative and effective gestures. Nonetheless, the common problem is the usability. Although many proposed gestures are effective, they are not natural (see e.g. Kurtenbach, Hulteen, 1990 or Schindler *et al.*, 2008). Therefore, the user must be taught how to control the device. This has direct influence on the usability. Even Apple Inc. during years has designed a new set of gestures for multi-tasking (switch between applications, display

2: *Using tablet as a filter for large screen data (Source: IEEE Computer, April 2012, p. 50)*

running applications etc.). Although these gestures are also used among iOS users, their adoption is substantially lower than adoption of the "original" gestures (pinch and zoom, swipe, scroll). Obviously, they are not as natural as the original ones. The answer is in so called metaphors (described e.g. in Caroll *et al.*, 1988; Apple, 2013 or many others). An action is for the user natural in case there is a known parallel outside computer. Examples may include: drag-and-drop an object or move with an image/page as with a sheet of paper. To switch between photos or document pages by a movement of fingers from right to left is natural because it is similar to switching pages in a paper book. Nonetheless, there is no common parallel (metaphor) to action "show all running applications".

The last approach is based on the usage of special devices. As an example can be taken the solution described in Beaudoin-Lafon *et al.*, 2012. In this particular case a tablet is used to present additional information to the screen content (see Fig. 2). The tablet is a personal screen through which required data can be obtained. Certainly this approach is suitable mostly for collaboration on complex data processing and similar cases. Further example from consumer electronics, is the Sony Playstation Move device. The Move "wand" controller has on the end a color light that allows fast gamer movements tracking and buttons for additional actions. Again, this approach is based on the idea to control an application with natural user movements.

3 EXPERIMENT WITH GESTURE BASED PRESENTATION

Our main goal was to design and implement the standalone application, which enables creating common presentations and controlling them using natural hand gestures. Each presentation consists of several pages (slides) ordered horizontally. These slides contain customized texts, photographs and videos. A customer should be able to toggle between pages using natural hand gestures immediately, without any additional information. We have focused on finding simple swipe gestures in the first phase of development. We have chosen the depth maps of the scanned scene to detect movements of the customer, which is standing in front of the presentation. The depth maps are obtained from the Kinect device (see e.g. Khoshelham, Elberink, 2012).

The control gestures should be simple and intuitive. That was the reason, why we have focused only on the simple "swipe left" and "swipe right" gestures in the first development phase. The initial design was based only on the ideas of the development team members. The design was based on the assumption that the customer's palm is in the approximately same height as the shoulder in the case the user tries to perform movements of the slides in the presentation. As we later realized, this assumptions was wrong.

Implementation

We have chosen the Microsoft Kinect SDK libraries and the C# language to implement the gestures mentioned above. The whole application is developed using the MS Visual Studio 2010. The SDK provides functions for detecting up to 8 skeletons. The skeleton, which is the nearest one from the center of the scanned scene, is considered as the active one. It means that this skeleton is able to control the presentation movements. The SDK also provides positions of all joints of the skeleton. The *HandLeft*, *HandRight* and *ShoulderCenter* points are detected and saved into the internal FIFO structure together with the time stamp for later analysis. This information is sufficient for finding the gestures mentioned above. Our gesture analysis is based on finding sequences of points with a specific change of horizontal position (X coordinates) and small changes of the timestamp. The size of distance and time changes were set experimentally. This analysis is running in a separate thread. Whenever a swipe is detected, the *New Gesture Event Handler* is raised to inform the visualization engine. The reaction is an appropriate slide change.

As the visualization engine, we have selected the *Windows Presentation Foundation* (WPF). WPF is written in C# and is easily connected with *KinectSDK*. The entire visualization part, as well as all application configurations, are stored in an XML file. The XML file has been chosen to separate the visualization part from the rest of the application. XML file contains all the necessary information to generate a GUI and to fill it with an appropriate text, image and other elements.

3: *KiWall presentation application. Left: Screen from the presentation. Right: User switching between slides using Kinect standing on the computer.*

4 USER EXPERIENCE TESTING

Microsoft Kinect or Sony PlayStation Move have caused massive expansion of gesture-based interaction in consumer electronics. On the other hand, we argue that this method of user input requires more attention for user interface designing. Hence, we decided to make a thorough user experience test of our hand gestures controlled presentation. Our user experience evaluation was based on the testing method described in Koubek, Procházka, 2012. We decided to use the qualitative part of this approach. Our application provides only horizontal switching between screens realized by from-left-to-right and from-right-to-left gestures. These gestures were chosen because they represent essential functionality of any computer based system – switching between object. The main goal of the testing process was to get information about implemented hand gestures quality (*are they logical and natural for the user?*) and application functionality (*are there gestures sufficient for the application purpose?*). The other goal was to collect user's ideas about other possible application functions.

Our evaluation was done with a 20 person-large users group that consisted of students of Mendel University in Brno. Approx. half of them were students of computer science study programs; in the other half of the group were mostly students of Agronomy or Forestry faculty. All users individually got the instruction (*application is controlled by hand gestures*). Then they went through a set of tasks. After that a technician made an interview with them. Two cameras filmed the testing process. The first camera filmed our application and the second one took users' behavior.

During the test, the user had to find out how to control the program. If the right gesture was not found out in the three-minute time limit, it was shown to the user. 15 users found the right movement for switching screens in the given time limit; 9 of them found it in 30 seconds, thereof 2 users executed it at first try. The most of users (except 2 of them) used some variations of that correct hand movement. This behavior, together with data from the interview (default gesture is considered as the best by 15 users), confirms the used gestures correctness. In this part of testing we found two important problems in our gesture recognition engine. The first one was based on hand movement variance. We anticipated that movements would be more similar. Our engine was not able to recognize some of the gestures; even though the overall movement was in accordance with the gesture shape. The other important problem was the delayed response to user behavior. Both problems make gesture control irritating and lead to user confusion. During the interview, 14 users mentioned unsatisfactory gestures registration.

Hand movement variance

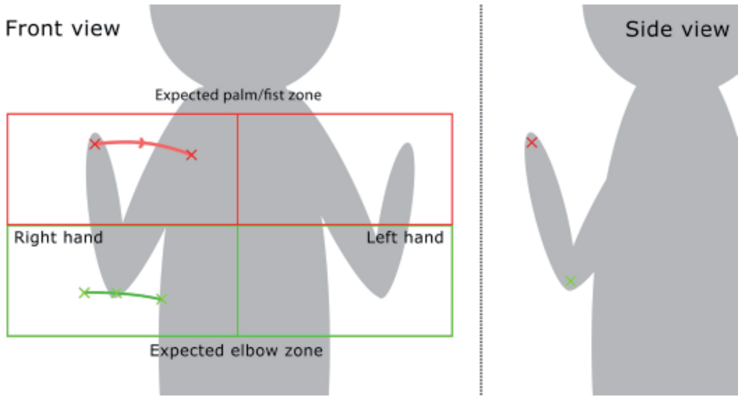
We assumed the position of the forearm will be in almost right angle with the rest of the arm, and the elbow location is under the palm in front view. Positions of elbow, forearm and arm were crucial for gesture detection. (see Fig. 4). Tested users showed many varieties of this movement, even with different hand position; e.g. palm was under elbow (see Fig. 5).

Response time to user behavior

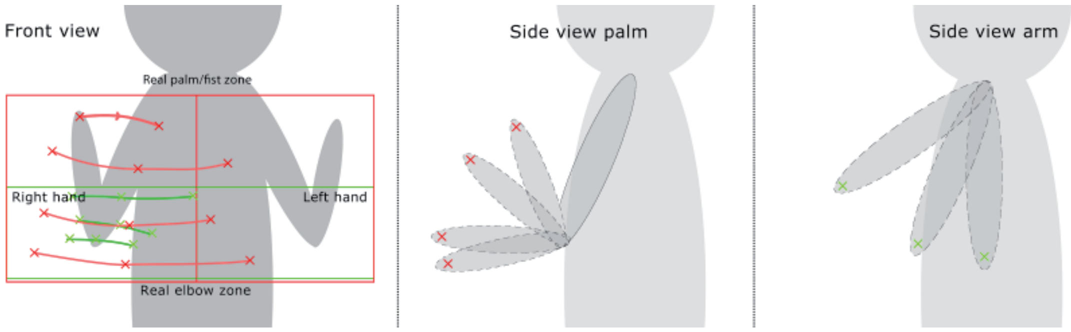
The difference between a user gesture and the program reaction was in some cases up to 2 seconds. This lag was determined as unacceptable for common usage. In this case the user performed usually additional gestures until the application responded. This behavior was misleading for all users, even for those who had known the right gesture set. 11 users complained about application delay. Therefore, the application response time must be up to 1 second.

5 DISCUSSION

Users were also asked to perform gestures for hypothetical application functionality – opening of the contextual menu, switching among vertical screens, rotation of 3D object, resizing of the object and playing video. The most common gesture for menu opening was pushing hand towards screen (7 users). 3 users would prefer a gesture similar to the mobile OS notification screen – menu pulling from the top of the screen to the bottom. Other users did some other gesture.



4: *Assumed gestures for horizontal screen switching*



5: *Performed gestures for horizontal screen switching*

A remarkable idea was counting by fingers – a number represents each item in the menu and the user can show the item number. Another habit from touch devices was used for the resizing gesture. 5 users performed a pinch-and-zoom gesture by fingers, 13 users by two hands.

Notable conformity was achieved in another hypothetical function. 19 users would switch among vertical screens by swinging to the bottom or to the top. It is analogous to the basic gesture of our application.

3D object rotation was presented by these gestures: 12 users grabbed imaginary the object by both hands pointing to the screen and rotated with the object in all 3 axis. 4 users did it by one hand.

The last gesture was the start of video playback. The most of the users tried something similar to opening the menu – pushing against screen (9 users). 3 users clapped by hands.

5 CONCLUSION

As mentioned above, there are several aspects that must be taken into account when a developer designs a control mechanism based on gestures. The first one is simplicity. None of the gestures can be long or complicated. Otherwise it would be a problem as to perform the gesture as to detect the gesture.

The second one is that the gesture must be natural for its users (usually must have a metaphor in the common movements, e.g. turning a book page). This is a key aspect in case when no user training is possible. Even in cases where the user can be trained (e.g. special applications such as described in Yan *et al.*, 2012 or in Riener, 2012), natural gestures should be preferred. As presented in the user evaluation part, it can be very useful to question target users how they think the gestures should be designed. In many cases, a substantial part can agree independently on some model. Therefore we assume that there can exist a widely accepted set of natural spatial gestures that can be comparable with basic mobile gestures.

In case it is possible, the developer should not create own gestures, but let the target user group to design the advanced gestures. Obviously, the majority of users can agree on completely different gestures than the developer. Therefore, we propose to use some kind of “user-design driven development”. Nonetheless, a long and complex user evaluation is a prerequisite to the development itself.

The third key aspect is the implementation. Performance is very important. If there is no immediate feedback whether the action (gesture) has been done correctly, the user is confused. Therefore the time for gesture recognition and feedback (e.g. turning slides) is very limited. Response time above 1 second is completely unacceptable. Natural response is up to approx. 0.5 second.

SUMMARY

Our paper is focused on the standardization of spatial gestures. The first part provides an overview of spatial gestures topic and related projects. The main emphasis is placed on the usability point-of-view of the projects. Our paper presents the usability as a key issue enabling the wide adoption of spatial gestures. As an example, design and implementation of our presentation software, which is controlled by gestures is used. This is described in the second part of the paper. Follows the usability testing results. As is outlined, we have tested our application on a group of users not instructed in the implemented gestures design. These results were compared with the other ones, obtained with our previous implementation. Presented evaluation can be used as the basis for implementation of similar projects. On the basis of our experiments, we formulated three key principles of spatial gesture design. The first one is simplicity. None of the gestures can be long or complicated. Otherwise it would be a problem as to perform the gesture as to detect the gesture. The second one is that the gesture must be natural for its users (usually must have a metaphor in the common movements, e.g. turning a book page). This is a key aspect in case when no user training is possible and even in cases where the user can be trained natural gestures should be preferred. The third key aspect is the implementation. E.g. performance is very important. If there is no immediate feedback whether the action (gesture) has been done correctly, the user is confused. Therefore the time for gesture recognition and feedback (e.g. turning slides) is very limited.

Acknowledgement

This work was supported by grant IGA FBE MENDELU 18/2013 (Enhancement of property management effectiveness using point clouds).

REFERENCES

Apple, Inc., 2013: iOS Human Interface Guidelines [on-line]. Cited: 21. 3. 2013. Available at: http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/Principles/Principles.html.

BEAUDOUIN-LAFON, M., HUOT, S., NANCEL, M., MACKAY, W., PIETRIGA, E., PRIMET, R., WAGNER, J., CHAPUIS, O., PILLIAS, C., EAGAN, J. R., GJERLUFSEN, T., KLOKMOSE, C., 2012: Multisurface Interaction in the WILD Room, *Computer*, 45, 4: 48–56. ISSN 0018-9162.

CARROLL, J. K., MACK, R. L. and KELLOGG, W. A., 1998: Interface Metaphors and User Interface Design. In: HELANDER, M. (ed.), *Handbook of Human-Computer Interaction*, Holland: Elsevier, 1167. ISBN 04-448-1862-6

KHOSHELHAM, K., ELBERINK, S. O., 2012: Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors,* 12, 2: 1437–1454. ISSN 1424-8220.

KOUBEK, T., PROCHÁZKA, D., 2012: Empirical evaluation of augmented prototyping effectiveness. *Acta Univ. Agric. et Silvic. Mendel. Brun.*, 60, 2: 143–150. ISSN 1211-8516.

KURTENBACH, G. and HULTEEN, E., 1990: Gestures in Human-Computer Communication The Art and Science of Interface Design. In: Brenda Laurel (ed.), *The Art of Human-Computer Interface Design*, Reading, Mass.. Addison: Wesley Publishing Co., pp. 309–317. ISBN 02-015-1797-3.

REINER, A., 2012: Gestural Interaction in Vehicular Applications, *Computer*, 45, 4: 42–47. ISSN 0018-9162.

SCHINDLER, K. Van GOOL, L., de GELDER, B., 2008: Recognizing emotions expressed by body pose: A biologically inspired neural model. *Neural Networks*, 21, 9: 1238–1246, ISSN 0893-6080.

YAN, X., FU, C.-W., HANSON, A. J., 2012: Multitouching the Fourth Dimension, *Computer*, 45, 9: 80–88. ISSN 0018-9162.

Address

Ing. David Procházka, Ph.D., Ing. Jaromír Landa, Ing. Tomáš Koubek, Ing. Vít Ondroušek, Ph.D., Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, 613 00, Brno, Czech Republic, e-mail: david.prochazka@mendelu.cz