MoKey – A Motion based Keyboard Interpreter

Martina Eckert, Marcos López, Carlos Lázaro, Juan Meneses, José F. Martínez Ortega

Abstract— **In this paper the authors present a new middleware for mapping gestures, obtained by a motion sensing camera device, to key events which are consumed by a standard off-the-shelf application. The aim is twofold: accessibility for users which are not able to use the keyboard because of physical impairments and the use of standard games for doing physical exercises. Hereby, special attention is laid on the adaptability to user requirements and easiness of configuration for the user himself and non-expert assistants. The actual state of our system is compared to similar proposals in terms of usability and performance, finally future working directions are outlined.**

I. INTRODUCTION

When the first motion sensing cameras appeared on the market, people instantly understood their usefulness as different, more natural interfaces for HCI (Human Computer Interaction) in normal life applications. In the state of the art of current investigations, two main lines can be found: one focusses on the employment of such a camera as an alternative interface for existing applications, i.e. to substitute keyboard and mouse as resumed in [1], the other line focusses on the capabilities to exploit the user movements, e.g. for physical exercises, muscle movement measures etc., which usually is realized with software explicitly created for that purpose. Here, the second line is more frequently applied to e-health tasks than the first one, but also a keyboard substitution could be very useful for handicapped people who have fine-motoric difficulties. And going a step further, the gestures could be exploited as, or transformed into, physical exercises.

In this sense, only one proposal has been found in literature: the Flexible Action and Articulated Skeleton Toolkit (FAAST), firstly published by Suma et al. in 2011 [2] and further improved in [3], which constitutes a powerful middleware between the camera and any off-the-shelf application.

Other groups proposed different kinds of amendments to FAAST, e.g. an "Application Wrapper" [4] or an interface which combines voice and hand gesture recognition [5]. The latest publication found about FAAST was its description in [6], but to date, the development has also been stopped. So, to our knowledge, there is currently no freely available supported middleware that enables a generic and adaptive usage of off-the-shelf software with a motion capture device in spite of being a very important and useful contribution. This general lack, in combination with new ideas and the motivation to improve FAAST, inspired the here presented work. The objective is to create a new, simple to configure middleware (especially for the user himself or a non-expert assistant), which could be used to exploit standard programs (preferably games) for serious purposes as physical exercises and will be fully adaptive to the user's capabilities and necessities.

This paper is the first public presentation of MoKey, which is still under development, so it aims to show its global structure, design differences in comparison with FAAST and preliminary tests results. Finally, the ongoing enhancements are also lined out.

II. SYSTEM DESCRIPTION

Being a very powerful tool, which is thought to cover every imaginable possibility to configure gestures, FAAST shows a very important drawback which is its tedious configuration procedure. It takes time to understand and even experienced users need several minutes to configure or reconfigure the system. Additionally, the way to define the gestures is via numerical values which are not intuitive, and the user itself cannot find out or change on the fly. A further disadvantage of FAAST is that it seems to be quite memory and CPU consuming. Tests have been performed to measure user friendliness and performance and are presented in comparison to the here proposed interface in the next chapter.

The proposed system is designed like an interpreter of a person's movements captured by the motion capture camera which are assigned to key-events and sent to the application as shown in Fig. 1.

MoKey can be configured for standing pose with twelve predefined movements or in seated mode applying six of them related to the upper body parts. This pose is especially interesting for users depending on a wheelchair. In contrast to FAAST, where different movements of body parts have to be combined to create a gesture, here simple gestures have been predefined, aiming to be generic, intuitive and executable by persons with different agilities, including disabilities, see the list in the user interface shown in Fig.2.

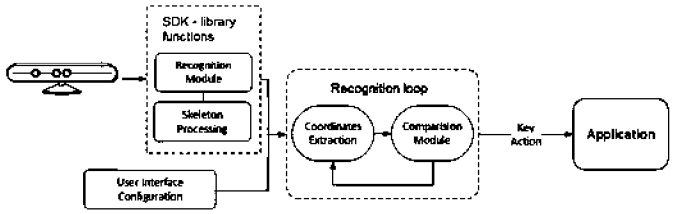

Fig. 1: MoKey. System flow from left (gesture capture) to right (application).

This work was sponsored by Spanish National Plan for Scientific Tech. Res. and Innov. TEC2013-48453-C2-2-R.

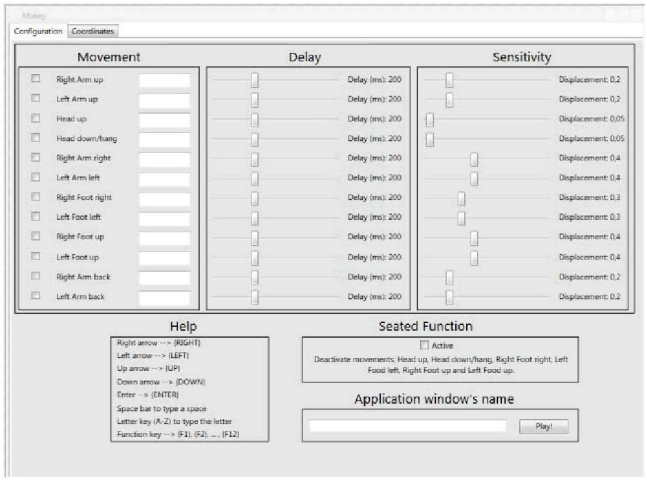

Fig. 2: MoKey user interface.

Nevertheless, the defined gestures are not fixed, they can be easily adapted to individual needs with help of two simple sliders: sensitivity (displacement range) and delay (time in-between repetitive key events). The sensitivity regulation permits an adaptation to particular necessities of the user, e.g. in case of usage as an instrument for physical exercises. The modification of the delay is useful in case that the application needs repetitive key events or when the user has difficulties to get back to the original pose in time.

To make the tool still more useful and easier to adapt to individual needs, a motion recording functionality is currently implemented. The goal is to provide the possibility to train the application with individual user specific gestures, which could be helpful e.g. for special physical exercises (the application only responds if the user is doing a gesture well) or for severe motor impairments (e.g. the range of movement is very small or slow).

III. Preliminary Test Results

So far, only basic tests have been performed, mainly to prove the functionality and usability and in comparison with FAAST. The implementation is using the SDK v.1.8 with a first generation camera and is running on a 64 bit, 2 CPU, 2.13 GHz, 3.25 GB RAM, personal computer.

To compare the easiness of configuration between MoKey and FAAST, two persons have been requested to setup the system with two simple gestures: "right arm right" to press A and "left arm left" to press B. Person 1 was an expert, person 2 saw both applications for the first time. Results for configuration times are shown in Table 1 and clearly confirm that FAAST is not easy to set up.

TABLE I: Example Setup Times

User	MoKey	FAAST*
Expert	22s	120s
Non-expert	72s	204s **

*the application needs additionally 36s to connect to the Kinect
** the user didn't achieve to configure the system

For playing trials, Tetris was chosen as example application, as it is widely known, easy to use and the high-scores gives some feedback about usability of the middleware. Nevertheless, it works with every other key-stroke driven software. 6 test persons between 10 and 25 years played the game with MoKey and FAAST, among them a 13 year-old non-ambulant boy with Duchenne Muscular Dystrophy who has already a limited range of arm movements. As the tests have been only informative, no data is presented, but it has been observed that both tools require a similar training times and afterwards the scores are also similar. Soon, complex tests will be performed and published, above all with impaired users.

Another interesting aspect to compare was the performance of both tools on a medium range machine. Table 2 shows the CPU and RAM usage of both applications when paused and in use, with 1 and 4 gestures assigned to key events. As can be clearly seen, FAAST, being more complicated, needs double processor power and more memory to achieve the same functionalities.

TABLE II : Predefined Configurable Gestures

	CPU %		RAM (kB)	
	MoKey	FAAST	MoKey	FAAST
no user	14	15	73820	65016
1 gesture, quiet	36	88	68804	83360
1 gesture, moving	37	86	68408	83360
4 gestures, quiet	37	86	70208	83032
4 gestures, moving	38	87	70584	83168

IV. Conclusions

The needs in e-health are wide, and a middleware could be an excellent solution to exploit any kind of application for physical exercises or to adapt it for impaired users.

As the experiments show, the presented tool, although covering much less configuration possibilities then FAAST, is easier to use while promising to provide sufficient user adaptability. Work is going on to further improve the tool and achieve a fully adaptive interface, easy to use by the user himself as well as by unexperienced assistants.

References

[1] T. Saldok and F. Ott, "Vergleich verschiedener Maus-Emulatoren für Microsoft Kinect", available online: http://www.soziotech.org/vergleich-verschiedener-maus-emulatoren-fuer-microsoft-kinect/ (accessed on 24 March 2015)

[2] E. A. Suma, B. Lange, A. S. Rizzo, D. M. Krum, and M. Bolas, "FAAST: The Flexible Action and Articulated Skeleton Toolkit," *IEEE Virt. Reality Conf.*, pp. 247–248, Mar. 2011.

[3] E. A. Suma, D. M. Krum, B. Lange, S. Koenig, A. Rizzo, and M. Bolas, "Adapting user interfaces for gestural interaction with the flexible action and articulated skeleton toolkit," *Computer Graphics*, vol. 37, no. 3, pp. 193–201, May 2013.

[4] F. Lamberti, A. Sanna, G. Paravati, C. Demartini, and P. Torino, "Endowing Existing Desktop Applications with Customizable Body Gesture-based Interfaces," in *ICCE.*, pp. 558–559, 2013.

[5] S. Lee and S. Oh, "A Kinect Sensor based Windows Control Interface," *Int. J. Control Autom.*, vol. 7, no. 3, pp. 113–124, 2014.

[6] S. Koenig, A. Ardanza, C. Cortes, A. de Mauro and B. Lange, "Introduction to Low-Cost Motion-Tracking for Virtual Rehabilitation", in J. L. Pons and D. Torricelli (eds.), *Emerging Therapies in Neurorehabilitation*, Biosystems & Biorobotics 4, Springer-Verlag Berlin Heidelberg, pp. 287-303, 2014.