Increasing Kinect Application Development Productivity by an Enhanced Hardware Abstraction

Bernardo Reis, João Marcelo Teixeira, Felipe Breyer, Luis Arthur Vasconcelos, Aline Cavalcanti, André Ferreira, Judith Kelner
Virtual Reality and Multimedia Research Group – Computer Science Center
Federal University of Pernambuco, Recife – Pernambuco – Brazil
{bfrs, jmxnt, fbb, lalv, asc3, amf2, jk}@cin.ufpe.br

ABSTRACT
Designing and implementing the interaction behavior for body tracking capable systems requires complex modeling of actions and extensive calibration. Being the most recent and successful device for robust interactive body tracking, Microsoft’s Kinect has enabled natural interaction by the use of consumer hardware, providing detailed and powerful information to designers and developers, but little tooling. To fulfill this lack of adequate tools for helping developers in the prototyping and implementation of such interfaces, we present Kina, a toolkit that makes the development not fully conditional to the existence of a sensor. By providing playback capabilities together with an online movement database, it reduces the physical effort found while performing testing activities.

Author Keywords
Application testing; development methodology; interaction database.

ACM Classification Keywords
D.2.5 Software Engineering: Testing and Debugging - Testing tools

INTRODUCTION
Microsoft’s Kinect sensor was originally developed as an accessory for the Xbox 360 videogame console that would allow interaction without touching a game controller, through natural postures and gestures [9]. Soon enough the scientific community realized that it could be applied to a wide range of applications, from robot navigation support [1] to high quality dense 3D reconstruction [10].

To test such applications is often a hard and time consuming activity because it is difficult to redo tests using the same reference input, and furthermore it requires considerable physical effort from the users to perform movements and body gestures.

Some issues are particularly interesting in the field of developing Kinect-based applications:

- Equipment restriction - the development phase demands a Kinect sensor to be available all the time. For example, a development team consisting of 4 users should have 4 different sensors in order to make possible different and independent tests at the same time;

- Physical effort during tests - testing applications requires at least one user to be placed on the view area of the sensor and to perform some specific movements. To repeat the same tests means that the user must perform the same physical actions over and over, which can cause physical weariness resulting from exertion;

- Some body gestures require specific knowledge - while developing health support applications, for example, such as functional exercises or physiotherapy evaluations, the support of a specialist is needed in order to verify whether the movement being used as input is correct. Therefore, the figure of a specialist is required during most of development and testing phases.

Some work has been done concerning the first issue, but only few are applied specifically to the Kinect device itself. One way of sharing the same device for more than just one user is to adopt the concept of virtual USB ports [3]. This way, the device is connected to a single PC and its information is passed over the network to any of the connected users. Unfortunately, it does not solve the problem of simultaneous accesses, since only one access is permitted at a time. In [14], the authors propose an open source tool for distributing data from Kinect over the web. By using their solution it is possible to share and access the data independently, which solves the first issue. However, scalability and data transmission still are bottlenecks, as they limit the number of simultaneous users that can access the device.

The second and third issues, regarding physical effort and specialized information, respectively, could be solved by recording user input and then reproducing the data that was captured. This way, during the development test phase, a single user input would be necessary for each different movement. The Fakenect library [2] is an existing solution that captures the dump of the Kinect sensor and later reproduces the information to the application (more details about this library will be given in the next sections).