Teaching Natural User Interaction Using OpenNI and the Microsoft Kinect Sensor

Norman Villaroman
Brigham Young University
Provo, Utah
normanhv@byu.edu

Dale Rowe, Ph.D.
Brigham Young University
Provo, UT
dale_rowe@byu.edu

Bret Swan, Ph.D.
Brigham Young University
Provo, UT
bswan@byu.edu

ABSTRACT
The launch of the Microsoft Kinect for Xbox (a real-time 3D imaging device) and supporting libraries spurred the development of various applications including those with natural user interfaces. We propose that using Kinect offers opportunities for novel approaches to classroom instruction on natural user interaction. A number of development frameworks has come up that can be used to facilitate this instruction. We evaluate the current state of this technology and present an overview of some of its development frameworks. We then present examples of how Kinect-assisted instruction can be used to achieve some of the learning outcomes in Human Computer Interaction (HCI) courses as outlined in IT2008. We have verified that OpenNI, with its accompanying libraries, can be used for these activities in multi-platform learning environments.

Categories and Subject Descriptors
K.3.1 [**Computers and Education**]: Computer Uses in Education – Computer-Assisted Instruction

General Terms
Design, Experimentation, Human Factors

Keywords
HCI, Kinect, OpenNI, Natural User Interface, Education

1. INTRODUCTION
Human-computer interaction (HCI) is a fundamental pillar in the Information Technology discipline.[1] Any interactive computing system involves one or more interfaces with which a user can provide commands and get results. Interactive computer systems started with command-line interfaces that are still used today. The development of graphical user interfaces has allowed users with varying levels of computer skills to use a wide variety of software applications. Recent advancements in HCI have facilitated the development of natural user interfaces that provide more intuitive ways of interacting with the computing device. The development of natural user interfaces is expected to make it easy for users to learn how to use the interface in the quickest possible way. [1]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGITE'11, October 20–22, 2011, West Point, New York, USA.
Copyright 2011 ACM 978-1-4503-1017-8/11/10...$10.00.

The desire to develop natural user interfaces has existed for decades. Since the last world war, professional and academic groups have been formed to enhance interaction between "man and machine" [2, 3]. While computer user interfaces started with punch cards and keyboard-like devices, attempts to develop interfaces that process more natural movements emerged relatively early. An example of such an attempt involved the study of pattern recognition of natural written language using a pen-like interface called the RAND tablet and stylus in 1967 [4, 5]. An estimation of the value of the ability of a machine to process audio commands was published in 1968 [6]. The first touch device, called the Elograph was created in 1971, with further advancements from the same company continuing in the following decades. In the late 1970s, VIDEOPLACE, a system with video cameras and projectors allowed graphical interaction and communication between users without the aid of user-attached or controlled devices, was developed. The 1980s saw the beginnings of multi-touch systems with a master's thesis by Nimish Mehta that resulted in the development of the *Flexible Machine Interface*. In the same decade, touch screen interfaces began to be prevalent in commercial establishments. At this time, *Hewlett-Packard 150*, a personal computer with basic touch input, was launched. In the 1990s, touch screen interfaces made its way to mobile devices with *Simon*, launched by IBM and Bell South in 1994 [7]. Touch screens continued to spread in different consumer applications. In the first decade of this century, more advanced gestural technology and accompanying applications have surfaced with a plethora of PDAs, smartphones, and multimedia devices. Natural interfaces have become more prevalent in recent years in gaming systems that use such input such as the *Nintendo Wii Remote*, the *PlayStation Eye/Move*, and the *Microsoft Kinect for Xbox.*

The emergence of these natural user interface and gestural technologies has motivated researchers and hardware enthusiasts to look for novel ways that they can be used. In addition to their use in consumer applications, more of these technologies are finding their way into the classrooms, particularly for the study of HCI. We suggest that the Kinect is an excellent addition to the toolset for teaching such topics for reasons discussed in this paper.

[1] The terms for and the definitions of the different kinds of interfaces including those not mentioned may vary from one expert to another and may even overlap. This paper does not intend to clarify the scope of such terms. Suffice it to say that there have been continuous advancements in making the user interface easier and more intuitive for the user to learn.

Our paper begins with a discussion and assessment of the current state of Kinect-enabled technology. We then make recommendations for how this technology can be applied in HCI course instruction. Finally, we discuss the advantages and disadvantages of a Kinect-assisted HCI instruction.

2. DEVELOPMENT FRAMEWORK

Before presenting ideas on how Kinect can be used in the classroom, the following shows the state of the technology as it exists today.

2.1 Hardware

The Kinect is based on a sensor design developed by *PrimeSense Ltd*. The technical specifications described here are taken from the reference design of the *PrimeSense* sensor.[8]

Light Coding ™ is the technology that allows it to construct 3D depth maps of a scene in real-time with the amount of detail that it does. Structured near-infrared light is cast on a region of space and a standard CMOS image sensor is used to receive the reflected light. The PS1080 SoC (System on a Chip) is able to control the structured light with the aid of a light diffuser and is able to process the data from the image sensor to provide real-time depth data [9].

The depth image size from the PS1080 has a maximum resolution of 640 x 480. At 2m from the sensor it is able to resolve down to 3mm for height and width and 1cm for depth. It operates at a range between 0.8m and 3.5m. Experimentation has shown that Kinect is only able to process depth data at a frame rate of 30 fps.

The sensor also has an integrated RGB camera with a maximum resolution of 1600x1200 (UXGA) to match the depth data with real images. It also has built-in microphones for audio transduction. Connectivity is enabled by a USB 2.0 interface.

While it has to be noted that the Kinect sensor is not the only device that uses the *PrimeSense* reference design (e.g. *ASUS Xtion Pro*), the presented activities in this paper were accomplished using it.

Its basic applications do not require special or powerful computer hardware. A dual-core machine with 2GB RAM and standard video graphics processor can handle these applications just fine.

2.2 Software

At the moment of this writing, there are three major projects with freely available libraries that can be used to collect and process data from a Kinect sensor – OpenKinect [10], OpenNI [11], and CL NUI [12].

Before the official release of Microsoft Kinect on November 4th, 2010, *Adafruit Industries* announced that they would give monetary reward to hardware enthusiasts who can write software that will enable a PC to access and process the data coming from the sensor [13]. A developer with an alias of *AlexP* was able to successfully accomplish the challenge first at two days after the launch but it was Hector Martin who was recognized as the winner as the former was not willing to open-source his code at the time [14].

The efforts of these two individuals led to the beginning of two of the projects mentioned. Hector Martin released his code on November 10th, which marks the beginning of *OpenKinect* [15]. Among other things, this project provides drivers for Kinect, wrappers for different languages and other projects, and an analysis library that are all open-source. This project is distributed under both the Apache 2.0 and the GPL v2 license [16].

AlexP's code was used in the development of the *Windows Kinect Driver/SDK - CL NUI Platform* of *Code Laboratories*, version *1.0* of which was released officially on December 8th. The driver and the SDK with C/C++/C# libraries are freely available for use by the public.

In the same month of the Kinect's launching, OpenNI [11, 17] was formed by a group of companies, which included *PrimeSense Ltd.,* as a not-for-profit organization that aims to set an industry-standard framework for the interoperability of natural interaction devices. With the framework they have released came middle-ware libraries that can convert raw sensor data from a compliant device to application-ready data. These libraries are available for applications written in C, C++, and C#. Because Kinect is an OpenNI-compliant device, this software can be used to build applications for it.

OpenNI is written in C/C++ so that, among other reasons, it can be used across different platforms. While OpenNI is officially supported on later versions of Windows from *Windows XP* and *Ubuntu* from 10.10, [18] its use in other Linux distributions and Mac OS X has been documented to work in online forums.

Microsoft Research has announced the release of a non-commercial Kinect for Windows SDK in the spring of 2011, with a commercial version coming later. [19] The initial release promises, among other things, audio processing with source localization.[20] In the three development platforms mentioned, processing audio signals from the sensor's microphones is not enabled and is currently under development. This official Microsoft SDK may also provide the support necessary to accomplish the learning outcomes discussed here once it is available.

3. EDUCATIONAL USE

Familiarity with a Kinect-related SDK will help students be able to develop and understand applications that work with the Kinect. Any of the three software packages mentioned can be used to create HCI learning activities. For the purposes of this paper, OpenNI, with its accompanying middleware library called NITE, was investigated for its academic value. OpenNI was selected for reasons that will be discussed hereafter.

In this section the advantages and limitations of using Kinect-based applications to teach various topics in HCI are discussed. An enumeration of some of these topics is given. And some learning activities are outlined.

3.1 Suggestions for HCI Instruction Using Kinect

HCI is a required and fundamental area in the IT discipline. With the current industry trend, gesture-based, natural user interaction should be part of the context of the different topics in HCI. Kinect can provide activities that aid in the study of natural user interaction that would otherwise be unavailable.

Learning activities using Kinect could easily be developed to fulfill core and advanced learning outcomes outlined in the *IT 2008: Curriculum Guidelines in Undergraduate Degree Programs in Information Technology* [1]. The following

subsections enumerate topics listed in IT2008 and suggest some Kinect-assisted learning activities for each. For the sake of brevity and clarity, the use of device-free, natural, gestural and possibly multimodal interfaces enabled by Kinect will be called "Kinect-controlled" or "Kinect-enabled" interfaces in the following sections. These applications can be used in these learning activities "out of the box" or can easily be modified to provide basic applications that students can test or even build on.

3.1.1 Human Factors

Sample activities for core learning outcomes

- Study how cognitive principles, affordance, and feedback should influence the design of Kinect-controlled interfaces in desktop computers.
- Discuss if and how Kinect-enabled interfaces can cause repetitive stress syndrome.
- Analyze the abilities of users of different age groups to use Kinect-controlled interfaces.

Sample activities for advanced learning outcomes

- Design and implement a Kinect-controlled interface accounting for appropriate cognitive, physical, cultural, and affective ergonomic considerations..
- Experiment on various conditions that would increase or decrease the usability and user experience of a particular Kinect-controlled interface.

3.1.2 HCI Aspects of Application Domains

Sample activities for core learning outcomes

- Describe the advantages and disadvantages of using Kinect-control in web browsing.
- Describe conditions of the Kinect-enabled interface that would enhance usability and user experience in web browsing

Sample activities for advanced learning outcomes

- Design and implement a Kinect-controlled interface for a specific application domain – such as web browsing.
- Discuss how a particular application can benefit from multimodal interfaces that include Kinect-enabled gesture interactions.

3.1.3 Human-Centered Evaluation

Sample activities for core learning outcomes

- Perform usability and user experience evaluations for a commercial Kinect-enabled application
- Come up with usability guidelines and standards for Kinect-enabled interfaces.

Sample activities for advanced learning outcomes

- Perform a heuristic evaluation of the usability of a Kinect-enabled application.
- Determine and design performance, usability, and user experience metrics for Kinect-enabled interfaces.
- Evaluate the appropriateness of traditional usability and user experience testing methods to assess Kinect-enabled interactions (both single and multimodal).

3.1.4 Developing Effective Interfaces

Sample activities for core learning outcomes

- Determine the types of user processes, workflows, and tasks that are more or less appropriate for Kinect-enabled interfaces.
- Implement a Kinect-enabled user interface that integrates and triggers typical cursor, mouse, and keyboard events.

Sample activities for advanced learning outcomes

- Apply different cognitive models or interaction frameworks to assess Kinect-enabled interface design.
- Study how Kinect-enabled interfaces can be adapted to be able to adjust to different characteristics of user personas and demographics (e.g. age, size, etc.). If appropriate, test and redesign available calibration methods for such a purpose.
- Implement a Kinect-enabled interface that uses skeleton detection, pose detection, or depth map processing.

3.1.5 Accessibility

Sample activities for core learning outcomes

- Identify different ways that a Kinect sensor can assist users with certain disabilities.
- Consider the limits and capabilities of Kinect for disabled users and analyze the risks of such assistance.

Sample activities for advanced learning outcomes

- Discuss how 3D depth maps or gestures can improve computer use for users with disabilities or special needs.

3.1.6 Emerging Technologies

Sample activities for core learning outcomes

- As an emerging technology, explore how Kinect can help advance the field of gesture-based, natural user interaction.
- Give examples of how Kinect-controlled interfaces can be used for ubiquitous computing environments.

Sample activities for advanced learning outcomes

- Compare the Kinect as a video game input device with comparable devices marketed by its competitors such as the *Nintendo's Wii Remote Controller* and the *PlayStation Eye/Move,* as well as other traditional (e.g., mouse, keyboard, and game controllers) and emerging (e.g., voice-recognition) interaction technologies.
- Illustrate how existing applications can be improved upon by Kinect-enabled interfaces.

3.1.7 Human-Centered Computing

Sample activities for core learning outcomes

- Test and analyze whether usability and user experience requirements can be met with the current capabilities of Kinect-enabled user interfaces. Determine what technological improvements are necessary to meet those requirements, if not currently met.

Sample activities for advanced learning outcomes

- Test an existing Kinect-based application in a user environment, and redesign the interface for that application to better satisfy user experience and usability requirements.

3.2 Advantages & Limitations

The following discusses some of the advantages of using Kinect in the classroom. Some caution is also given on its use.

Kinect is revolutionary as a computer vision sensor because of depth data it can generate of a scene in great detail, in real-time, and at a significantly lower cost compared to alternatives. The technology allows an interface without user-attached controllers. Several years ago, computer vision techniques for 3D processing on readily available hardware are virtually non-existent.[21] And even with the development of such in the last several years, they are usually computationally expensive and difficult to implement because the 3D reconstruction from 2D sensors is done in software. Inherent in Kinect's technology is its ability to reduce such computing by getting the depth image in hardware.[8]

Some time-of-flight 3D sensors can be used and have been used in the academe in the fields of computer vision and HCI to generate depth data from a scene. While there is an advantage in their being able to do in hardware what would otherwise have to be done in software, these can cost tens of times more expensive and at the same time provide depth data at a much lower resolution than the Kinect. Clearly, the availability and affordability of Kinect sensors make it a good learning tool to use in courses where they will be useful.

If resources limit the number of sensors that can be used by students, OpenNI allows the recording of sensor data streams that can be played back and processed without an actual sensor. These recordings provide a standard data set for students to process where such a condition is required.

Because Kinect makes the user the controller, as opposed to physically controlling an input device, it requires a greater level of involvement on the part of the user or the developer testing it. This can make any learning activity that uses it quite engaging. Graphical toolkits can easily be used to display the processed data, as shown in the included sample applications, which makes it visually engaging.

A simple Internet search on Kinect-enabled applications will show scores of applications that have already been developed soon after the Kinect's release. Even though they are based on the same sensor technology, these applications vary widely in purpose and usage. Having a plethora of explored and unexplored applications provides a great opportunity for students to exercise creative thought and innovation.

OpenNI was selected over the other two platforms because it provides important advantages for students. It can be used in multiple operating systems so that learning is not platform dependent. It is designed and maintained by its member companies to be an industry standard, which implies a sufficient degree of stability and maturity. It is open source, which increases the rate of maturity and allows more in-depth study of the framework. Perhaps the greatest advantage of its use by students and educators is that documentation of the framework and the APIs at the time of this writing far surpasses that of the others.

The NITE middleware library it comes with work seamlessly with it and provides a wide range of functionalities and sample applications that are more than sufficient for a structured course on natural user interaction. These include (though not by any means complete):

- Recognition of push, steady, swipes, waves and other gestures
- Skeleton detection and tracking of individual skeleton joints
- Pose detection
- User segmentation and multiple user detection
- Accessing depth and video data
- Multiple point tracking
- Various calibration and smoothing functions to enhance recognition
- Sample applications of gesture-controlled interfaces, user segmentation (Figure 1), point tracking (Figure 2), skeleton tracking (Figure 3), and of other functionalities previously mentioned.

Figure 1. Multiple user segmentation

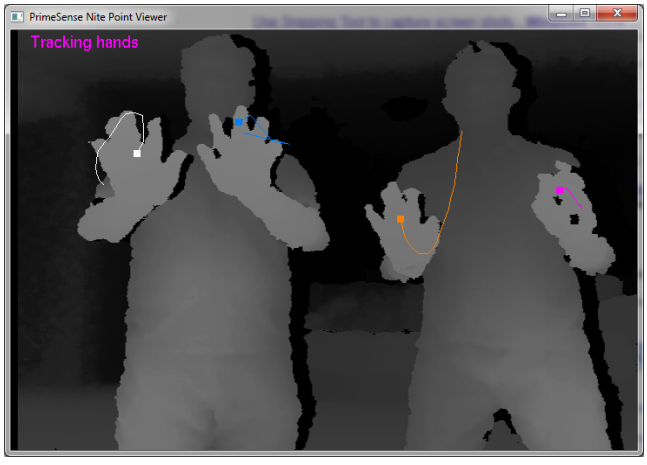

Figure 2. Multiple point tracker

Figure 3. Skeleton tracking

Students who have gone through an undergraduate advanced programming course in C/C++ will find it relatively easy to develop applications directly from the project libraries. If the students do not have this advanced level of understanding but only have basic programming skills, wrapper classes or incomplete implementations can be written to make its use simpler for them. Even just using the sample applications out of the box could be useful in various topics of HCI.

These and the other standard features that are included in OpenNI/NITE could be used in learning activities for HCI. While these could very well be sufficient, there are many other open-source and commercial applications that can be used if necessary.

The technology behind Kinect also has the potential to allow significant advances in natural user interaction technology, as evidenced by the wide variety of applications that have been developed from the very beginning of its launch. Students interested in the field of computer vision and/or natural user interaction will benefit from keeping up with this advancement.

While Kinect-enabled technology can be a very helpful aid in learning HCI topics, caution is given to keep its use appropriate and balanced with other technologies that are also useful for HCI instruction.

4. CONCLUSION

The Microsoft Kinect sensor and its supporting development platforms provide significant learning opportunities in various topics of Human Computer Interaction. A number of such platforms are currently available. In this paper, we evaluated OpenNI. OpenNI and its libraries was found to be sufficient and advantageous in enabling Kinect-assisted learning activities.

Kinect and OpenNI can be used to provide students with hands-on experience with this gesture-based, natural user interaction technology. Since gesture-based interaction technologies are becoming a standard part of commercial systems, IT students will benefit from integrating this technology in their education, such as in HCI courses, in order to learn how to keep up with advancements in this exciting technology.

5. REFERENCES

[1] Lunt, B., Ekstrom, J., Reichgelt, H., *et al.*, IT 2008, *Communications of the ACM*, Vol 53, Iss. 12, pp 133.

[2] Birmingham, H. P., Human Factors in Electronics-Historical Sketch, *Proceedings of the IRE*, Vol 50, Iss. 5, pp 1116-1117, 1962.

[3] Sanders, M. S., *Human factors in engineering and design: Mark S. Sanders, Ernest J. McCormick*. New York: McGraw-Hill.

[4] Hornbuckle, G. D., The Computer Graphics User/Machine Interface, *Human Factors in Electronics, IEEE Transactions on*, Vol HFE-8, Iss. 1, pp 17-20, 1967.

[5] Davis, M. R. and Ellis, T. O., The RAND tablet: a man-machine graphical communication device. *In Proceedings of the Proceedings of the October 27-29, 1964, fall joint computer conference, part I* (San Francisco, California), ACM.

[6] Lea, W., Establishing the value of voice communication with computers, *Audio and Electroacoustics, IEEE Transactions on*, Vol 16, Iss. 2, pp 184-197, 1968.

[7] Saffer, D., *Designing Gestural Interfaces*. Sebastopol : O'Reilly Media, In., Sebastopol.

[8] Ltd., P., The PrimeSensor (TM) Reference Design 1.08. *PrimeSense Ltd*, http://www.primesense.com/files/FMF_2.PDF (Last Accessed: April 2011).

[9] Zalevsky, Z., Shpunt, A., Maizels, A., *et al.*, Method and System for Object Reconstruction. 2006. http://www.wipo.int/patentscope/search/en/detail.jsf?docId=WO2007043036.

[10] OpenKinect, OpenKinect Main Page. http://openkinect.org/ (Last Accessed: April 2011).

[11] OpenNI, OpenNI. http://openni.org/ (Last Accessed: April 2011).

[12] Laboratories, C., About: CL NUI Platform. *Code Laboratories*, http://codelaboratories.com/kb/nui (Last Accessed: April 2011).

[13] Industries, A., The Open Kinect project – THE OK PRIZE – get $3,000 bounty for Kinect for Xbox 360 open source drivers. http://www.adafruit.com/blog/2010/11/04/the-open-kinect-project-the-ok-prize-get-1000-bounty-for-kinect-for-xbox-360-open-source-drivers/ (Last Accessed: February 2011).

[14] Giles, J., Inside the Race to Hack the Kinect. 2010. http://www.newscientist.com/article/dn19762-inside-the-race-to-hack-the-kinect.html?full=true.

[15] OpenKinect, OpenKinect History. http://openkinect.org/wiki/History (Last Accessed: April 2011).

[16] OpenKinect, OpenKinect Policies. http://openkinect.org/wiki/Policies (Last Accessed: April 2011).

[17] OpenNI, About. http://openni.org/about (Last Accessed: May 2011).

[18] OpenNI, OpenNI User Guide. 2011. http://openni.org/images/stories/pdf/OpenNI_UserGuide_v3.pdf.

[19] Knies, R., Academics, Enthusiasts to Get Kinect SDK. Microsoft Research, http://research.microsoft.com/en-us/news/features/kinectforwindowssdk-022111.aspx (Last Accessed: April 2011).

[20] Research, M., Kinect for Windows SDK beta. http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/ (Last Accessed: May 2011).

[21] Branislav Kisaécanin, V. P., Thomas S. Huang, Real- time vision for human-computer interaction. *Germany: Springer Verlag*. 2006.