User Interface Issues of 3D Gesture Mobile Devices and Visual Feedback of Physiotherapy Application

Ryan Rho

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-88
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-88.html

May 17, 2013

User Interface Issues of 3D Gesture Mobile Devices and Visual Feedback of Physiotherapy Application

Ryan Rho
with Ross Yeager, Aarthi Ravi, Zhaochen Liu, Hassan Elahi
Electrical Engineering and Computer Science, University of California, Berkeley
mKinect Team
ryanrho@berkeley.edu

Abstract

We introduce and discuss new user interface issues raised by the advent of gesture recognition-enabled mobile devices. Compared to traditional hand gestures on touch-screens of mobile devices, the effectiveness of gesture recognition on mobile devices depends on their screen sizes, the distance between the user and the device, and how the user holds the device. Due to these user interface issues, software developers who would like to apply gesture recognition to their applications should redesign their user interfaces different from typical mobile applications. To effectively visualize feedback of a physical therapy application for mKinect, the purpose of which is to bring the users Kinect experience onto their smartphones, we present a novel approach of visual feedback for a gesture recognition-enabled mobile devices. We display a shadow of the player and arrows of how the user should correct their joints when performing an exercise so that not only it is easy for the user to receive feedback but it is also accurate enough to perform a physiotherapy exercise.

Author Keywords

motion detection, 3D gesture, gesture recognition, mobile devices, smartphones, tablets, kinect, mKinect, user interface, physiotherapy

1. Introduction

Motion detection technology to consumer products has become popular through game consoles such as Xbox Kinect. Along with the popularity of mobile devices, as of April of 2013, there have been early attempts to implement motion detection on smartphones. For instance, Samsung and Pantech introduced a basic hand motion gesture feature on their smartphones [3, 11]. They are mainly targeting users who are temporarily unable to touch their smartphones due to driving or cooking.

1.1 Gesture Recognition in Mobile Devices

It is not trivial to implement fully supported gesture recognition into portable devices due to their limited performance, power and battery, compared to desktop computers. Computer vision algorithms for gesture recognition need enough power of computation with limited hardware on mobile devices. As a result, motion detection on mobile devices is either aimed to provide as a simple feature, such as answering call by waving your hand on your smartphone, or provide rich gesture recognition by adding additional hardware such as sensors and CPUs [5, 6].

Among these two different perspectives, we believe that as portable batteries have been getting smaller and more powerful [14], these constraints will eventually reduce, which will result in successfully implementing fully-functional gesture recognition on mobile devices in the long run. For instance, PrimeSense has recently demonstrated Capri in the International Consumer Electronic Show 2013, which is the smallest yet affordable 3D sensor that can be attachable to a small device and provide 3D sensing platform [6].

There are several advantages if mobile devices have gesture recognition capabilities. For instance, users do not have to touch the screen in order to interact with mobile devices. This is especially great for the case when it is difficult for the user to provide touch input, such as driving, cooking, or giving a presentation. In addition, gesture recognition introduces a new way for users to interact with mobile devices, which produces potential to create innovative mobile applications. For instance, Samsung introduced face recognition in its smartphone product Galaxy S3. Facebook recognition is a form of gesture recognition in that the user triggers a command with his or her face. The users can unlock their devices by placing their faces in front of cameras of their devices [15].

1.2 Kinect

One of popular gesture recognition sensors available as a consumer product is Microsoft's Kinect sensor. Kinect was originally introduced in 2010 as an Xbox gaming device. Kinect tracks color and depth information through a camera and an infrared sensor, mainly providing the software developers to track the player's movement.

By using a smartphone's camera and an additional infrared sensor for receiving color and depth data, we found that Kinect could be simulated with Windows Phone devices. In addition, both Kinect and Windows Phone run on .NET Framework so that they are compatible each other. As a result, we found it may be possible to implement a mobile version of Kinect.

1.3 mKinect

In addition to Kinect, as Microsoft has recently introduced compact and affordable Capri sensor, we expect that applying gesture recognition on Windows Phone products is becoming possible. Assuming that gesture recognition for smartphones will be available soon, we started project mKinect, the purpose of which is to bring a gesture recognition technology on Windows Phones, which provides users Kinect experience on their Windows phones and tablets.

With a mKinect-enabled device, the user can trigger commands with motions such as hand gestures and body movements. With this feature provided, Windows Phone developers can apply mKinect to a variety of applications that involve actions such as games, exercises, and healthcare.

We have implemented a prototype of mKinect which mimics the actual device with a computer. As a prototype of mKinect, we mainly target on users with tablet-size devices which are relatively easier for the users to trigger commands with richer gesture recognition compared to smartphones with smaller screens.

1.4 User Interface Issues

As there have been several attempts to implement gesture recognition technology onto mobile devices, several challenges regarding user interaction have been introduced. There are three major user interface challenges while implementing mKinect:

1. Using gesture recognition technology on a smaller screen makes the user hard to receive visual feedback from the application.

2. Since the user can either hold the mobile device or place it on a surface while she interacts with the device, its application needs more than one approach to interact with the user, unlike other typical applications.

3. It is hard for software developers to implement gesture recognition-enabled applications with a decent quality of user interaction due to a variety of screen sizes of mobile devices and lack of the user interface documentation.

Applying gesture recognition technology into a smaller device introduces new user interface issues for feedback to the user due to the smaller sizes of the mobile devices, unlike the larger screen of television when using Kinect. It is hard for users to receive visual feedback from the mobile device screen while using gesture recognition since it requires a certain distance to work motion detection properly.

In addition, since gesture recognition on mobile devices has not been widely introduced yet, there are not many references regarding its user interface. Fewer user interface challenges of 3D gesture mobile devices have been raised compared to the current motion detection technologies, such as Kinect, because they assume that users interact with a big television screen in a certain position of the room.

1.5 Physiotherapy Application for mKinect

We have faced these user interface issues while we implemented a physiotherapy application for mKinect, which simulates a gesture recognition application for mobile devices. The goal of this application is to direct users to perform exercises and give them visual feedback whether their exercises are correct.

1.6 Visual Feedback of Physiotherapy Application

Implementing a physiotherapy application for 3D gesture mobile devices has several user interface challenges.

First of all, unlike most other 3D gesture applications such as Kinect applications which require basic movements and hand gestures, a physiotherapy application aims to accurately correct the user's exercise. Regardless of the accuracy of current consumer products of 3D gesture sensors such as Xbox Kinect, most of its applications do not specify the direction of correct movement if the user's exercise is incorrect.

Figure 1: Performing a yoga exercise in Your Shape Fitness Evolved for Xbox Kinect. It gives the user enough visual instruction for the exercise by displaying joints and bones. It does not correct the user when the exercise poorly done. Excerpted from http://thecontrolleronline.com/2010/11/review-your-shape-fitness-evolved/

For example, Figure 1 shows a demonstration of a yoga exercise of Your Shape Fitness Evolved for

Xbox Kinect. Although it displays check mars to indicate whether the exercise is correct, it does not give the user feedback of how the exercise should be corrected. Rather, the application either gives the user a low score as a result of the incorrect exercise or simply indicates the exercise is incorrect without explaining why it is incorrect.

However, this may not be suitable for a physiotherapy application, the goal of which is to provide the users proper treatments by providing detailed feedback of an exercise. As a goal of the physiotherapy application for mKinect, we have decided to design and implement visual feedback of an exercise accurate enough to be used as a healthcare application.

1.7 Structure of this Paper

Along with these user interface issues and a need for a physiotherapy application, this paper will further explain the details of user interface issues and discuss how the physiotherapy application for mKinect is implemented with a consideration of user interface issues brought up by gesture recognition-enabled devices. Furthermore, it will discuss additional user interface issues discovered during the implementation of the application and suggested solutions.

2. User Interface Issues

2.1 Trends of Mobile Gesture Recognition

There have been a few approaches of user interface guidelines for gesture recognition on mobile devices with several different approaches which are mainly divided into two categories:

1. computer vision-based gesture recognition providing full body motion sensing such as Capri sensor,
2. computer vision-based gesture recognition that provides partial motion sensing such as hand gestures like Leap Motion [17], and
3. non computer vision-based gesture recognition such as tracking muscles like MYO [16].

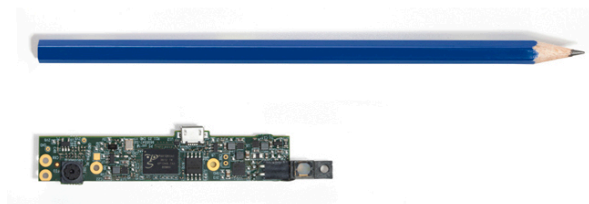

Figure 2: Capri sensor by PrimeSense compared with a pencil. Capri is small enough to be embedded in a palm-size device. Excerpted from http://www.primesense.com/solutions/sensor/

Current similar projects such as Capri focus on reducing performance constraints of power and battery by providing an additional small sensor that can be attached to a device [6]. Figure 2 shows that the Capri sensor is small enough to be inserted into a small device.

On the other hand, Samsung and Pantech's Android-based smartphones support basic hand gestures such waving a hand to the camera in order to receive calls [3, 11].

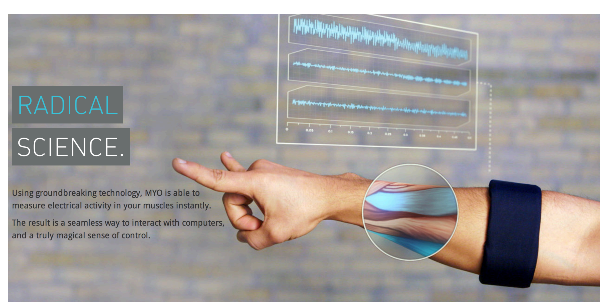

Figure 3: MYO sensor tracks the user's muscle activity and motions from the armband the user wears. Excerpted from https://getmyo.com

Another recent trend of gesture recognition is a wearable armband that tracks the user's arm muscle activity and motions which result in effortlessly reducing the cost of gesture recognition [10].

One of recent trends in mobile devices is to have gesture and retina recognition. It is expected that gesture and retina recognition on smartphones will be popular within the next five years [4].

Among these trends, we assumed that mKinect is a computer vision-based mobile that that tracks full-body motion or hand gesture.

2.1 Limited Visual Feedback

Since the size of the screen of a mobile device is smaller, providing visual feedback to its user is harder than regular devices such as televisions and computers. In addition, before a device has a gesture recognition feature, the user interacts with the device at a short distance unless he or she uses a remote controller. However, motion detection on mobile devices introduces a new way for users to interact with motion gestures at a longer distance in an application that requires the user's full body. Since the distance between the mobile device and the user may be farther than the possible distance for him or her to read the texts on screen, it is crucial for the device to provide users a better visual feedback.

Since the texts on the screen are not as readable as the texts in the regular use, the size of the font should be increased. As a result, displaying the number of words on screen becomes limited, along with distraction to other objects on screen. In order to effectively use limited amount of space for displaying texts, Rapid Serial Visual Presentation should be considered. RSVP is a technique that displays texts word-by-word in a fixed location. Users can understand as many as 12 words per second with RSVP. [2] Increasing the size of the font and applying RSVP will improve readability.

In addition, using symbols and animations reduces the amount of text as long as it provides clear meaning to the user. Intuitive symbols are easier to read than texts. For example, arrow symbols indicate a movement from one position to another.

2.2 Various Postures to Interact with Mobile Devices

Unlike the game consoles which expect users to be at a certain fixed location to control, mobile devices can be used while users hold the devices or place them on tables. As a result, the motion detection-enabled device may view the user's full body, hand only, face only, or upper body only. In each case, the user may have different expectations of how the mobile device is supposed to work. Therefore, the mobile device should take each case separately when it comes to user interaction.

When the user's face is only targeted through the camera, it means that the user wants to use motion detection with his or her head or facial expression. In this case, detailed motion from the facial expression should be focused for better user interaction. On the other hand, sensing the full body through the camera means that the user is going to move with wider range but simpler motion than facial expression such as lifting an arm. In this case, motion detection algorithm for detailed motion can be disabled for optimization. Especially in this mode, the user is farther from the device than other modes, so the user interface should have more vivid visual feedback.

In order to improve the user interaction in these several modes of motion detection on mobile devices, the documentation of the mobile device must clearly state which modes the mobile device has. Depending on usage, when the mobile device is near to the user, he or she would like to use facial recognition or hand gesture. In this case, the skeleton algorithm for body motion tracking is not needed. If the mobile device is far from the user, it means that he or she would like to use the entire body or upper body. In this case, the user interaction guideline applies the same rules in Kinect Human Interface Guidelines, except for visual feedback in the smaller screen.

2.3 Improving the Quality of Applications

Since gesture recognition directly related to the user's movements, it is important for developers to focus more on user-friendly application by enhancing the quality of user experience. In order to gives them proper direction to better quality of user interaction, the providers of gesture recognition-based mobile devices should provide an easy application programming interface (API) for the presentation layer so that developers provide user-friendly interface without much knowledge of computer vision or graphics.

In addition, the mobile device providers should emphasize the importance of Model-View-Controller (MVC) so that developers can maintain one model with several views. Since one application has several modes for user interaction, it needs more than one view, a way of display based on the distance between the user and the device. Without maintaining the MVC structure, it is possible that the model can be unnecessarily duplicated along with multiple views, which result in more cost in software maintenance.

3. Visual Feedback for Physiotherapy Application

As a goal of the physiotherapy application for mKinect, we have decided to design and implement visual feedback for its exercises accurate and intuitive enough for the application to be used as healthcare software. While designing the visual feedback, we have faced several user interface challenges:

1. how to effectively display the user on the screen and the user's body movement, and
2. how to provide the user visual feedback accurate enough for the user to correct his or her incorrect exercise.

3.1 Avateering vs. Depth Projection of User

For displaying the user on the screen, we have thought of two major methods: avateering and depth projection. Avateering is a method of displaying the user by reconstructing the body into another graphical 3D human form on the screen so that the user can manipulate the avatar.

On the other hand, depth projection of the user is a method of displaying the depth information of the user with the attached depth sensor in mKinect. As a result, the depth projection looks similar to the user's shadow.

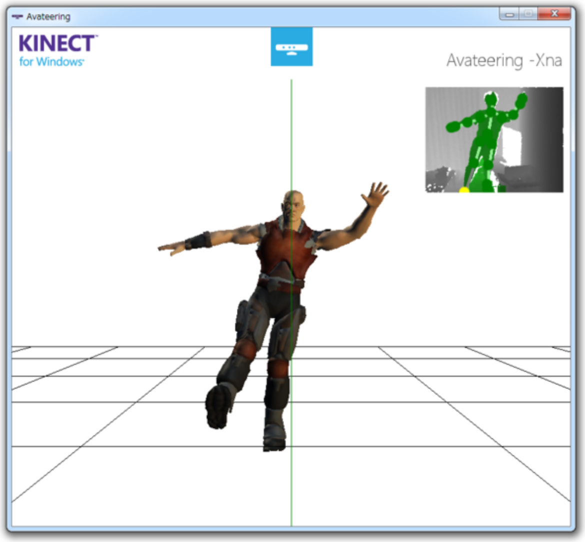

Figure 4: Avateering with Kinect sample code; The user is mapped to an avatar on the screen and can manipulate it. Excerpted from Avateering - Xna sample code of kinect for Windows.

Figure 4 shows a demonstration of avateering with the sample code on Kinect for Windows. Based on the user's movements, the kinect sensor tracks the user's joints and the application applies the same movement to the avatar. Avateering is a widely-used technique in action-based applications, especially video games.

We have found that manipulating an avatar with a Kinect sensor has several disadvantages for the physiotherapy application. Due to expensive calculation of avatar manipulation, the user experiences some delay from the screen. In order for the application to provide accurate visual feedback, the avatar should promptly copy the user's movements.

The delay of avateering results from filtering and refining the raw skeleton joint positions of the user in order to reduce the problem of models stretching, which occurs if the avatar size is different from the user's [12]. In addition, avateering does not fully copy the user's movements, lowering the accuracy. For example, avateering for Kinect does not track the user's detailed movements such as fingers. Although finger movements may not be used in the physiotherapy application, it reduces the quality of user experience because all the user's movements do not correspond with the avatar's movements.

Figure 5: Depth projection of the user. The depth is represented as a color, where the same color of two different areas mean that they are in the same range of depth like a contour graph.

As an alternative, displaying the depth projection of the user is suggested like a shadow. Figure 5 demonstrates a depth projection of the user using Kinect. Since depth information is directly received from the Kinect's depth sensor, the user's movements are accurately displayed on the screen so that the user can easily recognize the way he or she moves.

3.2 Visual Feedback of Incorrect Exercise

In order to for the physiotherapy application to provide accurate visual feedback to the user, two main user interface issues have been raised:

1. How much information of joints and bones of the user should be displayed to

improve the visual feedback of the user's movement?

2. How to display symbols for the indication of an incorrect exercise intuitively yet accurately?

Figure 6: Displaying a sub-skeleton of the user. The application renders the user's right arm.

We have found that displaying joints and bones of the user provides the user great feedback on the user's movement since the user can intuitively view the angles of two adjacent bones. However, displaying the full skeleton of may confuse the user since it is more feedback than the user would like to know.

For example, an arm exercise only contains a set of arm movements while other parts of the body are static. Like the sub-skeleton of the right arm in Figure 6, displaying only the skeleton of interest provides concise yet accurate visual feedback to the user. As a result, we have decided to only display the sub-skeleton of interest of an exercise.

However, displaying only an incorrect sub-skeleton raises a new user interface issue. If the sub-skeleton is not displayed, the user does not know whether the exercise is correct or the body is not being recognized. Even though an incorrect sub-skeleton is displayed, the user may not know whether Kinect correctly recognizes the body. The Kinect sensor may not recognize the user in certain circumstances. For example, it is harder for Kinect to track a body if there is an object placed in the similar depth of the body. In addition, since Kinect uses an infrared sensor to track depth information, Kinect may not work in a place with sunlight [18].

As a result, we have decided to display the full skeleton using several colors. In Figure 7, correct skeletons are colored as dark gray and an incorrect skeleton is colored as red to get the user's attention.

Figure 7: Visual feedback of an exercise with highlighting an incorrect sub-skeleton and an arrow. The direction of the arrow indicates the direction of the incorrect joint to be placed to perform a correct exercise. The whole skeleton is displayed with dark gray color and an incorrect sub-skeleton is red.

The other user interface issue for the physiotherapy application is to provide effective visual feedback to the user accurate enough to correct the incorrect exercise. At first displaying each exercise step as another skeleton on top of the user's depth project was suggested, so that the user can simply direct their body to the skeleton.

However, following a skeleton to perform an exercise may not be good for user experience since users are more familiar with following an instruction with an instructor rather than a skeleton. So we have decided to use a set of images of instructors on top of the screen, displaying one by one for each step of an exercise.

Figure 8: Exercise instruction as a step-by-step image of an instructor on top left of the screen.

In addition to step-by-step exercise instruction images, we have realized that the visual feedback should be conveyed as a 3D form. To effectively correct the exercise in terms of joints and bones, displaying an animated arrow around an incorrect joint is suggested. Every movement of an exercise is a sequence of movements of joints and bones. A movement can be divided into changes in angles of one or more joints. Based on this, we have found that providing visual feedback on incorrect joints is intuitive and accurate. Figure 7 demonstrates feedback arrow on joints of the red sub-skeleton. Since the arrow points to top-left, it means that the arm should be rotated counter-clockwise where the axis is the joint of the shoulder.

Since the application displays one arrow for an incorrect sub-skeleton, the current user interface problem cannot display more than one incorrect sub-skeletons. In order to resolve this issue, we have decided to use several colors for incorrect sub-skeletons. In Figure 9, incorrect sub-skeletons are colored as yellow and red. Yellow indicates that the sub-skeleton is incorrect and red means that the sub-skeleton is the most incorrect so that it has to be corrected. The feedback arrow on the screen corresponds to the red sub-skeleton. The application calculates the angle of between each of the user's bone and the corresponding bone of the skeleton of the correct exercise. The sub-skeleton that has the largest angle becomes the most incorrect sub-skeleton.

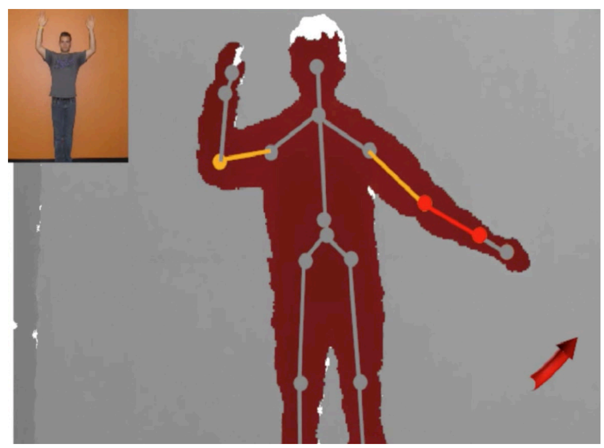

Figure 9: Yellow and red indicate incorrect sub-skeletons and the red one is the most incorrect sub-skeleton that has the largest incorrect angle among all the incorrect sub-skeletons.

4. Result

We have demonstrated this application on May 8th of 2013 to public and received feedback. Firstly, users do not fully understand the meaning of the colors of sub-skeleton. We believe that this is because the screen displays only one arrow while there may be multiple incorrect sub-skeleton. In order to resolve this issue, we have designed an alternative user interface for future implementation. As described in Figure 10, we display multiple feedback arrows next to the incorrect joints.

Figure 10: Displaying an incorrect sub-skeleton and arrows in each incorrect joint as an alternative visual feedback.

5. Conclusion

With the advent of motion detection on mobile devices, it raises issues of user interface with the devices. These are the lack of visual feedback to the mobile device user, the need for various ways to interact with users for better user experience, and the possibility of developers to implement applications with lower quality of user interaction. These can be resolved by introducing effective visual feedback overcoming the small size of mobile device screens, using audial feedback, providing a detailed and structured documentation for the better user interaction, and providing an easy API for developers to easily implement user interface of the applications.

If motion detection actually implemented onto mobile devices, it may introduce unexpected issues that can hard be expected. Thus, it is crucial to examine details of minor but potential issues. For example, one of them is diverse types of cameras on mobile devices. Current models of smartphones and tablets have both front and back cameras. They have a variety of models which have different resolution and lens sizes. These several factors may incur inconsistency of the motion detection technology.

In addition, we should be aware that the user interaction with motion detection on mobile devices are different from that in the current technology such as Xbox Kinect. Due to the different settings and uses, users may expect different ways to interact with the devices.

When it comes to visual feedback of the physiotherapy application, depth projection of the user is used for displaying the user on the screen. For better visual feedback of correcting an exercise a sub-skeleton of interest in an exercise is displayed along with 3D feedback arrows on joints one-by-one. Rather than displaying the several arrows, each arrow is displayed one-by-one starting from parent joints to child joints to make the feedback consistent.

Acknowledgement

I would like to firstly thank my advisors, Chin Chun Ming and Professor Bjoern Hartmann, to give us mKinect members great directions of technical advice. I would also like to thank Professor Jitendra Malik who firstly brought the user interaction problems due to the camera and small screen size of mobile devices. In addition, I appreciate the advice of Wooram Youn in Microsoft Windows Phone Department, who gave an insight of the current technical trend of Windows Phone. Lastly, I would like to thank the team members Ross Yeager, Aarthi Ravi, Hassan Elahi, and JJ Liu who help me achieve one of recent technical challenges to push technical innovation forward.

References

1. Human Interface Guidelines - Kinect for Windows v1.5.0, Microsoft (2012), 6-7.

2. J. Wobbrock, J. Forlizzi, S. Hudson, B. Myers, Evaluating Readability on Mobile Devices, ACM (2002)

3. Samsung squeezes Kinect Tech into a mobile camera: Movement-sensing Android incoming? http://www.electricpig.co.uk/2012/03/01/samsung-squeezes-kinect-tech-into-a-mobile-camera-movement-sensing-android-incoming. Retrieved 9/2012

4. V. Atluri, U. Cakmak, R. Lee, S. Varanasi, Making Smartphones Brilliant: Ten Trends, McKinsey & Company (2012), 3-4

5. Touch-Free Gesture Recognition for iOS and Android http://www.nanocritical.com/nanogest/. Retrieved 1/2013

6. Hands-On With the Next Generation of Kinect: PrimeSense Capri http://spectrum.ieee.org/automaton/robotics/robotics-hardware/handson-with-the-next-generation-kinect-primesense-capri. Retrieved 2/2013

7. Program Information and Costs, Clear Passage Physical Therapy, http://www.clearpassage.com/before-your-visit/program-costs/. Retrieved 3/2013

8. Physical, Occupational, & Speech Therapy, Metro Health, http://metrohealth.net/about-metro/quality-pricing/physical-occupational-therapy/physical-occupational-therapy-prices/. Retrieved 3/2013

9. Patient Price Information List, Akron General, http://www.akrongeneral.org/portal/page/portal/AGMC_PAGEGROUP/Price_guide/PRICE_GUIDE6. Retrieved 3/2013

10. The Next Generation of Gesture Control, MYO, https://getmyo.com. Retrieved 4/2013

11. Pantech add 3D gestures to Vega LTE smartphone, Will Shanklin, http://www.geek.com/mobile/pantech-adds-3d-gestures-to-vega-lte-smartphone-1435961/. Retrieved 1/2013

12. Avateering C# Sample, Microsoft, http://msdn.microsoft.com/en-us/library/jj131041.aspx. Retrieved 4/2013

13. Samsung Galaxy Beam Features, http://www.samsung.com/global/microsite/galaxybeam/feature.html. Retrieved 4/2013

14. Small in size, big on power: New microbatteries a boost for electronics, http://news.illinois.edu/news/13/0416microbatteries_WilliamKing.html. Retrieved 5/2013

15. Samsung Galaxy S III: Everything you need to know, http://www.pocket-lint.com/news/115350-samsung-galaxy-s3-specifications-highlights. Retrieved 5/2013

16. MYO - The Gesture Control Armband, https://www.thalmic.com/myo/. Retrieved 5/2013

17. Leap Motion, https://www.leapmotion.com, Retrieved 5/2013

18. MSDN - Kinect Sensor, http://msdn.microsoft.com/en-us/library/hh438998.aspx, Retrieved 5/2013