User Interface: 3D Feedback

Seyed Hassan Elahi

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-105
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-105.html

May 17, 2013

Abstract

In this project we examine the possibilities of bringing the gesture recognition to Windows Phone and Surface tablet devices by integrating Kinect to the aforementioned devices. We will demonstrate this by designing a physiotherapy application.

In physiotherapy exercises, it is very important and critical to the patient to do the exercises as accurately as possible. So our physiotherapy application should have the ability to give feedback to the users in order for them to do the exercises correctly. For this purpose, we decided to render 3D arrows that show users how to adjust incorrectly positioned limbs. One way to do this is to use 3D animation on top of the skeleton image from the Kinect, so the user will know how and how much he or she should change the angle of the wrong joint to correct the position. This paper presents an approach for modeling shapes in an open source 3D application, Blender, and subsequently importing and animating those shapes in an interactive application created in the Microsoft XNA framework.

1-Introduction:

In recent years, gestural interaction has received great attention in the computer industry and any related fields. Improvement in the size, power, and cost of microprocessors, memory, cameras, and any other sensing devices make it possible now for people to interact with their PCs and devices through body movements, touch, or voice [1].

Kinect, a depth-sensing camera, was introduced by Microsoft as a new human-computer interaction technology, initially targeting computer games. Soon after that, it became clear that there are other areas that can exploit such a user-friendly device. Nowadays, institutes are using Kinect in their educational programs and doctors are improving their treatments by means of monitoring capabilities that Kinect offers them[2].

Currently, Kinect applications require a dedicated PC or game console, and a separate display. We foresee a future in which depth sensors can be integrated into mobile devices."

The smartphone and tablet industry is growing very fast. ”As Pew Research Center reports in January of 2012, 45 million U.S adults owned tablets. By August, the number had risen to 59 million, or 25 percent of the adult population. That represents growth of 32 percent in just 8 months. In the same period, the number of U.S. smartphone owners grew at a more modest 15 percent to 117 million [3]”. We cannot completely replace Pcs with smartphones and tablets for most applications and businesses. Since the popularity of tablets is increasing and also smartphones are getting exceedingly prevalent, the companies are trying to change their strategies and pay more attention to these devices [4]. So those mobile and tablet companies who can adapt motion detection and gesture recognition to their product can have significant control in this large industry.

In this project we want to make an application that uses Kinect for Windows motion tracking technology in physical therapy.

In physiotherapy exercises, it is very important and critical to do the exercises as accurately as possible. From the point of view of Kinect and skeleton data, this would be considered as the right angles of the joint of the patient (user) body. In the physiotherapy application, after showing the exercise to users, we need to give them feedback to show them whether they are doing the exercise right or wrong. In order to give feedback, we need to display which joint's angles are wrong, then the user will find out how to correct that joint and how to do the exercise as precisely as possible. To show feedback, we decide to animate the arrows to help the user change the angle of the wrong joint to the correct position.

2-Litrature Review

2-1 Kinect

Kinect is a device with the ability of capturing, tracking and analyzing human voice and body movements in order to use them as commands to interact with digital units integrated in games or other computer programs. To put it differently, users are not restricted to keyboards, mice or joysticks anymore to control the software, offering them more real and sensible experiences [5].

Figure 1. Kinect Sensor

As announced by Microsoft, over one million Kinect sensors were purchased within first 10 days after its original launch in the US on 4 November, 2011 [6]. In spite of the fact that it was originally aimed to be used along with Microsoft Xbox 360 for entertainment experiences , the technology beyond Kinect showed its diverse potential in being used in real-world applications such as digital signage, virtual shopping, health related services and education [7].

For instance Kinect has the potential to become an essential technology in schools in a few years' time. Bearing in mind that Kinect is able to track each user individually; it gives both teachers and students the opportunity to control the learning materials using their body gestures and voice instead of the more inconvenient wired or wireless devices [8].

Additionally, attempts have been made to utilize Kinect technology in healthcare systems in order to assist doctors to track and tackle medical issues in better and more efficient ways. For example, a ‘’virtual healthcare app’’ has been designed which uses Kinect to monitor the patient movements in 3D, providing doctors with more data [9].

2-2 How Kinect works

Although the fact that Kinect looks like a single piece of hardware, various distinctive technologies have been integrated in this device to support the following features (see Figure 2)[10]:

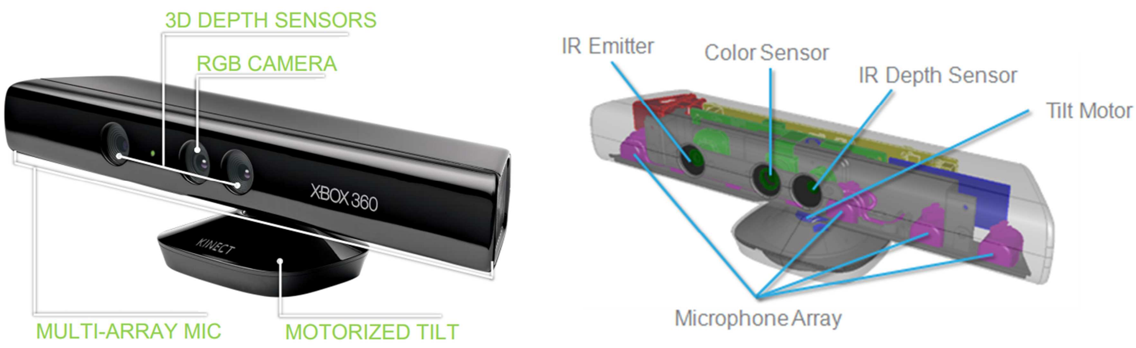

Figure2. Kinect with sensors [10]

- **Depth image**: The invisible light which is radiated in to the play space by an infrared emitter is captured by a CMOS sensor and then, depending on the way it’s reflected back, is processed as a grayscale image, in order to determine the scene depth to detect any movement in 3D space [10].
- **RGB image:** “An RGB camera is being used in Kinect for facial recognition, in-game snapshots and video chat” [10]

- **Tilt (Get and Set):** Depending upon the user's height and their distance the sensor is being tilted up and down automatically by a motor [10].
- **Microphone Array:** By four integrated capsule microphones along with an audio processor, Kinect not only can remove the background noise of the incoming voice, but also is able to determine its direction [10].
- **Skeleton Tracking**: Due to large amounts of data provided to algorithms which were developed by Microsoft, Kinect is able to track the location of 48 skeletal points in the user's body [10].

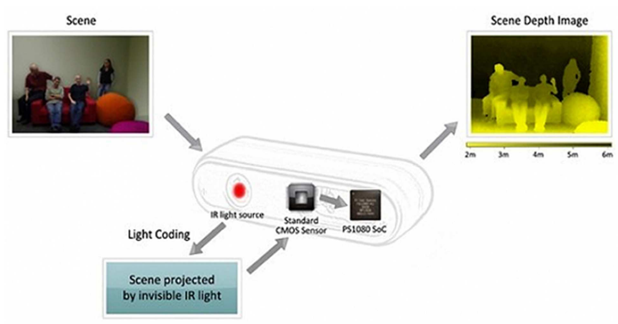

Figure 3: Depth and RGB Images [10]

2-3 Visual Feedback

There are several instructional Kinect applications in areas like Yoga, fitness or dancing, where Kinect is used to teach users and show them how to do exercises. In most of these applications, they use a human model (avatar) as instructor and this avatar will show the user how to do the exercises. Figure 4 shows some examples of using avatar as instructor.

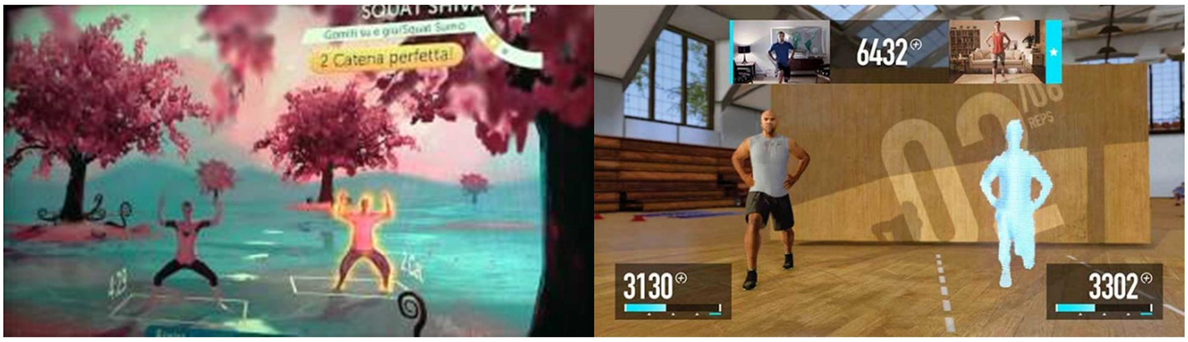

Figure 4: Avatar as an instructor

The purpose of using avatars in these applications is to teach the user how to do the exercises. So, the user will get some feedback if he or she is doing the exercise correctly or not. For example as shown in Figure 5 in “Nike + Kinect Training”, if the user is doing something wrong the color of his/her shadow will change and an instruction will pop up; or in “Kinect Yoga” the color of the wrong bone of the user will change (Figure 6).

Figure 5: Visual feedback “Nike + Kinect Training” application

Figure 6: Visual feedback “Kinect Yoga” application

By using Kinect we are dealing with 3D environments, so it is very important to give precise feedback to the user. Especially in physiotherapy application it is more critical, because if the users do the exercise wrongly they may hurt themselves.

3-Methodology

3-1 Introduction:

Kinect with its depth camera gives 3D position of the user's joints. In order to give a real feedback to the user, we also need to show the user the right directional guides of the wrong joint position in 3D. One way to do this is to use 3D animation on top of the skeleton image from the Kinect, so the user will know how and how much he or she should change his/her joint.

There are several ways to make an animation in 3D environment but we want our 3D model to interact with its 3D environment so that we could move parts of our 3D model through our code [11].

3-2 Blender:

There are several 3D modeling and animation applications available such as Anim8or, Autodesk 3d Max, Blender, Autodesk Maya. In this project, we used Blender as the modeling software which is one of free and open-source ones. Blender is 3D computer graphics software which is used for creating animation, visual effects, video games or 3D applications [12].

3-3 XNA:

For using the animated feedback we need to import the Model to graphical environment. "The XNA Framework provides support for both 2D and 3D model and allows use of Kinect". Since we want to animate a 3D model (the arrow) in our application we decided to use XNA as our graphical environment [13].

3-4 Background:

What makes a 3D game really attractive is the continuous movements of objects. The question that pops up here is how these items move in 3D games. It could be explained by how a person moves his body by using his bones. We can assume that an animated human has bones which can be moved separately while attached to each other. It should be noted that there are certain rules which are dominant in the movement of a human skeleton's bones. However, these rules need not necessarily be abided by in an animated human model in a 3D game. Due to the fact that bones should not be seen in an animated model, there is no need to put them inside the body as they are in the human body. Furthermore, the model is not often as much complex as a human body, so fewer of bones could be used make an animated model to move [13].

In order to make an arrow in XNA, we need to do the following tasks:

1. Make model in Blender (an arrow)
2. Add a Skin to the Mesh
3. Add Bones to the Model
4. Show the Mesh how to deform when the Bones move
5. Export the Mesh, Bones, and Skin in a format that can be used in XNA
6. Bring the Model into XNA
7. By using a code, move the Model through moving the Bones [13]

Also, in order to be able to use the model in XNA, we need to consider the following requirements:

1. A Mesh must have at least one Bone

2. Each vertex in our Model should be bound to at least one or more Bones

3. A Skin should be added on our Mesh [13]

3-5 Step One: creating the mesh

Figure 7 shows the initial view when getting Blender started. Blender starts with a cube, camera, and a light.

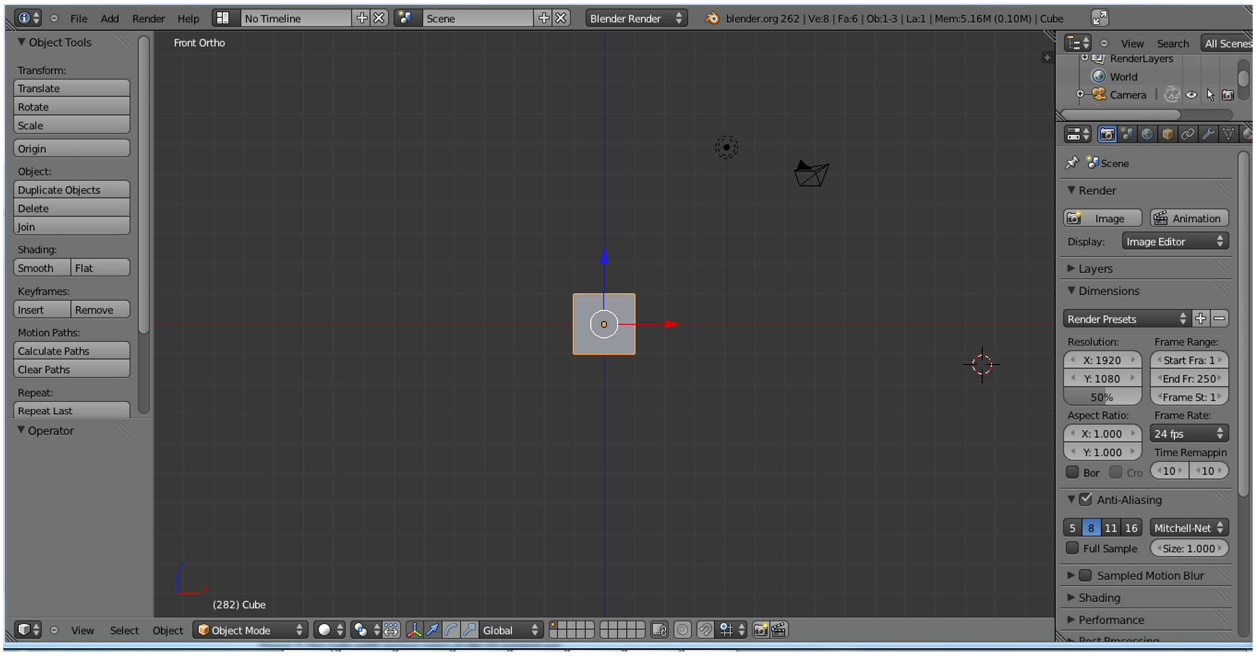

Figure 7: Initial view in blender

While XNA framework works on a right-handed coordinate system and DirectX works on a left-handed coordinate system (Figure 8), Blender works neither on a right- nor on a left-handed coordinate system [14].

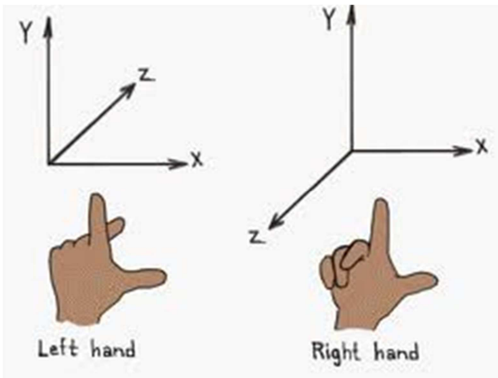

Figure 8: Left-handed and Right-handed coordinates

In Blender, the axis alignment is rather unique because the x-axis points to the right, the y-axis to the front, and the z axis vertically. This coordinate system has been shown in figure 9 [14].

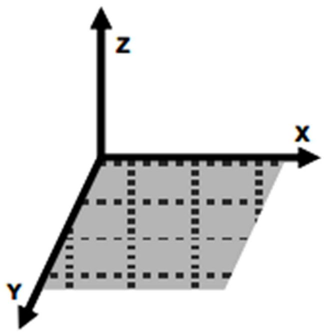

Figure 9: Blender axis alignment [14]

The Blender Model could be easily exported to XNA if we fit its coordinate to the right-handed coordinate system. In order to do this we can rotate the x axis to the right, the y-axis upward and z-axis to the front (as illustrated in Figure 10).

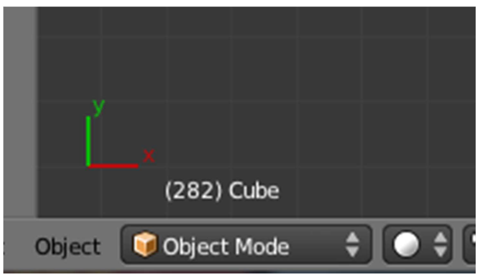

Figure 10: Right-handed coordinate system in Blender

It is very important that all the model objects (meshes) and the armature are centered at the same location, ideally zero (X = 0.0, Y = 0.0, Z = 0.0). Also the model objects must have a scale of 1.0 and not use rotation.

For making an arrow, we have to delete the cube and add a Torus. Figure 11 shows the Torus in Blender.

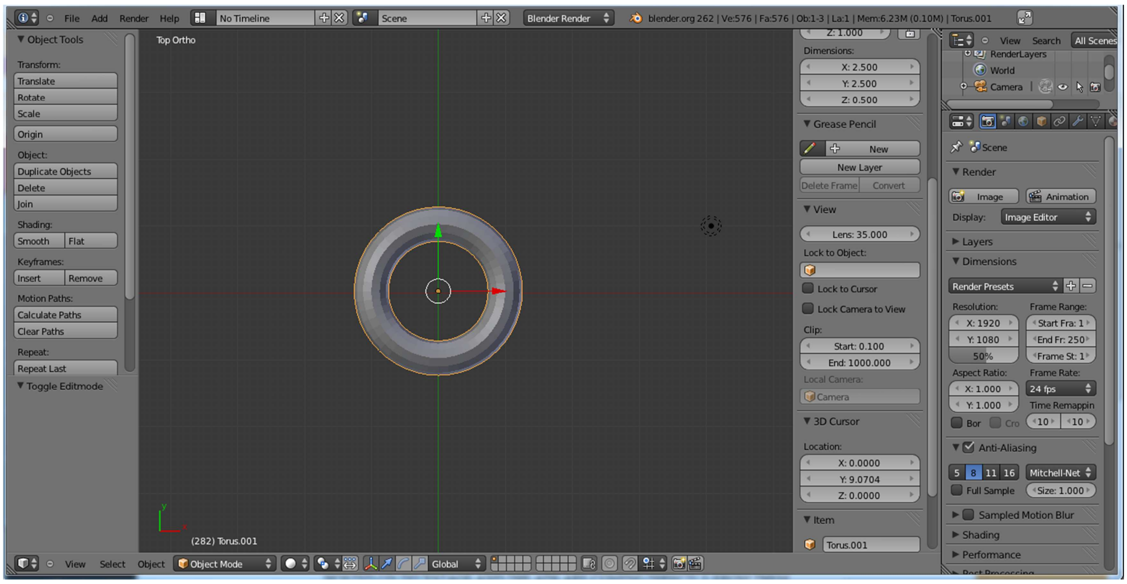

Figure 11: Torus in Blender

Each mesh in Blender has its own control parameters by which you can customize your mesh. For example, for the Torus we have a major radius control parameter which defines the main radius of the torus, a minor radius control which defines the minor radius of the torus. Minor and major segment parameters specify how many segments are used to define both radii. Figure 12 shows the Torus with its control options.

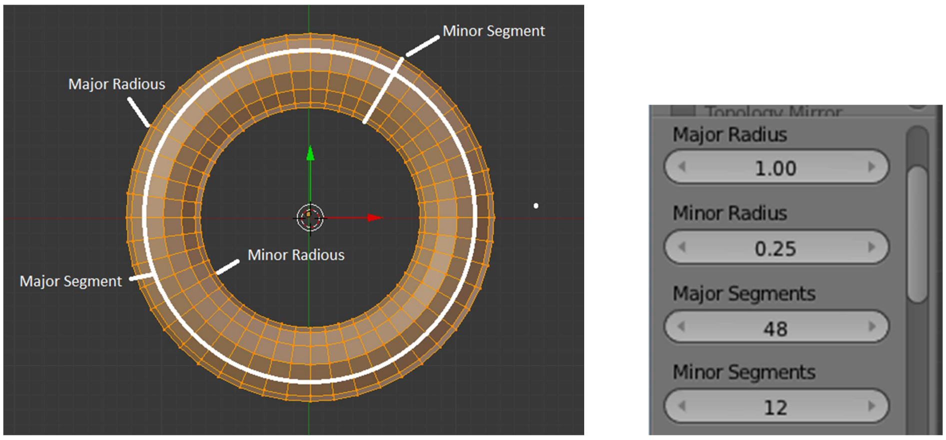

Figure 12: Torus with control parameters

Figure 13 illustrates the steps to creat the arrow.

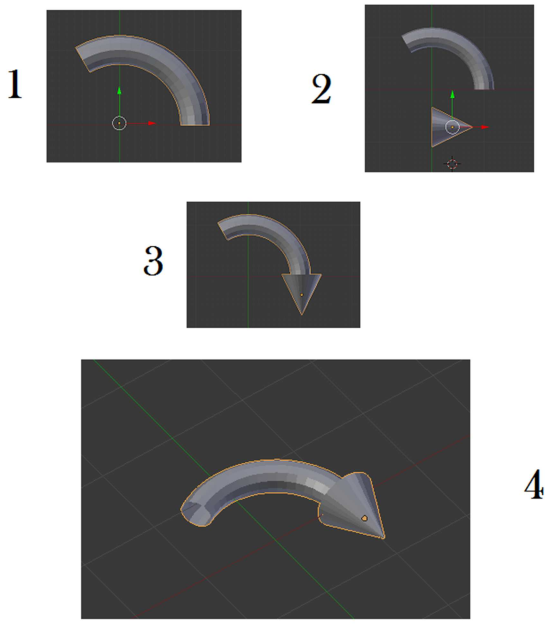

Figure 13: 3D arrow

3-6 Step Two: Add skin to the Mesh

3D models usually have texture and in order for our arrow to work in XNA, it needs to be wrapped with some skin. If the model has texture, 2D images can be added on its surface. We can create Texture in Blender in two ways:

1- Texture can be directly created in Blender

2- Texture can be created in other Graphic software like Paint or Photoshop.

The method used to create texture is called UV-mapping [13]. In UV-mapping we flatten the 3D surface of our Mesh into 2D so that we can edit and modify it with graphic software. However, we need to show to Blender how to unwrap the Model. In order to do this, we require defining seams along our model so Blender can use them to pull the arrow apart. We can define as many seams as we want on the Model in order to break the model down into small pieces [13].

The resulting 2D image (right) and the corresponding 3D model (left) can be seen in Figure 14. "Each vertex has 2D coordinate. These are referred to as UV-Vertices or 'UVs' and this process in Blender is called 'unwrapping'" [14].

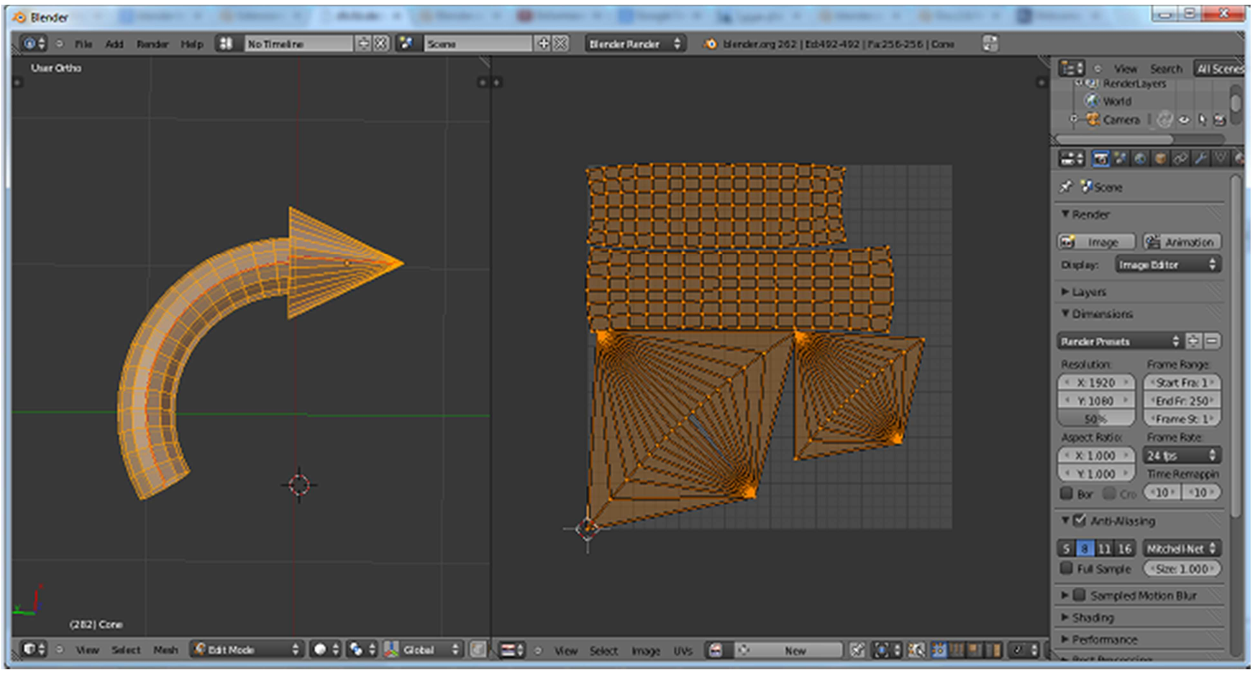

Figure 14: How to flip a mesh along a pre-defined line.

We can also export this UV map to other formats like TGA, in order to make further modification on the UV map in other graphic software products. We can then open the newly created texture in Blender and apply it to the Model. The 2D image on the right hand side of Figure 15 has been applied onto the model on left. Furthermore, in order to export the Model to XNA we need to save the 2D image for future use in XNA [14].

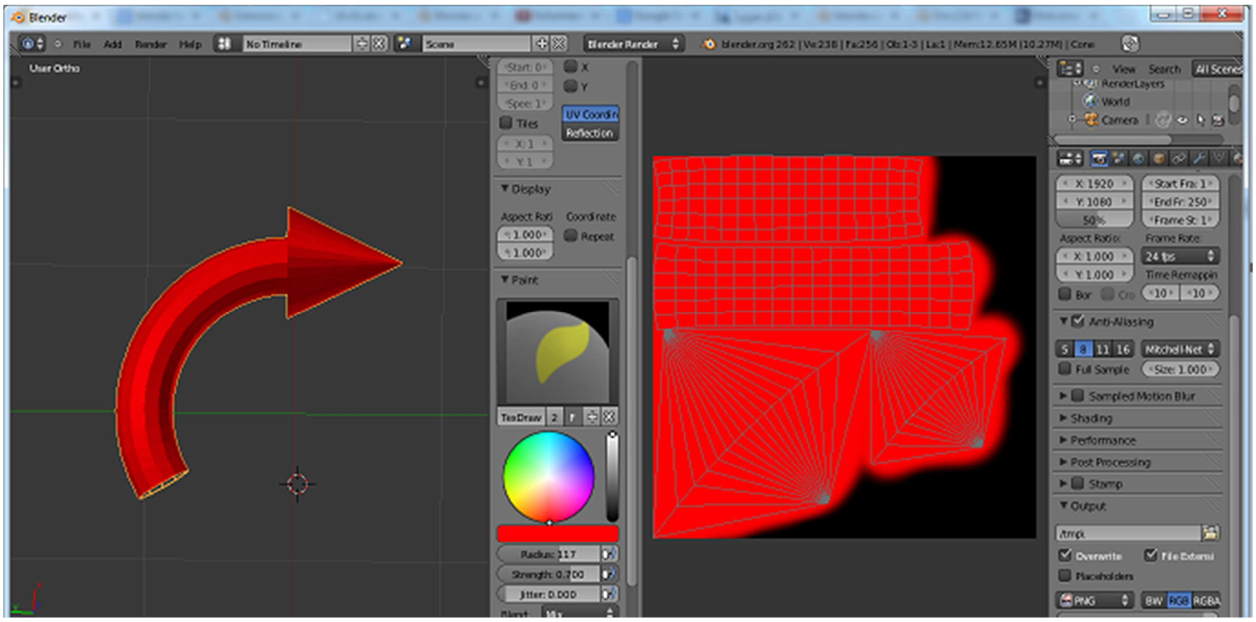

Figure 15: The 3D model and its texture in Blender

3-7 Step Three: Create Bones for the Model:

First we need to find the center of the arrow (center of the initial Torus) which should be at the zero location (X = 0.0, Y = 0.0, Z = 0.0). We can then add an Armature (Shift 'a' -> Armature -> Single bone). Figure 16 shows the arrow with one bone.

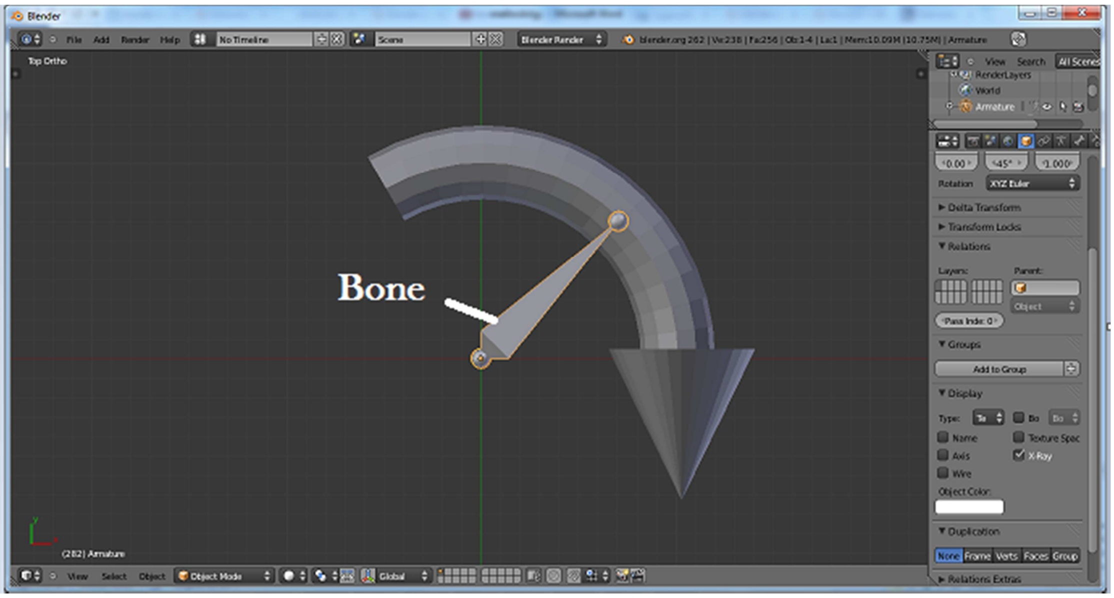

Figure 16: The 3D model with Bone

It should be noted that in order to find the bones in XNA and to manipulate them, it is very important to name the Bones.

3-8 Step Four: Binding the Arrow to the Bone

In order to deform the mesh by the Bones (Armature) we need to bind them together. First we have to select the arrow, then the Armature (Bones) and subsequently ”parent” the two items together. There are several ways to parent the Mesh to the Bones. The easiest option is “With Automatic Weights”. In the ”With Automatic Weights” option, Blender will assign weighting to each vertex proportional to the location of that vertex to the Bone (Figure 17). We could also do weighting manually by choosing the “Weighted paint” option [13].

To animate the Model in XNA, it is necessary to bind every vertex to at least one Bone. Therefore, when we want to load our Model to XNA, if we miss even one vertex, we will receive an error message.

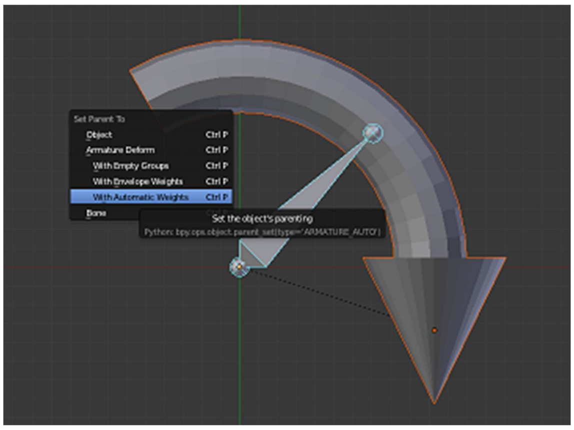

Figure 17: Parenting our arrow to our bones with 'Automatic Weights'

At next stage, we need to export the Model as Autodesk FBX (".fbx") in order to use it in XNA. We now have two files:

1- Skinning image file

2- .fbx file.

Now our Model and texture file are ready to be imported into XNA. To use the Model in our application, it must to be imported through the content pipeline. In order to import the model and skinning image files to XNA, first we need to add them into a project. By using the 'Solution Explorer' window in the XNA Game studio we can import the Model. Figure 18 shows the arrow in XNA.

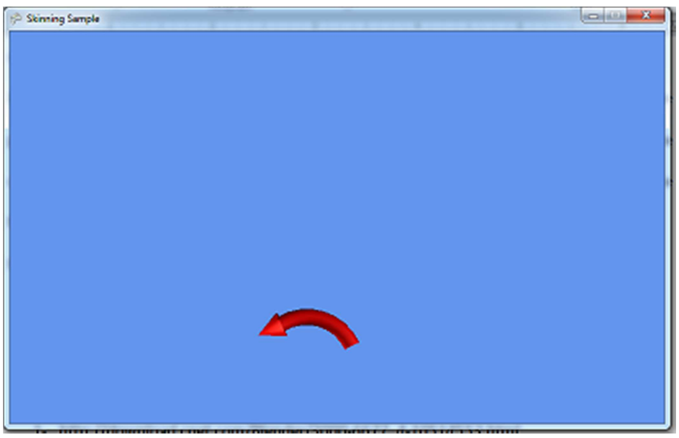

Figure 18: Arrow in XNA

By means of a piece of code we can move the Model through moving the Bones. Each Bone has 3 parameters that make it possible to rotate it from the joint in all the 3D dimensions. Our arrow has only one Bone letting us to rotate it in all directions. Figure 19 illustrates the rotation of the arrow around X-axis.

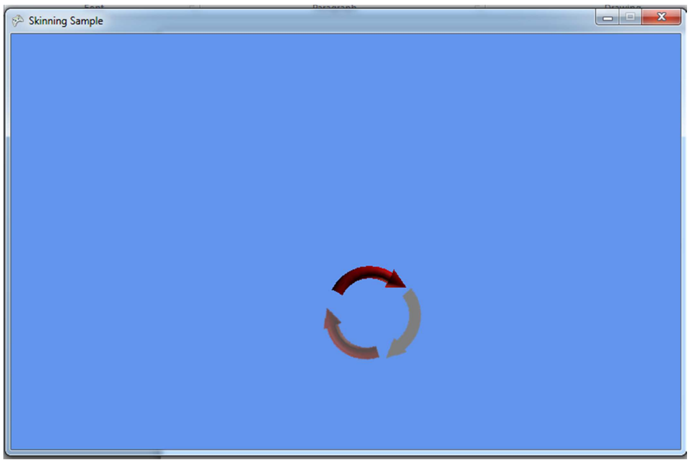

Figure 18: Rotation of arrow in XNA

XNA not only provides tools to move and rotate the Models in a 3D manner, but also enables the user to change the point of view (camera direction) toward the Model. Also the user can zoom in and out from the current view. Figure 20 illustrates the zoom in and out of the arrow.

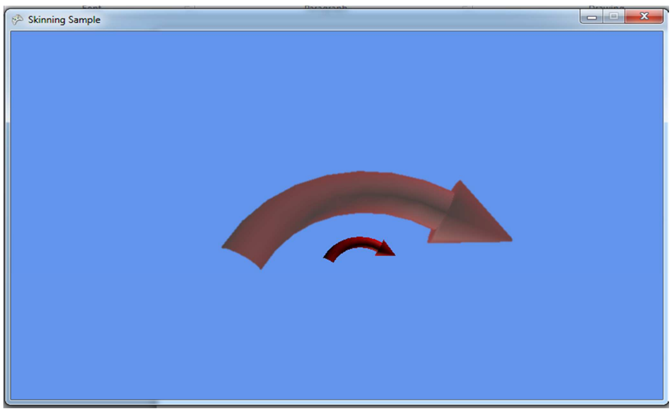

Figure 19: Zoom in and out in XNA

Next, this arrow should be integrated in the main physiotherapy application. In the environment of this application, whenever the user does a wrong movement, the application will detect the bone which is most responsible for the wrong pattern. Then, the arrow will show the correct direction in which the bone should move to correct for it. Figure 19 illustrates the arrow integrated in the main application.

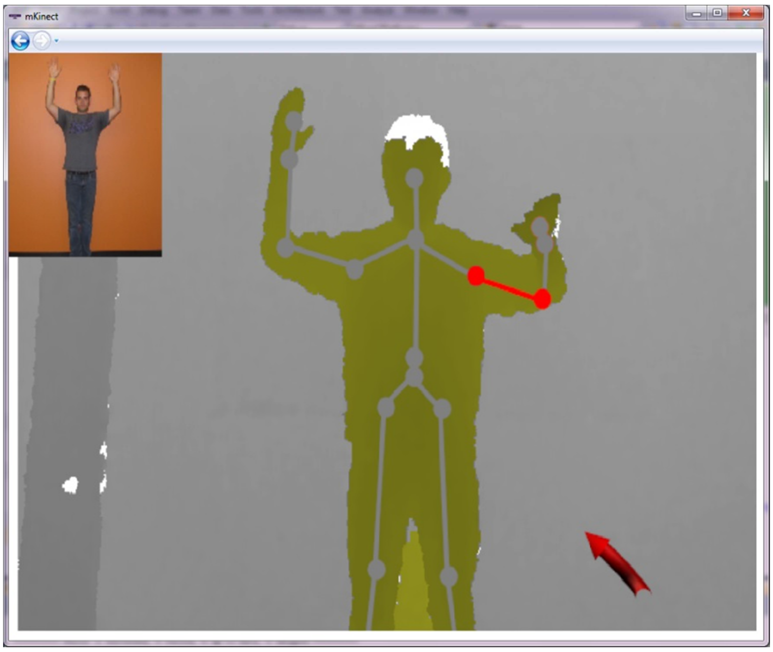

Figure 19: Arrows in the main physiotherapy application

4- Conclusion

The aim of this project is to make gesture recognition accessible on mobile devices. In order to examine its possibility, we made an application that uses ‘Kinect for Windows’ motion tracking technology in physical therapy. The program would provide interactive feedback and educational materials to patients.

In physiotherapy exercises, it is very important and critical to do the exercises as accurate as possible. So we need to give feedback to the users to show them whether they are doing the exercise right or not. In order to give a real feedback to the user, we also need to show the user the right directional guides of the wrong joint position with 3D arrow. This paper has dealt with the creation of 3D arrow in the Blender software tool and importing such models to the free XNA Game Studio tool.

References:

[1] Norman, Don, “Natural User Interfaces Are Not Natural”. Retrieved 11/2012 from http://www.jnd.org/dn.mss/natural_user_interfa.html

[2] “The Kinect Effect How the World is Using Kinect” Retrieved 11/2012 from http://www.xbox.com/en-CA/Kinect/Kinect-Effect.

[3] “The Growth in U.S. Tablet and Smartphone Owner”, Retrieved 11/2012 from http://www.iclarified.com/26376/the-growth-in-us-tablet-and-smartphone-owners-chart.

[4] Hamblen M.,” Smartphones and tablets growth exploding, especially in business, Gartner says”. Retrieved 11/2012 from http://www.computerworld.com/s/article/9233305/Smartphones_and_ tablets_growth_ exploding_especially_in_business_Gartner_says

[5] “Children Development with Kinect for Xbox 360”. Retrieved 10/2012 from http://kinectandphysicaleducation.weebly.com/social-development.html

[6] “Kinect”. Retrieved 10/2012 from http://en.wikipedia.org/wiki/Kinect#cite_note-releasedate-2

[7] Rouse, Margaret (March 2011). “Kinect”. Retrieved 10/2012 from http://searchhealthit.techtarget.com/definition/Kinect

[8] Hui-mei Justina Hsu,"The Potential of Kinect as Interactive Educational Technology",2011 2nd International Conference on Education and Management Technology, IPEDR vol.13 (2011) © (2011) IACSIT Press, Singapore.

[9] “Kinect-powered Virtual Healthcare”. Retrieved 10/2012 from http://www.kinecthacks.com/kinect-powered-virtual-healthcare/

[10] Nelson, Randy (11/2010). “Kinect review: The hardware and experience”. Retrieved 11/2012 from http://www.joystiq.com/2010/11/04/kinect-review/

[11] Boldock, Alan (7/2011). ”Animating single bones in a Blender 3D model with XNA”. Retrieved 4/2013 from http://www.codeproject.com/Articles/230540/Animating-single-bones-in-a-Blender-3D-model-with.

[12] “Blender”, Retrieved 4/2013 from http://en.wikipedia.org/wiki/Blender_(software)

[13] “Microsoft XNA”, Retrieved 4/2013 from http://en.wikipedia.org/wiki/Microsoft_XNA

[14] Fuchsloch, Andrea, Weidner, Katja, “Blender to XNA Tutorial – Quick and Easy”