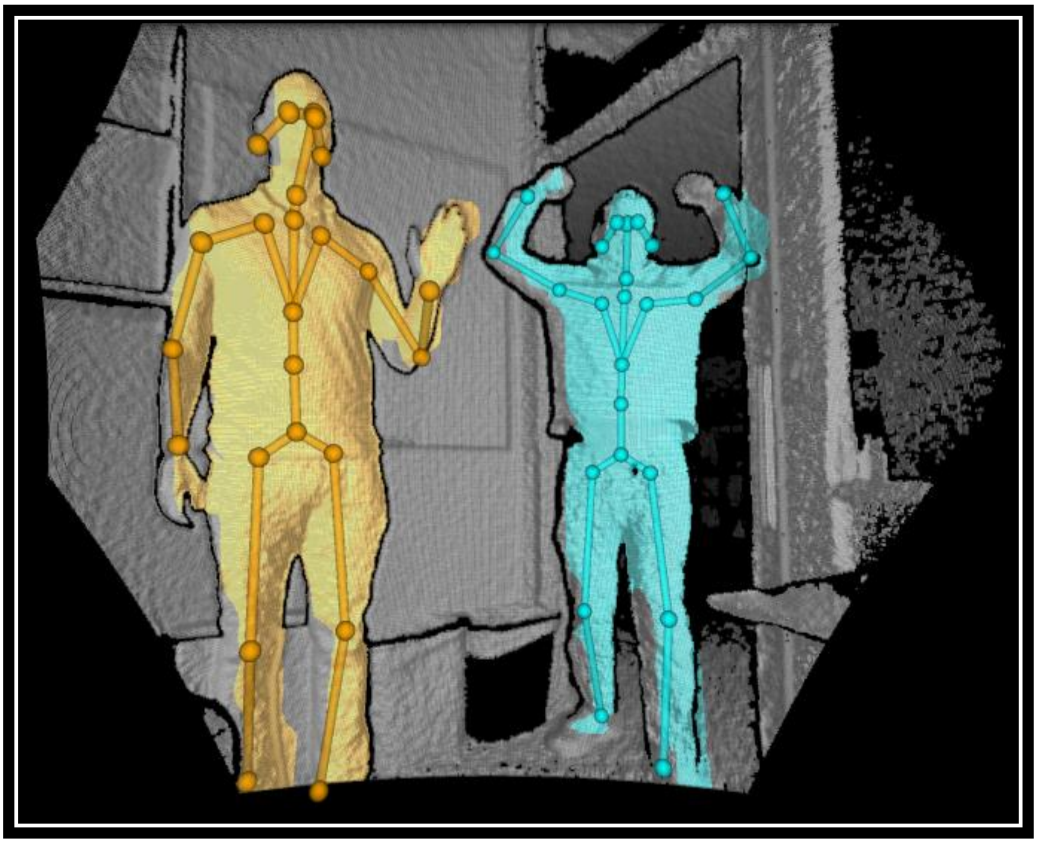
H!T
מכון טכנולוגי חולון
Holon Institute of Technology

MADE Multidisciplinary Assistive Design & Engineering Lab

A Practical Project for B.Sc. in Applied Mathematics

Marker-Less Body Tracking System for Cervical Dystonia



Author: Shay Shtern ID: 205891294

Lecturer: Dr. Jonathan Rubin

Submitted Date: 31.10.2021

Table of Contents:

Project summary:

This project is about creating base system that will help tracking human body movement, for creating future systems that will help people with disabilities.

Specifically, this project is based on the idea to help people that suffering from Cervical Dystonia – a disorder in which your neck muscles contract involuntarily, causing your head to twist or turn to one side. It can also cause your head to uncontrollably tilt forward or backward. This disorder has no cure. The project comes to help people by using it to create a system that will tell how humans that suffering from this disease are moving while the disorder.

Using body tracking is very useful for understating body behaviors. From that we can learn about the human body language, for example detecting specific movements.

In the past, doing body tracking was very expensive to make. It was needed to use 4 expensive cameras in a laboratory room, which will scan the room, and IR markers that were on specific location on the body that was tracked (for example- right hand, left hand, neck, left foot, etc.). Generally, the cameras looked for the points and needed to combine the point between the cameras.

Today, it's possible to do body tracking with only one camera, that is way less expensive, and to get great results from that.

In this project, for hardware part I used *Microsoft Azure Kinect Camera*, that is the newest model of the Kinect camera and PC, with a powerful GPU.
For the software part I used the *Body Tracking SDK* of the camera, and a software called OpenSim, that its goal is to visualize the skeleton of the body and be able to move parts of the body.

All the work was done in *'MADE'* laboratory that is at Holon Institute of technology (HIT), and planned to be develop there.

Introduction:

Your brain is your body's control center. It's part of the nervous system, which also includes the spinal cord and a large network of nerves and neurons. Together, the nervous system controls everything from your senses to the muscles throughout your body.

When your brain is damaged, it can affect many different things, including your memory, your sensation, and even your personality. Brain disorders include any conditions or disabilities that affect your brain. This includes conditions that are caused by illness, genetics, traumatic injury etc. This is a broad category of disorders, which vary greatly in symptoms and severity.

Dystonia is a kind of brain disorder. Dystonia is a movement disorder in which your muscles contract involuntarily, causing repetitive or twisting movements.

The condition can affect one part of your body (focal dystonia), two or more adjacent parts (segmental dystonia) or all parts of your body (general dystonia). The muscle spasms can range from mild to severe. They may be painful, and they can interfere with your performance of day-to-day tasks.

There's no cure for dystonia. But medications can improve symptoms. Surgery is sometimes used to disable or regulate nerves or certain brain regions in people with severe dystonia.

Dystonia affects different people in varying ways. Muscle contractions might:

- Begin in a single area, such as your leg, neck, or arm. Focal dystonia that begins after age 21 usually starts in the neck, arm or face and tends to remain focal or segmental.
- Occur during a specific action, such as handwriting.
- Worsen with stress, fatigue, or anxiety.
- Become more noticeable over time.

What is Cervical Dystonia:

Cervical Dystonia often referred to as spasmodic torticollis, is a particular type of focal dystonia that affects the muscles of the neck. It presents in different forms as follows: torticollis, laterocollis, anterocollis and retrocollis. Spasmodic Torticollis is the term commonly used to describe all forms of cervical dystonia. As the muscles of the neck are overactive, the sufferer's posture may be affected causing pain and stiffness in other muscles not directly affected by the dystonia. Information in the USA suggests that as many as 30 in 100,000 people have cervical dystonia. Accurate statistics are difficult to report because of those sufferers who continue to go undiagnosed in the community. The figure, therefore, may well be higher. Cervical dystonia is reported as more common in females and affects people of all ages though it usually occurs in the 40-50 age-group.

There are many different degrees of disability. Where some people with cervical will have few symptoms and little pain, others may suffer a high degree of disability and chronic pain, requiring treatment. Similarly, some sufferers have much fewer symptoms in the morning, particularly early in the disease, and symptoms return as the day goes on allowing more function early in the day.

There is no known cure for cervical dystonia. However, treatments that can improve life for the sufferer do exist.

Patients with cervical dystonia suffer from involuntary contraction of the neck muscles, which causes repetitive involuntary movements, which can be accompanied by pain, or remain in an abnormal head-neck position. Today, injecting a small amount of botulinum toxin into the muscles involved in movement or the dystonic position is the best and most effective option for treatment. Patients with cervical dystonia undergo injection treatments every 3-4 months. To determine the personal injection plan for each patient, which requires accuracy in the movement \ involuntary posture component, and to monitor the effectiveness of treatment after injection, it is necessary to define the facilitator's metrics and identify symptoms and in the second stage develop a support system for injection planning.

At present, follow-up is performed only on clinic visits (once every few months) and includes a visual assessment by the specialist doctor while the patient is sitting in front of him.

The goal in this project is to achieve objective quantitative definition of the severity of the disease using cameras and computer vision algorithms. Also, in the future the goal is get into evaluation more frequently also between hospital treatments, when the patient is at home (using a computer camera and appropriate algorithms).

Project planning and execution

As already mentioned, there is no solution or cure for this disease so because of that we needed to think creatively technologically and advanced about a solution to a problem that can help and facilitate the patient care process as simply as possible.

The solution that the project came to offer is to set up a system based on a computer that includes a camera that will monitor the patient's movements and thus you will be able to identify his types of movements.

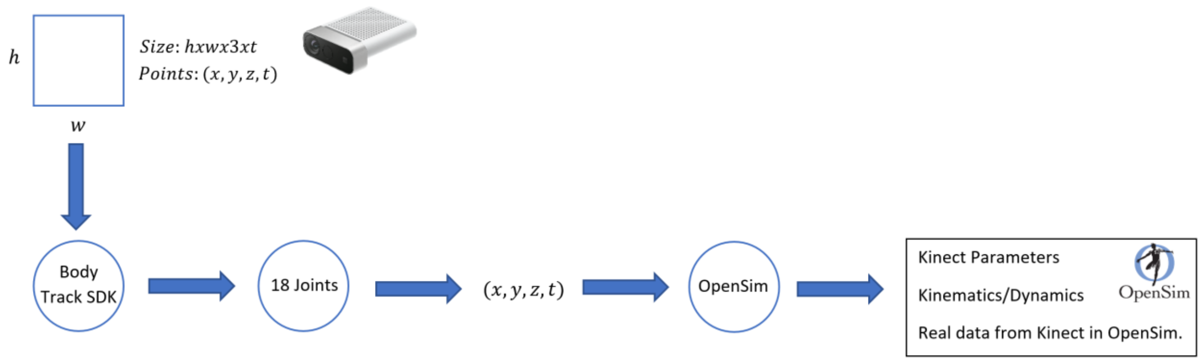
In the first decades the technology has evolved significantly, the most common technology of track human movements using computers was the marker-based techniques.

The marker-based techniques capture the motion information based on the tracking of markers, for example, reflective spheres, light-emitting diodes, or infrared markers. The markers are attached on the target regions, such as limbs, and their position and orientation are tracked. The recognition of marker-based object features in the video and their displacement information over time is used to record the continuous time-series data, which represents the dynamic limbs movement. For example, the research in proposed to attach different color markers on joints' locations and they are tracked in successive frames of the video. Similarly, two games are proposed in for upper limb stroke rehabilitation. The authors employed color objects that are attached to the upper limbs; the algorithm detects them using a calibration process and tracks them in the video to encode the motion information.

This project is coming with more advance technique that developing the latest years – tracking a human body only with camera, this idea called marker-less system. In the future everything needs to be simple, people want to use their smartphone camera for doing powerful things. There are differences between marker and marker-less systems – the main are:

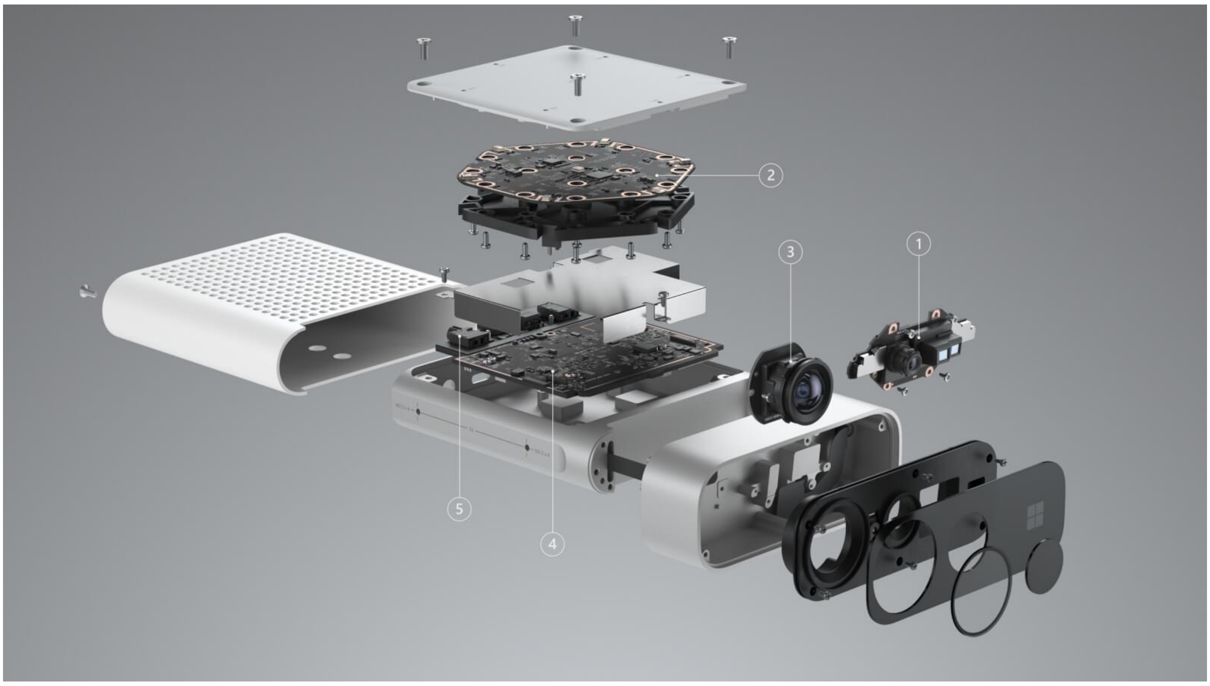
- Maker-less system cost less than marker systems.
- Tracking more easily without markers because you are not looking at specific points.
- Getting more information about the tracking object – camera looking the whole frame.

The project including Azure Kinect Camera, PC with Nvidia powerful GPU for running the algorithms and for the requirement of the camera. Also, this project will use a software called OpenSim© to get the data of the tracked body, view and modify it. This is the flow of this project:

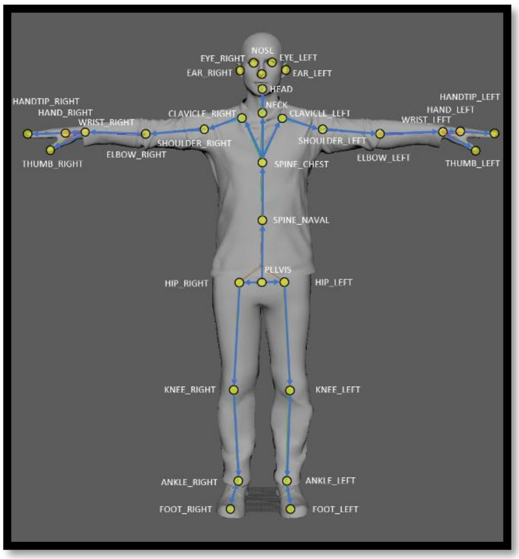


Microsoft Azure Kinect Camera:

This is a camera that was develop as a DK (Developer kit), released in March 2020. It has 1-MP depth sensor with wide and narrow field-of-view (FOV) options, 12-MP RGB video camera and in addition 7-microphone array along with Accelerometer and gyroscope (IMU). In this project we are using only the cameras sensor.



The Azure Kinect has its own SDK with body tracking option that allow the camera to detect a human body skeleton. This skeleton has 32 joints* that measured by its own algorithm from CNN by using the depth camera.



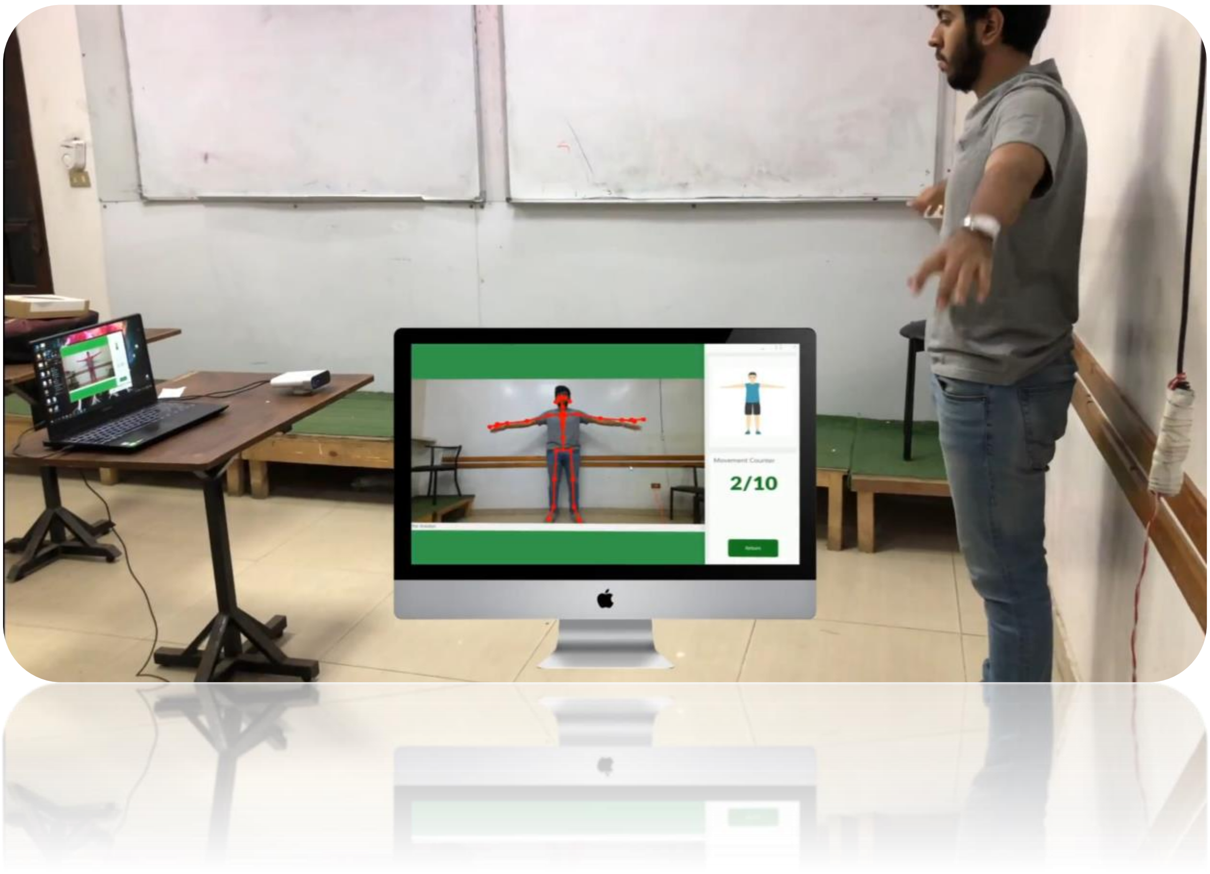
*A joint is the part of the body where two or more bones meet to allow movement.

Using Azure Kinect:

Cuda-enabled GPU is needed to use the Azure Kinect, because this camera is using CNN that require powerful hardware to run it.

Generally, this camera has its own SDK for controlling it, based on C++. For future using and to be able to use it more easily, I looked for a python-based SDK that will be easier to use. In this project I used an SDK called "*pyKinectAzure*" that is just the same functional SDK - in Python.

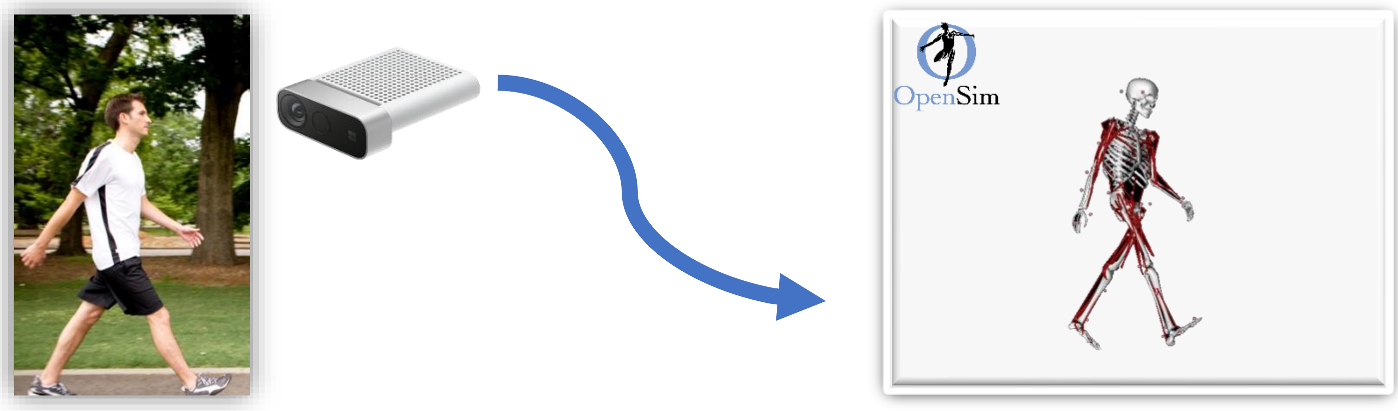
With this package you can control the cameras, show the image from the camera, get cloud points, run body tracking and control his joints.



OpenSim and Integration with Azure Kinect:

This is an open-source software system for biomedical modeling, emulation and analysis. Its purpose is to provide free and widely accessible tools for conducting biomechanics research and motor control science. It can do for example analysis of walking dynamics, studies of sports performance, simulations of surgical procedures, analysis of joint loads, design of medical devices, and animation of human and animal movement. The software performs inverse dynamics analysis and forward dynamics simulations. OpenSim is used to study movement in hundreds of biomechanics laboratories around the world and has a community of software developers contributing new features.

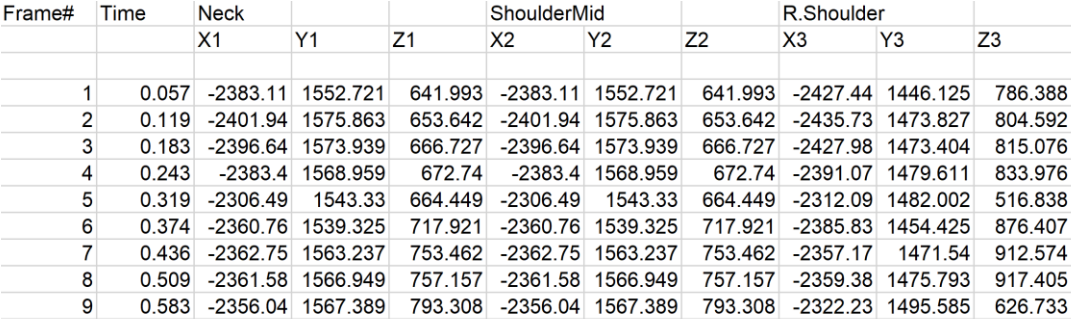
In this project I planned to find how to create the connection between Azure Kinect camera to OpenSim software, for getting the ability to see the skeleton moving in the software exactly like in the body tracking results from the Kinect.



Processing:

Getting the data from azure Kinect:

Every joint in the body is simply a point in the 3D space: (x, y, z). While writing a python script using the SDK of the camera, I got for every joint the value of the 3-demintional point in the space. In addition, it is necessary to save the point detection time. I decided to save the points in .xlsx file (an Excel file), for example:

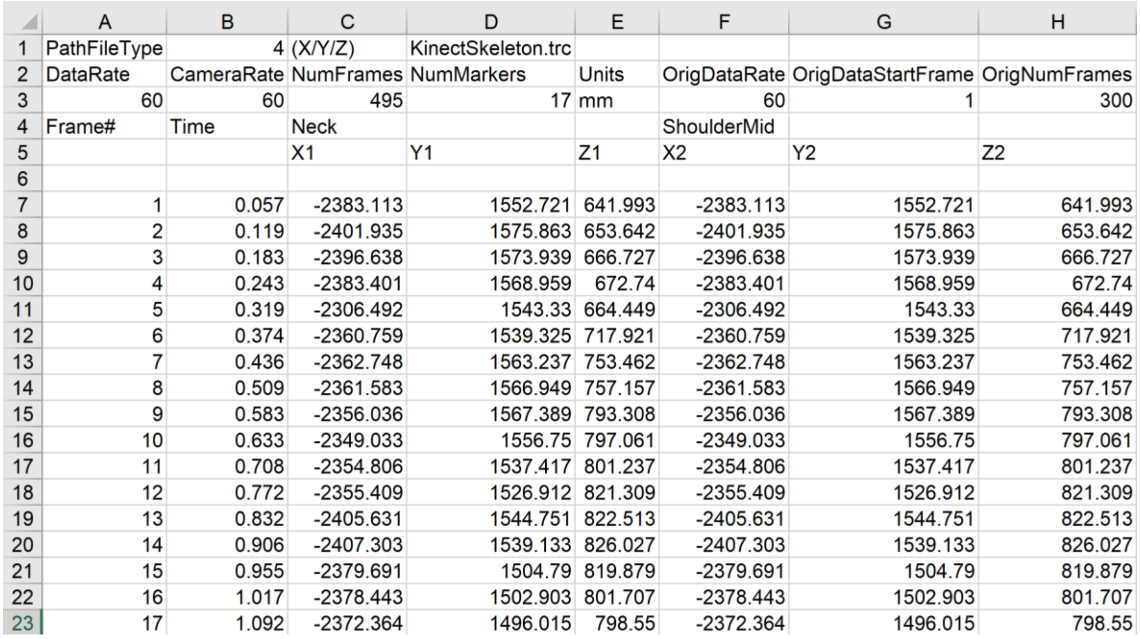


Frame#	Time	Neck			ShoulderMid			R.Shoulder		
		X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3
1	0.057	-2383.11	1552.721	641.993	-2383.11	1552.721	641.993	-2427.44	1446.125	786.388
2	0.119	-2401.94	1575.863	653.642	-2401.94	1575.863	653.642	-2435.73	1473.827	804.592
3	0.183	-2396.64	1573.939	666.727	-2396.64	1573.939	666.727	-2427.98	1473.404	815.076
4	0.243	-2383.4	1568.959	672.74	-2383.4	1568.959	672.74	-2391.07	1479.611	833.976
5	0.319	-2306.49	1543.33	664.449	-2306.49	1543.33	664.449	-2312.09	1482.002	516.838
6	0.374	-2360.76	1539.325	717.921	-2360.76	1539.325	717.921	-2385.83	1454.425	876.407
7	0.436	-2362.75	1563.237	753.462	-2362.75	1563.237	753.462	-2357.17	1471.54	912.574
8	0.509	-2361.58	1566.949	757.157	-2361.58	1566.949	757.157	-2359.38	1475.793	917.405
9	0.583	-2356.04	1567.389	793.308	-2356.04	1567.389	793.308	-2322.23	1495.585	626.733

…

Preparing the data for OpenSim:

After a lot of research about how this software getting data of a motion, I found there is a format that can be read in OpenSim: .trc – track row column, a file with the position of the point and the time it was taken. It has a specific form, here is an example:



	A	B	C	D	E	F	G	H
1	PathFileType	4	(X/Y/Z)	KinectSkeleton.trc				
2	DataRate	CameraRate	NumFrames	NumMarkers	Units	OrigDataRate	OrigDataStartFrame	OrigNumFrames
3	60	60	495	17	mm	60	1	300
4	Frame#	Time	Neck			ShoulderMid		
5			X1	Y1	Z1	X2	Y2	Z2
6								
7	1	0.057	-2383.113	1552.721	641.993	-2383.113	1552.721	641.993
8	2	0.119	-2401.935	1575.863	653.642	-2401.935	1575.863	653.642
9	3	0.183	-2396.638	1573.939	666.727	-2396.638	1573.939	666.727
10	4	0.243	-2383.401	1568.959	672.74	-2383.401	1568.959	672.74
11	5	0.319	-2306.492	1543.33	664.449	-2306.492	1543.33	664.449
12	6	0.374	-2360.759	1539.325	717.921	-2360.759	1539.325	717.921
13	7	0.436	-2362.748	1563.237	753.462	-2362.748	1563.237	753.462
14	8	0.509	-2361.583	1566.949	757.157	-2361.583	1566.949	757.157
15	9	0.583	-2356.036	1567.389	793.308	-2356.036	1567.389	793.308
16	10	0.633	-2349.033	1556.75	797.061	-2349.033	1556.75	797.061
17	11	0.708	-2354.806	1537.417	801.237	-2354.806	1537.417	801.237
18	12	0.772	-2355.409	1526.912	821.309	-2355.409	1526.912	821.309
19	13	0.832	-2405.631	1544.751	822.513	-2405.631	1544.751	822.513
20	14	0.906	-2407.303	1539.133	826.027	-2407.303	1539.133	826.027
21	15	0.955	-2379.691	1504.79	819.879	-2379.691	1504.79	819.879
22	16	1.017	-2378.443	1502.903	801.707	-2378.443	1502.903	801.707
23	17	1.092	-2372.364	1496.015	798.55	-2372.364	1496.015	798.55

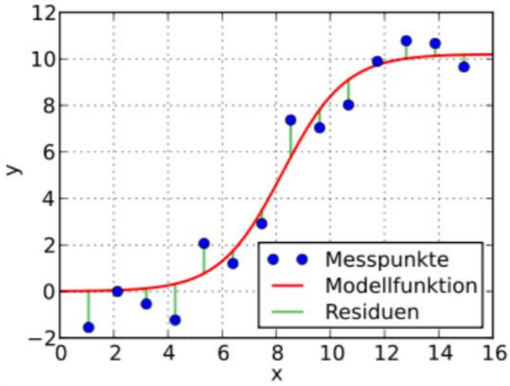
Inverse Kinematic Tool:

The Inverse Kinematics Tool of OpenSim is a possibility that allow to import .trc file. The IK Tool processes each recorded frame (image) individually and assigns the recorded coordinates to the markers on the selected model in the best possible way. This assignment is based on the mathematical method of least squares.

Least squares method:

The least squares method is the standard mathematical method for finding a curve in a data point cloud, that is as close as possible to the data points.

Here is an example of a curve that created from data set of points, represent the minimal distance between the points. To calculate the values for this curve, OpenSim uses a formula for the least square method that is tailored to the requirements.



$$\min_{\boldsymbol{q}}\left[\sum_{m=1}^{\#\,\text{markers}} w_m \left\|\boldsymbol{x}_m^{\text{exp}} - \boldsymbol{x}_m(\boldsymbol{q})\right\|^2 + \sum_{c=1}^{\#\,\text{coordinates}} \omega_c \left(q_c^{\text{exp}} - q_c\right)^2\right]$$

This formula is made up of two parts. w_m and ω_c stand for the value of the selected marker point or the selected angle coordinate. Depending on the importance, this can be increased, which allows a greater influence on the overall result. The marker points are considered on the left. x_m^{exp} stands for the experimental values of the project application, and $x_m(q)$ for the marker points on the model. On the right, there is the part of the formula that is for experimental angular coordinates responsible. q_c^{exp}, stands for the experimental angle data, and q_c for the existing Angular data on the model.

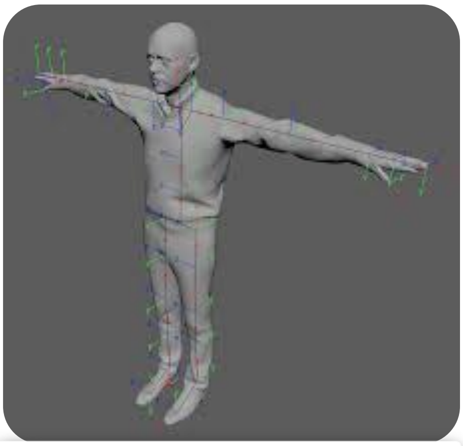
How Least Square Method works in OpenSim

Simply, The IK tool goes through each frame of motion and computes generalized coordinate values which positions the model in a pose that best matches experimental marker and coordinate values for that time step. This process done with the mathematical technique of weighted least squares problem, whose solution aims to minimize both marker and coordinate errors.

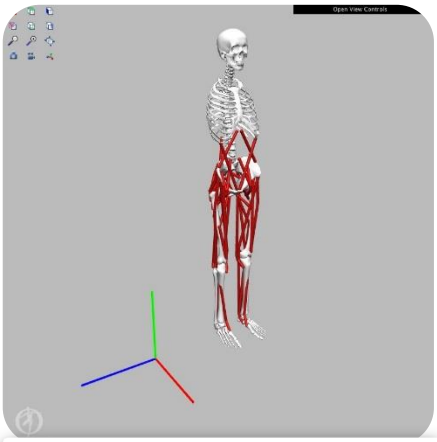
A marker error is the distance between an experimental marker and the corresponding marker on the model when it is positioned using the generalized coordinates computed by the IK solver. Each marker has a weight associated with it, specifying how strongly that marker's error term should be minimized.

From Azure Kinect Space to OpenSim Space:

First, I tried to load the .trc file to OpenSim as it is. Then the model I used in OpenSim with the pure data was not at the right direction, so I figured that the points needed to be transform.



Azure Kinect Coordinate System

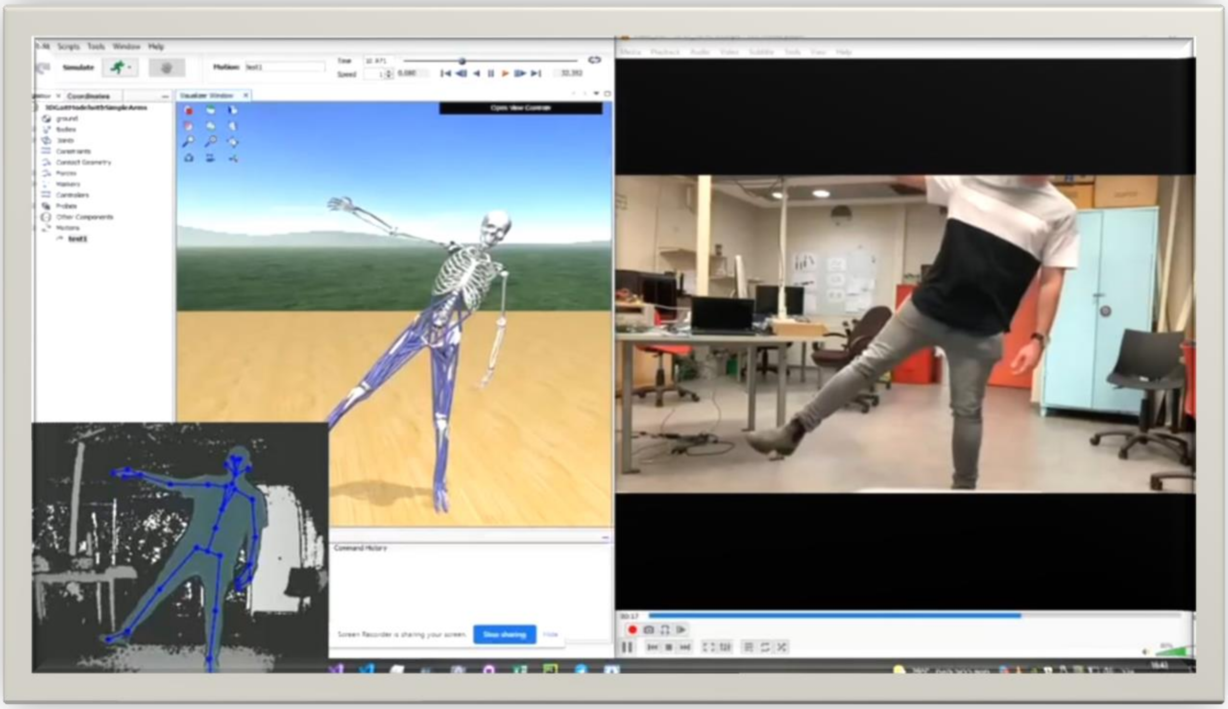


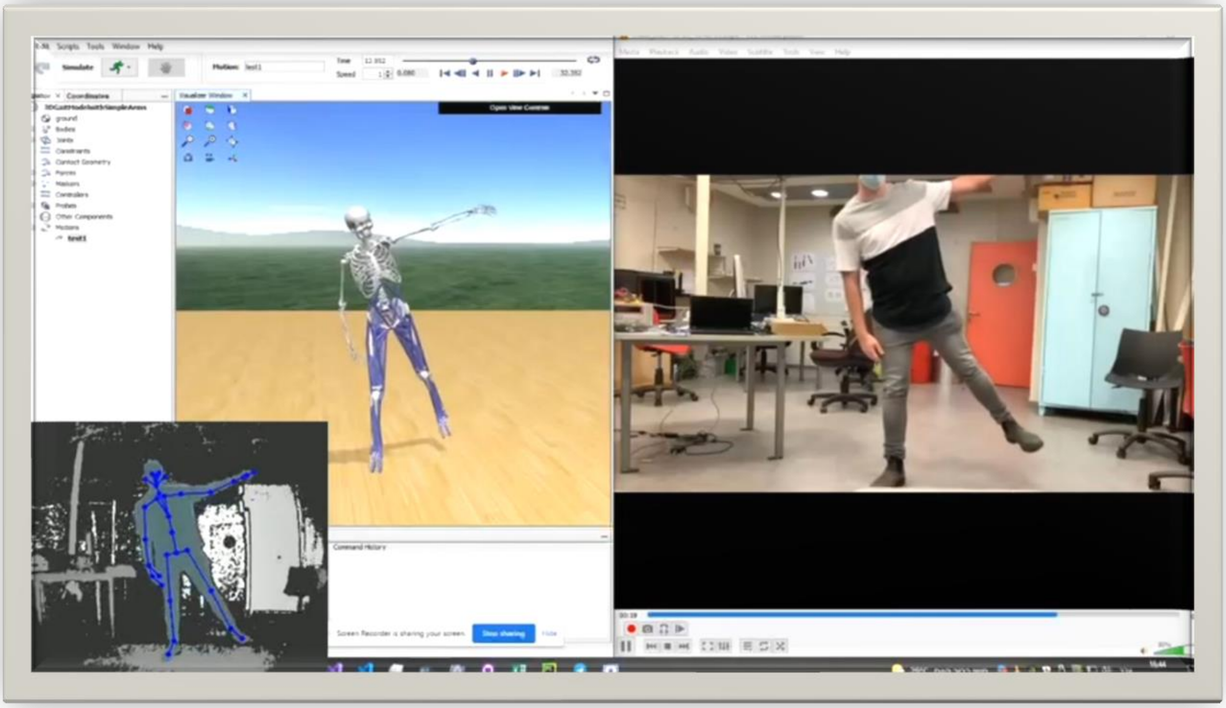
OpenSim Coordinate System

The transformation is rotate every point 180° in the X-axis and 90° in the Y-axis. Also add 700 to every point is required to place the model in OpenSim to stand on the floor.

Results:

Here is a frame from a video that shows the Azure Kinect image and the OpenSim motion moving at the same time. The skeleton in OpenSim moving according to data that was collected from the camera.





Summary and Conclusions:

The vision of this project is to develop a system as a simple application for those who suffer from the disorder. This app will be easy to install on every smartphone, while the patient will be able to easily take a video of himself according to the instruction in the app. After that, the application will analyze his/her movements the send it to his doctor.

The program will be based on Machine Learning and Deep Learning methods and algorithms. The plan is to build a system, that will get a lot of examples of people moving their head and neck in many situations and directions. From that, the machine will learn to be able to predict the future of the movements and at the end analyze the specific movement. This will help the doctors to know more about the client without the need to see their patients often, and the clients won't need to get to the hospital many times for a short procedure.

Finally, I found this project very helpful for people who suffer from the Cervical Dystonia. I learned How to work with the Azure camera and how to use it's SDK. It was a great experience for me to do this research and to take part in the process of the big project. The research was challenging and very interesting. I'm satisfied from the process, and glad I had the right to help people.

Bibliography:

[1]https://azure.microsoft.com/en-us/services/kinect-dk/

[2]https://github.com/microsoft/Azure-Kinect-Sensor-SDK

[3]https://github.com/ibaiGorordo/pyKinectAzure

[4]https://opensim.stanford.edu/