

RGB-D Cameras

Laboratorio di Realtà Virtuale

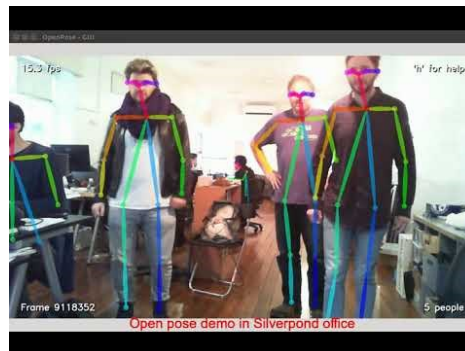


Manuel Pezzera – manuel.pezzera@unimi.it



Overview

- 3D Cameras: Introduction and theory
 - Microsoft Kinect, Orbbec Astra, Intel RealSense
- 2D Cameras and human pose estimation
 - OpenPose, PoseNet, wrnch.ai
- RGB-D cameras & Unity
 - OpenPose & PoseNet packages
 - Nitrack SDK
 - Orbbec - Unity Integration
 - Avateering in Unity



Overview

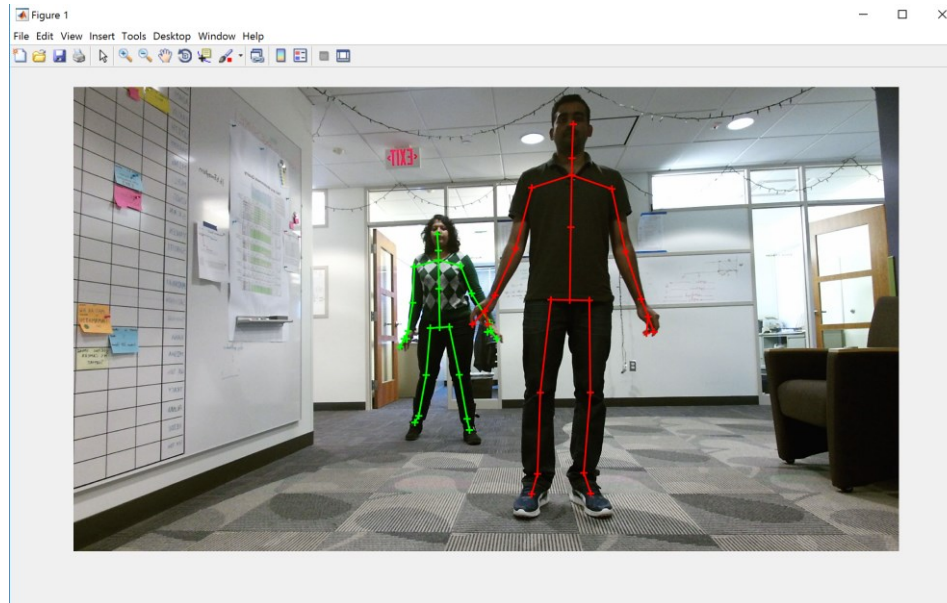
- RGB-D cameras, also called “Depth Cameras” or 3D cameras, are able to obtain a “3D photo” of the environment.
- It means that they are able to detect the distance between the camera itself and a subject.
- See the following images:



- The left image is recorded using a classical 2D cameras and for the computer is quite difficult understand which is the silhouette of the person. While, on the other hand, the picture on the right is captured using Kinect, and the background has a completely different color.

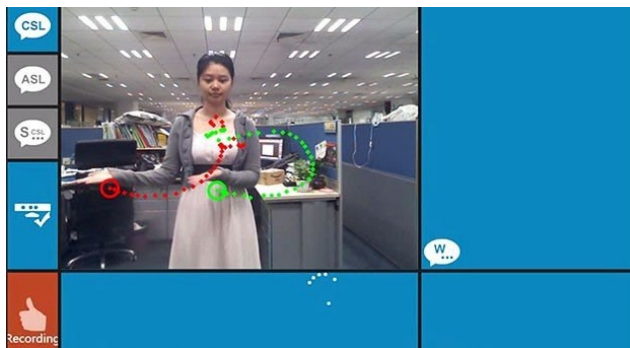
3D Cameras: overview

- Using the depth image is quite easy to detect human posture
 - Kinect 1, for example, is able to recognize 20 different body joints.
 - Kinect 2 has been improved and can recognize 26 joints.
- Born as a gaming device/controller, it has also been widely used in the research field.
- Lot of projects used it to record people's movement trying to diagnose diseases or helping with home-rehabilitation.

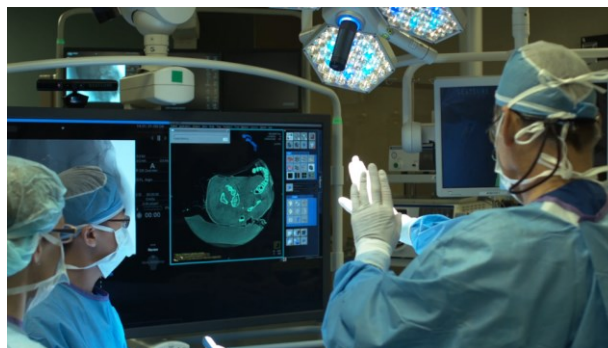


RGB-D cameras application

- Few innovative uses of RGB-D cameras
 - Produce high-quality 3D scans
 - E.g., Kinect Fusion
 - Help with stroke (or other diseases) recovery
 - Translate sign language
 - Retrieve data via gestures
 - Control robots with body movement



Translate sign language

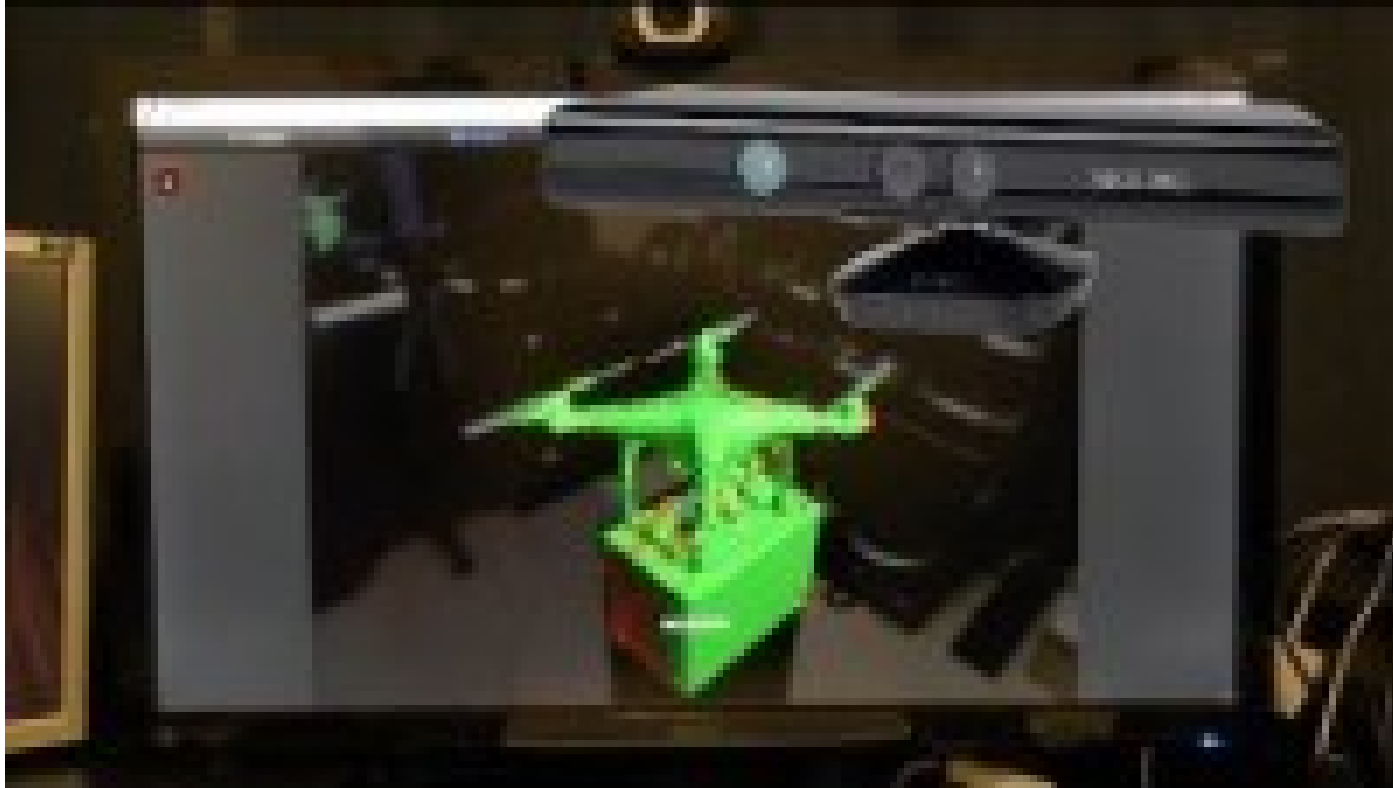


Retrieve data via gestures



Kinect Fusion

RGB-D cameras applications: 3D scanning



https://www.youtube.com/watch?v=mPtOr3lXiaA&ab_channel=baaelectronics

RGB-D cameras applications: Rehabilitation



https://www.youtube.com/watch?v=pfcWbt_7V-Y

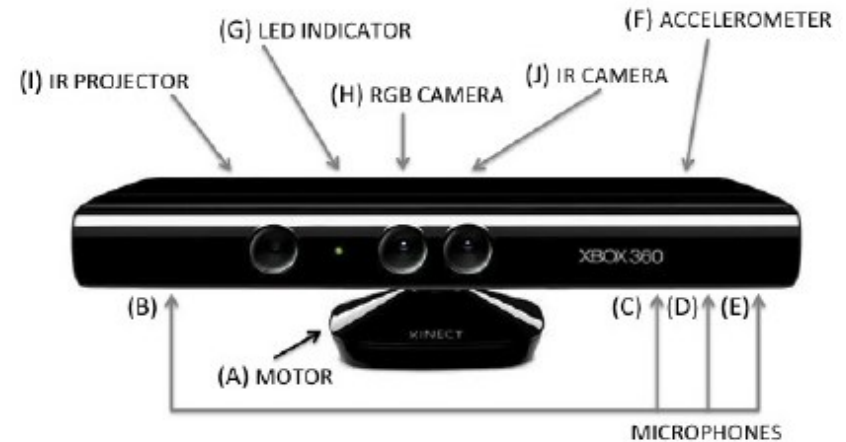
3D Cameras: overview

- Most common 3D cameras are:
 - Kinect 1
 - Kinect 2
 - Orbbec Astra
 - Intel RealSense
 - Azure Kinect DK



Kinect 1

- Launched in 2010 (for Xbox 360)
- Windows version came in 2012
 - Near mode added
- Kinect 1 has four main components:
 - An infrared projector
 - A RGB camera
 - 640x480 @ 30fps, 1280x960 @ 12fps
 - An infrared camera
 - 640x480 @ 30fps
 - Microphones array
 - 4 microphones



Kinect 1: how does it work?

1 The projector shoots an irregular pattern of dots invisible to humans.

2 CMOS sensors in the IR camera can detect the infrared light bounced off our subjects

3 Depth is calculated for every pixel in the scene: Kinect compares the data captured by the IR camera with the irregular pattern that has been projected. Triangulation methods are used to calculate the final distance.



Useful videos:

- <https://www.youtube.com/watch?v=uq9SEJxZiUg>
- <https://www.youtube.com/watch?v=dTKINGSH9Po>

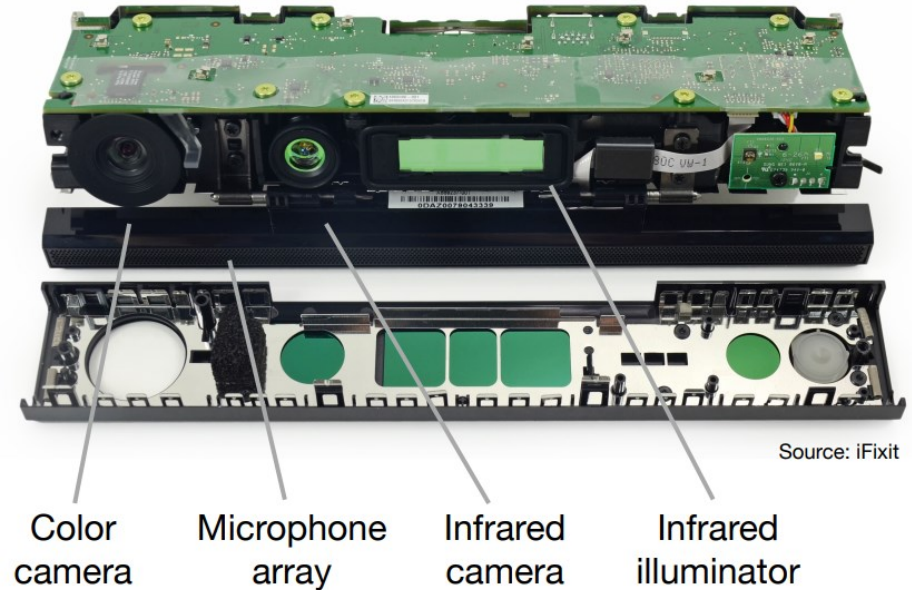
Kinect 1: how does it work?



- <https://www.youtube.com/watch?v=dTKINGSH9Po>

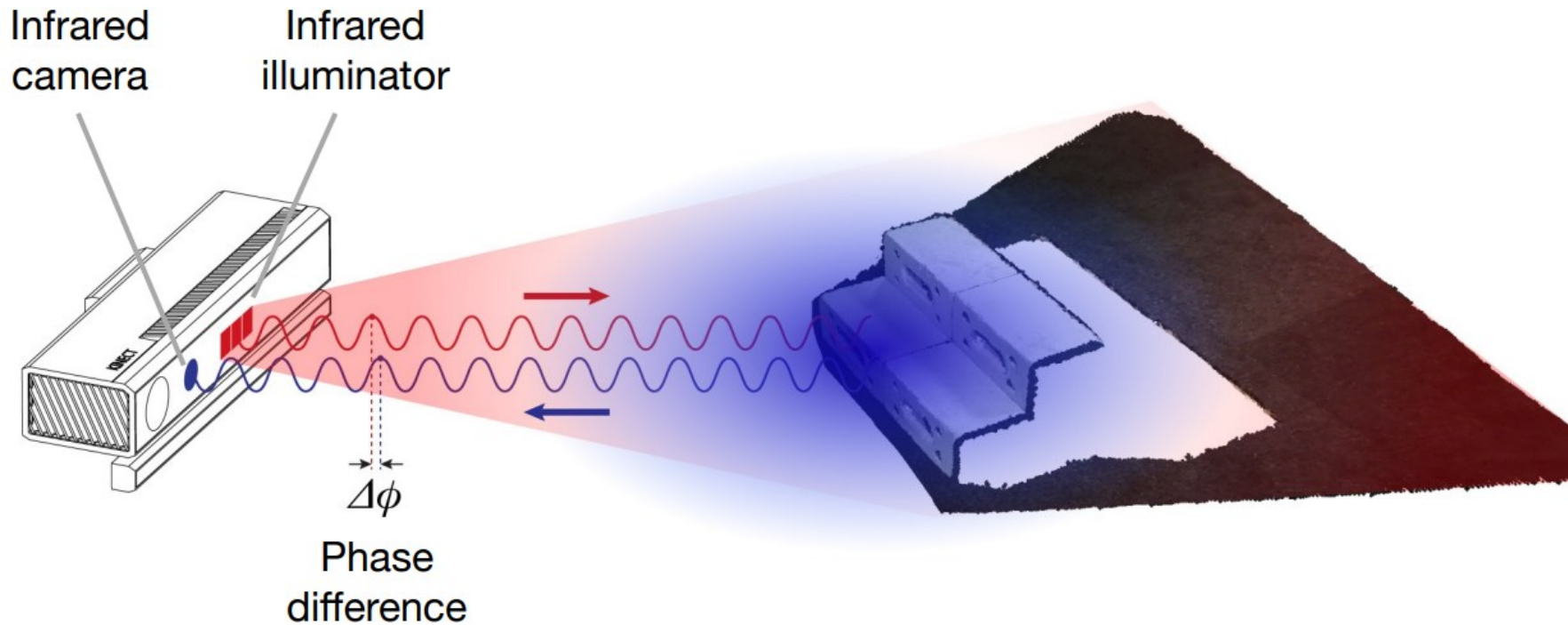
Kinect 2

- Launched in 2013 for Xbox One and PCs
- Main components:
 - A RGB camera
 - 1920x1080 @ 30fps
 - An infrared camera
 - 512x424 @ 30fps
 - Microphones array
 - 4 microphones



- Kinect 2 has the “new” Time of Flight (ToF) technology to calculate the depth image.
 - No dots are used here.

Time of Flight technology



- Useful link:
 - https://blogs.technet.microsoft.com/microsoft_blog/2013/10/02/collaboration-expertise-produce-enhanced-sensing-in-xbox-one/

Kinect: human recognition

- We have understood how Kinect 1 and Kinect 2 obtain the depth image.
- But how they use this image to recognition human posture?
- Random decision forest classifier are usually used
 - 1) Find body parts
 - 2) Compute joints positions



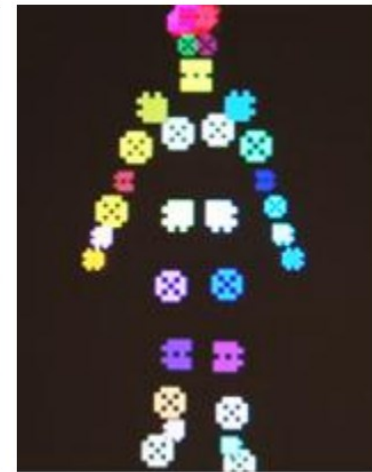
RGB



Depth



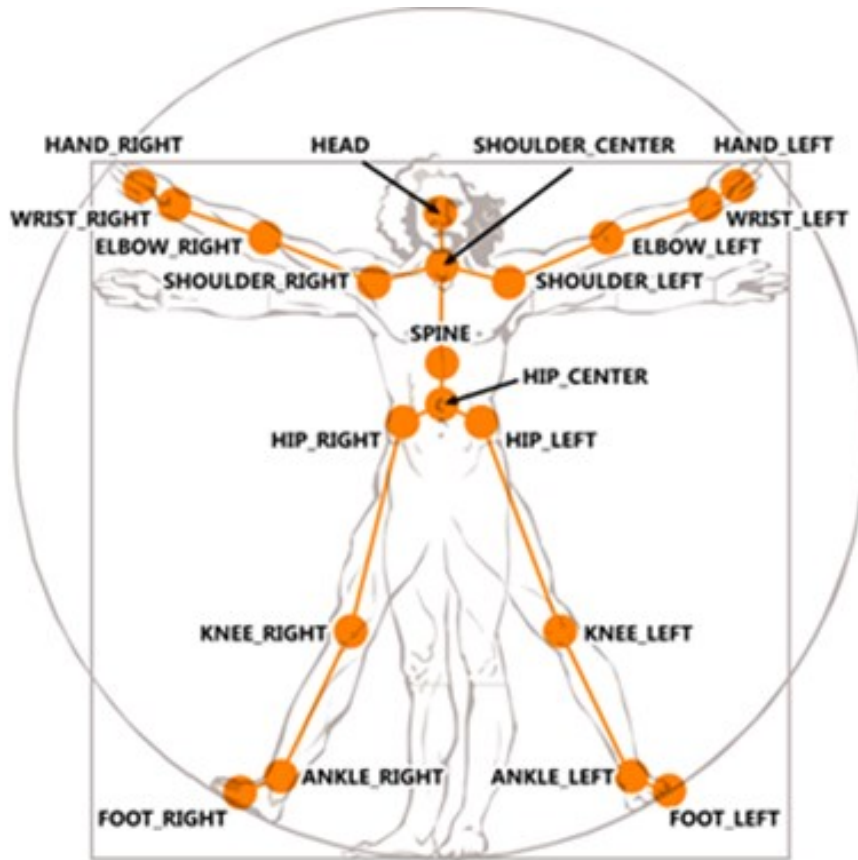
Part Label Map



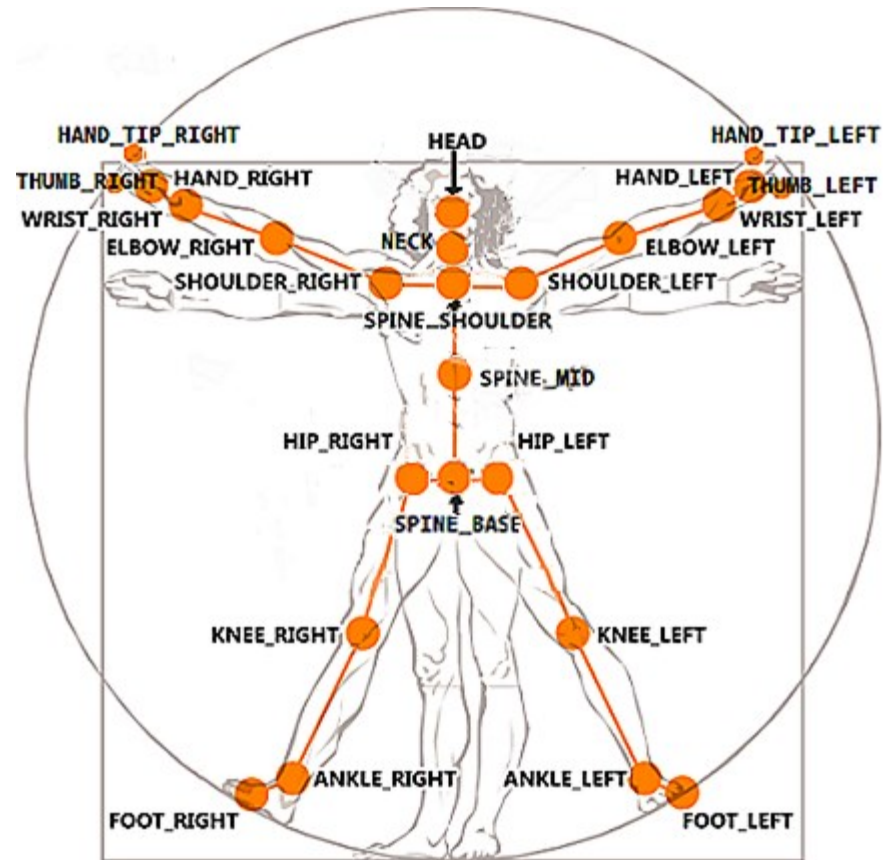
Joint Positions

Detailed information: [Real-Time Human Pose Recognition in Parts from a Single Depth Image](#)

Kinect 1 and Kinect 2 skeletons

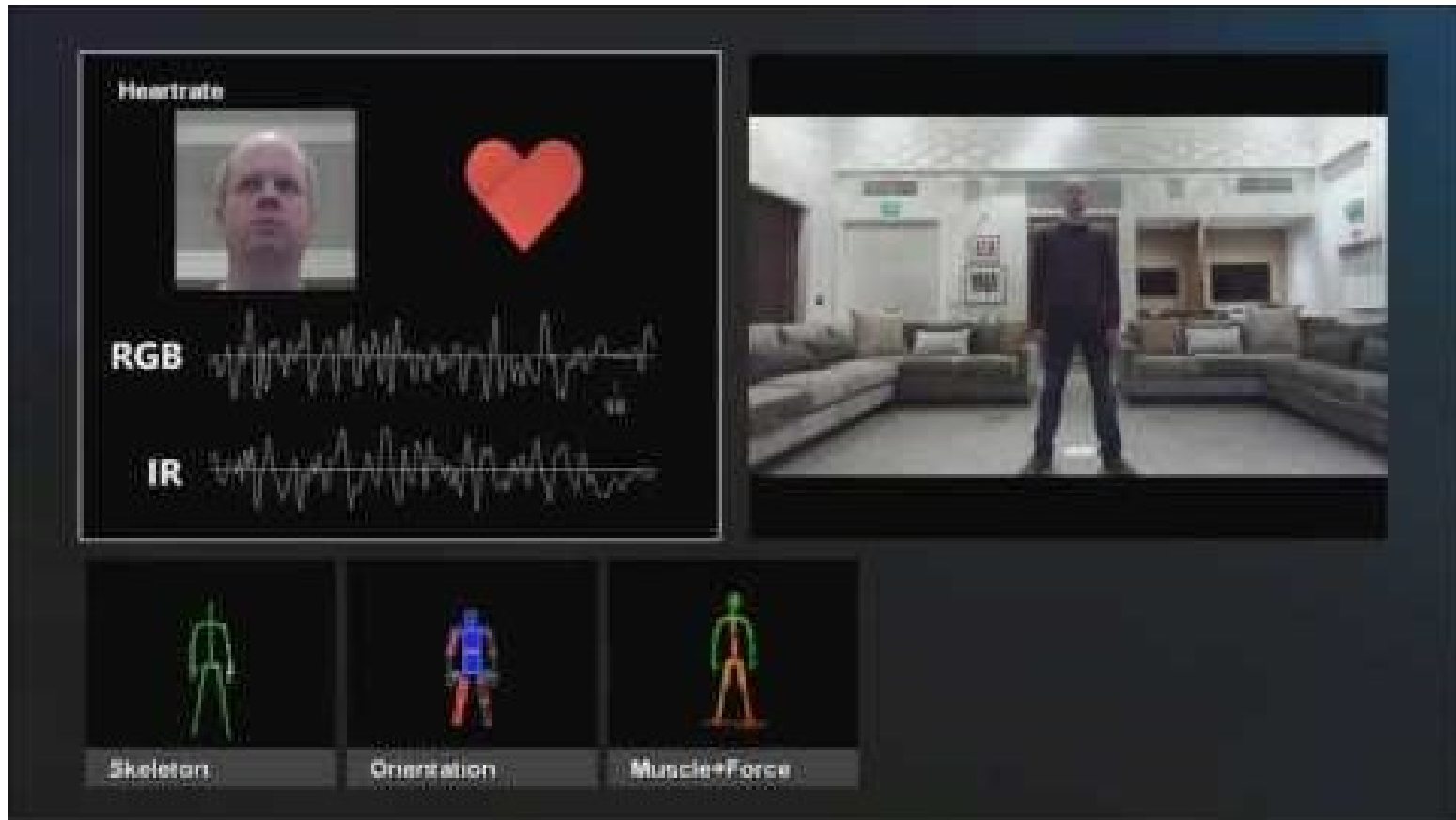


Kinect 1



Kinect 2

Kinect 2 Demo



<https://www.youtube.com/watch?v=OWzjn656kb4>

Kinect 2 Demo



https://www.youtube.com/watch?v=8_yMPLSZSEs

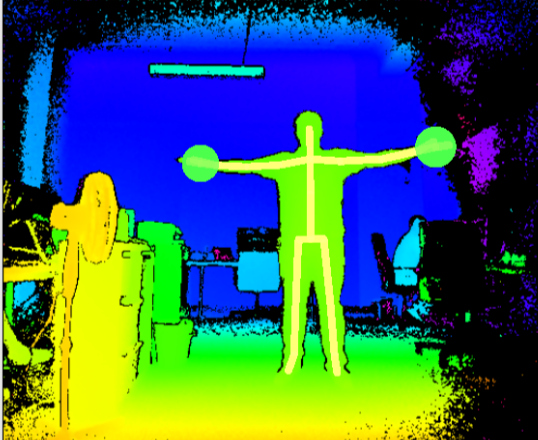
Kinect Studio

HandsAboveHead.xef [Read-Only] - Microsoft® Kinect Studio

FILE MONITOR RECORD PLAY LOG

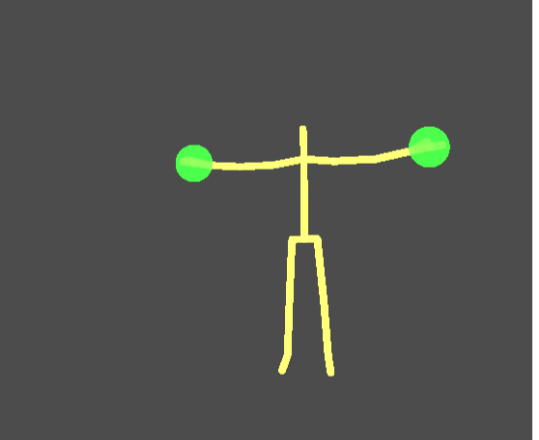
Monitor 2D View Monitor 3D View 2D Properties Metadata

MONITOR: FILE [Settings] [Zoom to fit] [Zoom In] [Zoom Out]



Playback 2D View Playback 3D View

INSPECT: FILE [Settings] [Zoom to fit] [Zoom In] [Zoom Out]

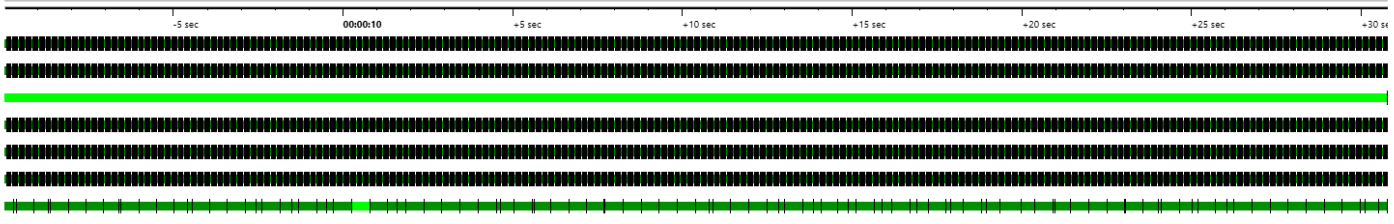


Connected [Play] [Stop] [Pause] Loop: Count 0 Iteration 0 [Stop] [Reset] [Close] [Menu]

HandsAboveHead.xef [Read-Only] Current Time: 00:00:10.7138336

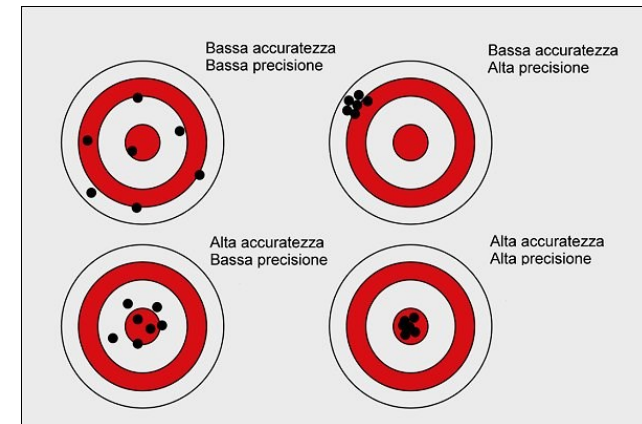
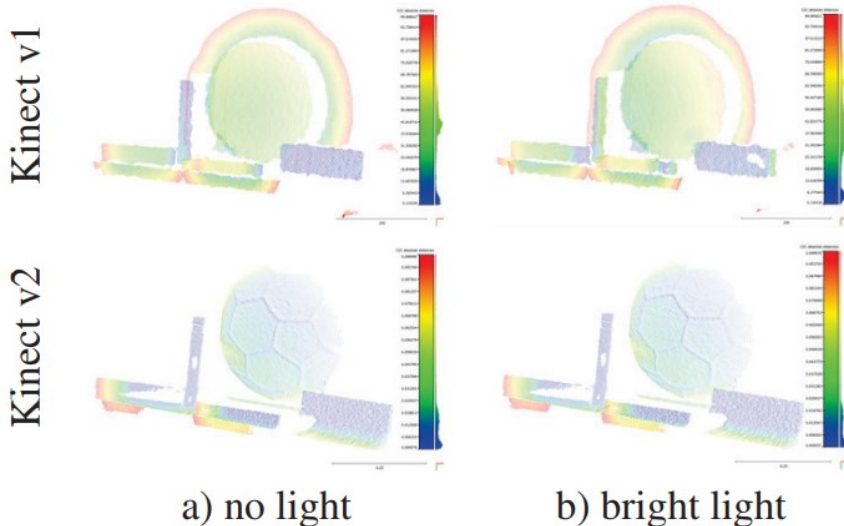
0 Streams (0 visible, 0 hidden)

- Nui Body Frame
- Nui Body Index
- Nui Calibration Data
- Nui Depth
- Nui IR
- Nui Opaque Data
- Nui Sensor Telemetry



Kinect 1 and Kinect 2: depth image comparison

- Kinect 1 is able to record reliable images just after starting, Kinect 2 needs to be pre-heat for at least 25 minutes in order to achieve reliable results.
- Accuracy decreases exponentially if we increase distance for Kinect 1, Kinect 2 does not have this problem.
- Artificial light can affect Kinect v1. Kinect 1 is total blindness in present of sunlight while Kinect 2 works a little bit better, but both are designed to be used indoor.
- Kinect 2 has better accuracy (even 50%), Kinect 1 has better precision.



- Wasenmüller, Oliver, and Didier Stricker. "Comparison of kinect v1 and v2 depth images in terms of accuracy and precision."
- Zennaro, S., et al. "Performance evaluation of the 1st and 2nd generation Kinect for multimedia applications."

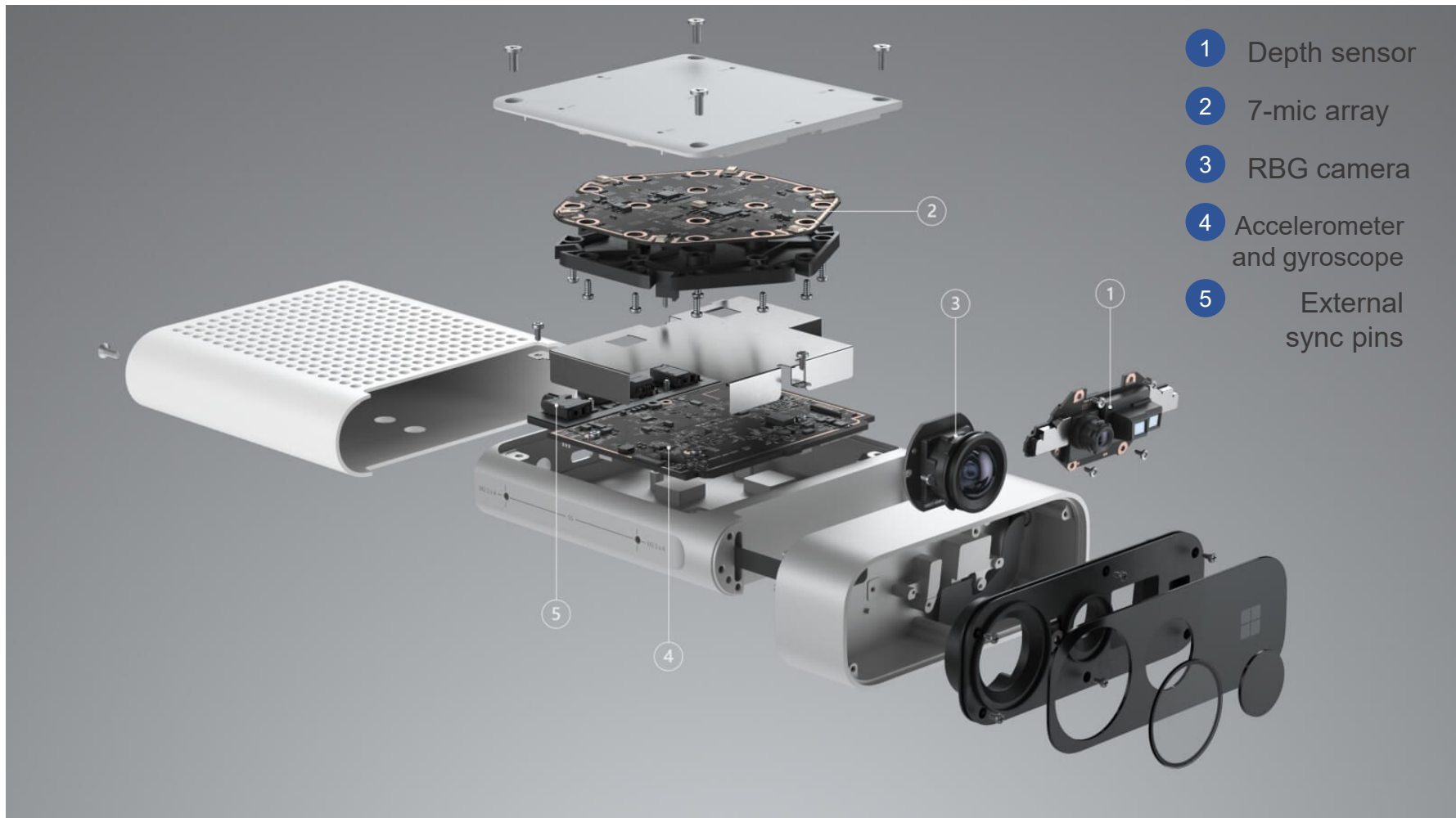
Azure Kinect

- Preorder started February 2019
- 12 MP RGB Camera
 - Up to 3840x2160 @ 30 FPS
- 1 MP Depth camera
 - 640x576 @ 30 FPS
 - 512x512 @ 30 FPS
 - 1024x1024 @ 15 FPS
 - Time-Of-Flight camera
- Motion sensor
 - Accelerometer and gyroscope
- Microphone array
 - 7 microphone circular array
- Vision API & Speech Service Sdk (from Azure)
- Body tracking SDK released March 2021



<https://www.mdpi.com/1424-8220/21/2/413/pdf#:~:text=The%20device%2C%20however%2C%20has%20to,convenient%20mostly%20for%20indoor%20applications.>

Azure Kinect



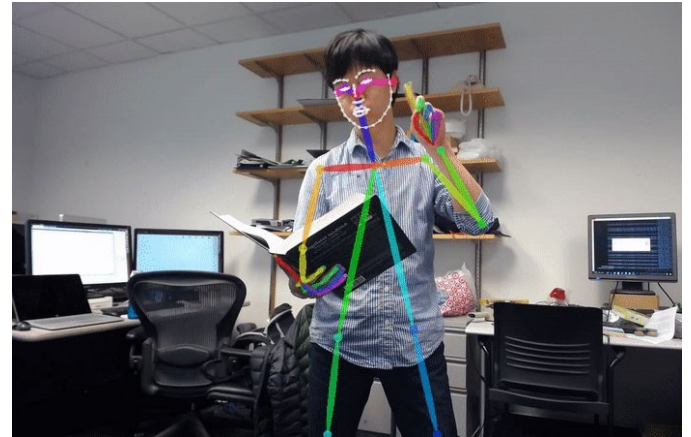
Azure Kinect



<https://www.youtube.com/watch?v=2URblck97y8>

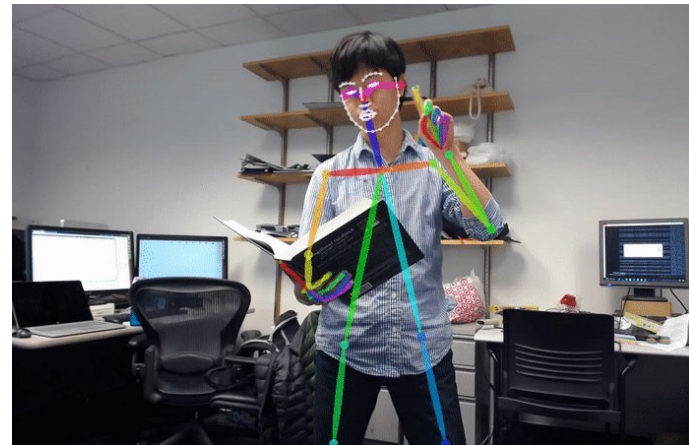
Human Pose Estimation with 2D Cameras

- In the last years, a few algorithms to estimate human pose from 2D cameras have been developed
 - E.g.: OpenPose, PoseNet, wrnch.ai
- These algorithms can extract joints positions from a classic 2D image.
- Both offline and real-time!



OpenPose

- Source code available:
 - <https://github.com/CMU-Perceptual-Computing-Lab/openpose>
- Real time
- Multi-person
- Able to detect human body, hand, facial and foot, for a total of 135 keypoints
 - Face has 70 keypoints!
- Unity plugin available:
 - https://github.com/CMU-Perceptual-Computing-Lab/openpose_unity_plugin



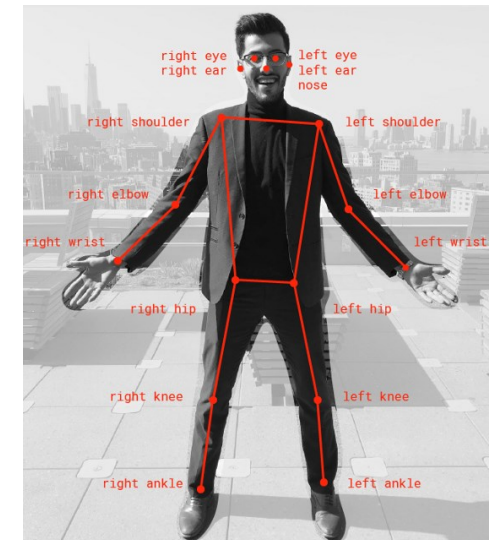
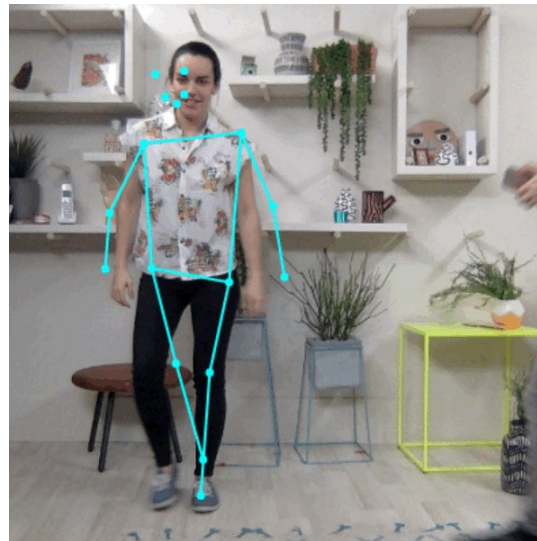
OpenPose

- Unbelievable!
- But...cons?
 - 9.75 FPS on a GTX 1080 Ti (body only)
 - All keypoints (body + hands + face): 3 FPS (4 if you are lucky)
 - With a two GTX 1080 Ti we can reach 18 FPS

Graphics card model	#GPUs	Millisec / frame body-only+	FPS body-only+	GPU memory body (MiB)+	Fps all (body,hands,face)+
GTX 1080 Ti (11GB)**	1		~9.75		~1/2 or 1/3 of body-only
Titan X Pascal (12GB)	1	115.7	8.6	1489	~1/2 or 1/3 of body-only
GTX 1080 (8GB)	1	119.4	8.4	1443	~1/2 or 1/3 of body-only
GTX 1070 (8GB)**	1		~8.75		~1/2 or 1/3 of body-only
Quadro M6000 (24GB)**	1		~6.6	~1500	~1/2 or 1/3 of body-only
GTX 980 (8GB)**	1		~6.5	~1500	~1/2 or 1/3 of body-only

PoseNet

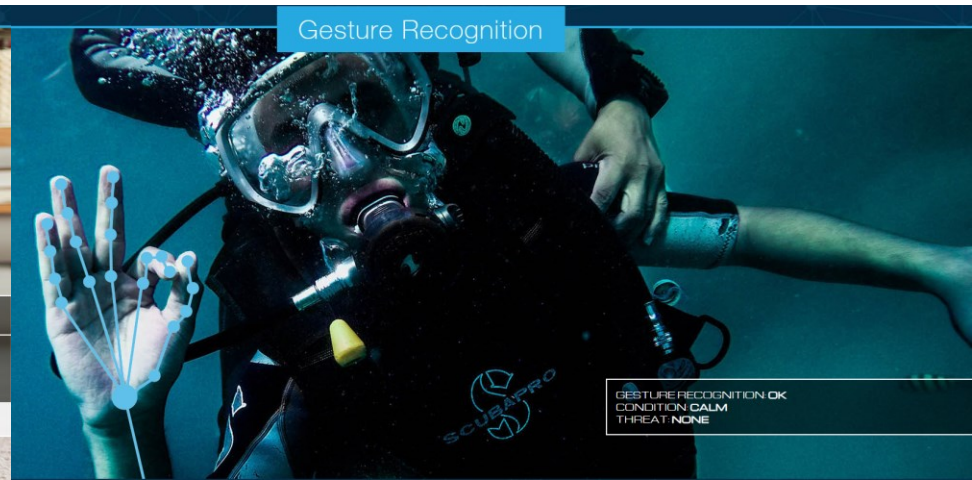
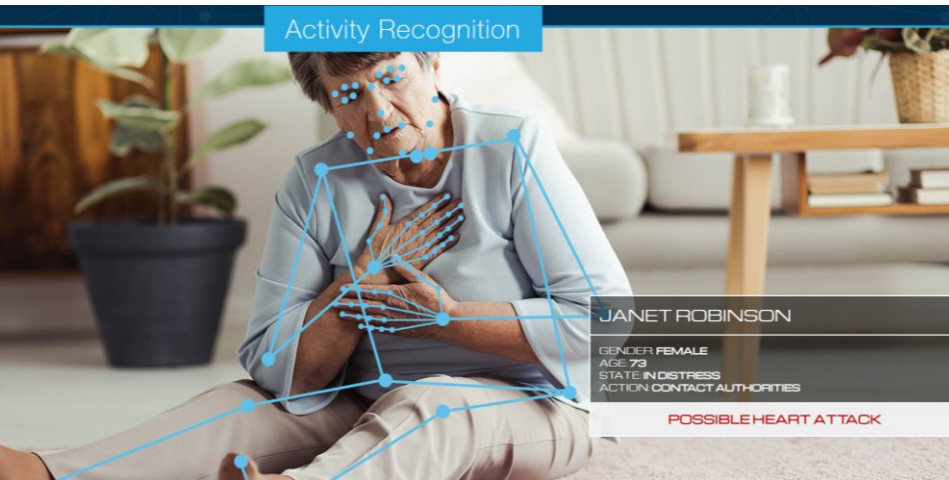
- Real-time multi-person human pose detection (same as OpenPose)
- Demo online available:
 - <https://storage.googleapis.com/tfjs-models/demos/posenet/camera.html>
 - If you have a webcam, you can try it
- Simpler than OpenPose, it has only 17 keypoints



PoseNet: how does it work?

- Source:
 - <https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5>

- Human pose detection
- Activity recognition
- Gesture recognition



wrnch.ai



https://www.youtube.com/watch?v=KBLlvNiI0Gs&ab_channel=wrnch

wrncn.ai vs OpenPose



https://www.youtube.com/watch?v=vTC0QKR_uM0&ab_channel=LearnOpenCV

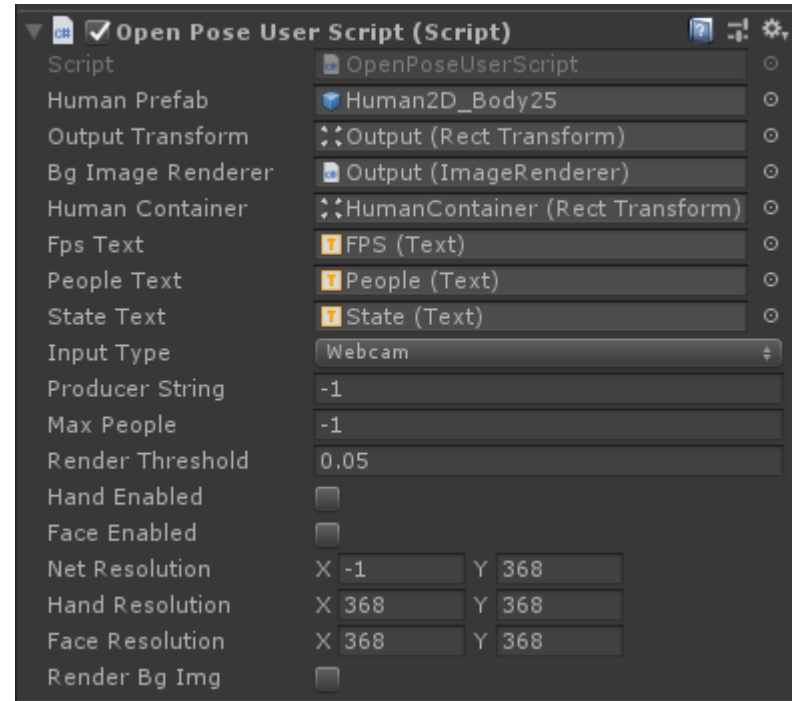
Unity

- Let's now see how to integrate these kinds of software/cameras in Unity.
- With RGB-D cameras, a skeleton will be animated following the human movements.
 - We will also briefly see how Unity animations work.



OpenPose & Unity

- OpenPose – Unity Plugin repository:
 - https://github.com/CMU-Perceptual-Computing-Lab/openpose_unity_plugin
- Unity 2018.4 recommended
- Clone or download the project.
- Run `getPlugins.bat`. This will automatically download and unzip OpenPose binaries and copy dlls to Unity.
- Run `getModels.bat`. This will automatically download required models for OpenPose.
- Open Unity and the `Demo.unity` scene
 - Located in `OpenPosePlugin/Assets/OpenPose/Examples/Scenes/`.
- Select OpenPose object from the hierarchy.



- You can test also with a video, set "Input Type" on "Video" and paste the local path of the video into the "Producer String" field.

PoseNet & Unity

- PoseNet – Unity Plugin repository:
 - <https://github.com/infocom-tpo/PoseNet-Unity>
- Unity 2018.4 recommended
- Clone or download the project; open the project in Unity.
- Import the following package:
 - <https://s3.amazonaws.com/unity-ml-agents/0.4/TFSharpPlugin.unitypackage>
- Open one of the two scenes in the Assets folder.

Nuitrack & Unity

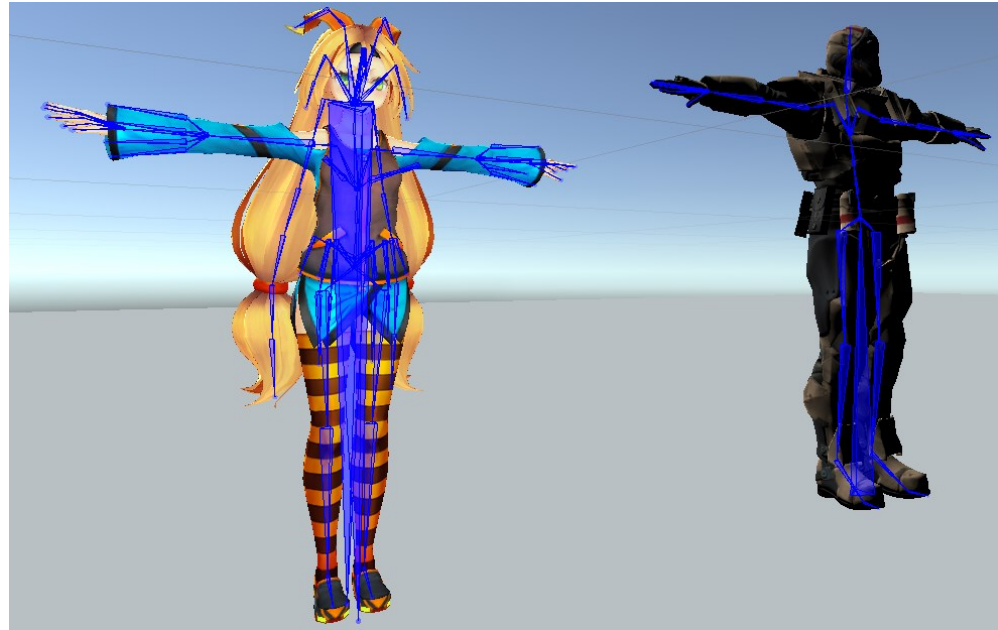
- If you have an RGB-D camera, you can download the Nuitrack package:
 - <https://github.com/3DiVi/nuitrack-sdk/blob/master/Unity3D/NuitrackSDK.unitypackage>
- Unity 2019.4 recommended
- Open Assets/NuitrackSDK/Tutorials/First Project
- Open NativeAvatar/TranslationAvatar.cs
 - In Update:
 - CurrentUserTracker has a static variable **CurrentSkeleton**
 - **CurrentSkeleton** is updated every frame, and it contains the information about the posture of the player

Nuitrack & Unity

- Open Assets/NuitrackSDK/Tutorials/Avatar Animation/Scenes/RiggedModel.scene
 - See:
 - NuitrackScripts -> Nuitrack Manager.cs
 - RiggedAvatar -> RiggedAvatar.cs
- Open the package manager
 - Advanced -> Show preview packages
 - Install «Animation Rigging»
- Select RiggedAvatar from the hierarchy.
 - Enable it
 - From the top menu: Animation Rigging -> Bone Renderer Setup

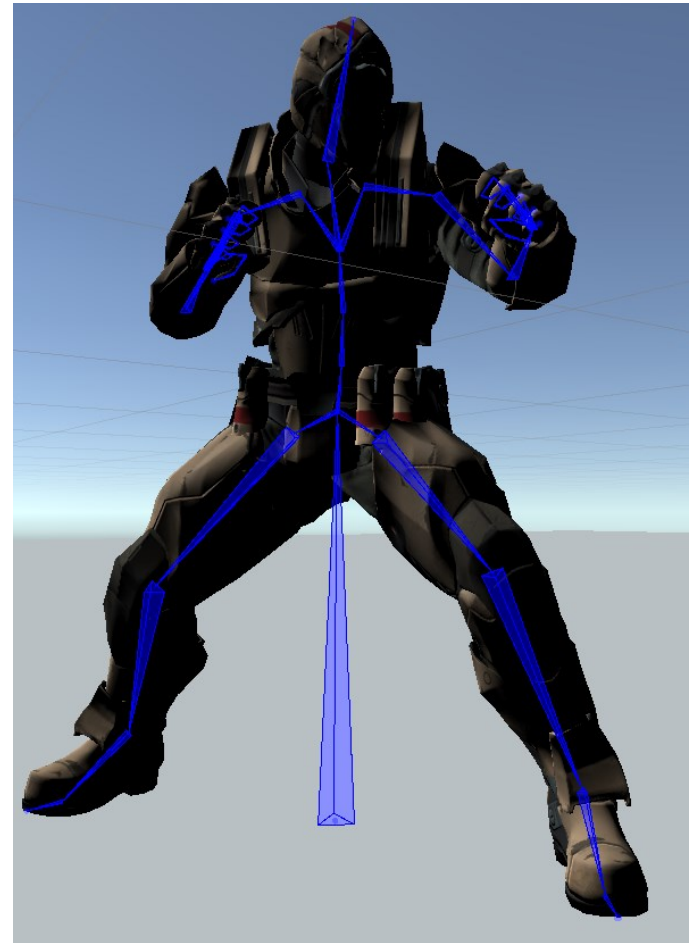
Mixamo: Characters

- You can download models and animations from Mixamo.
 - <https://www.mixamo.com/>
- Download the character you prefer and drag the .fbx file into Unity
 - It will not have the texture, to fix it:
 - Select the .fbx file
 - Select the Materials tab
 - Location -> Use External Materials (Legacy)
 - Apply. It could appear a windows telling you that you are using a texture as normal map, click on «Fix now».



Mixamo: Animation

- Download an animation from Mixamo.
 - After pressing on «Download» button, select «Without Skin» in the Skin option.
 - We have already download the model with the skin, so we do not need another skin.
- Select the model you have previously downloaded and added to the scene.
 - Add the «Animator Component»
 - Create a new Animator Controller
 - Open it and drag the animation you have just downloaded.
 - It will create a connection between the Entry node and the animation you have added.
 - Close the Animator tab, select the animation .fbx file, Animation tab -> set Loop Time to True
- Start the game



Unity Animation

- Let's play with Unity Animation

```
using UnityEngine;

public class PlayPunchingAnimation : MonoBehaviour
{
    public bool startPunching = false;

    private Animator aController;

    void Start()
    {
        aController = this.GetComponent<Animator>();
    }

    void Update()
    {
        if (startPunching && !aController.GetBool(name: "isPunching"))
            aController.SetBool(name: "isPunching", value: true);

        if (!startPunching && aController.GetBool(name: "isPunching"))
            aController.SetBool(name: "isPunching", value: false);
    }
}
```

