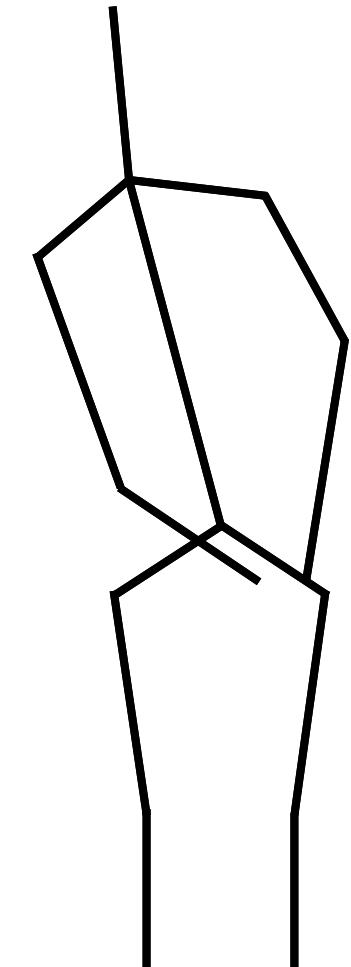


The Power of tracking the invisible: Geoinformatics and Human Social Behaviour

Violeta Ana Luz Sosa León
University of Münster



Personal Intro

Software Developer and Project Manager

Pursuing a Doctorate in **Geoinformatics**
in Germany

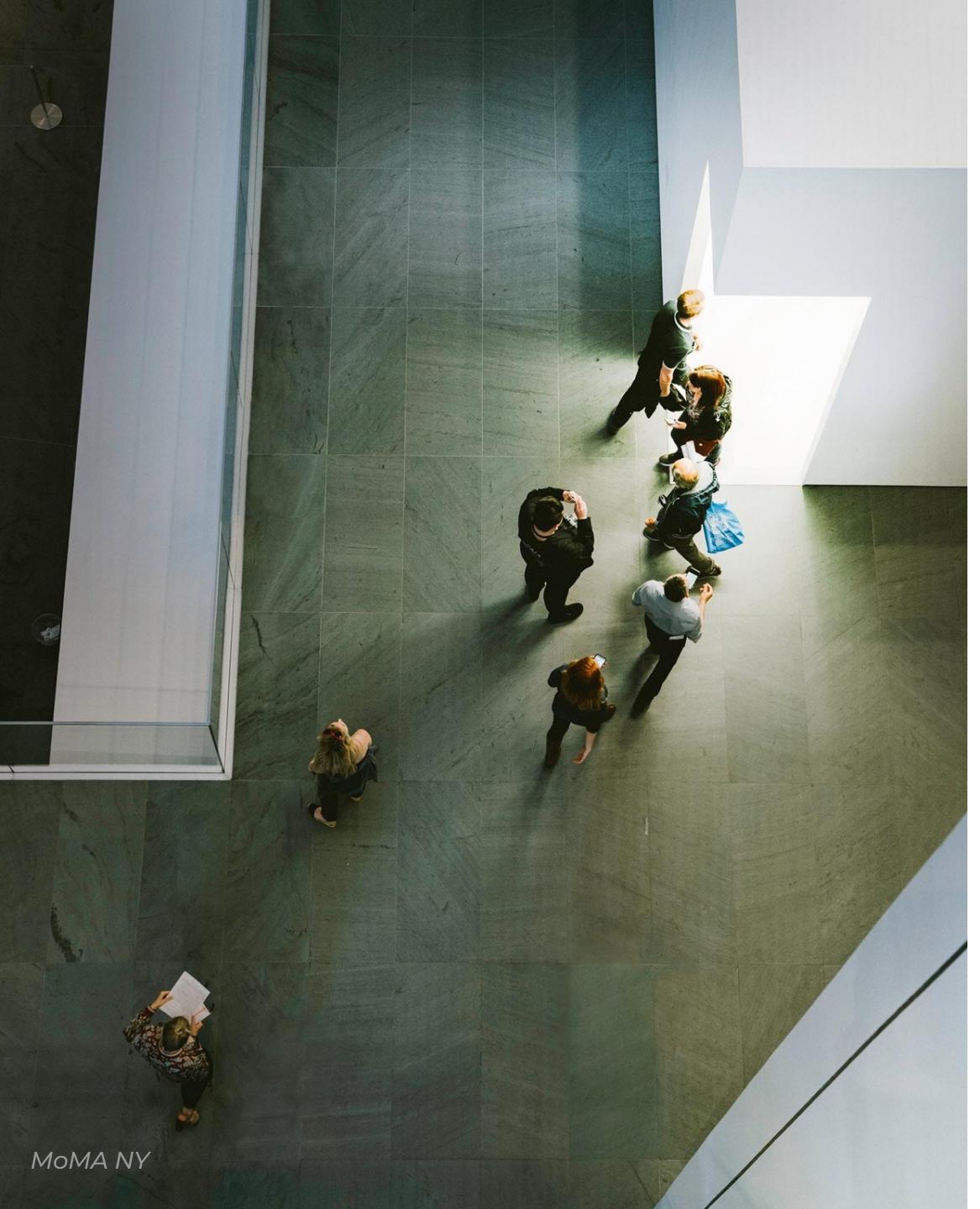
Interested in integrating **Spatial Cognition**
and **Computer Science**



Spatial Science

goes beyond geographic

Maps



MoMA NY

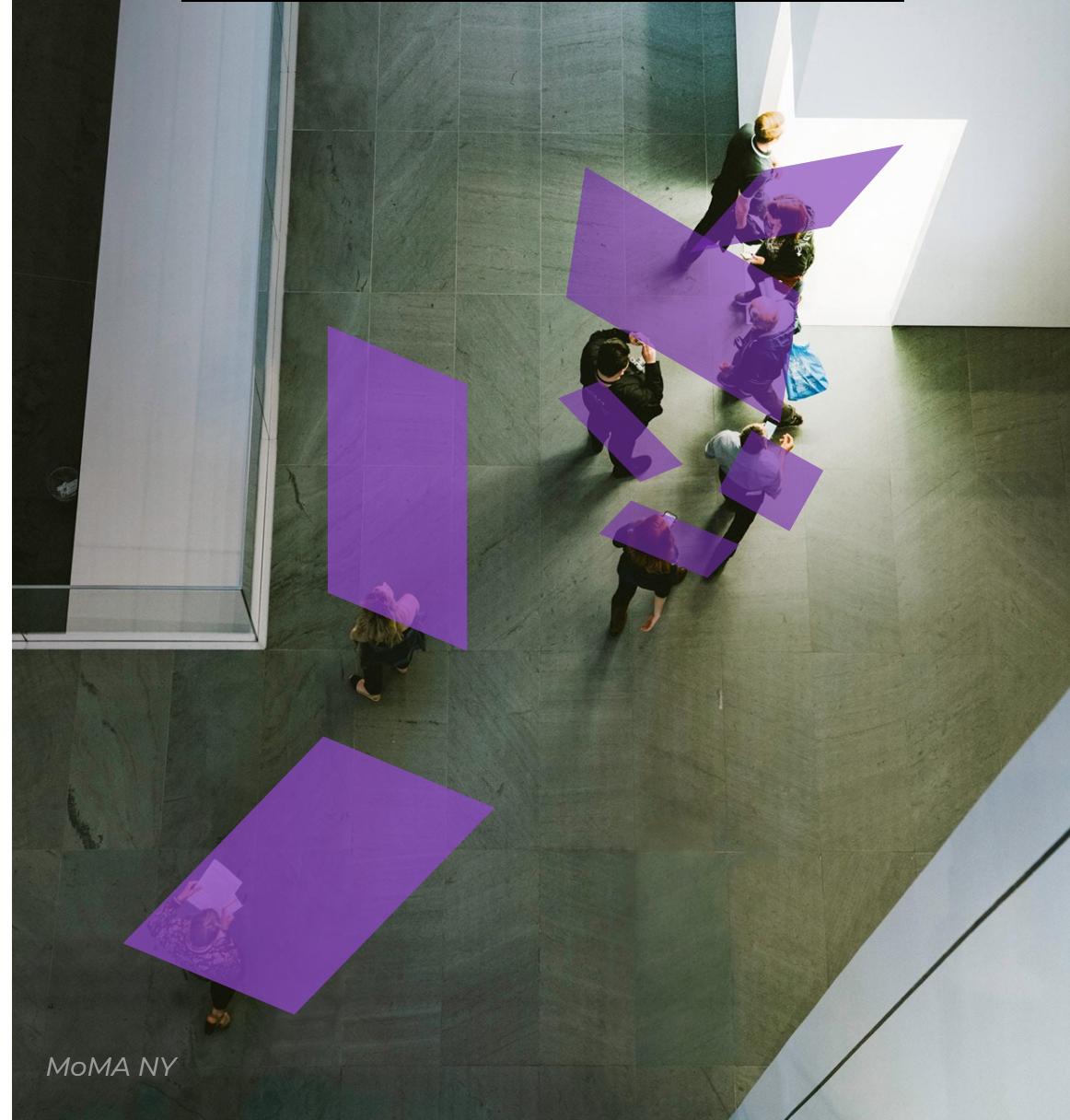
Physically occupied space

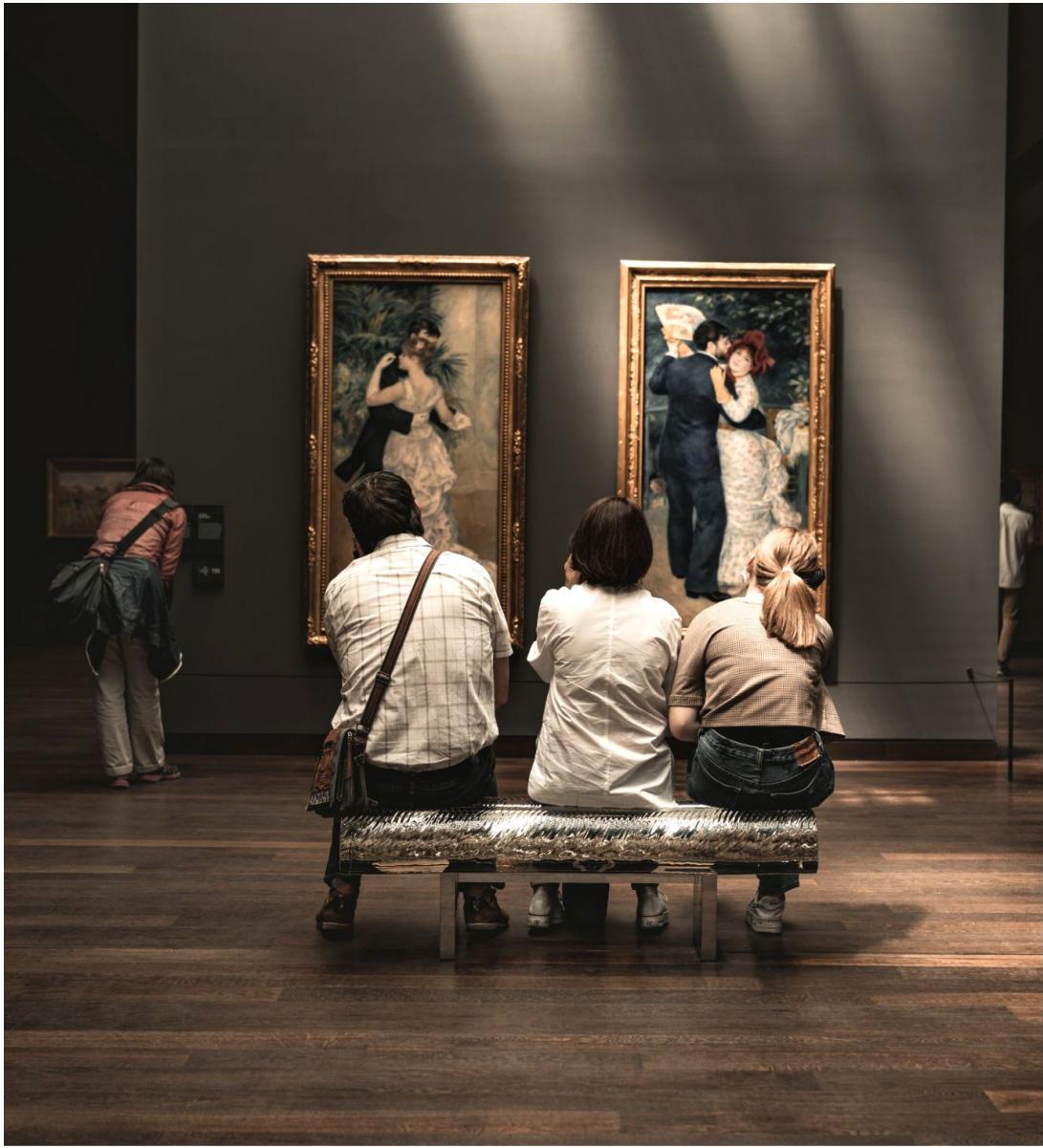


Physically occupied space



Socially occupied space







Why to track?

Improve architectural and exhibitions design



Offer better services in stores



Analyse behaviour during interactions

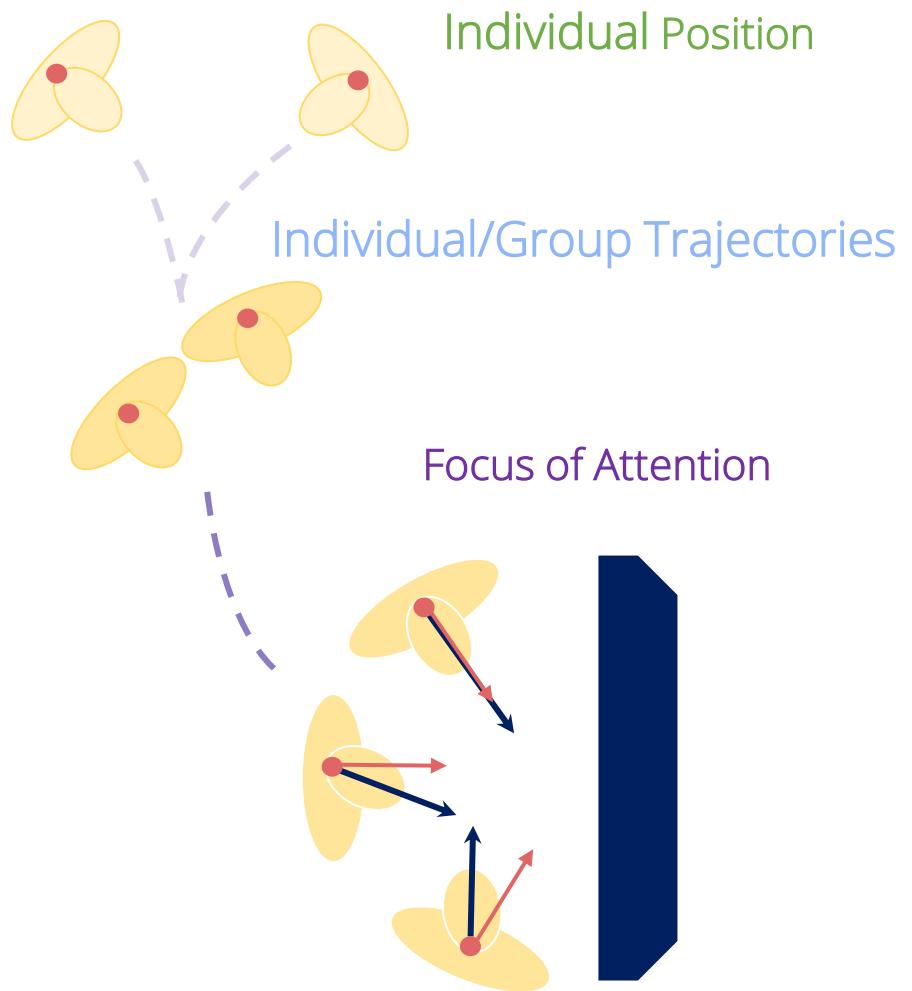


Appropriate social behaviour for robots

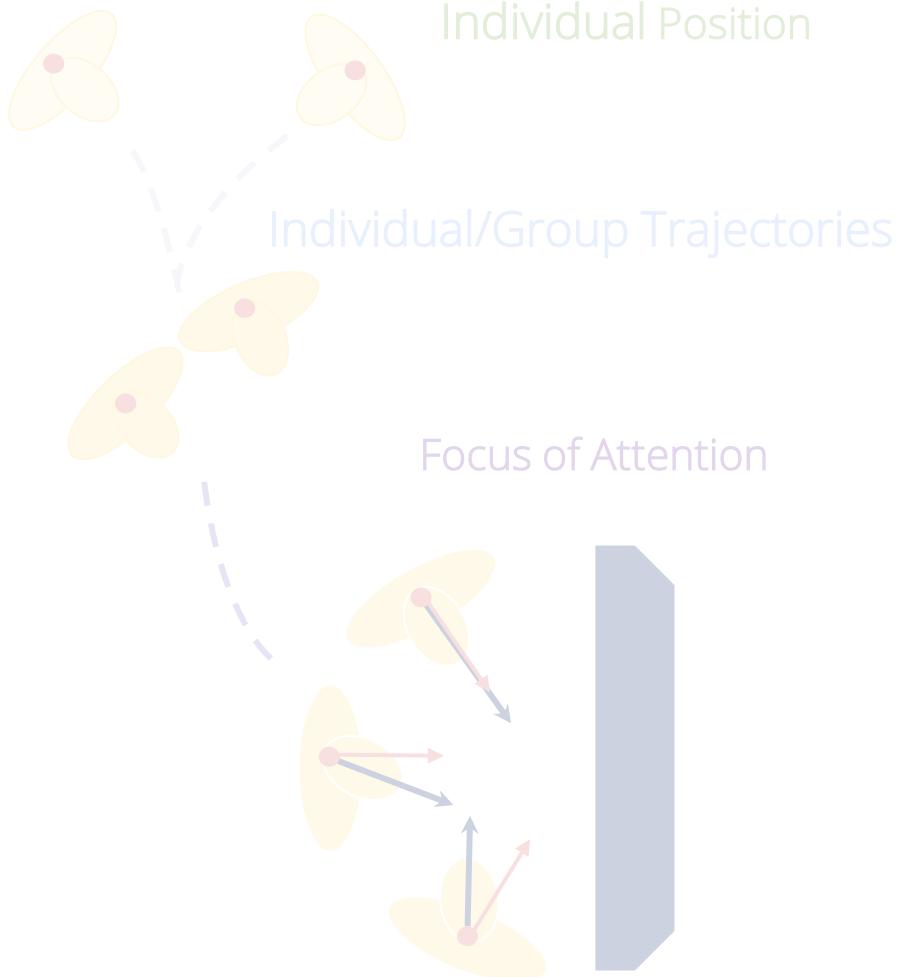


How can we detect
human behavior
automatically?

What to track?



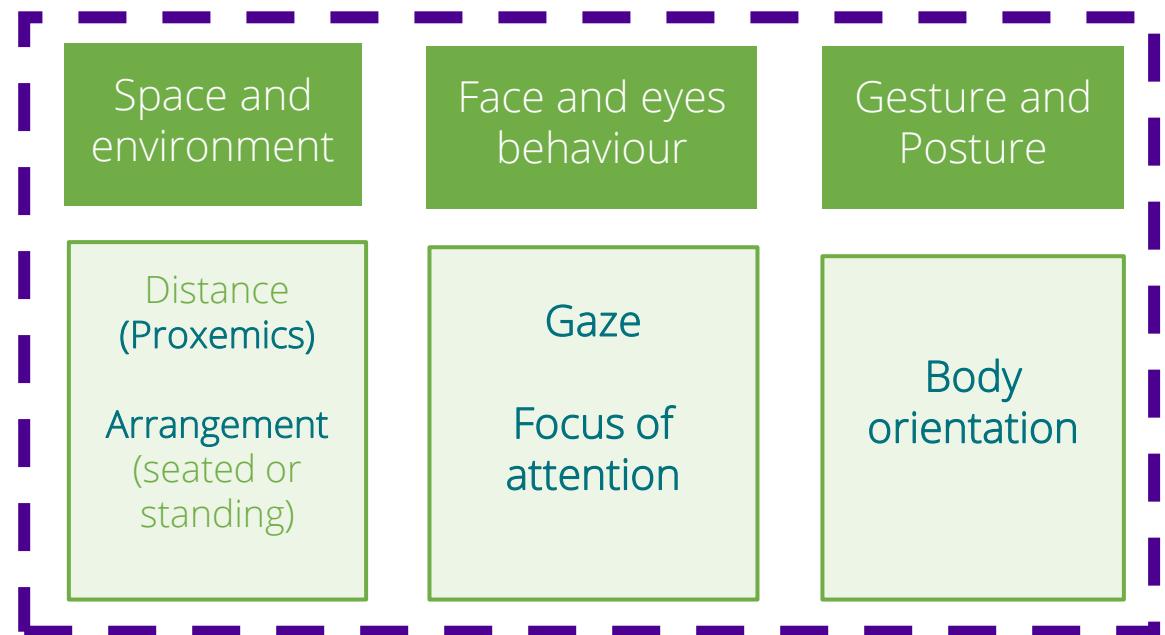
Social Signals
(Vinciarelli, 2009)



Groups

Interaction

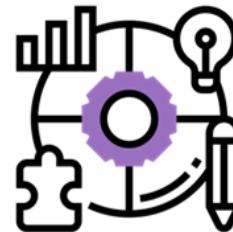
People



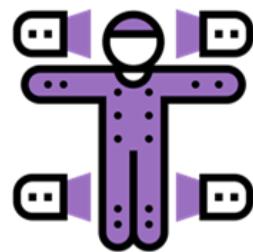
How to extract this data?



No manual
observation



Simple design
and setup

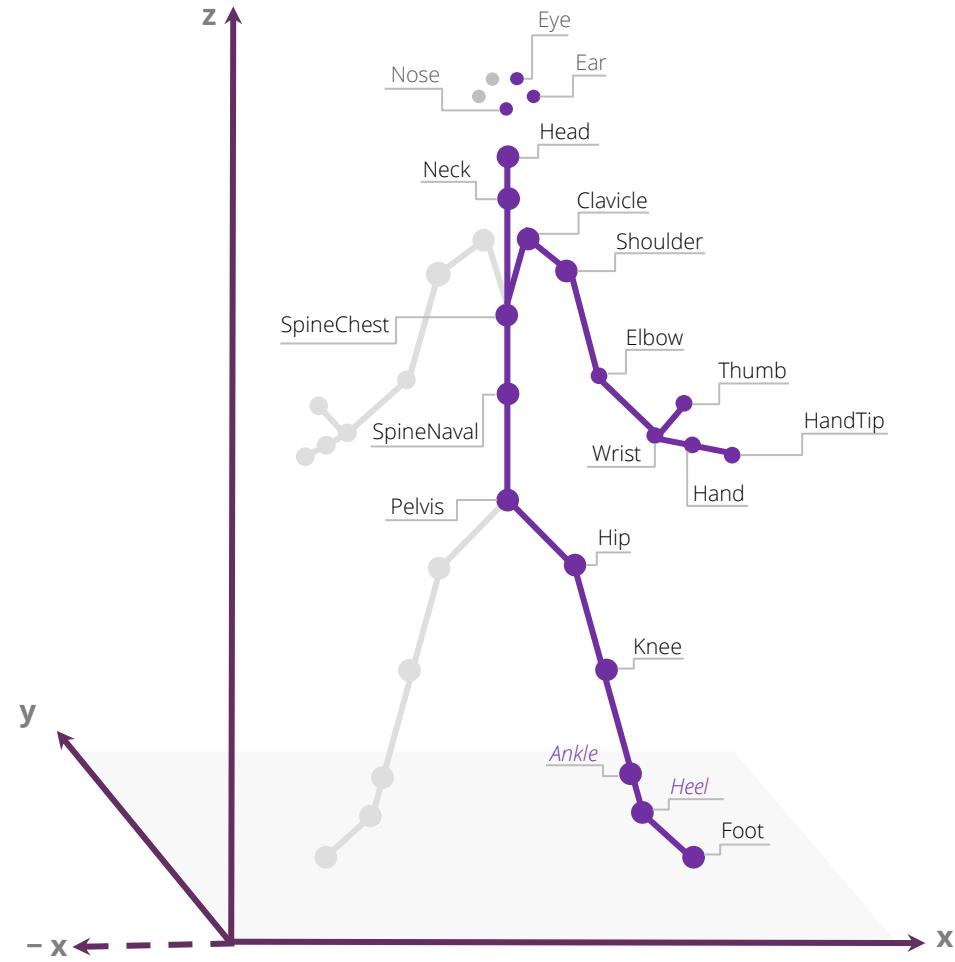


Unobtrusive

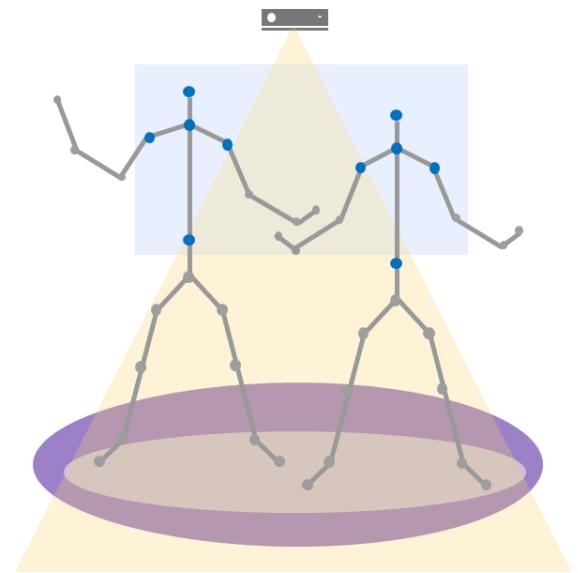
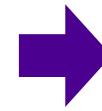
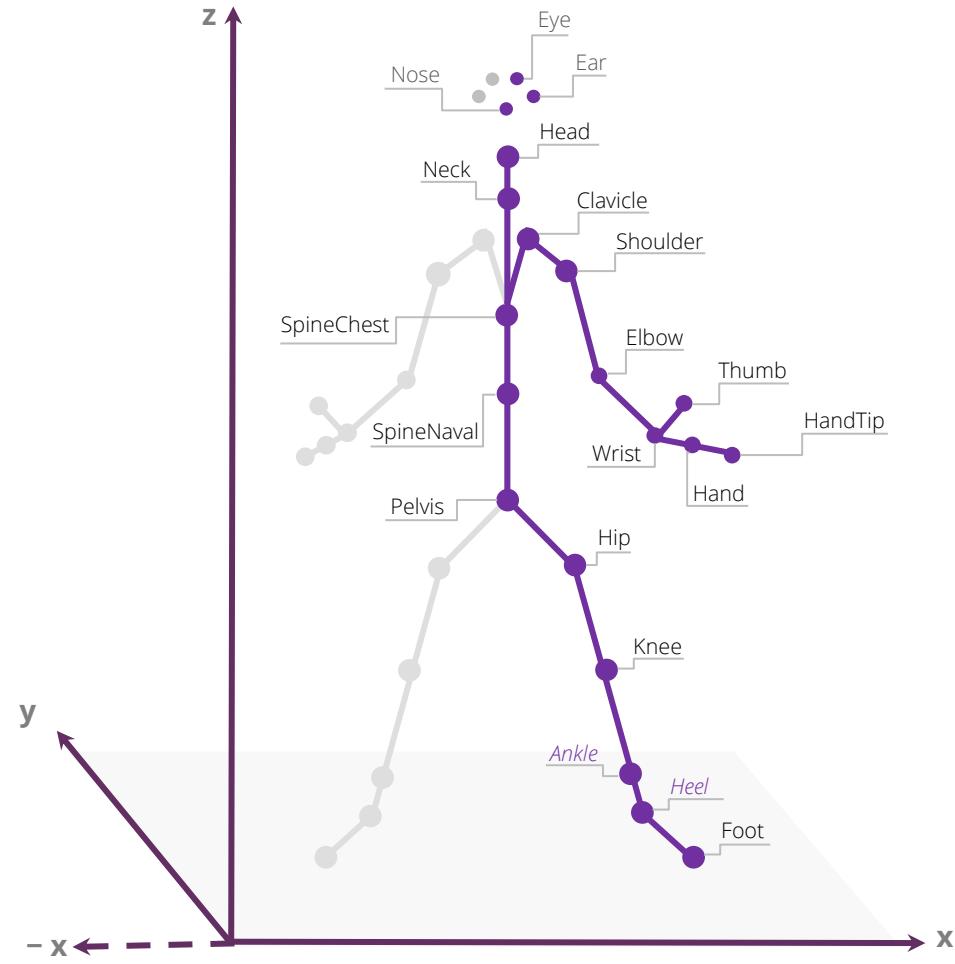


Identity safe

Using Skeleton Data



Using Skeleton Data



Using Skeleton Data



Which
technology?



Optical and vision

WebCam
Python, C++



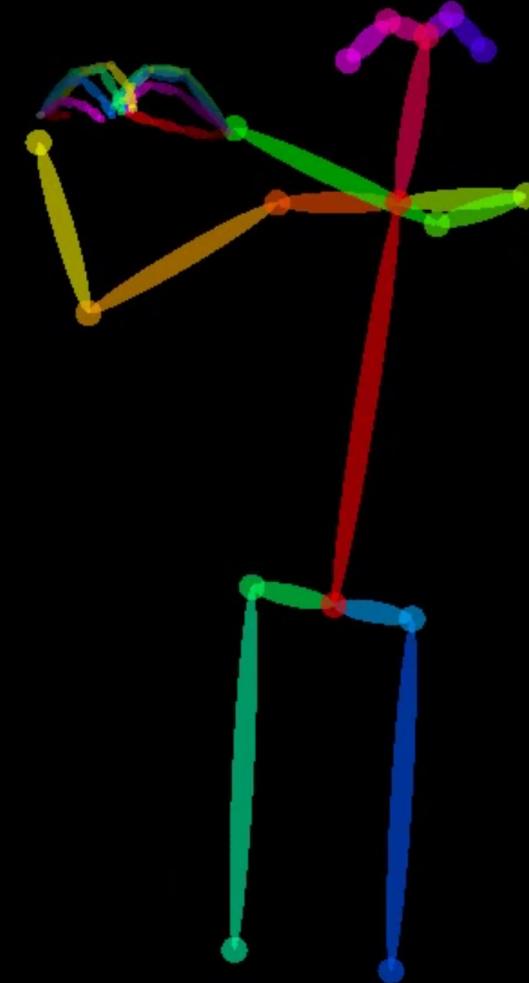
 STEREO LABS

Stereo Depth-Camera

Dedicated camera
Python, C++

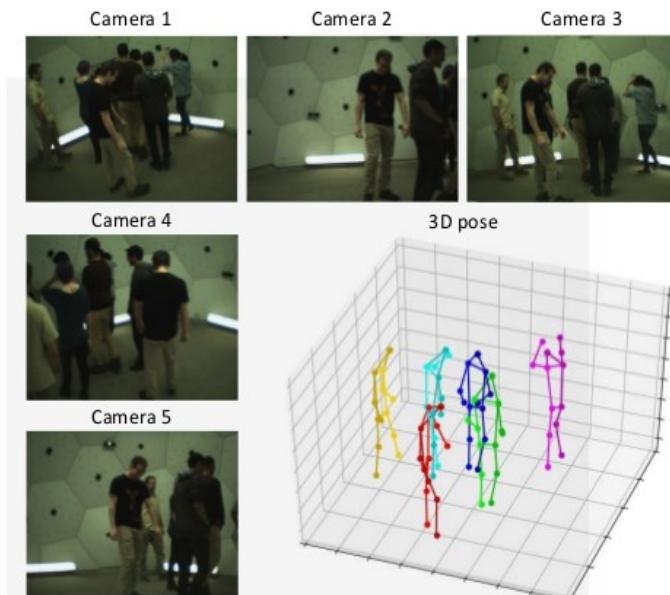
OpenPose

Optical and vision

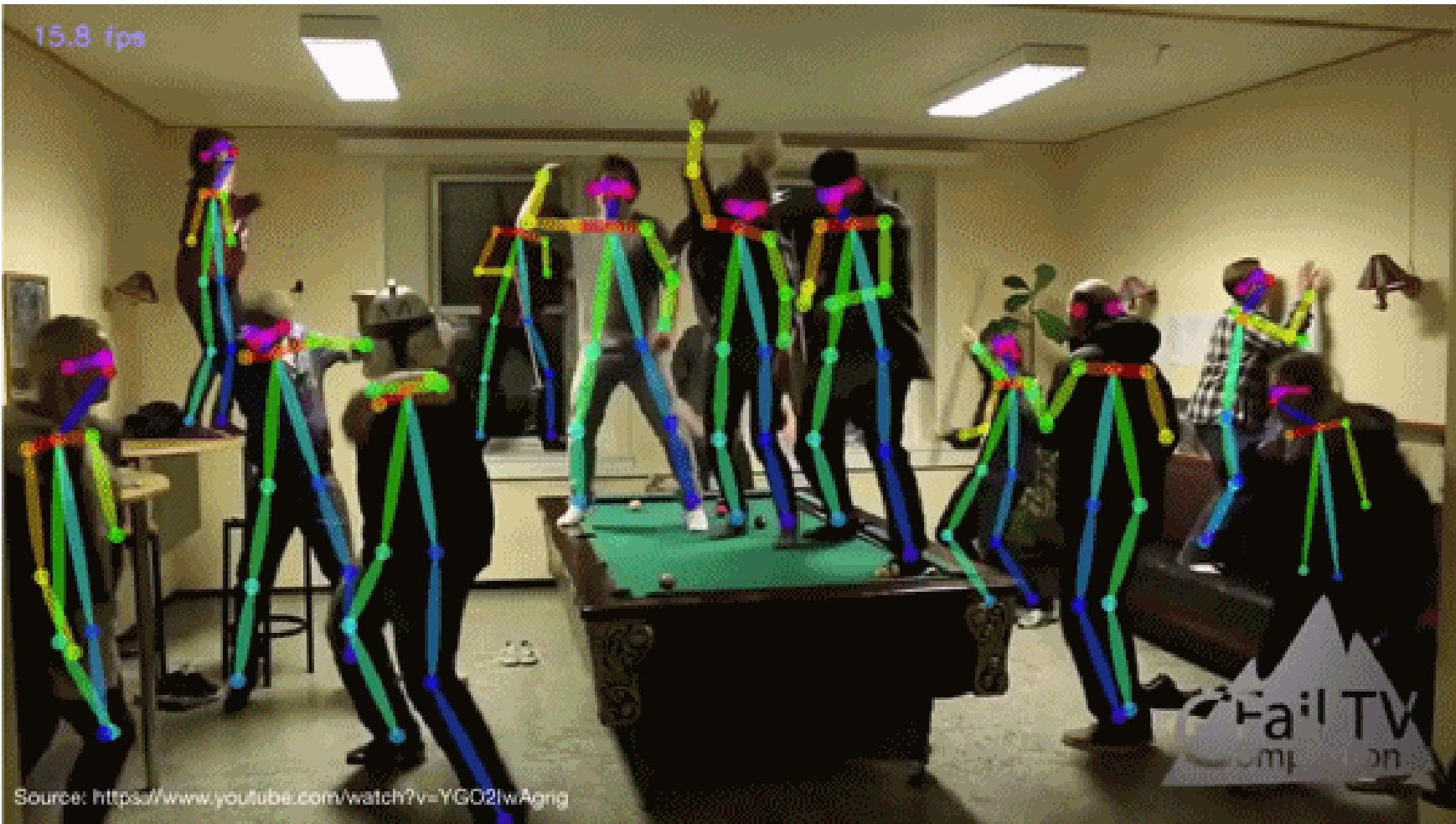


VISION

Accuracy	1cm - 1m
Cost	Medium - High
Complexity	High
Scalability	Low



[Multi-person and multi-view 3D pose reconstruction example from the Panoptic dataset by Dong et al. 2019](#)

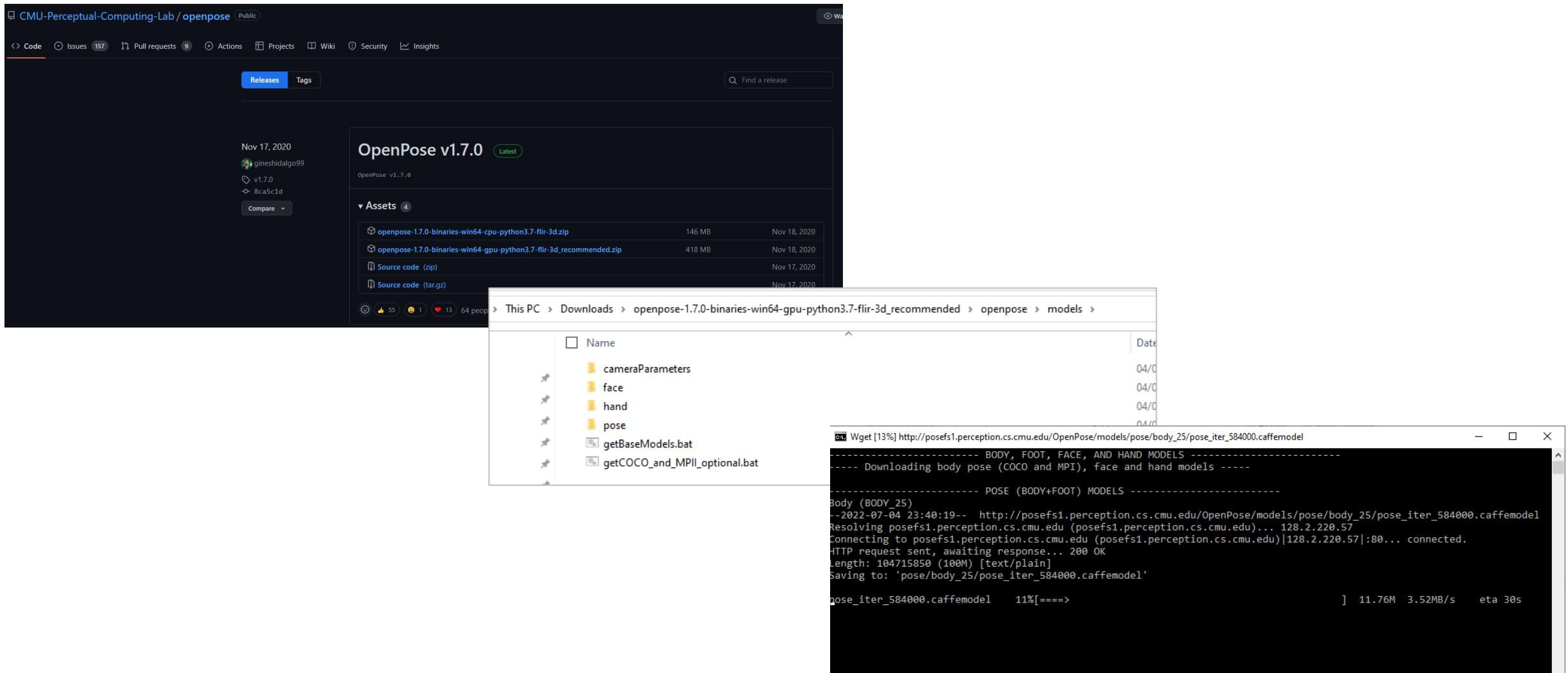


OpenPose: Install and Configure

Detailed steps in:

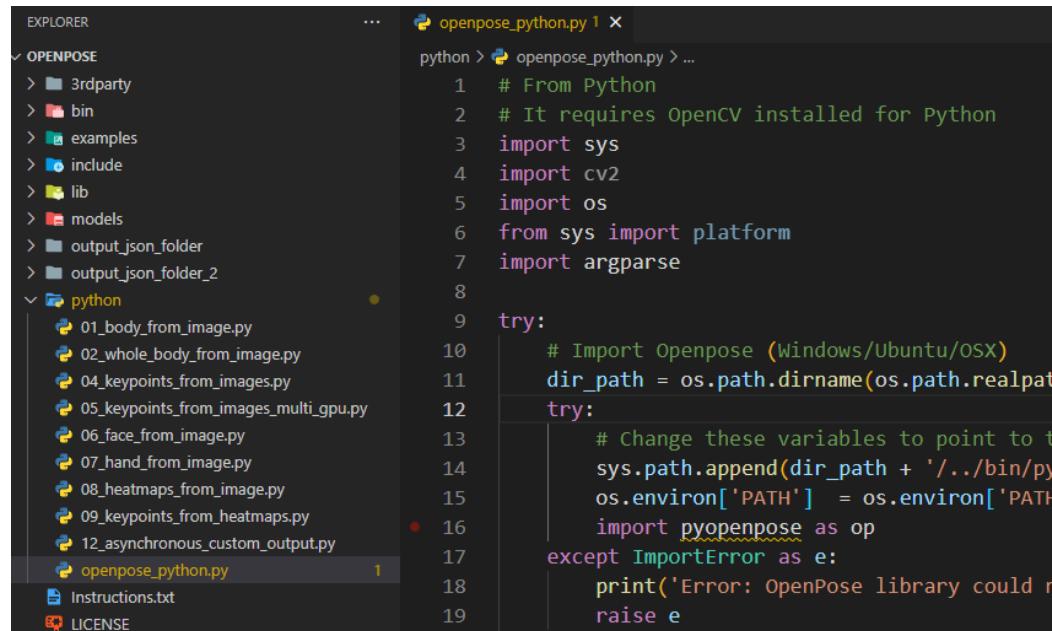
- [openpose/doc at master · CMU-Perceptual-Computing-Lab/openpose \(github.com\)](https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/installation.md)

Library setup



Development environment setup

Manual Install

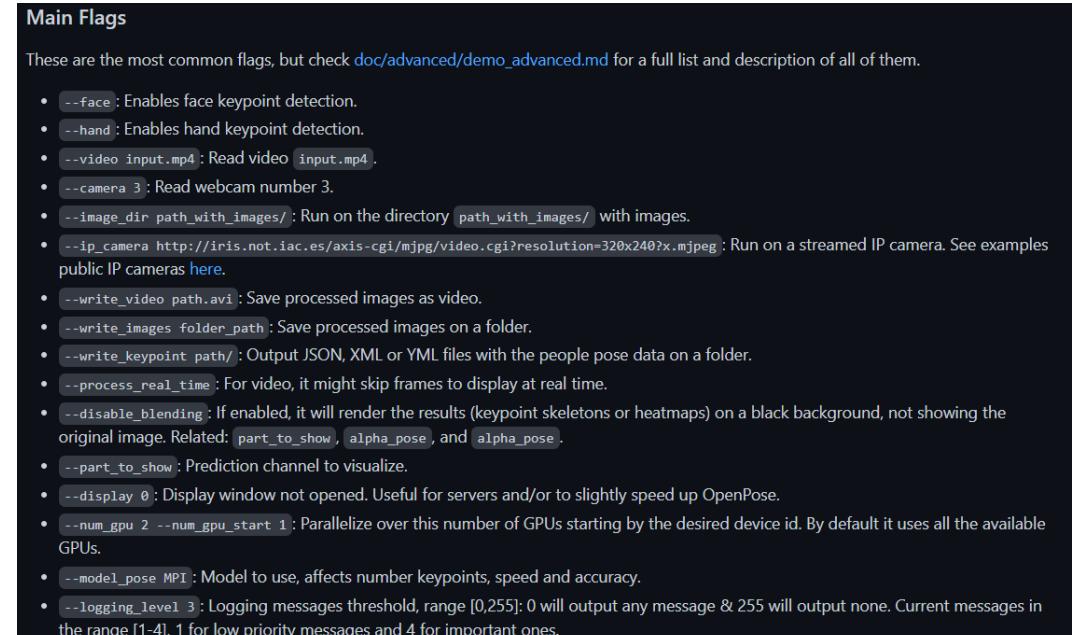


A screenshot of a code editor showing the `openpose_python.py` script. The code imports necessary modules and sets up a try-except block to handle errors. It includes comments explaining the import of Openpose and the modification of the PATH variable. The code editor has a dark theme with syntax highlighting for Python.

```
EXPLORER ... openpose_python.py 1 x
OPENPOSE
> 3rdparty
> bin
> examples
> include
> lib
> models
> output_json_folder
> output_json_folder_2
python
01_body_from_image.py
02_whole_body_from_image.py
04_keypoints_from_images.py
05_keypoints_from_images_multi_gpu.py
06_face_from_image.py
07_hand_from_image.py
08_heatmaps_from_image.py
09_keypoints_from_heatmaps.py
12_asynchronous_custom_output.py
openpose_python.py 1
Instructions.txt
LICENSE

python > openpose_python.py > ...
1 # From Python
2 # It requires OpenCV installed for Python
3 import sys
4 import cv2
5 import os
6 from sys import platform
7 import argparse
8
9 try:
10     # Import Openpose (Windows/Ubuntu/OSX)
11     dir_path = os.path.dirname(os.path.realpath(__file__))
12     try:
13         # Change these variables to point to the
14         # directory containing the main OpenPose library
15         # and its dependencies
16         sys.path.append(dir_path + '/../bin/python')
17         os.environ['PATH'] = os.environ['PATH'] + ':.' + dir_path
18         import pyopenpose as op
19     except ImportError as e:
20         print('Error: OpenPose library could not be '
21             'imported. Did you correctly install '
22             'it? Refer to the documentation at '
23             'https://github.com/CMU-Perception-Lab/OpenPose#installation.')
24         raise e
```

Terminal



A screenshot of a terminal window displaying the "Main Flags" section of the documentation. It lists various command-line flags with their descriptions, such as `--face` for face keypoint detection and `--ip_camera` for running on a streamed IP camera.

Main Flags

These are the most common flags, but check [doc/advanced/demo_advanced.md](#) for a full list and description of all of them.

- `--face`: Enables face keypoint detection.
- `--hand`: Enables hand keypoint detection.
- `--video input.mp4`: Read video `input.mp4`.
- `--camera 3`: Read webcam number 3.
- `--image_dir path_with_images/`: Run on the directory `path_with_images/` with images.
- `--ip_camera http://iris.not.iac.es/axis-cgi/mjpg/video.cgi?resolution=320x240?x.mjpeg`: Run on a streamed IP camera. See examples public IP cameras [here](#).
- `--write_video path.avi`: Save processed images as video.
- `--write_images folder_path`: Save processed images on a folder.
- `--write_keypoint path/`: Output JSON, XML or YML files with the people pose data on a folder.
- `--process_real_time`: For video, it might skip frames to display at real time.
- `--disable_blending`: If enabled, it will render the results (keypoint skeletons or heatmaps) on a black background, not showing the original image. Related: `part_to_show`, `alpha_pose`, and `alpha_pose`.
- `--part_to_show`: Prediction channel to visualize.
- `--display 0`: Display window not opened. Useful for servers and/or to slightly speed up OpenPose.
- `--num_gpu 2 --num_gpu_start 1`: Parallelize over this number of GPUs starting by the desired device id. By default it uses all the available GPUs.
- `--model_pose MPI`: Model to use, affects number keypoints, speed and accuracy.
- `--logging_level 3`: Logging messages threshold, range [0,255]: 0 will output any message & 255 will output none. Current messages in the range [1-4], 1 for low priority messages and 4 for important ones.

OpenPose: Implementation

The screenshot shows a GitHub repository page for 'CMU-Perceptual-Computing-Lab / openpose'. The repository is public. The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, there are links for Code (highlighted with a red box), Issues (157), Pull requests (9), Actions, Projects, Wiki, Security, and Insights.

The current view is on the 'Code' tab, specifically the file 'openpose / doc / 01_demo.md'. A red box highlights the file path 'openpose / doc / 01_demo.md' in the breadcrumb navigation area. The file was last updated by 'luzpaz' on March 11, 2024, with commit e835524. It has 2 contributors.

The file content is titled 'OpenPose Doc - Demo'. It contains the following text:

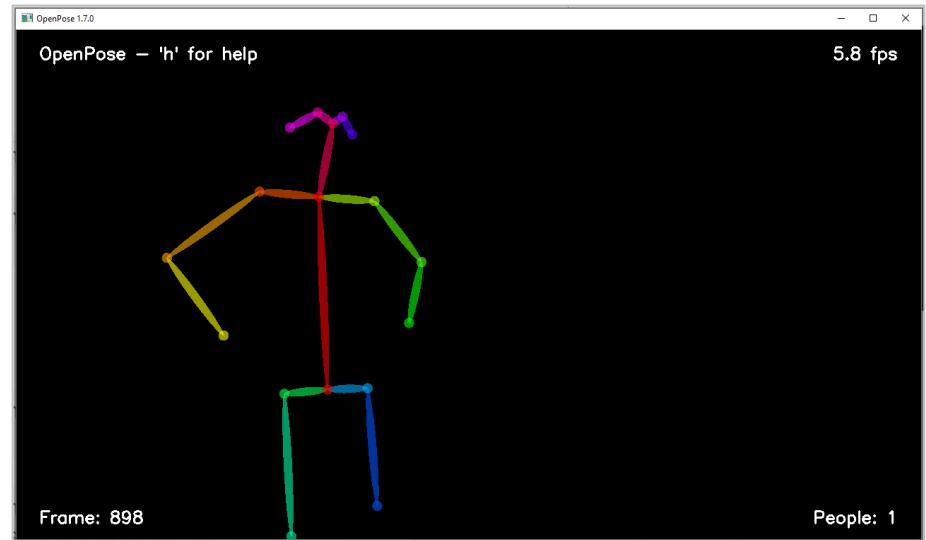
Forget about the OpenPose code, just download the portable Windows binaries (or compile the code from source) and use the demo by following this tutorial!

Contents

1. Quick Start
 - i. Running on Images, Video, or Webcam
 - ii. Face and Hands
 - iii. Different Outputs (JSON, Images, Video, UI)
 - iv. Only Skeleton without Background Image
 - v. Not Running All GPUs
 - vi. Maximum Accuracy Configuration

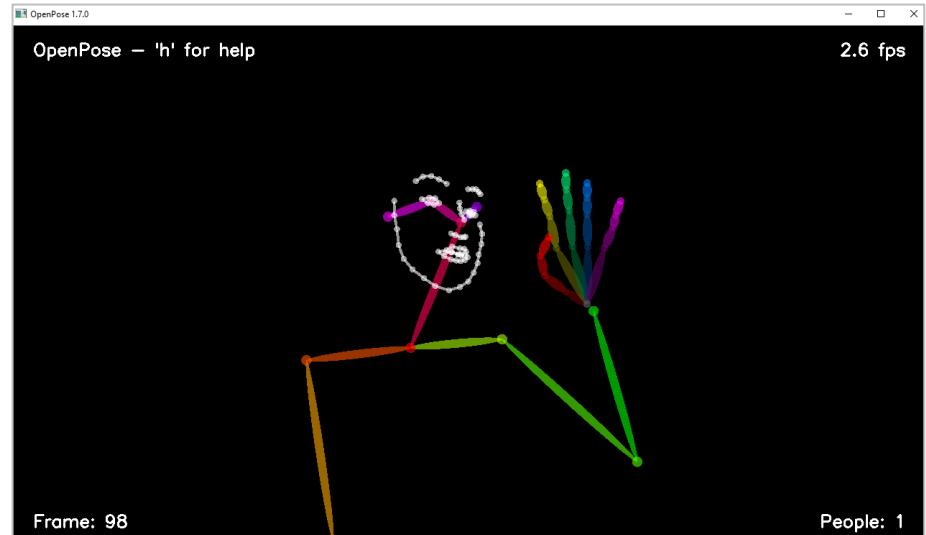
```
$ OpenPosePath .\bin\OpenPoseDemo.exe --camera 0
```

```
$ OpenPosePath .\bin\OpenPoseDemo.exe --camera 0  
--disable_blending
```



```
$ OpenPosePath .\bin\OpenPoseDemo.exe --camera 0  
--disable_blending --hand --face
```

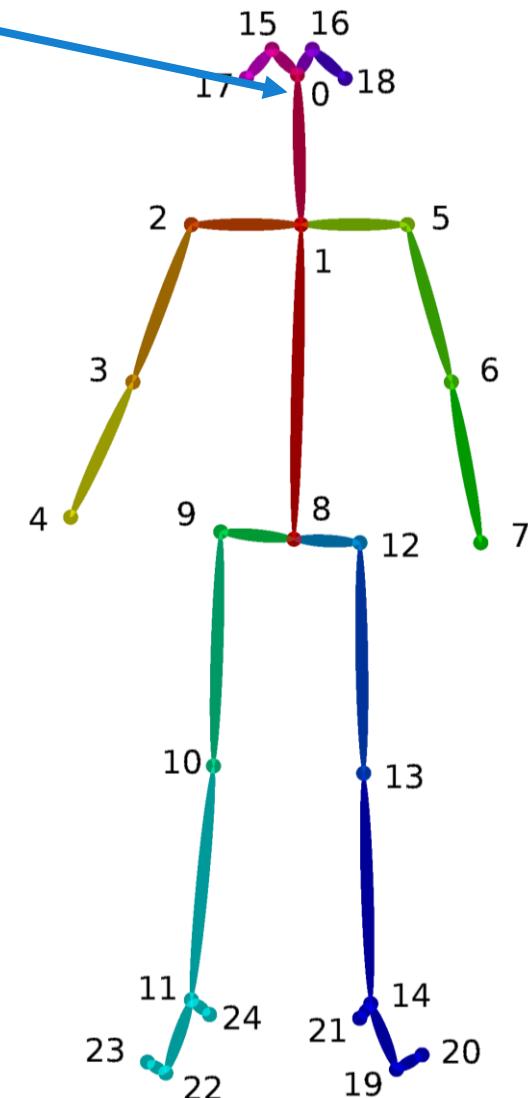
```
$ OpenPosePath .\bin\OpenPoseDemo.exe --camera 0  
--disable_blending --write_json output_dir/
```



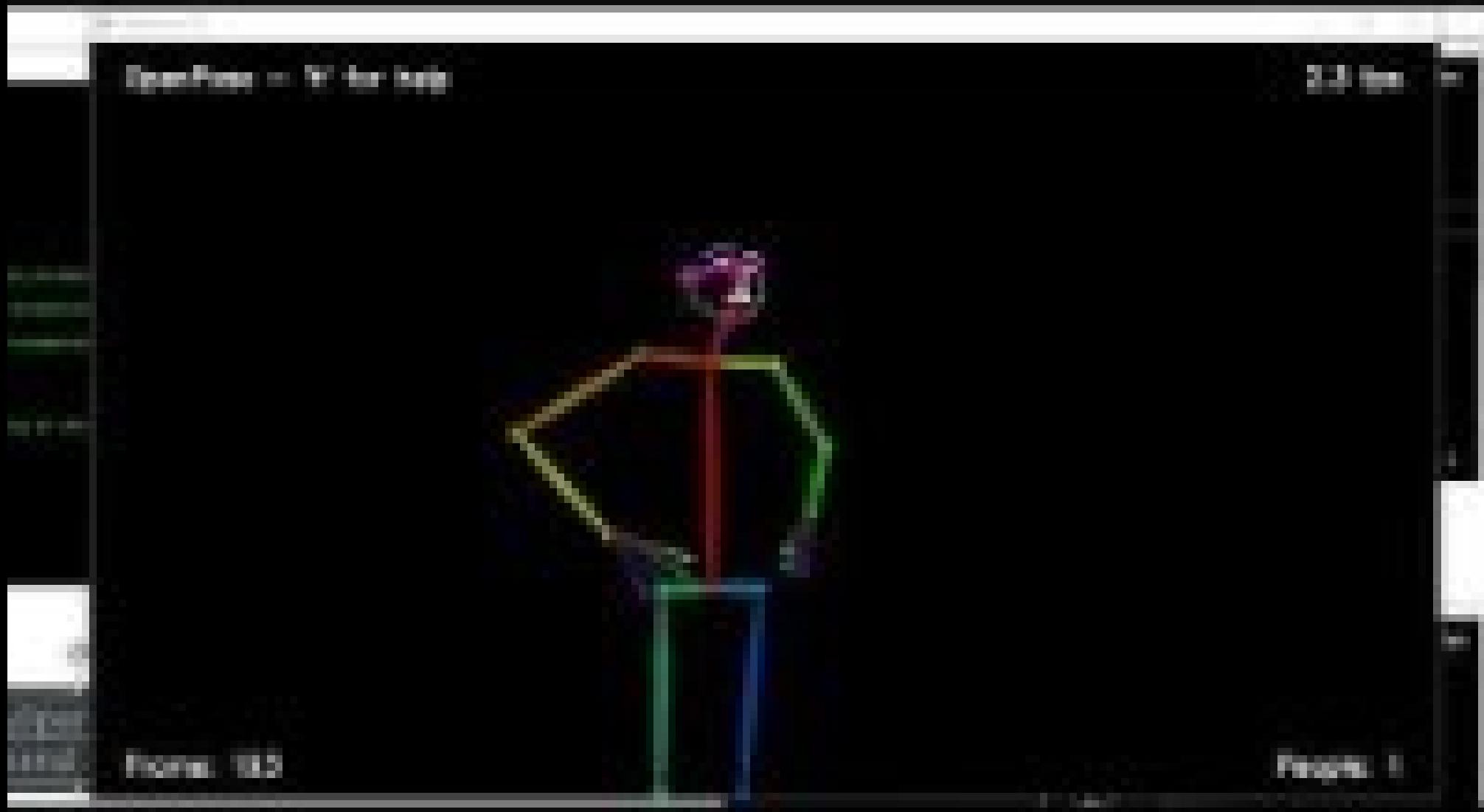
Coordinates [x, y]

```
{  
    "version": 1.3,  
    "people": [  
        {  
            "person_id": [-1],  
            "pose_keypoints_2d": [],  
            "460.801, 335.437, 0.850702,  
            400.14, 505.938, 0.582054,  
            280.543, 490.276, 0.35841,  
            0, 0, 0,  
            0, 0, 0,  
            525.609, 513.785, 0.599199,  
            388.262, 715.022, 0.582557,  
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
            421.75, 319.851, 0.86914,  
            478.469, 333.515, 0.868618,  
            355.145, 353.095, 0.75606,  
            494.225, 378.595, 0.0532065  
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0  
        ],  
        "face_keypoints_2d": [],  
        "hand_left_keypoints_2d": [],  
        "hand_right_keypoints_2d": [],  
        "pose_keypoints_3d": [],  
        "face_keypoints_3d": [],  
        "hand_left_keypoints_3d": [],  
        "hand_right_keypoints_3d": []  
    ],  
    "part_candidates": [  
        {"0": [  
            -80.448, 229.773, 0.150741,  
            886.051, 259.087, 0.0928216,  
            982.02, 296.339, 0.159637,  
            460.801, 335.437, 0.850702  
        ]}  
    ]  
}
```

Joint key

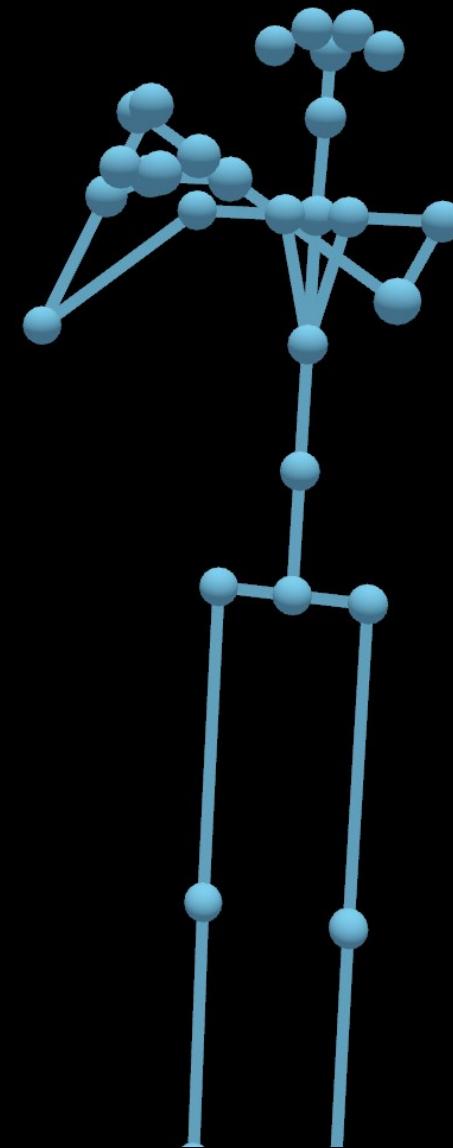


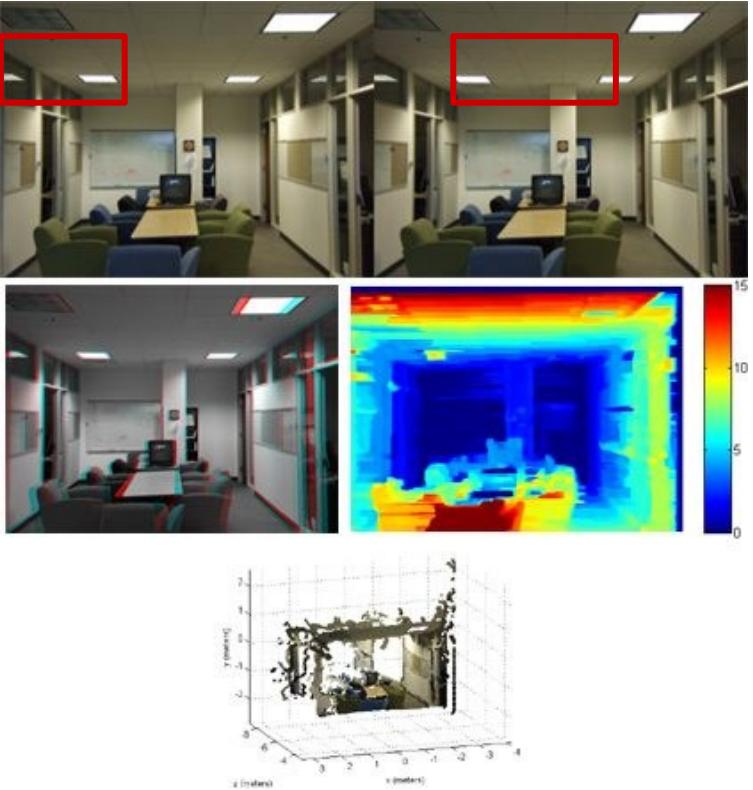
OpenPose: Test



Zed 2i

Stereo Depth Camera





[Mathworks Stereo Vision](#)

STEREOVISION

(stereoscopic or camera array)

Accuracy	mm to cm Base-line (spacing between cameras) dependent Mid range
Range	1-20m
Cost	Medium - High
Complexity	High
Scalability	Medium
Light performance	Low: Weak - Bright: Good



Measurement and
Inspection

Augmented Reality



Identification



Location

Zed2i: Install and Configure

Detailed steps in

- [Zed 2i · violetasdev/bodytrackingdepth course Wiki \(github.com\)](#)
- [Stereolabs Docs: API Reference, Tutorials, and Integration](#)

Camera setup

Requirement	Detail
Operative system	Windows 10, Ubuntu 16.04, 18.04
Processor	Quad-core 2,7GHz or faster
RAM	8 GB
Graphics driver	NVIDIA GPU with Compute Capabilities >3.0 - Support CUDA
Graphics card	NVIDIA GEFORCE GTX 1060 or higher
USB connector	USB 3.0 port this is very important



Camera setup

SDK Downloads

The ZED SDK allows you to add depth, motion sensing and spatial AI to your application. Available as a standalone installer, it includes applications, tools and sample projects with source code. Please check out our [GitHub](#) page and [SDK documentation](#) for additional resources.

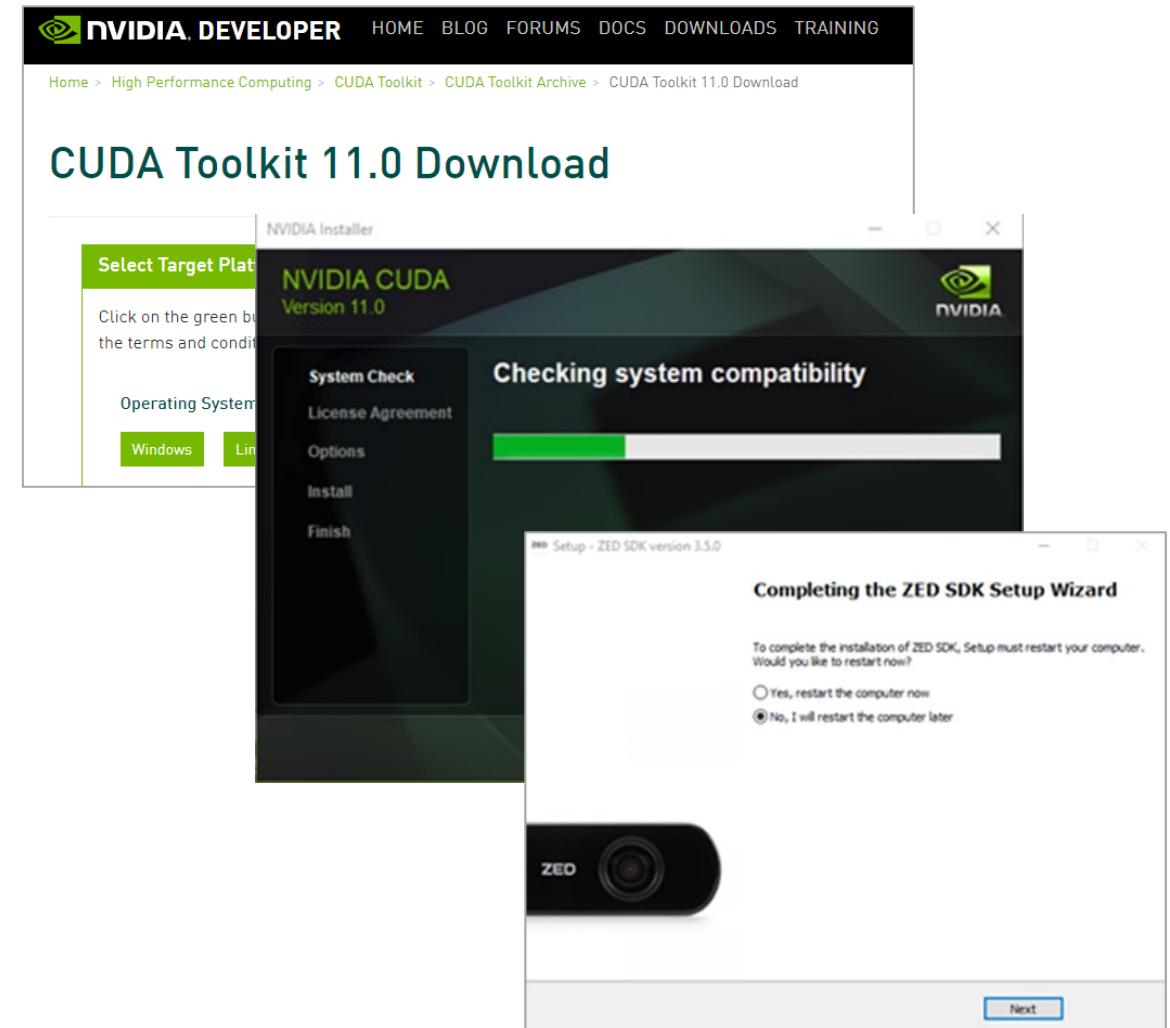
CUDA 11.X (11.0*->11.5)

ZED SDK for Windows 10/11 3.7.4 (*CUDA 11.0 is not supported)

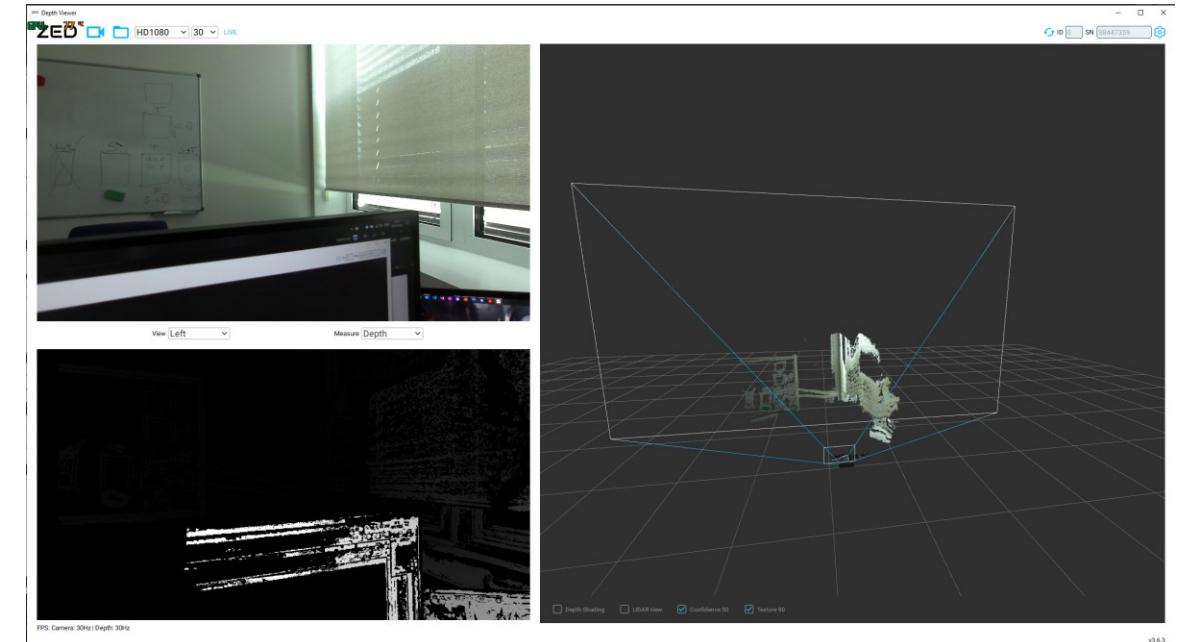
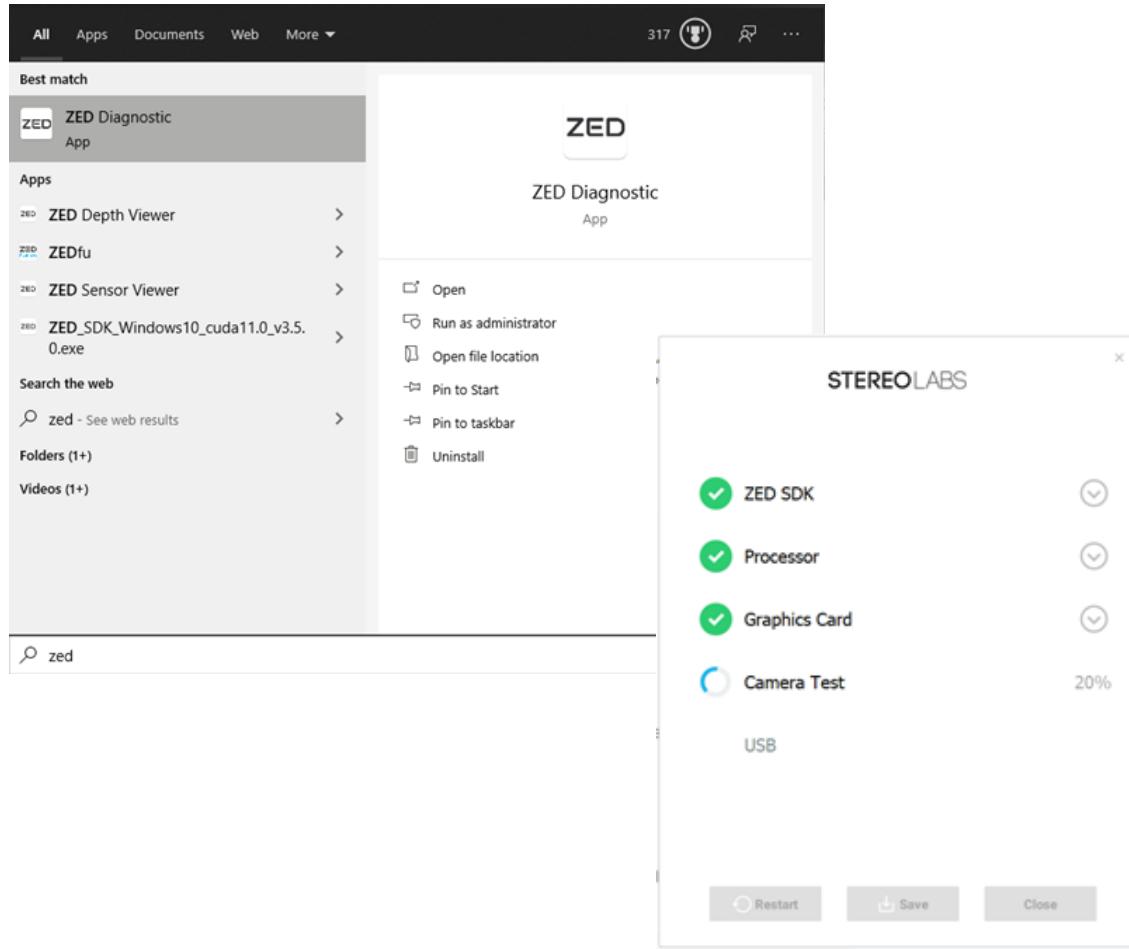
ZED SDK for Ubuntu 20 3.7.4

ZED SDK for Ubuntu 18 3.7.4

Stereolabs



Camera setup



Development environment setup

Manual Install

Windows

```
(zed_camera) PS C:\Program Files (x86)\ZED SDK> python3.exe .\get_python_api.py
-> Downloading to 'C:\Program Files (x86)\ZED SDK'
Detected platform:
    win
    Python 3.9
    ZED SDK 3.6
-> Checking if https://download.stereolabs.com/zedsdk/3.6/win/py39 exists and is available
```

Linux

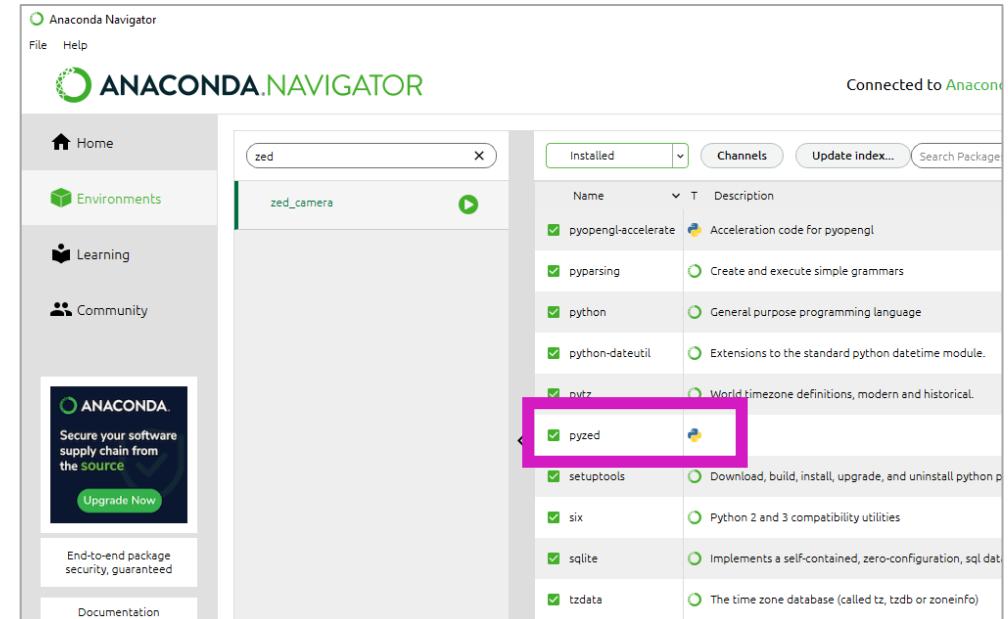
```
$ cd "/usr/local/zed/"
$ python3 get_python_api.py

# The script displays the detected platform versions
Detected platform:
    ubuntu18
    Python 3.6
    CUDA 11.0
    ZED SDK 3.5

# Downloads the corresponding whl package
Checking if https://download.stereolabs.com/zedsdk/3.5/ubuntu18/cu110/py36 exists and is available
```

Stereolabs

Anaconda Environment



Zed2i: Implementation

 Search or jump to... / Pull requests Issues Marketplace Explore

 + 

 **stereolabs / zed-examples** Public

Watch 24 Fork 339 Star 463

<> **Code** Issues 22 Pull requests Actions Wiki Security Insights

master ▾ 3 branches 2 tags Go to file Add file ▾ **Code ▾**

 adujardin Update README.md ... 8cdb122 4 days ago 176 commits

File/Folder	Description	Last Commit
.github	Update stale_issues.yml	3 months ago
body tracking	3.7 sample upgrade (#483)	2 months ago
camera control	3.7 sample upgrade (#483)	2 months ago
camera streaming	3.7 sample upgrade (#483)	2 months ago
depth sensing	3.7 sample upgrade (#483)	2 months ago
object detection	Update README.md	4 days ago
other	3.7 sample upgrade (#483)	2 months ago
plane detection	3.7 sample upgrade (#483)	2 months ago
positional tracking	3.7 sample upgrade (#483)	2 months ago

About

ZED SDK Example projects

 www.stereolabs.com/developers/

tutorial slam stereo-vision
zed-camera

Readme MIT license 463 stars 24 watching 339 forks

Releases 2

 v2.X Latest on Jan 24, 2020

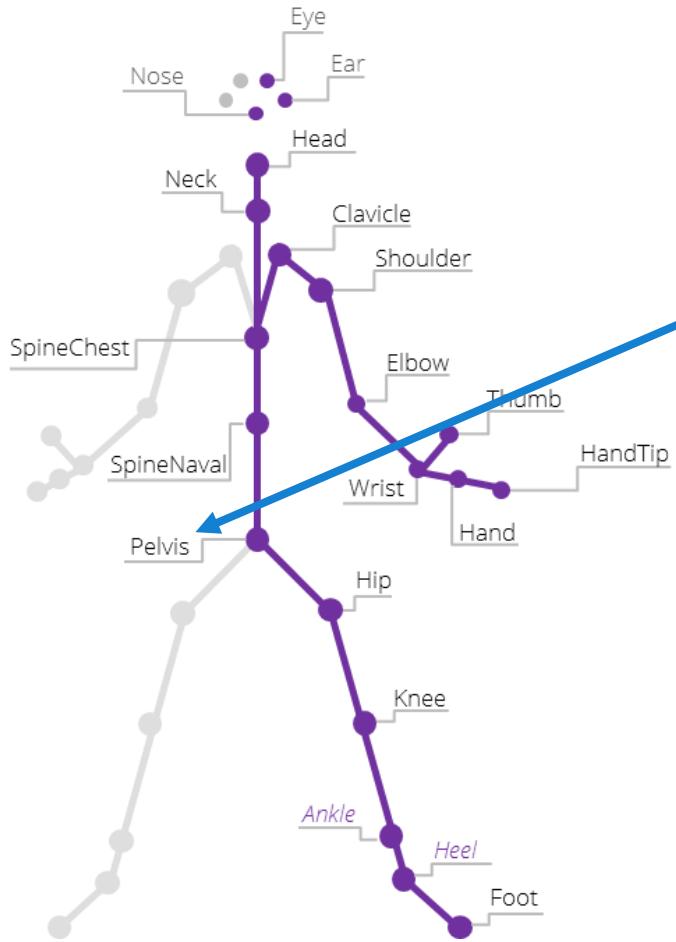
```
body_tracking > python > body_tracking.py > ...
24 """
25 import cv2
26 import sys
27 import pyzed.sl as sl
28 import ogl_viewer.viewer as gl
29 import cv_viewer.tracking_viewer as cv_viewer
30 import numpy as np
31
32 if __name__ == "__main__":
33     print("Running Body Tracking sample ... Press 'q' to quit")
34
35     # Create a Camera object
36     zed = sl.Camera()
37
38     # Create a InitParameters object and set configuration parameters
39     init_params = sl.InitParameters()
40     init_params.camera_resolution = sl.RESOLUTION.HD1080 # Use HD1080 video mode
41     init_params.coordinate_units = sl.UNIT.METER           # Set coordinate units
42     init_params.depth_mode = sl.DEPTH_MODE.ULTRA
43     init_params.coordinate_system = sl.COORDINATE_SYSTEM.RIGHT_HANDED_Y_UP
44
45
46     # Open the camera
47     err = zed.open(init_params)
48     if err != sl.ERROR_CODE.SUCCESS:
49         exit(1)
50
51     # Enable Positional tracking (mandatory for object detection)
52     positional_tracking_parameters = sl.PositionalTrackingParameters()
53     # If the camera is static, uncomment the following line to have better performances and better results
54     # positional_tracking_parameters.set_as_static = True
55     zed.enable_positional_tracking(positional_tracking_parameters)
56
57     obj_param = sl.ObjectDetectionParameters()
58     obj_param.enable_body_fitting = True          # Smooth skeleton move
59     obj_param.enable_tracking = True              # Track people across images flow
60     obj_param.detection_model = sl.DETECTION_MODEL.HUMAN_BODY_FAST
61     obj_param.body_format = sl.BODY_FORMAT.POSE_18 # Choose the BODY_FORMAT you wish to use
```

SDK library

Camera invoke

Coordinate System

Skeleton data format



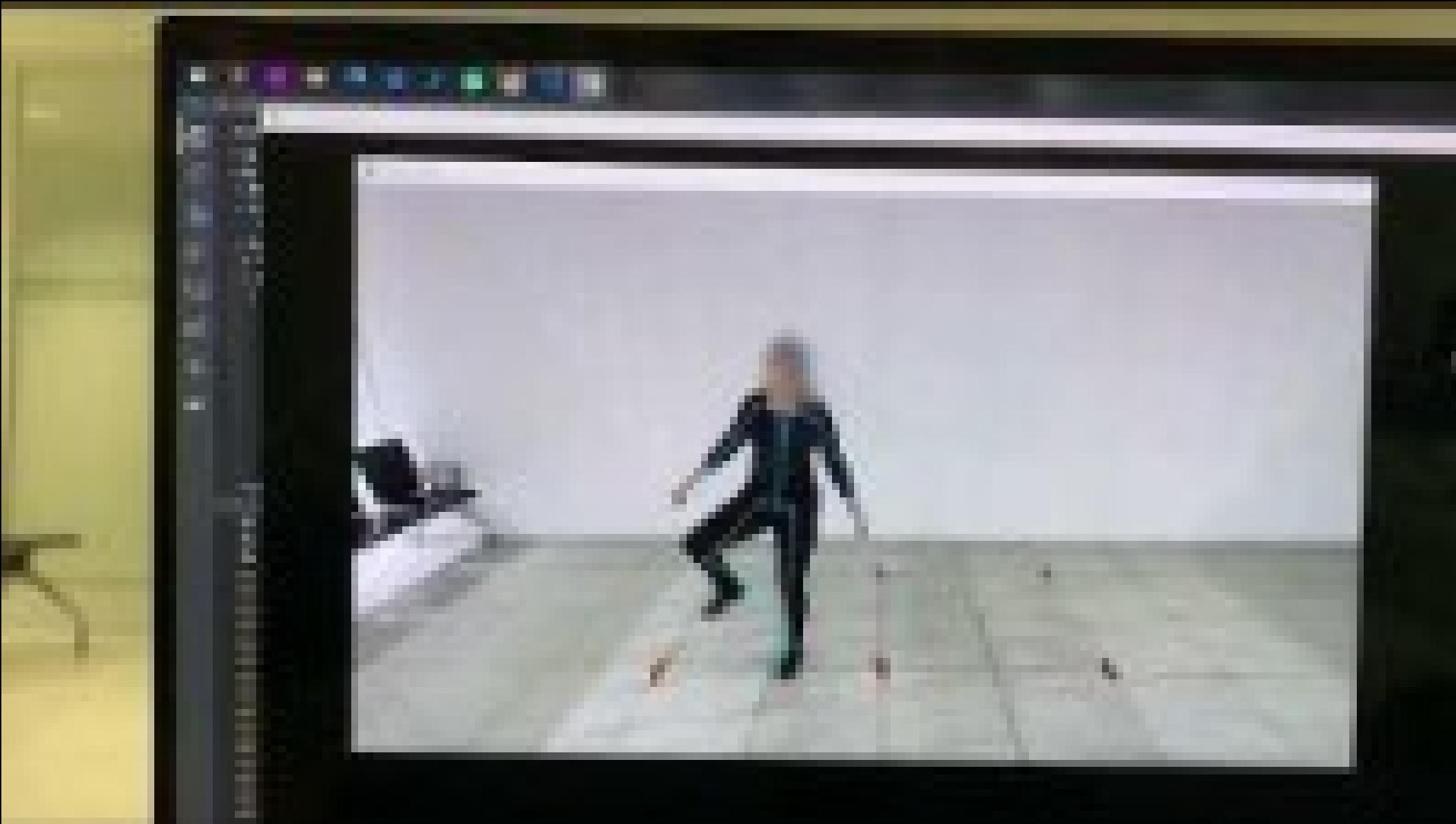
Joint key

Coordinates [x, y, z]

```
# elif body_format == sl.BODY_FORMAT.POSE_34:
    #Get skeleton data joints
    joints={
        'Pelvis': [obj.keypoint[0][0]*-1,obj.keypoint[0][1],obj.keypoint[0][2]],
        'NavalSpine':[obj.keypoint[1][0]*-1,obj.keypoint[1][1],obj.keypoint[1][2]],
        'ChestSpine':[obj.keypoint[2][0]*-1,obj.keypoint[2][1],obj.keypoint[2][2]],
        'Neck':[obj.keypoint[3][0]*-1,obj.keypoint[3][1],obj.keypoint[3][2]],

        'ClavicleLeft':[obj.keypoint[4][0]*-1,obj.keypoint[4][1],obj.keypoint[4][2]],
        'ShoulderLeft':[obj.keypoint[5][0]*-1,obj.keypoint[5][1],obj.keypoint[5][2]],
        'ElbowLeft':[obj.keypoint[6][0]*-1,obj.keypoint[6][1],obj.keypoint[6][2]],
        'WristLeft':[obj.keypoint[7][0]*-1,obj.keypoint[7][1],obj.keypoint[7][2]],
```

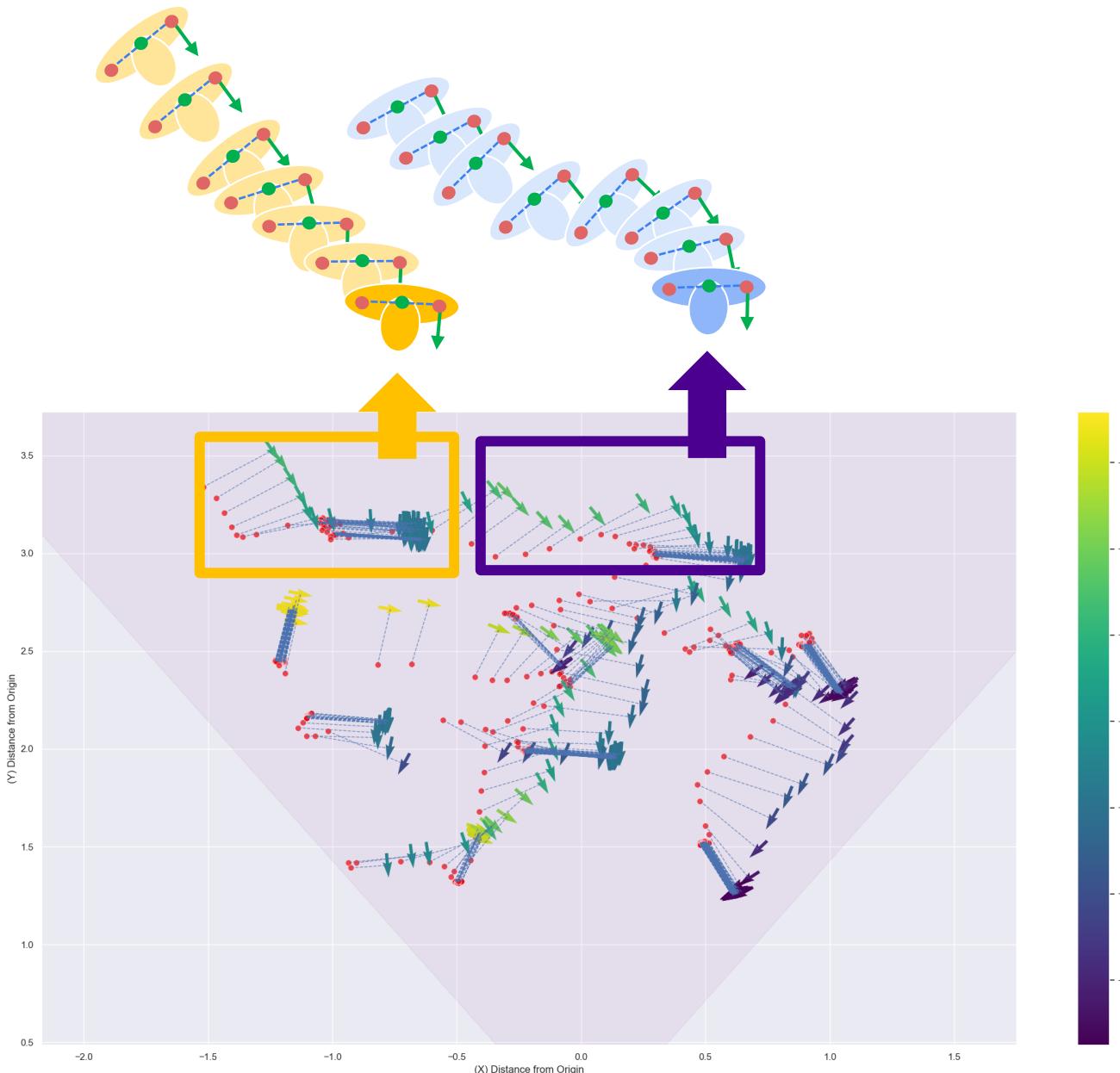
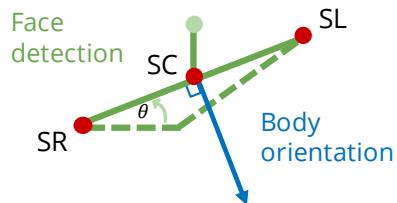
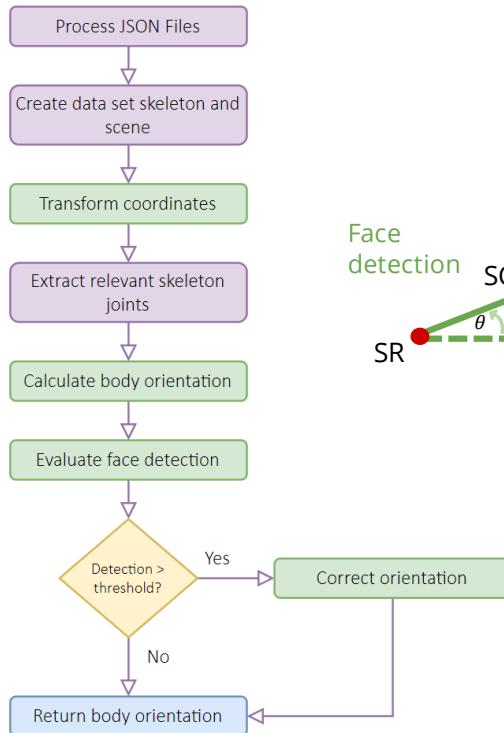
Zed2i: Test

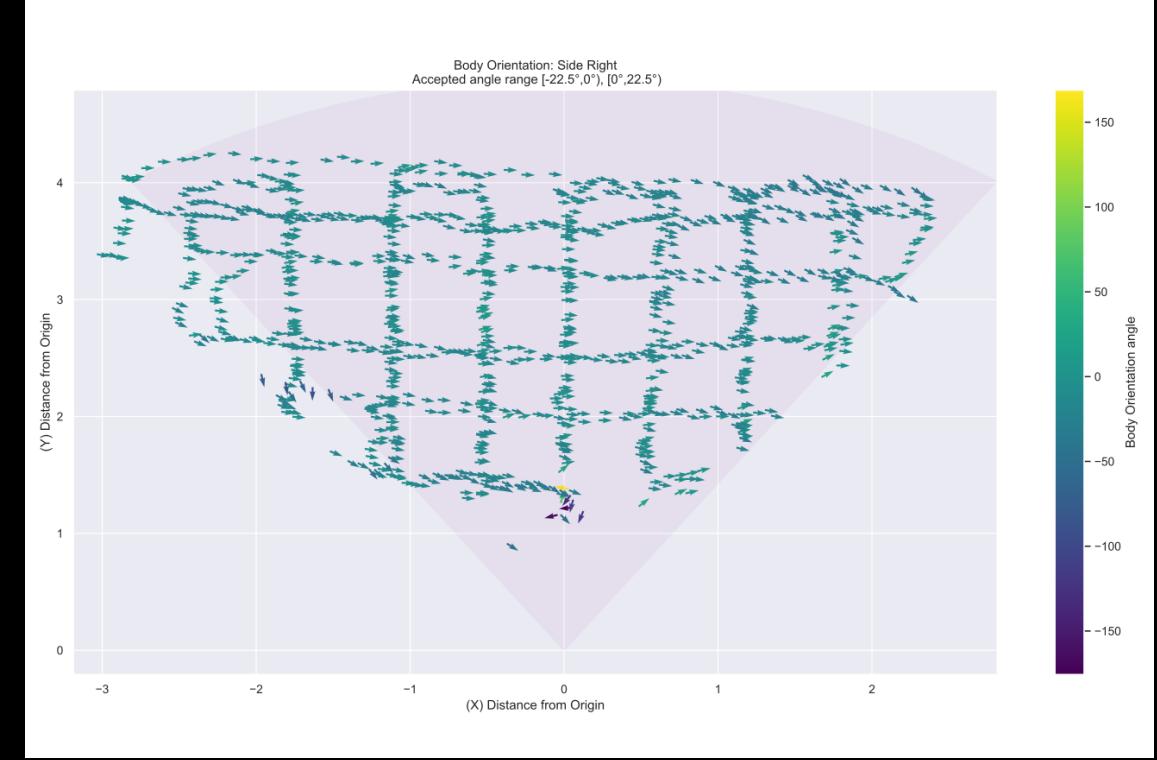
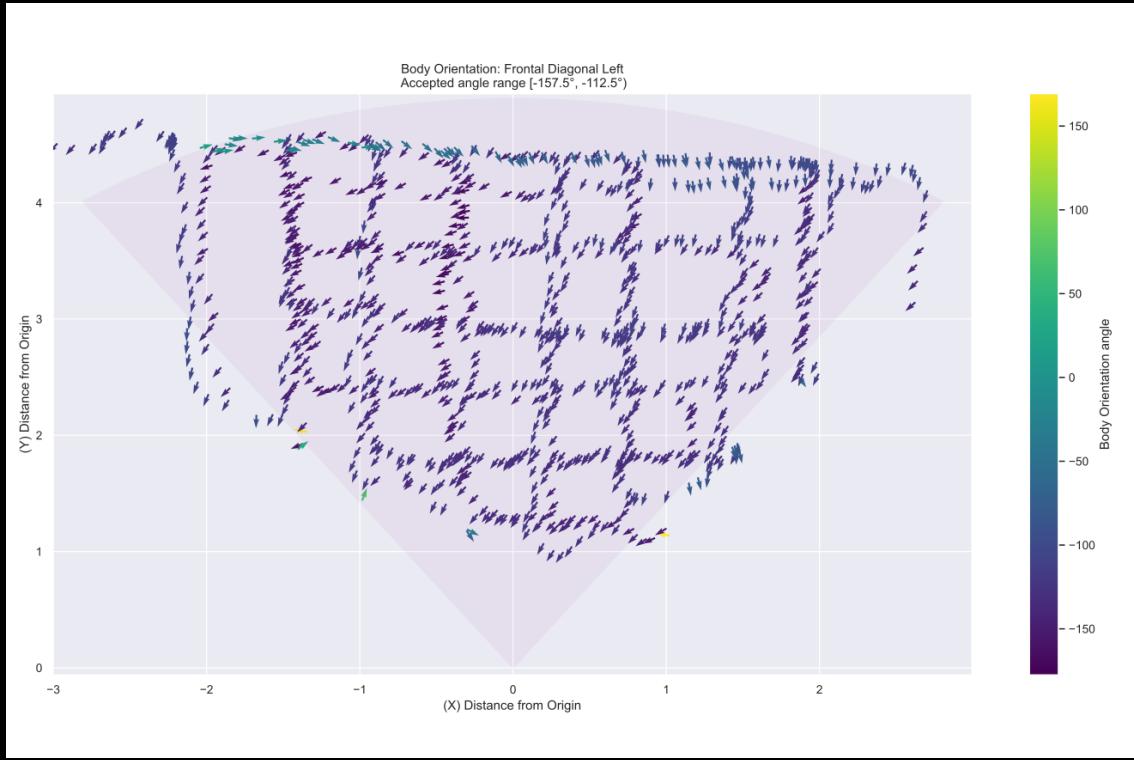


[Zed2i: Skeleton tracking test - YouTube](#)

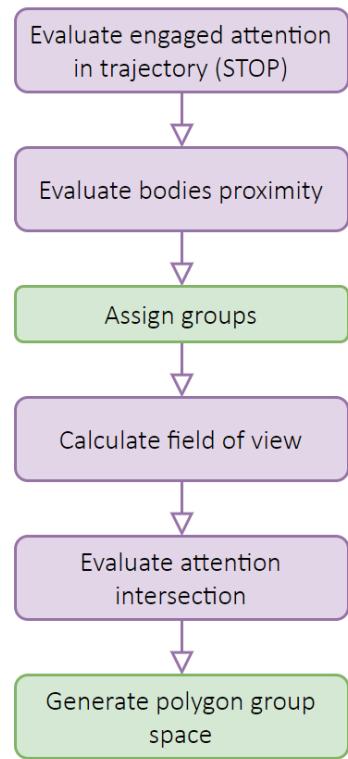
Application:
Detecting social groups

Skeleton data processing algorithm

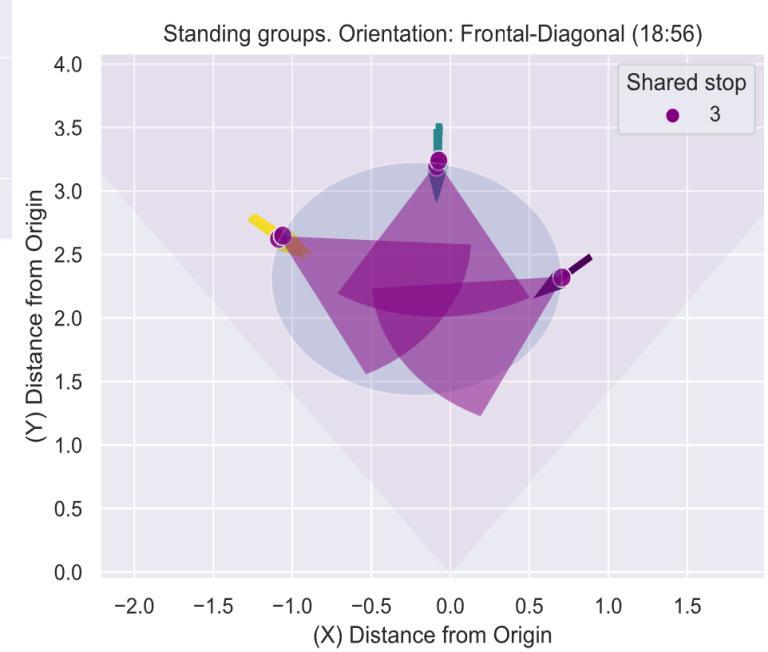
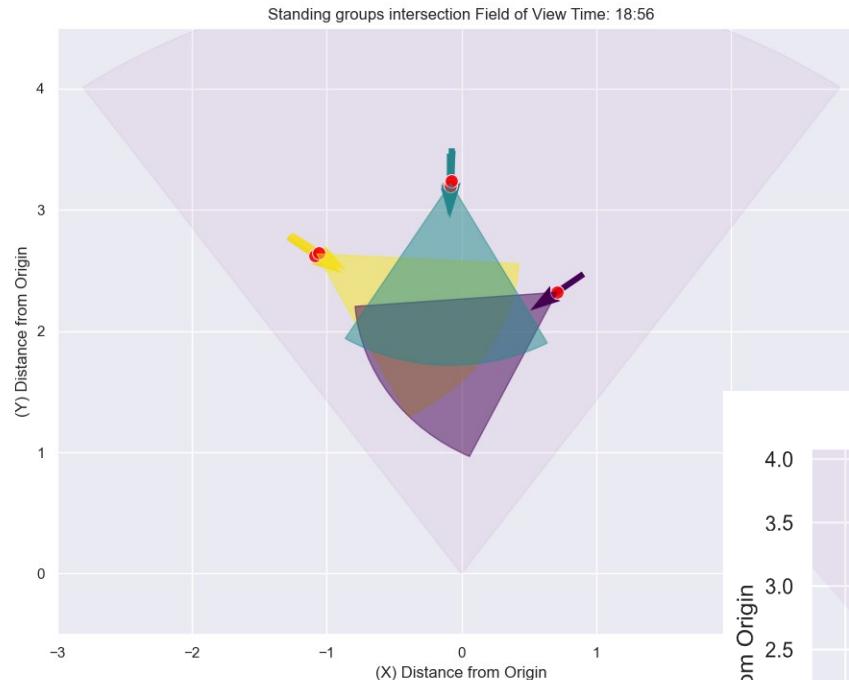




Group detection algorithm



> 90%



Challenges



Social

Cultural context

Privacy concerns

Bias in algorithms (height, weight,
skin color behavior)



Social Technical

Cultural context

Privacy concerns

Bias in algorithms (height, weight,
skin color behavior)

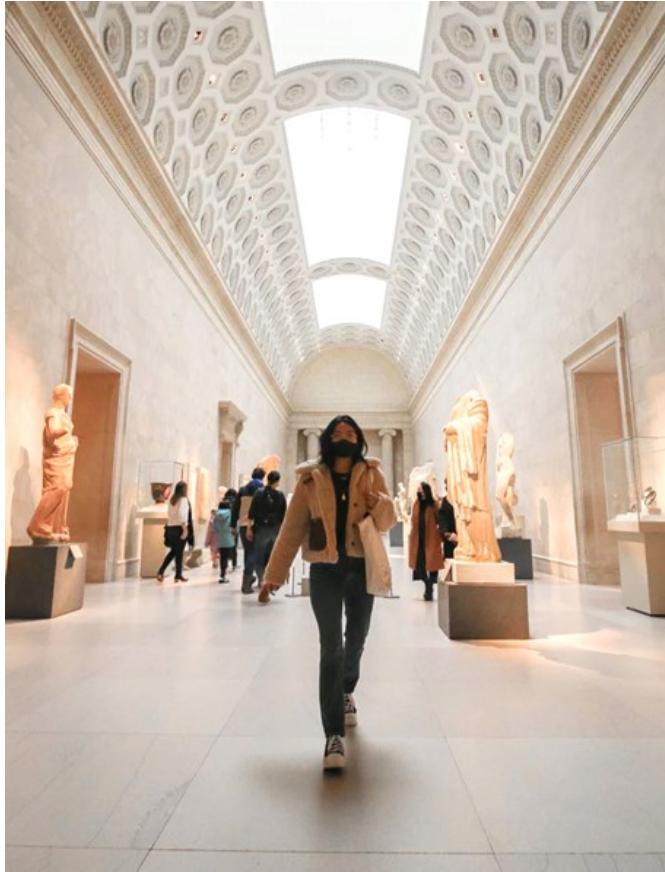
Computational power

Environmental factors

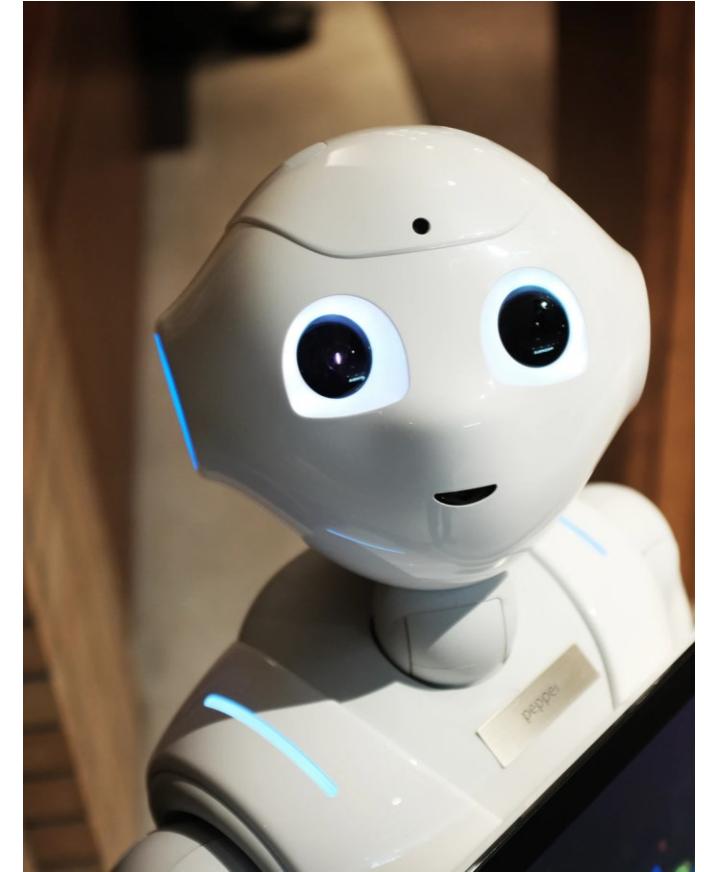
Set-up configuration and placement



Services we use



Places we explore



Artificial agents we create

The Power of tracking the invisible: Geoinformatics and Social Human Behavior

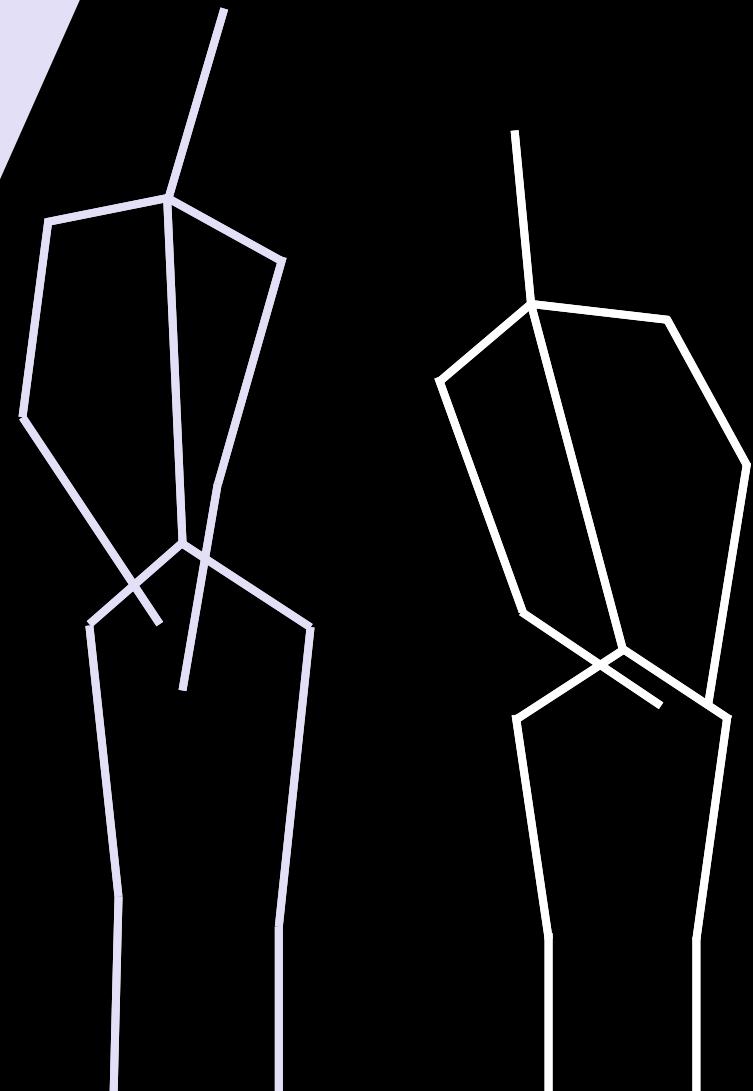
Violeta Ana Luz Sosa León

www.violetasdev.com

@violetasdev



PYCON 22
COLOMBIA



References

- [violetasdev/zed-examples: ZED SDK Example projects \(github.com\)](https://github.com/violetasdev/zed-examples)
- [V. A. L. S. León and A. Schwering, "Detecting socially occupied spaces with depth cameras: evaluating location and body orientation as relevant social features," 2021 International Conference on Indoor Positioning and Indoor Navigation \(IPIN\), 2021, pp. 1-8, doi: 10.1109/IPIN51156.2021.9662607](https://doi.org/10.1109/IPIN51156.2021.9662607)
- [Sosa-Leon, V., Schwering, A., \(2022\). Evaluating Automatic Body Orientation Detection for Indoor Location from Skeleton Tracking Data to Detect Socially Occupied Spaces Using the Kinect v2, Azure Kinect and Zed 2i. Sensors 2022, 22, 3798. https://doi.org/10.3390/s22103798](https://doi.org/10.3390/s22103798)
- [Mathworks Stereo Vision](#)
- [Stereolabs](#)
- [Zed 2i · violetasdev/bodytrackingdepth_course Wiki \(github.com\)](https://github.com/violetasdev/bodytrackingdepth_course)
- [Stereolabs Docs: API Reference, Tutorials, and Integration](#)
- [openpose/doc at master · CMU-Perceptual-Computing-Lab/openpose \(github.com\)](https://github.com/CMU-Perceptual-Computing-Lab/openpose)

Images from Unsplash