Project Report – Age Estimation from Posture

Lotem Nadir

October 26, 2021

1 Goal

The main purpose of this project is to develop algorithms and a pipeline for age estimation from RGB-D images, using the posture of the subject. The project consists of two tasks: (1) posture estimation from RGB-D images and (2) age classification from posture.

2 Dataset

A dataset of 15 recording sessions was collected. In each recording session, a person was recorded in various poses known to reflect age-differences: standing-still; standing on one leg; standing with feet-together stance; squatting consecutively for 30 seconds. Each recording was done using 3 RealSense cameras (RGB-D) and Vicon sensors (3D coordinates) as shown in figure 1. The RealSense cameras record 3 different angles: front, back and side.

Figure 1: Examples of the different RealSense shooting angles, and the Vicon points.

Working with multiple sensors raises the following challenges:

- The frames from the different RealSense cameras and the Vicon system are not aligned in time, and synchronizing is required.

- The RealSense cameras and the Vicon sensors are not calibrated. Calibration is required in order to find the transformation between the Vicon coordinate system and the RealSense camera coordinate system.
- The FPS of the RealSense cameras is 30, and the FPS of the Vicon system is 120. In some recordings, the FPS of the RealSense cameras is 15.
- Since two of the RealSense cameras in each recording session are connected to the same laptop, there is a frame drop in the output of these cameras.

3 Methods

In this section I will be referring to each of the tasks separately.

3.1 Posture estimation from RGB-D images

My main contribution was by cleaning the dataset and finding a method to calibrate RealSense cameras and the Vicon sensors. For the dataset cleaning, automatic method was first examined, using the OpenPose [1] model in order to detect the T-pose at the beginning of each recording. OpenPose performed poorly of non-frontal shooting angles as shown in figure 2. Due to the frame-drop in the RealSense cameras, manual fixes on the "front" and "back" angles were required frequently. Since this method was not effective nor accurate enough, the T-pose was manually detected in all recordings. In order to deal with the different FPS of the sensors, every 4th frame was taken from the Vicon recordings. Another method of averaging every 4 frames in the Vicon recordings was considered. In order to check which method is better, an angle in the neck was calculated in both methods. The difference between the two methods was negligible. In order to deal with the frame-drop in the RealSense cameras, the differences in the frames numbers caused by the frame-drop were extracted, and the correlated Vicon frames were trimmed to fit the RealSense data. This process is shown in figure 3.

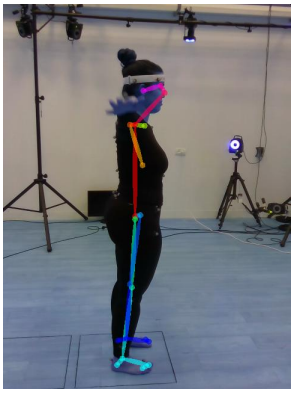

Figure 2: OpenPose perform poorly on non-frontal frames

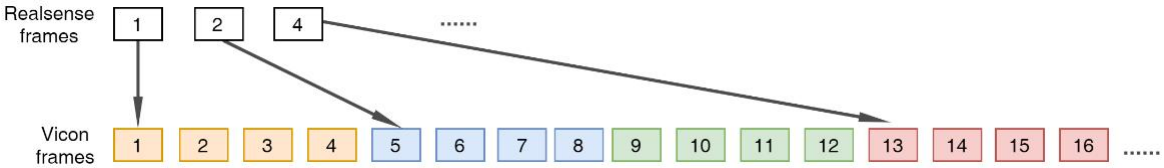

Figure 3: Trimming the Vicon frames according to the RealSense frames. Some frames were removed in the RealSense due to the frame-drop, e.g frame #3 in this example. Their correlated frames in the Vicon, e.g frame #9 in this example, were not taken to the dataset.

Recordings were cleaned and trimmed for all 15 sessions. Validation of the synchronizing was done manually. For the calibration process, Kabsch's algorithm [2] was used in order to find the transformation between the Vicon coordinate system and the RealSense camera coordinate system: given 2 sets of N paired points in D dimensions, Kabsch's algorithm calculates the rotation matrix that minimizes the RMSE between the two sets, using singular value decomposition (SVD). The calibration is done using a single frame from the RealSense camera and the corresponding frame from the Vicon system. Kabsch's algorithm alone performed poorly. Several improvements were made to the data before applying Kabsch's algorithm on it: Removing points with noisy depth value, averaging the depth value of each point with neighboring pixels, sampling sub-group of the points with lowest projection error. After applying the improvements, the current projection error rate is 60mm (RMSE). The projection before and after the improvements is shown in figure 4.

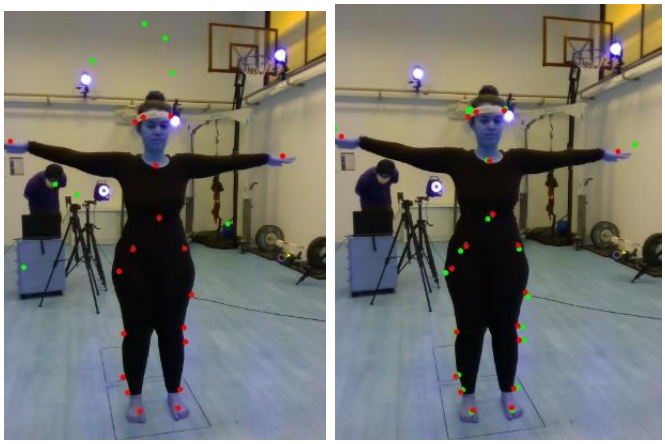

Figure 4: Projecting the Vicon points, after applying on them the transformation calculated with Kabsch's algorithm without improvements (Left) and with improvements (Right). The red points are "ground truth", the green points are the projected points.

3.2 Age classification from posture

Two methods were examined for this purpose: (1) Predicting the age directly from the 39 3d points, using the network of PointNet [3], and (2) Predicting the age from four angles calculated based on the 3d Vicon points, using classical classifiers. In both method the ages of the subject were converted into binary labels.

For predicting the age directly from the 39 3d points, PointNet was trained as a binary classifier ("old" or "young"). The network was trained "as is", except for changing the dimensions of the last softmax layer. The training resulted in high overfitting on the trainset, as can be shown in figure 5. This might be caused by the fact that the data has low variance due to the Vicon high FPS. In order to increase the variance in the data, the dataset was re-generated, this time only frames that have a difference of at least 80mm were kept to the dataset. This process is shown in figure 6.

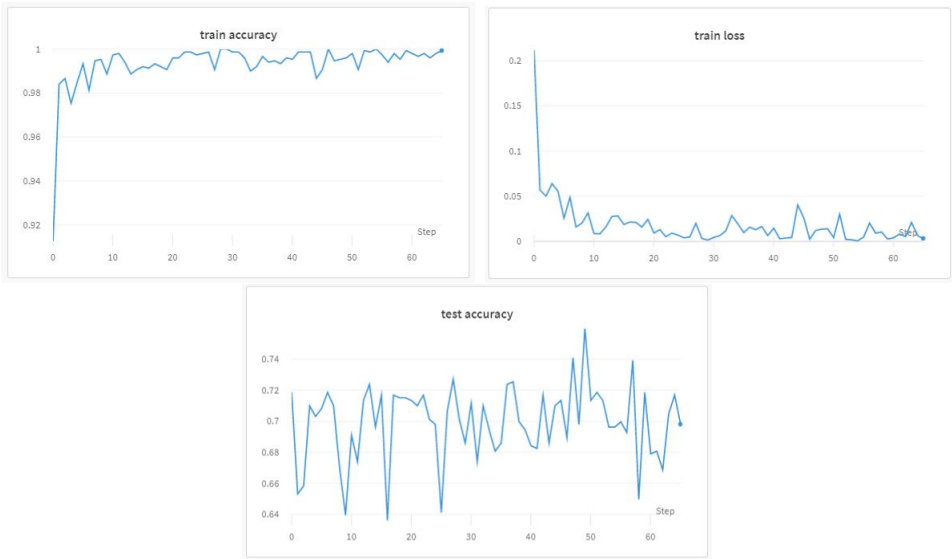

Figure 5: PointNet training results. The network has learned well to classify the train data, but failed to generalize it's learning on the test set.

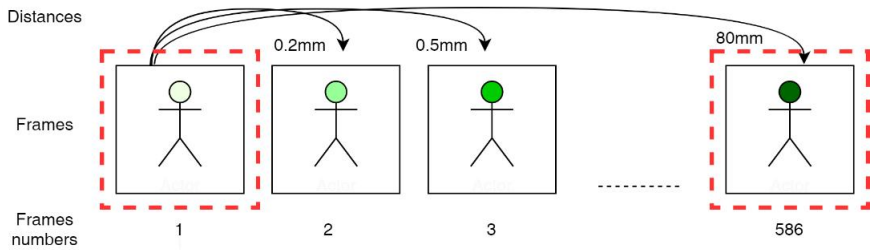

Figure 6: For each recording, the first frame was taken to the dataset. For each consecutive frame, its average euclidean distance from the first frame was calculated. If that distance is equal or larger than 80mm, the frame was taken to the dataset as well, and so on.

For predicting the age from the angles, four angles represent the human posture were chosen. Each one was defined by 3 3D points from the Vicon data. The angles are described in appendix 5.1. For each frame in the dataset, the four angles were extracted, and the age was converted into a binary label ("old" or "young") for that sample. Dimensionality reduction algorithms were applied on the data in order to visualize it, as shown in figure 7.

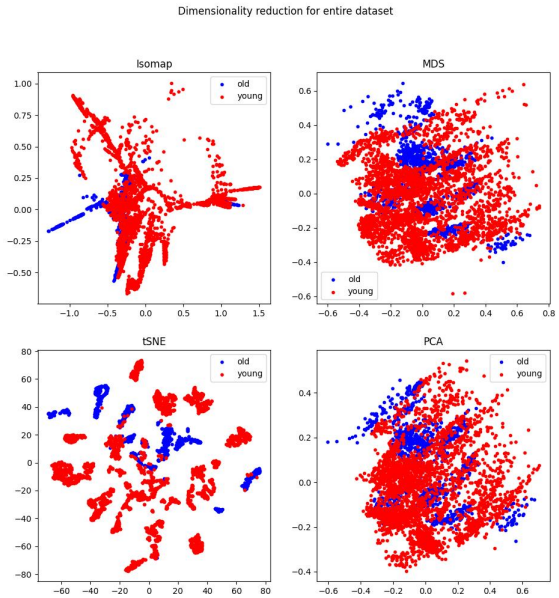

Figure 7: Data after applying several dimensionality reduction algorithms

As can be seen from the figure, the data is not separable. Nonetheless, several classical classifiers (KNN, SVM, random forest) were trained on the dataset. The result were poor, as shown in appendix 5.2. Results were also highly depended on the data splitting to train and test sets. This indicates more data is required for this problem.

4 Future Work

My work has shown that more data is required to solve this problem. Collecting data in the lab using the RealSense and the Vicon sensors is the most accurate method, but we will not be able to achieve thousands of samples this way. Perhaps a new paradigm is required, such as using 3d human body datasets available online. These datasets are frequently given in meshes, but their vertices can be seen as human body point clouds. Perhaps, developing a method for estimating the Vicon points from the mesh's vertices might be useful. A list of such datasets is described in appendix 5.3. For the data collection in the lab, it is necessary to develop automatic method for synchronizing the recordings in time. Such method can be to develop automatic T-pose detector or to start the recordings with constant time gaps. Moreover, the calibration must be calculated using a constant object. For that purpose, a special calibration device was built, as shown in figure 8.

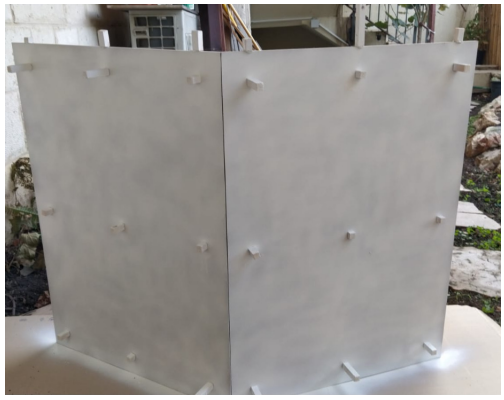

Figure 8: Calibration device

5 Appendix

5.1 List of angles used for age classification

The angles were calculated with the following points:

1. (C7, STRN, T10)
2. (RSHO, C7, LSHO)
3. (CLAV, C7, middle of RFHD and LFHD)
4. (STRN, C7, middle of RASI and LASI)

5.2 Classical classifiers results

Classifier	Best Parameters	Train Accuracy	Test Accuracy
KNN	n_neighbors: 5	0.821	0.745
SVM	C: 10, kernel: poly	0.829	0.614
Random Forest	max_depth: 50, n_estimators: 50	0.835	0.684

Table 1: Classical classifiers results on the angles data. Results were highly depended on the data splitting to train and test sets.

5.3 3D human body datasets

1. 3DPeople: 80 different subjects, mostly young, RGB-D, 3D skeleton.
2. ScanDB: 114 different subjects, meshes.
3. People-Snapshot: 24 different young subjects, meshes.

4. Human3.6M: 11 different young subjects, meshes.
5. Buff: 6 different young subjects, meshes.
6. MPII: 4300 different people, old & young, meshes.
7. USCS: 3000 different people, old & young, meshes. Might be overlapping with MPII.

5.4 Code

My code is maintained in GitHub in the following location:
https://github.com/Lotemn102/skeleton-RGBDto3D
Documentation comments were added throughout the entire code during my work. E-mails and meeting summaries were added as well to the repository.

References

[1] OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, Cao et al., 2018

[2] A solution for the best rotation to relate two sets of vectors, W. Kabsch, 1972

[3] PointNet: Deep Learning on Point Sets for 3D Classification and Segmentationl, Charles R. Qi et al., 2016.