3D Human Pose Estimation (3DHPE)

Phase 1 Report

Farhanur Rahim Ansari, Vidhey Oza, Minji Lee

Abstract

The project aims to perform 3D human pose estimation on RGB images and videos and make an interactive tool that helps in using this technology with convenience. Building pipelines and prototypes are imperative to go further and achieve our goal: to build interactive tools (to be explored during Phase 2). Existing research and pre-trained models for 3D-HPE have their strong points and weaknesses. We scrutinized models and found the most promising model combination: OpenPose and GAST-Net. After OpenPose receives input image or video and produces 2D output, GAST-Net receives the 2D input and returns the 3D output.

Summary

Human Pose Estimation is an important problem and has enjoyed the attention of the Computer Vision community for the past few decades. It is an essential step towards understanding people in images and videos. Given an image of a person, 3D pose estimation is the task of mapping the spatial position of a person into a simulated 3D space.

Human pose estimation is a well-solved problem in real-world scenarios for the 2D case. However, recovering 3D pose from 2D RGB images is considered more difficult than 2D pose estimation due to the larger 3D pose space and more ambiguities. An algorithm has to be invariant to factors like background scenes, lighting, clothing, skin color, and image imperfections, among others. 3D-HPE has immediate applications in various tasks such as action understanding, surveillance, human-robot interaction, motion capture, and CGI. We can enjoy the fruits of 3D-HPE while sitting on a comfortable couch and watching Disney movies, or we can actively engage with 3D-HPE technology by playing against a virtual tennis player on Nintendo Wii or Xbox Kinect.

The main goals of Phase 1 were to look into as many models as possible, make prototypes, and design robust pipelines for the project. 3D-HPE research is well tested in the indoor and lab settings, but not too much on in the wild dataset. Our exploration, which will be detailed out in the following sections, provided us insights into which direction we should march on.

As previously mentioned in our proposal and progress pitch, our final goal is to make an interactive tool that can be used to perform the task of 3D-HPE easily by anyone for fun and make improvements on the quality of its output. The crucial part of our goal is to go from proof-of-concept to real-world (in-the-wild) images and videos. This project will not only automate the tasks being done manually previously but also will help us gather interesting insights from the extracted information that was previously not possible.

Datasets

It is quite challenging to build an in-the-wild dataset. A 3D pose estimation dataset is built using Motion Capture (MOCAP) systems, which are suitable for an indoor environment. MOCAP systems require an elaborate setup with multiple sensors and bodysuits, which is impractical to use outside. So the lack of ground truth data is a significant bottleneck. In order to achieve our targeted goal, we have divided our overall approach into two parts. The details about this are mentioned in the next section. Each part utilizes a different dataset combination in order to train its model.

I. Input-to-2D

For training our model to produce 2D pose predictions from the input video, we have worked on the COCO, and MPII Human Pose dataset. These two datasets collect images in diverse scenarios that contain many real-world challenges, such as crowding, scale variation, occlusion, and contact.

COCO is a publicly available large scale object detection, segmentation, and captioning dataset with 1.5 million object instances from 80 objects. The annotations here are in the JSON file format, which has a dictionary as a top value. Along with the images, the metadata is also present.

MPII Human Pose dataset is a state of the art benchmark for evaluation of articulated human pose estimation. It includes around 25K images containing over 40K people with annotated body joints covering 410 human activities. It has images extracted from youtube, making it a truly in-the-wild dataset. The annotations here are also extracted in a JSON file format.

II. 2D-to-3D

For transforming the estimated 2D pose into 3D, the models were trained and tested on two publicly available datasets: Human3.6M and HumanEva-I. They have multiple subjects performing common poses in an indoor environment. These are standard datasets for 3D Human Pose Estimation.

Human3.6M captures data through four synchronized cameras at 50 Hz and contains 3.6 million video frames with 11 professional subjects performing 15 daily activities (i.e., walking and sitting). Subjects 1, 5, 6, 7, 8 were employed for training and subjects 9, 11 for testing for the GAST-Net model (discussed in the preceding section).

HumanEva-I is a much smaller dataset and captures data through three camera views at 60 Hz. The time-series data from three actions (walk, jog, box) is split between training and testing. The dataset has been downloaded from the official website as 2D, and 3D ground-truth poses on 15-joint skeleton objects in a .npz file format. It had MATLAB dependencies that needed tweaking the ConvertHumanEva.m file provided on Facebook’s VideoPose3D Github repository [6] according to the system requirements.

Fig1: Sample images from the Human3.6M dataset

Methods

There are two main approaches for estimating 3D poses from monocular RGB images. Given an input video, the one-step method directly estimates 3D poses from RGB images in an end-to-end fashion. Second is a two-step method that first predicts the 2D key points from RGB images and then lifts them to 3D poses. Out of the two, the two-step method is more advantageous than the one-step method. It is compatible with the existing 2D pose estimation algorithms and so can be easily built upon them. Most importantly, it avoids the influence of the environment backgrounds and human surface features, which makes it a better generalization for in the wild scenarios. Also, the models have an auxiliary point of convergence, so it is mathematically more feasible to reach the optimal weights. Overall it is a better alternative for us to achieve our goal.

The 3D pose estimation problem, in general, comes with certain challenges, which makes it difficult for them to be implemented in the wild scenarios. One of the biggest challenges is that of self-occlusion. As humans are articulated objects, self-occlusion will inevitably occur in the input videos. And the other general challenge is to handle visual jitter, which yields because of the motion estimated from monocular RGB images. Beside these two general challenges, there is also the challenge of depth ambiguity, which is a bit specific to the two-step process. For a monocular model, this challenge arises because of the multiple possible 3D poses for a given 2D pose since there is no information about depth in the video or the 2D pose.

Our end goal for this project is to develop a 3D pose estimation engine that can be generalized for in the wild cases and which can handle the challenges of the domain introduced in the previous paragraph with ease. In order to achieve our goal, we have made use of and developed our project on top of 2 of the benchmark works in the field of human pose estimation - OpenPose and GAST-Net. We have developed a pipeline where the input video data is first passed to the OpenPose Framework and gets transformed into an estimated 2D pose video along with the JSON file containing the information of the human skeleton key points as output. This output is then passed to the GAST-Net Framework, which further augments it and transforms it into an estimated 3D pose as the final output.

I. Input-to-2D Pose Estimation – OpenPose

The OpenPose model developed by the CMU Perceptual Lab (initially proposed in [5] and improved in [3]) is considered as the state of the art approach for real-time 2D human pose estimation of multiple people in an image. The model has been trained and tested on the COCO and MPII datasets. Here, the image is analyzed using pre-trained convolutional networks like VGG-19, mobileNet, and ResNet for producing feature maps. It uses a non-parametric representation, referred to as Part Affinity Fields (PAFs), a set of 2D vector fields that encode the location and orientation of limbs over the image domain, to learn the associated body parts with individuals in an image. This approach performs exceptionally well and achieves high accuracy and real-time performance, regardless of the number of people in the frame.

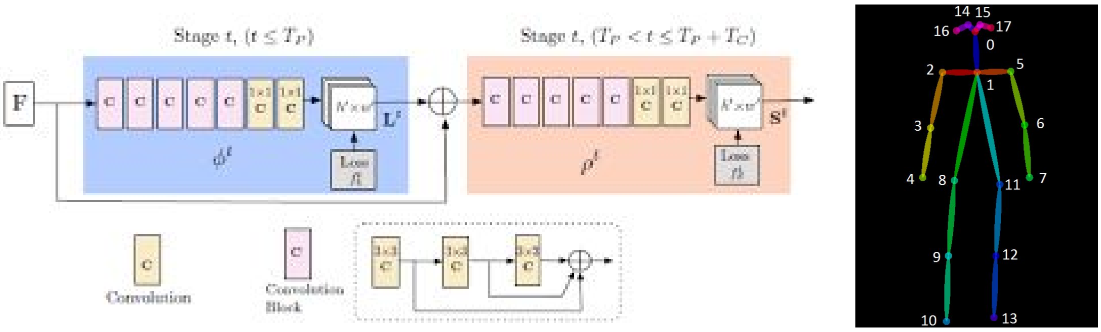


Fig 2: OpenPose - (a) Architecture for the Multi-stage CNN (b) 18-Keypoints skeleton for COCO Dataset .

OpenPose has models trained for many different skeleton formats, so we decided to develop a working prototype with their codebase and then build on top of it. Currently, we have used the COCO dataset format of 18 keypoints (skeleton shown in Fig. 2b) since it is the most used dataset across the pipeline. OpenPose stores the pose estimation output of a video in frame-wise JSON files, so we had to develop a way to merge them all into a single format that is easily accessible by Stage II of our pipeline.

2D keypoint predictions are evaluated using the mAP metric, which is typically used for measuring the accuracy of object detectors. It stands for mean Average Precision and computes the average precision value for recall. On the MPII dataset, OpenPose outperforms the previous state of the art bottom-up methods by 8.5% mAP. Comparing the results on the COCO dataset, OpenPose has the highest drop in accuracy when considering only people of higher scales. And it has an impressive overall mAP score of 64.2 in this case,

II. 2D-to-3D Pose Estimation – GAST-Net

GAST-Net is a benchmark approach for solving the problem of 2D-to-3D human pose estimation [4]. It exploits the information about the space occupied by the object in the frame and the time of the video to solve challenges associated with this domain. Its main idea is that temporal information is just as valuable as spatial to resolving occlusion and depth ambiguity in 3D pose estimation. It proposes simple yet effective graph attention Spatio-temporal convolutional network that comprises of interleaved temporal convolutional and graph attention blocks.

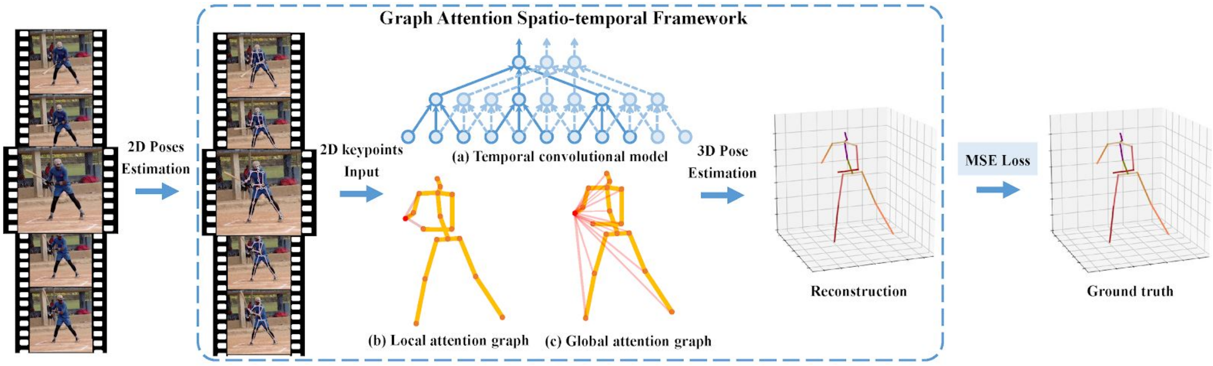
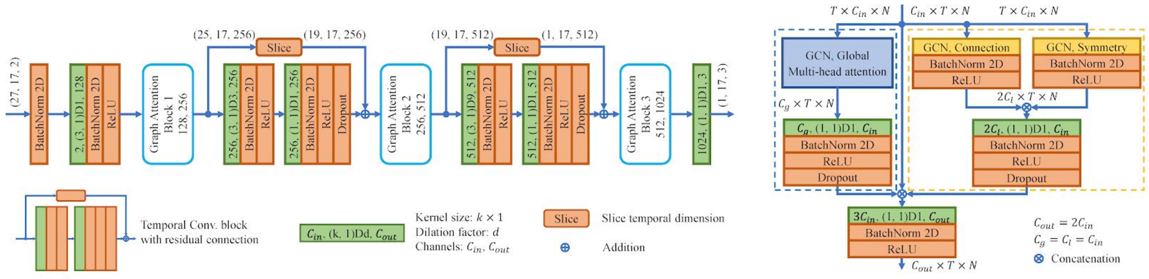

Graph Attention Spatio-temporal Framework
2D Poses Estimation
2D keypoints Input
(a) Temporal convolutional model
3D Pose Estimation
MSE Loss
Reconstruction
Ground truth
(b) Local attention graph
(c) Global attention graph

Fig 3: Schematic overview of the GAST-Net Framework. Input: 2D pose estimates from RGB images. Output: a sequence of reconstructed 3D poses from the corresponding 2D keypoints. GAST-Net synergistically interleaves 3 components: (a) a dilated temporal convolutional model (model forward-propagation from bottom to top) with (b) a set of local attention mechanisms for visualized joints (i.e., the right-wrist) including local

kinematic dependencies and symmetric relations, and (c) a global attention mechanism that informs about posture semantics.

The temporal component is designed from dilated TCNs to tackle long-term patterns. As for the spatial components, the local spatial attention network models the hierarchical and symmetrical structure of the human skeleton and a global spatial attention network to adaptively extract global semantic information to encode the human body's spatial characteristics better.



(a) Graph attention spatio-temporal convolutional networks

(b) Graph attention block

Fig. 4: (a) An instantiation of GAST-Net for 3D pose estimation. The GAST-Net consists of 2 Temporal Convolution Blocks and 3 Graph Attention Blocks. Given a 2D pose sequence, the output is a sample 1-frame prediction. Dimensions are enclosed in parenthesis: e.g. (27, 17, 2) denotes a receptive field of 27 frames, 17 joints, and 2 channels. (b) The graph attention block architecture. The left dotted box indicates the local graph attention layer. The right dotted box indicates the global graph attention layer. The layer output is concatenated followed by a 2D convolution layer before outputting the spatio-temporal features

Using one of the supported formats of GAST-Net, we arrived at the right way to connect the two modules and make a proper pipeline from the beginning, i.e., any video recorded or taken from the internet, till the making of the skeleton in simulated 3D space. With this pipeline in place, we now have a working prototype of a monocular 3D pose estimation system.

GAST-Net uses a rigid alignment (Procrustes analysis) with the ground truth before calculating the mean per joint positioning error (P-MPJPE), where the absolute distance between predicted and ground-truth joints is measured in mm. This is one of the more common methods to evaluate 2D as well as 3D pose estimation models.

GAST-Net is a very generalized model with a low P-MPJPE score, and it is apparent in the results they have shown in their paper [4]. For the Human3.6M dataset, they show scores very close to the best models in their respective actions, which leads to one of the best performances overall. They boast an average P-MPJPE score of 35.2 while the state-of-the-art is at 34.5. This is noteworthy since the state-of-the-art model is best at only half of all the actions of the dataset, while GAST-Net is close to the best in all actions.

For the HumanEva-I dataset, GAST-Net is state-of-the-art in all the three actions for all the three subjects. They boast an average P-MPJPE score of 21.32, which is the best across all models. With these results in HumanEva-I GAST-Net proves to be the best-generalized model, and that is key in being able to replicate the results in the wild since videos are highly diverse for YouTube videos or self-recordings.

GAST-Net has a very interesting codebase where they only provide the 2D to 3D projection module, and has no code for 2D pose estimation at all. With this in place, we used our modified OpenPose output to get the 2D pose estimation, then bridged the gap between the two modules by developing a mapping system between formats of those modules. OpenPose had a COCO-inspired skeleton format, but it was converted so that it is of the format of the H3.6M skeleton.

Results

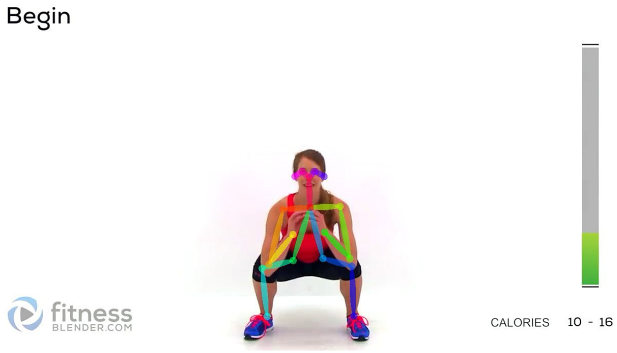


Fig. 5: OpenPose output example.

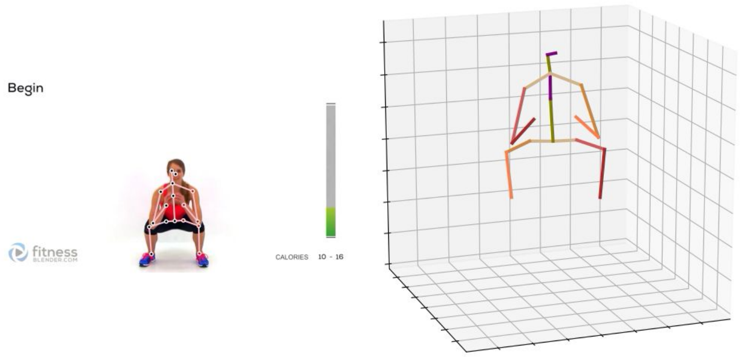


Fig. 6: Left – still image of a person squatting. Right – a 3D projection of her body in simulated 3D space. Notice the elbow and knee angles detected from an angle that is not helpful for the model to detect the full pose.

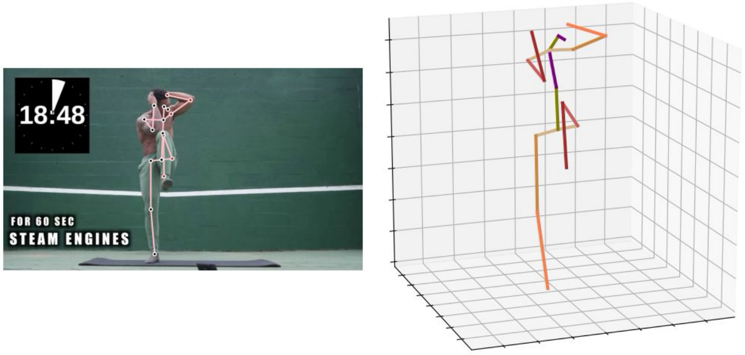


Fig. 7: Left – snapshot from a YouTube workout video. Right – 3D projection. Notice the highly convoluted posture of the person and the model's efficacy to model it properly in 3D.

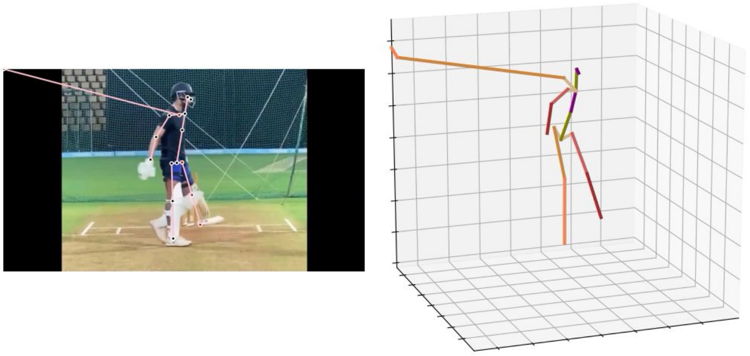


Fig. 8: The problem of self-occlusion: the left hand of the batsman is not visible, and since the 2D pose estimation module does not detect and track it correctly, the error is propagated to the 3D projection module as well.

Figs. 6-8 showcase our 3D pose estimation pipeline in action on various indoors and outdoors scenarios. Fig. 7 shows how the pipeline handles contorted bodies well without disturbing the 3D projection of the same. It is worth noting that the model only has information in 2D, so the model predicts the depth dimension all by itself by understanding how humans move and how our body parts work in conjunction with one another.

With this pipeline prototype developed, we want to work on and tackle problems that monocular pose estimation brings to the table. Specifically, we want to focus on self-occlusion and flickering/jittering. In Fig. 8, the problem of self-occlusion can be seen clearly: the part that is not visible on camera is difficult to predict. While building the pipeline, we have seen that the Stage I of 2D pose estimation is a big contributor to this issue, which propagates into the Stage II module as well. Even though GAST-Net claims to solve the problem by a large margin, it is observed that more work needs to be done on that front.

Flickering is a problem intrinsic to any detection problem in video-based computer vision, but with pose estimation, it is not only more severe of a problem but also more pronounced. We believe that working with better quality datasets and fine-tuning the entire pipeline will be very helpful in alleviating this issue significantly, if not entirely fixed.

Discussion

During Phase One, we aimed to look into pre-existing models and make prototypes. We tried OpenPose, AlphaPose, and HRNet for 2D and VideoPose3D, GAST-Net, and Integral Human Pose for 3D. Reading papers and examining how models work brought us insights on the trend of Human Pose Estimation and all the nitty-gritty parts of them.

While struggling to understand models with a lack of experience in Computer Vision, we learned valuable lessons. We got to understand the necessity of a proper multi-stage pipeline, find a way how to use a different dataset to fine-tune models which trained on other inaccessible datasets (Human 3.6M), and learn modular testing of 2D models and 2D-to-3D models.

There are a few roadblocks in the journey that could have been handled differently. Cloud platforms such as Discovery and Google Cloud Platform turned out to spend more time than we expected. We decided to invest in Google Colab fully, but the time we spent could have been used in more dataset gathering and model testing. This would have led to more rigorous model testing before deploying in the pipeline.

Overall, we met our Phase 1 goal. Prototypes we built are working not only on the datasets but also on any videos we feed. In the process, we learned that we have to fine-tune 2D models with a robust dataset. We learned that 3D output needs to be cleaned for future use further down the pipeline as well. These successes and lessons brought us to a better starting line of Phase 2.

Statement of Contributions

- Farhanur Rahim Ansari – Dataset exploration, Pipeline design
- Vidhey Oza – Model exploration, Pipeline design
- Minji Lee – Platform exploration, Model exploration

Github Link

https://github.com/mjlee2121/3D-HumanPoseEstimation

References

1. *Raj, B. (2019, May 01). An Overview of Human Pose Estimation with Deep Learning. Retrieved October 01, 2020, from Medium.*
2. *Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., & Theobalt, C. (2017). Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision. 2017 International Conference on 3D Vision (3DV), 506-516.*
3. *Hidalgo, G., Raaj, Y., Idrees, H., Xiang, D., Joo, H., Šimon, T., & Sheikh, Y. (2019). Single-Network Whole-Body Pose Estimation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 6981-6990.*
4. *Liu, J., Guang, Y., & Rojas, J. (2020). GAST-Net: Graph Attention Spatio-temporal Convolutional Networks for 3D Human Pose Estimation in Video. arXiv preprint arXiv:2003.14179.*
5. *Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei and Y. A. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2019.2929257.*
6. *Pavllo, D., Feichtenhofer, C., Grangier, D., & Auli, M. (2019). 3d human pose estimation in video with temporal convolutions and semi-supervised training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7753-7762).*