Temporally-aware Human Pose Estimation on 3D videos: A review of the State of the Art

Joris LIMONIER[1][0000−0002−0393−2247], Frédéric PRECIOSO[2][0000−0001−8712−1443], and Lucile SASSATELLI[2][0000−0003−1232−1787]

[1] Université Côte d'Azur, Biot, France joris.limonier@etu.univ-cotedazur.fr
[2] {frederic.precioso, lucile.sassatelli}@univ-cotedazur.fr

Abstract. The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: Computer Vision · Human Pose Estimation · 3D Human Pose Estimation · Human Pose Estimation in Videos

1 Introduction

The Deep Learning revolution, coupled with increasing computing power and the improved use of GPU opened new opportunities in the field of Computer Vision. New architectures arose and new techniques suggested numbers of parameters that hadn't been seen before, some of them reaching hundreds of millions of parameters [4]: up to 94.9M for Faster R-CNN [5], 127.3M for Cascade R-CNN [6], 51.0M for FCOS [7], 210.1M for CenterNet [8], 135.2M for Cascade Mask R-CNN [6], 138.2M for Hybrid Task Cascade [9] and 63.4M for Mask R-CNN [5]. Such complex networks manage to segment images, detect objects in images or identify the pose of a person. Our interest goes to the latter task. The task of Human Pose Estimation (HPE) aims at detecting joints of a human being in a frame. This could be considered a solved problem in the 2D case when the person is clearly visible. Some other cases are more challenging, one of which is when some body parts are hidden (occlusions) in a 2D image. Another challenging case is inferring 3D coordinates from a 2D image, this will be our focus throughout this study.

Furthermore, one can consider HPE applied to images but also HPE applied to videos. We want to focus on videos because we are interested in exploiting the temporal dimension rather than a frame-per-frame joints detection. Doing so brings its set of challenges and its complexity to the problem, which makes it even more interesting.

We want to study the current state of the art for 3D HPE on videos while considering the temporal dimension. In order to do so, we will list and examine related work, then we will gather existing datasets and metrics while analysing their strengths and weaknesses. Subsequently, we will evaluate and compare existing methods on common datasets. Finally, we will give our conclusions and propose study pathways for the future of this study.

2 Related Work

HPE encompasses several subtasks, not all of which we are interested in. In this section, we describe the tasks we want to tackle, then we describe existing approaches.

2.1 Task Description

As mentioned in section 1, HPE can aim to obtain 2D (planar coordinates), as well as in 3D (spatial coordinates), our focus is on the 3D case. This means that we want to predict the 3D location of body joints. Several industries find applications of these techniques, namely the movie & animation industry and the sport industry, only to name a few.

When talking about images that have been taken from one viewpoint only, we talk about *monocular* images. Such images result in a projection from the 3D space we live in to the 2D space that is the image. As such, each point in the image is the projection of one of an infinite number of points (namely, the whole straight line starting at the camera and going infinitely in the direction of the projected point). For this reason, the 3D HPE monocular problem is ill-posed.

Although ill-posed, inferences could be made in the the 3D HPE monocular problem thanks to the temporal awareness. This is why we want to work with videos and take advantage of the consecutiveness of their frames. We want the algorithm to understand that the frames are in a sequence, not simply making a frame-per-frame prediction. Predicting one frame at a time is the simplest approach, but completely diregards the advantage given by a video over an equivalent number of independent frames.

2.2 Approaches

Several methods can be used to solve the 3D video HPE problem, we will detail them in this section. Some of the solutions are presented for completeness, but they focus on a per-frame basis, which is not what we will want to study eventually.

We will focus on 3D HPE from RGB images as this is the simplest, most common case. In this setting, one uses a conventional camera to capture videos. One can use one or multiple cameras to capture images. We will start with the one-camera case. As mentioned previously, this is an ill-posed problem as the captured 2D images come from a projection of the 3D space onto a 2D screen. Reverting the operation seems impossible in the general, mathematically-posed problem but using some commonsensical tricks (including spatial and temporal depth cues) could help achieve better results in some cases. For instance, the length of a human arm lies within a fixed-radius 3D ball for all human beings, which a smart algorithm could use to infer a set of plausible positions to lift such an arm from 2D to 3D. We will start with single-view, single person 3D HPE, then study single-view, multi-person 3D HPE, then finally multi-view 3D HPE.

Single view, single person 3D HPE. In this setting, we distinguish three types of tasks: skeleton-only, the kinematic model and Human Mesh Recovery (HMR). We will detail each of these techniques in their own paragraph.

The Skeleton-only method aims at predicting the spatial (*i.e.* 3D) coordinates of the joints of a person. Estimating a 3D skeleton can be done in two ways. The first way is called direct estimation. It consists of training a neural network to predict the 3D joints directly from the image. The other way is to use an existing 2D joint predictor, which are very performant as we mentioned before, then try to lift the joints from 2D to 3D with a custom neural network.

The kinematic model is another task which consists in estimating the segments between joints. This task allows to pass custom rules to the network and leverage prior knowledge such as restricting joint rotation angles or fixing bone-length ratios.
Some of the best performing methods for skeleton-only methods here are lifting methods such as HRNet [4] by or MHFormer [3].

HMR is the final task in this list. It aims at superimposing a 3D, volumetric body model correctly in space. The body is seen as a unique, connected (in the topological sense) structure where the limbs are able to evolve within their respective ranges of motion. Some notable methods for HMR are Mesh reconstruction with transformers [11] and Mesh Graphormer [12].

Single-view, multi-person 3D HPE. In this setting, there are two main approaches. These are the top-down approaches and the the bottom-up ones. We will start by detailing each of those, then we will compare them.

The top-down approaches start by detecting each individual person in the frame, then they find a so-called “root” for each person, which represents a center joint, almost like a center of mass. Subsequently, these approaches use the root of each person in order to infer their respective 3D poses and position the joints in space. Two of the best-performing techniques for top-down approaches are HMOR [13] and the technique by Yu Cheng et al. [14].

The bottom-up approaches, contrary to the top-down approaches, first find two things. On the one hand, they identify all body joints in the image. On the other hand, they produce depth maps, which are used to group joints and roots into a person’s joints and therefore deduce the pose. The difficulty in bottom-up methods lies in the grouping of joints after they have been detected. Grouping methods run from more sophisticated ones, defining custom limb scores [15], to less sophisticated ones, simply using 3D Euclidian distance from the head joint (most confident join) [16]. Two performant bottom-up methods are XNect [17] and SMAP [18].

Comparison of top-down and bottom-up approaches. Top-down approaches tend to outperform bottom-up ones, at a cost of super-linear time complexity. Indeed, they manage to leverage 2D person detection methods, which as mentioned previously are extremely pertformant. However, the computational complexity (and therefore the inference time) may increase drastically with respect to the number of people in the scene. Increased time and computation is further enhanced when compared to the computation and time complexity of bottom-up approaches which is linear with respect to the number of people. This can be explained by the fact that top-down approaches need to optimize joint detection on each and every individual in the scene after finding their roots. This leads to multiple optimization processes going on, and therefore results in potentially important computational complexity and inference times. Moreover, since the top-down approaches set bounding boxes around the identified people in the image, they discard everything that is outside these bounding boxes and as a result they may loose contextual information. In conclusion, bottom-up approaches tend to be not as good as top-down approaches, but they may be faster and less computationally intense, especially as the number of people in the scene increases.

Multi-view 3D HPE. We previously mentioned that occlusions were challenging for 3D HPE. While this is true for single view HPE, it is not for the multi-view setting. Indeed in this case, an occlusion in a given view may not occur in another view. The complexity however lies in getting multi-view data and locating cameras with respect to one another. The multi-view setting is mostly used for multi-person cases [1], which is why we do not precise whether we work with single-person or multi-person images.
Although 3D methods are more likely to solve occlusions' challenges, reconstructing the 3D relative location of cameras is computationally expensive, especially for multi-person 3D HPE. To suppress this constraint, it may be worth noting that some effort has been made to remove the need for 3D reconstruction before performing 3D HPE [10], in particular through the use of transformers.

Conclusion. The last couple of years saw great improvements in 2D HPE, which bounced into the 3D world thanks to their tight relationship (*c.f.* the lifting methods above). These improvements are qualified, however, by the lack of abundant, real-world, diverse data in comparison the 2D setting.
Maybe also because the links between 2D and 3D are so tight, we see the same kind of problems regarding occlusions and computation as the number of people in the scene grows. Indeed, having an occluded and/or crowded scene on which one performs 2D HPE, then lifting still means that 2D HPE must be performed, hence the logical commonality between challenges in 2D and 3D.

3 Datasets and Metrics

Now that we introduced the approaches that can be used to identify people in images, we need two things: data and ways to evaluate these mathods. Our first focus will be to present datasets, what they offer, why they could be interesting and their pitfalls. Then we will dive into the various metrics that exist in the 3D case, which also come with their share of strengths and weaknesses.

3.1 Datasets

One of the challenges when working on 3D images is to find appropriate datasets. Indeed, in comparison, the 2D setting has many datasets to offer, mainly because

data collection and annotation is fairly easy. In the 3D world however, collecting data requires sensors to be placed on the protagonists' joints or to use some other tricks. Moreover, annotating joints in the 3D space is also a harder task than in the 2D world.
We will review a few datasets in depth, then mention a few others that may be less interesting for our problem.

Humans3.6M [19] is a dataset containing 3.6 millions of 3D human poses along with their respective annotations from accurate sensors. These sensors were placed on 11 actors (6 men, 5 women) performing 17 tasks. The tasks being performed include smoking, taking photo and talking on the phone. There exists a conventional train-test-split called Protocol #1 where subjects S1, S5, S6 and S7 are used for training and subjects S9 and S11 are used for testing. This dataset is commonly regarded as one of the most, if not the most, famous datasets for indoor 3D HPE. One of the reasons for its popularity may be that it can easily be downloaded from the internet.

MPI-INF-3DHP [20] is a dataset of 1.3 million frames captured in a multi-camera, green screen setting. It encompasses 8 actors (4 men, 4 women) performing 8 types of action, some of which are not dynamic. Some are indeed dynamic, such as exercising or doing sports, but some aren't like sitting on a chair of being on the ground. Thanks to the green screen setting, actors can easily be segmented, as well as modified (*e.g.* augmented) in finalized clips. This dataset can also be easily downloaded from the internet.

3.2 Metrics

We saw different approaches and datasets in this paper, but we need some common ground to evaluate them. Metrics in 3D HPE are quite different from the usual accuracy, recall, f1-score that the data science community may be more used to. For this reason, we will detail them in this subsection.

Mean Per Joint Position Error (MPJPE) is the most commonly used metric to evaluate 3D HPE algorithms. It is computed as the average 3D Euclidian distance between predicted joints and their respective ground truth.
Let J_i and J_i^* be the coordinates of the ground truth and prediction of the $i-th$ joint respectively. Let N be the number of joints. Then, the MPJPE is given by:

$$MPJPE = \frac{1}{N} \sum_{i=1}^{N} \|J_i - J_i^*\|_2 \tag{1}$$

Normalized MPJPE (NMPJPE) is similar to MPJPE, but after normalizing the prediction coordinates with respect to the reference.

Mean Per Vertex Error (MPVE) computes the 3D Euclidian distance between the ground truth vertices and the predicted ones.
Let V and V_i^* be the ground truth and the predicted coordinates for the i-th vertex. Then the MPVE is given by:

$$MPVE = \frac{1}{N}\sum_{i=1}^{N} \|V_i - V_i^*\|_2 \tag{2}$$

4 Evaluation and Comparison

5 Conclusion and Perspectives

5.1 Conclusion

5.2 Perspectives

References

1. Zheng C, Wu W, Chen C, Yang T, Zhu S, Shen J, Kehtarnavaz N, Shah M (2022) Deep learning-based human pose estimation: A survey. In: arXiv.org. https://arxiv.org/abs/2012.13392.
2. Wu HY, Nguyen L, Tabei Y, Sassatelli L. Evaluation of deep pose detectors for automatic analysis of film style. InProceedings of the ACM on Human-Computer Interaction 2022 Apr 28.
3. Li W, Liu H, Tang H, Wang P, Van Gool L. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022 (pp. 13147-13156).
4. Wang J, Sun K, Cheng T, Jiang B, Deng C, Zhao Y, Liu D, Mu Y, Tan M, Wang X, Liu W. Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence. 2020 Apr 1;43(10):3349-64.
5. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).
6. Cai Z, Vasconcelos N. Cascade R-CNN: high quality object detection and instance segmentation. IEEE transactions on pattern analysis and machine intelligence. 2019 Nov 28;43(5):1483-98.
7. Tian Z, Shen C, Chen H, He T. Fcos: Fully convolutional one-stage object detection. InProceedings of the IEEE/CVF international conference on computer vision 2019 (pp. 9627-9636).
8. Duan K, Bai S, Xie L, Qi H, Huang Q, Tian Q. Centernet: Keypoint triplets for object detection. InProceedings of the IEEE/CVF international conference on computer vision 2019 (pp. 6569-6578).
9. Chen K, Pang J, Wang J, Xiong Y, Li X, Sun S, Feng W, Liu Z, Shi J, Ouyang W, Loy CC. Hybrid task cascade for instance segmentation. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019 (pp. 4974-4983).

10. Zhang J, Cai Y, Yan S, Feng J. Direct multi-view multi-person 3d pose estimation. Advances in Neural Information Processing Systems. 2021 Dec 6;34:13153-64.
11. Kevin Lin, Lijuan Wang, and Zicheng Liu. 2021. End-to-end human pose and mesh reconstruction with transformers. In CVPR. 1954–1963.
12. Lin K, Wang L, Liu Z. End-to-end human pose and mesh reconstruction with transformers. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021 (pp. 1954-1963).
13. Li J, Wang C, Liu W, Qian C, Lu CH. Hierarchical multi-person ordinal relations for monocular multi-person 3d pose estimation. arXiv 2020. arXiv preprint arXiv:2008.00206.
14. Cheng Y, Wang B, Yang B, Tan RT. Graph and temporal convolutional networks for 3d multi-person pose estimation in monocular videos. InProceedings of the AAAI Conference on Artificial Intelligence 2021 May 18 (Vol. 35, No. 2, pp. 1157-1165).
15. Zanfir A, Marinoiu E, Sminchisescu C. Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018 (pp. 2148-2157).
16. Fabbri M, Lanzi F, Calderara S, Alletto S, Cucchiara R. Compressed volumetric heatmaps for multi-person 3d pose estimation. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020 (pp. 7204-7213).
17. Mehta D, Sotnychenko O, Mueller F, Xu W, Elgharib M, Fua P, Seidel HP, Rhodin H, Pons-Moll G, Theobalt C. XNect: Real-time multi-person 3D motion capture with a single RGB camera. Acm Transactions On Graphics (TOG). 2020 Jul 8;39(4):82-1.
18. Zhen J, Fang Q, Sun J, Liu W, Jiang W, Bao H, Zhou X. Smap: Single-shot multi-person absolute 3d pose estimation. InEuropean Conference on Computer Vision 2020 Aug 23 (pp. 550-566). Springer, Cham.
19. Ionescu C, Papava D, Olaru V, Sminchisescu C. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence. 2013 Dec 12;36(7):1325-39.
20. Mehta D, Rhodin H, Casas D, Fua P, Sotnychenko O, Xu W, Theobalt C. Monocular 3d human pose estimation in the wild using improved cnn supervision. In2017 international conference on 3D vision (3DV) 2017 Oct 10 (pp. 506-516). IEEE.