Temporally-aware monocular 3D Human Pose Estimation on 2D videos

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. Human Pose Estimation (HPE) is the task of locating human body parts in a frame. This problem has drawn quite some interest in the past decade and it will probably continue to do so in the future. Indeed, beyond the scientific community, it may find come useful in augmented reality, the military or the monitoring of crowded areas. In this paper, we focus on 3D HPE. We split it into "single view, single person", "single view, multi-person" and "multi-view", detailing the challenges of each approach. Subsequently, we review the existing, most performant techniques available for each task. Our goal is to make a currated list of the best performing techniques currently available, compare them, understand them and comment on their strengths and weaknesses. We discuss the challenges of occlusion, availability of data and appropriately choosing metrics. Finally, we choose our future working direction which will be focusing on 3D HPE using the temporal dimension of videos. We also comment on the challenges, as well as the opportunities, that such a work may bring.

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 had never been seen before, some of them reaching hundreds of millions of parameters [3]: up to 94.9M for Faster R-CNN [4], 127.3M for Cascade R-CNN [5], 51.0M for FCOS [6], 210.1M for CenterNet [7], 135.2M for Cascade Mask R-CNN [5], 138.2M for Hybrid Task Cascade [8] and 63.4M for Mask R-CNN [4]. 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 single 2D image (monocular HPE), 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 monocular 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, also called single-view. 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 (SVSP) 3D HPE, then study Single View, Multi-Person (SVMP) 3D HPE, then finally Multi-View (MV) 3D HPE.

SVSP 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 [3] by or MHFormer [2].

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 [10] and Mesh Graphormer [11].

SVMP 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 [12] and the technique by Yu Cheng et al. [13].

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 [14], to less sophisticated ones, simply using 3D Euclidian distance from the head joint (most confident join) [15]. Two performant bottom-up methods are PandaNet [16] and SMAP [17].

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.

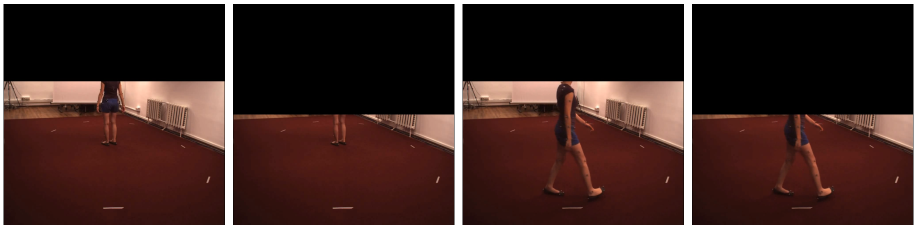
MV 3D HPE. We previously mentioned that occlusions were challenging for 3D HPE. While this is true for single view HPE, it is not for the MV setting. Indeed in this case, an occlusion in a given view may not occur in another view. The complexity however lies in getting MV data and locating cameras with respect to one another. The MV 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 [9], in particular through the use of transformers. Some top-performing methods for MV 3D HPE are Transfusion [18] and MetaFuse [19].

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 Methodology

We saw that occlusions remain a challenge for the monocular 3D HPE task. We want to challenge the existing SVSP methods by testing them on occluded videos and see how they perform. To do so, we choose a video dataset, manually add occlusions and evaluate the performance of the state-of-the-art methods on this occluded dataset, compared to the non-occluded counterpart. We want to answer several questions: first, are these methods able to predict the hidden keypoints? Second, are they able to predict visible keypoints on occluded videos? How well do they predict visible keypoints on occluded compared to non-occluded videos?
We perform predictions thanks to the MMPose library [23], which is a PyTorch-based open-source library for pose estimation. The steps performed are as follows: First, a pre-trained object detector model is used to detect people in the input video frames. The specific detector model used is based on the Faster R-CNN architecture [24] with a ResNet-50 backbone [25]. Then, the detected people are passed through a pre-trained 2D human pose estimator model to estimate their 2D joint positions in the input video frames. The specific pose estimator model used is based on the HRNet-W48 architecture [3]. Third, the estimated 2D joint positions are then passed through a pre-trained 3D human pose lifter model to estimate their corresponding 3D joint positions. The pose lifter model used is based on a fully convolutional neural network trained on the Human3.6M dataset. Finally, the output of the pipeline is a sequence of estimated 3D joint positions for each person detected in the input video frames.
The pipeline presented above is the basis pipeline for our predictions. In order to evaluate the capabilities of the network on occluded scenes, we keep this pipeline, but we feed it an already-occluded video. We then compare the results of the pipeline on the occluded video to the results of the pipeline on the non-occluded video. Our occluded video contains a black rectangle starting from the top of the image, covering the full width of the image. The vertical proportion of the image that is occupied by the rectangle can vary depending on several factors. The first factor is the difficulty we want to set for the prediction. The second one is the position of the actor in the scene; actors can stand anywhere and a given occlusion ratio will not cover the same number of joints from one actor to another (see figure 1). This argument is further complexified by the fact that actors move throughout the video. As a result, some joints may go from being occluded to being visible, or vice-versa.



(a) Actor in the center, occlusion 0.35. (b) Actor in the center, occlusion 0.50. (c) Actor in the bottom, occlusion 0.35. (d) Actor in the bottom, occlusion 0.50.

Fig. 1: Comparison of occlusion ratios and actors positions. 1a vs 1b and 1c vs 1d compare occlusion ratios of 0.35 and 0.50 respectively. 1a vs 1c and 1b vs 1d compare actors positions in the center and at the bottom of the image respectively.

4 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 methods. 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.

4.1 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 heavy manual labeling. Indeed, annotating joints in the 3D space is also a harder task than in the 2D world. We will review a few datasets in depth, which may be of interest for our problem.

Human3.6M [20] 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 photos and talking over 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.

MPI-INF-3DHP [21] 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 these actions are dynamic, such as exercising or doing sports, but some are not, like sitting on a chair or 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.

MuPoTS-3D [22] is a multi-person 3D dataset of 8 people performing 20 real-world scenes, 5 of which are indoors and 15 of which are outdoors. It is a challenging dataset that contains numerous occlusions and lens flare in some outdoor images.

4.2 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. Let N be the number of vertices, then the MPVE is given by:

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

According to some sources, MPJPE depends on the predicted scale of human shape and skeleton, therefore it does not always accurately indicate an accurate pose [1].

3D Percentage of Correct Keypoints (3DPCK) is a metric that considers that a joint is correcly identified if it lies closer that a given distance from ground truth, usually $150mm$. This metric turns the regression problem into a classification one, which may be more convenient for some applications.

5 Evaluation and Comparison

A summary of the best-performing techniques in each category can be found in Table 1. We note that having multiple views on the Human3.6M dataset almost halves the MPJPE compared to single views. Comparing performances on different datasets is already tricky, but the fact that we do not have the same metrics for SVSP and SVMP simply makes the comparison impossible. One additional information on top of the table is that both best-performing skeleton techniques are lifting techniques rather than direct ones.

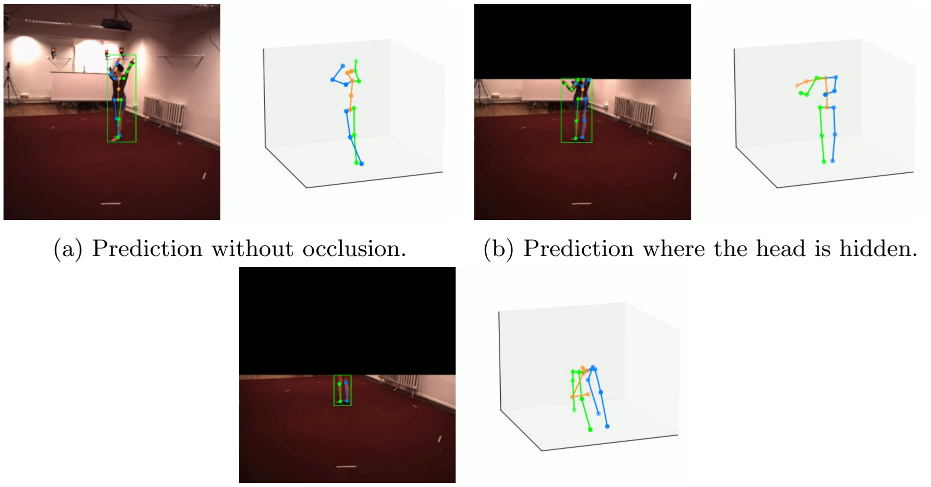
Table 1: Presentation of different, top-performing 3D HPE techniques.

Task	Type	Method	Dataset	Metric	Score
SVSP	Skeleton	[2]	Human3.6M (protocol 1)	MPJPE ↓	43.0
SVSP	Skeleton	[3]	Human3.6M (protocol 1)	MPJPE ↓	42.6
SVSP	HMR	[10]	Human3.6M (protocol 1)	MPJPE ↓	54.0
SVSP	HMR	[11]	Human3.6M (protocol 1)	MPJPE ↓	51.2
SVMP	Top-down	[12]	MuPoTS-3D	3DPCK ↑	82.0
SVMP	Top-down	[13]	MuPoTS-3D	3DPCK ↑	87.5
SVMP	Bottom-up	[16]	MuPoTS-3D	3DPCK ↑	72.0
SVMP	Bottom-up	[17]	MuPoTS-3D	3DPCK ↑	73.5
MV	-	[18]	Human3.6M	MPJPE ↓	25.8
MV	-	[19]	Human3.6M	MPJPE ↓	29.3

6 Results

In section 3, we stated that we wanted to answer several questions about SVSP methods: are they able to predict the occluded keypoints? Are they able to predict visible keypoints on occluded videos? How well do they predict visible keypoints on occluded compared to non-occluded videos? We will now answer these questions.

Ability to predict the occluded keypoints. The model is not able to infer occluded keypoints from the position of the visible ones. When trying to predict the occluded keypoints, the model seems to only predict them in the visible part of the image. This is probably due to the fact that the model is trained on non-occluded videos, and therefore does not have enough data to learn how to predict occluded keypoints. This deficiency suffers drastically from the increase of the number of occluded keypoints. Indeed, when the occlusion ratio is increased (figure 2), or when the subject moves towards the upper part of the image, the occluded keypoints stay in the visible part of the image, thus creating non-sensical predictions, such as predicting that the hands are next to the feet (figure 2c).



(a) Prediction without occlusion.

(b) Prediction where the head is hidden.

(c) Prediction where half of the body is hidden.

Fig. 2: Predictions with varying occlusions ratios.

Ability to predict visible keypoints on occluded videos. Although the visible keypoints may be cluttered and hard to distinguish in figure 2c, the model still seems to be able to predict them somewhat coherently. This is further confirmed by figure 2b, where the model is able to coherently predict the visible keypoints even though the head is hidden. This property is made even more plausible by performing predictions on other images that are not shown here.

Prediction quality of visible keypoints on occluded compared to non-occluded videos. As mentioned previously, the prediction of visible keypoints in occluded scenes seems close enough to the prediction of visible keypoints in non-occluded scenes. However, it seems that the network is less confident about visible keypoints in occluded scenes, as can be noticed by their unstability. Indeed, the prediction of the keypoints seems to "shake". In other words, they show a higher variability than the predicted keypoints in the non-occluded scene setting. This phenomenon is further amplified for occluded keypoints.

7 Analysis of results

8 Conclusion and Perspectives

8.1 Conclusion

Throughout this study of the State of the Art, we saw that 3D HPE can be splitted into three tasks: SVSP , SVMP and MV. The first one can consist of

finding joints (skeleton) or superimposing an artificial 3D body model (HMR). The second one can consist of either identifying each person in the image then finding their joints (top-down), or finding all joints plus a depth map before grouping these joints into the person they belong to (bottom-up). Finally, the last one avoids the strugles of other tasks, such as occlusions, thanks to its multiple views but it requires having multiple cameras and determining precisely where each of them lies with respect to the others.
We also saw that neither for datasets, nor for metrics is there one perfect, universal standard. For datasets, we have to choose between an "easy", maybe utopic, one and other ones with less data. For metrics, we may need to use several of them to properly evaluate the performances of any given model.

8.2 Perspectives

For the future of this study, we could go in many directions but there is one we are particularly interested in, occlusions in 3D videos. This is indeed a challenging topic and we believe we could benefit from the temporal consistency of motion to infer positions when a person becomes occluded. There are some challenges we expect to face, such as choosing an appropriate dataset and metric(s). As we mentioned previously, there is no silver bullet here so we will have to make compromises. If the dataset is too easy, we will not challenge our technique enough. If it is too hard on the other hand, we may want to increase our network model. This may lead to unacceptable training complexities on an already computationally expensive task. We will have to find the right balance, which makes this task both challenging and interesting.

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. (link).
2. 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).
3. Wang J, Yan S, Xiong Y, Lin D. Motion guided 3d pose estimation from videos. InEuropean Conference on Computer Vision 2020 Aug 23 (pp. 764-780). Springer, Cham.
4. 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).
5. 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.
6. 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).
7. 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).

8. 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).
9. 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.
10. Kevin Lin, Lijuan Wang, and Zicheng Liu. 2021. End-to-end human pose and mesh reconstruction with transformers. In CVPR. 1954–1963.
11. 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).
12. 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.
13. 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).
14. 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).
15. 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).
16. Benzine A, Chabot F, Luvison B, Pham QC, Achard C. Pandanet: Anchor-based single-shot multi-person 3d pose estimation. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020 (pp. 6856-6865).
17. 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.
18. Ma H, Chen L, Kong D, Wang Z, Liu X, Tang H, Yan X, Xie Y, Lin SY, Xie X. Transfusion: Cross-view fusion with transformer for 3d human pose estimation. arXiv preprint arXiv:2110.09554. 2021 Oct 18.
19. Xie R, Wang C, Wang Y. Metafuse: A pre-trained fusion model for human pose estimation. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020 (pp. 13686-13695).
20. 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.
21. 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.
22. Mehta D, Sotnychenko O, Mueller F, Xu W, Sridhar S, Pons-Moll G, Theobalt C. Single-shot multi-person 3d pose estimation from monocular rgb. In2018 International Conference on 3D Vision (3DV) 2018 Sep 5 (pp. 120-130). IEEE.
23. MMPose Contributors. OpenMMLab Pose Estimation Toolbox and Benchmark, 2020. Available at: `https://github.com/open-mmlab/mmpose`.
24. Girshick, R., 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

25. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).