SimpleDepthPose: fast and reliable human pose estimation with RGBD-images

Daniel Bermuth, Alexander Poeppel, Wolfgang Reif

Angaben zur Veröffentlichung / Publication details:

Bermuth, Daniel, Alexander Poeppel, and Wolfgang Reif. 2025.
“SimpleDepthPose: fast and reliable human pose estimation with RGBD-images.”
arXiv. arXiv. https://arxiv.org/abs/2501.18478.

SimpleDepthPose: Fast and Reliable Human Pose Estimation with RGBD-Images

Daniel Bermuth
ISSE
University of Augsburg, Germany
daniel.bermuth@uni-a.de

Alexander Poeppel
ISSE
University of Augsburg
poeppel@isse.de

Wolfgang Reif
ISSE
University of Augsburg
reif@isse.de

Abstract

In the rapidly advancing domain of computer vision, accurately estimating the poses of multiple individuals from various viewpoints remains a significant challenge, especially when reliability is a key requirement. This paper introduces a novel algorithm that excels in multi-view, multi-person pose estimation by incorporating depth information. An extensive evaluation demonstrates that the proposed algorithm not only generalizes well to unseen datasets, and shows a fast runtime performance, but also is adaptable to different keypoints. To support further research, all of the work is publicly accessible.

1. Introduction

In many human-centric applications, determining the precise location and pose of individuals is crucial. Pose estimation, which typically involves identifying the positions of a person's joints, is therefore essential for tasks ranging from motion capture to human-computer interaction.

Traditional methods for pose estimation often rely on markers attached to the body, which can be tracked by specialized cameras. While this approach yields high accuracy, it is also cumbersome as it requires individuals to wear specific clothes, which may not be practical or not possible, for example in public settings or in an operating room. Markerless methods, on the other hand, offer greater convenience since they can extract the poses directly from images. However, they face greater computational challenges due to the need to accurately interpret the images.

Using multiple cameras to capture scenes from various angles enhances robustness against occlusions and improves accuracy. Besides using standard RGB cameras, depth cameras can also be employed to provide additional information that can enhance pose estimation.

This work presents a simple but fast and reliable algorithm to detect the joints of multiple humans using RGBD-images from multiple views. It does not require additional training and generalizes well across different scenarios.

The source-code of the presented method can be found at: https://gitlab.com/Percipiote/

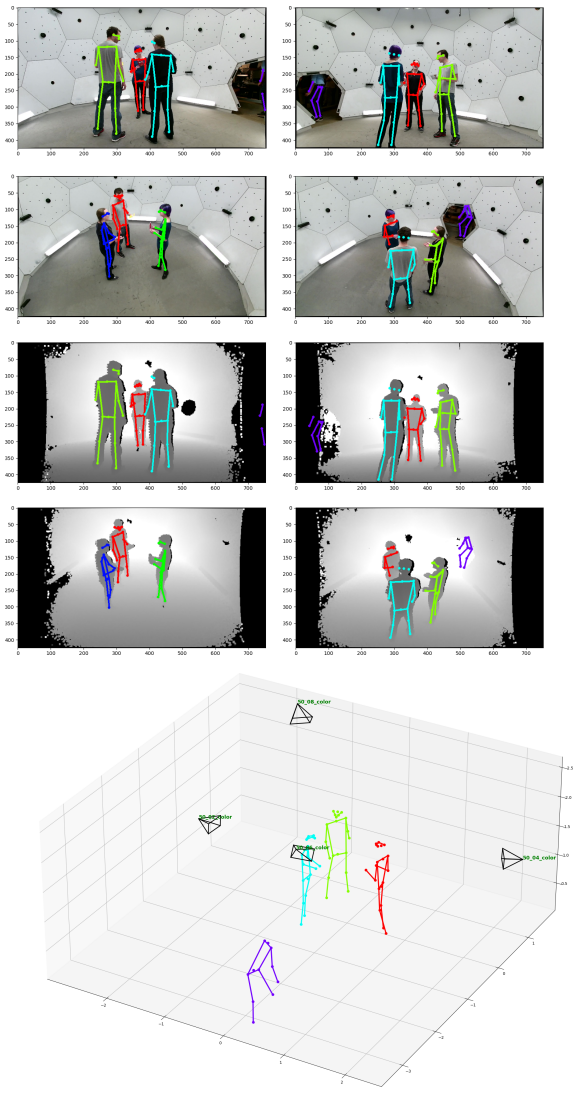

Figure 1. Example of a multi-person pose estimation from multiple camera views (from the *panoptic* dataset [10]). Using the 2D pose detections from the color images (top), a distance to the cameras is extracted from the aligned depth images (center), and the resulting 3D poses of each view are filtered and merged into a final result (bottom).

2. Related Work

Traditionally, human pose estimation is addressed by a two-phase method. Initially, 2D poses are derived from each image, and subsequently, these are fused to estimate 3D poses. The algorithms can be categorized based on their usage of algorithmic strategies versus learning-based methods.

From a learning-based perspective, *VoxelPose* [22] was one of the first concepts, extending the work of *Iskakov et al.* [8] to multi-person estimations. It projects the joint heatmaps from the 2D images into 3D voxelized space and then estimates a rough center for each person, which is then used to create and extract a concentrated cube around each individual. Following this, the locations of the joints are computed using a second neural network. *Faster-VoxelPose* [25] refined this approach by restructuring the 3D voxel space into multiple 2D and 1D projections to improve efficiency. *TEMPO* [4] and *TesseTrack* [17] introduced a temporal dimension to the voxel space to track the poses across frames. Other methods like *PRGnet* [24] use a graph-based method or directly regress the 3D poses from the 2D features, as in *MvP* [23]. *SelfPose3d* [20] is a recent approach that uses self-supervised training. It adopts the structure of *VoxelPose*, and trains both the 2D and 3D networks with randomly augmented 2D poses.

In terms of algorithmic methods, *mvpose* [6] addresses the problem in two phases: initially, it identifies matching 2D poses across images based on geometric and visual similarities, and then it triangulates these poses to construct the final output. *mv3dpose* [21] employs a graph-matching strategy to allocate poses through epipolar geometry and integrates temporal data to compensate for any missing joint information. *PartAwarePose* [5] speeds up the pose-matching process by utilizing poses from the previous frame, and applies a joint-based filter to correct keypoint inaccuracies caused by occlusions. *VoxelKeypointFusion* [1] uses a voxel-based triangulation concept to predict 3D joint proposals from overlapping views and then uses their reprojections to assign them to persons in those views before grouping them into a final result.

Since some cameras are capable of capturing depth data in addition to color images, incorporating this depth information could potentially enhance the accuracy of the pose predictions. A few algorithms have already been developed to leverage this additional information.

OpenPTrack [2, 15], which is frequently employed in robotics, initially calculates the 2D keypoints for each image, and then leverages depth images to determine the distance of each joint to the cameras. It generates a 3D person proposal from each view, which is converted into global world coordinates. Subsequently, these proposals are associated with specific individuals, and a *Kalman-Filter* is applied for joint filtering and temporal smoothing to refine the results. *MVDeep3DPS* [12] utilizes a trainable filter to eliminate inaccurate person proposals before combining them in 3D space. After merging, the method refines these proposals by a calculated confidence score for each body part. *Ryselis et al.* [18] employed a straightforward strategy of just averaging the 3D poses from the different views. *Hansen et al.*[7] generated keypoint heatmaps from depth images and utilized a point cloud to estimate each person's center. They then projected these heatmaps and depth data into a voxelized space to create a 3D pose using a *V2V*[14] network architecture, similar to that of *VoxelPose*. Their source code is not available. *PointVoxel* [16] is a recent work that adopts a similar concept but distinguishes itself by using two separate *V2V*-branches for keypoint and depth voxelmaps instead of merging them directly. It then combines the outputs from these branches. Additionally, it features a synthetic data generator to facilitate generalization across different setups. The source code for this method was not available at the time of writing. *VoxelKeypointFusion* [1] includes a simple voxel-based depth masking approach to remove voxel projections that are not visible in the depth images.

3. SimpleDepthPose

The new algorithm called *SimpleDepthPose* follows a very simple concept with the following steps:

1. Predict joint coordinates for each color image
2. Extract the distance of each detected visible joint from the aligned depth images
3. Group the 3D pose proposals into persons from the last time-step, or create new ones if necessary
4. Filter outliers in each proposal group
5. Average remaining proposals to get the final 3D pose

To predict the joint coordinates in step (1), basically every off-the-shelf pose estimator can be used. One important requirement is though, that the 2D pose estimation model is able to predict only directly visible keypoints, but no occluded ones, because they would result in extracting wrong depth values. Here a *HigherHrNet* [3] model is used, trained and then finetuned on *COCO* [13] to predict only visible joints, and without the refinement step, because this was likely to add occluded joints again. A simpler grouping approach for the association scores was evaluated as well, which even resulted in overall better scores, but only if the refinement was kept, so it can not be used here.

In step (2), the distance is extracted from the depth image by extracting the median value of pixels around the joint coordinates obtained from the RGB-based 2D pose estimator (see Figure 2). The depth values are selected using two rectangular cutouts that form a cross-shape, to emphasize the center region and reduce outliers that would occur at the square's edges. Afterward, a static per-joint offset is added

to the distance, depending on the type of the joint, because a depth camera normally measures the distance to the surface, but the target location is the center of the joint. These offsets are estimated using normal human proportions. For example, $3\,cm$ are added for shoulders and knees, or $1\,cm$ for the wrists. To account for larger persons or (thick) clothing those default values can be adapted. At last, the poses are transformed from camera into world coordinates using the extrinsic calibrations.

Figure 2. Visualization of the cross-shape used to extract the depth value for each joint in a zoom-in of the depth image. All pixels inside the cross are used to calculate the median depth distance.

In step (3), each 3D person proposal is then assigned to a person from the last time-step, by finding the closest match below a distance threshold. If no match is found, a new person is created. Old persons that were not matched are dropped after a certain number of frames.

Following this, a simple outlier removal step (4) is applied. The filter calculates the distance of each joint proposal to the averaged center of its neighboring joints, and if it is above a threshold, it is discarded. In the case of a knee, for example, the neighbors are the hip and ankle. The idea is to remove proposals that are very far from the other joints, creating impossibly long limbs, which are likely to be wrong. Therefore the threshold (default $0.5m$) should cover all limb lengths of normal-sized persons ($< 2m$).

Then, in the last step (5), if there are enough proposals for a joint, a center between them is calculated, and only the *topk* (default 3) closest proposals to this center are averaged into the new joint location. Wrong joint proposals are either caused by poor keypoint predictions or by errors in the depth image (especially at object edges), both resulting mostly in proposals far from the correct location of the joint. See Figure 3 for an example of the proposals and their fused result.

In a direct comparison with *OpenPTrack*, which is architecturally closest, *SimpleDepthPose* is much simpler, but, as can be seen later, also more accurate. Two key differences are to account for the joint visibility and the improved concept of depth extraction. Another one is the filtering and merging approach. While *OpenPTrack* uses a *Kalman-Filter* to smooth the joint locations, which can lag behind in fast motions, *SimpleDepthPose* only uses the joints from the last time-step, the other views, and the joint's direct neighbors to filter outlier joints before merging the remaining proposals to the final results.

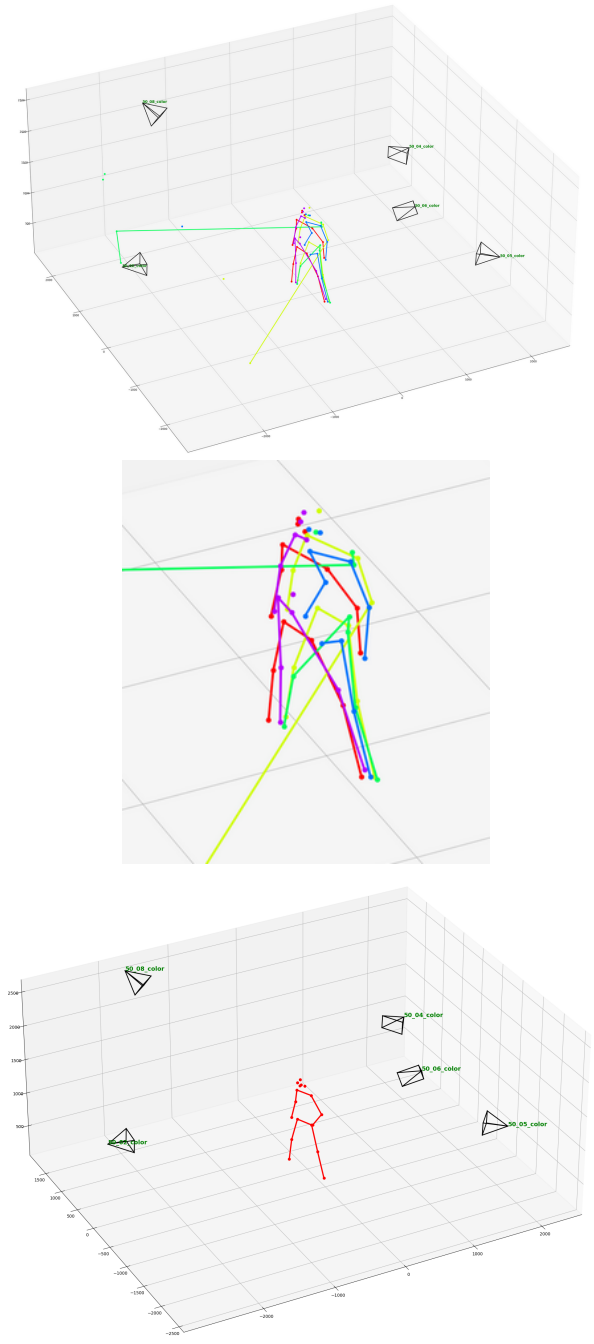

Figure 3. Example of the proposals for each view with some joint errors (top), a zoom-in on the per-view poses (center), and their fused result with the final joints (bottom).

Method	PCP	PCK@100/500		MPJPE	Recall@100/500		Invalid	F1	FPS
MV3DReg [11]	-	-	-	176	-	-	-	-	-
VoxelPose	28.2	10.8	35.9	119	19.5	36.8	15.8	51.2	19.2
VoxelPose (synthetic)	36.3	27.8	65.9	201	15.2	72.4	76.7	35.3	8.3
Faster-VoxelPose	43.3	31.1	55.2	120	29.1	56.0	24.6	64.3	29.3
Faster-VoxelPose (synthetic)	37.9	28.1	45.5	**109**	29.4	46.1	**7.7**	61.5	**30.0**
MvP	0	0	0.1	343	0	0.1	99.9	0.1	8.8
PRGnet	4.9	3.6	6.2	120	3.4	6.4	44.3	11.5	11.9
TEMPO	10.4	7.9	12.6	102	8.8	12.7	14.0	22.2	20.3
SelfPose3d	48.8	36.2	67.7	143	31.1	70.2	36.5	66.7	13.0
mvpose	45.9	32.9	60.2	127	27.1	61.3	18.5	70.0	0.8
mv3dpose	1.3	0.7	2.8	235	0.4	3.1	57.7	5.8	3.0
PartAwarePose	15.3	10.3	25.5	201	6.3	27.7	20.5	41.1	6.9
VoxelKeypointFusion	**54.5**	**43.9**	**75.1**	128	**35.9**	**76.6**	24.2	**76.2**	11.3
MVDeep3DPS [12]	-	-	-	213	-	-	-	-	-
OpenPTrack	11.7	9.9	26.8	323	0.8	33.6	83.9	21.7	1.9
VoxelKeypointFusion	54.0	44.2	72.2	119	36.3	73.4	12.9	79.7	10.9
SimpleDepthPose	**74.0**	**62.0**	**94.3**	113	**54.1**	**96.6**	23.5	**85.4**	**37.2**

Table 1. Transfer to *mvor* [19] without and with depth. All other results without extra citations are taken from [1].

4. Dataset Evaluation

In general, the number of multi-view multi-person depth datasets is very low, only two datasets were found that are suitable for this evaluation, *mvor* [19] and *panoptic* [10]. Both datasets contain RGB and depth images, and the poses of the persons are labeled in 3D. The evaluation is performed using the same metrics as in *VoxelKeypointFusion* [1], which evaluate the error of 13 keypoints (2 shoulders, 2 hips, 2 elbows, 2 wrists, 2 knees, 2 ankles, 1 nose/head). The metrics are described in more detail in [1]. The FPS was measured on a Nvidia-3090 as well.

Multi View Operation Room (MVOR) [19] is more often used in literature and records an operation room from three different viewpoints. It is a relatively complicated dataset, since there is much occlusion, and the persons all have similar clothing. Only the upper body of a person is labeled, and most, but not all persons are labeled.

Table 1 shows that many models have great problems with this setup, and only a few of them, including *SimpleDepthPose*, reach a decent performance on this dataset. The problem is mainly caused by the many occlusions, which result in some persons, or most parts of them, being visible in only one image. All triangulation-based methods consequently struggle to detect such persons at all. When additionally using depth information, on the other hand, one view is enough to correctly detect them.

In this dataset it was notable that *SimpleDepthPose* has large errors at hip joints, which on average are around $170\,mm$ off. In comparison, the upper body joints have an average error between $60\,mm$ to $100\,mm$. This is likely caused by the fact that the hip joints are much more often occluded, which sometimes leads to an incorrect assignment of depth values. Other than that, it significantly outperforms all others in terms of the detected persons and keypoints. It is also faster than every other approach.

The second option for evaluation is the *Panoptic* [10] dataset, which is commonly used for evaluations of RGB-only approaches, but also contains depth recordings. The cameras are mounted in a dome-like structure and point to the center of it. The same evaluation approach as in *VoxelKeypointFusion* [1] was used here as well. Note that since the depth cameras were not time synchronized, their alignment to the color images and to the pose labels is not perfect. They were considered as belonging together if the time difference was below a threshold.

As the results in table 2 show, the RGBD-based approaches outperform half of the algorithmic RGB-based ones in terms of the percentage of detected persons. *SimpleDepthPose* shows a detection rate of persons and joints that is on the same level as the learned approaches that were trained in this setup, while being faster than most. This is especially relevant in safety-critical applications, like in human-robot collaboration for example, where inaccurate

Method	PCP	PCK@100/500		MPJPE	Recall@100/500		Invalid	F1	FPS
VoxelPose	98.5	97.9	98.7	19.3	98.7	98.7	1.1	98.8	8.0
Faster-VoxelPose	99.4	98.6	**99.9**	20.5	99.7	**99.9**	**1.0**	**99.5**	**18.0**
MvP	97.6	97.2	98.3	18.7	98.0	98.5	15.8	90.8	8.9
PRGnet	**99.5**	**99.1**	**99.9**	17.1	**99.9**	**99.9**	2.0	99.0	6.8
TEMPO	98.1	97.4	98.5	**16.8**	98.4	98.4	2.4	98.0	5.1
SelfPose3d	99.3	98.7	99.8	24.9	99.7	**99.9**	8.0	95.7	7.1
mvpose	90.5	75.9	97.5	83.6	73.5	98.5	10.0	94.0	0.1
mv3dpose	84.5	79.4	86.1	48.8	81.8	86.4	15.6	85.4	1.3
PartAwarePose	89.8	79.9	92.1	60.5	83.1	93.0	1.4	95.8	1.5
VoxelKeypointFusion	**97.1**	**94.0**	**99.7**	**47.8**	**97.3**	**99.9**	2.4	98.7	4.2
OpenPTrack	83.0	70.9	95.1	97.6	68.9	97.2	15.5	90.4	1.8
VoxelKeypointFusion	92.6	90.0	96.9	60.1	85.4	97.8	**0.1**	**98.9**	4.0
SimpleDepthPose	96.9	91.2	**100**	**45.5**	**98.6**	**100**	4.7	97.6	**17.7**

Table 2. Replication of *panoptic* results and transfer without and with depth. All other results are taken from [1].

Method	PCP	PCK@100/500		MPJPE	Recall@100/500		Invalid	F1	FPS
SDP (panoptic, with occluded kpts)	95.9	90.3	99.8	49.1	96.1	100	29.6	82.6	17.4
SDP (panoptic, without joint offsets)	95.6	88.8	100	52.9	97.0	100	5.9	97.0	17.5
SDP (panoptic, cameras=1)	65.7	58.6	84.2	155	38.5	89.3	5.6	91.8	50.4
SDP (panoptic, cameras=3)	92.0	82.5	99.5	64.3	90.7	99.7	3.2	98.2	27.1
SDP (panoptic, cameras=10)	98.5	96.8	99.9	37.5	98.1	100	4.8	97.5	9.5
SDP (mvor, pc2vmap)	63.7	50.0	91.1	142	29.5	94.7	21.2	86.0	3.1
SDP (mvor, pc2dimg)	74.0	61.7	94.5	114	52.9	96.8	24.3	85.0	1.2
SDP (panoptic, pc2vmap)	91.3	82.3	99.4	70.4	86.1	99.7	18.4	89.7	1.3
SDP (panoptic, pc2dimg)	96.6	90.8	100	47.4	97.7	100	7.9	95.9	0.5

Table 3. Ablation experiments with SimpleDepthPose.

joint or person estimations can be better handled, for example by generally increasing the required distances, than a missing one. Some of the invalid predictions might be persons entering the room, which are often not labeled. *SimpleDepthPose* already detects them if they are visible in one image, while many other approaches require at least two.

5. Ablation studies

The visibility finetuning of *SimpleDepthPose*, so that only directly visible joints are detected, has a relatively small impact on most metrics, but without it, the number of invalid predictions strongly increases. The added per-joint depth offsets notably improve the average position accuracy.

Even with only a single camera, the algorithm is able to detect most persons, even though the localization accuracy strongly decreases. As expected, the results get better with more cameras, but the inference time increases as well. In case there are many cameras with overlapping views, implementing that persons need to be seen by multiple cameras to be valid could further reduce the number of invalid persons.

Due to the use of depth information from the cameras, the (learned) triangulation step can be skipped, and the algorithm is very fast. On *Panoptic* the average time is about $2.6\,ms$ for the depth extraction and $0.4\,ms$ for multi-view fusion. For better performance, the fusion part is implemented in *C++* and called through a *Python* interface.

Besides directly pairing color and depth images, another option would be to fuse the depth information from all cameras first. For this all depth images are converted to point-clouds which are merged together. After that two different options were evaluated, the first one was to convert the point-cloud back to depth images again (*pc2dimg*), and the second one was to convert it to a 3D voxel-map (*pc2vmap*, voxel resolution $5cm$). These concepts might be interesting if the depth information is not generated by depth cameras, but by other sensors instead, or if a point-cloud is already available, which is often the case in robotic applications. As can be seen in the results, both options output usable detections, with *pc2dimg* being better. In comparison to the original approach of using the depth images directly without fusing them, the fusion takes some extra time (though the current implementation is not very efficient, so this could be faster), while the results are similar, so the fusion step is not considered necessary if depth images are available.

6. Whole-body estimation

Similar to *VoxelKeypointFusion* the algorithmic approach of *SimpleDepthPose* can be easily extended to handle different input keypoints. This can for example be used to predict whole-body keypoints, which include additional face, foot, and finger keypoints.

While the 3D algorithm is easy to adjust, the 2D pose estimation part remains a challenge. Extending the *HigherHrNet* model did not work with its bottom-up concept, especially with the finger keypoints, since often only some of the visible fingers are labeled, and the default training process penalized predictions of unlabeled keypoints. Instead, it is possible to use the whole-body keypoint model from *RTMPose* [9], the same as in *VoxelKeypointFusion*. Here only the visibility finetuning could not be included, because the face and hand keypoints do not contain information about their occlusion status.

This problem results in improvement possibilities for future works. Since very often fingers are occluded by other fingers, they, as a result, have a poor localization performance. So currently the persons and all/most fingers and/or the face keypoints should be directly visible for whole-body estimations.

7. Conclusion

This paper showed, through an evaluation of different datasets, that the proposed *SimpleDepthPose* algorithm is a very fast and reliable approach if depth data is available, and also shows the best generalization results among other methods, without requiring any additional training.

One limitation is that since there is no neural refinement of the resulting 3D poses, the location accuracy is highly dependent on the accuracy of the depth images. Each keypoint also has to be visible in at least one image to be detected.

Following the results of the experiments, the use of depth data does not necessarily lead to more accurate results in terms of joint localization, but can significantly increase the number of detected keypoints and persons, especially in strongly occluded settings. Since in many applications it is more important to detect persons and their poses at all, than to have a very accurate joint localization, using or integrating depth sensors can therefore be recommended, because as shown, they can increase the algorithm's performance.

References

[1] Daniel Bermuth, Alexander Poeppel, and Wolfgang Reif. VoxelKeypointFusion: Generalizable Multi-View Multi-Person Pose Estimation. *arXiv preprint arXiv:2410.18723*, 2024. 2, 4

[2] Marco Carraro, Matteo Munaro, Jeff Burke, and Emanuele Menegatti. Real-time marker-less multi-person 3D pose estimation in RGB-depth camera networks. In *Intelligent Autonomous Systems 15: Proceedings of the 15th International Conference IAS-15*, pages 534–545. Springer, 2019. 2

[3] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5386–5395, 2020. 2

[4] Rohan Choudhury, Kris M Kitani, and László A Jeni. TEMPO: Efficient multi-view pose estimation, tracking, and forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14750–14760, 2023. 2

[5] Hau Chu, Jia-Hong Lee, Yao-Chih Lee, Ching-Hsien Hsu, Jia-Da Li, and Chu-Song Chen. Part-aware measurement for robust multi-view multi-human 3d pose estimation and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1472–1481, 2021. 2

[6] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and Robust Multi-Person 3D Pose Estimation from Multiple Views. 2019. 2

[7] Lasse Hansen, Marlin Siebert, Jasper Diesel, and Mattias P Heinrich. Fusing information from multiple 2D depth cameras for 3D human pose estimation in the operating room. *International journal of computer assisted radiology and surgery*, 14:1871–1879, 2019. 2

[8] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7718–7727, 2019. 2

[9] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose. *arXiv preprint arXiv:2303.07399*, 2023. 5

[10] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015. 1, 4

[11] Abdolrahim Kadkhodamohammadi and Nicolas Padoy. A generalizable approach for multi-view 3d human pose regression. *Machine Vision and Applications*, 32(1):6, 2021. 4

[12] Abdolrahim Kadkhodamohammadi, Afshin Gangi, Michel de Mathelin, and Nicolas Padoy. A multi-view RGB-D approach for human pose estimation in operating rooms. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 363–372. IEEE, 2017. 2, 4

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[14] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018. 2

[15] Matteo Munaro, Alex Horn, Randy Illum, Jeff Burke, and Radu Bogdan Rusu. OpenPTrack: People tracking for heterogeneous networks of color-depth cameras. In *IAS-13 Workshop Proceedings: 1st Intl. Workshop on 3D Robot Perception with Point Cloud Library*, pages 235–247. Citeseer, 2014. 2

[16] Zhiyu Pan, Zhicheng Zhong, Wenxuan Guo, Yifan Chen, Jianjiang Feng, and Jie Zhou. PointVoxel: A Simple and Effective Pipeline for Multi-View Multi-Modal 3D Human Pose Estimation. *arXiv preprint arXiv:2312.06409*, 2023. 2

[17] N Dinesh Reddy, Laurent Guigues, Leonid Pishchulin, Jayan Eledath, and Srinivasa G Narasimhan. Tessetrack: End-to-end learnable multi-person articulated 3d pose tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15190–15200, 2021. 2

[18] Karolis Ryselis, Tautvydas Petkus, Tomas Blažauskas, Rytis Maskeliūnas, and Robertas Damaševičius. Multiple Kinect based system to monitor and analyze key performance indicators of physical training. *Human-Centric Computing and Information Sciences*, 10:1–22, 2020. 2

[19] Vinkle Srivastav, Thibaut Issenhuth, Abdolrahim Kadkhodamohammadi, Michel de Mathelin, Afshin Gangi, and Nicolas Padoy. MVOR: A multi-view RGB-D operating room dataset for 2D and 3D human pose estimation. *arXiv preprint arXiv:1808.08180*, 2018. 4

[20] Vinkle Srivastav, Keqi Chen, and Nicolas Padoy. SelfPose3d: Self-Supervised Multi-Person Multi-View 3d Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2502–2512, 2024. 2

[21] Julian Tanke and Juergen Gall. Iterative Greedy Matching for 3D Human Pose Tracking from Multiple Views. In *German Conference on Pattern Recognition*, 2019. 2

[22] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[23] Tao Wang, Jianfeng Zhang, Yujun Cai, Shuicheng Yan, and Jiashi Feng. Direct Multi-view Multi-person 3D Human Pose Estimation. *Advances in Neural Information Processing Systems*, 2021. 2

[24] Size Wu, Sheng Jin, Wentao Liu, Lei Bai, Chen Qian, Dong Liu, and Wanli Ouyang. Graph-based 3d multi-person pose estimation using multi-view images. In *ICCV*, 2021. 2

[25] Hang Ye, Wentao Zhu, Chunyu Wang, Rujie Wu, and Yizhou Wang. Faster VoxelPose: Real-time 3D Human Pose Estimation by Orthographic Projection. In *European Conference on Computer Vision (ECCV)*, 2022. 2