International Journal of Computer Vision (2021) 129:703–718
https://doi.org/10.1007/s11263-020-01398-9

AdaFuse: Adaptive Multiview Fusion for Accurate Human Pose Estimation in the Wild

Zhe Zhang[1] · Chunyu Wang[2] · Weichao Qiu[3] · Wenhu Qin[1] · Wenjun Zeng[2]

Received: 23 December 2019 / Accepted: 26 October 2020 / Published online: 16 November 2020

Abstract
Occlusion is probably the biggest challenge for human pose estimation in the wild. Typical solutions often rely on intrusive sensors such as IMUs to detect occluded joints. To make the task truly unconstrained, we present *AdaFuse*, an adaptive multiview fusion method, which can enhance the features in occluded views by leveraging those in visible views. The core of *AdaFuse* is to determine the point-point correspondence between two views which we solve effectively by exploring the sparsity of the heatmap representation. We also learn an adaptive fusion weight for each camera view to reflect its feature quality in order to reduce the chance that good features are undesirably corrupted by "bad" views. The fusion model is trained end-to-end with the pose estimation network, and can be directly applied to new camera configurations without additional adaptation. We extensively evaluate the approach on three public datasets including Human3.6M, Total Capture and CMU Panoptic. It outperforms the state-of-the-arts on all of them. We also create a large scale synthetic dataset *Occlusion-Person*, which allows us to perform numerical evaluation on the occluded joints, as it provides occlusion labels for every joint in the images. The dataset and code are released at https://github.com/zhezh/adafuse-3d-human-pose.

Keywords Human pose estimation · Multiple camera fusion · Epipolar geometry

1 Introduction

Accurately estimating 3D human pose from multiple cameras has been a longstanding goal in computer vision (Liu et al. 2011; Bo and Sminchisescu 2010; Gall et al. 2010; Rhodin et al. 2018; Amin et al. 2013; Burenius et al. 2013; Pavlakos et al. 2017; Belagiannis et al. 2014). The ultimate goal is to recover absolute 3D locations of the body joints in a world coordinate system from multiple cameras placed in natural environments. The task has attracted a lot of attention because it can benefit many applications such as augmented and virtual reality (Starner et al. 2003), human-computer-interaction and intelligent player analysis in sport videos (Bridgeman et al. 2019).

The task is often addressed by a simple two-step framework. In the first step, it tries to detect the 2D poses in all camera views, for example, by a convolutional neural network (Cao et al. 2017; Xiao et al. 2018). Then in the second step, it recovers the 3D pose from the multiview 2D poses either by analytical methods (Burenius et al. 2013; Pavlakos et al. 2017; Belagiannis et al. 2014; Qiu et al. 2019; Amin et al. 2013) or by discriminative models (Iskakov et al. 2019; Tu et al. 2020). The camera parameters are usually assumed known in these approaches. The development of powerful network architectures such as Newell et al. (2016) has notably improved the 2D pose estimation quality, which in turn reduces the 3D error remarkably. For example, in Qiu

Communicated by Mei Chen.

Zhe Zhang and Chunyu Wang have contributed equally to this work.

Work done when Zhe Zhang is an intern at Microsoft Research Asia.

✉ Wenhu Qin
qinwenhu@seu.edu.cn

Zhe Zhang
zhangzhecnjs@gmail.com

Chunyu Wang
chnuwa@microsoft.com

Weichao Qiu
qiuwc@gmail.com

Wenjun Zeng
wezeng@microsoft.com

[1] Southeast University, Nanjing, China

[2] Microsoft Research Asia, Beijing, China

[3] The Johns Hopkins University, Baltimore, MD, USA

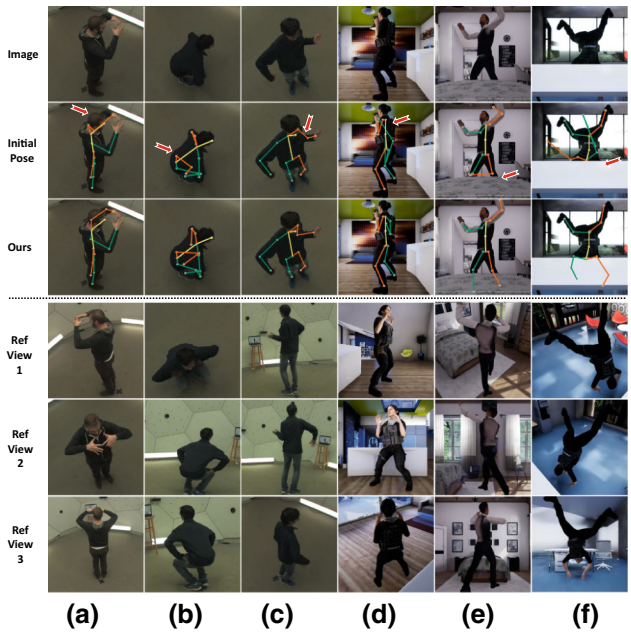

Fig. 1 Our approach accurately detects the poses even though they are occluded by leveraging the features in other views. The bottom three rows are images from other view angles of the scene for readers to better perceive the 3D poses of the actors

et al. (2019), the 3D error on Human3.6M (Ionescu et al. 2014) decreases significantly from 52 to 26 mm.

However, obtaining small errors on benchmark datasets does not imply that the task has been truly solved unless the challenges such as background clutter, human appearance variation and occlusion encountered in real world applications are well addressed. In fact, a growing amount of efforts (Zhou et al. 2017; Ci et al. 2019; Yang et al. 2018; Rogez and Schmid 2016; Pavlakos et al. 2018; Ci et al. 2020) have been devoted to improving the pose estimation performance in challenging scenarios, for example, by augmenting the training dataset (Zhou et al. 2017; Yang et al. 2018; Varol et al. 2017) with more images or by using more robust sensors such as IMUs (Trumble et al. 2017). We will discuss about this type of work in more details in Sect. 2.

In this work, we propose to solve the problem in a different way by multiview feature fusion. The approach is orthogonal to the previous efforts. As shown in Fig. 1, our approach can accurately detect the joints even when they are occluded in certain views. The motivation behind our approach is that a joint occluded in one view may be visible in other views. So it is generally helpful to fuse the features at the corresponding locations in different views. To that end, we present a flexible multiview fusion approach termed *AdaFuse*. Figure 2 shows the pipeline. It first uses camera parameters to compute the point-line correspondence between a pair of views. Then it "finds" the matched point on the line by exploring the sparsity of the heatmap representation without performing the challenging point-point matching. Finally, the features of the matched points in different views are fused. The approach can effectively improve the feature quality in occluded views. In addition, for a new environment with different camera poses, we can directly use *AdaFuse* without re-training as long as the camera parameters are available. This improves the applicability of the approach in real applications.

The performance of *AdaFuse* is further boosted by learning an adaptive fusion weight for each view to reflect its feature quality. This weight is leveraged in fusion in order to reduce the impact of low-quality views. If a joint is occluded in one view, its features are also likely corrupted. In this case, we hope to give a small weight to this view when performing multiview fusion such that the high-quality features in the visible views are dominant, and are free from being corrupted by low-quality features. We add some simple layers to the pose estimation network to predict heatmap quality based on the heatmap distribution and cross view consistency. We observe in our experiments that the use of adaptive fusion notably improves the performance.

We evaluate our approach on three public datasets including Human3.6M (Ionescu et al. 2014), Total Capture (Trumble et al. 2017) and CMU Panoptic (Joo et al. 2019). It outperforms the state-of-the-arts demonstrating the effectiveness of our approach. In addition, we also compare it to a number of standard multiview fusion methods such as RANSAC in order to give more detailed insights. We evaluate the generalization capability of our approach by training and testing on different datasets. We also create a synthetic human pose dataset in which human are purposely occluded by objects. The dataset allows us to perform evaluation on the occluded joints.

The rest of the paper is organized as follows. In Sect. 2, we discuss the related work on multiview 3D human pose estimation with special focus on the approaches that aim to improve the performance in challenging environments. Section 3 introduces the basics for multiview feature fusion to lay the groundwork for *AdaFuse*. Then we describe how we learn adaptive weight for each camera view to reflect the feature quality, as well as the details of *AdaFuse*. In Sects. 5 and 6, we introduce the experimental datasets and results, respectively. Section 7 concludes this work.

2 Related Work

We first review the related work on multiview 3D human pose estimation in section 2.1. Then Sect. 2.2 summarizes the techniques that are used to improve the in-the-wild performance. Finally, in Sect. 2.3, we discuss the approaches on consensus learning such as RANSAC. This is necessary for multiple sensor fusion because the sensors could have con-

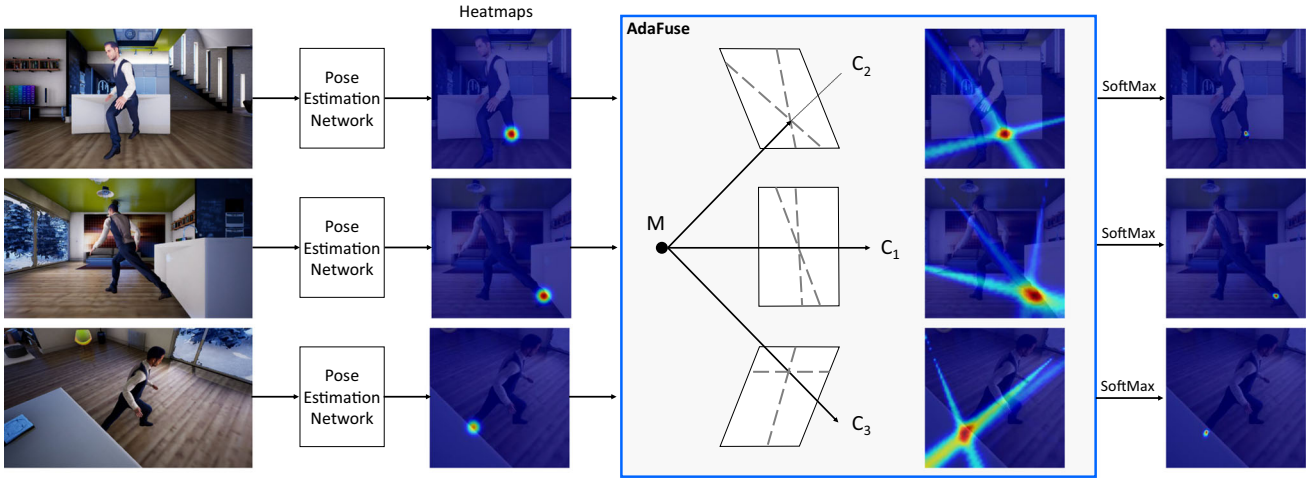

Fig. 2 Overview of *AdaFuse*. It takes multiview images as input and outputs 2D poses of all views jointly. It first uses a pose estimation network to obtain 2D heatmaps for each view. Then on top of epipolar geometry, the heatmaps from all camera views are fused. Finally, we apply the SoftMax operator to suppress the small noises introduced in fusion. Consequently, pose estimation in each view benefits from other views

tradictory predictions and the outliers should be removed to ensure the good fusion quality.

2.1 Multiview 3D Human Pose Estimation

We briefly classify the multiview 3D human pose estimation methods into two classes. The first class is model-based approaches which are also known as analysis-by-synthesis approaches (Liu et al. 2011; Gall et al. 2010; Moeslund et al. 2006; Sigal et al. 2010; Perez et al. 2004). They first model human body by simple primitives such as sticks and cylinders. Then the parameters of the model (i.e. poses) are continuously updated according to the observations in multiview images until the model can be explained by the image features. The resulted optimization problem is usually non-convex. So expensive sampling techniques are often used. The main difference among those approaches lies in the adopted image features and the optimization algorithms. We refer the interested readers to earlier survey papers such as Moeslund et al. (2006).

The advantage of the model-based approaches lies in its capability to handle occlusion because of the inherent structure prior embedded in human model. These approaches aggregate the local features as evidence to infer the global model parameters with the inherent human body structure as constraints. So if a joint is occluded, it can still rely on other joints to guess the possible locations that are consistent with the prior. However, the model-based approaches get larger 3D errors than the model-free approaches due to the difficult optimization problems.

The second class is model-free approaches (Qiu et al. 2019; Iskakov et al. 2019; Burenius et al. 2013; Pavlakos et al. 2017; Dong et al. 2019; Amin et al. 2013; Belagiannis et al. 2014; Xie et al. 2020) which often follow a two-step framework. They first detect 2D poses in images of all camera views. Then with the aid of camera parameters, they recover the 3D pose using either triangulation (Amin et al. 2013; Iskakov et al. 2019) or pictorial structure models (Burenius et al. 2013; Pavlakos et al. 2017; Dong et al. 2019). Recursive pictorial structure model is introduced in Qiu et al. (2019) to speed up the inference process. The authors in Iskakov et al. (2019) also propose to use learnable triangulation (Hartley and Zisserman 2003) for human pose estimation which is more robust to inaccurate 2D poses. If the 2D poses are accurate, the recovered 3D poses are guaranteed to be accurate without worrying about being trapped in local optimum as the model-based methods.

The development of more powerful network architectures (Newell et al. 2016; Sun et al. 2019) has dramatically improved the 2D pose estimation accuracy on benchmark datasets, which in turn also decreases the 3D pose estimation error. For example, on the most popular benchmark Human3.6M (Ionescu et al. 2014), the 3D MPJPE error has decreased to about 20 mm which can meet the requirements of many real-life applications.

2.2 Improving "In the Wild" Performance

Sensors Occlusion is probably the biggest challenge for in-the-wild scenarios. One straightforward solution is to use additional sensors such as IMUs (Trumble et al. 2017) and radio signals (Zhao et al. 2019), which are not impacted by occlusion. For example, Roetenberg et al. (2009) place 17 IMUs at the rigid bones. If the measurements are accurate,

the 3D pose is fully determined. In practice, however, the accuracy is limited by the drifting problem. To that end, some approaches (Trumble et al. 2017; von Marcard et al. 2018; Gilbert et al. 2019; Malleson et al. 2017; Zhang et al. 2020) propose to fuse images and IMUs to achieve more robust pose estimation. Some works (Zhao et al. 2019; Li et al. 2019; Zhao et al. 2018) leverage the fact that wireless signals in the WiFi frequencies traverse walls and reflect off the human body, and propose a radio-based system that can estimate 2D poses even when persons are completely occluded by walls. However, these approaches also have their own problems. For example, how to effectively fuse visual and inertial signals for IMU-based approaches? Besides, wearing sensors on the body is intrusive, and is not acceptable in some scenarios such as football games. On the other hand, the WiFi-based solutions cannot deal with self-occlusion which is a big limitation.

Data Augmentation Collecting more images for model training is an effective approach to improve the generalization performance. For example in Zhou et al. (2017) and Qiu et al. (2019), the authors propose to use the MPII (Andriluka et al. 2014) and the COCO (Lin et al. 2014) datasets to help train the 2D module of the 3D pose estimators which effectively reduces the risk of over-fitting to simple training datasets. However, annotating a sufficiently large pose dataset is expensive and time consuming. So some approaches (Rogez and Schmid 2016; Varol et al. 2017; Hoffmann et al. 2019; Chen et al. 2016; Lassner et al. 2017) propose to generate synthetic images. The main issue is to bridge the gap between the synthetic and real images such that the model trained on synthetic images can be applied to real images. To that end, some approaches such as Peng et al. (2018) propose to use generative adversarial networks to generate realistic images.

Spatial-Temporal Context Models Some approaches propose to use spatial-temporal context models to jointly detect all joints in a video sequence such that each joint can benefit from other joints in the same or neighboring frames. Intuitively, if a body joint is occluded thus is difficult to be detected according to its own appearance, they can use the locations of other joints to guess the possible location. For example, in a previous work (Cao et al. 2017; Kreiss et al. 2019), the authors propose to detect body parts, i.e. the links connecting two joints, in addition to the individual joints. This provides a chance to mutually enhance the detection of the two linked joints. In Cheng et al. (2019) and Pavllo et al. (2019)), temporal convolution is utilized to deal with occlusion in current frames. Some works such as Qiu et al. (2019) propose to establish the spatial correspondence across multiple camera views, and leverage multi-view features for robust joint detection. Significant performance improvement has been achieved for the occluded joints on several benchmark datasets. The main drawback of the approach (Qiu et al. 2019) is the lack of flexibility in practice since it needs to train a separate fusion network for every possible camera placement. Our work differs from Qiu et al. (2019) in that it can be applied to new environments with different numbers of cameras and different camera poses without additional adaptation. We will compare the two methods in the experiments.

2.3 Consensus Learning

A fundamental problem in multi-sensor fusion is to detect and remove outliers as the sensors may produce inconsistent measurements. RANSAC (Fischler and Bolles 1981) is the most commonly used outlier detection method. The main assumption is that the dataset consists of inliers. It produces reasonable results only with a certain probability which increases as the number of inliers. In practice, when the number of sensors is small, the probability of detecting the real outliers is also small. For example, in multiview human pose estimation, the number of cameras is only four to eight for most benchmark datasets (Ionescu et al. 2014; Trumble et al. 2017). For such cases, we observe that RANSAC may not be the best option.

In recent years, uncertainty learning (Kendall and Gal 2017; Gal and Ghahramani 2015; Lakshminarayanan et al. 2017; Zafar et al. 2019; Lakshminarayanan et al. 2017; Pleiss et al. 2017) has attracted a lot of attention which is particularly important for high-risk applications such as autonomous driving and medical diagnosis (Gal 2016; Ghahramani 2016). The main idea is that, when a model makes a prediction, it also outputs a score reflecting the confidence of the prediction. Consider an autonomous car that uses a neural network to detect people. If the network is not confident about the prediction, the car could probably rely on other sensors for making the correct decision. Uncertainty is introduced to computer vision in Kendall and Gal (2017), Kreiss et al. (2019), He et al. (2019) and Ilg et al. (2018). Another branch of approaches such as Guo et al. (2017) and Pleiss et al. (2017) propose to learn uncertainty by calibration. They propose to train the model such that the probability associated with the predicted class label agrees with its ground truth correctness likelihood.

The concept of uncertainty can be leveraged to reduce the impact of outliers. For example, in Iskakov et al. (2019), the authors propose to predict an uncertainty score for each joint in each view. The score is used to weigh each view when doing triangulation. This dramatically reduces the 3D pose estimation error. Inspired by the success of uncertainty learning in computer vision tasks, we propose to learn uncertainty for multiview feature fusion. The predicted uncertainty is used as a weight when fusing multiview features. We show this adaptive feature fusion could effectively improve the fusion quality.

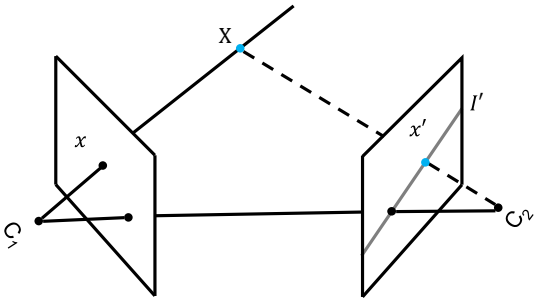

Fig. 3 Illustration of the point-line correspondence in two views. For an arbitrary point $\mathbf{x}$ in one view, the corresponding point $\mathbf{x}'$ in another view has to lie on the epipolar line $\mathbf{I}'$. This is the core of *AdaFuse* for finding corresponding points in other views

Fig. 4 Epipolar geometry based heatmap fusion. For each location $\mathbf{x}$ in the first view, we first compute the corresponding epipolar lines in the other two views. Then we find the largest responses on the two lines, respectively and add them to the original response at $\mathbf{x}$

3 The Basics for Multiview Fusion

We first introduce the basics for multiview fusion to lay the groundwork for *AdaFuse*. In particular, we discuss how to establish the point-point correspondence between two views such that the features correspond to the same 3D space point can be fused together. The narrow baseline correspondence can be solved efficiently by local feature matching. However, in the context of multiview human pose estimation where only a small number of cameras are placed far away from each other, the local features cannot be robustly detected and matched especially for texture-less human regions. This poses a serious challenge.

To solve the problem, we present a coarse-to-fine approach to find matched points. It first establishes the point-to-line correspondence between two views by epipolar geometry, and then implicitly determine the point-to-point correspondence by exploring the sparsity of the heatmap representations. The approach notably simplifies the task because it avoids the challenging step of finding the exact correspondence. We first introduce epipolar geometry in Sect. 3.1 in order to determine the point-to-line correspondence. Then in Sect. 3.2, we describe how we adapt epipolar geometry to perform multiview heatmap fusion. Finally, we discuss the side effect caused by the simplified fusion strategy and our solution in Sect. 3.3.

3.1 Epipolar Geometry

Let us denote a point in 3D space as $\mathbf{X} \in \mathcal{R}^{4\times1}$ as shown in Fig. 3. This could be the location of a body joint in the context of pose estimation. Note that homogeneous coordinate and column vector are used to represent a point. The 3D point is imaged in two camera views, at $\mathbf{x} = \mathbf{P}\mathbf{X}$ in the first, and $\mathbf{x}' = \mathbf{P}'\mathbf{X}$ in the second, where $\mathbf{x}$ and $\mathbf{x}' \in \mathcal{R}^{3\times1}$ represent 2D points in images, $\mathbf{P}$ and $\mathbf{P}' \in \mathcal{R}^{3\times4}$ are the projection matrix for each camera. Since the two 2D points correspond to the same 3D point and have the same semantic meanings, their features can be safely fused such that each view benefits from the other view.

The epipolar geometry (Hartley and Zisserman 2003) between two views is essentially the geometry of the intersection of the image planes with the pencil of planes having the baseline as axis. The baseline is the line joining the camera centers C_1 and C_2. In particular, for each location $\mathbf{x}$ in the first view, it helps us to determine the location of the corresponding point $\mathbf{x}'$ in the second view without having to know $\mathbf{X}$.

We can see from Fig. 3 that the image points $\mathbf{x}$ and $\mathbf{x}'$, the 3D point $\mathbf{X}$, and the camera centers $\mathbf{C_1}$ and $\mathbf{C_2}$ lie on the same plane $\boldsymbol{\pi}$. The plane intersects with the two image planes at epipolar lines $\mathbf{I}$ and $\mathbf{I}'$, respectively. In particular,

$$\begin{aligned} \mathbf{I}' &= \mathbf{F}\mathbf{x} \\ \mathbf{I} &= \mathbf{F}^\top \mathbf{x}', \end{aligned} \tag{1}$$

where $\mathbf{F} \in \mathcal{R}^{3\times3}$ is fundamental matrix which can be derived from $\mathbf{P}$ and $\mathbf{P}'$. Readers can refer to Hartley and Zisserman (2003) for detail derivation. In addition, the rays back-projected from $\mathbf{x}$ and $\mathbf{x}'$ intersect at $\mathbf{X}$, and the rays are coplanar, lying in $\boldsymbol{\pi}$. It is straightforward to derive that the location of $\mathbf{x}'$ which corresponds to $\mathbf{x}$ is guaranteed to lie on the epipolar line $\mathbf{I}'$. However, we have to leverage additional information such as appearance to determine the exact location of $\mathbf{x}'$ on $\mathbf{I}'$.

In the context of multiview feature fusion, for every image point $\mathbf{x}$, we need to find the corresponding point $\mathbf{x}'$ in the second view so that we can fuse the features at $\mathbf{x}$ with those at $\mathbf{x}'$ and obtain more robust pose estimations. Since we do not know the depth of $\mathbf{X}$, it could move freely on the line defined by the camera center $\mathbf{C_1}$ and image point $\mathbf{x}$. However, we know that $\mathbf{x}'$ cannot span the entire image plane but is restricted to the line $\mathbf{I}'$. In the following Sect. 3.2, we will describe how we perform multiview feature fusion based on epipolar geometry.

Sampson Distance In practice, usually we have 2D measurements $\mathbf{x}$ and $\mathbf{x}'$ corresponding to the same 3D location $\mathbf{X}$ which is unknown. Due to measurement noise and errors,

the line $\mathbf{C_1 x}$ and $\mathbf{C_2 x}'$ might not intersect exactly at location $\mathbf{X}$. To obtain the optimal estimation for $\mathbf{X}$, we search for $\hat{\mathbf{X}}$ subject to

$$d^2_{Reproj} = \min_{\hat{\mathbf{X}}} d^2\left(\mathbf{x}, \mathbf{P}\hat{\mathbf{X}}\right) + d^2\left(\mathbf{x}', \mathbf{P}'\hat{\mathbf{X}}\right), \tag{2}$$

where $d(\cdot)$ denotes Euclidean distance, d_{Reproj} represents the reprojection distance between $\mathbf{x}$ and $\mathbf{x}'$. Since there is optimization process when obtaining d_{Reproj}, we adopt an one-step method which is its first-order approximation (Hartley and Zisserman 2003). This approximation is also called Sampson distance as

$$d_{Sampson} = \frac{\mathbf{x}'^{\top}\mathbf{F}\mathbf{x}}{(\mathbf{F}\mathbf{x})_1^2 + (\mathbf{F}\mathbf{x})_2^2 + \left(\mathbf{F}^{\top}\mathbf{x}'\right)_1^2 + \left(\mathbf{F}^{\top}\mathbf{x}'\right)_2^2}, \tag{3}$$

where $\mathbf{F}$ is fundamental matrix, the subscript 1 or 2 denotes the first or second element of a vector. By using Sampson distance, we can directly obtain distance between a pair of locations without knowing the intermediate $\hat{\mathbf{X}}$. In *AdaFuse*, we use Sampson distance to represent to what extent a pair of 2D joint detections support each other.

3.2 Heatmap Fusion

Multiview fusion is applied to heatmaps rather than intermediate features as shown in Fig. 2. This is because heatmap has the nice property of sparsity which can simplify the point-point matching. A heatmap produces a per-pixel likelihood for joint locations in the image. Specifically, it is generated as a two-dimensional Gaussian distribution centered at the coordinate of the joint. So it has a small number of large responses near the joint location, and a large number of zeros at other locations. See Fig. 4a for an example heatmap of the right knee joint.

The sparse heatmaps allow us to safely skip the exact point-point matching because the features at the "zero" locations on the epipolar line are not contributing to the feature fusion. As a result, instead of trying to find the exact corresponding location in the other view, *we simply select the largest response on the epipolar line as the matched point*. This is a reasonable simplification because the corresponding point usually has the largest response. For example, in Fig. 4, for each location $\mathbf{x}$, we first compute the corresponding epipolar lines in the other two camera views. Then we find the largest responses on the two epipolar lines, respectively and fuse them with the response at $\mathbf{x}$.

Let us denote the heatmap in view v as $\mathbf{H}^v$. The response at the location $\mathbf{x}$ of the heatmap is denoted as $\mathbf{H}^v(\mathbf{x})$. The corresponding epipolar line of $\mathbf{x}$ in view u is denoted as $\mathbf{I}^u(\mathbf{x})$ which consists of a number of discrete locations on the heatmap $\mathbf{H}^u$. The epipolar line can be analytically computed based on the camera parameters for every location $\mathbf{x}$. Then we formulate multiview fusion as

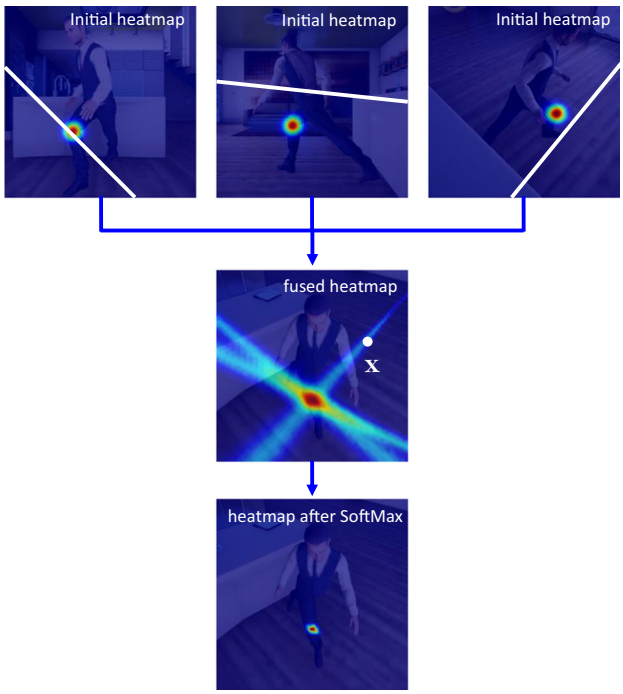

Fig. 5 The ambiguity problem in our simplified multiview fusion approach and our solution. We can see from the "fused heatmap" that the correct location has the largest response which is as expected. However, for an incorrect location $\mathbf{x}$, there is also a chance that the response is also enhanced by at most one view. Fortunately, the correct location will be enhanced more times (three times in this example) leading to the largest response. So we apply the SoftMax operator to the fused heatmap to reduce the responses at incorrect locations

$$\hat{\mathbf{H}}^v(\mathbf{x}) = \lambda \mathbf{H}^v(\mathbf{x}) + \frac{1-\lambda}{N}\sum_{u=1}^{N} \max_{\mathbf{x}' \in \mathbf{I}^u(\mathbf{x})} \mathbf{H}^u(\mathbf{x}'), \tag{4}$$

where $\hat{\mathbf{H}}$ denotes the fused heatmap and N is the number of camera views which contribute to the fusion of current view. The parameter λ balances the responses in the current and other views.

3.3 Side Effect and Solution

One side effect caused by the simplified fusion model [i.e. Eq. (4)] is that some background locations may be enhanced undesirably. We visualize an example in the second row of Fig. 5. We can see that many background pixels, for example $\mathbf{x}$, have non-zero responses which are caused by fusion. This phenomenon happens because multiple epipolar lines (in other views) may pass the ground truth joint location which has large responses, and some of the epipolar lines actually correspond to background pixels in the current view. This is explained in Fig. 5. For a location $\mathbf{x}$ in the current view,

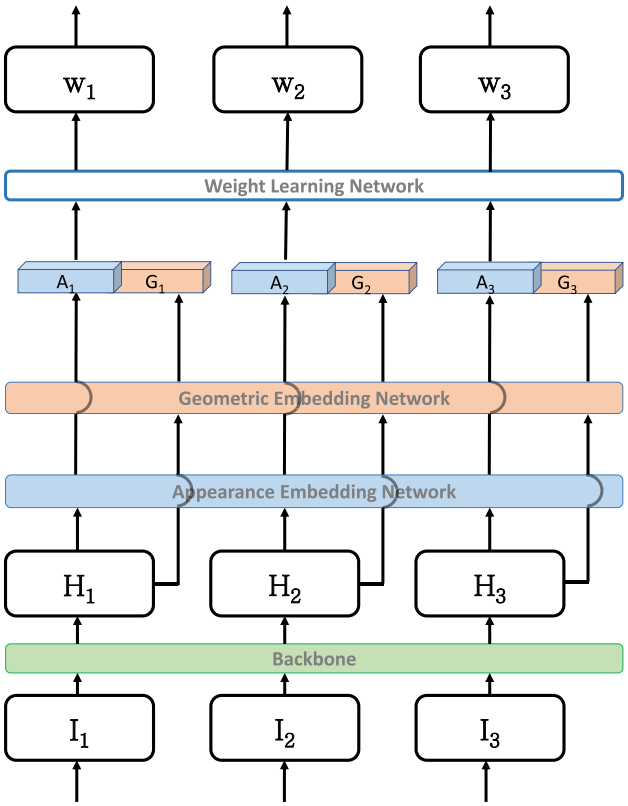

Fig. 6 Network for learning adaptive fusion weights. The backbone network for pose estimation is used to extract heatmaps $\mathbf{H}_v$ for each view $\mathbf{I}_v$. The heatmaps are fed to *appearance embedding network* and *geometry embedding network*, respectively, to extract features, which are concatenated and fed to a *weight learning network* to learn the fusion weights which reflect the heatmap quality in each view. The weights are used for multiview fusion

the corresponding epipolar lines in the other three views are drawn in the first row. We can see that although $\mathbf{x}$ is not at a meaningful joint location, the epipolar line in the first view passes the ground truth knee joint and leads to a large unexpected response for $\mathbf{x}$.

Fortunately, there are patterns for the background pixels that could be undesirably impacted. In general, the pixels that are impacted by a high response location in another view are guaranteed to lie on the same line. More importantly, the lines that correspond to different views do not overlap. It means, for a location $\mathbf{x}$ in the background, its response can only be enhanced by at most one view. In contrast, the location which corresponds to meaningful body joints will be enhanced by multiple views. In other words, the correct location is guaranteed to have the largest response for general cases. So we take advantage of this observation and directly apply the SoftMax operator to remove the small responses. See the third row in Fig. 5 for the effect. We can see that only the large responses around the joint location are preserved.

3.4 Implementation Details

It is worth noting that the above fusion method does not have learnable parameters. So we only need to train the backbone network such as SimpleBaseline (Xiao et al. 2018) to estimate pose heatmaps. The loss function for training the backbone network is defined as MSE loss between the estimated heatmaps and ground truth heatmaps. In the testing stage, given the heatmaps estimated by SimpleBaseline, we fuse them deterministically by our approach.

4 Adaptive Weight for Multiview Fusion

The fusion strategy introduced in the previous section treats all views evenly without considering the feature quality of each view. Note that the fusion weight is $\frac{1-\lambda}{N}$ for the N views in Eq. (4). However, the strategy is problematic in some cases where the heatmaps of some camera views are incorrect. This is because those features may undesirably mess up the features in good views, leading to a completely incorrect 2D pose estimation results.

To solve this problem, we present a weight learning network to learn an *adaptive weight* for each view to faithfully reflect its heatmap quality. It takes inputs of the heatmaps of N-views extracted by the pose estimation network, and regresses N weights ω^u. Then multiview fusion is rewritten to consider the weights as follows

$$\hat{\mathbf{H}}^v(\mathbf{x}) = \omega^v \mathbf{H}^v(\mathbf{x}) + \sum_{u=1}^{N} \omega^u \max_{\mathbf{x}' \in \mathbf{I}^u(\mathbf{x})} \mathbf{H}^u(\mathbf{x}'), \tag{5}$$

The prediction of the adaptive fusion weight ω is implemented by a lightweight neural network as shown in Fig. 6. On top of the heatmaps $\mathbf{H}$ provided by the pose estimation network, we extract two types of information for making the prediction. The first is the appearance embedding which extracts information such as the distribution characteristics of the heatmaps. The second is the geometry embedding which considers the cross-view location consistency. The two terms are complementary to each other. The proposed weight learning network can be joined with the pose estimation network for end-to-end training without enforcing supervision on the weights.

4.1 The Appearance Embedding

The heatmap of each joint actually contains rich information to infer its heatmap quality. For example, if the predicted heatmap has a desired shape of Gaussian kernel, then in many cases, the heatmap quality is good. In contrast, if the predicted heatmap has random and small responses all over the space

Fig. 7 The appearance embedding network for predicting the fusion weight. i is the index of camera views. The parameters in the network are shared for all views and joints. See also Fig. 6 for how the appearance embedding A_i is used for determining the fusion weight

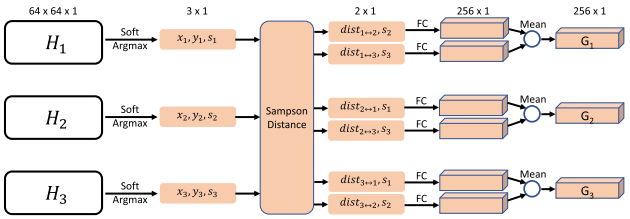

Fig. 8 The geometry embedding network for predicting the fusion weight. For each joint in each camera view (three views are shown in this example), it generates a 256-dimensional embedding to reflect the heatmap (pose) quality. Note that the FC is shared for all branches

(for example, when the joint is occluded), then the quality is likely to be bad.

We propose a simple network to extract appearance embeddings for each joint in each camera view. Figure 7 shows the network structure. Starting from the heatmaps $\mathbf{H}_i$, we apply a convolutional layer to extract features. Then the features are down-sampled by average pooling and fed to a Fully Connected (FC) layer for extracting the appearance embeddings. Different joint types and camera views share the same weights. We only show the network for a single view and a single joint for simplicity. The appearance embedding network is jointly learned end-to-end with the pose estimation network.

4.2 The Geometry Embedding

The appearance embedding alone is not sufficient for some challenging cases where the heatmaps have the desired shape of Gaussian kernel but at the wrong locations. One such example is when the left knee is detected at the location of right knee which is usually known as the "double counting" problem to the community. To solve this problem, we propose to leverage the location consistency information among all camera views. Our core motivation is that the predicted joint location in one camera view is more reliable if it agrees with the locations in other views.

We implement this idea by a geometry embedding network as shown in Fig. 8. Starting from the heatmaps $\mathbf{H}$, we first apply the "soft-argmax" operator (Sun et al. 2018) to obtain the location (x, y) of the joint in each view. We also get the heatmap response value s in that location to reflect its confidence. Then we compute the Sampson distance (Hartley and Zisserman 2003) $dist_{i\leftrightarrow j}$ between the current view and other views to measure the correspondence or consistency

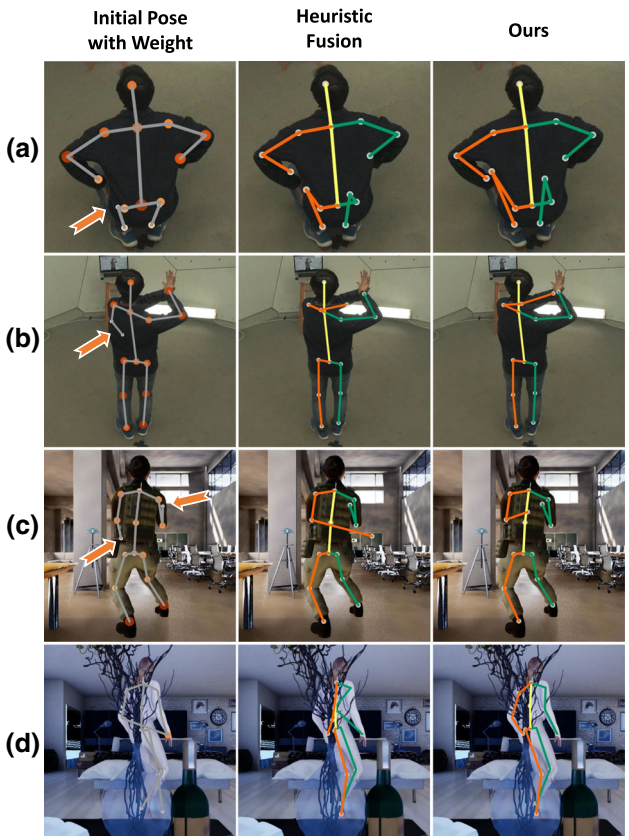

Fig. 9 We visualize the predicted fusion weights by the size of the markers in the first column. A large marker denotes a larger weight. The rest two columns show the poses estimated by *HeuristicFuse* and *AdaFuse*, respectively. Our *AdaFuse* has clearly better estimations due to the consideration of the feature quality in every view

error. A small $dist_{i\leftrightarrow j}$ means the joint locations in the two views are consistent. Intuitively, the location that is consistent with most views is more reliable. Finally, we propose to use a FC layer to embed the Sampson distance into a feature vector. The feature vectors of all camera pairs are then averaged to obtain the final geometry embedding.

4.3 Weight Learning Network

We propose a simple network consisting of three FC layers to transform the concatenated appearance and geometric embeddings to regress the final weight. It is worth noting that we do not train the weight learning network independently. Instead, we join it with the pose estimation network to minimize the fused 2D heatmap loss without enforcing intermediate supervision on the fusion weights. The first column in Fig. 9 shows some example weights predicted by our approach. We can see that when the joints are occluded, and are localized at incorrect locations, the corresponding fusion weights are indeed smaller than other joints.

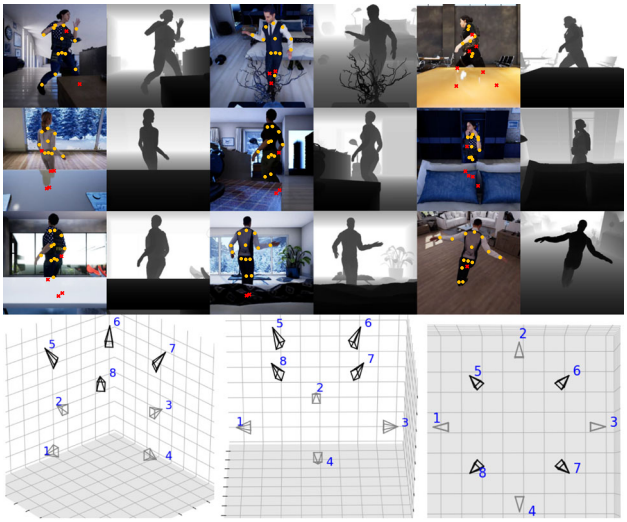

Fig. 10 We show some typical images, ground-truth 2D joint locations and the depth maps from the *Occlusion-Person* dataset. The joint represented by red "x" means it is occluded. The bottom row shows spacial configuration of the eight cameras used in the dataset from different view angles

Table 1 The statistics of the public multiview pose estimation datasets

Dataset	Frames	Cameras	Occluded joints
Human3.6M	784k	4	–
Total Capture	236k	8	–
Panoptic	36k	31	–
Occlusion-Person	73k	8	20.3%

Only the *Occlusion-Person* dataset provides occlusion labels

5 Datasets and Metrics

We introduce the three datasets used for evaluation and the corresponding metrics. We also describe how we construct the synthetic person dataset *Occlusion-Person* which has a large amount of human-object occlusion.

5.1 Datasets

The Human3.6M Dataset (Ionescu et al. 2014) It provides synchronized images captured by four cameras. There are seven subjects performing daily actions. We use a cross-subject evaluation scheme where subjects 1, 5, 6, 7, 8 are used for training and 9, 11 for testing. We also use the MPII dataset (Andriluka et al. 2014) to augment the training data to avoid over-fitting to the simple background. Since the MPII dataset provides only monocular images, we only train the backbone network before multiview fusion.

The Total Capture Dataset (Trumble et al. 2017) It provides synchronized person images captured by eight cameras. Following the dataset convention, the training set consists of "ROM1,2,3", "Freestyle1,2", "Walking1,3", "Acting1,2" and "Running1" on subjects 1, 2 and 3. The testing set consists of "Freestyle3 (**FS3**)", "Acting3 (**A3**)" and "Walking2 (**W2**)" on subjects 1,2,3,4 and 5.

The CMU Panoptic Dataset (Joo et al. 2019) This recently introduced dataset provides images captured by dozens of cameras. We uniformly select six cameras to evaluate the impact of the number of cameras on 3D pose estimation. In particular, the cameras 1, 2, and 10 are firstly selected to construct a 3-view experiment setting. Then the cameras 13, 3 and 23 are sequentially added to the previous three cameras to construct a four, five and six view experiment setting, respectively. We follow the practice of the previous work (Xiang et al. 2019) to select the training and testing sequences which consist of only one person. Since few works have reported numerical results on this dataset, we only compare our approach to the baselines.

The Occlusion-Person Dataset The previous benchmarks do not provide occlusion labels for the joints in images which prevents us from performing numerical evaluation on the occluded joints. In addition, the amount of occlusion in the benchmarks is limited. To address the limitations, we propose to construct this synthetic dataset *Occlusion-Person*. We adopt UnrealCV (Qiu et al. 2017) to render multiview images and depth maps from 3D models. In particular, thirteen human models of different clothes are put into nine different scenes such as living rooms, bedrooms and offices. The human models are driven by the poses selected from the CMU Motion Capture database. We purposely use objects such as sofas and desks to occlude some body joints. Eight cameras are placed in each scene to render the multiview images and the depth maps. The eight cameras are placed evenly every 45 degree on a circle of two meters radius at about 0.9 and 2.3 meters high, respectively. We provide the 3D locations of 15 joints as ground truth. Figure 10 shows some sample images from the dataset and spacial configuration of the cameras.

The occlusion label for each joint in an image is obtained by comparing its depth value (available in the depth map), to the depth of the 3D joint in the camera coordinate system. If the difference between the two depth values is smaller than 30cm, then the joint is not occluded. Otherwise, it is occluded. Table 1 compares this dataset to the existing benchmarks. In particular, about 20% of the body joints are occluded in our dataset. We use 75% of the dataset for training and 25% for validation.

5.2 Metrics

2D Metrics The Percentage of Correct Keypoints (PCK) metric introduced in Andriluka et al. (2014) is commonly used for 2D pose evaluation. PCKh@t measures the percentage of the estimated joints whose distance from the ground-truth

Table 2 The 2D pose estimation accuracy (PCKh@t) of the baseline methods and our approach on the Human3.6M dataset

Methods	Root	Belly	Neck	Nose	Head	Hip	Knee	Ankle	Shlder	Elbow	Wrist	Mean
NoFuse	95.8	77.1	60.4	86.4	86.2	79.3	81.5	58.6	65.1	78.3	70.1	74.8
HeuristicFuse	96.0	79.3	60.7	**88.4**	**86.8**	83.1	84.5	60.0	**66.9**	82.1	75.2	77.3
ScoreFuse	96.2	79.3	61.6	88.3	86.2	83.3	84.3	60.5	66.6	83.1	77.4	77.8
AdaFuse (Ours)	**96.2**	**79.3**	**61.6**	88.3	86.3	**83.5**	**86.4**	**61.1**	66.7	**86.0**	**80.1**	**78.8**

The best result for each column is highlighted in bold
We report results for each individual joint and the average over all joints

Table 3 The 3D pose estimation error (mm) of the baseline methods and our approach on the Human3.6M dataset

Methods	Belly	Neck	Nose	Head	Hip	Knee	Ankle	Shlder	Elbow	Wrist	Mean
NoFuse	21.6	16.8	15.7	11.3	**17.8**	25.8	35.8	22.0	26.8	34.1	22.9
HeuristicFuse	21.6	16.8	15.7	11.0	17.9	23.0	32.7	21.9	25.0	25.7	21.0
ScoreFuse	21.4	16.7	15.8	10.9	18.3	21.3	30.8	21.8	23.3	23.2	20.1
RANSAC	21.6	16.8	**15.7**	11.2	17.9	23.9	34.6	22.0	25.8	28.2	21.8
AdaFuse (Ours)	**21.3**	**16.7**	15.8	**10.9**	18.3	**20.6**	**30.2**	**21.8**	**21.3**	**21.1**	**19.5**

The best result for each column is highlighted in bold

joints is smaller than t times of the head length. Following the previous works, we report results when t is $\frac{1}{2}$. Since the head length is not provided in the used three benchmarks, we approximately set it to be 2.5% of the human bounding box width for all benchmarks.

3D Metrics The 3D pose estimation accuracy is measured by Mean Per Joint Position Error (MPJPE) between a ground truth 3D pose $y = [p_1^3, \cdots, p_M^3]$ and an estimated 3D pose $\bar{y} = [\bar{p_1^3}, \cdots, \bar{p_M^3}]$: MPJPE $= \frac{1}{M}\sum_{i=1}^{M} \|p_i^3 - \bar{p_i^3}\|_2$ where M is the number of joints in a pose. We do not align the estimated 3D poses to the ground truth by Procrustes. This is referred to as protocol 1 in some works (Martinez et al. 2017; Tome et al. 2018)

6 Experimental Results

We compare our approach to four baselines. The first is *NoFuse* which estimates 2D poses independently for each view without multiview fusion. The second is *HeuristicFuse* which assigns a fixed fusion weight for each view according to Eq. (4). The parameter λ is set to be 0.5 by cross-validation. The third baseline is *ScoreFuse* which uses the same formulation as AdaFuse, i.e. Eq. (5), for feature fusion. It differs from AdaFuse only in the way we compute ω. In particular, ScoreFuse computes ω as the maximum value of the heatmap $\mathbf{H}$. Our approach is denoted as *AdaFuse* which uses the predicted weight for fusion as in Eq. (5). All of the four methods use triangulation (Hartley and Zisserman 2003) to estimate 3D pose from the multiview 2D poses. We also compare to a baseline *RANSAC* which does not perform multiview fusion, but uses RANSAC to remove the outliers in triangulation.

6.1 Results on Human3.6M

2D Pose Estimation Results The 2D pose estimation results are presented in Table 2. All multiview fusion methods remarkably outperform *NoFuse*. The improvement is most significant for the Elbow and Wrist joints because they are frequently occluded by human body. The results demonstrate that multiview fusion is an effective strategy to handle occlusion. *AdaFuse* achieves the highest average accuracy among all fusion methods validating that learning appropriate fusion weights can effectively reduce the negative impact caused by the features of low-quality views.

3D Pose Estimation Results Table 3 shows the 3D pose estimation errors of the baselines and our approach. We can see that *NoFuse* gets an average error of 22.9 mm. This is a very strong baseline whose error is only slightly larger than the state-of-the-arts (see Table 4). On top of this strong baseline, we observe that adding multiview fusion can further reduce the 3D pose estimation errors.

HeuristicFuse gets a smaller error than *NoFuse* which is consistent with the 2D results in Table 2. The mean error only decreases by 1.9 mm because most examples are relatively easy leaving little space for improvement. However, significant improvement is achieved for the challenging joints such as Wrist. The *ScoreFuse* gets a smaller error than *HeuristicFuse*. It means assigning small weights to low-quality views helps improve the quality of the fused heatmaps. Finally, our approach *AdaFuse*, which determines the fusion weight by considering both appearance cues and geometry consistency, notably decreases the average error to 19.5 mm. Considering the baseline is already very strong, the improvement is significant. We notice that AdaFuse achieves slightly worse

results on a small number of joints such as hip and head. This is mainly because these joints are rarely occluded in the datasets so the 2D pose estimator can obtain very accurate estimations for them. Further applying cross view fusion will introduce small noise to heatmaps leading to slightly worse 2D pose estimation accuracy. But when occlusion occurs which is often the case in practice, the benefit brought by cross view fusion will be much more significant than the harm caused by the small noise.

RANSAC is the de facto standard for solving robust estimation problems. As shown in Table 3, it outperforms *NoFuse* by removing some outlier 2D poses in triangulation. However, it is not as effective as the multiview fusion methods because the latter also attempt to refine, in addition to removing, the outlier poses. Another reason is that the number of cameras in this task is small which reduces the chance of finding the true outliers. In addition, we find that *RANSAC* is very sensitive to the threshold used for determining whether a data point is inlier or outlier. In our experiments, we set the threshold by cross validation.

To better understand the improvement brought by *AdaFuse*, we divide the testing samples of the Human3.6M dataset into six groups according to the 3D errors of *NoFuse*. Then we compute the average error for each group. Figure 11 shows the results of various baselines. We can see that *AdaFuse* achieves the most significant improvement when the original error of *NoFuse* is large. However, even when the pose estimations of *NoFuse* are already accurate, *AdaFuse* can still reduce the error slightly.

Ablation Study on Fusion Weights One typical situation where *ScoreFuse* fails is when the pose estimation network generates large scores at *inaccurate* locations. In this case, *AdaFuse* can outperform *ScoreFuse* by leveraging the multiview geometry consistency. To support this conjecture, we visualize some typical heatmaps and the corresponding fusion weights predicted by the two methods, respectively, in Fig. 12. We find that the heatmap responses are large for the four views although the locations are inaccurate for the first and third view. *ScoreFuse* gives large weights for all views which finally leads to a corrupted heatmap. In contrast, *AdaFuse* identifies that the predicted locations in the first and third view are inconsistent with the other two views in spite of their large scores. So it decreases the weights to ensure the good quality of the fused heatmap.

In addition, we also conduct ablation study on *AdaFuse* by using only one of two embedding networks. When we only use either the *appearance embedding* or *geometry embedding*, the 3D errors increase to 20.3 mm and 19.9 mm, respectively. Note that the improvement is actually much larger on those challenging examples. The results validate that the two embeddings are complementary.

Comparison to the State-of-the-arts Table 4 compares our approach to the state-of-the-arts. We can see that our

Table 4 The 3D pose estimation errors (mm) of the state-of-the-arts and our approach on the Human3.6M dataset

Methods	Direct	Disc.	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	*MPJPE*
Trumble et al. (2017)	92.7	85.9	72.3	93.2	86.2	101.2	75.1	78.0	83.5	94.8	85.8	82.0	114.6	94.9	79.7	87.3
Pavlakos et al. (2017)	41.2	49.2	42.8	43.4	55.6	46.9	40.3	63.7	97.6	119.0	52.1	42.7	51.9	41.8	39.4	56.9
Tome et al. (2018)	43.3	49.6	42.0	48.8	51.1	64.3	40.3	43.3	66.0	95.2	50.2	52.2	51.1	43.9	45.3	52.8
Qiu et al. (2019)	24.0	26.7	23.2	24.3	24.8	22.8	24.1	28.6	32.1	26.9	30.9	25.6	25.0	28.0	24.4	26.2
T- Iskakov et al. (2019)	20.4	22.6	20.5	19.7	22.1	20.6	19.5	23.0	25.8	33.0	23.0	21.6	20.7	23.7	21.3	22.6
V- Iskakov et al. (2019)	18.8	20.0	19.3	**18.7**	20.2	19.3	**18.7**	22.3	**23.3**	29.1	21.2	**20.3**	19.3	21.6	19.8	20.8
NoFuse	20.1	22.2	20.2	22.2	23.9	18.2	20.6	25.9	37.0	24.6	22.4	22.5	18.2	22.8	18.5	22.9
AdaFuse (Ours)	**17.8**	**19.5**	**17.6**	20.7	**19.3**	**16.8**	18.9	**20.2**	25.7	**20.1**	**19.2**	20.5	**17.2**	**20.5**	**17.3**	**19.5**

The best result for each column is highlighted in bold
We report results for each of the 15 actions individually and also the average error over all actions. T- Iskakov et al. (2019) means triangulation is used. V- Iskakov et al. (2019) means volumetric method is used

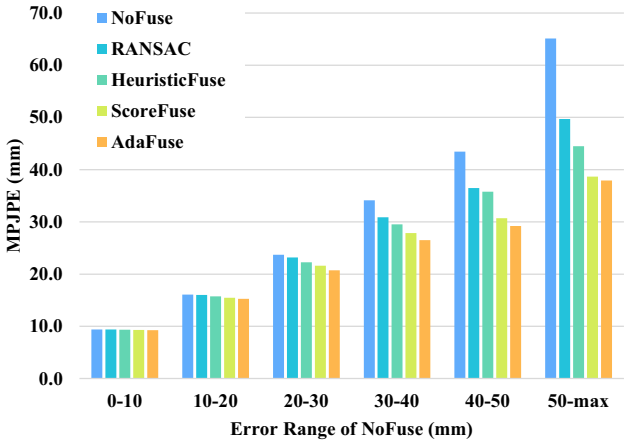

Fig. 11 We divide the test set of Human3.6M into to six groups according to the error of *NoFuse*. We compute the average error for every baseline and every group, respectively

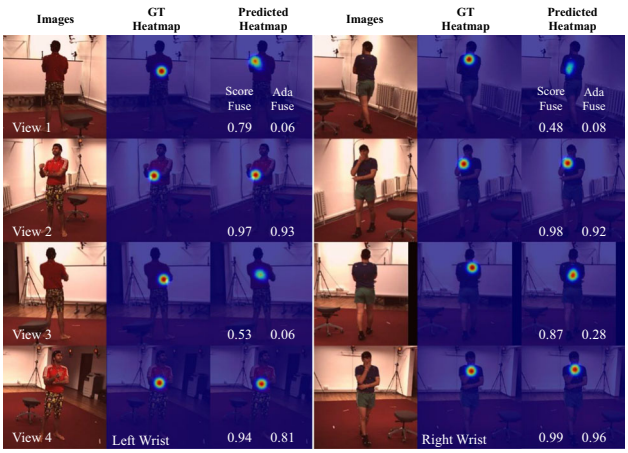

Fig. 12 We visualize the weights predicted by the *ScoreFuse* and *AdaFuse*, respectively. For example, in the first example (left sub-figure), the pose estimation network generates a high response at the wrong location for the first view. Consequently, *ScoreFuse* undesirably gives a large weight. In contrast, *AdaFuse* gives a small weight by identifying that its location are inconsistent with other views

approach outperforms all of them. Note that two approaches, i.e. *Triangulation* and *Volumetric*, are used in Iskakov et al. (2019) to lift 2D poses to 3D. The *Triangulation* approach is more comparable to ours. Our approach *AdaFuse* decreases the error of Iskakov et al. (2019) by about 13%($= \frac{22.6-19.5}{22.6}$). The improvement is significant considering that the error of the state-of-the-art is already very small.

6.2 Results on Panoptic

We evaluate the impact of the number of cameras on this dataset. Figure 13 shows the mean 3D errors when three to six cameras are used, respectively. In general, the error decreases when more cameras are used for most baselines. However, we observe that the error of *NoFuse* actually becomes larger

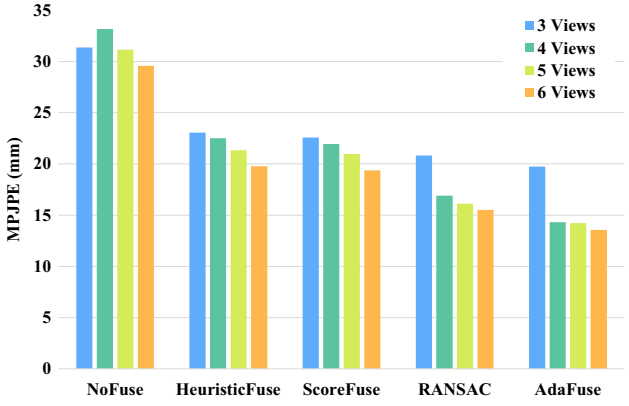

Fig. 13 The 3D pose estimation errors on the Panoptic dataset when different numbers of cameras are used

when the camera number increases from three to four. This undesirable phenomenon happens because the new camera view is very challenging thus the 2D pose estimation results are inaccurate. However, for our approach *AdaFuse*, the negative impact of low-quality heatmaps in individual views is limited due to the adaptive multiview fusion. We can see that the error of *AdaFuse* consistently decreases when the number of cameras increases. Since there is not a commonly adopted evaluation protocol and very few works have reported results on this new dataset, we do not compare our approach to the other approaches.

6.3 Results on Occlusion-Person

2D Pose Estimation Results Table 5 shows the results on the *occluded* joints. Only about 30.9% of the occluded joints can be accurately detected by *NoFuse*. The result is reasonable because the features of the occluded joints are severely corrupted. All of the three multiview fusion methods remarkably improve the accuracy. In particular, more than 90% of the occluded joints are correctly detected by *AdaFuse*. The results demonstrate the advantages of our strategy for learning the fusion weights.

3D Pose Estimation Results We show the 3D pose estimation error (*mm*) for each joint type in Table 6. *NoFuse* results in a large error of 48.1 mm. By improving the 2D pose estimation results on the occluded joints, the 3D errors are also significantly reduced, especially for the joints on the limbs such as Ankles and Wrists. In particular, our approach decreases the 3D error significantly to 12.6 mm.

Impact of Number of Occluded Views We also evaluate the impact of the number of occluded views on this dataset. In particular, we classify each joint into one of five groups according to the number of occluded views, and report the average joint error for each group, respectively. The results are shown in Table 7. We can see that when the joints are visible in all views, the simple baseline *NoFuse* also achieves a

Table 5 The 2D pose estimation accuracy (PCKh@t) of the baselines and our approach for the **occluded joints** on the *Occlusion-Person* dataset

Methods	Hip	Knee	Ankle	Shlder	Elbow	Wrist	Avg
NoFuse	63.4	21.5	17.0	29.5	14.6	12.4	30.9
HeuristicFuse	76.9	59.0	73.4	63.5	49.0	54.8	65.0
ScoreFuse	90.9	88.6	88.1	86.0	93.2	86.8	89.8
AdaFuse	**96.5**	**96.0**	**92.5**	**94.1**	**98.3**	**93.2**	**95.5**

The best result for each column is highlighted in bold
We report results for each joint type individually, and also the average accuracy over all joint types

Table 6 The 3D pose estimation error (*mm*) of the baselines and our approach on the *Occlusion-Person* dataset

Occluded (%)	Root 14.3%	Belly 13.7%	Neck 7.6%	Hip 23.0%	Knee 25.0%	Ankle 23.5%	Shlder 16.8%	Elbow 25.3%	Wrist 21.7%	Mean
NoFuse	10.0	12.2	12.5	16.8	61.1	113.9	28.0	63.7	60.3	48.1
HeuristicFuse	8.8	10.7	**11.5**	14.2	21.1	19.2	17.5	23.6	24.1	18.0
ScoreFuse	8.4	12.6	12.6	14.7	17.5	17.1	16.1	13.2	16.9	15.0
RANSAC	8.6	11.2	11.7	12.9	18.8	17.9	17.1	14.5	19.7	15.5
AdaFuse (Ours)	**7.2**	**10.6**	**11.6**	**11.7**	**13.8**	**15.7**	**14.2**	**9.9**	**14.4**	**12.6**

The best result for each column is highlighted in bold
We report the result on each joint individually and also the average over all joints. The second row shows the percentage of the joints that are occluded for each joint type

Table 7 The 3D pose estimation error (*mm*) of the baseline methods and our approach on the Occlusion-Person dataset

Occluded Views Percentage	4 2%	3 15%	2 38%	1 35%	0 10%
NoFuse	82.6	70.2	59.7	33.7	13.0
HeuristicFuse	30.5	19.9	15.9	13.5	11.1
ScoreFuse	25.0	18.1	15.2	13.4	12.6
RANSAC	36.5	24.5	19.4	14.3	11.7
AdaFuse (Ours)	**21.7**	**14.8**	**12.5**	**11.5**	**10.8**

The best result for each column is highlighted in bold
We group the the 3D joints by number of occluded views (8 views in all). We show each group's joint number percentage in the second row

very small error of 13.0 mm. However, the error increases dramatically to 82.6 mm when four views are occluded. Recall that there are eight views in total for this dataset. In contrast, the multiview fusion methods, especially our *AdaFuse*, achieves consistently smaller errors than *NoFuse*. More importantly, the error increase is much slower than *NoFuse* when more camera views are occluded which validates the robustness of our approach to occlusion.

Generalization Power The only learnable parameters in our fusion approach are in the appearance embedding and geometry embedding networks. In this section, we evaluate whether the *AdaFuse* weight prediction network learned on *Occlusion-Person* can be directly applied to the other datasets. In particular, we append the *AdaFuse* weight prediction network learned on *Occlusion-Person* to the 2D pose estimators trained on each dataset itself as the final model for evaluation. Table 8 shows the 3D pose estimation results on various datasets. We find that the fusion network learned on the synthetic *Occlusion-Person* dataset achieves similar performance on the three realistic datasets compared to the networks learned on each of the target dataset, respectively. The promising results validate that the fusion model

Table 8 The 3D pose estimation errors MPJPE (*mm*) when *AdaFuse* weight prediction network is trained on *Occlusion-Person* or directly trained on the Evaluation dataset, respectively

Evaluation Dataset	AdaFuse Trained on		NoFuse	HeuristicFuse	ScoreFuse	RANSAC
	Evaluation Dataset	Occlusion-Person				
Human3.6M	19.5	19.4	22.9	21.0	20.1	21.8
Panoptic 4 views	14.7	14.6	33.2	22.5	21.9	16.9
Panoptic 6 views	13.6	13.9	29.6	19.8	19.4	15.5
Total Capture	19.2	20.1	29.4	20.0	20.5	20.5

The 2D pose estimators for generating the initial heatmaps are trained on each Evaluation dataset separately

Table 9 The 3D pose estimation errors MPJPE (*mm*) of different methods on the Total Capture dataset

Methods	IMUs	Temporal	Subjects(S1,2,3)			Subjects(S4,5)			Mean
			W2	A3	FS3	W2	A3	FS3	
(Trumble et al. 2017)	✓	✓	48.3	94.3	122.3	84.3	154.5	168.5	107.3
(Wei et al. 2016)			79.0	106.5	112.1	79.0	73.7	149.3	99.8
(Gilbert et al. 2019)	✓		19.2	42.3	48.8	24.7	58.8	61.8	42.6
(Trumble et al. 2018)		✓	13.0	23.0	47.0	**21.8**	40.9	68.5	34.1
(Qiu et al. 2019)			19	21	28	32	33	54	29
NoFuse			15.9	18.5	29.9	33.9	33.8	60.0	29.4
HeuristicFuse			7.8	11.6	19.6	23.3	26.9	44.8	20.0
ScoreFuse			9.7	13.1	19.9	23.9	27.2	41.4	20.5
RANSAC			8.4	11.6	20.5	23.3	27.2	45.7	20.5
AdaFuse (Ours)			**7.2**	**10.8**	**18.5**	22.8	**26.6**	**42.9**	**19.2**

The best result for each column is highlighted in bold

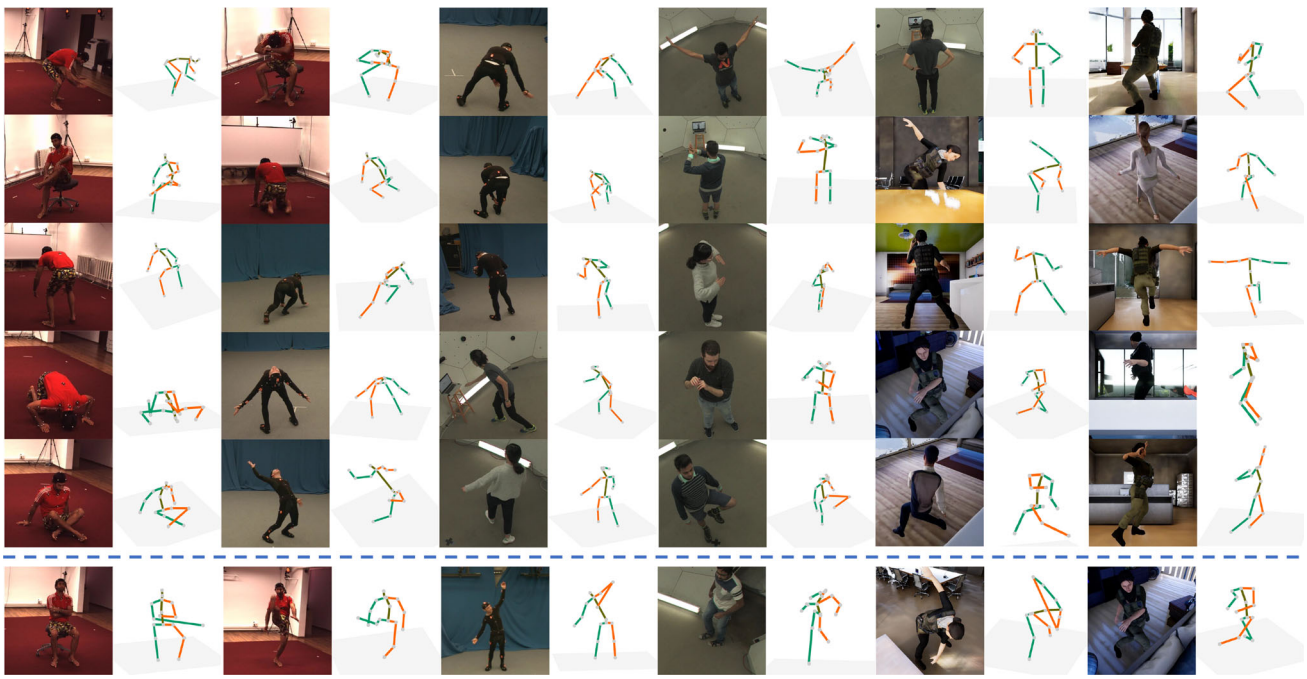

Fig. 14 We demonstrate some 3D pose estimation examples obtained by *AdaFuse*. The last row shows some failure cases

has strong generalization power. It is also worth noting that our approach can naturally handle different numbers of cameras for two reasons. First, the parameters in the appearance embedding network and the geometry embedding network are shared for all camera views. Second, the "Mean" operator in the geometry embedding network makes it independent of the number of views as shown in Figure 7 and Figure 8. In summary, *AdaFuse* is ready to be deployed in new environments of different camera poses without additional adaptation.

6.4 Results on Total Capture

We report the 3D pose estimation results on the Total Capture dataset in Table 9. It is worth noting that some methods also use IMUs in addition to the multiview images. We can see that our approach outperforms all of the previous methods. We notice that the error of our approach is slightly larger than LSTM-AE (Trumble et al. 2018) for the "W2 (walking)" action of S4,5. We tend to think it is because LSTM can get significant benefits when it is applied to periodic actions such as "walking". This is also observed independently in another work (Gilbert et al. 2019).

We show some 3D pose estimation examples in Fig. 14. In most cases, our approach can accurately estimate the 3D

poses. One typical situation where the approach fails is when 2D pose estimation results are inaccurate for many camera views. For example in the Panoptic dataset, when human begin to enter the dome, they may be occluded in multiple views. In this case, the heatmaps in each view are of low-quality. Therefore the fused heatmaps will also have degraded quality, leading to inaccurate 2D pose estimations.

7 Summary and Future Work

We present a multiview fusion approach *AdaFuse* to handle the occlusion problem in human pose estimation. *AdaFuse* has practical values in that it is very simple and can be flexibly applied to new environments without additional adaptation. In addition, it can be combined with any 2D pose estimation networks. We extensively evaluate the effectiveness of the approach on three benchmark datasets. The approach outperforms the state-of-the-arts remarkably. We also construct a large scale human dataset which has severe occlusion to promote more research along this direction. Our next step of work is to leverage temporal information to further improve the pose estimation accuracy.

References

Amin, S., Andriluka, M., Rohrbach, M., & Schiele, B. (2013). Multi-view pictorial structures for 3D human pose estimation. In *BMVC*.

Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR* (pp. 3686–3693).

Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., & Ilic, S. (2014). 3d pictorial structures for multiple human pose estimation. In *CVPR* (pp. 1669–1676).

Bo, L., & Sminchisescu, C. (2010). Twin gaussian processes for structured prediction. *IJCV*, *87*(1–2), 28.

Bridgeman, L., Volino, M., Guillemaut, J. Y., & Hilton, A. (2019). Multi-person 3d pose estimation and tracking in sports. In *CVPRW*.

Burenius, M., Sullivan, J., & Carlsson, S. (2013). 3D pictorial structures for multiple view articulated pose estimation. In *CVPR* (pp. 3618–3625).

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR* (pp. 7291–7299).

Chen, W., Wang, H., Li, Y., Su, H., Wang, Z., Tu, C., et al. (2016). Synthesizing training images for boosting human 3d pose estimation. In *3DV* (pp. 479–488). IEEE.

Cheng, Y., Yang, B., Wang, B., Yan, W., & Tan, R. T. (2019). Occlusion-aware networks for 3d human pose estimation in video. In *ICCV* (pp. 723–732).

Ci, H., Wang, C., Ma, X., & Wang, Y. (2019). Optimizing network structure for 3d human pose estimation. In *ICCV* (pp. 915–922).

Ci, H., Ma, X., Wang, C., & Wang, Y. (2020). Locally connected network for monocular 3d human pose estimation. In *T-PAMI*.

Dong, J., Jiang, W., Huang, Q., Bao, H., & Zhou, X. (2019). Fast and robust multi-person 3d pose estimation from multiple views. In *CVPR* (pp. 7792–7801).

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Gal, Y. (2016). Uncertainty in deep learning. PhD thesis, PhD thesis, University of Cambridge.

Gal, Y., & Ghahramani, Z. (2015). Dropout as a Bayesian approximation: Insights and applications. In *Deep learning workshop* (Vol. 1, p. 2). ICML.

Gall, J., Rosenhahn, B., Brox, T., & Seidel, H. P. (2010). Optimization and filtering for human motion capture. *IJCV*, *87*(1–2), 75.

Ghahramani, Z. (2016). A history of Bayesian neural networks. In *NIPS workshop on Bayesian deep learning*.

Gilbert, A., Trumble, M., Malleson, C., Hilton, A., & Collomosse, J. (2019). Fusing visual and inertial sensors with semantics for 3d human pose estimation. *IJCV*, *127*(4), 381–397.

Guo, C., Pleiss, G., Sun, Y., Weinberger, K. Q. (2017). On calibration of modern neural networks. In *ICML* (pp. 1321–1330), JMLR.org .

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

He, Y., Zhu, C., Wang, J., Savvides, M., & Zhang, X. (2019). Bounding box regression with uncertainty for accurate object detection. In *CVPR* (pp. 2888–2897).

Hoffmann, D. T., Tzionas, D., Black, M. J., & Tang, S. (2019). Learning to train with synthetic humans. In *German conference on pattern recognition* (pp. 609–623). Springer.

Ilg, E., Cicek, O., Galesso, S., Klein, A., Makansi, O., Hutter, F., et al. (2018). Uncertainty estimates and multi-hypotheses networks for optical flow. In *ECCV* (pp. 652–667).

Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3. 6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*, 1325–1339.

Iskakov, K., Burkov, E., Lempitsky, V., & Malkov, Y. (2019). Learnable triangulation of human pose. arXiv preprint arXiv:1905.05754.

Joo, H., Simon, T., Li, X., Liu, H., Tan, L., Gui, L., et al. (2019). Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(1), 190–204.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS* (pp. 5574–5584).

Kreiss, S., Bertoni, L., & Alahi, A. (2019). Pifpaf: Composite fields for human pose estimation. In *CVPR* (pp. 11977–11986).

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS* (pp. 6402–6413).

Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., & Gehler, P. V. (2017). Unite the people: Closing the loop between 3d and 2d human representations. In *CVPR* (pp. 6050–6059).

Li, T., Fan, L., Zhao, M., Liu, Y., & Katabi, D. (2019). Making the invisible visible: Action recognition through walls and occlusions. In *ICCV* (pp. 872–881).

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft coco: Common objects in context. In *ECCV* (pp. 740–755). Springer.

Liu, Y., Stoll, C., Gall, J., Seidel, H. P., & Theobalt, C. (2011). Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR* (pp. 1249–1256). IEEE.

Malleson, C., Gilbert, A., Trumble, M., Collomosse, J., Hilton, A., & Volino, M. (2017). Real-time full-body motion capture from video and imus. In *3DV* (pp. 449–457). IEEE.

von Marcard, T., Henschel, R., Black, MJ., Rosenhahn, B., & Pons-Moll, G. (2018). Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV* (pp. 601–617).

Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017). A simple yet effective baseline for 3D human pose estimation. In *ICCV* (p. 5).

Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, *104*(2–3), 90–126.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *ECCV* (pp. 483–499). Springer.

Pavlakos, G., Zhou, X., Derpanis, K. G., & Daniilidis, K. (2017). Harvesting multiple views for marker-less 3D human pose annotations. In: CVPR (pp. 1253–1262).

Pavlakos, G., Zhou, X., & Daniilidis, K. (2018). Ordinal depth supervision for 3d human pose estimation. In *CVPR* (pp. 7307–7316).

Pavllo, D., Feichtenhofer, C., Grangier, D., & Auli, M. (2019). 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR* (pp. 7753–7762).

Peng, X., Tang, Z., Yang, F., Feris, R. S., & Metaxas, D. (2018). Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *CVPR* (pp. 2226–2234).

Perez, P., Vermaak, J., & Blake, A. (2004). Data fusion for visual tracking with particles. *Proceedings of the IEEE*, *92*(3), 495–513.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., & Weinberger, K. Q. (2017). On fairness and calibration. In *NIPS* (pp. 5680–5689).

Qiu, H., Wang, C., Wang, J., Wang, N., & Zeng, W. (2019). Cross view fusion for 3d human pose estimation. In *ICCV* (pp. 4342–4351).

Qiu, W., Zhong, F., Zhang, Y., Qiao, S., Xiao, Z., Kim, T. S., et al. (2017). Unrealcv: Virtual worlds for computer vision. In *Proceedings of the 25th ACM international conference on multimedia* (pp. 1221–1224).ACM.

Rhodin, H., Spörri, J., Katircioglu, I., Constantin, V., Meyer, F., Müller, E., Salzmann, M., et al. (2018). Learning monocular 3d human pose estimation from multi-view images. In *CVPR* (pp. 8437–8446).

Roetenberg, D., Luinge, H., & Slycke, P. (2009). *Xsens mvn: full 6dof human motion tracking using miniature inertial sensors*. Xsens Motion Technologies BV, Tech Rep 1.

Rogez, G., Schmid, C. (2016). Mocap-guided data augmentation for 3d pose estimation in the wild. In *NIPS* (pp. 3108–3116).

Sigal, L., Balan, A. O., & Black, M. J. (2010). Humaneva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, *87*(1–2), 4.

Starner, T., Leibe, B., Minnen, D., Westyn, T., Hurst, A., & Weeks, J. (2003). The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3d reconstruction for augmented desks. *Machine Vision and Applications*, *14*(1), 59–71.

Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In *CVPR* (pp. 5693–5703).

Sun, X., Xiao, B., Wei, F., Liang, S., & Wei, Y. (2018). Integral human pose regression. In *ECCV* (pp. 529–545).

Tome, D., Toso, M., Agapito, L., & Russell, C. (2018). Rethinking pose in 3D: Multi-stage refinement and recovery for markerless motion capture. In *3DV* (pp. 474–483).

Trumble, M., Gilbert, A., Malleson, C., Hilton, A., & Collomosse, J. (2017). Total capture: 3D human pose estimation fusing video and inertial sensors. In *BMVC* (pp. 1–13).

Trumble, M., Gilbert, A., Hilton, A., & Collomosse, J. (2018). Deep autoencoder for combined human pose estimation and body model upscaling. In *ECCV* (pp. 784–800).

Tu, H., Wang, C., & Zeng, W. (2020). Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *ECCV* (pp. 1–16).

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, MJ., Laptev, I., et al. (2017). Learning from synthetic humans. In *CVPR* (pp. 109–117).

Wei, S. E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In *CVPR* (pp. 4724–4732).

Xiang, D., Joo, H., & Sheikh, Y. (2019). Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*.

Xiao, B., Wu, H., & Wei, Y. (2018). Simple baselines for human pose estimation and tracking. In *ECCV* (pp. 466–481).

Xie, R., Wang, C., & Wang, C. (2020). Metafuse: A pre-trained fusion model for human pose estimation. In *CVPR*.

Yang, W., Ouyang, W., Wang, X., Ren, J., Li, H., & Wang, X. (2018). 3d human pose estimation in the wild by adversarial learning. In *CVPR* (pp. 5255–5264).

Zafar, U., Ghafoor, M., Zia, T., Ahmed, G., Latif, A., Malik, K. R., et al. (2019). Face recognition with Bayesian convolutional networks for robust surveillance systems. *EURASIP Journal on Image and Video Processing*, *1*, 10.

Zhang, Z., Wang, C., Qin, W., & Zeng, W. (2020). Fusing wearable imus with multi-view images for human pose estimation: A geometric approach. In *CVPR* (pp. 2200–2209).

Zhao, M., Li, T., Abu Alsheikh, M., Tian, Y., Zhao, H., Torralba, A., et al. (2018). Through-wall human pose estimation using radio signals. In *CVPR* (pp. 7356–7365).

Zhao, M., Liu, Y., Raghu, A., Li, T., Zhao, H., Torralba, A., et al. (2019). Through-wall human mesh recovery using radio signals. In *ICCV* (pp. 10113–10122).

Zhou, X., Huang, Q., Sun, X., Xue, X., & Wei, Y. (2017). Towards 3D human pose estimation in the wild: A weakly-supervised approach. In *ICCV* (pp. 398–407).