Learnable Triangulation of Human Pose

Karim Iskakov [1]   Egor Burkov [1,2]   Victor Lempitsky [1,2]   Yury Malkov [1]

[1]Samsung AI Center, Moscow   [2]Skolkovo Institute of Science and Technology, Moscow

{k.iskakov, e.burkov, v.lempitsky, y.malkov}@samsung.com

Abstract

We present two novel solutions for multi-view 3D human pose estimation based on new learnable triangulation methods that combine 3D information from multiple 2D views. The first (baseline) solution is a basic differentiable algebraic triangulation with an addition of confidence weights estimated from the input images. The second solution is based on a novel method of volumetric aggregation from intermediate 2D backbone feature maps. The aggregated volume is then refined via 3D convolutions that produce final 3D joint heatmaps and allow implicit modelling a human pose prior. Crucially, both approaches are end-to-end differentiable, which allows us to directly optimize the target metric. We demonstrate transferability of the solutions across datasets and considerably improve the multi-view state of the art on the Human3.6M dataset. Video demonstration, annotations and additional materials will be posted on our project page[1].

1. Introduction

3D human pose estimation is one of the fundamental problems in computer vision, with applications in sports, action recognition, computer-assisted living, human-computer interfaces, special effects, and telepresence. To date, most of the efforts in the community are focused on *monocular* 3D pose estimation. Despite a lot of recent progress, the problem of in-the-wild monocular 3D human pose estimation is far from being solved. Here, we consider a simpler yet still challenging problem of multi-view 3D human pose estimation.

There are at least two reasons why multi-view human pose estimation is interesting. First, multi-view pose estimation is arguably the best way to obtain ground truth for monocular 3D pose estimation [5, 27] in-the-wild. This is because the competing techniques such as marker-based motion capture [13] and visual-inertial methods [24] have

certain limitations such as inability to capture rich pose representations (e.g. to estimate hands pose and face pose alongside limb pose) as well as various clothing limitations. The downside is, previous works that used multi-view triangulation for constructing datasets relied on excessive, almost impractical number of views to get the 3D ground truth of sufficient quality [5, 27]. This makes the collection of new in-the-wild datasets for 3D pose estimation very challenging and calls for the reduction of the number of views needed for accurate triangulation.

The second motivation to study multi-view human pose estimation is that, in some cases, it can be used directly to track human pose in real-time for the practical end purpose. This is because multi-camera setups are becoming progressively available in the context of various applications, such as sports or computer-assisted living. Such practical multi-view setups rarely go beyond having just a few views. At the same time, in such a regime, modern multi-view methods have accuracy comparable to well-developed monocular methods [22, 15, 6, 20, 16]. Thus, improving the accuracy of multi-view pose estimation from few views is an important challenge with direct practical applications.

In this work, we argue that given its importance, the task of multi-view pose estimation has received disproportionately little attention. We propose and investigate two simple and related methods for multi-view human pose estimation. Behind both of them lies the idea of *learnable* triangulation, which allows us to dramatically reduce the number of views needed for accurate estimation of 3D pose. During learning, we either use marker based motion capture ground truth or "meta"-ground truth obtained from the excessive number of views. The methods themselves are as follows: (1) a simpler approach based on algebraic triangulation with learnable camera-joint confidence weights, and (2) a more complex volumetric triangulation approach based on dense geometric aggregation of information from different views that allows implicitly modelling a human pose prior. Crucially, both of the proposed solutions are fully differentiable, which permits end-to-end training.

Below, we review related work in monocular and multi-view human pose estimation, and then discuss the details

---

[1] https://saic-violet.github.io/learnable-triangulation

of the new learnable triangulation methods. In the experimental section, we perform an evaluation on the popular Human3.6M [3] and CMU Panoptic [5] datasets, demonstrating state-of-the-art accuracy of the proposed methods and and their ability of cross-dataset generalization.

2. Related work

Single view 3D pose estimation. Current state-of-the-art solutions for the monocular 3D pose estimation can be divided into two sub-categories. The first category is using high quality 2D pose estimation engines with subsequent separate lifting of the 2D coordinates to 3D via deep neural networks (either fully-connected, convolutional or recurrent). This idea was popularized in [11] and offers several advantages: it is simple, fast, can be trained on motion capture data (with skeleton/view augmentations) and allows switching 2D backbones after training. Despite known ambiguities inherent to this family of methods (i.e. orientation of arms' joints in current skeleton models), this paradigm is adopted in the current multi-frame state of the art [16] on the Human3.6M benchmark [3].

The second option is to infer the 3D coordinates directly from the images using convolutional neural networks. The present best solutions use volumetric representations of the pose, with current single-frame state-of-the-art results on Human3.6M [3], namely [20].

Multi-view view 3D pose estimation. Studies of multi-view 3D human pose estimation are generally aimed at getting the ground-truth annotations for the monocular 3D human pose estimation [17, 5, 9]. The work [6] proposed concatenating joints' 2D coordinates from all views into a single batch as an input to a fully connected network that is trained to predict the global 3D joint coordinates. This approach can efficiently use the information from different views and can be trained on motion capture data. However, the method is by design unable to transfer the trained models to new camera setups, while the authors show that the approach is prone to strong over-fitting.

Few works used volumetric pose representation in multi-view setups [15, 5]. Specifically, [5] utilized unprojection of 2D keypoint probability heatmaps (obtained from a pre-trained 2D keypoint detector) to volume with subsequent non-learnable aggregation. Our work differs in two ways. First, we process information inside the volume in a learnable way. Second, we train the network end-to-end, thus adjusting the 2D backbone and alleviating the need for interpretability of the 2D heatmaps. This allows to transfer several self-consistent pose hypotheses from 2D detectors to the volumetric aggregation stage (which was not possible with previous designs).

The work [22] used a multi-stage approach with an external 3D pose prior [21] to infer the 3D pose from 2D joints'

coordinates. During the first stage, images from all views were passed through the backbone convolutional neural network to obtain 2D joints' heatmaps. The positions of maxima in the heatmaps were jointly used to infer the 3D pose via optimizing latent coordinates in 3D pose prior space. In each of the subsequent stages, 3D pose was reprojected back to all camera views and fused with predictions from the previous layer (via a convolutional network). Next, the 3D pose was re-estimated from the positions of heatmap maxima, and the process repeated. Such procedure allowed correcting the predictions of 2D joint heatmaps via indirect holistic reasoning on a human pose. In contrast to our approach, in [22] there is no gradient flow from the 3D predictions to 2D heatmaps and thus no direct signal to correct the prediction of 3D coordinates. Recently, there were proposed few methods that utilized similar unprojection techniques as a step to aggregate the information from multiple views producing state-of-the-art result in different applications. Specifically unprojection was used in [19] for free-point-view rendering, in [23, 7] for volumetric 3D object reconstruction.

3. Method

Our approach assumes we have synchronized video streams from C cameras with known projection matrices P_c capturing performance of a single person in the scene. We aim at estimating the global 3D positions $\mathbf{y}_{j,t}$ of a fixed set of human joints with indices $j \in (1..J)$ at timestamp t. For each timestamp the frames are processed independently (i.e. without using temporal information), thus we omit the index t for clarity.

For each frame, we crop the images using the bounding boxes either estimated by available off-the-shelf 2D human detectors or from ground truth (if provided). Then we feed the cropped images I_c into a deep convolutional neural network backbone based on the "simple baselines" architecture [26].

The convolutional neural network backbone with learnable weights θ consists of a ResNet-152 network (output denoted by g_θ), followed by a series of transposed convolutions that produce intermediate heatmaps (the output denoted by f_θ) and a 1×1 - kernel convolutional neural network that transforms the intermediate heatmaps to interpretable joint heatmaps (output denoted by h_θ; the number of output channels is equal to the number of joints J). In the two following sections we describe two different methods to infer joints' 3D coordinates by aggregating information from multiple views.

Algebraic triangulation approach. In the algebraic triangulation baseline we process each joint j independently of each other. The approach is built upon triangulating the 2D positions obtained from the j-joint's backbone heatmaps
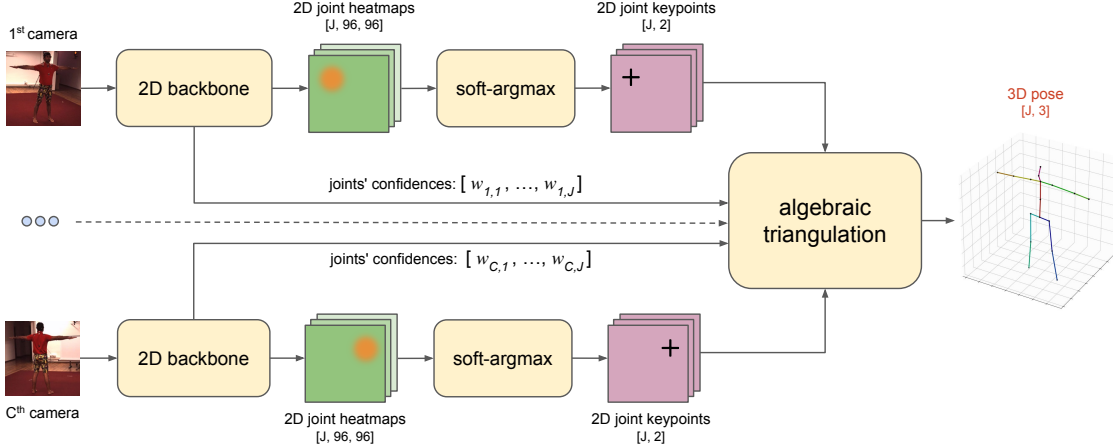
Figure 1. Outline of the approach based on algebraic triangulation with learned confidences. The input for the method is a set of RGB images with known camera parameters. The 2D backbone produces the joints' heatmaps and camera-joint confidences. The 2D positions of the joints are inferred from 2D joint heatmaps by applying soft-argmax. The 2D positions together with the confidences are passed to the algebraic triangulation module that outputs the triangulated 3D pose. All blocks allow backpropagation of the gradients, so the model can be trained end-to-end.

from different views: $H_{c,j} = h_\theta(I_c)_j$ (Figure 1). To estimate the 2D positions we first compute the softmax across the spatial axes:

$$H'_{c,j} = \exp(\alpha H_{c,j}) / \Big(\sum_{r_x=1}^{W} \sum_{r_y=1}^{H} \exp(\alpha H_{c,j}(\mathbf{r})) \Big), \quad (1)$$

where parameter α is discussed below. Then we calculate the 2D positions of the joints as the center of mass of the corresponding heatmaps (so-called soft-argmax operation):

$$\mathbf{x}_{c,j} = \sum_{r_x=1}^{W} \sum_{r_y=1}^{H} \mathbf{r} \cdot \big(H'_{c,j}(\mathbf{r}) \big) \quad (2)$$

An important feature of soft-argmax is that rather than getting the index of the maximum, it allows the gradients to flow back to heatmaps H_c from the output 2D position of the joints $\mathbf{x}$. Since the backbone was pretrained using a loss other than soft-argmax (MSE over heatmaps without softmax [20]), we adjust the heatmaps via multiplying them by an 'inverse temperature' parameter $\alpha = 100$ in (1), so at the start of the training the soft-argmax gives an output close to the positions of the maximum.

To infer the 3D positions of the joints from their 2D estimates $\mathbf{x}_{c,j}$ we use a linear algebraic triangulation approach [1]. The method reduces the finding of the 3D coordinates of a joint $\mathbf{y}_j$ to solving the overdetermined system of equations on homogeneous 3D coordinate vector of the joint $\tilde{\mathbf{y}}$:

$$A_j \tilde{\mathbf{y}}_j = 0, \quad (3)$$

where $A_j \in \mathbb{R}^{(2C,4)}$ is a matrix composed of the components from the full projection matrices and $\mathbf{x}_{c,j}$ (see [1] for full details).

A naïve triangulation algorithm assumes that the joint coordinates from each view are independent of each other and thus all make comparable contributions to the triangulation. However, on some views the 2D position of the joints cannot be estimated reliably (e.g. due to joint occlusions), leading to unnecessary degradation of the final triangulation result. This greatly exacerbates the tendency of methods that optimize algebraic reprojection error to pay uneven attention to different views. The problem can be dealt with by applying RANSAC together with the Huber loss (used to score reprojection errors corresponding to inliers). However, this has its own drawbacks. E.g. using RANSAC may completely cut off the gradient flow to the excluded cameras.

To address the aforementioned problems, we add *learnable* weights w_c to the coefficients of the matrix corresponding to different views:

$$(\mathbf{w}_j \circ A_j) \tilde{\mathbf{y}}_j = 0, \quad (4)$$

where $\mathbf{w}_j = (w_{1,j}, w_{1,j}, w_{2,j}, w_{2,j}, ..., w_{C,j}, w_{C,j})$; $\circ$ denotes the Hadamard product (i.e. i-th row of matrix A is multiplied by i-th element of vector $\mathbf{w}$). The weights $w_{c,j}$ are estimated by a convolutional network q^ϕ with learnable parameters ϕ (comprised of two convolutional layers, global average pooling and three fully-connected layers), applied to the intermediate output of the backbone:

$$w_{c,j} = \big(q^\phi(g^\theta(I_c)) \big)_j \quad (5)$$

This allows the contribution of the each camera view to be controlled by the neural network branch that is learned jointly with the backbone joint detector.

The equation (4) is solved via differentiable Singular Value Decomposition of the matrix $B = UDV^T$, from
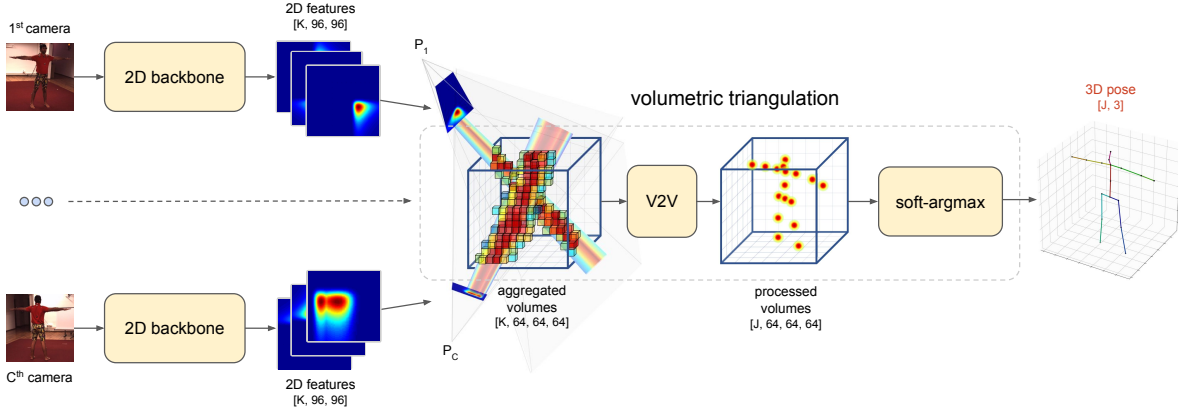
Figure 2. Outline of the approach based on volumetric triangulation. The input for the method is a set of RGB images with known camera parameters. The 2D backbone produces intermediate feature maps that are unprojected into volumes with subsequent aggreagation to a fixed size volume. The volume is passed to a 3D convolutional neural network that outputs the interpretable 3D heatmaps. The output 3D positions of the joints are inferred from 3D joint heatmaps by computing soft-argmax. All blocks allow backpropagation of the gradients, so the model can be trained end-to-end.

which $\tilde{\mathbf{y}}$ is set as the last column of V. The final non-homogeneous value of $\mathbf{y}$ is obtained by dividing the homogeneous 3D coordinate vector $\tilde{\mathbf{y}}$ by its fourth coordinate: $\mathbf{y} = \tilde{\mathbf{y}}/(\tilde{\mathbf{y}})_4$.

Volumetric triangulation approach. The main drawback of the baseline algebraic triangulation approach is that the images I_c from different cameras are processed independently from each other and fuses only sparse information, so there is no easy way to add a 3D human pose prior and no way to filter out the cameras with wrong projection matrices.

To solve this problem we propose to use a more complex and powerful triangulation procedure. We unproject the un-interpretable feature maps produced by the 2D backbone into 3D volumes (see Figure 2). This is done by filling a 3D cube around the person via projecting output of the 2D network along projection rays inside the 3D cube. The cubes obtained from multiple views are then aggregated together and processed. For such volumetric triangulation approach, the 2D output does not have to be interpretable as joint heatmaps, thus, instead of unprojecting H_c themselves, we use the output of a trainable single layer convolutional neural network o^γ with 1×1 kernel and K output channels (the weights of this layer are denoted by γ) applied to the input from the backbone intermediate heatmaps $f^\theta(I_c)$:

$$M_{c,k} = o^\gamma(f^\theta(I_c))_k \qquad (6)$$

To create the volumetric grid, we place a $L \times L \times L$ - sized 3D bound box in the global space around the human pelvis (the position of the pelvis is estimated by the algebraic triangulation baseline described above, L denotes the size of the box in meters) with the Y-axis perpendicular to the ground

and a random orientation of the X-axis. We discretize the bounding box by a volumetric cube $V^{\text{coords}} \in \mathbb{R}^{64,64,64,3}$, filling it with the global coordinates of the center of each voxel (in a similar way to [5]).

For each view, we then project the 3D coordinates in V^{coords} to the plane: $V_c^{\text{proj}} = P_c V^{\text{coords}}$ (note that $V_c^{\text{proj}} \in \mathbb{R}^{64,64,64,2}$) and fill a cube $V_c^{\text{view}} \in \mathbb{R}^{64,64,64,K}$ by bilinear sampling [4] from the maps $M_{c,k}$ of the corresponding camera view using 2D coordinates in V_c^{proj}:

$$V_{c,k}^{\text{view}} = M_{c,k}\{V_c^{\text{proj}}\}, \qquad (7)$$

where $\{\cdot\}$ denotes bilinear sampling. We then aggregate the volumetric maps from all views to form an input to the further processing that does not depend on the number of camera views. We study three diverse methods for the aggregation:

1. Raw summation of the voxel data:

$$V_k^{\text{input}} = \sum_c V_{c,k}^{\text{view}} \qquad (8)$$

2. Summation of the voxel data with normalized confidence multipliers d_c (obtained similarly to w_c using a branch attached to backbone):

$$V_k^{\text{input}} = \sum_c \left(d_c \cdot V_{c,k}^{\text{view}} \right) / \sum_c d_c \qquad (9)$$

3. Calculating a relaxed version of maximum. Here, we first compute the softmax for each individual voxel V_c^{view} across all cameras, producing the volumetric coefficient distribution $V_{c,k}^w$ with the role similar to scalars d_c:

$$V_{c,k}^w = \exp(V_{c,k}^{\text{view}}) / \sum_c \exp(V_{c,k}^{\text{view}}) \qquad (10)$$

Then, the voxel maps from each view are summed with the volumetric coefficients V_c^w:

$$V_k^{\text{input}} = \sum_c V_{c,k}^w \circ V_c^{\text{view}} \tag{11}$$

Aggregated volumetric maps are then fed into a learnable volumetric convolutional neural network u^ν (with weights denoted by ν), with architecture similar to V2V [14], producing the interpretable 3D-heatmaps of the output joints:

$$V_j^{\text{output}} = (u^\nu(V^{\text{input}}))_j \tag{12}$$

Next, we compute softmax of V_j^{output} across the spatial axes (similar to (1)):

$$V_j'^{\text{output}} = \exp(V_j^{\text{output}})/\Big(\sum_{r_x=1}^{W} \sum_{r_y=1}^{H} \sum_{r_z=1}^{D} \exp(V_j^{\text{output}}(\mathbf{r}))\Big), \tag{13}$$

and estimate the center of mass for each of the volumetric joint heatmaps to infer the positions of the joints in 3D:

$$\mathbf{y}_j = \sum_{r_x=1}^{W} \sum_{r_y=1}^{H} \sum_{r_z=1}^{D} \mathbf{r} \cdot V_j'^{\text{output}}(\mathbf{r}) \tag{14}$$

Unprojecting to 3D allows getting more robust results, as the wrong predictions are spatially isolated from the correct ones inside the cube, so they can be discarded by convolutional operations. The network also naturally incorporates the camera parameters (uses them as an input), allows implicit modelling the human pose prior and can reasonably handle multimodality in 2D detections.

Losses. For both of the methods described above, the gradients pass from the output prediction of 3D joints' coordinates $\mathbf{y}_j$ to the input RGB-images I_c making the pipeline trainable end-to-end. For the case of algebraic triangulation, we apply a soft version of per-joint Mean Square Error (MSE) loss to make the training more robust to outliers. This variant leads to better results compared to raw MSE or L1 (mean absolute error):

$$\mathcal{L}_j^{\text{alg}}(\theta, \phi) = \begin{cases} \text{MSE}(\mathbf{y}_j, \mathbf{y}_j^{\text{gt}}), & \text{if } \text{MSE}(\mathbf{y}_j, \mathbf{y}_j^{\text{gt}}) < \varepsilon \\ \text{MSE}(\mathbf{y}_j, \mathbf{y}_j^{\text{gt}})^{0.1} \cdot \varepsilon^{0.9}, & \text{otherwise} \end{cases} \tag{15}$$

Here, ε denotes the threshold for the loss, which is set to $(20\,\text{cm})^2$ in the experiments. The final loss is the average over all valid joints and all scenes in the batch.

For the case of volumetric triangulation, we use the L1 loss with a weak heatmap regularizer, which maximizes the prediction for the voxel that has inside of it the ground-truth joint:

$$\mathcal{L}^{\text{vol}}(\theta, \gamma, \nu) = \sum_j |\mathbf{y}_j - \mathbf{y}_j^{gt}| - \beta \cdot log(V_j^{\text{output}}(\mathbf{y}_j^{gt})) \tag{16}$$

Without the second term, for some of the joints (especially, pelvis) the produced output volumetric heatmaps are not interpretable, probably due to insufficient size of the training datasets [20]. Setting the β to a small value ($\beta = 0.01$) makes them interpretable, as the produced heatmaps always have prominent maxima close to the prediction. At the same time, such small β does not seem to have any effect on the final metrics, so its use can be avoided if interpretability is not needed. We have also tried the loss (15) from the algebraic triangulation instead of L1, but it performed worse in our experiments.

4. Experiments

We conduct experiments on two available large multi-view datasets with available ground-truth 3D pose annotations: Human3.6M [3] and CMU Panoptic [5, 25, 18].

Human3.6M dataset. The Human3.6M [3] is currently one of the largest 3D human pose benchmarks with many reported results both for monocular and multi-view setups. The full dataset consist of 3.6 million frames from 4 synchronized 50 Hz digital cameras along with the 3D pose annotations (collected using a marker-based MoCap system comprised of 10 separate IR-cameras). The dataset has 11 human subjects (5 females and 6 males) split into train, validation and test (only train and validation have the ground-truth annotations).

The 2D backbone for Human3.6M was pretrained on the COCO dataset [10] and finetuned jointly on MPII and Human3.6M for 10 epochs using the Adam optimizer with 10^{-4} learning rate. We use the 3D groundtruth and camera parameters provided by Martinez *et al.* [11]. We undistort the images by applying grid-sampling on the video frames. If not mentioned explicitly, all networks are trained using four cameras and evaluated using all available cameras (either three or four, as one of the subjects lacks data from one camera). All algorithms use the 2D bounding boxes annotations provided with the dataset. The networks are trained for 6 epochs with 10^{-4} learning rate for the 2D backbone and a separate learning rate 10^{-3} for the volumetric backbone.

Note that the volumetric triangulation approach uses predictions obtained from the algebraic triangulation (the whole system, however, potentially can be fine-tuned end-to-end). The size of volumetric cube L was set to 2.5 m, which can enclose all subjects even if there is a few tens of centimeter error in pelvis prediction (which is much higher than the average error by the algebraic triangulation baseline). The number of output channels from the 2D backbone was set to $K = 32$. We did not apply any augmentations during the training, other than rotating the orientation of the cube in volumetric triangulation around the vertical axis.

Protocol 1 (relative to pelvis)	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Monocular methods (MPJPE relative to pelvis, mm)																
Martinez *et al.* [11]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Sun *et al.* [20]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**49.6**
Pavllo *et al.* [16] (∗)	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	**46.8**
Hossain & Little [2] (∗)	48.4	50.7	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
Ours, volumetric single view (†)	41.9	49.2	46.9	47.6	50.7	57.9	41.2	50.9	57.3	74.9	48.6	44.3	41.3	52.8	42.7	49.9
Multi-view methods (MPJPE relative to pelvis, mm)																
Multi-View Martinez [22]	46.5	48.6	54.0	51.5	67.5	70.7	48.5	49.1	69.8	79.4	57.8	53.1	56.7	42.2	45.4	57.0
Pavlakos *et al.* [15]	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.* [22]	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
Kadkhodamohammadi & Padoy [6]	39.4	46.9	41.0	42.7	53.6	54.8	41.4	50.0	59.9	78.8	49.8	46.2	51.1	40.5	41.0	49.1
RANSAC (our implementation)	24.1	26.1	24.0	24.6	27.0	25.0	23.3	26.8	31.4	49.5	27.8	25.4	24.0	27.4	24.1	27.4
Ours, algebraic (w/o conf)	22.9	25.3	23.7	23.0	29.2	25.1	21.0	26.2	34.1	41.9	29.2	23.3	22.3	26.6	23.3	26.9
Ours, algebraic	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
Ours, volumetric (softmax aggregation)	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
Ours, volumetric (sum aggregation)	19.3	20.5	20.1	19.3	20.6	19.8	19.0	22.9	23.5	29.8	22.0	21.4	19.8	22.1	20.3	21.3
Ours, volumetric (conf aggregation)	19.9	**20.0**	18.9	18.5	20.5	19.4	**18.4**	22.1	22.5	28.7	21.2	20.8	19.7	22.1	20.2	**20.8**

Table 1. The results of evaluation on the Human3.6M dataset. The table presents the MPJPE error for the joints (relative to pelvis) for published state-of-the-art monocular and multi-view methods. The methods that are using temporal information during inference are marked by (∗). Note that our monocular method (labeled by †) is using the approximate position of the pelvis estimated from the multi-view.

Protocol 1, absolute positions, filtered validation	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Multi-view methods (absolute MPJPE, mm)																
RANSAC	21.6	22.9	20.9	21.0	23.1	23.0	20.8	22.0	26.4	26.6	24.0	21.5	21.0	23.9	20.8	22.8
Ours, algebraic (w/o conf)	21.7	23.7	22.2	20.4	26.7	24.2	19.9	22.6	31.2	35.6	26.8	21.2	20.9	24.6	21.1	24.5
Ours, algebraic	18.1	20.0	17.6	17.0	18.9	19.3	17.4	19.2	21.9	23.2	19.5	18.0	18.3	20.5	17.9	19.2
Ours, volumetric (softmax aggregation)	16.9	18.1	16.6	16.0	17.1	17.9	16.5	18.5	19.6	20.1	18.2	16.8	17.2	19.0	16.6	17.7
Ours, volumetric (sum aggregation)	17.7	18.5	17.2	16.5	17.8	18.4	17.0	18.9	19.8	20.9	18.9	17.8	17.8	19.2	17.3	18.3
Ours, volumetric (conf aggregation)	18.0	18.3	**16.5**	16.1	17.4	18.2	**16.5**	18.5	**19.4**	20.1	18.2	17.4	**17.2**	19.2	**16.6**	17.9

Table 2. The results of evaluation on the Human3.6M dataset. The table presents the absolute positions MPJPE error for our algorithms. Note that the validation set has been filtered by removing the scenes with erroneous ground-truth 3D pose annotations.

For Human3.6M we follow the most popular protocol with 17-joint subset and testing on the validation. We used the MPJPE (Mean Per Joint Position Error) metric, which is L2 distance between the ground-truth and predicted positions of the joints (in most cases, measured with respect to pelvis). We use every fifth frame for the evaluation. As a baseline, we implemented a simple triangulation method with RANSAC and Huber loss, which is the de-facto standard for solving robust estimation problems. The baseline uses the same pretrained 2D backbone. The results of the standard protocol are summarized in Table 1.

Our implementations surpass the previous art by a large margin, even for the simplest RANSAC baseline. The introduced volumetric methods performs the best, providing about 30% additional reduction in the error to the RANSAC, which is significant.

While most of the works on monocular 3D human pose evaluate the positions of the joints relative to the pelvis (in order to avoid the estimation of global coordinates, which is problematic for monocular algorithms), evaluating with respect to the global coordinates is more reasonable for multi-view setups. This, however, is not straightforward due to 3D pose annotation errors in the Human3.6M - the problem is that some scenes of the 'S9' validation actor (parts of 'Greeting', 'SittingDown' and 'Waiting', see our project page) have the ground truth erroneously shifted in 3D com-

pared to the actual position. Interestingly, the error is nullified when the pelvis is subtracted (as done for monocular methods), however, to make the results for the multi-view setup interpretable we must exclude these scenes from the evaluation. The results for the absolute MPJPE for our methods with these scenes excluded are presented in Table 2, giving a better sense of the magnitude of errors. Interestingly, the average MPJPE for volumetric is much smaller than the size of the voxel (3.9 cm), showing the importance of the subpixel resolving soft-argmax.

Our volumetric multi-view methods can be generalized to the case of a single camera view, naturally taking into account the camera intrinsics. To check the monocular performance we have done a separate experiment with training using a random number of cameras from 1 to 4. For the case of a single camera we used the L1 loss on joint positions relative to the pelvis (as is usually done for the monocular methods). Without any tuning this resulted in 49.9 mm error, which is close to the current state of the art. Note that the position of the cube center was estimated by triangulating the pelvis from all 4 cameras, so the performance of the methods might be somewhat overoptimistic. On the other hand, the average relative positions error of the monocular method lies within 4-6 cm, which corresponds to a negligible shift for the volumetric cube position (less than 2 voxels).
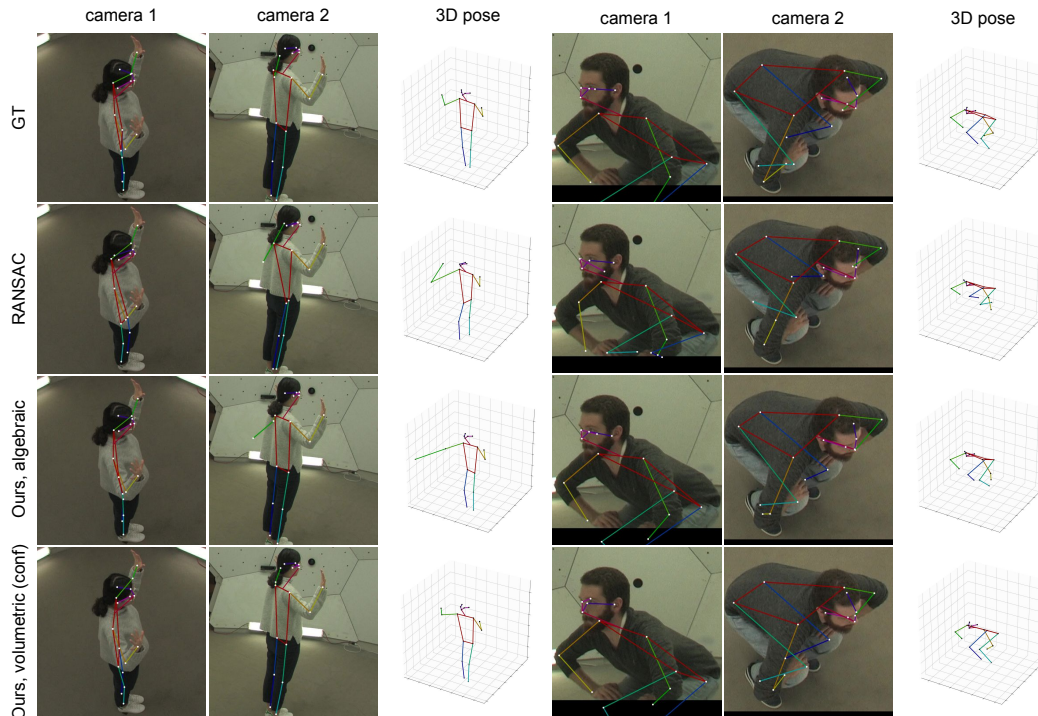
Figure 3. Illustration of the difference in performance of the approaches on the CMU dataset validation (using 2 cameras) that demonstrates the robustness of the volumetric triangulation approach.
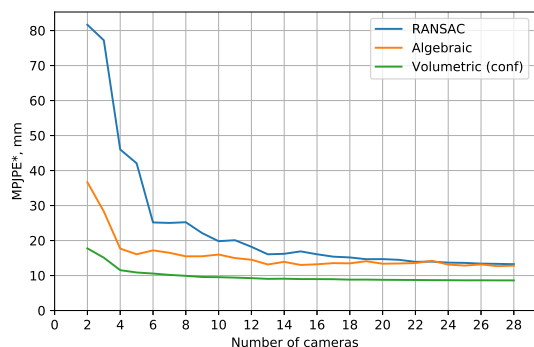

Figure 4. Estimate of the MPJPE absolute error on the subset of CMU validation versus the numbers of cameras (up to 28, treating the annotations from CMU as ground truth). Each value on the plot is obtained by sampling 50 random subsets of cameras followed by averaging.

Model	MPJPE, mm
RANSAC	39.5
Ours, algebraic (w/o conf)	33.4
Ours, algebraic	21.3
Ours, volumetric (softmax aggregation)	**13.7**
Ours, volumetric (sum aggregation)	**13.7**
Ours, volumetric (conf aggregation)	14.0

Table 3. Results of evaluation on the CMU dataset in terms of MPJPE error on the CMU dataset validation (using 4 cameras).

CMU Panoptic dataset. The CMU panoptic is a new multi-camera dataset maintained by the Carnegie Mellon University [5, 25, 18]. The dataset provides 30 Hz Full-HD videostreams of 40 subjects from up to 31 synchronized cameras.

The dataset is provided with annotations of the Full-HD cameras acquired via triangulation using all camera views. Since there are no other published results on the quality of multi-view pose estimation on CMU, we use our own protocol. For the tests we use the 17-subset of the 19-joint annotation format in the dataset, which coincides with the popular COCO format [10]. We used the same train/val split as in [25], which only has scenes with a single person at each point in time. Additionally, we split the dataset by camera views (4 cameras in val, up to 27 cameras in train), so there is no overlap between the test and validation both in terms of subjects and camera views. To get the human bounding boxes we use Mask R-CNN 2D detector with ResNet-152 backbone [12]). Note that we process the frames without taking into account the lens distortion which leads to some loss of accuracy.

The networks are trained using the Adam optimizer similarly to Human3.6M as described in the previous section. The number of views during training was selected randomly from 2 to 5.

The comparison between our multi-view methods is presented in Table 3 with absolute MPJPE used as the main metric. Here, the volumetric approach has a dramatic advantage over the algebraic one, and its superiority is far
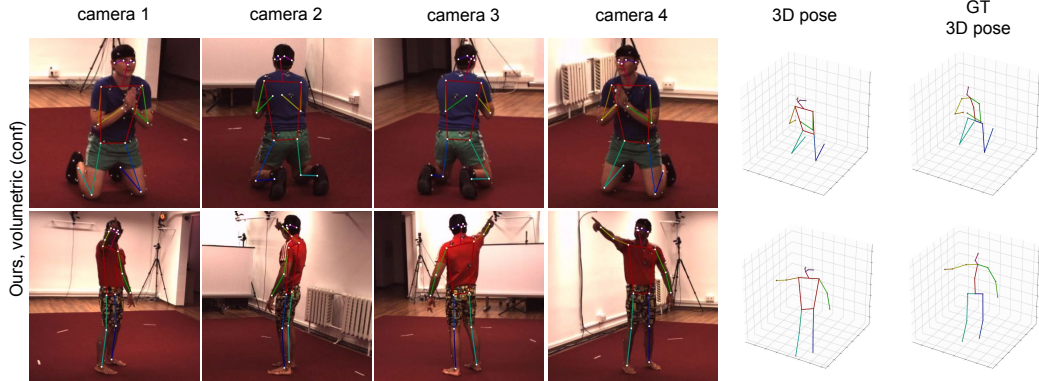
Figure 5. Demonstration of successful transfer of the solution trained on CMU dataset to Human3.6M scenes. Note that keypoint skeleton models on Human3.6M and CMU are different.

more evident than on Human3.6M. We believe that the main point in which CMU differs from Human3.6M is that in CMU most cameras do not have the full view of a person the whole time, leading to strong occlusions and missing parts. This suggests the importance of the 3D prior that can be learnt only by volumetric models. It seems that RANSAC performs worse compared to algebraic triangulation without confidences because it is not finetuned on the CMU data. The difference between the methods' prediction is illustrated in Figure 3. We have not observed any significant difference between the volumetric aggregation methods.

In Figure 4 we present a plot for the error versus the numbers of used cameras. The plot demonstrates that the proposed volumetric triangulation methods allow drastically reducing the number of cameras in real-life setups: the accuracy on "meta"-groundtruth for RANSAC approach with 28 cameras is surpassed by the volumetric approach with just four cameras.

We also conducted experiments to demonstrate that the learnt model indeed generalizes to new setups. For that we applied a CMU-trained model to Human3.6M validation scenes and scenes from KTH Multiview Football Dataset [8] (see our project page for the qualitative results). The qualitative results for the learnable triangulation are presented in Figure 5. Please see videos for all of the methods on our project page. To provide a quantitative measure of the generalizing ability, we have measured the MPJPE for the set of joints which seem to have the most similar semantics (namely, 'elbows', 'wrists' and 'knees'). The measured MPJPE is 36 mm for learnable triangulation and 34 mm for the volumetric approach, which seems reasonable when compared to the results of the methods trained on Human3.6M (16-18 mm, depending on the triangulation method).

5. Conclusion

We have presented two novel methods for the multi-view 3D human pose estimation based on learnable triangulation that achieve state-of-the-art performance on the Human3.6M dataset. The proposed solutions drastically reduce the number of views needed to achieve high accuracy, and produce smooth pose sequences on the CMU Panoptic dataset without any temporal processing, pointing that it can potentially improve the ground truth annotation of the dataset. An ability to transfer the trained method between setups is demonstrated for the CMU Panoptic → Human3.6M pair.

The volumetric triangulation strongly outperformed all other approaches both on CMU Panoptic and Human3.6M datasets. We speculate that due to its ability to implicitly learn a human pose prior this method is robust to occlusions and partial views of a person. Another important advantage of this method is that it explicitly takes the camera parameters as independent input. Finally, volumetric triangulation also generalizes to *monocular* images if human's approximate position is known, producing results close to state of the art.

One of the major limitations of our approach is that it supports only a single person in the scene. This problem can be mitigated by applying available ReID solutions to the 2D detections of humans. Another major limitation of the volumetric triangulation approach is that it relies on the predictions of the algebraic triangulation. This leads to the need for having at least two camera views, which might be a problem for some applications. The performance of our method can also potentially be further improved by adding multi-stage refinement in a way similar to [22].

References

[1] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3

[2] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *European Conference on Computer Vision*, pages 69–86. Springer, 2018. 6

[3] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 2, 5

[4] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 4

[5] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 2, 4, 5, 7

[6] Abdolrahim Kadkhodamohammadi and Nicolas Padoy. A generalizable approach for multi-view 3D human pose regression. apr 2018. 1, 2, 6

[7] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pages 365–376, 2017. 2

[8] Vahid Kazemi, Magnus Burenius, Hossein Azizpour, and Josephine Sullivan. Multi-view body part recognition with random forests. In *2013 24th British Machine Vision Conference, BMVC 2013; Bristol; United Kingdom; 9 September 2013 through 13 September 2013*. British Machine Vision Association, 2013. 8

[9] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[10] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 7

[11] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017. 2, 5, 6

[12] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. Accessed: 01 Feb 2019. 7

[13] Thomas B Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2-3):90–126, 2006. 1

[14] Gyeongsik Moon, Juyong 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[15] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3D human pose annotations. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:1253–1262, 2017. 1, 2, 6

[16] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. *arXiv*, abs/1811.11742, 2018. 1, 2, 6

[17] Helge Rhodin, Frederic Meyer, Jorg Sporri, Erich Muller, Victor Constantin, Pascal Fua, Isinsu Katircioglu, and Mathieu Salzmann. Learning Monocular 3D Human Pose Estimation from Multi-view Images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8437–8446. IEEE, jun 2018. 2

[18] Tomas Simon, Hanbyul Joo, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. *CVPR*, 2017. 5, 7

[19] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 2

[20] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018. 1, 2, 3, 5, 6

[21] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2500–2509, 2017. 2

[22] Denis Tome, Matteo Toso, Lourdes Agapito, and Chris Russell. Rethinking Pose in 3D: Multi-stage Refinement and Recovery for Markerless Motion Capture. In *2018 International Conference on 3D Vision (3DV)*, pages 474–483. IEEE, sep 2018. 1, 2, 6, 8

[23] Hsiao-Yu Fish Tung, Ricson Cheng, and Katerina Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2595–2603, 2019. 2

[24] Timo Von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. Human pose estimation from video and imus. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1533–1547, 2016. 1

[25] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. *arXiv preprint arXiv:1812.01598*, 2018. 5, 7

[26] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[27] Zhixuan Yu, Jae Shin Yoon, Prashanth Venkatesh, Jaesik Park, Jihun Yu, and Hyun Soo Park. Humbi 1.0: Human multiview behavioral imaging dataset. *arXiv preprint arXiv:1812.00281*, 2018. 1