Learnable Human Mesh Triangulation for 3D Human Pose and Shape Estimation

Sungho Chun[1] Sungbum Park[2] Ju Yong Chang[1]
[1]Dept of ECE, Kwangwoon University, Korea [2]NCSOFT, Korea
{asw9161,jychang}@kw.ac.kr, spark0916@ncsoft.com

Abstract

Compared to joint position, the accuracy of joint rotation and shape estimation has received relatively little attention in the skinned multi-person linear model (SMPL)-based human mesh reconstruction from multi-view images. The work in this field is broadly classified into two categories. The first approach performs joint estimation and then produces SMPL parameters by fitting SMPL to resultant joints. The second approach regresses SMPL parameters directly from the input images through a convolutional neural network (CNN)-based model. However, these approaches suffer from the lack of information for resolving the ambiguity of joint rotation and shape reconstruction and the difficulty of network learning. To solve the aforementioned problems, we propose a two-stage method. The proposed method first estimates the coordinates of mesh vertices through a CNN-based model from input images, and acquires SMPL parameters by fitting the SMPL model to the estimated vertices. Estimated mesh vertices provide sufficient information for determining joint rotation and shape, and are easier to learn than SMPL parameters. According to experiments using Human3.6M and MPI-INF-3DHP datasets, the proposed method significantly outperforms the previous works in terms of joint rotation and shape estimation, and achieves competitive performance in terms of joint location estimation.

1. Introduction

Human pose estimation from single or multi-view images is a long-standing computer vision problem. In many studies [9, 11, 30, 35], the human pose is simply represented as a set of 3D coordinates of the body joints. Compared to joint coordinate, human joint rotation and shape estimation has not received much attention. However, when 3D joint coordinates as well as joint rotations and human shape information are available together, the body of a person can be better described, as shown in Fig. 1(a) and (c). The estimated joint and shape information can also be used for human part segmentation [13] and detailed human mesh reconstruction [45, 46].

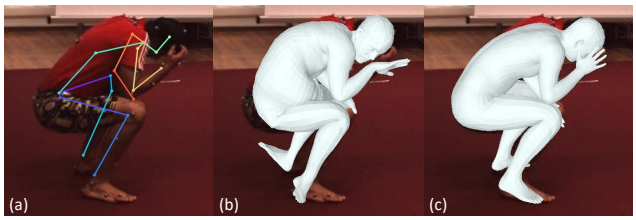

Figure 1: Results for (a) joint position estimation, (b) joint fitting, and (c) surface fitting are visualized. Joint fitting and surface fitting indicate that SMPL is fitted to the estimated joint set and vertex set, respectively.

The skinned multi-person linear model (SMPL) [22] is frequently used for multi-view human mesh reconstruction methods [32, 33, 38, 42], which can acquire joint rotations and human shape as well as joint coordinates. Among the methods, the most similar to our proposed method is [42]. This method first estimates 3D joints from multi-view images and then additionally computes joint rotation and shape information by fitting the SMPL-X [28] model to the 3D joints. However, this fitting framework heavily relies on regularization because joint coordinates do not provide enough information to resolve the ambiguity in the estimation of joint rotation and shape information. Nevertheless, the lack of such information can degrade joint rotation and human shape estimation performance, as shown in Fig. 1(b). The convolutional neural network (CNN)-based model proposed in [33] directly regresses SMPL pose and shape parameters from input multi-view images. However, the mapping function from the input image to the SMPL parameter is highly non-linear [27], which makes learning the model difficult.

In this paper, we propose a *Learnable human Mesh Triangulation (LMT)* method for SMPL-based human mesh reconstruction from sparse multi-view images. The proposed method can solve the above two problems. LMT first estimates human surface vertex coordinates, not human joints, from the input multi-view images, and then fits the SMPL model to the resultant vertices. Such surface vertex co-

ordinates provide strong constraints on joint rotation and human shape, which can help resolve the ambiguity problem. Also, many previous works [3, 6, 24, 34, 37, 39] verified that heatmap-based keypoint estimation can be easily learned through CNNs, especially fully convolutional networks. Our basic idea is to extend this heatmap-based keypoint estimation framework to SMPL mesh vertex estimation, which can solve the non-linearity problem in direct SMPL parameter regression.

To reconstruct SMPL-based human mesh vertices, we extend *Learnable Triangulation of human pose (LT)* [11], the heatmap-based method for estimating sparse joints to dense vertices. However, the application of LT to mesh vertices is non-trivial and raises two issues to be overcome. The first is high computational complexity. LT generates 3D heatmaps to estimate body joints. No problem is observed in the case of sparse joints (e.g., $\sim$20 for Human3.6M [10] and MPI-3DHP-INF [26]). In contrast, the use of a 3D heatmap may cause excessive GPU memory usage in the case of dense mesh vertices (e.g., 6890 for SMPL). However, the optimization process used to obtain SMPL parameters in the proposed method does not require full-vertices. Rather, estimating appropriately sampled sub-vertices can improve the performance of the model while solving the computational issue, which is proven through our experiments.

The second issue is the inconsistency between multi-view features. In our method, multi-view features are aggregated in each voxel after being unprojected into 3D space. In the case of voxel on the human surface, multi-view features aggregated into the voxel must be consistent. However, occlusion can lead to inconsistency between aggregated multi-view features, which makes vertex coordinate estimation difficult. To alleviate this problem, we propose to utilize the visibility information obtained from the single-view mesh reconstruction method. The basic idea is to use visibility information to increase the dependence of a certain voxel on features obtained from visible views and reduce the dependence on features obtained from invisible views. We experimentally show that utilizing visibility information alleviates the multi-view inconsistency problem and improves mesh reconstruction performance.

The contributions of this paper can be summarized as follows:

- We quantitatively and qualitatively prove that fitting the SMPL model to human surface vertices rather than human body joints leads to better mesh reconstruction results in terms of joint rotation and human shape.
- We show that the computational issue that makes it difficult to extend the heatmap-based framework to SMPL mesh vertices can be resolved through sub-vertices estimation, which also brings additional performance gain.
- Per-vertex visibility information is utilized to consider the consistency of multi-view features. Moreover, cross-dataset experiments show that the use of visibility improves the generalization performance of our model.
- Extensive experiments using Human3.6M and MPI-INF-3DHP datasets prove that the ideas of sub-vertices estimation and per-vertex visibility are effective. Consequently, the proposed framework outperforms previous methods in terms of joint rotation and human shape while showing competitive results in terms of 3D joint coordinates.

2. Related Work

2.1. Multi-view Joint Estimation

Many methods [9, 11, 12, 29, 30, 36, 40] have been proposed to estimate the 3D human pose in the form of joint coordinates from the input multi-view images. Among the methods for estimating the pose of a single person, the one most similar to our work is LT [11]. LT aggregates 2D features extracted from multi-view images in 3D voxel space and then applies the 3D convolution to the aggregated feature to estimate 3D pose. However, the final LT output is the 3D joint locations without joint rotation information. In contrast, in our method, the SMPL parameters are estimated, which enables a richer reconstruction of the human body, including joint rotations and human shape.

2.2. Multi-view Joint and Shape Estimation

Many studies [17, 32, 33, 38, 42, 43] have been conducted to estimate joint rotations or human shape as well as joint coordinates from input multi-view images. For SMPL and SMPL-X parameter estimation, the model is fitted to the predicted 3D joints in [42], and the 3D joints are fed into the feedforward network in [38]. In contrast, our method estimates SMPL parameters using 3D mesh vertices rather than 3D joints. Since the human surface provides richer information than joint coordinates for joint rotation and human shape estimation, our method can reconstruct rotation and shape more accurately than joint-based methods [38, 42]. In [17], the Mannequin dataset [18] is used to train a model that robustly predicts SMPL parameters in an in-the-wild environment. The dataset provides videos of static humans captured by a dynamic camera. The method in [17] performs 3D joint estimation by applying the structure-from-motion (SfM) algorithm to the input video. However, the SfM method is generally difficult to apply to sparse multi-view environments, e.g., Human3.6M and MPI-INF-3DHP datasets, which are the focus of this work. The geometry of a clothed human is reconstructed in [43] and multiple

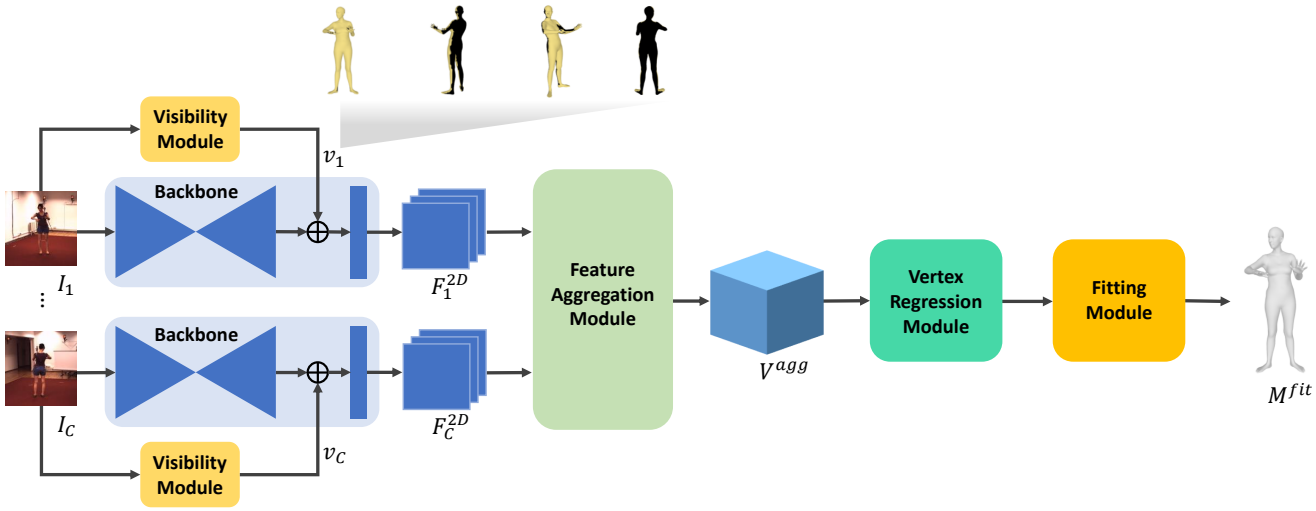

Figure 2: **Overall pipeline of the proposed method.** Visible vertices in the visibility map are colored in gold. $\oplus$ denotes concatenate operation.

images obtained from a dynamic camera are used in [32]. Their goals and settings are different from our work.

In [33], an existing work with the same goal as ours, SMPL parameters are directly regressed from multi-view images through a CNN model. However, learning the network in this method is difficult due to the high non-linearity of the regression function [4, 13, 15, 16, 19, 20, 27]. Therefore, our method learns a keypoint estimation network based on heatmap regression rather than parameter regression, and then obtains SMPL parameters by fitting SMPL to human mesh vertices predicted by the network.

3. Proposed Method

3.1. Overview of the Proposed Method

We propose a method (i.e., LMT) to estimate the SMPL-based 3D mesh of a single person from multi-view images obtained by C calibrated cameras. Fig. 2 shows the overall pipeline of the proposed method, which consists of the visibility module, CNN backbone, feature aggregation module, vertex regression module, and fitting module. The visibility module estimates per-vertex visibility $v_c \in \mathbb{R}^N$ for sub-sampled mesh from every single image $I_c \in \mathbb{R}^{H_0 \times W_0 \times 3}$, where N denotes the number of subsampled vertices. The CNN backbone computes visibility augmented image features $F_c^{2D} \in \mathbb{R}^{H \times W \times K}$ from the input multi-view image I_c and per-vertex visibility v_c. The feature aggregation module unprojects the input image feature F_c^{2D} into the 3D global voxel space to generate C volumetric unprojected features $V_c^{unproj} \in \mathbb{R}^{64 \times 64 \times 64 \times K}$, then aggregates the unprojected features $\{V_c^{unproj}\}_{c=1}^{C}$ to produce the volumetric aggregated feature $V^{agg} \in \mathbb{R}^{64 \times 64 \times 64 \times K}$. The vertex regression module generates 3D vertex coordinates $M \in \mathbb{R}^{N \times 3}$ of sub-sampled mesh from the aggregated feature V^{agg} using 3D convolution and soft-argmax operation [35]. The fitting module outputs the final joint coordinates, rotations, and shape information by fitting the SMPL model to the 3D vertex coordinates M from the vertex regression module.

3.2. Visibility Module

The visibility module calculates the per-vertex visibility map v_c from the single-view image I_c. We implement the visibility module using the I2L-MeshNet [27], one of the state-of-the-art single-view human mesh reconstruction methods, and the general visibility computation algorithm [1]. The detailed procedure is as follows. We first feed a single image I_c into the I2L-MeshNet and obtain the human mesh defined in the human-centered coordinate system of which the origin is defined as the pelvis joint. However, the visibility computation algorithm requires camera coordinates of the human mesh. Therefore, the algebraic triangulation method [11] is used to estimate the pelvis joint. The camera coordinates of the estimated pelvis joint are used to transform the human mesh obtained by I2L-MeshNet into the camera coordinate system. The visibility computation algorithm is then used to obtain the visibility map for full-vertices $v_c^{full} \in \mathbb{R}^{6890}$. To prevent overfitting of the proposed model, we apply additional mesh subsampling [31] to v_c^{full} and use the resultant per-vertex visibility map v_c of sub-vertices for subsequent processes.

3.3. Backbone

The CNN backbone outputs the visibility augmented image features $\{F_c^{2D}\}_{c=1}^{C}$ from input multi-view images $\{I_c\}_{c=1}^{C}$ and per-vertex visibility $\{v_c\}_{c=1}^{C}$. To construct the proposed backbone, according to [11], we remove the last classification and pooling layers of ResNet-152 [8] pre-trained on COCO [21] and MPII [1], and then add three

[1] https://github.com/MPI-IS/mesh

deconvolution layers and a 1×1 convolution layer to the back of the network. The last deconvolution layer of the backbone creates an intermediate feature $F_c^{deconv} \in \mathbb{R}^{H \times W \times 256}$. After v_c is extended to the spatial axis, it is concatenated with the intermediate feature F_c^{deconv}. An additional 1×1 convolution is applied to the concatenated feature to generate the visibility augmented image feature F_c^{2D}.

3.4. Feature Aggregation Module

In the feature aggregation module, the 2D feature F_c^{2D} from the backbone is unprojected into a cuboid defined in 3D world space to create a volumetric unprojected feature V_c^{unproj}. The volumetric aggregated feature V^{agg} is then calculated through the aggregation of the unprojected features $\{V_c^{unproj}\}_{c=1}^{C}$. In the proposed method, the estimation of the vertex coordinates of the subsampled mesh M depends on the unprojected 3D features in the cuboid. Therefore, the location and size of the cuboid should be set so that the cuboid contains the target human subject. Consequently, a cuboid with a side length of 2.0 m, centering on the pelvis of the target subject, is created.

The construction process of the unprojected feature V_c^{unproj} through the unprojection of F_c^{2D} is as follows. We first project the 3D coordinates of the cuboid voxels $V^{coords} \in \mathbb{R}^{64 \times 64 \times 64 \times 3}$ into the 2D image plane of each view using the camera projection matrix and obtain the 2D image coordinates $V_c^{proj} \in \mathbb{R}^{64 \times 64 \times 64 \times 2}$. Next, bilinear sampling is used to extract 2D features corresponding to each location of V_c^{proj} from F_c^{2D}, and, consequently, V_c^{unproj} is obtained:

$$V_c^{unproj} = F_c^{2D}\{V_c^{proj}\}, \tag{1}$$

where $\{\cdot\}$ denotes bilinear sampling. Then C unprojected features in 3D world space are aggregated using 3D softmax operation [11]. This can be written as:

$$V^{agg} = \sum_{c=1}^{C}(d_c \odot V_c^{unproj}), \tag{2}$$

$$d_c = \frac{\exp(V_c^{unproj})}{\sum_{c=1}^{C}\exp(V_c^{unproj})}, \tag{3}$$

where $d_c \in \mathbb{R}^{64 \times 64 \times 64 \times K}$ and $\odot$ denote the confidence weight and element-wise multiplication, respectively.

3.5. Vertex Regression Module

The vertex regression module with encoder-decoder structure composed of 3D convolution generates the vertex coordinates of the subsampled mesh M from the input aggregated feature V^{agg}. The encoder first computes a 3D feature with $2 \times 2 \times 2$ resolution and 128 channel dimension from V^{agg}, which is fed into the decoder to output a volumetric feature $V \in \mathbb{R}^{64 \times 64 \times 64 \times 32}$. Next, a $1 \times 1 \times 1$ 3D convolution is applied to V to produce 3D heatmaps $H^{3D} \in \mathbb{R}^{64 \times 64 \times 64 \times N}$ for the subsampled vertices. Details of the proposed encoder-decoder are presented in the Supplementary material.

A 3D soft-argmax operation is used to obtain vertex coordinates M from the 3D heatmaps H^{3D}:

$$\tilde{H}_n^{3D} = \frac{\exp(H_n^{3D})}{\sum_{i,j,k}\exp(H_n^{3D}(i,j,k))}, \tag{4}$$

$$M_n = \sum_{i,j,k} r \cdot \tilde{H}_n^{3D}(i,j,k), \tag{5}$$

where $r = [r_i, r_j, r_k]$ denotes the world coordinate vector of the voxel with indices (i,j,k) in the 3D heatmap. H_n^{3D}, $\tilde{H}^{3D}$, and M_n denote the n-th channel of the 3D heatmap, the normalized 3D heatmap, and the n-th row vector of M, respectively.

To train the proposed network, an L1 loss is applied to the vertices generated by the vertex regression module:

$$\mathcal{L}_M = \frac{1}{N}\sum_{n=1}^{N}\|M_n - M_n^*\|_1, \tag{6}$$

where M^* denotes the ground-truth mesh.

3.6. Fitting Module

The fitting module is used to acquire the SMPL parameters corresponding to the vertex coordinates M generated by the vertex regression module. Fitting module is based on optimization according to the existing works [23, 25, 28, 41, 42] and optimization parameters $\Theta = \{z \in \mathbb{R}^{32}, R \in \mathbb{R}^6, \beta \in \mathbb{R}^{10}, t \in \mathbb{R}^3\}$ contains VPoser's latent code z, global rotation with continuous representation [44] R, shape parameter β, and global translation t. From the latent code, VPoser $\mathcal{V}(\cdot)$ calculates the SMPL pose parameter $\theta = \mathcal{V}(z) \in \mathbb{R}^{69}$, which is fed into the SMPL decoder $\mathcal{M}(\cdot)$ together with R, β, and t to produce the SMPL mesh $M^{fit} = \mathcal{M}(\theta, R, \beta, t) \in \mathbb{R}^{6890 \times 3}$. The SMPL mesh is transformed into sub-vertices $M_{sub}^{fit} = sub(M^{fit}) \in \mathbb{R}^{N \times 3}$ by the mesh coarsening function [31] $sub(\cdot)$. The fitting module updates Θ iteratively to reduce the difference between the sub-vertices of the fitted mesh M_{sub}^{fit} and the regressed vertices M.

The cost function for fitting is defined as follows:

$$\mathcal{E}_{fit} = \mathcal{E}_{data} + \mathcal{E}_{reg}, \tag{7}$$

$$\mathcal{E}_{data} = \frac{1}{N}\sum_{n=1}^{N}\|M_{sub,n}^{fit} - M_n\|_2^2, \tag{8}$$

$$\mathcal{E}_{reg} = \lambda_z\mathcal{E}_z + \lambda_\beta\mathcal{E}_\beta + \lambda_w\mathcal{E}_{\theta_w} + \lambda_\alpha\mathcal{E}_\alpha, \tag{9}$$

where $M^{fit}_{sub,n}$ and $\theta_w \in \mathbb{R}^6$ denote the n-th row vector of M^{fit}_{sub} and the axis-angle representation of both wrist joints. $\mathcal{E}_z$, $\mathcal{E}_\beta$, and $\mathcal{E}_{\theta_w}$ are the L2 regularization terms for z, β, and θ_w, respectively. And $\mathcal{E}_\alpha$ is the exponential regularization term for preventing unnatural bending of the elbows and knees [2, 28]. Each λ represents regularization weight.

Joint coordinates $J = GM^{fit} \in \mathbb{R}^{17\times3}$ can be obtained from the fitted mesh M^{fit} using a pretrained joint regression matrix $G \in \mathbb{R}^{17\times6890}$. The obtained J is used to evaluate the joint coordinate estimation performance.

4. Experimental Results

4.1. Implementation Details

The spatial sizes of the input image I_c and 2D feature F^{2D}_c are set to $(H_0, W_0) = (384, 384)$ and $(H, W) = (96, 96)$, respectively. The bounding box provided in the datasets is used to crop the human region from the input image. Random rotation is applied to the cuboid [11] along the vertical axis of the ground, and other augmentation is not used. Except for the fitting module, our network is trained end-to-end. Learnable parameters are included in the backbone and vertex regression module, and their learning rates are set to 1e-4 and 1e-3, respectively. The mini-batch size, number of epochs, number of sub-vertices N, and channel of the feature map K is set to 3, 15, 108, and 32, respectively. The Adam optimizer [14] is used to train our network, which takes about 3.5 days using a single RTX 3090 GPU. Mesh sub-sampling algorithms [31] are applied to ground-truth human mesh vertices to obtain sub-sampled vertices, which are used for network training. The Adam optimizer is also used to update the optimization parameter Θ in the fitting module. The fitting module learning rate, number of iterations for fitting, λ_w, λ_z, λ_β, and λ_α are set to 6e-2, 500, 6e-2, 2e-6, 5e-6, and 5e-5, respectively. All regularization weights are simply determined through greedy search.

4.2. Datasets

Human3.6M [10] is a large-scale dataset for 3D human pose estimation, including 3.6M video frames and 3D body joint annotations acquired from four synchronized cameras. It includes 11 human subjects (five females and six males), and according to previous works [11, 33], S1, S5, S6, S7, and S8 are used for training and S9, and S11 for testing. The SMPL mesh obtained by applying MoSh [23] to Human3.6M is used for training and testing as ground-truth. The input image is undistorted before training and testing.

MPI-INF-3DHP [26] is a dataset for 3D human pose estimation and is obtained through the multi-camera markerless MoCap system. Since its test data includes single-view images, only train data composed of multi-view (i.e., 14) images are used in our experiments. Train data includes eight subjects. For a fair comparison, according to previous work [33], S1-S7 are used for training, S8 is used for testing, and views 0, 2, 7, and 8 are used among all cameras. MPI-INF-3DHP provides ground-truth 3D human joints, but does not provide ground-truth 3D human meshes, so pseudo ground-truth meshes are used to train the model. The pseudo ground-truth SMPL parameters are obtained by fitting the SMPL model to ground-truth 3D joints [28], but the pseudo parameters are not used for evaluation.

4.3. Evaluation Metrics

Mean-per-joint-position-error (MPJPE) is a metric that evaluates the performance of 3D human pose estimation based on the L2 distance between the predicted and ground-truth body joints. For LMT, joint coordinates in the world coordinate system can be estimated. Thus, following existing works [9, 11], the L2 distance between the two joint sets is computed without aligning the predicted and ground-truth pelvis joints [5, 13, 15, 16, 19, 20, 27].

Mean-per-vertex-error (MPVE) is a metric that evaluates the performance of human mesh reconstruction based on the L2 distance between predicted and ground-truth mesh vertices. The proposed method is evaluated through MPVE only for the Human3.6M dataset on which ground-truth human meshes are available.

MPJPE and MPVE are used for the evaluation of human mesh reconstruction methods in most existing works. However, because MPJPE and MPVE measure the position errors for joints and vertices, they do not provide information on whether the rotation of the body part is accurately estimated. Therefore, the angular distance d_{ang} [7] is used between the estimated and ground-truth joint rotations for evaluating the proposed method:

$$d_{ang} = 2\sin^{-1}\frac{\|R - R^*\|_F}{2\sqrt{2}}, \tag{10}$$

where, R, R^*, and $\|\cdot\|_F$ denote the predicted rotation matrix, ground-truth rotation matrix, and Frobenius norm, respectively. The joint rotation is defined relative to its parent joint. The rotation of the root joint (i.e., pelvis) denotes the global orientation of the entire body. All angular distances described in this paper are in degree units.

MPJPE averages the 3D position errors of all joints, so it cannot provide information about the case where only a specific joint has a large error. Therefore, 3DPCK [26] that computes the proportion of 3D joints with errors below a certain threshold is used. The AUC [26] is also presented for threshold-independent evaluation.

4.4. Ablation Experiments

The number of sub-vertices. The main problem of 3D heatmap-based prediction for SMPL mesh vertices is excessive GPU memory allocation for 3D heatmaps. This prob-

Number of vertices	MPJPE ↓	MPVE ↓	Angular ↓
6890	19.85	25.21	11.98
431	18.40	**24.15**	11.60
216	18.97	25.10	11.75
108	**18.10**	24.88	**11.54**
54	18.35	26.47	12.00

Table 1: **Ablation results for the number of estimated vertices on Human3.6M.** 3D heatmaps with $16 \times 16 \times 16$ resolution are used in all experiments in this table.

Heatmap resolution	MPJPE ↓	MPVE ↓	Angular ↓
$16 \times 16 \times 16$	18.10	24.88	11.54
$32 \times 32 \times 32$	18.02	24.19	11.45
$64 \times 64 \times 64$	**17.59**	**23.70**	**11.33**

Table 2: **Ablation results for the resolution of 3D heatmaps on Human3.6M.** 108 vertices are estimated in all experiments in this table.

lem can be solved by estimating fewer sub-vertices. For example, if 108 sub-vertices are used instead of 6890 SMPL vertices, the size of GPU memory for the 3D heatmap is reduced by about $6890/108 \approx 63.8$ times.

To investigate the effect of using sub-vertices on mesh reconstruction performance, MPJPE, MPVE, and angular distance results are presented according to the number of vertices in Table 1. To compare the full-vertices model and all sub-vertices models under the same condition, $16 \times 16 \times 16$ heatmap resolution is used. It is the maximum resolution at which a full-vertices model under our computing resources can be trained.

Table 1 shows that better quantitative results are obtained in most cases using sub-vertices than when using full-vertices. Only for the 54 sub-vertices, MPVE and angular distance performances deteriorate compared to full-vertices. This degraded performance is due to the fact that 54 sub-vertices do not provide sufficient information for joint rotation and shape reconstruction, given the supplementary material. We adopt the 108 vertices model that shows the best MPJPE and angular distance performance and requires a relatively smaller heatmap size.

Heatmap resolution. Experiments using various heatmap resolutions are conducted to investigate a model that can accurately estimate 108 sub-vertices. Table 2 shows the performance for the cases in which the heatmap resolution is set to $16\times16\times16$, $32\times32\times32$, and $64\times64\times64$. The proposed method shows the best performance at $64\times64\times64$ heatmap resolution. In this case, the memory allocation for the heatmap is $64 \times 64 \times 64 \times 108 \times 4byte = 113.2MB$, which is similar to the memory allocation of $16 \times 16 \times 16 \times 6890 \times 4byte = 112.9MB$ for the maximum heatmap resolution allowed by full-vertices. According to the performance comparison of the full-vertices model in Table 1 and the sub-vertices model in Table 2, the 108 vertices model with $64\times64\times64$ heatmap resolution achieves better performance for all evaluation metrics than the full-vertices model without additional memory cost.

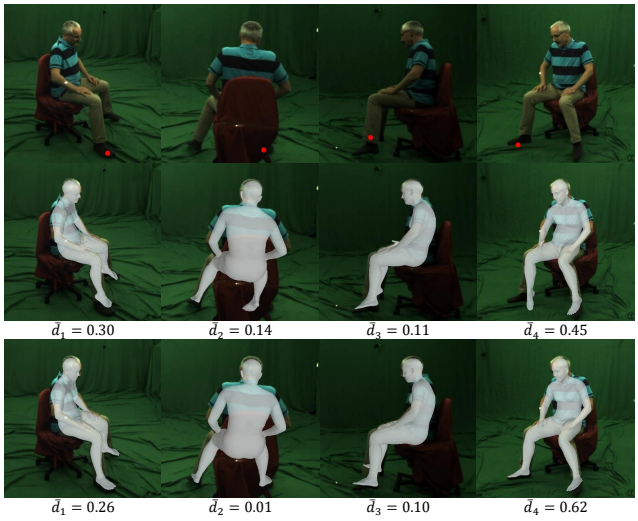

Figure 3: The first row visualizes the input multi-view images. The second and third rows show the reconstructed meshes generated from the softmax baseline and LMT, respectively. $\bar{d}_c$ is obtained by averaging the confidence weight d_c corresponding to the red pixel on the multi-view image along the channel axis. The red pixels are obtained by projecting the voxel including a ground-truth vertex on the right foot to each image plane. Therefore, $\bar{d}_c$ indicates how much the model depends on the image feature obtained from each view c to construct the aggregated feature of the voxel containing the right foot vertex.

Multi-view inconsistency. To prove that using visibility helps feature aggregation, we implement the *softmax baseline* that does not use visibility, and compare it to the LMT. The Softmax baseline generates the image feature F_c^{2D} by directly feeding F_c^{deconv} into the 1×1 convolution layer without concatenation with visibility v_c. When aggregating multi-view features, using only features obtained from views in which the human body surface is visible is desirable, because this leads to the consistency of aggregated multi-view features. However, in the softmax baseline, the multi-view inconsistency problem may arise, which can be mitigated through the use of visibility.

In the first and fourth views of Fig. 3, the right foot is clearly visible. In the third view, the right foot is not visible, but it can be contextually inferred that it is behind the left foot. However, in the second view, severe occlusion prevents the right foot from being estimated. Therefore, relying on the features obtained from the remaining views rather than the second view is preferable for estimating the position of the right foot. However, the softmax baseline shows a higher dependence on the second view than on the third view, which causes the model to incorrectly estimate the

Model	MPVE †	MPJPE	MPVE	Angular
Softmax	22.12	17.84	24.14	11.42
LMT	**21.50**	**17.59**	**23.70**	**11.33**

Table 3: **Comparison with the softmax baseline on Human3.6M.** † means that the regressed vertices M from the vertex regression module are evaluated.

Model	S1	S2	S3	S4	S5	S6	S7	S8	**Avg**
MPJPE ↓									
Softmax	85.83	64.03	63.42	82.55	65.18	71.68	66.02	70.74	70.66
LMT	**81.32**	**61.49**	**60.50**	**78.62**	**62.47**	**71.39**	**65.77**	**66.78**	**68.02**
3DPCK ↑									
Softmax	89.16	96.68	95.01	88.67	94.37	93.42	94.05	94.04	93.32
LMT	**90.28**	**97.80**	**95.85**	**89.81**	**95.21**	**93.52**	**94.91**	**95.22**	**94.07**
AUC ↑									
Softmax	53.00	59.97	62.10	58.01	60.11	58.06	60.44	57.53	58.91
LMT	**53.94**	**60.47**	**62.54**	**58.66**	**60.32**	**58.12**	**60.53**	**58.33**	**59.30**

Table 4: **Cross-dataset evaluation of the softmax baseline and LMT.** The two models are trained on Human3.6m and evaluated on MPI-INF-3DHP. S1-S8 denote the subjects in MPI-INF-3DHP.

Model	MPJPE ↓	MPVE ↓	Angular ↓
LT-fitting [11, 42]	**16.21**	35.20	15.73
LT-fitting [11, 42] (w/o reg)	16.40	42.99	22.94
LMT	17.59	**23.70**	**11.33**
LMT (w/o reg)	17.48	25.30	13.03

Table 5: **Comparison with joint fitting on Human3.6M.** "w/o reg" means that no regularization term $\mathcal{E}_{reg}$ is used.

right foot mesh. On the other hand, LMT uses visibility to reduce the dependence on the second view and increase the dependence on the remaining views. Consequently, LMT successfully reconstructs the right foot mesh.

Effect of using visibility. We investigate the quantitative results of using per-vertex visibility in terms of joint coordinates, rotations, and shape estimation. Table 3 shows the results when the softmax baseline and LMT are trained on Human3.6M train data and evaluated on Human3.6M test data. The second column in Table 3 shows the MPVE for sub-vertices estimated by the vertex regression module, which proves that using visibility helps the network to estimate the human surface accurately. Columns 3-5 of Table 3 show that the use of visibility helps to improve MPJPE, MPVE, and angular distance results even after fitting.

Generalization. The proposed method exploits geometry information (i.e., visibility) obtained from a single-view model for feature aggregation. This single-view model can be trained using more various datasets than the multi-view model. Therefore, the use of geometry information from the single-view model causes the effect of implicit learning through such various datasets and helps to improve the generalization performance of the proposed method. To prove this quantitatively, we train the softmax baseline and LMT using Human3.6M train data and evaluate them for all subjects of MPI-INF-3DHP. Despite the differences between

Figure 4: **Qualitative comparison with joint fitting on Human3.6M.** The first column shows the input images. The second and third columns visualize the meshes reconstructed by LT-fitting and LMT, respectively.

the two datasets, Table 4 shows that the LMT significantly outperforms the softmax baseline in all metrics evaluating the joint coordinate estimation performance.

Comparison with joint fitting. We demonstrate that fitting on the human surface brings more benefits than fitting on the human joint [42]. However, the method of [42] cannot be directly compared with LMT because it is for multi-person mesh reconstruction. Therefore, we design the *LT-fitting baseline* using the state-of-the-art multi-view joint estimation method LT [11]. LT-fitting modifies $\mathcal{E}_{data}$ to minimize the difference between the predicted and ground-truth joints, and uses the same regularization term as LMT.

Table 5 shows MPJPE, MPVE, and angular distance results of LT-fitting and LMT evaluated on Human3.6M. In both cases of LT-fitting and LMT, the use of regularization results in better joint rotation and shape estimation. And LT-fitting relies more heavily on regularization than LMT. However, LT-fitting with regularization shows worse MPVE and angular distance results than LMT without regularization. Table 6 shows the rotation errors of LT-fitting and LMT for each joint. For most joints, the rotation prediction performance of LMT is significantly better than that of LT-fitting. Fig. 4 shows the human mesh reconstruction by LT-fitting and LMT. LT-fitting cannot describe the subject's body shape well because it cannot resolve the ambiguity of the human shape. On the other hand, LMT shows a visually satisfactory result. All these results show that using the human surface rather than the human joint is beneficial for human pose and shape estimation.

4.5. Comparison on Human3.6M

Table 7 shows the results of previous multi-view human mesh reconstruction methods and LMT trained and evaluated on Human3.6M. The same input image size and the same backbone are used for a fair comparison with the *pa-*

Angular ↓	pelvis	L-hip	R-hip	torso	L-knee	R-knee	spine	L-ankl	R-ankl	chest	neck	L-thrx	R-thrx	head	L-shld	R-shld	L-elbw	R-elbw	L-wrst	R-wrst
LT-fitting [11,42]	8.18	10.10	9.37	10.75	9.17	9.21	7.8	17.31	16.86	5.88	**12.07**	10.72	11.64	12.52	**11.65**	14.18	20.24	16.14	43.00	43.20
LMT	**4.77**	**5.69**	**5.79**	**6.40**	**5.80**	**5.38**	**5.68**	**8.58**	**9.85**	**4.48**	12.32	**9.39**	**10.22**	**10.69**	11.86	**14.06**	**13.45**	**11.50**	**19.53**	**20.22**

Table 6: **Per-joint rotation error comparison with joint fitting on Human3.6M.**

Model	MPJPE ↓	MPVE ↓	Angular ↓
(R50-224) Parameter regr. [33]	46.90	-	-
(R50-224) LMT	**30.56**	42.28	14.61
(R152-384) LT-fitting [11,42]	**16.21**	35.20	15.73
(R152-384) LMT	17.59	**23.70**	**11.33**

Table 7: **Comparison results on Human3.6M.** "R50-224" means that ResNet-50 backbone and input image of 224×224 resolution are used. Similarly, "R152-384" denotes ResNet-152 and 384×384 resolution.

Model	MPJPE ↓	3DPCK ↑	AUC ↑
(R50-224) Parameter regr. [33]	50.20	**97.40**	65.60
(R50-224) LMT	**45.87**	96.59	**71.57**
(R152-384) LT-fitting [11,42]	**33.33**	**99.60**	**77.23**
(R152-384) LMT	33.70	99.37	77.09

Table 8: **Comparison results on MPI-INF-3DHP.**

rameter regression method of [33]. Since [33] does not provide MPVE and angular distance results, MPJPE is used for comparison, which shows that LMT significantly outperforms the method of [33]. These results show that the combination of heatmap-based vertex regression and subsequent SMPL fitting brings more accurate results than the method of directly regressing the SMPL parameters from input images. LT-fitting is different from LMT in that SMPL is fitted to the human joint rather than the human surface. Due to this difference, LT-fitting does not obtain enough information to resolve the ambiguity for joint rotation and human shape determination, and as a result achieves significantly lower MPVE and angular distance performance than LMT.

4.6. Comparison on MPI-INF-3DHP

Table 8 shows the results of previous multi-view human mesh reconstruction methods and LMT trained and evaluated on MPI-INF-3DHP. For a fair comparison with [33], the LMT model is pretrained on Human3.6M and then fine-tuned on MPI-INF-3DHP. For 3DPCK with a threshold of 150 mm, [33] shows better results than LMT, but for threshold-independent AUC, LMT shows better results. Also, as in Human3.6M, LMT shows better MPJPE performance. The LMT model shows a competitive joint coordinate estimation result with LT-fitting. In the case of MPI-INF-3DHP, ground-truth SMPL parameters are not provided, so joint rotation and shape estimation results are not presented. However, LMT gives qualitatively better mesh reconstruction results than LT-fitting, as shown in Fig. 5.

5. Conclusion

In this paper, a two-stage method consisting of visibility-based sub-vertices estimation and surface fitting is proposed to reconstruct a single human mesh from multi-view images. The estimation of sub-vertices rather than full-vertices solves the problem of excessive GPU memory usage. In addition, the use of per-vertex visibility improves the mesh vertices estimation performance by alleviating the multi-view inconsistency problem. Surface fitting is also demonstrated to help estimate joint rotations and human shape compared to joint fitting. According to the experimental results, the proposed LMT significantly outperforms the existing multi-view human mesh reconstruction methods on the Human3.6M and MPI-INF-3DHP datasets. However, since using a single-view mesh reconstruction model to acquire visibility complicates the proposed model, additional studies are needed for a more efficient method to obtain visibility information. In addition, the investigation of more diverse viewpoints and in-the-wild input images is another future work.

Figure 5: **Qualitative comparison with joint fitting on MPI-INF-3DHP.** The first column shows the input images. The second and third columns visualize the meshes reconstructed by LT-fitting and LMT, respectively.

Acknowledgement

This work was supported in part by NCSOFT, and in part by the Samsung Research Funding Center of Samsung Electronics under Project SRFCIT-1901-06.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, June 2014.

[2] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, pages 561–578, 2016.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *IJCV*, 2017.

[4] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020.

[5] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020.

[6] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Stefano Alletto, and Rita Cucchiara. Compressed volumetric heatmaps for multi-person 3d pose estimation. In *CVPR*, 2020.

[7] Richard I. Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *IJCV*, 103(3):267–305, 2013.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *CVPR*, pages 7779–7788, 2020.

[10] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 36(7):1325–1339, July 2014.

[11] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *ICCV*, 2019.

[12] Abdolrahim Kadkhodamohammadi and Nicolas Padoy. A generalizable approach for multi-view 3d human pose regression. *arXiv preprint arXiv:1804.10462*, 2018.

[13] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[15] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[16] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019.

[17] Vincent Leroy, Philippe Weinzaepfel, Romain Brégier, Hadrien Combaluzier, and Grégory Rogez. Smply benchmarking 3d human pose estimation in the wild. In *3DV*, 2020.

[18] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019.

[19] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021.

[20] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*, September 2014.

[22] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM TOG*, 34(6):248:1–248:16, Oct. 2015.

[23] Matthew M. Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. *ACM TOG*, 33(6):220:1–220:13, Nov. 2014.

[24] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. Rethinking the heatmap regression for bottom-up human pose estimation. In *CVPR*, pages 13264–13273, June 2021.

[25] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, Oct 2019.

[26] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017.

[27] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020.

[28] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019.

[29] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3D human pose annotations. In *CVPR*, 2017.

[30] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *ICCV*, 2019.

[31] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *ECCV*, pages 725–741, 2018.

[32] Nitin Saini, Elia Bonetto, Eric Price, Aamir Ahmad, and Michael J. Black. Airpose: Multi-view fusion network for aerial 3d human pose and shape estimation. *arXiv preprint arXiv:2201.08093*, 2022.

[33] Soyong Shin and Eni Halilaj. Multi-view human pose and shape estimation using learnable volumetric aggregation. *arXiv preprint arXiv:2011.13427*, 2020.

[34] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[35] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018.

[36] Denis Tome, Matteo Toso, Lourdes Agapito, and Chris Russell. Rethinking pose in 3d: Multi-stage refinement and recovery for markerless motion capture. In *3DV*, 2018.

[37] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[38] Tao Wang, Jianfeng Zhang, Yujun Cai, Shuicheng Yan, and Jiashi Feng. Direct multi-view multi-person 3d human pose estimation. *NeurIPS*, 2021.

[39] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[40] Size Wu, Sheng Jin, Wentao Liu, Lei Bai, Chen Qian, Dong Liu, and Wanli Ouyang. Graph-based 3d multi-person pose estimation using multi-view images. In *ICCV*, pages 11148–11157, October 2021.

[41] Yan Zhang, Michael J. Black, and Siyu Tang. We are more than our joints: Predicting how 3D bodies move. In *CVPR*, June 2021.

[42] Yuxiang Zhang, Zhe Li, Liang An, Mengcheng Li, Tao Yu, and Yebin Liu. Light-weight multi-person total capture using sparse multi-view cameras. In *ICCV*, 2021.

[43] Yang Zheng, Ruizhi Shao, Yuxiang Zhang, Tao Yu, Zerong Zheng, Qionghai Dai, and Yebin Liu. Deepmulticap: Performance capture of multiple characters using sparse multiview cameras. In *ICCV*, 2021.

[44] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, June 2019.

[45] Hao Zhu, Xinxin Zuo, Sen Wang, Xun Cao, and Ruigang Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *CVPR*, pages 4491–4500, 2019.

[46] Hao Zhu, Xinxin Zuo, Haotian Yang, Sen Wang, Xun Cao, and Ruigang Yang. Detailed avatar recovery from single image. *TPAMI*, 2021.