Letter

MeshLifter: Weakly Supervised Approach for 3D Human Mesh Reconstruction from a Single 2D Pose Based on Loop Structure

Sunwon Jeong and Ju Yong Chang *

Department of Electronics and Communication Engineering, Kwangju University,
Seoul 01897, Korea; bmycrew@kw.ac.kr

* Correspondence: jychang@kw.ac.kr; Tel.: +82-2-940-5136

Received: 30 June 2020; Accepted: 29 July 2020; Published: 30 July 2020

Abstract: In this paper, we address the problem of 3D human mesh reconstruction from a single 2D human pose based on deep learning. We propose MeshLifter, a network that estimates a 3D human mesh from an input 2D human pose. Unlike most existing 3D human mesh reconstruction studies that train models using paired 2D and 3D data, we propose a weakly supervised learning method based on a loop structure to train the MeshLifter. The proposed method alleviates the difficulty of obtaining ground-truth 3D data to ensure that the MeshLifter can be trained successfully from a 2D human pose dataset and an unpaired 3D motion capture dataset. We compare the proposed method with recent state-of-the-art studies through various experiments and show that the proposed method achieves effective 3D human mesh reconstruction performance. Notably, our proposed method achieves a reconstruction error of 59.1 mm without using the 3D ground-truth data of Human3.6M, the standard dataset for 3D human mesh reconstruction.

Keywords: 3D human mesh reconstruction; 3D human pose estimation; deep neural network; weakly supervised learning

1. Introduction

Nowadays, intelligent sensors such as the Microsoft Kinect can perform human body motion recognition and have been successfully used in various applications such as human–computer interaction, virtual reality, and intelligent robots. Moreover, the recent rapid development of data-driven approaches, including deep learning, has made it possible to use more general red, green, and blue (RGB) image sensors for human body motion analysis than depth sensors such as the Microsoft Kinect. In this paper, we address the problem of 3D human pose and shape reconstruction using a single monocular RGB sensor.

In the area of computer vision, research on 2D and 3D human pose estimation from a single RGB image have been improved considerably in recent years [1,2]. However, these studies only generate sparse keypoints of the human subject. We need dense shape information on the target subject to obtain a deeper understanding of the human image. Most recent studies use the 3D morphable model (3DMM) called skinned multi-person linear model (SMPL) [3] to reconstruct the 3D shape of a person from an image. SMPL parameterizes the variation of the 3D human mesh using low-dimensional latent variables, such as pose and shape.

The recent 3D human body mesh reconstruction method based on SMPL is typically classified into two classes. The first is an optimization-based method that minimizes the energy function to fit the parameterized body model to the 2D features extracted from the input image. This method has the advantage of accurately obtaining a human body mesh without training with a dataset.

However, this method has the following disadvantages. First, the optimization algorithm is sensitive to the initial point. If an appropriate initial point is not given, the optimization algorithm may fall into the local minima, which prevents a satisfactory reconstruction result. Second, the optimization process is generally very slow.

Methods using deep-learning-based regression networks have been proposed recently to overcome the disadvantages of the optimization-based approach. Deep learning networks run faster than optimization-based methods but have the following disadvantages. In most deep-learning-based methods, the network is trained using many pairs of inputs and outputs. Therefore, training a 3D human mesh reconstruction network based on SMPL requires a large dataset that includes many input images and their corresponding SMPL parameters. However, obtaining ground-truth SMPL parameters is generally very difficult. Therefore, most existing studies train the network indirectly using 3D poses instead of SMPL parameters. However, even 3D human poses are not easy to acquire in an in-the-wild environment. Hence, in this study, we propose a method to train a 3D human mesh reconstruction model without paired 2D and 3D data to solve the 3D data acquisition difficulty. Our method includes the following two contributions.

The first is MeshLifter, which is a deep learning model that outputs a 3D human mesh from an input 2D human pose and has the encoder–decoder structure. The encoder composed of residual blocks and fully connected layers outputs SMPL parameters from the input 2D human pose. The decoder with SMPL layer outputs 3D human mesh from the SMPL parameters generated from the encoder. A 3D pose can be also obtained from a 3D mesh using the pre-trained regression matrix included in the SMPL layer. Therefore, we can obtain the 3D mesh and 3D pose simultaneously from the input 2D human pose through the MeshLifter.

The second is a weakly supervised learning method based on a loop structure. The loop structure was first proposed in [4] to solve projection ambiguity where multiple 3D poses can be mapped to one 2D human pose. In [4], through the proposed loop structure, a lifting network that estimates a 3D pose from a 2D pose is trained without a 3D pose dataset. Our research goes further and proposes a method to learn MeshLifter, a model that can estimate 3D pose and 3D mesh from a single 2D pose. In the loop structure proposed in this study, 3D mesh lifting from a 2D pose through the MeshLifter and random rotation and 2D projection of a 3D pose computed through the MeshLifter are repeated, thereby providing a novel loop loss using only 2D pose data.

We reconstruct the 3D mesh by feeding the 2D pose estimated from the input image and not the ground-truth 2D pose into the MeshLifter to show the effectiveness of the proposed method. We use a general 2D human pose estimator based on a convolutional neural network (CNN). Through this experiment, we show that the MeshLifter can successfully reconstruct a 3D human mesh from a noisy input 2D human pose. In addition, through experiments using various datasets, such as Human3.6M [5], MPI-INF-3DHP [6], and MPII [7], we show that the proposed method achieves comparable performance with existing state-of-the-art methods. An overview of the proposed method is illustrated in Figure 1.

Figure 1. Overview of the proposed method.

2. Related Work

Recent 3D human body mesh reconstruction studies can be grouped into two main categories. The first is optimization-based methods that minimize the energy function to fit the parameterized body model to 2D features extracted from the input image. The second method regresses the parameters of the 3DMM directly from the input 2D feature through the deep neural network.

2.1. Optimization-Based SMPL Parameter Fitting

In [8], the 3D human mesh is reconstructed through the following two-step process. First, 2D joints are extracted from the input image through CNN. SMPL parameters are obtained through optimization of the proposed energy function using the extracted 2D joints. [9] is an extension of [8], which estimates SMPL parameters by minimizing the energy function based on body part segments and joint annotations. These optimization-based methods generally have high computational complexity and have the disadvantage of being easy to fall into local minima if a good initial point is not given.

2.2. Deep-Learning-Based SMPL Parameter Regression

Recently, many methods for directly regressing SMPL parameters using deep neural networks have been proposed. Kanazawa et al. [10] proposed a regression network that directly estimates SMPL parameters from image features extracted from an input image. This method also regularizes the output SMPL parameters through additional adversarial learning to enforce plausibility of the resultant 3D mesh. In [11], the extracted image feature and template mesh are fed into the graph CNN and the graph CNN generates intermediate mesh vertices. The shape and pose parameters are

regressed from the intermediate mesh vertices, which results in the finer human mesh. In [12], a data fusion module that allows not only RGB images but also RGB-D images to be used as input has been proposed. The module enables more robust mesh reconstruction. In addition, a probabilistic learning process has been proposed to simulate missing data in the process of using RGB-D data, which allows various datasets to be used for learning.

In [13], a method in which optimization and deep-learning-based regression are combined was proposed. In this method, the deep network regresses the SMPL parameter, which is used as the initial value of the iterative optimization routine [8] for fitting the SMPL model to 2D joints within the training loop. The SMPL parameter computed through optimization is used to explicitly supervise the deep network. The proposed method currently shows state-of-the-art performance and has been proved to be effective especially when 3D ground-truth is lacking or not available.

The proposed method belongs to a deep-learning-based direct regression approach. Therefore, it does not perform the optimization process, which requires proper initialization and has relatively high computational complexity. Also, unlike many methods that belong to the direct regression approach, our method does not require paired 2D and 3D data for training. Finally, our MeshLifter can use either a 2D human pose or an RGB image as input, unlike most other methods that only use RGB images to obtain a 3D human mesh. This flexibility in input makes the proposed method easy to use. A comparison of 3D human body mesh reconstruction methods, including the proposed method, is presented in Table 1.

Table 1. Comparison of our proposed method with other previous methods for 3D human body mesh reconstruction. “Optimization” indicates that the method depends on the optimization process, which requires parameter initialization and is generally slow. “Regression” indicates that the method is a deep neural network that requires large-scale training data. “Paired 2D–3D” indicates that paired 2D and 3D data should be used for training. “2D pose input” indicates that a 2D human pose can be used as the input to the method instead of an red, green, and blue (RGB) image.

Method	Optimization	Regression	Paired 2D–3D	2D Pose Input
SMPLify [8]	✓			✓
UP-3D [9]	✓			
HMR [10]		✓		
CMR [11]		✓	✓	
RGB-D [12]		✓	✓	
SPIN [13]	✓	✓		
Ours		✓		✓

3. Proposed Method

3.1. MeshLifter

MeshLifter consists of an encoder and a decoder and outputs a 3D mesh $M \in \mathbb{R}^{3 \times V}$ of $V(=6890)$ vertices and a 3D pose $\mathbf{P} \in \mathbb{R}^{3 \times N}$ from an input normalized 2D pose vector $\tilde{\mathbf{p}} \in \mathbb{R}^{2N}$ composed of pixel coordinates of $N(=14)$ joints. Figure 2 shows an overview of the MeshLifter structure composed of the sequential combination of encoder and decoder.

Figure 2. Overview of the MeshLifter.

3.1.1. SMPL

We use 3DMM to represent the 3D human shape, in which 3D geometry and texture variations of human subjects are represented by a low-dimensional parameter vector. SMPL is a 3DMM proposed in [3] to represent human body mesh. In SMPL, the principal component analysis is applied to thousands of 3D body scans to determine shape parameters $\beta \in \mathbb{R}^{10}$ and pose parameters $\theta \in \mathbb{R}^{72}$. First, the shape parameter β represents shape variations that the human body can have, such as fatness. The pose parameter θ represents a 72-dimensional ($3 \times 23 + 3 = 72$) vector that parameterizes the global rotation of the body mesh and the local rotations of 23 joints in an axis-angle fashion. The joints of the body mesh rotate according to the pose parameter, which enables a representation of the body shape that corresponds to various postures by deforming the mesh with the shape parameters.

3.1.2. Encoder

The input 2D pose $\{\mathbf{p}_i\}_{i=1}^N$ ($\mathbf{p}_i \in \mathbb{R}^2$) is first converted to the normalized 2D pose $\{\tilde{\mathbf{p}}_i\}_{i=1}^N$ with zero mean and unit variance. The normalized 2D pose vector $\tilde{\mathbf{p}} = [\tilde{\mathbf{p}}_1; \dots; \tilde{\mathbf{p}}_N]$ is then used as the input of the encoder:

$$\tilde{\mathbf{p}}_i = \frac{\mathbf{p}_i - \mathbf{m}}{\sigma}, \quad (1)$$

where $\mathbf{m}$ and σ denote mean vector and standard deviation, respectively, and are calculated as follows:

$$\mathbf{m} = \sum_{i=1}^N \frac{\mathbf{p}_i}{N}, \quad (2)$$

$$\sigma = \sqrt{\sum_{i=1}^N \frac{\|\mathbf{p}_i - \mathbf{m}\|^2}{N}}. \quad (3)$$

From the input normalized 2D pose vector, the encoder outputs low-dimensional feature vectors $\Theta = (\theta, \beta)$ that control the human mesh. θ and β represent the SMPL pose and shape parameters, respectively, and are fed into the decoder.

3.1.3. Decoder

The decoder generates the 3D mesh M from the shape and pose parameters output from the encoder based on the SMPL model. The decoder also computes the 3D pose $\mathbf{P}$ from the reconstructed mesh by using the pretrained regression matrix $W \in \mathbb{R}^{V \times N}$ as follows:

$$\mathbf{P} = MW. \quad (4)$$

The decoder consists of only differentiable operations and thus, the MeshLifter that contains it can be trained based on back-propagation.

3.2. Weakly Supervised Learning Based on Loop Structure

The ground-truth dataset is required for the input 2D pose and its corresponding output 3D pose or SMPL parameter to train the MeshLifter that outputs 3D human mesh and pose in a supervised learning manner. However, acquiring ground-truth data for 3D pose and SMPL parameters is generally not easy. Therefore, we propose the following method to learn the MeshLifter from a large 2D pose dataset in a weakly supervised manner using the loop structure introduced for unsupervised 3D human pose estimation in [4].

First, the overview of the loop structure is shown in Figure 3. The loop structure consists of two MeshLifters that share parameters, random rotation, and its inverse rotation, projection, and normalization processes. Random rotation rotates the 3D pose around the z-axis through the angle sampled from $[-60^\circ, 60^\circ]$, as shown in Figure 4. Projection converts the 3D pose into the 2D pose according to the orthographic projection model.

Figure 3. Overview of the loop structure.

Figure 4. Rotation around z-axis.

After the input 2D human pose p is normalized, it is fed into the MeshLifter to output the 3D pose P . A rotated 3D pose Q is obtained by applying a random rotation R to the output 3D pose P . Q is projected onto the 2D image plane through orthographic projection. After the projected 2D pose q is normalized by Equation (1), it is fed into the MeshLifter to output Q' . P' is obtained by applying the

inverse transform R^{-1} of the rotation R to the 3D pose $\mathbf{Q}'$. Finally, $\mathbf{P}'$ is projected onto the 2D image plane and then normalized to obtain $\mathbf{p}'$. If the MeshLifter is learned correctly, the normalized 2D pose $\mathbf{p}$ and the 2D pose $\mathbf{p}'$ obtained through the loop structure should be the same. Therefore, we define the loss L_{Loop} based on L1 norm as follows:

$$L_{Loop} = \|\tilde{\mathbf{p}} - \tilde{\mathbf{p}}'\|_1. \quad (5)$$

For the successful learning of the MeshLifter, the self-supervision loss L_{Self} that is motivated by self-supervised learning is introduced in addition to L_{Loop} . L_{Self} is defined as the difference between the input normalized 2D pose $\tilde{\mathbf{p}}$ and the 2D pose $\tilde{\mathbf{p}}''$ obtained by projecting the 3D pose $\mathbf{P}$ from the first MeshLifter onto 2D image plane and normalizing the result:

$$L_{Self} = \|\tilde{\mathbf{p}} - \tilde{\mathbf{p}}''\|_1. \quad (6)$$

This provides the additional constraint so that the MeshLifter produces a more accurate 3D pose.

3.3. Adversarial Training

3.3.1. Mesh Adversarial Training

The MeshLifter can be trained using the losses proposed above. However, these losses do not supervise the output SMPL parameters directly and thus, these losses alone cannot prevent the MeshLifter from outputting anthropometrically implausible body meshes. Therefore, we prevent this through adversarial training introduced in [10]. We consider the encoder of the MeshLifter as generator $\mathcal{G}$ and perform adversarial training using the discriminator network $\mathcal{D}$ for SMPL parameters. Discriminator $\mathcal{D}$ distinguishes whether the SMPL parameter output from the encoder corresponds to a real or fake human mesh, and generator $\mathcal{G}$ is trained to output a 3D shape and pose parameter that represents a plausible human mesh to deceive the discriminator.

We use different networks as discriminators for pose θ and shape β parameters, both of which are denoted by $\mathcal{D}_\theta$ and $\mathcal{D}_\beta$, respectively. $\mathcal{D}_\beta$ outputs the probability that input β corresponds to the real human mesh. $\mathcal{D}_\theta$ consists of a total of $K + 1$ discriminators that include K discriminators to learn the possible rotation range of joints, and a discriminator that determines the holistic plausibility of the mesh from full pose parameters. Here, $K (= 23)$ is the number of joints in the SMPL model, and is different from N which is the number of joints for the 2D and 3D pose. The loss function for generator $\mathcal{G}$ is as follows:

$$L_{Adv_{mesh}}(\mathcal{G}) = \sum_i \mathbb{E}_{\Theta \sim p_{\mathcal{G}}} \left[(\mathcal{D}_i(\mathcal{G}(\tilde{\mathbf{p}})) - 1)^2 \right], \quad (7)$$

where $\Theta = (\theta, \beta)$ and i represent the output SMPL parameters of the encoder and the index of the discriminator, respectively. The Mosh dataset [14] with ground-truth SMPL parameters is used to train the discriminator. The discriminator is trained to determine the parameter output from the encoder as fake and determine the parameter of the Mosh dataset as real. The cost function for discriminator learning is as follows:

$$L_{Disc_{mesh}}(\mathcal{D}_i) = \mathbb{E}_{\Theta \sim p_{mosh}} \left[(\mathcal{D}_i(\Theta) - 1)^2 \right] + \mathbb{E}_{\Theta \sim p_{\mathcal{G}}} \left[(\mathcal{D}_i(\mathcal{G}(\tilde{\mathbf{p}})))^2 \right]. \quad (8)$$

3.3.2. 2D Pose Adversarial Training

Additional adversarial training is performed using 2D human pose data, of which the ground-truth data is relatively easy to obtain. To this end, we consider the 2D pose $\tilde{\mathbf{q}}$ obtained by applying random rotation and projection to the 3D pose generated from the first MeshLifter as the output of the generator. The discriminator is trained to judge the $\tilde{\mathbf{q}}$ produced by the generator

as fake, and to judge the 2D pose $\tilde{\mathbf{r}}$ obtained through random sampling and normalization from the ground-truth 2D human pose dataset as real. Therefore, the loss function for the generator is as follows:

$$L_{Adv_{2d}}(\mathcal{G}) = \sum_i \mathbb{E}_{\tilde{\mathbf{q}}} \left[(\mathcal{D}_i(\tilde{\mathbf{q}}) - 1)^2 \right], \quad (9)$$

and the loss function for the discriminator is as follows:

$$L_{Disc_{2d}}(\mathcal{D}) = \mathbb{E}_{\tilde{\mathbf{r}}} \left[(\mathcal{D}(\tilde{\mathbf{r}}) - 1)^2 \right] + \mathbb{E}_{\tilde{\mathbf{q}}} \left[\mathcal{D}(\tilde{\mathbf{q}})^2 \right]. \quad (10)$$

3.4. Regularization

In addition to adversarial training, we adopt a regularization term for the plausibility of the reconstructed human mesh. The regularization term for this is defined as follows:

$$L_{Reg} = \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1, \quad (11)$$

where $\boldsymbol{\beta}$ is the shape parameter generated from the first MeshLifter and $\boldsymbol{\beta}_0$ is the shape parameter of the template mesh.

Therefore, the final loss function for training the proposed MeshLifter in a weakly supervised manner is as follows:

$$L_{Total} = L_{Loop} + \omega_1 L_{Self} + \omega_2 L_{Adv_{mesh}} + \omega_3 L_{Adv_{2d}} + \omega_4 L_{Reg}, \quad (12)$$

where $\omega_1, \omega_2, \omega_3, \omega_4$ are weights that control the relative importance of each loss constituting the total loss function. We set $\omega_1, \omega_2, \omega_3, \omega_4$ to 1.0, 2.0, 0.1, 0.05 in all our experiments, respectively. These numbers for weights were determined by the following simple greedy search. We first initialize the four weights. Then, after fixing the three weights, the optimal value for the other weight is selected using a small number of pre-sampled candidates. This process is repeated for each of the four weights.

4. Experiments

4.1. Datasets

The Human3.6M [5] and MPI-INF-3DHP [6] datasets containing RGB human images and corresponding ground-truth 3D human poses are used for training and evaluation of the proposed method. The Mosh [14] dataset containing only the ground-truth SMPL parameters without RGB images is also used for discriminator learning. The Human3.6M dataset [5] provides 3.6 million 2D and 3D human poses and their corresponding RGB images. To construct the dataset, 17 actions (e.g., discussion, smoking, taking the photo, ...) of 11 subjects were acquired through a motion capture system using four cameras. We used the subjects S1, S5, S6, S7, and S8 for training and the subjects S9 and S11 for testing according to the conventional protocol. We also sampled one frame every five frames and used it for an experiment to reduce the redundancy of the dataset. The MPI-INF-3DHP dataset [6] consists of approximately 100,000 learning images acquired through a markerless motion capture system indoors and approximately 3000 test images acquired indoors and outdoors. All images in the MPI-INF-3DHP dataset are annotated with 3D human poses. The Mosh dataset [14] was constructed by converting 3D human poses of subjects captured using a marker-based motion capture system into SMPL parameters. Approximately 410,000 SMPL parameters were used for our experiment. Lastly, the MPII dataset [7] containing in-the-wild images is used for the qualitative evaluation of the proposed method. The MPII dataset cannot be used for quantitative evaluation because only 2D poses are annotated. The descriptions of the datasets used in our experiment are summarized in Table 2.

Table 2. Description of the datasets used in our experiment.

Dataset	Human3.6M [5]	MPI-INF-3DHP [6]	Mosh [14]	MPII [7]
Data acquisition	Marker-based motion capture	Marker-less motion capture	Marker-based motion capture	YouTube search
2D image	✓	✓		✓
2D human pose	✓	✓		✓
3D human pose	✓	✓	✓	
SMPL parameters			✓	
Number of subjects	11	8	39	40K
Number of examples	3.6M	100K	410K	40K
Purpose of use	Training and evaluation	Training and evaluation	Adversarial training	Qualitative evaluation

4.2. Evaluation Metrics

Under the perspective projection assumption, the 3D shape can be reconstructed only up to a scale factor. Therefore, the proposed method cannot be used to determine the actual body size of the human subject. In consideration of this, we use the reconstruction error, which computes the mean per joint position error (MPJPE) after adjusting the scale and global rotation of the predicted 3D pose and ground-truth 3D pose, according to the Procrustes analysis [15], as the evaluation metric. MPJPE is defined as the average Euclidean distance between the predicted joint $\mathbf{P}_i$ and ground-truth joint $\mathbf{P}_i^*$ as follows:

$$MPJPE = \frac{1}{N} \sum_{i=1}^N \|\mathbf{P}_i - \mathbf{P}_i^*\|_2, \quad (13)$$

where i denotes the index of the joint.

4.3. Implementation Details

Our code is released at <https://github.com/sunwonlikeyou/MeshLifter>. The python 3.6 and PyTorch 1.2.0 [16] are used to implement the proposed method. The initial learning rate and the number of epochs are set to 1×10^{-4} and 100, respectively, to learn the MeshLifter and discriminator. The learning rate decays with a rate of 0.1 after the 50th epoch. Figure 5 shows the curves of all our losses during training. We can observe that the losses except $L_{Adv_{2d}}$ are minimized and converged. One exception, $L_{Adv_{2d}}$, increases with epoch and converges to a value of 1.0. It indicates that the generator, the MeshLifter, fails to produce realistic results that can deceive the 2D pose discriminator. Nevertheless, according to the ablation study in Section 4.4, $L_{Adv_{2d}}$ significantly improves the quantitative performance of the proposed method. We believe that this is because $L_{Adv_{2d}}$ works effectively as a kind of regularization term for plausible 2D poses. The implementation details for MeshLifter and discriminators are as follows.

Figure 5. Curves of all our losses during training.

MeshLifter has an encoder–decoder structure. The encoder consists of linear layer, ReLU [17], dropout [18], batch-norm [19], and residual connection [20]. The decoder includes an SMPL layer composed of only differentiable operations and thus, its parameters can be updated through back-propagation.

Discriminator for SMPL parameter is composed of two networks, $\mathcal{D}_\theta$ and $\mathcal{D}_\beta$, that correspond to pose and shape parameters as described in Section 3. $\mathcal{D}_\beta$ includes two fully connected layers composed of 10 and 5 hidden units and one ReLU layer. $\mathcal{D}_\theta$ includes a Rodrigues layer, two convolutional layers, and two branches. The Rodrigues layer converts a pose parameter expressed in the axis-angle format to a 3×3 rotation matrix according to the Rodrigues formula. The two convolutional layers consist only of 1×1 size convolution filters, and the number of input and output channels is (9,32) and (32,32), respectively. The first branch contains fully connected layers composed of 736, 1024, and 1024 hidden units. The second branch contains fully connected layers with 32 hidden units for all joints.

Discriminator for 2D pose consists of two consecutive residual blocks and two fully connected layers for input and output of the network. Each residual block includes fully connected layer with 3000 hidden units, batch-norm, dropout, ReLU, and residual connection.

The 2D pose predicted from the RGB image using CNN is used as the input of MeshLifter for a fair comparison with previous methods. The ResNet101 [20] model is used as a backbone network for 2D pose estimation, and the last layer is modified to output a heatmap of 64×64 resolution. We also added the soft-argmax layer [21] to obtain continuous 2D joint coordinates free from quantization error from the heatmap. The network outputs the 2D human pose $\mathbf{p}$ from an input RGB image of 256×256 size. Table 3 shows the performance of the 2D human pose estimation.

Table 3. Performance of 2D human pose estimation. The numbers denote mean Euclidean distances in pixels.

Dataset	Pixel Error
Human3.6M	3.2
MPII	6.3

4.4. Ablation Study

An ablation study is performed to investigate the effects of the proposed losses on the performance of our model, and Table 4 shows its quantitative results. The MeshLifter trained with only L_{Self} is considered as the baseline. Loop, Mesh, 2D, and Reg indicate L_{Loop} , $L_{Adv_{mesh}}$, $L_{Adv_{2d}}$, and L_{Reg} are used to learn the MeshLifter, respectively.

Table 4 shows that L_{Loop} , $L_{Adv_{mesh}}$, and $L_{adv_{2d}}$ are significantly helpful for improving the performance of the MeshLifter. According to Table 4, the results obtained using all loss functions

except L_{Reg} show the best quantitative performance. However, Figure 6 shows that L_{Reg} plays an important role in the qualitative performance of the reconstructed human mesh. The left, middle, and right columns in Figure 6 show the input image, the 3D mesh output by the MeshLifter trained without L_{Reg} , and the 3D mesh output by the MeshLifter trained with L_{Reg} , respectively. The use of L_{Reg} prevents the monstrous mesh output and helps in reconstructing the anthropometrically plausible human mesh.

Table 4. Ablation experiments with various combinations of losses. The numbers denote reconstruction errors in mm.

Loss Variations	Reconstruction Error
Self (baseline)	157.0
Self + Loop	136.1
Self + Loop + Mesh	83.5
Self + Loop + Mesh + 2D	58.8
Self + Loop + Mesh + 2D + Reg	59.1

Figure 6. Input images (*left*), and the reconstruction results of using (*right*) and not using (*middle*) the regularization term.

4.5. Quantitative Result

Tables 5 and 6 provide the quantitative results of the proposed method and recent existing methods for the Human3.6M and MPI-INF-3DHP datasets, respectively. They show the proposed method achieved the state-of-the-art performance among methods that do not use 3D pose data as direct supervision for learning. This result shows that the proposed weakly supervised method based on the loop structure effectively learns our MeshLifter.

Table 5. Quantitative results of the proposed model and the existing state-of-the-art methods for the Human3.6M dataset. The numbers denote reconstruction errors in mm.

Method	Reconstruction Error
SMPLify [8]	82.0
Pavlakos et al. [22]	75.9
HMR-unpaired [10]	66.5
SPIN-unpaired [13]	62.0
Ours	59.1

Table 6. Quantitative results of the proposed model and the existing state-of-the-art methods for the MPI-INF-3DHP dataset. The numbers denote reconstruction errors in mm.

Method	Reconstruction Error
HMR-unpaired [10]	113.2
VNect [23]	98.0
Ours	96.0
SPIN-unpaired [13]	80.4

4.6. Qualitative Result

Figures 7–9 show the qualitative results of the proposed method for Human3.6M, MPI-INF-3DHP, and MPII datasets, respectively. The figures show that the proposed method can successfully reconstruct 3D human meshes from various input images acquired in controlled and in-the-wild environments.

Figure 7. Qualitative results on the Human3.6M dataset.

Figure 8. Qualitative results on the MPI-INF-3DHP dataset.

Figure 9. Qualitative results on the MPII dataset.

4.7. 3D Hand Mesh Reconstruction

Additional experiments for hand mesh reconstruction are conducted to investigate the general applicability of the proposed method. To this end, we used MANO [24], a 3DMM for human hands, as a decoder for the MeshLifter. For training and evaluation, we used the Rendered Handpose Dataset (RHD) [25], which includes 24,619 training images and 1459 test images. In the case of hands, a dataset that provides ground-truth 3DMM parameters, such as the Mosh dataset for the body, is not available, and thus, the mesh adversarial training in Section 3.3 cannot be performed. Therefore, we introduce the following regularization term that constrains pose and shape parameters together:

$$L_{Reg_{mano}} = \|\Theta_h - \Theta_0\|_1, \quad (14)$$

where $\Theta_h = [\theta_h; \beta_h] \in \mathbb{R}^{58}$ is the MANO shape and pose parameters output from the encoder, and $\Theta_0 = [\theta_0; \beta_0]$ is the shape and pose parameters of the template hand mesh. Finally we train the MeshLifter for hands using the following loss function:

$$L_{Total} = L_{Loop} + \omega_1 L_{Self} + \omega_2 L_{Adv_{2d}} + \omega_3 L_{Reg_{mano}}, \quad (15)$$

where ω_1 , ω_2 , and ω_3 are set to 1.0, 0.3, and 0.1, respectively.

Table 7 shows a quantitative comparison between the proposed method and existing studies for 3D hand pose estimation. As an evaluation metric for comparison, the reconstruction error is used as in the body. Table 7 shows that the proposed method does not achieve the best performance quantitatively. However, all methods except the proposed method train the network in a supervised manner, and output only sparse 3D hand joints. Meanwhile, the proposed method generates a dense 3D hand mesh in a weakly supervised fashion, which shows the effectiveness of the proposed method. In addition, Figure 10 shows that the proposed method performs qualitatively successful 3D hand mesh reconstruction.

Table 7. Quantitative comparison with existing methods on the Rendered Handpose Dataset (RHD) dataset. The numbers denote reconstruction errors in mm.

Method	Reconstruction Error
Zimmermann and Brox [25]	30.42
Yang and Yao [26]	19.95
Spurr et al. [27]	19.73
Yang et al. [28]	13.14
Ours	14.02

Figure 10. Qualitative results on the RHD dataset.

4.8. Discussion

In this subsection, we present the usability of the proposed method, limitations, and future works to overcome them. Our proposed method can reconstruct the 3D mesh of the target human object in the form of SMPL parameters. By using the Equation (4), 3D joints can also be obtained from the reconstructed mesh. This 3D skeleton information is used for gesture or action recognition and can be applied to various fields such as human–computer interaction and visual surveillance. Meanwhile, the SMPL parameters reconstructed by the proposed method directly include 3D rotation information of limbs that make up the body beyond merely 3D coordinates of body joints. The rotation

information enables motion retargeting between characters and can be used for computer graphics and augmented/virtual reality.

The proposed method uses a single 2D human pose to reconstruct the 3D mesh of the target person. This 2D pose alone does not provide enough information to obtain reliable body shape information, so the proposed method relies heavily on the regularization term to estimate the SMPL shape parameters encoding human body shape information. Therefore, we plan to investigate how additional image features other than 2D poses can have an advantage in estimating shape parameters. Also, the proposed 3D human reconstruction method relies on a single 2D pose or a single RGB image. It makes the temporal prior, which can alleviate the ambiguity of the 3D reconstruction problem, unavailable in the proposed method. It also makes the proposed method produce results that lack temporal consistency when applied to an input 2D pose or image sequence. Therefore, our next future work is to extend the proposed method to 2D pose sequence input and pursue a method in which the proposed model can adopt the temporal prior.

5. Conclusions

In this study, we addressed the problem of reconstructing a 3D human mesh from a single 2D human pose. We proposed the MeshLifter, a network that can output 3D mesh and 3D pose from a single 2D pose, and the loop-structure-based method to learn it in a weakly supervised manner. We confirmed that the proposed method efficiently generates a 3D human mesh from a 2D human pose without direct 3D supervision through various experiments and showed that it has comparable performance to the state-of-the-art methods. The MeshLifter trained by our proposed weakly supervised learning method achieves a reconstruction error of 59.1mm for the Human3.6M dataset, the standard dataset for 3D human mesh reconstruction. Through additional experiments, we also showed that our method could be used to reconstruct human hands.

Author Contributions: methodology, S.J.; software, S.J.; validation, S.J.; investigation, S.J.; writing—original draft preparation, S.J.; writing—review and editing, J.Y.C.; visualization, S.J.; supervision, J.Y.C.; project administration, J.Y.C.; funding acquisition, J.Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Samsung Research Funding Center of Samsung Electronics (No. SRFC-IT1901-06) and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1C1C1008462). The present research has been conducted by the Research Grant of Kwangwoon University in 2020.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, Y.; Tian, Y.; He, M. Monocular human pose estimation: A survey of deep learning-based methods. *Comput. Vis. Image Underst.* **2020**, *192*, 102897. [[CrossRef](#)]
2. Zheng, X.; Chen, X.; Lu, X. A joint relationship aware neural network for single-image 3d human pose estimation. *IEEE Trans. Image Process.* **2020**, *29*, 4747–4758. [[CrossRef](#)] [[PubMed](#)]
3. Loper, M.; Mahmood, N.; Black, M.J. SMPL: A skinned multi-person linear model. *Acm Trans. Graph. TOG* **2015**, *34*, 1–16. [[CrossRef](#)]
4. Chen, C.-H.; Tyagi, A.; Agrawal, A.; Drover, D.; Stojanov, S.; Rehg, J.M. Unsupervised 3d pose estimation with geometric self-supervision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
5. Ionescu, C.; Papava, D.; Olaru, V.; Sminchisescu, C. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1325–1339. [[CrossRef](#)] [[PubMed](#)]
6. Mehta, D.; Rhodin, H.; Casas, D.; Fua, P.; Sotnychenko, O.; Xu, W.; Theobalt, C. Monocular 3d human pose estimation in the wild using improved cnn supervision. In Proceedings of the International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017.

7. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2d human pose estimation: New benchmark and state of the art analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3686–3693.
8. Bogu, F.; Kanazawa, A.; Lassner, C.; Gehler, P.; Romero, J.; Black, M.J. Keep it SMPL: Automatic estimation of 3d human pose and shape from a single image. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016.
9. Lassner, C.; Romero, J.; Kiefel, M.; Bogu, F.; Black, M.J.; Gehler, P.V. Unite the people: Closing the loop between 3d and 2d human representations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
10. Kanazawa, A.; Black, M.J.; Jacobs, D.W.; Malik, J. End-to-end recovery of human shape and pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
11. Kolotouros, N.; Pavlakos, G.; Daniilidis, K. Convolutional mesh regression for single-image human shape reconstruction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
12. Li, R.; Cai, C.; Georgakis, G.; Karanam, S.; Chen, T.; Wu, Z. Towards robust rgb-d human mesh recovery. *arXiv* **2019**, arXiv:1911.07383.
13. Kolotouros, N.; Pavlakos, G.; Black, M.J.; Daniilidis, K. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 2252–2261.
14. Loper, M.; Mahmood, N.; Black, M.J. MoSh: Motion and shape capture from sparse markers. *ACM Trans. Graph. TOG* **2014**, *33*, 1–13. [[CrossRef](#)]
15. Gower, J.C. Generalized procrustes analysis. *Psychometrika* **1975**, *40*, 33–51. [[CrossRef](#)]
16. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8024–8035.
17. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010; pp. 807–814.
18. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
19. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
21. Sun, X.; Xiao, B.; Wei, F.; Liang, S.; Wei, Y. Integral human pose regression. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 529–545.
22. Pavlakos, G.; Zhu, L.; Zhou, X.; Daniilidis, K. Learning to estimate 3d human pose and shape from a single color image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 459–468.
23. Mehta, D.; Sridhar, S.; Sotnychenko, O.; Rhodin, H.; Shafiei, M.; Seidel, H.P.; Xu, W.; Casas, D.; Theobalt, C. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Trans. Graph. TOG* **2017**, *36*, 1–14. [[CrossRef](#)]
24. Romero, J.; Tzionas, D.; Black, M.J. Embodied hands: Modeling and capturing hands and bodies together. *ACM Trans. Graph. TOG* **2017**, *36*, 245. [[CrossRef](#)]
25. Zimmermann, C.; Brox, T. Learning to estimate 3d hand pose from single rgb images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4903–4911.
26. Yang, L.; Yao, A. Disentangling latent hands for image synthesis and pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 9877–9886.

27. Spurr, A.; Song, J.; Park, S.; Hilliges, O. Cross-modal deep variational hand pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 89–98.
28. Yang, L.; Li, S.; Lee, D.; Yao, A. Aligning latent spaces for 3d hand pose estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 2335–2343.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).