

Article

Towards Single 2D Image-Level Self-Supervision for 3D Human Pose and Shape Estimation

Junuk Cha ¹, Muhammad Saqlain ¹, Changhwa Lee ², Seongyeong Lee ², Seungeun Lee ², Donguk Kim ¹, Won-Hee Park ^{3,*} and Seungryul Baek ^{1,*}

¹ AI Graduate School, Ulsan National Institute of Science and Technology, Ulsan 44919, Korea; jucha@unist.ac.kr (J.C.); m.saqlain1240@yahoo.com (M.S.); dukim@unist.ac.kr (D.K.)

² Department of Computer Science and Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, Korea; changhwalee@unist.ac.kr (C.L.); skwithu@unist.ac.kr (S.L.); selee@unist.ac.kr (S.L.)

³ Railway Safety Research Division, Korea Railroad Research Institute, Uiwang-si 16105, Korea

* Correspondence: whpark@krri.re.kr (W.-H.P.); srbaek@unist.ac.kr (S.B.)

Abstract: Three-dimensional human pose and shape estimation is an important problem in the computer vision community, with numerous applications such as augmented reality, virtual reality, human computer interaction, and so on. However, training accurate 3D human pose and shape estimators based on deep learning approaches requires a large number of images and corresponding 3D ground-truth pose pairs, which are costly to collect. To relieve this constraint, various types of weakly or self-supervised pose estimation approaches have been proposed. Nevertheless, these methods still involve supervision signals, which require effort to collect, such as unpaired large-scale 3D ground truth data, a small subset of 3D labeled data, video priors, and so on. Often, they require installing equipment such as a calibrated multi-camera system to acquire strong multi-view priors. In this paper, we propose a self-supervised learning framework for 3D human pose and shape estimation that does not require other forms of supervision signals while using only single 2D images. Our framework inputs single 2D images, estimates human 3D meshes in the intermediate layers, and is trained to solve four types of self-supervision tasks (i.e., three image manipulation tasks and one neural rendering task) whose ground-truths are all based on the single 2D images themselves. Through experiments, we demonstrate the effectiveness of our approach on 3D human pose benchmark datasets (i.e., Human3.6M, 3DPW, and LSP), where we present the new state-of-the-art among weakly/self-supervised methods.

Keywords: deep learning; human body pose estimation; human body mesh estimation; neural rendering; self-supervised learning



Citation: Cha, J.; Saqlain, M.; Lee, C.; Lee, S.; Lee, S.; Kim, D.; Park, W.-H.; Baek, S. Towards Single 2D Image-Level Self-Supervision for 3D Human Pose and Shape Estimation. *Appl. Sci.* **2021**, *11*, 9724. <https://doi.org/10.3390/app11209724>

Academic Editor: Athanasios Nikolaidis

Received: 31 August 2021

Accepted: 12 October 2021

Published: 18 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Tremendous progress has been made on estimating 3D human poses and shapes from a single image [1–12]. In this context, deep learning-based approaches have been successful over the last decades [1–8]. It was initially decided to estimate the 2D/3D skeletal representation of the human bodies [1–4]. The skeletal representation is efficient to describe key characteristics of the human motions, although the shape information is lacking. Recently, Kanazawa et al. [5] proposed the estimation of both poses and shapes of the human bodies by incorporating a differentiable 3D mesh representation—i.e., a skinned multi-person linear model (SMPL) [13]—in the deep learning framework. More recently, several frameworks that improve the 3D mesh estimation network [5] have been proposed to deal with temporal consistency [6], multi-person cases [7], domain differences [8], and so on.

One of the fundamental issues in constructing pose estimation frameworks is that they consume large numbers of 2D or 3D ground-truth poses (i.e., x, y, and z coordinate values of the 3D human bodies) for a given 2D RGB (i.e., red, green, blue) or 2.5D depth

input image to secure good accuracy in the mesh estimation task. Many researchers have proposed million-scale data pairs to properly train such frameworks [14]. However, it is challenging to acquire large-scale datasets containing diverse variations and quality 3D annotations. Manually annotating such 3D coordinate values is non-trivial, and it takes a great deal of time and manual effort. One team attempted to relieve the issue by using synthetic datasets based on graphics engines [15,16]. However, the appearance of images obtained from the graphics engines showed an observable gap to the real samples [17], and it is well known that the models trained by pure synthetic datasets do not generalize well to the real testing datasets [18].

In this paper, we attempted to relieve the issue of insufficient data by proposing self-supervision losses to train the 3D human pose estimation framework without explicit 2D/3D skeletal ground-truths. Our self-supervision losses consist of four types of supervision whose ground-truths are defined based on single 2D images: jigsaw puzzling [19], rotation prediction [20], image in-painting [21], and image projection [22] tasks (as illustrated in Figure 1). In the jigsaw puzzling task, the image is divided into the $D \times D$ tiles and shuffled; then, the jigsaw puzzle solver is trained to estimate the order of the tiles. In the rotation prediction task, the image is rotated and the rotation predictor is trained to estimate the rotation degrees. In the image in-painting task, sub-patches of the image are erased and the image in-painting decoder is learned to generate the erased patches with the aid of the adversarial loss [23]. Lastly, in the image projection task, the neural 3D mesh renderer is used to differentially generate the 2D images by projecting the estimated 3D meshes. This task directly provides gradients to the estimated 3D meshes and updates the parameters of the 3D mesh estimator, while the former three tasks indirectly enrich the feature vector extracted from the 2D images via the feature extractor. Via the combination of the proposed four losses, the 3D mesh estimator produces more accurate 3D human meshes.

Via a series of experiments, we have observed that the proposed self-supervised losses with additional image databases are able to enrich the pre-trained 3D human pose estimator and achieve state-of-the-art accuracy among self/weakly/semi-supervised works. The major contributions of this paper are summarized as follows:

- We construct a 3D human pose and shape estimation framework that could be trained by 2D single image-level self-supervision without the use of other forms of supervision signals, such as explicit 2D/3D skeletons, video-level, or multi-view priors;
- We propose four types of self-supervised losses based on the 2D single images themselves and introduce a method to effectively train the entire networks. In particular, we investigate which are the most promising combinations of losses to effectively achieve the 2D single image-level self-supervision for 3D human mesh reconstruction;
- The proposed method outperforms the competitive 3D human pose estimation algorithms, proving that leveraging single 2D images could be used for strong supervision to train networks for the 3D mesh reconstruction task.

We have made our code and data publicly available at <https://github.com/JunukCha/SSPSE> accessed on October 13, 2021.

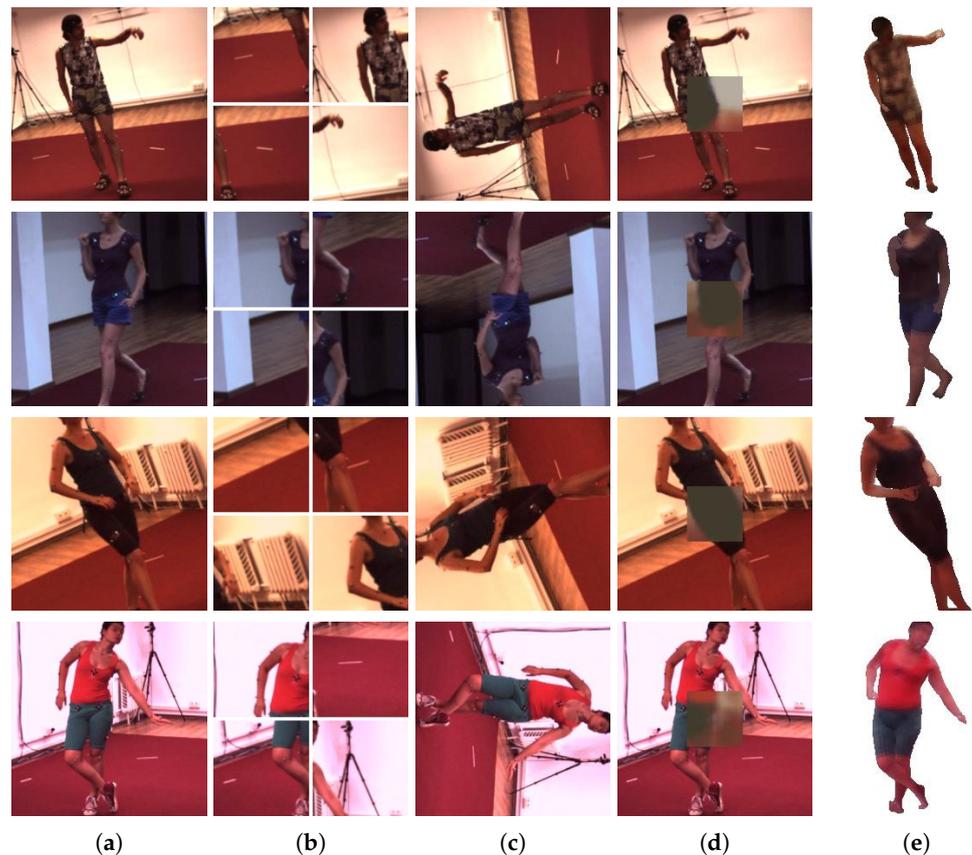


Figure 1. Example image-level self-supervision: We applied three types of image manipulation and one image projection task to train our human mesh estimation network. Each row shows examples from the Human3.6M dataset, and each column corresponds to (a) input RGB images, (b) shuffled patches used for the jigsaw puzzling task, (c) rotated RGB images used for rotation prediction task, (d) output images from part inpainting task, and (e) neural rendered images from estimated 3D meshes and textures, respectively.

2. Related Works

In this section, we review the recent literature on 3D human pose estimation and 3D human pose and shape estimation works. Then, we analyze recent weak/semi-supervised approaches designed for 3D human pose and shape estimation that are closely related to ours.

2.1. Three-Dimensional Human Pose Estimation

Many recent studies have focused on estimating 2D [11] or 3D keypoint locations on the human body [12]. Normally, these keypoint locations include major body joints such as the wrists, ankles, elbows, neck, shoulders, and knees. The architectures of 2D pose detectors have been designed to map images into the 2D pose vector more accurately. Wei et al. [2] proposed a sequential architecture composed of convolutional neural networks (CNNs) with multiple sizes of receptive fields. Zhou et al. [3] proposed a method using both a 3D geometric prior and temporal smoothness prior to treat considerable uncertainties in 2D joint locations. Newell et al. [1] proposed stacked hourglass networks based on the successive architecture, which is composed of pooling and upsampling layers. As we are living in the 3D space, understanding poses in the 3D space is the natural extension to 2D pose estimation. Three-dimensional pose estimation aims to locate the key 3D joints of the human body from an image or a video, which are either in the single-view or multi-view setting. Most approaches are based on the two-stage approaches that first predict 2D joint locations using 2D pose detectors [1–3,24] and then predict 3D joint locations

from the 2D joints or features by performing regression [4,25–27] or model fitting [3,28–32]. Xu et al. [24] proposed a graph stacked hourglass network for 3D human pose estimation, which consists of four stacked hourglasses. Estimated 2D joints are fed into the network, and the 3D pose is predicted. Martinez et al. [4] proposed a simple framework composed of fully connected layers to lift a 2D pose to a 3D angular pose. Moreno et al. [25] proposed a framework composed of a 2D joint detector based on CNN and inferred 3D poses from the detected 2D joints. These approaches have made great progress in improving the performance of 3D human pose estimation.

2.2. Three-Dimensional Human Pose and Shape Estimation

Followed by progresses in the field of 3D human pose estimation, simultaneously estimating the human body pose and shape has become a recent trend [10,33–41]. There are two main approaches for estimating 3D human poses and shapes (i.e., meshes): one is the optimization-based methods [9,42,43], and the other is the regression-based methods [5–8,36,37]. SMPLify [9] is one of the representative optimization-based methods. It first estimates the 2D skeletons from the images and fits the graphical 3D body model (i.e., SMPL [13]) to it using the optimization method. Kanazawa et al. [5] proposed the deep learning-based regression framework. This is an end-to-end framework for reconstructing a 3D human mesh from a single RGB image. Omran et al. [36] proposed neural body fitting. This framework generates 12 semantic human body parts with a semantic segmentation CNN. From these semantic human body parts, pose and shape parameters of SMPL are estimated and a 3D human mesh is reconstructed. Pavlakos et al. [37] proposed a framework that estimates heatmaps related to pose parameters of SMPL and silhouettes related to shape parameters of SMPL simultaneously. Using these heatmaps and silhouettes, a 3D human mesh is reconstructed. A method [44] has also been introduced that exploits the merits of both optimization-based and regression-based methods for 3D mesh estimation. The temporal dynamics of the human body and shape have also been incorporated in the framework of [6]. The framework used to overcome the adaptation of a pre-trained model of human mesh estimation to the out-of-domain field was proposed in [8]. In addition, non-parametric body mesh reconstruction methods have been proposed [38,45]. Tan et al. [38] proposed an encoder–decoder network: first, a decoder is trained to predict a body silhouette from the SMPL parameter input; second, an encoder is trained on a real image and corresponding silhouette pairs while the decoder is fixed. Lin et al. [45] proposed end-to-end human pose and mesh reconstruction with a transformer [46].

2.3. Weakly/Semi-Supervised Learning in 3D Human Mesh Estimation

For most human pose estimation methods [36,37,43,47–50], supervised learning prevails; however, securing the 3D mesh ground truth is non-trivial. Weakly or semi-supervised methods are designed to solve the issue of the lack of quality annotation by using the available easier annotations. In the context of the 3D human mesh estimation task, semi-supervised learning uses 3D skeletons that are coarser than 3D meshes, while weakly-supervised learning uses either 2D annotation [36,51,52] or pseudo-3D annotation [53–56]. In [36], 2D keypoint annotations are exploited to estimate SMPL body model parameters from CNNs to recover human 3D meshes following predicted body part segmentation masks. In [51], two anatomically inspired losses are proposed and used with a weakly supervised learning framework to jointly learn from large-scale in-the-wild 2D and indoor/synthetic 3D data. The authors of [52] suggest weakly-supervised solutions by adopting multi-view consistency loss in 2.5D pose representations leveraged by independent 2D annotated samples. In [53–56], multi-view geometry is used to resolve 3D depth ambiguity. We developed our original methods in the self-supervised setting only using the input 2D images; however, to compare with these weakly or semi-supervised approaches, we extended our model to be additionally learned by 2D or 3D skeletons if they are available. In this manner, we compared our method to the state-of-the-art self/weakly/semi-supervised approaches.

generates 2048-dimensional feature vectors $\mathbf{f} \in F$ from the input image \mathbf{x} , whose size is resized to a $224 \times 224 \times 3$ dimensional array.

Mesh estimator $f^{\text{Mesh}} : F \rightarrow M \subset \mathbb{R}^{6890 \times 13,776}$. After extracting the feature vector $\mathbf{f} \in F$, we mapped it to the corresponding 85-dimensional SMPL [13] parameters $\mathbf{h} \in H \subset \mathbb{R}^{85 \times 1}$ (we used 3, 10, and 72-dimensional vectors as camera, shape, and pose parameters, respectively, in the same manner as in [5]). These parameters were used to differentially generate the corresponding human body meshes $\mathbf{m} \in M$ with 6890 vertices and 13,776 faces using the SMPL layer (refer to [5,13] for more details).

Jigsaw puzzle solver $f^{\text{Jigsaw}} : F \times F \times F \times F \times F \rightarrow O \subset \mathbb{R}^{C \times 1}$. The jigsaw puzzle solver was used to solve the first self-supervision task, called the jigsaw puzzling task. In this task, we divided the input 2D image \mathbf{x} into $D \times D$ tiles and permuted them to generate a jigsaw puzzling image $\mathbf{x}_{\text{jigsaw}}$. Then, we input D^2 tiles combined with the original image into the feature extractor f^{Feat} by resizing all images into $224 \times 224 \times 3$ dimensional images. The resultant features from five images were concatenated, retaining the permuted orders, and mapped into the vector $\mathbf{o} \in O$, whose entry denotes the probability for the permutation order of the tiles in the whole images. The class number $C = (D \times D)!$ is equivalent to the number of permutations for $D \times D$ tiles. We found that $D = 2$ best encodes the human bodies, balancing it from the background regions (see Figure 1); thus, $C = (2 \times 2)! = 24$ was used throughout the experiment.

Rotation predictor $f^{\text{Rot}} : F \rightarrow R \subset \mathbb{R}^{R \times 1}$. The rotation predictor was used to solve the second self-supervision task, called the rotation prediction task. In this task, we rotated the input 2D image \mathbf{x} with R fixed angles θ to generate a new image \mathbf{x}_{rot} . Humans are in the upright position in images with 0 degrees. We first applied f^{Feat} on the input image \mathbf{x}_{rot} to obtain the feature vector \mathbf{f} . Then, the rotation predictor f^{Rot} was applied on \mathbf{f} to predict the vector $\mathbf{r} \in R$, which contained the probability for R possible angles. In this work, we set $R = 4$ by setting the rotation angles to 0, 90, 180, and 270 for simplicity; however, we were able to achieve a meaningful accuracy improvement using this simple setting.

Part in-painter $f^{\text{Inp}} : F \rightarrow P \subset \mathbb{R}^{P \times P \times 3}$. The part in-painter was used to solve the third self-supervision task, called the part in-painting task. In this task, we erased a $P \times P \times 3$ -sized patch \mathbf{p} from the center of an input image \mathbf{x} , resulting in \mathbf{x}_{hole} . Then, the part in-painter f^{Inp} was responsible for predicting $\hat{\mathbf{p}}$ resembling \mathbf{p} . We set $P = 64$ considering the patch size compared to the size of the human bodies in images (Figure 1d describes the example in-painted patches).

Neural mesh renderer $f^{\text{NMR}} : T \times M \rightarrow X \times S \subset \mathbb{R}^{224 \times 224 \times 3} \times \mathbb{R}^{224 \times 224}$. The neural mesh renderer [22] was employed to solve the last self-supervision task, called the image projection task. In this task, 3D meshes were projected to the 2D images by an operation similar to graphics rendering [22]. Either the RGB human images $\mathbf{x} \in X$ or a binary segmentation mask $\mathbf{s} \in S$ could be generated from the 3D meshes \mathbf{m} . To render RGB images, an additional texture array $\mathbf{t} \in T$ was required that described RGB values for faces of the 3D meshes. To infer the texture array, we additionally involved the texture network f^{Tex} , as explained below.

Texture network $f^{\text{Tex}} : F \rightarrow T \subset \mathbb{R}^{13,776 \times 3}$. As there are 13,776 faces in the SMPL [13] model, the dimension of the texture array \mathbf{t} is $13,776 \times 3$, as it represents RGB values corresponding to each mesh face. Furthermore, we first inferred the $13,776 \times 3$ dimensional array \mathbf{t} from f^{Tex} and differentially reshaped this towards a $13,776 \times Z \times Z \times 3$ dimensional array. $Z = 2$ was used as the visual quality based on this minimum dimension, which was sufficient for our purpose.

Discriminator $f^{\text{Disc}} : \mathbf{x} \rightarrow [0, 1]$. The discriminator was involved to capture the distribution of the real human images and give gradients to the networks to make the realistic images, in a similar manner to [23]. The discriminator f^{Disc} played a crucial role, especially when

training the part in-painter f^{Inp} and texture network f^{Tex} , as these tasks require training networks to generate the realistic patches or textured images.

Tables 1–3 present the detailed structure of our network architectures. For $f^{\text{r-cnn}}$, f^{NMR} , and f^{Feat} , we employed the implementation and weights from [5,22,58], respectively. For the mesh estimator f^{Mesh} as explained in Table 4, we involved the SMPL layer of [5], which can output the SMPL human body meshes from its estimated pose (72), shape (10), and camera (3) parameters.

Table 1. Architecture of jigsaw puzzle solver f^{jigsaw} . Inputs are concatenated feature maps of four tiles of x_{jigsaw} and an original image x , obtained from f^{Feat} .

Layer	Operation	Kernel	Dimensionality
	Input: Feature map	-	10,240 × 1
1	Linear + ReLU + Dropout (0.5)	-	2048 × 1
2	Linear	-	24 × 1

Table 2. Architecture of rotation predictor f^{Rot} . Input is a feature map of the rotated image from $f^{\text{Feat}}(x_{\text{rot}})$.

Layer	Operation	Kernel	Dimensionality
	Input: Feature map	-	10,240 × 1
1	Linear + ReLU + Dropout(0.5)	-	2048 × 1
2	Linear	-	24 × 1

Table 3. Architecture of part in-painter f^{Inp} . Input is a feature map of the center hole image from $f^{\text{Feat}}(x_{\text{hole}})$.

Layer	Operation	Kernel	Dimensionality
	Input: Feature map	-	2048 × 1
1	ConvT. + B.N. + ReLU	4 × 4	4 × 4 × 512
2	ConvT. + B.N. + ReLU	4 × 4	8 × 8 × 256
3	ConvT. + B.N. + ReLU	4 × 4	16 × 16 × 128
4	ConvT. + B.N. + ReLU	4 × 4	32 × 32 × 64
5	ConvT. + Tanh	4 × 4	64 × 64 × 3

Table 4. Architecture of mesh estimator f^{Mesh} .

Layer	Operation	Kernel	Dimensionality
	Input: Feature map + Pose, Shape, Camera param.	-	2205
1	Linear + Dropout (0.5)	-	1024
2	Linear + Dropout (0.5)	-	1024
3-1	Linear	-	72
3-2	Linear	-	10
3-3	Linear	-	3
4	SMPL layer [5]	-	6890 × 3

3.2. Training Method

Our aim during the training stage was to enrich both feature extractor f^{Feat} and mesh estimator f^{Mesh} , which were responsible for estimating 3D meshes. Our losses were proposed to improve the two networks (i.e. feature extractor f^{Feat} and mesh estimator f^{Mesh}) based on the 2D single image-level self-supervised losses, which were designed to solve three image manipulation tasks (i.e., jigsaw puzzle, rotation prediction, and part

in-painting) and one image projection task. In this subsection, we explain the entire training strategy and introduce the individual losses that we used.

3.2.1. Summary of the Entire Training Process.

The entire training process was divided into two stages: (1) pre-training stage and (2) mesh training stage.

At the (1) pre-training stage, five networks (i.e. jigsaw puzzle solver f^{Jigsaw} , rotation predictor f^{Rot} , part in-painter f^{Inp} , discriminator f^{Disc} , and texture network f^{Tex}) were pre-trained using the corresponding losses defined in Equations (2)–(6), respectively. We pre-trained these networks to constitute them for the self-supervision losses used in the subsequent mesh training stage.

Then, at the (2) mesh training stage, we trained the feature extractor f^{Feat} and mesh estimator f^{Mesh} , which were responsible for the 3D mesh estimation using the loss L defined as follows:

$$L(f^{\text{Feat}}, f^{\text{Mesh}}, f^{\text{Disc}}) = L_{\text{IM}}(f^{\text{Feat}}) + L_{\text{NR}}(f^{\text{Feat}}, f^{\text{Mesh}}, f^{\text{Disc}}) \quad (1)$$

where the image manipulation loss L_{IM} and neural rendering loss L_{NR} are defined in Equations (7) and (9), respectively. The actual improvement of the 3D mesh reconstruction was obtained during this stage. Furthermore, the discriminator f^{Disc} was further trained by discriminating between real and generated images to provide richer supervision.

The entire training process is summarized in the Algorithm 1.

3.2.2. Pre-Training Stage

The aim of this stage was to train five different networks (f^{Jigsaw} , f^{Rot} , f^{Inp} , f^{Disc} , and f^{Tex}) that were used to constitute the losses in the subsequent “mesh training” stage. We trained the networks by running $T_1 = 10$ epochs of training with the Adam optimizer on the losses L_{jigsaw} , L_{rot} , L_{inp} , L_{adv} and L_{RGB} (Equations (2)–(6)) with learning rates of 0.01, 5×10^{-4} , 1×10^{-4} , and 1×10^{-4} , and 0.01, respectively.

Jigsaw puzzle loss L_{jigsaw} . As in [19], the jigsaw puzzling task could be formulated as the standard classification task. We applied the cross-entropy loss L_{jigsaw} for the jigsaw puzzle solver f^{Jigsaw} to make its output close to the permutation order:

$$L_{\text{jigsaw}}(f^{\text{Jigsaw}} | \mathbf{x}, \mathbf{x}_{\text{jigsaw}}) = - \sum_{c=1}^C y_{c, \text{jigsaw}} \log(\hat{\mathbf{o}}_c) \quad (2)$$

where $y_{c, \text{jigsaw}}$ is the c -th dimensional element of the one-hot encoded vector from the permutation ground-truth, and $\hat{\mathbf{o}}_c$ is the c -th dimensional response of the f^{Jigsaw} network’s output.

Rotation prediction loss L_{rot} . The rotation prediction could be also formulated as the standard classification task as in [20]. Images were rotated with four possible angles θ , 0, 90, 180, and 270, and then mapped to the classification label set $\{1, 2, 3, 4\}$. To achieve this, the cross-entropy loss L_{rot} was applied on the output of the rotation predictor f^{Rot} :

$$L_{\text{rot}}(f^{\text{Rot}} | \mathbf{x}_{\text{rot}}) = - \sum_{c=1}^4 y_{c, \text{rot}} \log(\hat{\mathbf{r}}_c) \quad (3)$$

where $y_{c, \text{rot}}$ is the c -th dimensional element of the one-hot encoded vector from the rotation ground-truth, and $\hat{\mathbf{r}}_c$ is the c -th dimensional response of the f^{Rot} network’s output.

Algorithm 1: The summary of our entire training process**Input:**

- Training data $D = [D_{H36M}, D_{COCO}, D_{MPII}, D_{LSP}, D_{Youtube}, D_{MPI-INF-3DHP}]$ whose size is N .
- RGB image of training data;
- Hyper-parameters: number T_1, T_2 of epochs, size N' of mini-batch;

Output:

- Jigsaw puzzle solver output $\hat{\mathbf{o}}_c$;
- Rotation predictor output $\hat{\mathbf{r}}_c$;
- Part in-painter output $\hat{\mathbf{p}}$;
- Rendered RGB image $\hat{\mathbf{x}}_{\text{render}}$.

Initialization:

- Pre-train $f^{\text{Feat}}, f^{\text{Mesh}}$ based on [5];
- Initialize $f^{\text{r-cnn}}$ from [58];
- Implement f^{NMR} from [22];
- Randomize (parameters) of $f^{\text{jigsaw}}, f^{\text{Rot}}, f^{\text{Inp}}, f^{\text{Disc}}, f^{\text{Tex}}$.

for $t = 1, \dots, T_1 + T_2$ **do**

for $n = 1, \dots, N/N'$ **do**

if $t \leq T_1$ **then**

- For each data RGB image \mathbf{x} in the mini-batch D_n , output $\hat{\mathbf{o}}_c, \hat{\mathbf{r}}_c$, and $\hat{\mathbf{p}}$;
- Calculate gradient $\nabla L_{\text{jigsaw}}, \nabla L_{\text{rot}}, \nabla L_{\text{inp}}$ and ∇L_{adv} (Equations (2)–(5)) with respect to (the weights of) $f^{\text{jigsaw}}, f^{\text{Rot}}, f^{\text{Inp}}$ and f^{Disc} on D_n , and update $f^{\text{jigsaw}}, f^{\text{Rot}}, f^{\text{Inp}}$ and f^{Disc} ;
- Calculate gradient ∇L_{RGB} (Equation (6)) with respect to (the weights of) f^{Tex} on D_n , and update f^{Tex} ;

end

if $t > T_1$ **then**

- For each data RGB image \mathbf{x} in the mini-batch D_n , output $\hat{\mathbf{o}}_c, \hat{\mathbf{r}}_c, \hat{\mathbf{p}}$, and generate $\hat{\mathbf{x}}_{\text{render}}$;
- Calculate gradient ∇L (Equation (1)) with respect to (the weights of) $f^{\text{Feat}}, f^{\text{Mesh}}$ and f^{Disc} on D_n , and update $f^{\text{Feat}}, f^{\text{Mesh}}$, and f^{Disc} ;

end

end

end

Part in-painting loss L_{inp} . The part in-painting task could be framed as the image generation task as in [21]. We erased a patch \mathbf{p} from the center of the RGB images \mathbf{x} to make an image with a hole \mathbf{x}_{hole} . The image \mathbf{x}_{hole} was inputted to the $f^{\text{Inp}}(f^{\text{Feat}}(\mathbf{x}_{\text{hole}}))$, and the part in-painter f^{Inp} was trained to reconstruct the output $\hat{\mathbf{p}} = f^{\text{Inp}}(f^{\text{Feat}}(\mathbf{x}_{\text{hole}}))$. We trained the network using both the mean square error (MSE) loss and the GAN-type adversarial loss by the discriminator f^{Disc} to reconstruct a realistic patch $\hat{\mathbf{p}}$. The MSE loss was enforced to make the reconstructed patch $\hat{\mathbf{p}}$ look similar to the original patch \mathbf{p} , and GAN-type adversarial loss was enforced to make the reconstructed image $\hat{\mathbf{x}}_{\text{inp}}$ that combined \mathbf{x}_{hole} and $\hat{\mathbf{p}}$ look more realistic. f^{Inp} was pre-trained using the following rules:

$$L_{\text{inp}}(f^{\text{Inp}}|\mathbf{x}_{\text{hole}}) = \lambda_{\text{inp}} \|\mathbf{p} - \hat{\mathbf{p}}\|_2^2 + \lambda_{\text{disc}} \|f^{\text{Disc}}(\hat{\mathbf{x}}_{\text{inp}}) - 1\|_2^2 \quad (4)$$

where $\hat{\mathbf{x}}_{\text{inp}}$ is the in-painted image that fills the hole in \mathbf{x}_{hole} with the estimated $\hat{\mathbf{p}}$, and λ_{inp} and λ_{disc} are set at 0.999 and 0.001, respectively.

Adversarial loss L_{adv} . The discriminator f^{Disc} was trained to discriminate original images \mathbf{x} from images $\hat{\mathbf{x}}_{\text{inp}}$ —that is, the combination of \mathbf{x}_{hole} and $\hat{\mathbf{p}}$ —outputted from the f^{Inp} using the LSGAN [23] objective, as follows:

$$L_{\text{adv}}(f^{\text{Disc}}|\mathbf{x}, \hat{\mathbf{x}}_{\text{inp}}) = \|f^{\text{Disc}}(\mathbf{x}) - 1\|_2^2 + \|f^{\text{Disc}}(\hat{\mathbf{x}}_{\text{inp}}) - 0\|_2^2. \quad (5)$$

RGB rendering loss L_{RGB} . When projecting 3D meshes into the RGB images, RGB values for each vertex of 3D mesh were required. The texture network f^{Tex} was pre-trained to infer such RGB values using the loss, defined as follows:

$$L_{RGB}(f^{Tex}|\mathbf{x}) = \|\hat{\mathbf{x}}_{render} - \mathbf{x}_{render}\|_2^2 \quad (6)$$

where \mathbf{x}_{render} and $\hat{\mathbf{x}}_{render}$ denote the pseudo ground-truth obtained from $f^{r-cnn}(\mathbf{x}) \odot \mathbf{x}$ and rendered meshes $f^{NMR}(f^{Mesh}(f^{Feat}(\mathbf{x})), f^{Tex}(\mathbf{x}))$, respectively. The operation \odot denotes the pixel-wise multiplication and f^{r-cnn} denotes the pre-trained Mask-RCNN [58] network. The rationale behind the use of the Mask-RCNN [58] for obtaining the pseudo ground-truth \mathbf{x}_{render} is that 3D mesh reconstruction is more challenging than the 2D segmentation mask prediction.

3.2.3. Mesh Training Stage

At this stage, we trained the feature extractor f^{Feat} , mesh estimator f^{Mesh} , and the discriminator f^{Disc} with the aid of pre-trained networks obtained from the pre-training stage. We ran the training for $T_2 = 10$ epochs using the loss L (Equation (1)) with a learning rate of 5×10^{-6} . The learning rate was decreased using the exponential learning rate decay, whose decay rate was set to 0.99 for each epoch. In the remainder of this subsection, we elaborate the sub-loss of the loss L : L_{IM} and L_{NR} .

Image manipulation loss L_{IM} . This loss consisted of sub-losses reflecting three self-supervision tasks (i.e., solving jigsaw puzzling, rotation prediction and part in-painting tasks). To enrich the feature extractor f^{Feat} using jigsaw puzzling, rotation prediction, and part in-painting tasks, we used the loss L_{IM} that combined losses defined in Equations (2)–(4). We used the same form of losses as Equations (2)–(4) while changing the target training network to f^{Feat} :

$$\begin{aligned} L_{IM}(f^{Feat}) = & L_{jigsaw}(f^{Feat}|\mathbf{x}, \mathbf{x}_{jigsaw}) \\ & + L_{rot}(f^{Feat}|\mathbf{x}_{rot}) \\ & + L_{inp}(f^{Feat}|\mathbf{x}_{hole}). \end{aligned} \quad (7)$$

Neural rendering loss L_{NR} . This loss reflected the last self-supervision task (i.e., solving image projection task). Given a 2D image \mathbf{x} , 3D human meshes are estimated by the sequential combination of f^{Mesh} and f^{Feat} : $f^{Mesh}(f^{Feat}(\mathbf{x}))$. Then, we can render estimated 3D meshes \mathbf{m} to images $\hat{\mathbf{x}}_{render}$ and segmentation masks $\hat{\mathbf{x}}_{mask}$ using the neural mesh renderer f^{NMR} . L_{RGB} and L_{mask} are responsible for making two rendered images (i.e., $\hat{\mathbf{x}}_{render}$ and $\hat{\mathbf{x}}_{mask}$) close to their original inputs, respectively. Additional losses L_{disc} and L_{real} are defined to train the discriminator and obtain more realistic images. The combination of four sub-losses is called the neural rendering loss L_{NR} :

$$\begin{aligned} L_{NR}(f^{Feat}, f^{Mesh}, f^{Disc}) = & L_{RGB}(f^{Feat}, f^{Mesh}|\mathbf{x}) \\ & + L_{real}(f^{Feat}, f^{Mesh}|\mathbf{x}) \\ & + L_{mask}(f^{Feat}, f^{Mesh}|\mathbf{x}) \\ & + L_{disc}(f^{Disc}|\mathbf{x}, \hat{\mathbf{x}}_{render,bg}, \hat{\mathbf{x}}_{inp}) \end{aligned} \quad (8)$$

where $\hat{\mathbf{x}}_{render,bg}$ is the image that combines the rendered image $\hat{\mathbf{x}}_{render}$ and a random background image \mathbf{x}_{bg} . $L_{RGB}(f^{Feat}, f^{Mesh}|\mathbf{x})$ is in the same form as $L_{RGB}(f^{Tex}|\mathbf{x})$ in Equation (6);

however, the target training network is changed to f^{Feat} and f^{Mesh} . $L_{\text{mask}}(f^{\text{Feat}}, f^{\text{Mesh}}|\mathbf{x})$, $L_{\text{real}}(f^{\text{Feat}}, f^{\text{Mesh}}|\mathbf{x})$, and $L_{\text{disc}}(f^{\text{Disc}}|\mathbf{x}, \hat{\mathbf{x}}_{\text{render}}, \hat{\mathbf{x}}_{\text{inp}})$ are newly defined as follows:

$$L_{\text{real}}(f^{\text{Feat}}, f^{\text{Mesh}}|\mathbf{x}) = \|f^{\text{Disc}}(\hat{\mathbf{x}}_{\text{render},\text{bg}}) - 1\|_2^2, \quad (9)$$

$$L_{\text{mask}}(f^{\text{Feat}}, f^{\text{Mesh}}|\mathbf{x}) = \|\mathbf{x}_{\text{mask}} - \hat{\mathbf{x}}_{\text{mask}}\|_2^2, \quad (10)$$

$$L_{\text{disc}}(f^{\text{Disc}}|\mathbf{x}, \hat{\mathbf{x}}_{\text{render}}, \hat{\mathbf{x}}_{\text{inp}}) = \|f^{\text{Disc}}(\mathbf{x}) - 1\|_2^2 \\ + \|f^{\text{Disc}}(\hat{\mathbf{x}}_{\text{render},\text{bg}})\|_2^2 \\ + \|f^{\text{Disc}}(\hat{\mathbf{x}}_{\text{inp}})\|_2^2 \quad (11)$$

where \mathbf{x}_{mask} is the binary mask estimated from the Mask-RCNN [58]. Using Equation (11), the discriminator f^{Disc} is trained to discriminate the original images \mathbf{x} as real while the rendered images $\hat{\mathbf{x}}_{\text{rendered}}$ and in-painted images $\hat{\mathbf{x}}_{\text{inp}}$ are determined as fakes.

Joint loss L_{joints} . When 2D or 3D skeleton joints are available, this loss is used to close the keypoints regressed from 3D mesh vertices to their ground-truth locations (in the SMPL [13] model, the model that geometrically regresses 3D skeletons from mesh vertices is accompanied). This is an optional loss that is not used for the self-supervised setting (i.e., *Ours (self-sup)*); however, we used this loss for the weakly-supervised setting (i.e., *ours (weakly-sup)*); or for the semi-supervised setting (i.e., *ours (semi-sup)*), in which either of 2D or 3D skeletons are available, respectively.

Incorporation of different human images. As we constituted our framework with a self-supervised loss that did not require any 2D/3D skeleton annotations, we were able to train our network with any RGB human images outside of our training dataset. We additionally involved 2D images from the MPI-INF-3DHP [59], MPII [60], COCO [61], and LSP [62] datasets, which have been involved in the weakly/semi-supervised settings, plus our own collections of wild YouTube data. When detecting the humans in the images, we used the Mask-RCNN network [58] to inspect if there was a human bounding box or not. The results in Table 5 using “Full w/o Y” (row 4) and “Full” (row 5) datasets can be compared to see the performance improvement by the incorporation of human images, where “Full” and “Full w/o Y” denote the model trained by datasets with and without our own collections of wild YouTube images, respectively. We could observe that the accuracy was improved by the incorporation of human images.

Table 5. Ablation study on Human3.6M to analyze the effectiveness of the objective functions (first 2 rows) and results of weakly-supervised and semi-supervised approaches (last 2 rows). We also conducted an ablation study depending on the different training dataset compositions: “Full” denotes the full training dataset that we described in Section 4.1. Rows 3–4 show the ablation results using different training datasets, where “H36M” and “Full w/o Y” denote the Human3.6M dataset and full datasets without the collection of wild YouTube data, respectively. We use protocol-2 for “H36M”.

Methods	Dataset	MPJPE(↓)	PA-MPJPE(↓)
<i>Ours (self-sup)</i> — L_{real} — L_{mask}	Full	96.8	62.1
<i>Ours (self-sup)</i> — L_{real}	Full	95.7	60.3
<i>Ours (self-sup)</i>	H36M	90.4	55.4
<i>Ours (self-sup)</i>	Full w/o Y	86.5	54.9
<i>Ours (self-sup)</i>	Full	84.2	54.4
<i>Ours (weakly-sup)</i>	Full	80.7	52.7
<i>Ours (semi-sup)</i>	Full	65.8	44.9

4. Experiments

In this section, we explain the datasets and evaluation method and then analyze the obtained results in a qualitative and quantitative manner. We also provide ablation results by varying the parameters of our framework.

4.1. Dataset

We trained our model using the following datasets: Human3.6M [14], MPI-INF-3DHP [59], MPII [60], COCO [61], LSP [62], and a collection of wild YouTube data consisting of 62k, 96k, 14k, 28k, 1k, and 3k data entities, respectively. We tested using the following datasets: Human3.6M [14], 3DPW [63], and LSP [62].

The Human3.6M dataset is a widely used dataset with paired images and 2D/3D pose annotations. This is an in-studio dataset and consists of four multi-view images; however, we did not use multi-view images for training. Like [5], we used subjects S1, S5, S6, S7, S8 for training and S9 and S11 for testing. We computed the mean per joint position error (MPJPE) and Procrustes analysis-MPJPE (PA-MPJPE) on the Human3.6M test dataset to evaluate the 3D pose estimation results.

MPI-INF-3DHP has 2D images and corresponding 2D/3D skeleton annotation pairs. This is an in-studio dataset and consists of eight multi-view images; however, we did not use multi-view images for training. The MPII dataset and COCO dataset consist of indoor and outdoor images with only 2D annotations.

The LSP dataset is a wild athletic dataset that consists of wild images with 2D annotations. This dataset is used for evaluation, with their (part-)segmentation ground truths secured by [43]. The 3DPW dataset is an in-the-wild outdoor dataset and is used only for evaluation purposes. We further collected images from YouTube of jogging or dancing motions, with diverse backgrounds and viewpoints (the dataset is available in our Github repository).

4.2. Evaluation Method

We used two approaches to evaluate the estimated 3D poses: MPJPE and PA-MPJPE. MPJPE is the mean Euclidean distance between the ground truth and prediction for a joint. PA-MPJPE is Procrustes analysis-MPJPE, which aligns the estimated 3D poses to their ground-truths by a similarity transformation called Procrustes analysis before computing the MPJPE. We also used two approaches to evaluate the estimated 3D shapes: accuracy and F1 score for FG-BG and part-segmentation masks. We measured their accuracy and F1 score by a pixel-wise comparison between estimated masks and their ground truths.

4.3. Results

For the Human3.6M dataset, we compare our method to recent fully/semi/weakly/self-supervised methods [5,36,37,43,47–50,64] that output the 3D human poses by protocol-2 in Table 6. Our method outperformed previous methods in all three types of supervision settings (i.e., self/weakly/semi-supervised). Kundu et al. [64]’s method is the competitive self-supervised learning framework using the video priors; however, our method outperformed it even without any video priors. Li et al. [54]’s method is the only baseline that outperforms our method and uses the self-supervised setting. However, the work is not directly comparable to ours, as its setting is easier as it estimates only skeletal joints and using multi-view priors. We provide full 3D meshes inferring both pose and shape and use only 2D images as the supervision signals.

To show the generalization ability of our method, we evaluated it on 3DPW, the data of which were never seen in our training process. For the 3DPW dataset, we compared the accuracy of our method to those of recent state-of-the-art approaches [4,5,9,47,51,64] in Table 7. For the LSP dataset, as the dataset does not contain 3D pose annotations, we only measured the accuracy and F1 score of FG-BG segmentation and six-part segmentation, respectively, for comparison with other algorithms. We compared our method to state-of-the-art optimized-based methods [9,37,65] and regression-based methods [5,49,50,64] in Table 8. Optimization-based methods tend to have better performance on segmentation tasks than regression-based methods; however, our method outperformed most of them even in the self-supervised setting (i.e., *Ours (self-sup)*); our approach was also recorded as the best when involving more supervisions (i.e., *Ours (weakly-sup)* and *Ours (semi-sup)*).

Table 6. Evaluation with Human3.6M(Protocol-2). Methods in the first 10 rows use equivalent 2D and 3D pose supervision. Methods in rows 11–14 and rows 15–18 are weakly-supervised and self-supervised approaches, respectively. * indicates methods that output only 3D joints, not 3D meshes.

Methods	PA-MPJPE(↓)
Fully-supervised & Semi-supervised	
Lassner et al. [43]	93.9
Pavlakos et al. [37]	75.9
Omran et al. [36]	59.9
HMR [5]	56.8
Temporal-HMR [47]	56.9
Arnab et al. [48]	54.3
Kolotouros et al. [49]	50.1
TexturePose [50]	49.7
Kundu et al. [64]	48.1
<i>Ours (semi-sup)</i>	44.9
Weakly-supervised	
HMR unpaired [5]	66.5
Kundu et al. [64]	58.2
Iqbal et al. [52]	54.5
<i>Ours (weakly-sup)</i>	52.7
Self-supervised	
Kundu et al. [64]	90.5
* Chang et al. [66] (<i>multi-view-sup</i>)	77.0
Kundu et al. [64] (<i>multi-view-sup</i>)	74.1
<i>Ours (self-sup)</i>	54.4
* Li et al. [54] (<i>multi-view-sup</i>)	45.7

Table 7. Evaluation with wild 3DPW dataset in a *fully-unseen* setting. Unlike Temporal-HMR, we do not use any temporal supervision. Methods in the first seven rows use equivalent 2D and 3D pose supervision.

Methods	MPJPE(↓)	PA-MPJPE(↓)
Martinez et al. [4]	-	157.0
SMPLify [9]	199.2	106.1
TP-NET [51]	163.7	92.3
HMR [5]	130.0	76.7
Temporal-HMR [47]	127.1	80.1
Kundu et al. [64] (<i>semi-sup</i>)	125.8	78.2
<i>Ours (semi-sup)</i>	106.0	66.0
Kundu et al. [64] (<i>weakly-sup</i>)	153.4	89.8
<i>Ours (weakly-sup)</i>	107.4	69.1
Kundu et al. [64] (<i>self-sup</i>)	187.1	102.7
<i>Ours (self-sup)</i>	124.7	88.5

In Figure 3, we visualized qualitative examples of our method on Human3.6M, 3DPW, and LSP datasets. We overlaid predicted meshes, joints, part-segmentation masks, and neural rendered images on the input images. Even though our purpose was not to estimate textures at the testing stage; we could observe that our method faithfully estimates textures for unseen testing images.

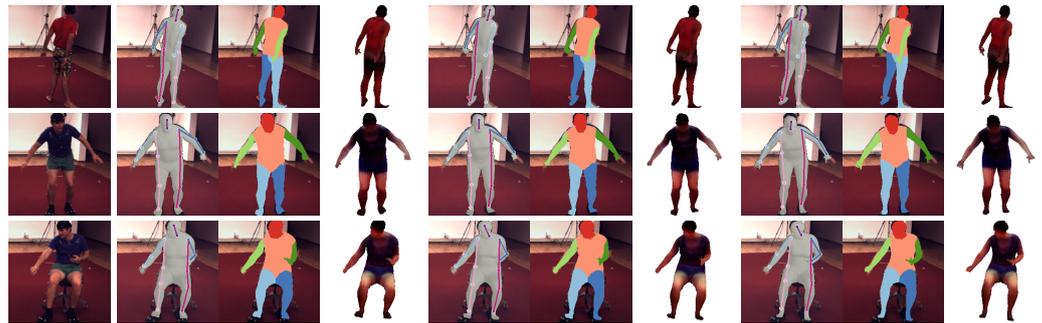
Table 8. Evaluation of FG-BG and six-part segmentation with LSP test set. First 4 rows: *optimization-based* methods. Last 10 rows: *regression-based* methods.

Methods	FG-BG Seg.		Part Seg.	
	Acc. (↑)	F1 (↑)	Acc. (↑)	F1 (↑)
Optimization-based				
SMPLify <i>oracle</i> [9]	92.17	0.88	88.82	0.67
SMPLify [9]	91.89	0.88	87.71	0.64
SMPLify on [37]	92.17	0.88	88.24	0.64
Bodynet [65]	92.75	0.84	-	-
Fully-supervised & Semi-supervised				
HMR [5]	91.67	0.87	87.12	0.60
Kolotouros et al. [49]	91.46	0.87	88.69	0.66
TexturePose [50]	91.82	0.87	89.00	0.67
Kundu et al. [64]	91.84	0.87	89.08	0.67
Ours (<i>semi-sup</i>)	93.28	0.90	89.90	0.70
Weakly-supervised				
HMR unpaired [5]	91.30	0.86	87.00	0.59
Kundu et al. [64]	91.70	0.87	87.12	0.60
Ours (<i>weakly-sup</i>)	93.31	0.90	89.86	0.70
Self-supervised				
Kundu et al. [64]	91.46	0.86	87.26	0.64
Ours (<i>self-sup</i>)	92.62	0.89	87.72	0.63

4.4. Ablation Study

To analyze the effectiveness of our entire training objective (Equation (1)), we conducted ablation experiments on our losses, as presented in Table 5. In our initial experiment, we used the combination of three images' manipulation task losses (i.e., jigsaw puzzling, rotation prediction, and part in-painting) and one RGB rendering loss L_{RGB} for the "mesh training stage". However, given only these four types of losses, we observed that 3D meshes were trained in the wrong way, inferring camera scale parameters to be very small, as this might be an easier way to reduce the RGB rendering loss L_{RGB} . To relieve this, we proposed the addition of more constraints in our loss by the segmentation mask loss L_{mask} , defined in Equation (10): we observed that the uses of loss L_{mask} and the discriminator loss L_{adv} are helpful for preventing this phenomenon. In Table 5, by comparing results from "Ours (*self-sup*)- L_{real} - L_{mask} ", "Ours (*self-sup*)- L_{real} ", and "Ours (*self-sup*)", we could conclude that results improved as more supervision was used. The corresponding qualitative results are also shown in Figure 4. Furthermore, we experimented on different configurations of datasets for training in the same table (Table 5): Rows 3–4 of Table 5 show the accuracy using fewer datasets for training where "H36M" and "Full w/o Y" denote the Human3.6M dataset and full datasets without the collection of wild YouTube data, respectively. "Full" denotes the full training data that we described in Section 4.1, including the collection of wild YouTube data compared to "Full w/o Y". We can see that involving more image data results in improved accuracy.

A. Results on Human3.6M dataset (in-studio)



B. Results on 3DPW dataset (in-the-wild)



C. Results on LSP dataset (in-the-wild)

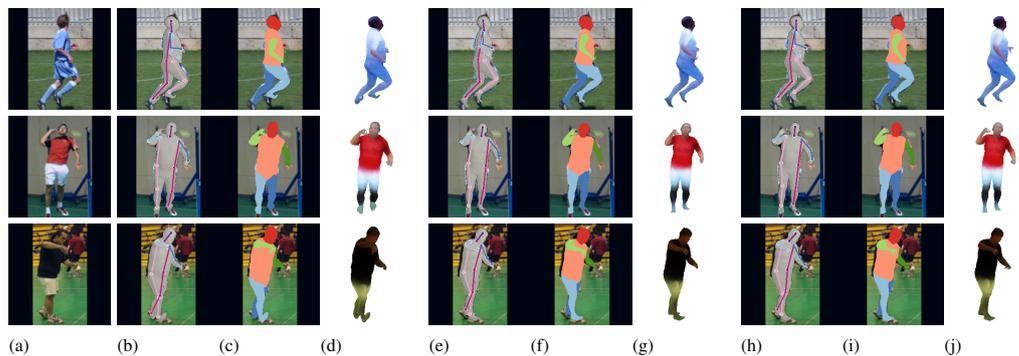


Figure 3. Qualitative examples of three databases (Human3.6M, 3DPW, and LSP). (a) Input image; (b–d), (e–g), and (h–j) denote our results obtained from self/weakly and semi-supervised models, respectively. (b,e,h) show images with joints and 3D meshes, (c,f,i) show images with part-segmentation masks, and (d,g,j) show neural mesh rendered images, respectively.

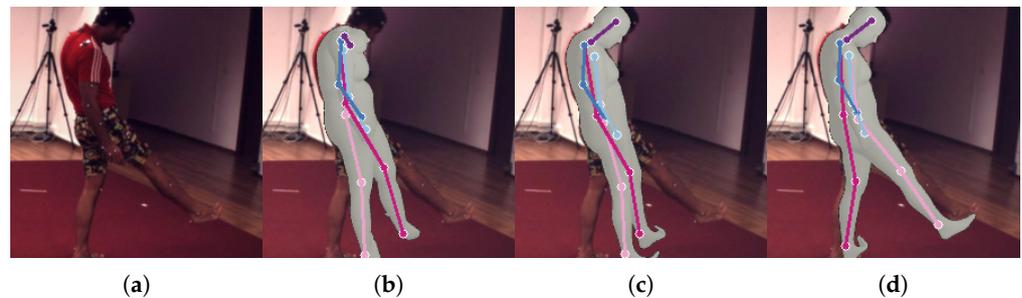


Figure 4. Qualitative examples depending on losses: (a) input, (b) results from ‘Ours (self-sup)- $L_{real}-L_{mask}$ ’, (c) results from ‘Ours (self-sup)- L_{real} ’, and (d) results from ‘Ours (self-sup)’. Results become better involving more supervision.

We further analyzed the effectiveness of the individual image-level supervision (i.e., L_{jigsaw} , L_{rot} , and L_{inp}) in the Table 9. We conducted the experiment by applying one of our self-supervision losses for our self/weakly and semi-supervised baselines. For example, "Ours (self-sup)- $L_{\text{rot}}-L_{\text{inp}}$ ", "Ours (self-sup)- $L_{\text{jigsaw}}-L_{\text{inp}}$ ", and "Ours (self-sup)- $L_{\text{rot}}-L_{\text{jigsaw}}$ " denote the network trained without a rotation prediction loss L_{rot} and a part in-painting loss L_{inp} , the network trained without a jigsaw puzzle loss L_{jigsaw} and a part in-painting loss L_{inp} , and the network trained without a rotation prediction loss L_{rot} and a jigsaw puzzle loss L_{jigsaw} for our self-supervision setting, respectively. From these experiments, we can see that (1) none of three losses clearly prevail over the other remaining, and (2) applying all the losses consistently outperforms the results when applying one of the losses. Thus, we need to combine three image-level self-supervisions (i.e., L_{jigsaw} , L_{rot} , and L_{inp}) together in our main experiments.

Table 9. Ablation study on Human3.6M, 3DPW, and LSP to analyze the effectiveness of image manipulation task losses.

Methods	Human3.6M		3DPW		LSP	
	MPJPE (\downarrow)	PA-MPJPE (\downarrow)	MPJPE (\downarrow)	PA-MPJPE (\downarrow)	BG-FG Seg.	Part Seg.
					Acc. (\uparrow)	Acc. (\uparrow)
<i>Ours (self-sup) - $L_{\text{rot}} - L_{\text{inp}}$</i>	90.1	55.5	126.7	89.5	92.80	88.15
<i>Ours (self-sup) - $L_{\text{jigsaw}} - L_{\text{inp}}$</i>	93.1	60.0	137.5	98.4	92.86	88.06
<i>Ours (self-sup) - $L_{\text{rot}} - L_{\text{jigsaw}}$</i>	91.2	58.2	135.9	98.7	92.76	87.70
<i>Ours (self-sup)</i>	84.2	54.4	124.7	88.	92.62	87.72
<i>Ours (weakly-sup) - $L_{\text{rot}} - L_{\text{inp}}$</i>	83.7	53.0	112.1	69.3	93.39	89.78
<i>Ours (weakly-sup) - $L_{\text{jigsaw}} - L_{\text{inp}}$</i>	86.3	54.7	110.5	68.2	93.37	89.76
<i>Ours (weakly-sup) - $L_{\text{rot}} - L_{\text{jigsaw}}$</i>	84.1	53.9	109.6	68.2	93.45	89.88
<i>Ours (weakly-sup)</i>	80.7	52.7	107.4	69.1	93.31	89.86
<i>Ours (semi-sup) - $L_{\text{rot}} - L_{\text{inp}}$</i>	67.7	46.0	105.8	64.9	93.18	89.84
<i>Ours (semi-sup) - $L_{\text{jigsaw}} - L_{\text{inp}}$</i>	68.4	45.9	108.0	65.7	93.23	89.79
<i>Ours (semi-sup) - $L_{\text{rot}} - L_{\text{jigsaw}}$</i>	67.6	46.5	106.9	65.0	93.22	89.88
<i>Ours (semi-sup)</i>	65.8	44.9	106.0	66.0	93.28	89.90

5. Conclusions

In this paper, we present a self-supervised learning framework for recovering human 3D meshes from a single RGB image. We proposed the use of the combination of three image manipulation task losses and one neural rendering loss to enrich the feature space and boost the mesh reconstruction accuracy. We also find that the combination of RGB rendering, mask rendering, and adversarial losses is essential to properly achieve image-level self-supervision without using video or multi-view priors. Experiments were conducted on three popular benchmarks, and our algorithm achieved the best accuracy among competitive self/weakly/semi and fully-supervised algorithms. Via these results, we could conclude that single 2D image-level supervision could be a strong supervision method for training 3D human pose and shape estimation. While we showed promising results using the proposed 2D single image-level self-supervision losses, avoiding the effort of collecting other types of supervision signals, some small inconvenience remains in our framework in terms of manually setting hyper-parameters. Future work should explore the possibilities of developing a fully automatic pipeline that resolves this inconvenience by including schemes to select the optimal hyper-parameters for our losses (e.g., tile number in jigsaw puzzle loss, rotation angles for rotation prediction loss and etc.).

Author Contributions: Conceptualization, J.C., W.-H.P. and S.B.; Methodology, J.C. and M.S.; Software, J.C. and M.S.; Validation, J.C., M.S. and S.B.; Formal Analysis, C.L., S.L. (Seongyeong Lee) and S.L. (Seungeun Lee); Writing—Original Draft Preparation, M.S., W.-H.P. and S.B.; Writing—Review and Editing, M.S., W.-H.P. and S.B.; Visualization, C.L., S.L. (Seongyeong Lee) and D.K.; Supervision, W.-H.P. and S.B.; Project Administration, S.B.; Funding Acquisition, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant from the R&D Program of the Korea Railroad Research Institute, Republic of Korea.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
2. Wei, S.E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional pose machines. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
3. Zhou, X.; Zhu, M.; Leonardos, S.; Derpanis, K.G.; Daniilidis, K. Sparseness meets deepness: 3d human pose estimation from monocular video. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
4. Martinez, J.; Hossain, R.; Romero, J.; Little, J.J. A simple yet effective baseline for 3d human pose estimation. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
5. Kanazawa, A.; Black, M.J.; Jacobs, D.W.; Malik, J. End-to-end Recovery of Human Shape and Pose. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
6. Kocabas, M.; Athanasiou, N.; Black, M.J. VIBE: Video inference for human body pose and shape estimation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
7. Sun, Y.; Bao, Q.; Liu, W.; Fu, Y.; Black, M.J.; Mei, T. CenterHMR: Multi-person center-based human mesh recovery. *arXiv* **2020**, arXiv:2008.12272.
8. Guan, S.; Xu, J.; Wang, Y.; Ni, B.; Yang, X. Bilevel online adaptation for out-of-domain human mesh reconstruction. *arXiv* **2021**, arXiv:2103.16449.
9. Bogo, 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 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
10. Pavlakos, G.; Choutas, V.; Ghorbani, N.; Bolkart, T.; Osman, A.A.; Tzionas, D.; Black, M.J. Expressive body capture: 3d hands, face, and body from a single image. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
11. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
12. Yang, W.; Ouyang, W.; Wang, X.; Ren, J.; Li, H.; Wang, X. 3d human pose estimation in the wild by adversarial learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5255–5264.
13. Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; Black, M.J. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.* **2015**, *34*, 248. [[CrossRef](#)]
14. 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.* **2013**, *36*, 1325–1339. [[CrossRef](#)] [[PubMed](#)]
15. Varol, G.; Laptev, I.; Schmid, C.; Zisserman, A. Synthetic humans for action recognition from unseen viewpoints. *arXiv* **2019**, arXiv:1912.04070.
16. Varol, G.; Romero, J.; Martin, X.; Mahmood, N.; Black, M.J.; Laptev, I.; Schmid, C. Learning from synthetic humans. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
17. Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; Webb, R. Learning from simulated and unsupervised images through adversarial training. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
18. Rad, M.; Oberweger, M.; Lepetit, V. Feature mapping for learning fast and accurate 3D pose inference from synthetic images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
19. Noroozi, M.; Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
20. Gidaris, S.; Singh, P.; Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv* **2018**, arXiv:1803.07728.
21. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.

22. Kato, H.; Ushiku, Y.; Harada, T. Neural 3D mesh renderer. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
23. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.K.; Wang, Z.; Smolley, S.P. Least squares generative adversarial networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
24. Xu, T.; Takano, W. Graph Stacked Hourglass Networks for 3D Human Pose Estimation. *arXiv* **2021**, arXiv:2103.16385.
25. Moreno-Noguer, F. 3d human pose estimation from a single image via distance matrix regression. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
26. Chen, C.H.; Ramanan, D. 3d human pose estimation= 2d pose estimation+ matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7035–7043.
27. Rayat Intiaz Hossain, M.; Little, J.J. Exploiting temporal information for 3D pose estimation. *arXiv* **2017**, arXiv:1711.08585.
28. Akhter, I.; Black, M.J. Pose-conditioned joint angle limits for 3D human pose reconstruction. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
29. Ramakrishna, V.; Kanade, T.; Sheikh, Y. Reconstructing 3d human pose from 2d image landmarks. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012.
30. Fish Tung, H.Y.; Harley, A.W.; Seto, W.; Fragkiadaki, K. Adversarial inverse graphics networks: Learning 2D-to-3D lifting and image-to-image translation from unpaired supervision. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
31. Sanzari, M.; Ntouskos, V.; Pirri, F. Bayesian image based 3d pose estimation. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
32. Zhou, X.; Zhu, M.; Leonardos, S.; Daniilidis, K. Sparse representation for 3D shape estimation: A convex relaxation approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1648–1661. [[CrossRef](#)] [[PubMed](#)]
33. Agarwal, A.; Triggs, B. Recovering 3D human pose from monocular images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 44–58. [[CrossRef](#)] [[PubMed](#)]
34. Balan, A.O.; Black, M.J. The naked truth: Estimating body shape under clothing. In Proceedings of the 10th European Conference on Computer Vision, Marseille, France, 12–18 October 2008.
35. Guler, R.A.; Kokkinos, I. Holopose: Holistic 3d human reconstruction in-the-wild. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
36. Omran, M.; Lassner, C.; Pons-Moll, G.; Gehler, P.; Schiele, B. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018.
37. 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 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
38. Tan, J.K.V.; Budvytis, I.; Cipolla, R. Indirect Deep Structured Learning for 3D Human Body Shape and Pose Prediction. In Proceedings of the British Machine Vision Conference (BMVC), London, UK, 4–7 September 2017; BMVA Press: Durham, UK, 2017; pp. 15.1–15.11.
39. Tung, H.Y.F.; Tung, H.W.; Yumer, E.; Fragkiadaki, K. Self-supervised learning of motion capture. *arXiv* **2017**, arXiv:1712.01337.
40. Saito, S.; Huang, Z.; Natsume, R.; Morishima, S.; Kanazawa, A.; Li, H. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
41. Saito, S.; Simon, T.; Saragih, J.; Joo, H. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
42. Huang, Y.; Bogo, F.; Lassner, C.; Kanazawa, A.; Gehler, P.V.; Romero, J.; Akhter, I.; Black, M.J. Towards accurate marker-less human shape and pose estimation over time. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017.
43. Lassner, C.; Romero, J.; Kiefel, M.; Bogo, F.; Black, M.J.; Gehler, P.V. Unite the people: Closing the loop between 3d and 2d human representations. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
44. 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 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
45. Lin, K.; Wang, L.; Liu, Z. End-to-end human pose and mesh reconstruction with transformers. *arXiv* **2021**, arXiv:2012.09760 .
46. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 6000–6010.
47. Kanazawa, A.; Zhang, J.Y.; Felsen, P.; Malik, J. Learning 3d human dynamics from video. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
48. Arnab, A.; Doersch, C.; Zisserman, A. Exploiting temporal context for 3D human pose estimation in the wild. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.

49. Kolotouros, N.; Pavlakos, G.; Daniilidis, K. Convolutional mesh regression for single-image human shape reconstruction. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
50. Pavlakos, G.; Kolotouros, N.; Daniilidis, K. Texturepose: Supervising human mesh estimation with texture consistency. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
51. Dabral, R.; Mundhada, A.; Kusupati, U.; Afaque, S.; Sharma, A.; Jain, A. Learning 3d human pose from structure and motion. In Proceedings of the 15th European Conference, Munich, Germany, 8–14 September 2018.
52. Iqbal, U.; Molchanov, P.; Kautz, J. Weakly-Supervised 3D Human Pose Learning via Multi-view Images in the Wild. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
53. 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 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
54. Li, Y.; Li, K.; Jiang, S.; Zhang, Z.; Huang, C.; Da Xu, R.Y. Geometry-driven self-supervised method for 3d human pose estimation. *AAAI Conf. Artif. Intell.* **2020**, *34*, 11442–11449. [[CrossRef](#)]
55. Kocabas, M.; Karagoz, S.; Akbas, E. Self-supervised learning of 3d human pose using multi-view geometry. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
56. Rhodin, H.; Salzmann, M.; Fua, P. Unsupervised geometry-aware representation for 3d human pose estimation. In Proceedings of the 15th European Conference, Munich, Germany, 8–14 September 2018.
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
58. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
59. 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 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017.
60. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2d human pose estimation: New benchmark and state of the art analysis. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
61. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014.
62. Johnson, S.; Everingham, M. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In Proceedings of the British Machine Vision Conference, BMVC 2010, Aberystwyth, UK, 31 August–3 September 2010.
63. von Marcard, T.; Henschel, R.; Black, M.J.; Rosenhahn, B.; Pons-Moll, G. Recovering accurate 3d human pose in the wild using imus and a moving camera. In Proceedings of the 15th European Conference, Munich, Germany, 8–14 September 2018.
64. Kundu, J.N.; Rakesh, M.; Jampani, V.; Venkatesh, R.M.; Babu, R.V. Appearance Consensus Driven Self-supervised Human Mesh Recovery. In Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020.
65. Varol, G.; Ceylan, D.; Russell, B.; Yang, J.; Yumer, E.; Laptev, I.; Schmid, C. Bodynet: Volumetric inference of 3d human body shapes. In Proceedings of the 15th European Conference, Munich, Germany, 8–14 September 2018.
66. Chang, I.; Park, M.G.; Kim, J.; Yoon, J.H. Multi-View 3D Human Pose Estimation with Self-Supervised Learning. In Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Jeju Island, Korea, 13–16 April 2021.