Skeleton2Mesh: Kinematics Prior Injected Unsupervised Human Mesh Recovery

Zhenbo Yu[1,2]*, Junjie Wang[1,2]*, Jingwei Xu[1,2], Bingbing Ni[1,2]†
Chenglong Zhao[1,2], Minsi Wang[1,2], Wenjun Zhang[1,2]
[1]Shanghai Jiao Tong University, [2]Shanghai Key Lab of Digital Media Processing & Transmission
{yuzhenbo, dreamboy.gns, xjwxjw, nibingbing, cl-zhao,mswang1994, zhangwenjun}@sjtu.edu.cn

Abstract

In this paper, we decouple unsupervised human mesh recovery into the well-studied problems of unsupervised 3D pose estimation, and human mesh recovery from estimated 3D skeletons, focusing on the latter task. The challenges of the latter task are two folds: (1) ***pose failure*** *(i.e., pose mismatching – different skeleton definitions in dataset and SMPL , and pose ambiguity – endpoints have arbitrary joint angle configurations for the same 3D joint coordinates). (2)* ***shape ambiguity*** *(i.e., the lack of shape constraints on body configuration). To address these issues, we propose Skeleton2Mesh, a novel lightweight framework that recovers human mesh from a single image. Our Skeleton2Mesh contains three modules, i.e., Differentiable Inverse Kinematics (DIK), Pose Refinement (PR) and Shape Refinement (SR) modules. DIK is designed to transfer 3D rotation from estimated 3D skeletons, which relies on a minimal set of kinematics prior knowledge. Then PR and SR modules are utilized to tackle the pose ambiguity and shape ambiguity respectively. All three modules can be incorporated into Skeleton2Mesh seamlessly via an end-to-end manner. Furthermore, we utilize an adaptive joint regressor to alleviate the effects of skeletal topology from different datasets. Results on the Human3.6M dataset for human mesh recovery demonstrate that our method improves upon the previous unsupervised methods by 32.6% under the same setting. Qualitative results on in-the-wild datasets exhibit that the recovered 3D meshes are natural, realistic. Our project is available at https://sites.google.com/view/skeleton2mesh.*

1. Introduction

Recovering human mesh from in-the-wild monocular images has been a promising goal in the vision community. This is considered as a crucial step for a variety of downstream applications such as robot interaction [38], augmented reality [16], animation industry [1], etc. Recent methods based on parametric models, such as SCAPE [2], SMPL [22] and SMPL-X [41] can be simply divided into two categories: regression-based and optimization-based.

Model-based methods	Paired 3D Sup.	Unpaired 3D Sup.	Temporal Information	Optimized Module
SMPLify [4]	✗	✓	✗	✓
Song et al. [46]	✗	✓	✗	✓
NBF [40]	✓	✗	✗	✓
Pavlakos et al. [42]	✓	✗	✗	✗
HMR [22]	✗	✓	✗	✗
SPIN [27]	✗	✓	✗	✓
PoseNet [48]	✗	✗	✓	✗
Ours	✗	✗	✗	✗

Table 1: Characteristic comparison of our method against previous model-based methods, in terms of supervision signals and the usage of optimized module.

Regression-based methods [22, 49] or Optimization-based methods [4, 13, 30] rely on 3D annotations or optimized module. Different from above, our method requires 3D supervision training scheme but free from 3D annotation (i.e., 3D skeleton, $\boldsymbol{\beta}$ or $\boldsymbol{\theta}$ in SMPL), optimized module, and temporal information (illustrated in Tab. 1).

Specifically, unsupervised human mesh recovery aims to recover the SMPL model, which is comprised of pose parameters (3D rotation) and shape parameters. **(a)** In terms of pose parameters, most existing methods [22, 14] directly regress 3D rotation from images or 2D pose. However, these methods all heavily rely on paired or unpaired 3D annotations. However, we can easily see that the SMPL model with 3D rotation alone is similar to the corresponding 3D skeleton, disregarding the shape information. Recent unsupervised 3D pose estimation [6] has achieved promising performance, which motivates us to use estimated 3D skeleton to facilitate human mesh recovery [14, 48]. HybrIK exploits inverse kinematics process to establish strict correspondence between 24 3D joints and 24 3D rotations provided by SMPL model, which heavily relies on supervised

*equal contribution
†corresponding author

3D annotation. Notably, 24 3D joints (includes hands and feet) and 24 3D rotations are highly difficult to obtain. **(b)** In terms of shape parameters, most recent methods [22, 27] exploit discriminator by unpaired 3D pose (such as CMU prior [27]) or simple regularizer via the average shape [48] to obtain more valid 3D human mesh. However, unpaired 3D pose is also expensive to capture and a simple regularizer based on the average shape is unable to capture more reasonable shape for specific human character. This inspires us to use silhouettes to obtain more valid shape.

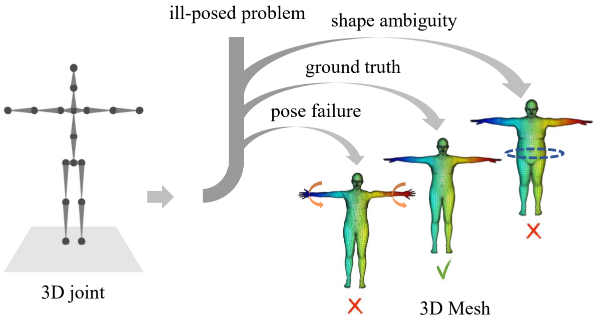

Figure 1: Transforming 3D skeleton to mesh is an ill-posed problem with no unique solution. Notably, pose failure is comprised of pose mismatching and pose ambiguity.

To this end, we decouple human mesh recovery into the well-studied problems of unsupervised 3D pose estimation [6], and unsupervised human mesh recovery from estimated 3D skeleton, focusing on the latter. Specifically, the challenges of the latter task are two folds (see Fig. 1): **(1a.) Pose mismatching.** Different skeleton definitions, mismatching joint numbers, and the cases that a single 3D skeleton possibly corresponds to multiple 3D meshes, which causes the large accuracy gap between pose estimation and reconstruction [14, 27, 43, 51]. **(1b.) Pose ambiguity.** Pose ambiguity refers to ambiguities in rotations of endpoints. In other words, endpoints have arbitrary joint angle configurations for the same 3D joint coordinates. **(2.) Shape ambiguity.** It can be easily seen that we are unable to obtain sufficient shape information from 3D skeleton.

In this paper, we propose Skeleton2Mesh, a novel lightweight framework that recovers human mesh from a single image. Our Skeleton2Mesh consists of three modules, i.e., DIK, PR, and SR modules. These three modules would be discussed in detail as follows: **(a) Differentiable inverse kinematics module.** Inverse kinematics methods have been studied to enable robots to imitate human body motion from a person. We are thus motivated to design the differentiable inverse kinematics (DIK) module to infer 3D rotations from the estimated 3D skeletons. DIK module relies on a minimal set of prior knowledge that defines the underlying kinematic 3D structure, and it can be incorporated into our framework seamlessly without any trainable parameters. **(b) Pose refinement module.** Most existing unsupervised 3D pose estimation methods commonly output 3D skeleton with 14-17 joints [6, 32]), which do not estimate hand or foot. Furthermore, the head positions cross datasets are different from each other. For example, the head position in Human3.6M dataset [18] and that in 3DHP dataset [36] are different. Thus it is unreasonable to transform these joints to the corresponding uniform 3D rotation. To this end, we use a pose refinement module to address above issues. **(c) Shape refinement module.** Extra shape information is obtained by silhouettes from the off-the-shelf detector. We exploit the shape refinement module to alleviate shape ambiguity. In summary, all the modules can be integrated into the lightweight framework seamlessly in an end-to-end manner.

We benchmark the proposed approach on various 3D human pose datasets and it outperforms state-of-the-art unsupervised methods [48, 49] by **4.0** mm PMPJPE on Human3.6M [18], **7.6** AUC on MPI-INF-3DHP [36] and **11.8** mm PMPJPE on Surreal [50].

2. Related Work

Unsupervised 3D Pose Estimation. Previous unsupervised 2D to 3D approaches can be widely classified into unsupervised 3D pose estimation [44, 6, 29, 24] and unsupervised human mesh recovery [4, 30, 26, 48]. Rhodin et al. [44] propose to learn a geometry-aware body representation from generated multi-view images without 3D labels, which exploits the consistency in camera geometry and multi-view information. Geometric self-supervision is presented by Chen et al. [6] without requiring any multi-view correspondence. It provides a simple yet effective baseline for unsupervised 3D pose estimation, which is also adopted in our work. Kundu et al. [29] exploit a minimal set of prior kinematics knowledge or encoder and decoder module in a self-supervised manner to facilitate pose estimation. Despite considerable progress in unsupervised 3D pose estimation, unsupervised 3D human mesh recovery still remains challenging due to the lack of 3D mesh supervision, which is more difficult to capture compared with 3D joints.

Unsupervised 3D Human Mesh Recovery. Unsupervised human mesh recovery is much more difficult than unsupervised 3D pose estimation due to richer reconstruction information. Recent model-based methods [4, 30, 22, 49, 42, 26, 48] can be simply divided into two categories: optimization-based methods and regression-based methods. SMPLify [4] and Lassner et al. [30] are the earliest end-to-end approaches, which fit the SMPL body model to 2D evidence(predicted 2D keypoints or silhouettes). HMR [22] directly regresses SMPL parameters from images using adversarial learning to exploit unpaired 3D data. Recently, SPIN [27] combines optimization-based methods

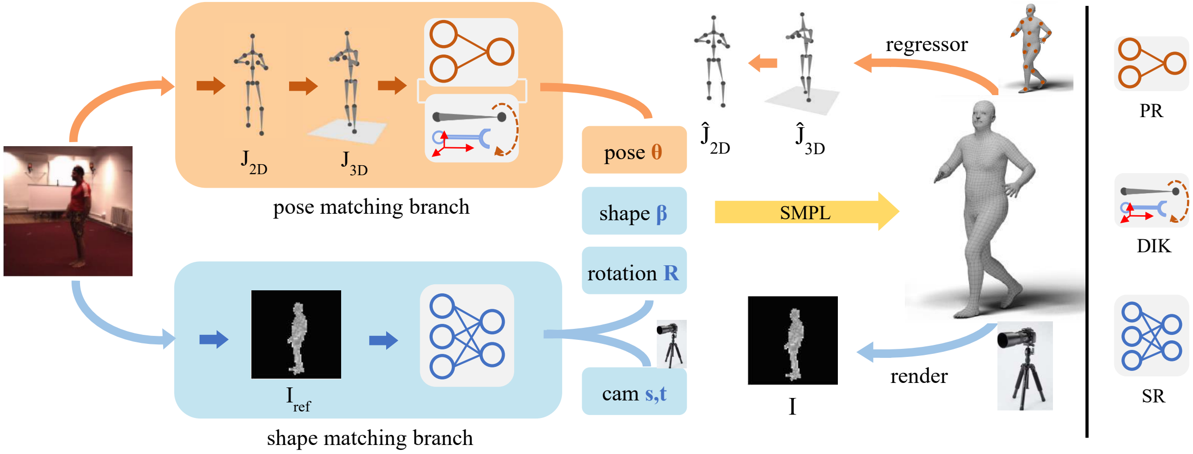

Figure 2: Detailed architecture of Skeleton2Mesh framework. Given a single image, 2D joints are estimated by a pre-trained 2D pose detector (e.g., CPN [7]), and masks are predicted by a new universal human parsing agent named "Graphonomy" [12]. Specifically, PR, DIK, and SR in the left denote pose refinement, differentiable inverse kinematics, and shape refinement, respectively. $\hat{\mathbf{J}}^{2D}$ and $\mathbf{J}^{2D}$ denote 2D joints. $\hat{\mathbf{J}}^{3D}$ and $\mathbf{J}^{3D}$ denote 3D skeletons. $\mathbf{I}$ and $\mathbf{I}_{ref}$ denote silhouettes.

and regression-based methods to form a self-improving cycle. However, the embedded optimization module is still time-consuming, making it hard to apply to real-time task. Also, most existing methods aforementioned use unpaired 3D supervision, which is also expensive and tedious to obtain. In comparison, our method does not use any form of 3D annotation, optimization module.

Inverse Kinematics. There have been sufficient works [53, 19, 9, 10] that try to encode kinematics priors into the learning paradigm of 2D/3D human pose estimation. Inverse kinematics (IK) calculates the variable joint parameters (e.g. rotation vectors) needed to place the end of a kinematic chain in a given position and is widely used in human imitation [39, 33] and robotic control [8]. Typically IK solvers are based on iterative optimization [5, 11, 17, 25]. There also exist heuristic methods (FABRIK [3], IK-FA[45]) designed to speed up the convergence and analytical solutions designed for some special applications [21, 47]. The concurrent work HybrIK [31] is most correlated to ours, which also integrates inverse kinematics in an end-to-end human mesh recovery pipeline. HybrIK decouples joint rotation into an analytical solved swing component and a learnable twist component. Different from HybrIK, we focus on totally unsupervised setting and provide efficient analytical IK solutions for the human body system. Our DIK module is efficient and can be easily plugged into any learning paradigm.

3. Method

3.1. Overview

The overall framework of Skeleton2Mesh is summarized in Fig 2. We can see that Skeleton2Mesh contains pose matching and shape matching branch. More concretely, Pose matching branch includes lifting 2D joint to 3D skeleton, transforming 3D skeleton to 3D rotation (DIK and PR modules). Shape matching branch contains SR module. However, we only introduce DIK, PR, and SR modules in Sec. 3. The detailed information of lifting 2D joint to 3D skeleton can be seen in the supplementary materials.

3D Body Representation We encode the 3D mesh of a human body using the Skinned Multi-Person Linear (SMPL) model. The model is parameterized by Θ that contains the pose and shape parameters $\boldsymbol{\theta} \in \mathbb{R}^{72}$ and $\boldsymbol{\beta} \in \mathbb{R}^{10}$ respectively. Pose parameters are comprised of global body rotation $\mathbf{R}$ and the relative rotation of 23 joints in axis-angle format, while the shape parameters are the first 10 coefficients of a PCA shape space. SMPL is a differentiable function, $\mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta}) \in \mathbb{R}^{6890\times 3}$, which shapes a template mesh based on forward kinematics constrained by $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. 3D skeleton $\hat{\mathbf{J}}^{3D}$ can be obtained from mesh vertices via $\hat{\mathbf{J}}^{3D} = \mathbf{RW}\mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta})$ utilizing an adaptive regressor $\mathbf{W}$.

Camera Projection As the inverse kinematics module is designed to be view-invariant, we rely on estimates of the camera intrinsics $\boldsymbol{\pi}$ in the canonical system C, to obtain 2D landmarks of the skeleton. Note that, these 2D landmarks are expected to register with the corresponding joint locations in the input image. Thus, 2D landmarks are obtained as, $\hat{\mathbf{J}}_i^{2D} = \mathcal{P}(\hat{\mathbf{J}}_i^{3D}, \boldsymbol{\pi})$, where $\mathcal{P}$ denotes the projection function of a weak-perspective camera.

3.2. Pose Matching Branch

In this section, the pose matching branch aims to generate the same body movement as the corresponding SMPL with pose parameters alone from the generated 3D skeleton. We identify two types of pose mismatching as follows:

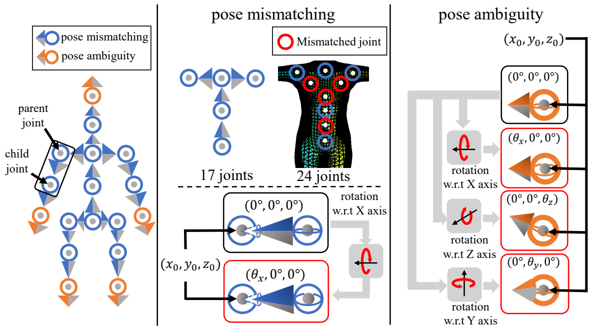

Figure 3: Illustration of pose failure. (a). Pose mismatching and pose ambiguity focus on different types of joints, i.e., blue and orange ones, respectively, resulting from different types of reasons. Specifically, pose mismatching refers to joint number mismatch, e.g., 17 joints defined in dataset and 24 joints in SMPL [35], and joint angle mismatch, e.g., ambiguities in rotations around an axis (*addressed by DIK*, see Fig. 3(b)). Pose ambiguity, Fig. 3(c), refers to the rotations of endpoints which are *can not be constrained by DIK*.

- **Joint number mismatch.** Joints estimated in 3D pose estimation (14-17 joints) are commonly less than local 3D rotations in SMPL (23 local 3D rotations), which lacks enough information to recover accurate 3D rotation in SMPL from estimated 3D skeleton.
- **Joint angle mismatch.** Root orientation is unable to be computed by analytical solution from 3D skeleton. Pose estimation and reconstruction have different forms of representation, which causes a large accuracy gap between these two types of representations [14, 27, 43, 51]. This can be addressed by IK methods.

As is illustrated in Fig. 3, 3D joints (blue ones) are utilized to match the corresponding local 3D rotations ($\boldsymbol{\theta}_{main}$). 3D joints (orange ones) in 3D skeleton (including head, hand, and foot) lack sufficient kinematics constraint (please refer to DIK module), thus we use PR module to learn the suitable local 3D rotations ($\boldsymbol{\theta}_{PR}$) from silhouettes. Additionally, some local 3D rotations ($\boldsymbol{\theta}_{other}$) have little effect on SMPL. To this end, we do not do any matching operations these local 3D rotations ($\boldsymbol{\theta}_{other}$) and set these ones to the default value in SMPL. Please refer to the supplementary materials for more detail.

Differentiable Inverse Kinematics. Inverse kinematics methods have been studied to enable robots to imitate human body motion [33, 39], we thus motivated to design DIK module to transform 3D skeleton to 3D rotation. DIK module relies a minimal set of prior knowledge that defines the underlying kinematic 3D structure. On the basis of the kinematic skeletal structure (i.e., skeletal joint connectivity information in SMPL), a unique mapping equation is applied for each unit via inverse kinematics. That is to say, for each joint (blue circle) in 3D skeleton, we use DIK module to calculate the corresponding local 3D rotations via the specific matching equations respectively.

We choose suitable axis definition to drive skeleton to match SMPL directly, and the definition of coordinate system is exactly similar as SMPL. To clarify this, we describe the detailed matching process of right elbow (see Fig. 4). Specifically, we consider 3D rotation in SMPL as multi-rigid-body system in order to express convenience. Rigid bodies in this system (correspond to joints in 3D skeleton) are then termed as units, which are divided into two categories: connect units and leaf units (shown in Fig. 4). The coordinate system of joint in 3D skeleton is denoted as $[\mathbf{x}_c,\ \mathbf{y}_c,\ \mathbf{z}_c]$, then the parent coordinate system of right elbow $[\mathbf{x}_p,\ \mathbf{y}_p,\ \mathbf{z}_p]$ is calculated by Eqn. 1,

$$\begin{cases} [\mathbf{x}_p,\ \mathbf{y}_p,\ \mathbf{z}_p] = [\dfrac{-\mathbf{l}_{re\_rs}}{|\mathbf{l}_{re\_rs}|},\ \dfrac{\mathbf{l}_{re\_rw} \otimes \mathbf{l}_{re\_rs}}{|\mathbf{l}_{re\_rw} \otimes \mathbf{l}_{re\_rs}|},\ \dfrac{\mathbf{y}_p \otimes \mathbf{x}_p}{|\mathbf{y}_p \otimes \mathbf{x}_p|}] \\ [\mathbf{x}_c,\ \mathbf{y}_c,\ \mathbf{z}_c] = [\dfrac{\mathbf{l}_{re\_rw}}{|\mathbf{l}_{re\_rw}|},\ \mathbf{y}_p,\ \dfrac{\mathbf{y}_c \otimes \mathbf{x}_c}{|\mathbf{y}_c \otimes \mathbf{x}_c|}] \end{cases} \tag{1}$$

where each item (e.g., $\mathbf{x}_c$) is a 3×1 vector in camera coordinate system. $\mathbf{l}_{a\_b}$ indicates the vector pointing from joint b to joint a. Subscripts re, rs, and rw denote right elbow, right shoulder, and right wrist, respectively. Specifically, $\mathbf{l}_{re\_rs}$ means the vector pointing from the right elbow joint to right shoulder joint. The coordinate is defined as left-handed coordinate system, which is the same as SMPL. After declaring the configuration of coordinate system bound with each joint, we can obtain rotation matrix via Eqn. 2,

$$\mathbf{T}_c^p = \mathbf{T}_p^{c\,T} = \begin{bmatrix}\mathbf{x}_p & \mathbf{y}_p & \mathbf{z}_p\end{bmatrix} \begin{bmatrix}\mathbf{x}_c & \mathbf{y}_c & \mathbf{z}_c\end{bmatrix}^T \tag{2}$$

where $\mathbf{T}_\mathbf{c}^\mathbf{p}$ is the transport matrix between the child and the parent coordinate system. Then relative rotation vector $\boldsymbol{\theta}_{re} \in \mathbb{R}^3$ can be calculated via Eqn. 3,

$$\begin{cases} |\boldsymbol{\theta}_{re}| = arccos(\dfrac{tr(\mathbf{T}_c^p) - 1}{2}) \\ \begin{bmatrix} 0 & -\mathbf{r}_z & \mathbf{r}_y \\ \mathbf{r}_z & 0 & -\mathbf{r}_x \\ -\mathbf{r}_y & \mathbf{r}_x & 0 \end{bmatrix} = \dfrac{\mathbf{T}_c^p - \mathbf{T}_c^{p\,T}}{2sin|\boldsymbol{\theta}_{re}|} \end{cases} \tag{3}$$

where $|\boldsymbol{\theta}_{re}|$ is the norm of $\boldsymbol{\theta}_{re}$, and $[\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z]^T = \boldsymbol{\theta}_{re}/|\boldsymbol{\theta}_{re}|$. After performing the similar mapping operation on the other nine units, we can obtain the corresponding ten local 3D rotations termed $\boldsymbol{\theta}_{DIK}$. For more matching details of the other nine units, please refer to the supplementary materials. Note that we only align ten local 3D rotations in SMPL, thus DIK module is able to be generalized to all the datasets with different topologies with little modification of the matching operation.

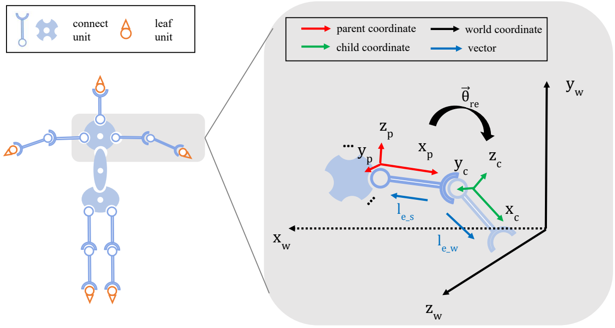

Figure 4: The left is multi-rigid-body system, whose units are rigid bodies including connect unit and leaf unit. Intuitively, connect unit has parent node and child node, and leaf unit only has parent node.The right is process of DIK module taking right elbow as example.

Pose Refinement. In DIK module, we perform the explicit pose mapping in a simple yet efficient manner. Eqn. 2 and Eqn. 3 show that child unit and parent unit are both required to calculate specific local 3D rotation. For example, the local 3D rotation of wrist estimated in 3D pose estimation is commonly unable to be derived by DIK module due to lack of the corresponding parent unit (i.e., hand joint). We thus propose PR module, which aims at capturing end-point local rotations. PR module takes feature maps, encoding silhouette information, as inputs and outputs 3D rotations of endpoint parts (i.e., head, hands, and feet, which are not addressed in DIK module). We experimentally find that PR and SR modules are complementary with each other. Finally, $\boldsymbol{\theta} = \boldsymbol{\theta}_{root} \cup \boldsymbol{\theta}_{DIK} \cup \boldsymbol{\theta}_{PR} \cup \boldsymbol{\theta}_{other}$. where $\cup$ is vector concatenation, $\boldsymbol{\theta}_{PR}$ is the output of PR module, $\boldsymbol{\theta}_{root}$ is the root orientation and $\boldsymbol{\theta}_{other}$ is all other 3D rotations, as in Fig. 4. We also add a regularization term $\mathcal{L}_{preg} = \sum_{i \in \mathcal{S}} \|\boldsymbol{\theta}_{PR,i}\|$ to penalize the magnititude of limb rotations, where $\boldsymbol{\theta}_{PR,i} \in \mathbb{R}^3$ is the ith rotation vector, and set $\mathcal{S}$ indicates the human parts that need refinement. After obtaining refined human pose $\boldsymbol{\theta}$ and shape $\boldsymbol{\beta}$, we obtain the 3D mesh $\mathbf{M} = \mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta})$, 3D keypoints $\hat{\mathbf{J}}^{3D} = \mathbf{RWM}$ and 2D keypoints $\hat{\mathbf{J}}^{2D} = \mathcal{P}(\hat{\mathbf{J}}^{2D}, \boldsymbol{\pi})$. Then we add loss terms to enforce consistency on 2D and 3D keypoints

$$\mathcal{L}_{2D} = \|\hat{\mathbf{J}}^{2D} - \mathbf{J}^{2D}\| \ , \ \mathcal{L}_{3D} = \|\hat{\mathbf{J}}^{3D} - \mathbf{J}^{3D}\| \qquad (4)$$

where $\mathbf{J}^{2D}$ can be ground-truth 2D annotations or predictions of 2D detector. $\mathbf{J}^{3D}$ is predictions from a pretrained lifting module.

3.3. Shape Matching Branch

Compared with 3D skeletons, silhouettes embody abundant cues about body shapes and body orientation. We thus exploit resnet18 [15] with parallel layers attached to learn body shapes $\boldsymbol{\beta}$, the global 3D rotation $\mathbf{R}$, and camera intrinsics $\boldsymbol{\pi}$ respectively from silhouettes.

Specifically, we use a differentiable renderer $\mathcal{F}$ (NMR [23]) taking human mesh $\mathbf{M}$ and a weak-perspective camera $\boldsymbol{\pi}$ as input to render a mask (i.e., $\mathbf{I} = \mathcal{F}(\mathbf{M}, \boldsymbol{\pi})$). Formally, the pixel-level re-projection loss is defined as follows:

$$\mathcal{L}_{mask} = \mathcal{D}(\mathbf{I}, \mathbf{I}_{ref}) \qquad (5)$$

$\mathcal{D}(\cdot, \cdot)$ is a distance function, which can take the form of IoU(Intersection Over Union) and MSE(Mean Squared Error) between rendered mask $\mathbf{I}$ and reference mask $\mathbf{I}_{ref}$ (obtained from an off-the-shelf detector [12]). In addition, $\mathcal{L}_{sreg} = \|\boldsymbol{\beta}\|$ is used to penalize the norm of $\boldsymbol{\beta}$.

3.4. Adaptive Joint Regressor

Joint regressor mapping dense human vertices to 3D skeleton from [35] is coarse due to the following two folds: **(a)** The similar meshes correspond to similar skeletons given specific joint regressor. However, similar meshes in different datasets (e.g. Human3.6M [18] and SMPL [35]) commonly have diverse skeletons. **(b)** Original joint regressor from [35] is trained using ground-truth $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. However, we are unable to obtain 3D annotations. To address these issues, we adopt an adaptive joint regressor $\mathbf{W}$, and then pre-train a joint regressor with objective $\mathcal{L}_{wpre}$,

$$\mathcal{L}_{wpre} = \|\mathbf{J}^{3D} - \mathbf{W}\mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta})\| + \lambda \sum_i \| \sum_j \mathbf{W}_{ij} - 1\|_1 \qquad (6)$$

where $\mathbf{J}^{3D}$ indicates pseudo ground-truth 3D keypoints obtained from pretrained lifting module, the second term is a regularization term encouraging each joint to be represented as a convex combination of vertices. λ is a hyper-parameter. $\mathbf{W}$ is integrated into the model seamlessly without fixing the corresponding parameters, and then optimized by the additional regularization $\mathcal{L}_{wreg}$ (same as the second term in Eqn. 6) is utilized to optimize $\mathbf{W}$.

4. Experiments

4.1. Implementation Details

Network Design. Following [6], we use the residual block as the building block in our framework. We adopt Resnet18 [15] as the CNN feature extractor from silhouettes, where four parallel fully connected layers are attached to perform shape refinement, pose refinement, learn global orientation and learn camera intrinsics.

Training strategy. Without access to the source code, we first re-implement a lifting module according to [6], train the module in a totally unsupervised manner and then freeze all parameters. Details about the lifting module can be found in supplementary material. Besides, we train an adaptive joint regressor as in Sec 3.4. Then we combine loss

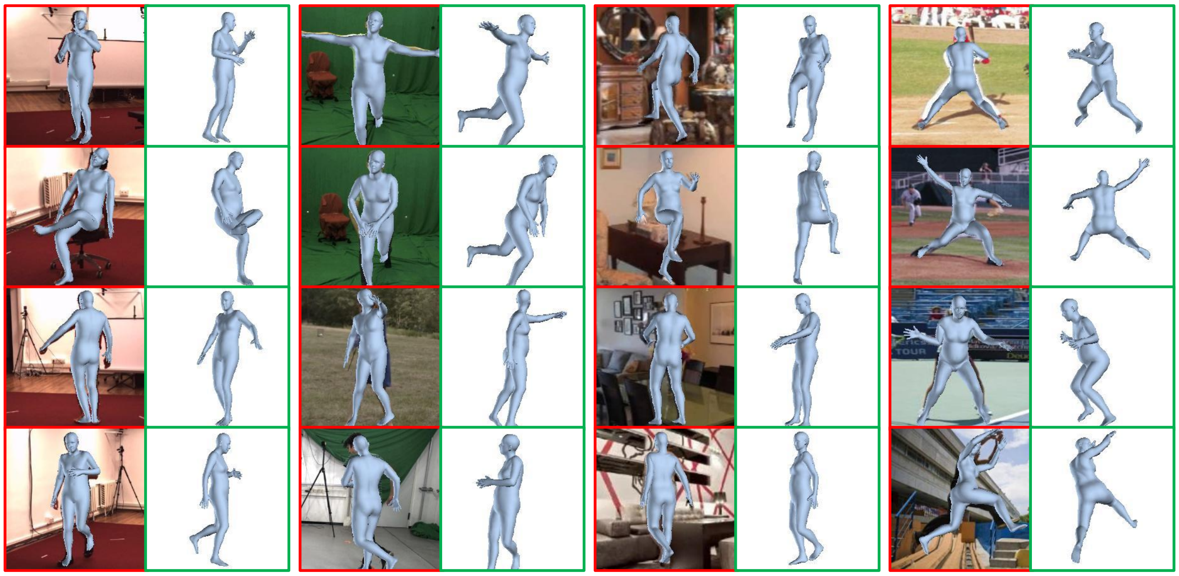

Figure 5: Qualitative results on 4 different datasets. **1st Column:** Human3.6M dataset [18]. **2nd Column:** MPI-INF-3DHP dataset [36]. **3rd Column:** Surreal dataset [50]. **4th Column:** LSP dataset [20]

functions in Eqn. 4, Eqn. 5 and other regularization terms to train our framework.

$$\begin{aligned} \mathcal{L} =& w_{2D}\mathcal{L}_{2D} + w_{3D}\mathcal{L}_{3D} + w_{mask}\mathcal{L}_{mask} \\ &+ w_{sreg}\mathcal{L}_{sreg} + w_{wreg}\mathcal{L}_{wreg} + w_{preg}\mathcal{L}_{preg} \end{aligned} \tag{7}$$

where $w_{2D} = 1.0$, $w_{3D} = 2.5$, $w_{mask} = 0.15$, $w_{sreg} = 0.06$, $w_{wreg} = 1.0$, $w_{preg} = 0.05$ respectively. We adopt IoU as the distance function in SR and downsample the number of vertices in SMPL, as in [28], to speed up the rendering process. We set $\lambda = 0.4$ for joint regressor pretraining. We set batch size to 512, learning rate of all components to $3e-5$ with decay rate 0.95 per epoch. We adopt Adam optimizer and train our framework for 200 epochs.

4.2. Datasets And Metrics.

Human3.6M [18]. Human3.6M is one of the largest indoor datasets with Mosh [34] available. We report mean per-joint position error (MPJPE) and PMPJPE (MPJPE after rigid alignment).

MPI-INF-3DHP [36]. MPI-INF-3DHP is collected both indoors and outdoors. In addition to PMPJPE, we report the Percentage of Correct Keypoints (PCK) thresholded at 150mm and the Area Under the Curve (AUC).

Surreal [50]. Surreal contains many video clips with human characters of various shapes and poses. We report Per-Vertex-Error (PVE) and PPVE (PVE after rigid alignment) to show body shape capture performance.

LSP [20]. LSP consists of 2000 in-the-wild images without ground-truth 3D annotation. We perform qualitative evaluation to illustrate the generalization ability.

4.3. Qualitative Results

Qualitative results on Human3.6M [18], 3DHP [36], Surreal [50] and LSP [20] are exhibited in Fig. 5. Note that to demonstrate the generalization ability of the proposed model, the human mesh on LSP [20] is recovered with the model trained on Human3.6M [18]. As illustrated in Fig. 5, we visualize rendered meshes on background images. Our method can generally provide reasonable and promising results. More visualization can be found in the supplementary material for reference.

4.4. Quantitative Evaluation

Algorithm	3D Data	MPJPE	PMPJPE
HMR [22] CVPR'2018	$\mathcal{P}$	87.9	58.1
HoloPose [14] CVPR'2019	$\mathcal{P}$	-	46.5
SPIN [27] ICCV'2019	$\mathcal{P}$	-	41.1
HybrIK [31] CVPR'2021	$\mathcal{P}$	54.4	34.5
SMPLify [4] ECCV'2016	Pose Prior	-	82.3
HMR [22] CVPR'2018	$\mathcal{U}$	106.8	67.5
SPIN [27] ICCV'2019	Pose Prior	-	62.0
*VIBE [26] CVPR'2020	$\mathcal{U}$	65.6	41.4
Lassner et al. [30] CVPR'2017	None	-	93.9
*PoseNet [48] 3DV'2020	None	-	59.4
Ours	None	**87.1**	**55.4**

Table 2: Results on the test set of Human3.6M[18]. * indicates methods using temporal information. $\mathcal{P}$ and $\mathcal{U}$ indicates paired and unpaired 3D supervision respectively.

Results on Human3.6M [18]. As illustrated in Tab. 2, we obtain 2D joints and silhouettes using an off-the-shelf detector and present mesh recovery results in terms of MPJPE and PMPJPE. We show results from using paired 3D annotation, unpaired 3D annotation(or pose prior), and no 3D annotation. Our method surpasses Lassner et al. [30] under the same settings by a significant margin (55.4 vs. 93.9) in terms of PMPJPE, which is possibly boosted by DIK module. Moreover, we outperform PoseNet3D [48] using temporal information by about 5% in terms of PMPJPE. Moreover, we surpass some methods that use paired 3D data(e.g. NBF [40], HMR [22]) or unpaired 3D supervision(e.g. SPIN [27], proving the effectiveness of our design.

Results on MPI-INF-3DHP [36]. As shown in Tab. 3, we obtain 2D keypoints and silhouettes using off-the-shelf models and present mesh recovery results in terms of PCK and AUC after rigid alignment. Besides recent work PoseNet3D [48], we also compare with previous works that use paired 3D data or unpaired 3d data. Only trained on MPI-INF-3DHP [36], our model is able to outperform VNect [37] with paired supervision and HMR [22] with unpaired supervision. Furthermore, when transferred from Human3.6M [18], our method is able to surpass [48], proving the generalization ability of our model.

Results on Surreal [50]. Surreal [50] is one of the largest synthetic dataset, which has high diversity in human body configuration. We report quantitative results in Tab. 4. Following [49], we use ground-truth 2D keypoints and silhouettes as inputs. Tung et al. [49] use supervised pretraining with paired 3D data, but we surpass their method in terms of PMPJPE and is comparable in terms of PPVE.

Algorithm	3D Data	Training Set	Rigid Alignment		
			PCK	AUC	PMPJPE
Vnect [37]	$\mathcal{P}$	H3.6M+3DHP	83.9	47.3	98.0
HMR [22]	$\mathcal{P}$	H3.6M+3DHP	86.3	47.8	89.8
SPIN [27]	$\mathcal{P}$	Various	92.5	55.6	67.5
HMR [22]	$\mathcal{U}$	H3.6M+3DHP	77.1	40.7	113.2
SPIN [27]	$\mathcal{U}$	Various	87.0	48.5	80.4
*PoseNet [48]	None	H3.6M	81.9	43.2	102.4
Ours	None	H3.6M	83.9	42.5	100.8
Ours	None	3DHP	**87.0**	**50.8**	**87.4**

Table 3: Results on the test set of MPI-INF-3DHP [36]. $\mathcal{P}$ indicates paired supervision and $\mathcal{U}$ indicates unpaired supervision. * indicates methods using temporal information.

Algorithm	3D Data	MPJPE	PMPJPE	PVE	PPVE
*Zhe et al. [52]	$\mathcal{P}$	-	37.1	-	-
Tung et al. [49]	$\mathcal{P}$	203.9	64.4	-	**74.5**
Ours	None	99.5	53.1	**107.8**	75.1
Ours(w/o SR)	None	**95.1**	**52.6**	112.9	80.8
Ours(w/o PR)	None	97.8	53.9	111.1	82.5

Table 4: Results on the validation set of Surreal [50]. * indicates pose estimation method. $\mathcal{P}$ indicates paired 3D supervision used in training or pretraining.

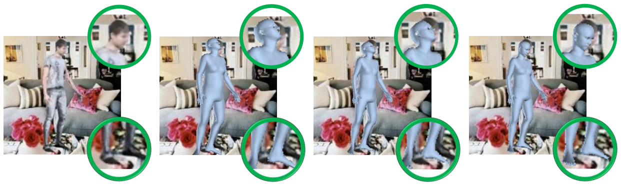

Figure 6: **1st** column: input image. **2nd** column: model w/o differentiable renderer and w/o pose refinement. **3rd** column: model w/ differentiable renderer and w/o pose refinement. **4th** column: model with all components.

Method	Inference Time	Mean Joint Angle Error
Iterative Baseline	30.6s	0.592
DIK	**0.019s**	0.391
DIK + PR	0.169s	**0.389**
DIK + PR^{all}	0.169s	0.404

Table 5: Quantitative results on the DIK module. Evaluation is performed on Human3.6M dataset. Superscript **all** indicates we refine all the local rotations in PR module.

4.5. Ablation Studies

Analysis on DIK module. (a) **Quantitative results**. In DIK module, we directly infer 3D rotations from estimated 3D skeletons. To verify the effectiveness of such kind of DIK module, we compare our method with a learning-based alternative, which learns human poses from estimated 3D skeletons via several residual blocks. As shown in Tab. 7, this learning-based method shows worse performance. Also, we calculate the joint angle error [1] produced by our DIK module and compare with an iterative inverse kinematics counterpart [2]. The results are shown in Tab. 5. Notably, we experiment with refining only endpoint rotations and all the local rotations, and find that the former obtains better performance. This may be caused by an absence of 3D supervision. Our DIK module outperforms simple iterative solver in terms of both speed (*0.019s/it vs 30.6s/it*) and accuracy (*0.391 vs 0.592 joint angle error*). Furthermore, we illustrate that the pose refinement module can improve the joint angle error by correcting endpoint rotations. We perform rigid alignment since the optimization of iterative IK is sensitive to global rotation due to high non-convexity. (b) **Stability of DIK module.** Considering that the discontinuity on the rotation is hard to be formally defined, to verify such kind of property, we compute the maximum position change of the right elbow joint between

[1] Refer to https://github.com/aymenmir1/3dpw-eval

[2] https://github.com/CalciferZh/Minimal-IK

two consecutive frames, which are 0.0647m (ground truth) and 0.0680m (recovered from the outputs of DIK module). The difference is less than 5mm, proving the continuity of the joints. Furthermore, if temporal information is available(e.g. the input is a video sequence), we can identity discontinuous frames by checking the neighbours and correct them. The performance before/after temporal correction is reported in Tab. 6. The discontinuity rarely happens and has negligible effects on the whole sequence.

Sample	PPVE(before TC)	PPVE(after TC)
Discontinuous Frames	81.98	69.70
The Whole Sequence	51.09	51.03

Table 6: Performance before/after TC(temporal correction). Evaluation is performed on Human3.6M dataset.

Analysis on SR module. In Fig. 6, we show qualitative results generated by our method with/without the differentiable renderer (i.e., 2nd and 3rd column). It is obvious that more valid and reasonable human meshes are achieved with the differentiable renderer. Quantitatively, in Tab. 7, we can observe that PVE drops 1.8 points with renderer on Human3.6M [18]. Compared with Human3.6m containing limited subjects, we would like to highlight that for challenging dataset (e.g., Surreal [50]) with diverse body shapes, significant improvement can be observed in terms of PVE and PPVE in Tab. 4. It can be seen that the renderer results in worse performance in terms of MPJPE. The reason is that the differentiable renderer optimizes our model at mesh-level, which does not necessarily mean better performance in terms of joint metrics.

Analysis on PR module. We conduct ablation studies on PR module, show a qualitative comparison in Fig. 6 and report quantitative results in Tab. 4 (Surreal [50]) and Tab. 7 (Human3.6M [18]). As can be seen in Fig. 6, our model with PR module can better capture the poses of limb ends (e.g. head, foot). Since the method only fits silhouettes and 3D joint positions (learned in an unsupervised manner) without any ground truth rotations available, it is hard to learn limb orientations such as head orientations and twisting movements very well. On Human3.6M [18], PR module only improves PVE by 0.5% and PPVE by 0.8%. On Surreal [50], the improvement is more obvious (3.0% on PVE and 9.0% on PPVE). To evaluate the ability to capture limb rotations, we calculate the joint angle error acrross the validation set of Surreal [50], the average error decreases from *0.325* (*w/o* PR module) to *0.314* (*w/* PR module). Moreover, the PR module is complementary to the SR module and can help to capture body shapes. If optimized together, the improvement is more substantial (reported in Tab. 4).

Analysis on adaptive joint regressor. We visualize our joint regressor along with that from GraphCMR [28] in Fig. 7. SMPL based meshes from GT pose/shape and pose/shape predictions (obtained by our framework) are given in the left and right part respectively. In the left part (GT space), our adaptive regressor (blue) gives larger MPJPE w.r.t. ground-truth 3D skeletons (green, directly obtained from 3D annotations) compared with [28] (red). However, in the learnt parameter space, joints from our regressor give a smaller error (85mm) compared with GraphCMR (89mm). This illustrates that during learning process, our regressor can adaptively map SMPL parameters to more accurate 3D joints. Also, from Tab. 7, we experimentally find that our model without adaptive regressor has much worse performance.

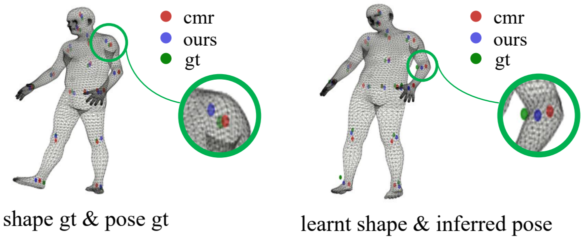

Figure 7: Comparison between our joint regressor and that from GraphCMR [28] on Human3.6M [18]. Green keypoints are directly obtained from 3D annotations. Red keypoints and blue keypoints are obtained from the mesh using joint regressor in [28] and ours respectively. **Left**: Mesh given by ground-truth $\boldsymbol{\theta}$, $\boldsymbol{\beta}$. **Right**: Mesh given by predicted $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ from our framework.

DIK	SR	Regressor	PR	MPJPE	PMPJPE	PVE	PPVE
✗	✓	✓	✓	159.1	111.9	-	-
✓	✗	✓	✓	90.4	**55.3**	122.6	81.5
✓	✓	✗	✓	108.6	63.7	-	-
✓	✓	✓	✗	87.7	**55.3**	121.8	81.4
✓	✓	✓	✓	**87.1**	55.4	**120.8**	**80.8**

Table 7: The analysis on different component modules. Performance is evaluated on test set of Human3.6M [18].

5. Conclusion

In this paper, we decouple unsupervised human mesh recovery into the well-studied problems of unsupervised 3D pose estimation, and human mesh recovery from estimated 3D skeleton. Proposed Skeleton2Mesh, a novel lightweight framework which relies on a minimal set of kinematics prior knowledge. In future, we would like to extend such framework for real-time robot or carton character control.

Acknowledgment This work was supported by National Science Foundation of China (U20B2072,61976137). The authors would like to give a personal thanks to the Student Innovation Center of SJTU for providing GPUs.

References

[1] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.*, 39(4):62, 2020. 1

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005. 1

[3] Andreas Aristidou and Joan Lasenby. FABRIK: A fast, iterative solver for the inverse kinematics problem. *Graph. Model.*, 73(5):243–260, 2011. 3

[4] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter V. Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: automatic estimation of 3d human pose and shape from a single image. In *ECCV*, pages 561–578, 2016. 1, 2, 6

[5] Samuel R. Buss and Jin-Su Kim. Selectively damped least squares for inverse kinematics. *J. Graph. Tools*, 10(3):37–49, 2005. 3

[6] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith MV, Stefan Stojanov, and James M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *CVPR*, pages 5714–5724. Computer Vision Foundation / IEEE, 2019. 1, 2, 5

[7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7103–7112. IEEE Computer Society, 2018. 3

[8] Akos Csiszar, Jan Eilers, and Alexander Verl. On solving the inverse kinematics problem using neural networks. In *24th International Conference on Mechatronics and Machine Vision in Practice, M2VIP 2017, Auckland, New Zealand, November 21-23, 2017*, pages 1–6. IEEE, 2017. 3

[9] Haoshu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6821–6828. AAAI Press, 2018. 3

[10] Georgios Georgakis, Ren Li, Srikrishna Karanam, Terrence Chen, Jana Kosecká, and Ziyan Wu. Hierarchical kinematic human mesh recovery. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVII*, volume 12362 of *Lecture Notes in Computer Science*, pages 768–784. Springer, 2020. 3

[11] Michael Girard and Anthony A. Maciejewski. Computational modeling for the computer animation of legged figures. In Pat Cole, Robert Heilman, and Brian A. Barsky, editors, *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1985, San Francisco, California, USA, July 22-26, 1985*, pages 263–270. ACM, 1985. 3

[12] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *CVPR*, pages 7450–7459. Computer Vision Foundation / IEEE, 2019. 3, 5

[13] Peng Guan, Alexander Weiss, Alexandru O. Balan, and Michael J. Black. Estimating human shape and pose from a single image. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 1381–1388. IEEE Computer Society, 2009. 1

[14] Riza Alp Güler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *CVPR*, pages 10884–10894. Computer Vision Foundation / IEEE, 2019. 1, 2, 4, 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 5

[16] David C. Hogg. Model-based vision: a program to see a walking person. *IVC*, 1(1):5–20, 1983. 1

[17] Charles W. Wampler II. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Trans. Syst. Man Cybern.*, 16(1):93–101, 1986. 3

[18] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 36(7):1325–1339, 2014. 2, 5, 6, 7, 8

[19] Hossam N. Isack, Christian Häne, Cem Keskin, Sofien Bouaziz, Yuri Boykov, Shahram Izadi, and Sameh

Khamis. Repose: Learning deep kinematic priors for fast human pose estimation. *CoRR*, abs/2002.03933, 2020. 3

[20] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12. 6

[21] Marcelo Kallmann. Analytical inverse kinematics with body posture control. *Comput. Animat. Virtual Worlds*, 19(2):79–91, 2008. 3

[22] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, pages 7122–7131, 2018. 1, 2, 6, 7

[23] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, pages 3907–3916. IEEE Computer Society, 2018. 5

[24] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *NIPS*, pages 3809–3819, 2019. 2

[25] Charles A. Klein and Ching-Hsiang Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Trans. Syst. Man Cybern.*, 13(2):245–250, 1983. 3

[26] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: video inference for human body pose and shape estimation. *CoRR*, abs/1912.05656, 2019. 2, 6

[27] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, pages 2252–2261. IEEE, 2019. 1, 2, 4, 6, 7

[28] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, pages 4501–4510. Computer Vision Foundation / IEEE, 2019. 6, 8

[29] Jogendra Nath Kundu, Siddharth Seth, Rahul M. V., Mugalodi Rakesh, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Kinematic-structure-preserved representation for unsupervised 3d human pose estimation. In *AAAI2020*. 2

[30] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *CVPR*, pages 4704–4713, 2017. 1, 2, 6, 7

[31] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. *CoRR*, abs/2011.14672, 2020. 3, 6

[32] Yang Li, Kan Li, Shuai Jiang, Ziyue Zhang, Congzhentao Huang, and Richard Yi Da Xu. Geometry-driven self-supervised method for 3d human pose estimation. In *AAAI2020*. 2

[33] Minas V. Liarokapis, Panagiotis K. Artemiadis, and Kostas J. Kyriakopoulos. Mapping human to robot motion with functional anthropomorphism for teleoperation and telemanipulation with robot arm hand systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, page 2075. IEEE, 2013. 3, 4

[34] Matthew Loper, Naureen Mahmood, and Michael J. Black. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220:1–220:13, 2014. 6

[35] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACMMM*, 34(6):248:1–248:16, 2015. 4, 5

[36] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved CNN supervision. In *3DV*, pages 506–516. IEEE Computer Society, 2017. 2, 6, 7

[37] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: real-time 3d human pose estimation with a single RGB camera. *ACM Trans. Graph.*, 36(4):44:1–44:14, 2017. 7

[38] Itay Mosafi, Eli (Omid) David, and Nathan S. Netanyahu. Deepmimic: Mentor-student unlabeled data based training. In *Artificial Neural Networks and Machine Learning - ICANN, pages = 440–455, year = 2019,*. 1

[39] Jaesung Oh, Buyoun Cho, and Jun-Ho Oh. Remote control for redundant humanoid arm using optimized arm angle. In *17th IEEE-RAS International Conference on Humanoid Robotics, Humanoids 2017, Birmingham, United Kingdom, November 15-17, 2017*, pages 324–331. IEEE, 2017. 3, 4

[40] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *3DV*, pages 484–494. IEEE Computer Society, 2018. 1, 7

[41] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*. 1

[42] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *CVPR*, pages 459–468, 2018. 1, 2

[43] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, pages 7753–7762, 2019. 2, 4

[44] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, volume 11214, pages 765–782. Springer, 2018. 2

[45] Nizar Rokbani, Alicia Casals, and Adel M. Alimi. Ik-fa, a new heuristic inverse kinematics solver using firefly algorithm. In Ahmad Taher Azar and Sundarapandian Vaidyanathan, editors, *Computational Intelligence Applications in Modeling and Control*, volume 575 of *Studies in Computational Intelligence*, pages 369–395. Springer, 2015. 3

[46] Jie Song, Xu Chen, and Otmar Hilliges. Human body model fitting by learned gradient descent. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX*, volume 12365 of *Lecture Notes in Computer Science*, pages 744–760. Springer, 2020. 1

[47] Deepak Tolani, Ambarish Goswami, and Norman I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graph. Model.*, 62(5):353–388, 2000. 3

[48] Shashank Tripathi, Siddhant Ranade, Ambrish Tyagi, and Amit Agrawal. Posenet3d: Unsupervised 3d human shape and pose estimation. *CoRR*, abs/2003.03473, 2020. 1, 2, 6, 7

[49] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *NIPS*, pages 5236–5246, 2017. 1, 2, 7

[50] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 2, 6, 7, 8

[51] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. *CoRR*, abs/2004.13985, 2020. 2, 4

[52] Zhe Wang, Daeyun Shin, and Charless C. Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. *CoRR*, abs/2004.03143, 2020. 7

[53] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3073–3082. IEEE Computer Society, 2016. 3