PLIKS: A Pseudo-Linear Inverse Kinematic Solver for 3D Human Body Estimation

Karthik Shetty1,2  Annette Birkhold2  Srikrishna Jaganathan1,2  Norbert Strobel2,3  Markus Kowarschik2  Andreas Maier1  Bernhard Egger1
1FAU Erlangen-Nürnberg, Erlangen, Germany  2Siemens Healthineers AG, Forchheim, Germany  3University of Applied Sciences Würzburg-Schweinfurt, Germany
karthik.shetty@fau.de

Abstract

We introduce PLIKS (Pseudo-Linear Inverse Kinematic Solver) for reconstruction of a 3D mesh of the human body from a single 2D image. Current techniques directly regress the shape, pose, and translation of a parametric model from an input image through a non-linear mapping with minimal flexibility to any external influences. We approach the task as a model-in-the-loop optimization problem. PLIKS is built on a linearized formulation of the parametric SMPL model. Using PLIKS, we can analytically reconstruct the human model via 2D pixel-aligned vertices. This enables us with the flexibility to use accurate camera calibration information when available. PLIKS offers an easy way to introduce additional constraints such as shape and translation. We present quantitative evaluations which confirm that PLIKS achieves more accurate reconstruction with greater than 10% improvement compared to other state-of-the-art methods with respect to the standard 3D human pose and shape benchmarks while also obtaining a reconstruction error improvement of 12.9 mm on the newer AGORA dataset.

1. Introduction

Estimating human surface meshes and poses from single images is one of the core research directions in computer vision, allowing for multiple applications in computer graphics, robotics and augmented reality [14, 46]. Since humans have complex body articulations and the scene parameters are typically unknown, we are essentially dealing with an ill-posed problem that is difficult to solve in general.

Thanks to models such as SMPL [35] and SMPL-X [44] additional constraints on body shape and pose became available. They made the problem somewhat more tractable. Most state-of-the-art methods [19, 21, 24, 27, 52] directly regress the shape and pose parameters from a given input image. These approaches rely completely on neural networks, while making several assumptions about the image generation process. One typical assumption is the use of a simplified camera model such as the weak perspective camera. In this scenario, the camera is assumed to be far away from the subject, which is generally realized by setting a large focal length constant for all images. A weak perspective camera can be described based on three parameters, two with respect to translation in the horizontal and vertical directions, and the third being scale. While these methods can estimate plausible shape and pose parameters, it can happen that the resulting meshes are either misaligned in the 2D image space or in the 3D object space. This is because the underlying optimization problem is often not constrained enough such that it is difficult for the underlying networks to optimize between the 2D re-projection loss and the 3D loss.

Some existing methods [23, 25, 31] propose a workaround by tackling the problem using a hybrid...
approach involving learning-based and optimization-based techniques while incorporating a full perspective camera [23]. Optimization-based approaches are, however, prone to local minima, and they are computationally expensive. In [25], the authors propose to regress the SMPL parameters by conditioning on features from a CamCalib network meant to predict the camera parameters. Unfortunately, this camera prediction network needs a specialized dataset to train on, which is very hard to acquire in practice. It also prevents end-to-end learning.

On the other hand, recent non-parametric or model-free approaches [33, 40] directly regress the mesh vertex coordinates based on their 2D projections, aligning well to the input image. However, by ignoring the effects of a perspective camera, even these methods suffer from the same limitations as the parametric models.

Motivated by the above observations, we present a novel approach, named PLIKS, for 3D human shape and pose estimation that incorporates the perspective camera while analytically solving for all the parameters of the parametric model. The pipeline of our approach comprises of two modules, namely the mesh regressor and PLIKS. The mesh regressor provides a mapping between an image and the 3D vertices of the SMPL model. Given a single image, any off-the-shelf Convolution Neural Network (CNN) can be used for feature extraction. The extracted features can then be used to obtain a mesh representation either by using 1D CNNs [40], GraphCNNs [28], or even transformers [33]. This way, correspondences to the image space can be found and a relative depth estimate can be computed. From the image-aligned mesh prediction, we can roughly estimate the rotations with respect to a template mesh in canonical space with the application of Inverse Kinematics (IK), denoted in this work as the Approximate Rotation Estimator (ARE). Finally, we reformulate the SMPL model as a linear system of equations, with which we can use the 2D pixel-aligned vertex maps and any known camera intrinsic parameters to fully estimate the model without the need for any additional optimization. As our approach is end-to-end differentiable and fits the model within the training loop, it is self-improving in nature. The proposed approach is benchmarked against various 3D human pose and shape datasets, and significantly outperforms other state-of-the-art approaches.

To summarize, the contribution of our paper is the following: (1) We bridge the gap between the 2D pixel-aligned vertex maps and the parametric model by reformulating the SMPL model as a linear system of equations. Since the proposed approach is fully differentiable, we can perform end-to-end training. (2) We propose a 3D human body estimation framework that reconstructs the 3D body without relying on weak-perspective assumptions. (3) We show that our approach can improve upon other state-of-the-art methods when evaluated across various 3D human pose and shape benchmarks.

2. Related Work

Recovering human pose and shape from monocular images has been extensively studied using both model-based [21, 24, 52] and model-free approaches [11, 33, 40]. Model-based methods estimate the parameters of a parametric body model such as SMPL [35] based on a single input RGB image. The use of parametric body models makes it possible to enforce strong statistical priors of the human mesh. Model-based methods can be further split into optimization and regression techniques. Optimization-based approaches [3, 6] make use of 2D keypoints estimated by a Deep Neural Network (DNN) which are iteratively fit with the SMPL model. These methods however are sensitive to initialization and are susceptible to local minima. Regression-based techniques based on DNNs directly estimate the pose and shape parameters [10, 21, 42, 45, 52, 54]. However, these approaches typically require a large amount of training data. SPIN [27] provided a revolutionary architecture combining optimization-based techniques with regression-based methods. This allowed for much stronger supervision and improved performance on mesh accuracy. However, due to the difficulty in directly estimating the mapping from a single image to the shape and pose space, the mesh alignment with respect to the input image is often imperfect [17, 29]. Model-free approaches directly regress the vertices based on intermediate representations, such as IUV maps [11, 56, 59], 2D/3D heatmaps [17, 29], silhouette [49, 57], and direct vertex regression [28, 32, 33, 40] where correspondence is established between the model and the input image.

To overcome the issues of the learning-based and optimization-based approaches, hybrid techniques combining both approaches have been proposed [7, 17, 29, 30]. In contrast to SPIN [27], here we have a closed-form solution. First, the learning-based approach localizes 3D human joint coordinates [4, 5, 16, 38], utilizing volumetric heatmaps or GraphCNNs for the target representation. From the localized 3D joints, the swing rotations are then determined analytically, whereas the twist rotations, shape parameters, and root translation are either predicted by a network or iteratively optimized [17, 29]. Typical human reconstruction methods [21, 27] take a cropped input image while using 3D predictions projected onto the cropped image for 2D supervision. This, however, ignores the effects of perspective warping. This happens when the cropped image is off-center resulting in inaccurate rotations [58]. To overcome this, some approaches [17, 23] perform iterative optimization [3] after the initial DNN-based predictions. SPEC [25] proposes to condition the image features from a camera calibration network. How-
ever, a few methods include perspective warping during the cropping process and add an implicit camera rotation as post-processing [36, 58]. CLIFF [31] addresses the problem by incorporating the bounding box information into the cropped image.

3. Methodology

In this section, we present our network architecture which includes an analytical solver for inverse kinematics and is end-to-end trainable. As illustrated in Fig. 2, our network consists of two parts, first, a mesh regressor and second, a Pseudo-Linear Inverse Kinematic Solver (PLIKS). In Sec. 3.1, we go over the SMPL model along with its forward kinematics process, and in Sec. 3.2, we explain the full pipeline for mesh reconstruction.

3.1. Parametric Mesh Representation

We use the Skinned Multi-Person Linear (SMPL) model to parameterize the human body [35]. The SMPL model is a statistical parametric function $M(\beta, \theta; \Phi)$. The output of this function is a triangulated surface mesh with $N = 6890$ vertices. The shape parameters $\beta$ are represented by a low dimensional principal component which maps the linear basis $B$ from $\mathbb{R}^{\beta|B}$ to $\mathbb{R}^{3N}$, representing offsets to the average mesh $\bar{x}_m$ as $x = \bar{x}_m + \beta B$. The pose of the model is defined with the help of a kinematics chain involving a set of relative rotation vectors $\theta = [\theta_1, \ldots, \theta_K] \in \mathbb{R}^{K \times 3}$, made up of $K = 24$ joints represented using axis-angle rotations. Additional model parameters summarized as $\Phi$ are involved in the deformation process of the SMPL model. They are used as joint regressor $J \in \mathbb{R}^{K \times 3N}$, blend weights $\mathcal{W} \in \mathbb{R}^{K \times N}$ and shape deformations conditioned on the body pose $B_P(\theta)$. Starting from a mean template mesh, the desired body mesh is obtained by applying forward kinematics based on the relative rotations $\theta$ and shape deformations $\beta$. The 3D body joints can be obtained by a linear combination of the mesh vertices using any desired linear regressor $J' (M(\beta, \theta; \Phi))$.

Simplified SMPL Although the SMPL model described above is intrinsically linear, it cannot be defined as a linear system of equations to solve the IK due to the forward kinematics and pose-related shape deformations $B_P(\theta)$. To deal with this problem, we make use of a simplified model, where we ignore the pose-related shape deformations $B_P(\theta)$ in the formulation. We further split the SMPL mesh into $K=24$ segments $S_k \in (1,K)$, corresponding to the influence of the 24 rotation vectors $\theta$ of the SMPL model as shown in Fig. 3. A vertex on the mesh with index $i$ belongs to a particular segment that has the maximum blend weight of all the joints, i.e., $\arg \max_{k \in (1,K)} W(i)$. For ease of notation, we represent a set of vertices/indices corresponding to a segment $s_k$ with the superscript $k$. Here, we also consider the individual segments to be a set of rigid 3D points which enables us to orient these segments from the template mesh $\bar{x}_m$ to a network predicted mesh $X^k$ to obtain an initial rotation estimate using ARE explained in the following section.

3.2. Pseudo-Linear Inverse Kinematic Solver

Solving Inverse Kinematics (IK) from 2D correspondences is a challenging task due to the inherent non-linearity that exists in the parameterized mesh generation process. We propose to solve IK from 2D pixel-aligned vertex inputs by building a linear system of equations of the form $Ax=b$. We add shape constraints to obtain the optimal world pose $\hat{\theta}$, shape $\hat{\beta}$, and world translation $t$. Linear-Least-Squares is used to estimate the optimal parametric solution making the entire pipeline end-to-end differentiable. From the world pose $\hat{\theta}$, the relative rotations $\theta$ can be inferred by recursively solving the kinematic tree. For the linear system, we assume the rotations as a first-order Taylor approximation [2] with the individual segments of the predicted mesh model approximately oriented $\hat{\theta}$ along the optimal solution. Based on this assumption, using the rotations estimated from ARE should provide the exact solution in a single optimization step.

Regression Network Assuming that some form of dense vertex correspondences exists that facilitates mapping between 3D vertices to pixels in the 2D image plane, we can then incorporate the IK solver into the network. As shown in Fig. 2, our architecture is comprised of an encoder, a mesh regressor, and a parameter regressor. The encoder is based on HRNet-W32 [51], the mesh regressor is based on MeshNet [40], whereas the parameter regressor is a set of fully connected layers.

The encoder acts as a feature extractor $F \in \mathbb{R}^{C \times c_x \times c_y}$, with a channel dimension $C=480$, height and width $c_x=58$. It takes a cropped image $I \in \mathbb{R}^{3 \times 224 \times 224}$ of a person as input. Similar to MeshNet [40], we make use of 1D convolutions to generate four feature vectors

$$P = \{\{P^u, P^v, P^d\} \in \mathbb{R}^{N \times 58}, w \in \mathbb{R}^N\},$$

for each mesh vertex. Here, $P^u$ and $P^v$ represent the features along the $u$ and $v$ axis, $P^d$ represents the features along the root-normalized depth, and $w$ represents the weighting factor. They are obtained as follows

$$P^u = f^u_{1D}(\text{avg}(u(F))), P^v = f^v_{1D}(\text{avg}(v(F))),$$

$$P^d = f^d_{1D}(\psi_d(\text{avg}(u,v(F)))),$$

$$w = \sigma(\text{avg}(f^d_{1D}(\psi_d(\text{avg}(u,v(F)))))),$$

\begin{equation}
\end{equation}
reconstructed body mesh is obtained via an analytical solution through the PLIKS process, fed with \( \tilde{v} \) vertices aligned to the image plane. The ARE module provides rotation estimates in world space for each joint segment of the SMPL. The assumed to be rigid.

Figure 2. Overview of the proposed framework: For a given RGB image input, the mesh regressor generates a mapping to the 3D mesh vertices aligned to the image plane. The ARE module provides rotation estimates in world space for each joint segment of the SMPL. The reconstructed body mesh is obtained via an analytical solution through the PLIKS process, fed with \( \tilde{\theta} \) comprising the segments’ initial world rotations and the 2D mesh projections \( X_{uv} \) as inputs.

Here \( f^{1D}(.) \) and \( \psi(.) \) represent a 1D convolution along the \( i \)-th dimension, which converts the features \( F \) from \( C \times c' \rightarrow N \times c \) and \( C \times 1 \rightarrow (c' \times C)^T \), respectively. The average function \( \text{avg}(.). \) averages the features along the \( i \)-th dimension. For the weighting factor \( w \), we average across the channel dimension, followed by applying the sigmoid activation function \( \sigma(.) \) \[41\]. More details about the weighting function \( w \) are explained in the following sections.

We then concatenate and process \( \tilde{P} = \{P^u, P^v, P^d \} \) using a graph convolution network (GCN) to predict \( uvd \in \mathbb{R}^{N \times 3} \). We use GCN rather than the heatmap-based approach from MeshNet \[40\] to avoid any truncation-based artifacts, which are prominent when partial images of humans are supplied as inputs. For the GCN we use the formulation from Kipf et al. \[22\], defined as \( G = \sigma(\mathbf{APW}) \), where \( \mathbf{A} \in \mathbb{R}^{N \times N} \) denotes the graph adjacency matrix, \( \mathbf{W} \in \mathbb{R}^{c \times l} \) denotes the trainable weights with \( l \) as the output channel dimension, and \( \sigma \) denotes the ReLU \[1\] activation function. We make use of 3 GCNs in series, with channel sizes 64, 32, and 3 respectively. The final output \( X_{uvd} \in \mathbb{R}^{N \times 3} \) acts as the vertex correspondence in the image coordinate system.

The parameter regressor is a set of fully connected layers to obtain an approximate shape \( \tilde{\beta} \) and a weak perspective camera \( \tilde{c} \in \mathbb{R}^4 \), which are used to determine the approximate world rotations later. We make use of the estimated depth \( \tilde{d} \) from the camera prediction to obtain a mesh in the world coordinate system \( X \) as follows,

\[
X = (K^{-1}X_{uvd}^T) \cdot (X_{uv}^T + \tilde{c}_d),
\]

where \( K \in \mathbb{R}^{3 \times 3} \) is an intrinsic matrix with fixed focal length of 1000.

Approximate Rotation Estimator (ARE) Given two sets of corresponding points in 3D space, it is possible to obtain an optimal rotation as a closed-form solution using the Kabsch solver \[20\]. For a given segment from network mesh prediction \( X^k \) we make use of the Kabsch solver to determine the rotation that a same segment from the template mesh needs to go through from its rest pose. As the mesh prediction \( X^k \) can represent a wide range of human shapes, we make use of the shape predictions \( \tilde{\beta} \) on the mean shape as \( x^k = \bar{x}_m^k + (\beta \mathbf{B})^k \). The pose solver minimizes the squared distances between a set of 3D correspondences to obtain an optimal pose as follows,

\[
(U \Sigma V^T)_{kk} = C_{kk} = \sum_{(X^k, x^k)} (X^k - \bar{x}_m^k)^T w^k \Lambda^k.
\]

Figure 3. We split the SMPL model into 24 segments based on the blend weights. These segments help in determining the initial rotation estimate in the ARE pipeline. Here the segments are assumed to be rigid.
Here $C_k$ is the covariance matrix between the correspondences. In this context, $\tilde{x}^k$ and $\tilde{X}^k$ represent the means over the associated point sets. The world rotation for the $k$-th segment $\tilde{\theta}_k = (VU^T)_k$ is obtained by applying singular value decomposition (SVD) over the covariance matrix. Since the SVD is differentiable, gradients can be backpropagated during the training process. Due to the blending process contributes. 

The obtained rotations are in the world space, whereas the SMPL model expects relative rotations in its axis space. The rotation around the pelvis $\tilde{\theta}_k^{pelvis}$ corresponds to the global root rotation. The relative rotations for the other joints can be recursively based on the parent rotation following a pre-defined kinematic tree for the SMPL model. To this end, let $\tilde{R}_k = R_k(\tilde{\theta}_k) \in SO(3)$ represent the estimated world rotation determined for segment $k$. Then the relative rotation for segment $k$ is

$$\hat{\theta}_k = \tilde{R}_k = \hat{R}_{p(k)}^{-1} \tilde{R}_k,$$  \hspace{1cm} (4)

where $p(k)$ represents the parent joint of $k$ and $\hat{\theta}_k$ the estimated relative pose.

**Pseudo-Linear Inverse Kinematic Solver (PLIKS)** An approximate SMPL model projected onto the image plane, with no pose-related blend shapes can be represented as

$$t^k = \tilde{K} \sum_{j=1}^{K} W_j^k (\Delta R_k \tilde{R}_k \tilde{x}_m^k + \beta B^k) + t_k.$$

$$\text{(5)}$$

Here, the superscript $k$ refers to the $k$-th’s subset of vertices defined in Sec. 3.1. The matrix $\tilde{K} \in \mathbb{R}^{3 \times 4}$ represents a perspective projection matrix taking into account the affine transformation (only crop and resize) for the image fed into the network as

$$\tilde{K} = \begin{bmatrix}
    f_x & 0 & p_x & 0 \\
    0 & f_y & p_y & 0 \\
    0 & 0 & 1 & 0 
\end{bmatrix}.$$

Further, $\tilde{R}_k$ represents the approximate world rotation obtained from the previous step, $\Delta R_k$ is defined as the additional rotation required to get an optimal solution, $t^k$ represents the joint translation, and $W$ are the blend weights. By combining all segments, we get $t = X^I_{\text{all}} \in \mathbb{R}^3$. This represents the correspondence between mesh vertices, defined in homogeneous coordinates, and the pixels in the image plane.

Since the additional rotation required is considered to be small we linearize the rotation matrix which needs to be determined based on Taylor expansion with angles $\alpha, \beta, \gamma$ along the $x, y$ and $z$ axes, respectively, as follows

$$\Delta R_k = \begin{bmatrix}
    1 & -\gamma_k & \beta_k & 0 \\
    \gamma_k & 1 & -\alpha_k & 0 \\
    -\beta_k & \alpha_k & 1 & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix}.$$

Further, we simplify the projected SMPL model in Eq. (5) by making some additional assumptions using the definitions introduced in Eq. (6). For the $x^k$ term we assume that the majority of the rotation is significantly effected by the rotation with respect to the primary segment, i.e., we ignore the impact of neighbouring rotations as they are usually minuscule. This assumption is also made for the term $B^k$ by assuming that for small rotations $\Delta R_k \tilde{R}_k \Delta B_k \approx \tilde{R}_k \beta B_k$.

$$t^k = \tilde{K} \left( \Delta R_k \sum_{j=1}^{K} W_j^k \tilde{R}_k \tilde{x}_m^k + \beta \sum_{j=1}^{K} W_j^k \tilde{R}_k B^k + t_k \sum_{j=1}^{K} W_j^k \right).$$

$$\text{(6)}$$

The equation is now linear, with 154 unknown parameters corresponding to $\Delta R_k$, $\beta$ and $t_k$. If the focal lengths $f_x, f_y$ and principal points $p_x, p_y$ are not known, we assume a fixed focal length of 1000 mm and the image center as the principal point. Note that the fixed values here apply to the entire image, i.e., not to the cropped and resized input fed to the network.

Using Direct Linear Transform (DLT) [12] we can rewrite Eq. (6) in the form $Ax = b$. The optimal parameters can be obtained by minimizing the analytical error using linear least square defined by $x = A^+b$, where $A^+$ represents the pseudo-inverse of $A$. As the pseudo-inverse of a tall matrix is differentiable [8], gradients can be backpropagated during the training process. One of the major drawbacks when using DLT is that it minimizes the analytical loss rather than the geometric loss. One option to overcome this is to use Iterative Reweighted Least Squares (IRLS) [9], which robustly minimizes the objective function in an iterative manner by reweighing the geometric loss. However, we make use of the network predicted weighting $w^k$ to weight the correspondences [26, 47]. To further enforce the predicted shape to be close to the mean shape, we add an additional constraint such that $||\beta||_2 \approx 0$ with a regularizing weight $\omega_{\beta}$ as

$$\arg\min_{\Delta R_k, \beta, t_k} \|w^k \left( t^k - \tilde{K} (\Delta R_k \tilde{x}_m^k + \beta B^k + t_k w^k) \right)\|_2 + \omega_{\beta} ||\beta||_2.$$  \hspace{1cm} (7)

To get the final relative pose $\theta_k$ for each joint $k$, we apply Eq. (4) on the obtained world rotations $\Delta R_k \tilde{R}_k$. The
global translation for the camera system $\hat{K}$ is obtained from the root joint as $t_1 = (\hat{j}_1 - \Delta R_1 \hat{R}_1 \hat{j}_1)$, where $\hat{j}_1$ is the root joint for a rest pose mesh with shape coefficient $\beta$.

### 4. Experiments

Following previous works, the base PLIKS model is trained on a combination of 3D datasets (Human3.6M [15] and MPI-INF-3DHP [36]), and 2D dataset (COCO [34]) with pseudo-GT labels obtained from EFT [19]. We evaluate on Human3.6M [15], 3DPW [55], MPI-INF-3DHP [36], AGORA [43], MuPoTs-3D [37], and 3DOH50K [61]. During the evaluation, we highlight our results for networks trained with any additional datasets.

#### 4.1. PLIKS Error Analysis

We measure the effective reconstruction capability of ARE and PLIKS with ground truth conditions for the 3DPW [55] test set and the AGORA [43] validation set. For the ARE module experiment, we provide the GT SMPL mesh vertex projections and the GT shape parameters. Additionally, we set the root depth as 7 m to the ARE module along with the GT bounding-box camera intrinsics as inputs. We also set the weights for the shape regularizer $\omega_\beta=0$. We measure the Mean Per Joint Position Error (MPJPE), and the Per Vertex Error (PVE). Following previous works [21,24,27], we use the LSP joint regressor [27] to determine the 14 joints which are regressed from the body mesh. We report the results in Table 2 showing that the assumptions made in Eq. (6) are reasonable providing a stable and accurate fit. We observe a larger error on the AGORA dataset with the PLIKS module, due to larger perspective warping effects, which results in incorrect global rotation estimation from the ARE module. As PLIKS uses a linearized rotation matrix, running the PLIKS module twice, that is iteratively optimizing the rotation estimates provides a more accurate fit. Though ARE is not an accurate IK solver, we can safely conclude about the drawbacks of using a weak perspective camera model.

#### 4.2. Comparison with the State-of-the-art

We compare our method with previous human mesh reconstruction approaches based on Human3.6M, MPI-INF-3DHP, and 3DPW datasets. To remain consistent with previous approaches, we use the LSP regressor [27] to obtain the 14 joints for Human3.6M and 3DPW datasets, while we
use 17 joints with the Human3.6M regressor [27] for
the MPI-INF-3DPW dataset. We measure Procrustes Aligned
Mean Per Joint Position Error (PA-MPJPE), Percentage of
Correct Keypoints (PCK), and Area Under Curve (AUC) on
the 3D pose results.

In Table 1, we provide quantitative results for previous
3D human shape and pose estimation results. We use the
best results reported in all the other works for our com-
parison. Our network outperforms all previous state-of-
the-art techniques on MPI-INF-3DHP and 3DPW datasets
along with the lowest MPIPE on Human3.6M. Further fine-
tuning with the AGORA dataset results in a significant per-
formance improvement. We also provide results of our
network when trained on a larger HRNet-W48 [51] back-
bone following previous works [31, 33, 60]. We provide
qualitative results of our approach comparing prior meth-
ods [29, 31] in Fig. 4. Though all methods shown in Fig. 4
align well in the 2D projected space, it is apparent that the
prior methods fail to align well in 3D space due to perspec-
tive warping. The effect of perspective warping is observed
significantly when the human is off-centered. On the MPI-
INF-3DHP dataset, the advantage of incorporating PLIKS
is more pronounced since the images are captured with a
wide Field-of-View (FOV) (60°~90°) showing a 26.3 mm
MPJPE improvement.

In Table 5, we list quantitative results on the AGORA
benchmark. It can be seen that the proposed approach sig-
nificantly outperforms all prior methods. AGORA addi-
tionally measures the F1 score to get Normalized Mean Joint
Error (NMJE) and Normalized Mean Vertex Error (NMVE),
which penalizes missed or faulty human detection. Here the
human bounding box and camera parameters are not pro-
vided for the test set. We make use of YOLO-v5 [18, 39] for
person detection. From Table 5, PLIKS was fine-tuned only
on AGORA, whereas, PLIKS† was trained on all the 2D and
3D datasets which helps in generalizing to real-world im-
ages. As the focal length is not available during the test, we
set \( f = 1 \) m for all images. Making use of the estimated fo-
cal length from CamCalib [25] on PLIKS† shows a 3.4 mm
NMJE improvement (PLIKS†).

Most methods reported in Table 1 generate root relative
meshes using a weak perspective camera. As our results
are in absolute camera coordinates, we present root local-
ization results in Table 3. We report previous results on
Human3.6M provided from [38] on the mean root position
error (MRPE). Here baseline [36, 48] depth is obtained us-
ing the least squares fit from the predicted 3D joints and
its corresponding 2D projections. Though [38] has explic-
itly been designed to learn the absolute depth, we observe
comparable results along the depth while performing better
in the horizontal and vertical directions. Using 3DPW and
AGORA for training results in significant improvement of
the root localization error. We also report the Absolute PCK
result on the MuPoTs-3D [37] dataset in Table 4. Note that
the results are not strictly comparable due to differences in
network backbones, the training datasets, and the joint pre-
diction techniques.

4.3. Occlusion Analysis

To validate the stability under occlusion, we evaluate
PLIKS on the object-occluded benchmark dataset
3DOH50K [61] and the person-occluded dataset 3DPW-
OCC [55]. Following [24], we use only COCO, Hu-
man3.6M, and 3DOH datasets for training. To arrive
at a fair comparison, we also train a network with
ResNet50 [13] as the backbone. We observe better occlu-
sion performance on both backbones from Table 6 for our
approach. This can be attributed to the fact, that our ap-
proach is intrinsically trained on dense correspondences.

4.4. Ablation Studies

In this context, we evaluate the individual components of
our approach. For all experiments, we use only the COCO,
Human3.6M, and MPI-INF-3DHP datasets for training and
evaluation on all 3D datasets. We use the 3DPW valida-
tion set to determine the best model. In Table 7, we provide
all the results for our ablation studies. The ARE module on
its own can reconstruct 3D models provided the shape and
camera parameters are regressed by the network. Hence,
we evaluate its performance without the PLIKS module.
Here, we observe that it performs poorly on all metrics for
different datasets except for the PA-MPJPE of 3DPW. Next, on
the same ARE module, we replace the HRNet-32 backbone
with a ResNet-50 backbone. In this setting, similar to the

\[ \begin{array}{|c|c|c|c|c|}
\hline
\text{Method} & \text{F1} & \text{NMVE} & \text{NMJE} & \text{MVE} & \text{MPIPE} \\
\hline
\hline
\text{MPJPE} & 0.77 & 193.4 & 199.2 & 148.9 & 153.4 \\
\text{SPEC} [25] & 0.84 & 113.6 & 118.8 & 103.4 & 108.1 \\
\text{BEV} [53] & 0.93 & 108.3 & 113.2 & 100.7 & 105.3 \\
\text{EFT} [19] & 0.96 & 90.2 & 95.5 & 84.8 & 89.8 \\
\text{CLIFF} [31] & 0.91 & 83.5 & 89.0 & 76.0 & 81.0 \\
\hline
\text{PLIKS} & 0.94 & 76.8 & 81.5 & 72.2 & 76.6 \\
\text{PLIKS} & 0.94 & 78.3 & 83.0 & 73.6 & 78.0 \\
\text{PLIKS} & 0.94 & 74.9 & 79.6 & 70.4 & 74.8 \\
\text{PLIKS} & 0.94 & 71.6 & 76.1 & 67.3 & 71.5 \\
\hline
\end{array} \]

Table 5. Reconstruction errors on the AGORA test set. All re-
sults are taken from the official evaluation platform. PLIKS† was
trained with all 2D and 3D datasets. PLIKS‡ represents the usage

\[ \begin{array}{|c|c|c|c|}
\hline
\text{Method} & \text{MPJPE} & \text{PA-MPJPE} & \text{MPIPE} & \text{PA-MPJPE} \\
\hline
\text{DOH} [51] & - & 72.2 & - & 58.5 \\
\text{EFT} [19] & 94.4 & 60.9 & 75.2 & 53.1 \\
\text{PARE} [24] & 90.5 & 56.6 & 63.3 & 44.3 \\
\text{PLIKS} [50] & 86.1 & 53.2 & 51.5 & 39.3 \\
\hline
\end{array} \]

Table 6. Evaluation on occlusion datasets 3DPW-OCC and 3DOH.
occlusion test, HRNet performs better than ResNet. Next, we evaluate the importance of accounting for camera intrinsics by training a network with the PLIKS module but with all cropped images given as input to the network having a fixed focal length of $f=0.3$ m. The low focal length value violates the assumption of a weak-perspective setting. We observe slight improvements compared to that of ARE on the Human3.6M, whereas we see on-par results with the other datasets. Finally, we determine the effect of using a focal length of $f=\sqrt{W^2 + H^2}$ proposed in [23] only during inference. While there is no significant difference in 3DPW and Human3.6M, we observe a drop in performance for the MPI-INF-3DHP dataset compared to PLIKS with known camera intrinsics. This is due to the actual FOV being larger than the estimated FOV. Our ablations clearly demonstrate the importance of incorporating camera intrinsics into a network.

5. Conclusion

In this paper, we bridge the gap between 2D correspondence and body mesh estimation by creating a pipeline from which the inverse kinematics can be solved in closed form. This can be effectively leveraged to use the full perspective projection rather than having to rely on weak-perspective counterparts. Our approach yields considerable improvements in both root-relative and absolute-3D estimation for human pose estimation. PLIKS further enables our method to be fully differentiable facilitating end-to-end training. We validated the effectiveness of our method on various 3D pose and shape datasets and achieved state-of-the-art on multiple benchmarks. Due to the inherent nature of a built-in solver, we can extend our work with additional constraints like using multi-view systems for even higher reconstruction accuracy or temporal constraints for smooth video reconstruction should be feasible and foster future research in this direction.

Disclaimer The concepts and information presented in this article are based on research and are not commercially available.
References


[52] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In ICCV, 2021. 1, 2, 6

[53] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. Putting people in their place: Monocular regression of 3d people in depth. In CVPR, 2022. 7


