MULTI-VIEW HUMAN MODEL FITTING USING BONE ORIENTATION CONSTRAINT AND JOINTS TRIANGULATION

Jordy Ajanohoun *Eric Paquette* *Carlos Vázquez*

École de technologie supérieure, Montreal, Quebec, Canada

ABSTRACT

We address 3D human pose and shape estimations from multi-view images. We use the SMPL body model, and regress the model parameters that best fit the shape and pose. To solve for the parameters, we first compute 3D joint positions from 2D joint estimations on images by using a linear algebraic triangulation. Then, we fit the 3D parametric body model to the 3D joints while imposing a bone orientation constraint between the 3D model and the corresponding body parts detected in the images. We do so by minimizing a new set of objective functions through a two-step optimization process that provides a good initialization for the refinement of the shape and pose parameters. Our approach is evaluated on the Human3.6M and HumanEva benchmarks, showing superior results with respect to state-of-the-art methods.

Index Terms— 3D reconstruction, shape and pose estimation, body model, multi-view

1. INTRODUCTION

3D body reconstruction benefits many applications like virtual reality, healthcare, human tracking, and video games. The emergence of realistic human body models, such as the Skinned Multi-Person Linear (SMPL) model [1], has resulted in substantial improvements in human pose and shape estimations. Many methods focus on estimating the shape and pose from images by trying to fit the SMPL 3D model to 2D features, such as silhouettes and 2D joint locations.

3D human pose estimation. Two paradigms stand out in the literature: the direct regression of 3D joints from images or the estimation of 2D joints followed by their lifting to 3D. Direct 3D joint regressions is mostly achieved by training a Convolutional Neural Network (CNN) in an end-to-end manner [2, 3, 4]. In two-stage methods, 2D pose estimation [5] is first performed, after which 2D estimates are lifted to 3D. To this end, various strategies have been applied, such as: neural networks [6], dictionary [7], pictorial structure models [8], triangulation [9], and 3D-aware 2D pose estimation [10].

Simultaneous shape and pose estimation. Most of these methods are based on a statistical human body model which encodes and parametrizes human shape and pose spaces. SMPL [1] is the leading body model used for this task. The main methods can be classified [11] as either optimization-based [12, 13] or CNN-based [14, 15, 16, 17]. Because end-to-end training data is lacking, CNN methods are generally less accurate than optimization-based methods [11, 17]. In addition, they do not generalize well either [11].

Optimization-based solutions require human-made priors and constraint terms to relax the objective function. SMPLify [13] uses an iterative optimization process to find the optimal shape and pose. The energy function is constructed from estimated 2D joint positions by means of a 2D pose estimator. SMPLify suffers from depth ambiguity issues since it relies on a single view. Rhodin *et al.* [18] adopt a multi-view setting using 2D optimization costs. MuVS [12] also adopts a multi-view setting to reduce depth ambiguity. It is built upon SMPLify [13] and uses a similar optimization process. The main difference is that the objective function considers 2D pose estimation through all views.

We also address the problem of 3D shape and pose estimation using multiple views. Rhodin *et al.* [18] use 2D optimization costs and MuVS [12] aggregates the 2D pose estimations from all the views into a single objective function. Conversely, we rely on two objective functions both directly integrating 3D joint positions. We triangulate these 3D joints from 2D joint estimations by weighting the contribution of each view to the final 3D joint position. To determine the influence of each view, we rely on the 2D pose estimator's confidence values. This leads to better estimates for the 3D joints which are later injected in our shape and pose objective functions. Furthermore, we design a different optimization process with a novel objective function. This function aims to achieve bone orientation consistency between the 3D skeleton and the SMPL model. Thanks to our bone orientation constraint (BOC), we are able to closely approximate the SMPL pose parameter and take advantage of this information when conducting the final optimization stage (simultaneous shape and pose refinement). We demonstrate that the semantic position of joints in SMPL and in the validation data sets do not exactly match. To account for this discrepancy we introduce, for each joint, a shift vector computed in the joint local space. Results on widely used benchmark data sets (HumanEva and Human3.6M) show that our approach has a higher accuracy than the state-of-the-art methods. We summarize the main contributions of our approach as: (1) a bone orientation con-

straint (BOC) to recover the pose parameter independently from the shape parameter, (2) a more precise initialization for the simultaneous optimization of pose and shape parameters thanks to the BOC, and (3) a two-step optimization process that improves the accuracy of the pose and shape estimations.

2. MULTI-VIEW 3D BODY RECONSTRUCTION

Given multi-view images of a human subject taken at the same time, together with camera parameters for each view, our goal is to generate a precise 3D body model. Like in other work [12, 13, 15, 19], we use the SMPL body model [1]. The challenge is to find the 3D shape and pose of the individual from the images. As in other work [12, 14, 17, 20], we use the 2D pose estimation to infer an accurate 3D pose and 3D shape. We first estimate 2D joints on each view. We then triangulate 3D joints (from estimated 2D joints using joint confidence values) to reduce the weight of incorrectly detected joints (Sec. 2.1). For instance, in Fig. 1, this allows us to correctly converge despite the inaccurate left elbow. Finally, we introduce a two-step optimization process (Sec. 2.2) with a new objective function to determine the SMPL pose and shape parameters with the aid of the triangulated 3D joints.

2.1. 3D pose triangulation

We first estimate the 2D pose on each view $v \in \{1, 2, 3, ..., V\}$, where V is the number of views. We use OpenPose [5] which provides 25 joint locations and a confidence value for each joint j. We then use the linear algebraic triangulation from Iskakov *et al.* [9] with OpenPose's joint confidence values to lift the 2D joints to 3D. Given a joint j, its 2D estimated position on each view v, and the camera parameters (intrinsic and extrinsic) of each view, the algebraic triangulation consists in solving the following system of equations:

$$((\mathbf{w}_j \cdot \mathcal{J}) \circ A_j)\tilde{\mathbf{y}}_j = 0 \ , \tag{1}$$

where $\tilde{\mathbf{y}}_j$ is the unknown location of the 3D joint j and $A_j \in \mathbb{R}^{2V \times 4}$ is a matrix that allows to calculate, for all V views, the difference between the estimated 2D joint locations and the projected 3D joint locations. The weights $\mathbf{w}_j = (w_{1,j}, w_{1,j}, ..., w_{V,j}, w_{V,j})^\intercal \in \mathbb{R}^{2V \times 1}$ correspond to confidence values. The weight $w_{v,j} \in [0, 1]$ denotes the confidence in the estimate of joint j on view v. These weights are multiplied (matrix product) by the all-ones row vector $\mathcal{J} \in \mathbb{R}^4$ and operator $\circ$ denotes the Hadamard product. The idea behind Eq. 1, is to recover the homogeneous coordinates, $\tilde{\mathbf{y}}_j \in \mathbb{R}^4$, of the joint j knowing its projection on the V images. The 3D joint $J_{3D_j} \in \mathbb{R}^3$ is computed from the homogeneous coordinates. The system is solved independently for each joint using a differentiable singular value decomposition. We refer the reader to the original paper [9] for full details. Iskakov *et al.* [9] use their own 2D pose estimator (and confidence values) trained to map Human3.6M joints. We use OpenPose instead because its joints are at the same semantic positions as in the SMPL model, which is not the case with Human3.6M (or HumanEva-I) joints (Fig. 2).

The weights $\mathbf{w}_j$ are crucial since they adjust the contribution of each view in the triangulation. When a joint is likely to be occluded in one of the views, the weight for this view is low, ensuring that other views with larger confidences will drive the convergence to the right location. Thus, the emanating 3D joint contains less uncertainty than the 2D ones.

2.2. Two-step optimization process

We now describe our process to infer SMPL parameters. SMPL has two vectors of parameters (shape $\vec{\beta}$ and pose $\vec{\theta}$). As it does not constrain invalid parameters, one may converge to a *non-human* pose or shape if the problem is not constrained enough. Furthermore, SMPL joint locations after posing, $J(\vec{\beta}, \vec{\theta}) \in \mathbb{R}^{3\times 23}$, depend on SMPL joint locations $J(\vec{\beta})$ which are a function of the shape. This means that each modification of the shape $\vec{\beta}$ leads to a change in $J(\vec{\beta}, \vec{\theta})$, even if the pose $\vec{\theta}$ remains unchanged. In SMPLify [13] and MuVS [12], the shape and pose are estimated simultaneously. Therefore, the cost functions used are complex and result in multiple local optima. That is the reason why optimization-based methods are sensitive to the initialization point. In our approach, we overcome these obstacles in a novel fashion. The triangulated 3D joints allow us to provide a robust initialization for $\vec{\theta}$. The proposed optimization process is decomposed into two steps: SMPL mesh bone orientation, followed by simultaneous posing and shaping.

First, $\vec{\beta}$ is initialized to the mean shape and $\vec{\theta}$ to the "A" pose. The first step (bone orientation constraint) estimates only the pose parameter. We want the 3D mesh to be posed as in the multiple views. To that end, we designed a new objective function. Let B be the set of bones of the triangulated 3D skeleton. A bone $b \in B$ is defined by two consecutive joints (child-parent) in the skeleton's kinematic tree. We name these joints child(b) and parent(b). Given a bone b and a 3D pose J (3D joint locations), the function $\Phi(J, b)$ returns the normalized orientation vector of the bone b in J:

$$\Phi(J, b) = \frac{J_{\text{child}(b)} - J_{\text{parent}(b)}}{||J_{\text{child}(b)} - J_{\text{parent}(b)}||_2} \ . \tag{2}$$

Then, our objective function is:

$$\begin{aligned} &E_{pose}(\vec{\theta}) = \lambda_\theta E_\theta(\vec{\theta}) + \\ &\lambda_{bone} \sum_{b \in B} ||\Phi(J(\vec{\beta}, \vec{\theta}), b) - \Phi(J_{3D}, b)||_2^2 \ , \end{aligned} \tag{3}$$

where J_{3D} denotes the triangulated 3D joints, $J(\vec{\beta}, \vec{\theta})$ denotes the SMPL mesh 3D joints, $\lambda_\theta = 1$ and $\lambda_{bone} = 100$ are weights, and $E_\theta(\vec{\theta})$ is a pose prior [12, 13]. The pose prior prevents convergence to *non-human* poses. During the optimization, $\vec{\beta}$ is kept fixed to the mean shape. With Eq. 3,

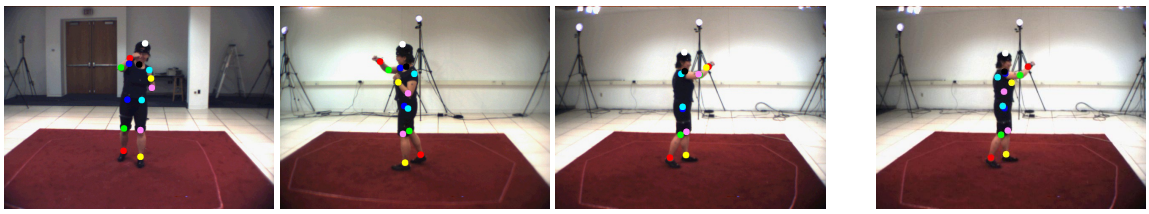

Fig. 1: The first three images correspond to 2D joint estimates. Note that in the third image, the left elbow (pink) is incorrect because of the occlusion. We automatically detect that this joint is likely to be inaccurate, and thus decrease its contribution to the triangulation process, resulting in an accurate 3D joint location thereafter (right-most image).

we are constraining the bones to have orientations consistent with the triangulated 3D joint positions. We are able to get a close approximation of $\vec{\theta}$ alone, without caring about the shape $\vec{\beta}$, because we get rid of the bone lengths by normalizing. Whatever the individual's shape, by minimizing Eq. 3, we are able to obtain bone orientations (*e.g.* $\vec{\theta}$). This strategy resolves concerns arising from the simultaneous optimization of shape and pose used in previous works [12, 13]. One important advantage with our technique is that we can then use this estimation of $\vec{\theta}$ to initialize the final optimization step.

In the final step, we also want to recover the shape, therefore, the bone lengths matter. Our energy function is then:

$$E_{final}(\vec{\beta},\vec{\theta},\vec{\gamma}) = \lambda_\theta E_\theta(\vec{\theta}) + \lambda_\beta E_\beta(\vec{\beta}) + \lambda_J ||J_{3D} - J(\vec{\beta},\vec{\theta}) + \vec{\gamma}||_2^2 \,, \quad (4)$$

where $\vec{\gamma} \in \mathbb{R}^3$ is the SMPL mesh global translation, $\lambda_\theta = 5, \lambda_\beta = 300$ and $\lambda_J = 1$ are weights, and $E_\beta(\vec{\beta})$ is the shape prior learnt from the SMPL shape training data [1]. Contrary to MuVS, we use 3D rather than 2D joints data in this last stage. Furthermore, the objective function of MuVS contains joint projection operations and its number of terms is a function of the number of views. In our case, the number of views comes up when triangulating 3D joints. However, the triangulation is solved for each joint independently, and through a singular value decomposition, which is simpler to solve than the minimization of the final MuVS energy function.

3. EXPERIMENTAL RESULTS AND DISCUSSION

We evaluate our approach on two widely used multi-view data sets: HumanEva-I [21] and Human3.6M [22, 23] (both contain ground-truth 3D joint locations). As others [12, 13], we use HumanEva-I to make design choices and validate our approach, whereas Human3.6M serves to gauge the solution's generalization. We measure the performance with the Mean Per Joint Position Error (MPJPE) metric. No Procrustes analysis is used unless stated. We manually set all cost function weights on the training data set of HumanEva.

Since HumanEva-I and Human3.6M joints differ from the SMPL joints (Fig. 2), we compute one shift vector (in the local bone coordinate) for each of the SMPL joints. Among the first 1000 frames of each video for Human3.6M, we took every 100th frame and computed the shift between the result of our optimization and the groundtruth. We then apply the mean of the shift vectors before computing the MPJPE for all of the other frames. For HumanEva-I we took every 20th frame among the first 300 frames. Shift vectors are only applied when computing the MPJPE for our approach since for the other methods (in Tables 1 and 2), the MPJPEs report the values found in the respective papers. For fairness of comparison, we also report our MPJPE without shift vectors.

We carry out a first validation on the HumanEva-I data set. Following common practice [12, 13], we report results for subjects S1, S2, and S3 on the "Walking" and "Box" actions of the validation set. We use all three views and the ground-truth camera parameters. Table 1 compares our approach with MuVS. The first row ("MuVS") refers to the MuVS optimiza-

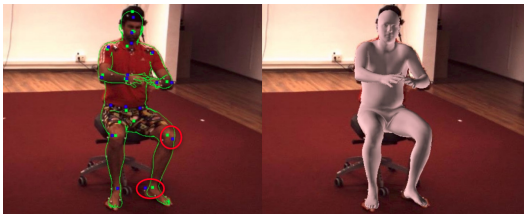

Fig. 2: Despite the fact that the SMPL mesh and its silhouette (green contour on the left image) match the individual on the image, there is a shift between the groundtruth Human3.6M joints (blue squares) and the SMPL joints (green squares).

Method	Walking	Box	Mean
MuVS	58.32	68.41	63.37
MuVS^{T}	56.68	67.79	62.23
Ours	48.59	61.45	55.02
Ours^{SV}	47.22	59.88	53.55
Ours^{BOC}	42.63	53.75	48.19
Ours$^{BOC,\ SV}$	**41.96**	**51.12**	**46.54**

Table 1: MPJPE (*mm*) comparison. Numerical results for MuVS were directly transcribed from the original paper [12].

Fig. 3: Qualitative results on Human3.6M's subjects 9 (left) and 11 (right). Body mesh silhouette in green and mesh in pink.

Method	Shape	PA	MV	MPJPE
Kanazawa *et al.* [14]	Yes	Yes	No	66.65
Trumble *et al.* [24]	No	No	Yes	62.50
Kolotouros *et al.* [11]	Yes	Yes	No	62.00
Pavlakos *et al.* [3]	No	No	Yes	56.89
MuVS^{T}	Yes	Yes	Yes	47.09
Ours	Yes	Yes	Yes	54.86
Ours^{SV}	Yes	Yes	Yes	39.56
Ours^{BOC}	Yes	Yes	Yes	46.37
Ours$^{BOC, SV}$	Yes	Yes	Yes	30.13
Iskakov *et al.* [9]	No	Yes	Yes	20.80
He *et al.* [10]	No	Yes	Yes	19.00

Table 2: Quantitative comparison on Human3.6M (subjects 9 and 11). "Shape" indicates if the method estimates the shape besides the pose. "PA" indicates if Procrustes analysis is applied before computing the MPJPE (*mm*). "MV" states if the method uses multiple views. Values for the compared methods were directly transcribed from the respective papers.

tion process using 2D joint error terms, silhouette consistency term, shape prior, and pose prior. The second row ("MuVS^{T}") corresponds to MuVS when adding temporal information as described by Huang *et al.* [12]. "Ours" refers to our approach without the BOC step in the optimization process. We notice that using 3D joints triangulated with OpenPose's confidence values significantly improves the MPJPE as compared to using 2D joints. Superscript SV means using the shift vectors when computing the MPJPE. We notice a slight improvement with the shift vectors. Finally, "Ours^{BOC}" illustrates the effectiveness of our BOC in further decreasing the error. Our approach outperforms MuVS even without taking advantage of temporal nor silhouette information.

We now look at the generalization of our approach with Human3.6M. The poses in this data set are more challenging than in HumanEva-I because of asymmetric and other complex poses. As others [3, 12, 24], we use subjects 9 and 11 for the evaluation. We use all four views and the ground-truth camera parameters. Table 2 compares our approach with other state-of-the-art methods. All are multi-view methods except Kanazawa *et al.* [14] and Kolotouros *et al.* [11]. We notice that almost all the multi-view methods perform better than the single-view ones, highlighting the fact that multiple views significantly improve the accuracy. Note that among the multi-view methods, only MuVS and our approach return a complete 3D human body mesh. The other methods optimize only for joint locations, which is a less constrained problem than simultaneously optimizing for shape and joint location. Unlike these methods, we compute the parameters for a full data-driven body shape, which incurs a trade-off between the accuracy of the pose and the body shape. Even so, our approach outperforms all methods that compute only joint locations except the methods of Iskakov *et al.* [9] and He *et al.* [10]. Finally, on Human3.6M our approach significantly outperforms the temporal version of MuVS, and the shift vectors significantly reduce the MPJPE.

Figure 3 shows some examples of the application of our approach to non-trivial poses from Human3.6M. Our approach is effective in recovering both shape and pose in these challenging situations with a very good correspondence between the projection of the 3D mesh and the images.

4. CONCLUSION AND FUTURE WORK

We have presented an approach to accurately estimate 3D human shape and pose in a multi-view setting. We have introduced several improvements in our proposed approach. First, we use 3D instead of 2D joints to infer the SMPL pose and shape parameters. We achieve this by triangulating 3D joints from 2D joints with a weighted algebraic triangulation. Second, we designed a new optimization process from the 3D joints to regress the SMPL parameters. This process incorporates a new bone orientation constraint (BOC) step which consists in solving a novel objective function to recover the SMPL pose parameter, independently from the shape. This results in a better initialization and leads to a better final mesh. Evaluation on benchmarks demonstrated the effectiveness of our approach compared to state-of-the-art methods.

5. REFERENCES

[1] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: a skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–16, 2015.

[2] M. Kocabas, S. Karagoz, and E. Akbas, "Self-Supervised Learning of 3D Human Pose Using Multi-View Geometry," in *IEEE CVPR*, 2019, pp. 1077–1086.

[3] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Harvesting Multiple Views for Marker-Less 3D Human Pose Annotations," in *IEEE CVPR*, 2017, pp. 1253–1262.

[4] H. Rhodin, F. Meyer, J. Sporri, E. Muller, V. Constantin, P. Fua, I. Katircioglu, and M. Salzmann, "Learning Monocular 3D Human Pose Estimation from Multi-view Images," in *IEEE CVPR*, 2018, pp. 8437–8446.

[5] Z. Cao, G. H. Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE TPAMI*, pp. 1–1, 2019.

[6] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A Simple Yet Effective Baseline for 3d Human Pose Estimation," in *IEEE ICCV*, 2017, pp. 2659–2668.

[7] H. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki, "Adversarial Inverse Graphics Networks: Learning 2D-to-3D Lifting and Image-to-Image Translation from Unpaired Supervision," in *IEEE ICCV*, 2017, pp. 4364–4372.

[8] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D Pictorial Structures for Multiple Human Pose Estimation," in *IEEE CVPR*, 2014, pp. 1669–1676.

[9] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable Triangulation of Human Pose," in *IEEE ICCV*, 2019, pp. 7717–7726.

[10] Y. He, R. Yan, K. Fragkiadaki, and S. Yu, "Epipolar transformers," in *IEEE CVPR*, 2020, pp. 7779–7788.

[11] N. Kolotouros, G. Pavlakos, M. Black, and K. Daniilidis, "Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop," in *IEEE ICCV*, 2019, pp. 2252–2261.

[12] Y. Huang, F. Bogo, C. Lassner, A. Kanazawa, P. V. Gehler, J. Romero, I. Akhter, and M. J. Black, "Towards Accurate Marker-Less Human Shape and Pose Estimation over Time," in *3DV*, 2017, pp. 421–430.

[13] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image," in *ECCV*. Springer, 2016, vol. 9909, pp. 561–578.

[14] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-End Recovery of Human Shape and Pose," in *IEEE CVPR*, 2018, pp. 7122–7131.

[15] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional Mesh Regression for Single-Image Human Shape Reconstruction," in *IEEE CVPR*, 2019.

[16] J. Liang and M. Lin, "Shape-Aware Human Pose and Shape Reconstruction Using Multi-View Images," in *IEEE ICCV*, 2019, pp. 4351–4361.

[17] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to Estimate 3D Human Pose and Shape from a Single Color Image," in *IEEE CVPR*, 2018, pp. 459–468.

[18] H. Rhodin, N. Robertini, D. Casas, C. Richardt, H. Seidel, and C. Theobalt, "General Automatic Human Shape and Motion Capture Using Volumetric Contour Cues," in *ECCV*. Springer, 2016, pp. 509–526.

[19] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural Body Fitting: Unifying Deep Learning and Model Based Human Pose and Shape Estimation," in *3DV*, 2018, pp. 484–494.

[20] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the People: Closing the Loop Between 3D and 2D Human Representations," in *IEEE CVPR*, 2017, pp. 4704–4713.

[21] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion," *Intl. Journ. Comp. Vision*, vol. 87, no. 1, p. 4, 2009.

[22] C. Ionescu, F. Li, and C. Sminchisescu, "Latent structured models for human pose estimation," in *IEEE ICCV*, 2011.

[23] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE TPAMI*, 2014.

[24] M. Trumble, A. Gilbert, A. Hilton, and J. Collomosse, "Deep Autoencoder for Combined Human Pose Estimation and Body Model Upscaling," in *ECCV*. Springer, 2018, vol. 11214, pp. 800–816.