Towards Accurate Markerless Human Shape and Pose Estimation over Time

Yinghao Huang  
Max Planck Institute for Intelligent Systems, Tübingen, Germany  
yinghao.huang@tue.mpg.de

Federica Bogo  
Microsoft Research  
febogo@microsoft.com

Christoph Classner  
Max Planck Institute for Intelligent Systems, Tübingen, Germany  
University of Tübingen  
classner@tue.mpg.de

Angjoo Kanazawa  
University of Maryland  
kanazawa@umiacs.umd.edu

Peter V. Gehler  
Max Planck Institute for Intelligent Systems, Tübingen, Germany  
University of Tübingen

Ijaz Akhter  
Australian National University  
ijaz.akhter@anu.edu.au

Michael J. Black  
Max Planck Institute for Intelligent Systems, Tübingen, Germany  
black@tue.mpg.de

Figure 1: Given multi-view videos, our method can not only yield more accurate 3D pose estimation results, but also more realistic and natural meshes than the other state-of-the-art. The entire process is fully automatic. From up to bottom: example input frames; meshes returned by [39]; meshes generated by our method.

Abstract

Existing marker-less motion capture methods often assume known backgrounds, static cameras, and sequence specific motion priors, which narrows its application scenarios. Here we propose a fully automatic method that
given multi-view video, estimates 3D human motion and body shape. We take recent SMPLify [12] as the base method, and extend it in several ways. First we fit the body to 2D features detected in multi-view images. Second, we use a CNN method to segment the person in each image and fit the 3D body model to the contours to further improves accuracy. Third we utilize a generic and robust DCT temporal prior to handle the left and right side swapping issue sometimes introduced by the 2D pose estimator. Validation on standard benchmarks shows our results are comparable to the state of the art and also provide a realistic 3D shape avatar. We also demonstrate accurate results on HumanEva and on challenging dance sequences from YouTube in monocular case.

1. Introduction

The markerless capture of human motion has been a long term goal of the community. While there have been many proposed approaches and even commercial ventures, existing methods only operate under restricted environments. Most commonly, such methods exploit background “subtraction” assuming a known and static background and the most accurate methods employ strong prior assumptions about the motion of the actor. In many cases the best results on benchmarks like HumanEva [42] are obtained by training on the same motion by the same actor as is evaluated at test time [5]. Here we provide a solution for markerless mocap that is more accurate than the recent state of the art but also less restrictive.

There are four key components to our approach. First our approach exploits SMPL [28], a realistic, low-dimensional, 3D parametric model of the human body. Second we use a convolutional neural network to compute putative 2D joint locations in multiple camera images. We then fit the 3D parametric model to the 2D joints robustly. This extends the SMPLify approach for pose and shape estimation [12] from one single image to multi-camera data. This is similar to recent work by Rhodin et al. [39], but we go beyond that work in several ways: our method is based on a “global” realistic human body model — SMPL [28], which naturally encodes the statistical dependency between different parts; we can not only achieve more accurate 3D joint estimation, but also better body mesh, which facilitates future modification and animation; what’s more, our method can be easily applied on monocular videos, this is not impossible for the method proposed in [39].

Third, we go beyond SMPLify [12] and other previous work to use a deep convolutional neural network to also segment people from images [27]. This removes the need for a background image and makes the approach more general. We fit our 3D body model to both the 2D joints and the estimated silhouettes and show that the silhouettes provide significantly improved accuracy and realism to the mocap.

Since 2D joints estimated by CNNs (Convolutional Neural Networks) sometimes confuse left and right parts of the body, the image evidence alone is not enough for a reliable solution. Consequently we exploit temporal information to resolve such errors. This leads to the fourth component in which we exploit a generic temporal prior based on the insight that human motions can be captured by a low-dimensional DCT (Discrete Cosine Transform) basis [4]. We implement this DCT temporal term robustly and show that it improves performance yet does not require training data.

We call the method MuVS (Multi-View SMPLify) and evaluate it quantitatively on HumanEva [42] and Human3.6M [24]. We find that MuVS gives the lowest error of any published result and more realistic meshes (see Figure 1), while having fewer restrictions. We evaluate the method with an ablation study to determine which design decisions are most important.

In particular, our approach also works in the monocular camera setting. We find that the temporal coherence term enables good quality reconstruction of pose from monocular video even with a moving camera, complex background, and challenging motions. We evaluate this quantitatively on HumanEva [42] and using challenging dance video sequences from Youtube. The software will be made available for research purposes.

2. Related Work

The majority of previous works only handle one aspect of the two closely related problems: 3D body shape an pose estimation. Some of them target at 3D pose estimation [5, 20, 17, 38, 43, 18]. They formulate it as a discriminative problem, directly inferring 3D pose from 2D image features, assuming no explicit human body model. Amin et al. [5] extend single-view based pictorial structure to multi-view cases, jointly inferring 2D joint location of all views, then use linear-triangulation to obtain the 3D joints. Yao et al. [52] propose a stochastic gradient based method for a Gaussian Process Latent Variable Model (GPLVM), which shows good optimization properties. Uncertainty over estimated pose 2D image features has also been considered. Zhou et al. [54] introduce sparsity prior over human pose, and jointly handle the pose and 2D location uncertainty, while Kazemi et al. [26] try to address the body part correspondence problem via optimizing latent variables. Similar idea is proposed by Simo-Serra et al. [44], which also estimate 2D and 3D pose at the same time. Twin Gaussian process [11] has also been used on this problem. Most recently deep learning methods achieve the most accurate pose estimation results [47, 18, 48, 30, 37]. To address the data-hungry issue, Yasin et al. [53] propose a dual-source approach. Pavlakos et al. [52] directly regress 3D pose from
RGB image via CNNs in a coarse-to-fine manner.

The second major set of approaches use an explicit intermediate human body representation, which effectively assists pose estimation but often lacks realism [16, 43, 10, 56]. Common human body representations include Articulated Human Body Model [17], 3D Pictorial Structures [10], Sum-of-Gaussian model [46], Triangulated Mesh Model [41], etc. These models are usually utilized to represent the structure of human body, thus facilitating the inference of pose parameters. Sometimes the body mesh is also considered, but in an abstract or coarse way, without consideration of the shape details. Estimating both the pose and surface mesh, usually requires complex global optimization [21, 20]. Often the silhouette of the body is assumed to be known [17] and manual initialization or a pre-scanned surface mesh is required [8, 15, 51, 49, 45, 9, 22, 25, 30, 24]. Balan et al. [8] address this problem by fitting a SCAPE body model [6] to multi-view silhouettes. Their initialization method was complex and they did not integrate information over time. Another new work concurrent with us is proposed in [33]. They also use CNNs to detect 2D joints, then fit a 3D pictorial structures model to the detections. But their method only returns 3D joints as the output, while ours estimate body shape and pose together. Compared with their approach, ours not only achieves comparable 3D joint estimation accuracy, but also yields realistic 3D mesh ready for animation.

The most similar recent work addresses fully automatic estimation of 3D pose and shape from monocular images [12] and multi-views videos [39]. The SMPLify algorithm proposed by Bogo et al. [12] makes it possible to simultaneously obtain accurate 3D pose and convincing shape from one single image, without requiring any user intervention, assuming background extraction or complex optimization techniques. Based on the state-of-the-art 3D human body model, SMPL [28], they infer human shape and pose parameters by fitting a projection of 3D SMPL joints to 2D joints estimated via a 2D joint detector like Deepcut or CPM [50, 34]. Ambiguity issues are handled by applying priors learned from the large-scale public CMU dataset [2], which is vital for their method to yield valid results. Rhodin et al. [39] propose a method that works on multi-view videos. Built upon a specially-tailored sum-of-Gaussian shape model [40, 46], their algorithm firstly initializes the pose of each Gaussian blob, then refines the pose and shape of each blob via the body contour approximated with image gradients. As in Bogo et al [12], they also use deep learning to estimate 2D joints to get rough joint projections on each view. Also they enforce temporal coherence by penalizing acceleration between frames. Note that their method does not generalize to monocular video sequences. To some extent this is due to the fact that they are using a “local” body model, which represents each human body part separately, without encoding the statistical dependency among these parts. Also their blob-based human body model is not as realistic as SMPL.

We go beyond SMPLify by extending it to multi-view and monocular videos in a straightforward and principled way. Though 2D joints encode a great amount of information about the underlying 3D pose and shape, we show that there are important addition cues, like the silhouette and temporal coherence. Also, in contrast to [39], we use explicit segmentation to obtain the body contour, and choose a Discrete-Cosine-Transformation (DCT) basis as the underlying temporal prior model. As a general temporal smoothness model, DCT can be applied in any video sequence, without the need of learning from a training dataset. Though conceptually similar, our algorithm turns out to be more accurate, and achieves much lower joint estimation error on public benchmarks. Additionally, compared with their blob based shape model, our estimated shape is more realistic, which favors later possible applications like body shape modification, pose editing, motion retargeting, etc.

3. 2D Joints and Segmentation

Our method takes as input a set of the major 2D body joints and a segmentation of the body from the background. Both of these are fully automatic and computed by CNNs [34, 50] trained on generic databases, which do not overlap with any of our test data. Illustrative joint estimation and human body segmentation results are shown in Figure 2.

As demonstrated in SMPLify [12], 2D joints alone encode many clues about the configuration of the corresponding 3D shape and pose. There a 3D generative model is fit such that the 3D joints of the model project to detected 2D joints in the image. For a direct quantitative comparison with SMPLify on standard test datasets, we use the same CNN-based joint estimator, DeepCut [39]. For more complex videos from the Internet, we use realtime convolution pose machines (CPM) [13] because we find that it is more reliable than DeepCut.

SMPLify obtains only approximate 3D shape given the 2D joints. More information is available in the silhouette of the body, particularly in multi-view imagery or video sequences where multiple silhouettes can constrain the shape. However, much of the previous work assumes known background segmentation or a static camera and background [43, 8, 20, 41, 14], which limits applications to real world videos. Consequently we use a CNN trained to estimate human segmentations [27]. In [27] they combine joints with segmentation to estimate 3D pose and shape but do not consider multi-view data or video sequences.
4. Multi-view SMPLify

Here we first extend SMPLify to multiple camera views, then extend it over time. Given the 2D joints and silhouettes for all the input frames for each camera view, we estimate the 3D pose for each time instant. Then we combine information from all the views to estimate a consistent 3D human shape over time. Consequently, our algorithm is composed of two consecutive stages described in detail below.

In the first stage, a separate SMPL model is fit to all views independently at each time instant. The extension of the public SMPLify code to multiple views is straightforward; we estimate a single shape and pose using information from all camera views. This gives a fully automatic approach to multi-camera markerless motion capture. In contrast, with as few as even 2 views, many of these ambiguities go away. After that, the silhouette is used to refine the estimated shape, which is then more faithful to the observed body.

In the second stage, we first estimate a consistent 3D shape across the entire sequence. We then treat a set of consecutive frames together and regularize the motion in time. We do this by fitting the projected joint error while requiring that the trajectory of each 3D joint can be well represented by a low-dimensional discrete cosine transform (DCT) [4]. This temporal smoothing helps remove errors caused by inaccurate 2D joint estimates, which are not perfect. In particular CNNs sometimes detect spurious points or are suffer left/right ambiguity.

4.1. Stage One

As in SMPLify, we use SMPL as our underlying shape representation. SMPL is a state-of-the-art statistical human body model [35, 36], which is controlled by two sets of parameters: one for the shape, the other for the pose. More formally, SMPL is defined as:

\[
M(\beta, \theta; \Phi) = W(T_p(\beta, \theta; T, s, p), J(\beta, J, s, t, \theta, W)),
\]

where \( \beta \) is the shape parameter, which is responsible for the body variation due to identify, and \( \theta \) is the pose parameter, which sets the pose configuration. And \( \Phi \) is the fixed parameters learnt from large quantity of 3D body meshes. For the detailed meaning of all these parameters, we refer the reader to [28] for further reference.

We first estimate the shape and pose parameters of SMPL model for each time instant. Given the corresponding 2D joint estimation \( \{J^1_{est}, J^2_{est}, \ldots, J^{\|V\|}_{est}\} \) for the different views \( V \), we formulate the energy function as the following:

\[
E_M(\beta, \theta) = E_P(\beta, \theta) + \sum_{v \in V} E_J(\beta, \theta; K_v, J^v_{est}),
\]

where \( E_p \) is the prior term, \( K_v \) are the camera parameters for view \( v \), and \( E_J \) is the joint fitting term. Note that here we remove the other priors used in SMPLify, because in multi-view cases the solution is better constrained. \( E_p \) is composed of two terms: a shape prior \( E_\beta \) and a pose prior \( E_\theta \). Both priors are learnt from the CMU dataset.

\[
E_P(\beta, \theta) = \lambda_\beta E_\beta(\theta) + \lambda_\theta E_\theta(\beta),
\]

The joint fitting term is formulated as follows:

\[
E_J(\beta, \theta; K_v, J^v_{est}) = \sum_{\text{joint } i} w_i \rho(R_{Ji}(J\beta_i - J^v_{est,i})),
\]

where \( J \) is the joint estimation function, which returns joint locations, and \( R \) is the rotation function, \( \Pi \) the projection function. Considering the inevitable detection noise even in the entire process, instead of standard square norm we adopt robust German-McClure statistics here, which is defined by:

\[
\rho(e, \sigma) = \frac{e^2}{\sigma^2 + e^2},
\]

here \( e \) is the residual error, and \( \sigma \) is the robustness parameter.

After obtaining the initial pose and shape estimation via fitting SMPL to 2D joints, we further refine them by adding silhouette information. The fitting error between rendered SMPL model and segmented contour is defined as:

\[
E_S(\beta, \theta; K_v, U_v) = \sum_{x \in S(\theta, \beta)} l(x, U_v)^2 + \sum_{x \in U_v} l(x, S(\theta, \beta)),
\]

where \( l(x, S) \) denotes the absolute distance from a point \( x \) to a silhouette \( S \); the distance is zero when the point is inside \( S \). The first term computes the distance from points
Figure 3: DCT based temporal prior helps to alleviate the leg swap problem. a): Pose detection with leg swap; b): MuVS without DCT prior; c): MuVS with DCT prior.

\[
\hat{S}(\theta, \beta) \quad \text{to the estimated silhouette } \quad \hat{S}^v(\theta, \beta) \quad \text{to the estimated silhouette } \quad \hat{S}(\theta, \beta). \quad \text{We find that the second term is noisier and we use the L1 distance, while the first term uses the L2 distance. Combined with the 2D joint fitting term, the final energy function is:}
\]

\[
E_1(\beta, \theta) = E_M(\beta, \theta) + \sum_{v \in V} E_S(\beta, \theta; K_v, U_v), \quad (7)
\]

We empirically found the solution can faster converge to better solution via the aforementioned hierarchical optimization strategy: firstly fitting SMPL to 2D joints can yield a coarse estimation of pose and shape parameter fast, then adding silhouette fitting term can boost the solution further.

4.2. Stage Two

One obvious shortcoming of the algorithm used in the first stage is that it doesn’t take into account the temporal relationship between consecutive frames, while in real life human motions usually present a strong consistent feature. What’s more, due to the lack of texture, occlusion, similarity to the background and other noise, the joint estimator be erroneous in ambiguous cases. One of these errors is leg swap, which is demonstrated in Figure 3. Sometimes these errors can be difficult to automatically correct in single frame. But treating some consecutive frames as a whole, and considering them at the same time, we can greatly alleviate the error.

To make our algorithm more time-efficient, in this stage, we don’t consider silhouette any more, and only use 2D joints. Using the obtained shape and pose parameters from the first stage, we try to optimize the following objective, which is composed of the 2D joint fitting term and low-dimensional DCT reconstruction term:

\[
E_2(\Theta, C; \beta, N) = \sum_{i=1}^{N} E_M(\beta, \Theta_i) + \sum_{\text{joint } i \in \{X, Y, Z\}} \lambda T E_T(C_i, \beta, \Theta, D_i, d) \quad (8)
\]

Here for the three different coordinate of each joint, we require that the trajectory can be well approximated by some low-dimensional DCT bases.

\[
E_T(c, \beta, d) = \sum_{i} \rho(d_i - Bc_i), \quad (9)
\]

where \( B \) is a matrix composed of the first several DCT components, \( c \) is the corresponding coefficients, \( i \) iterates over \( \{x, y, z\} \) three different coordinates, while \( d_i \) is the projected trajectory of the considered 3D joint on \( i \)-th coordinate.

4.3. Implementation Details

We implement our entire algorithm in pure Python language. The two involved optimization problems are conducted using Powell’s dogleg method [31], OpenDR [29] and Chumpy [1]. For the second stage, we choose 30 consecutive frames as a unit, and use the first 10 DCT components to act as the bases \( B \). All the weights are empirically chosen by running our method on the training dataset of HumanEva.

5. Evaluation

To evaluate the effectiveness of each stage of our method, we conducted extensive experiments on two commonly used datasets HumanEva-I [42] and Human3.6M [24], and compared with state-of-the-arts [39, 43, 10, 19, 5]. Both datasets are collected in controlled lab environment. And HumanEva-I is composed of 4 different subjects and 6 different motions, while Human3.6M is composed of capture performances 11 subjects each performing 15 different motions. To keep compatibility with SMPLify, we also use the first 10 shape parameters in all the experiments, and fine tune all the parameters on the training dataset of HumanEva-I.

5.1. Ablation study

To analyze the effect of different part of our algorithm, firstly we performed various ablation experiments on HumanEva. The results are shown in Table 1.
Table 1: Ablation results on HumanEva-I. 3D joint errors in mm. Here label 2/3 mean using the first 2/3 camera views; S means being with silhouette fitting term; T means being with temporal fitting term. The same notation is used in the rest paper.

| Method   | S1    | S2    | S3    | S1    | S2    | S3    | Mean | Median |
|----------|-------|-------|-------|-------|-------|-------|------|--------|
| MuVS^2   | 59.22 | 66.81 | 88.60 | 79.51 | 78.68 | 88.34 | 76.86 | 79.10  |
| MuVS^2,S | 54.35 | 56.06 | 80.95 | 70.27 | 72.01 | 79.01 | 68.78 | 71.14  |
| MuVS^2,S,T| 50.14 | 56.11 | 79.55 | 68.96 | 71.73 | 78.45 | 67.49 | 70.35  |
| MuVS^3   | 52.50 | 62.76 | 82.51 | 72.86 | 73.10 | 80.42 | 70.69 | 72.98  |
| MuVS^3,S | 47.21 | 52.72 | 75.04 | 64.88 | 68.39 | 71.98 | 63.37 | 66.64  |
| MuVS^3,S,T| 43.11 | 53.37 | 73.56 | 64.00 | 67.94 | 71.44 | 62.23 | 65.97  |

Table 2: Shape estimation error on HumanEva. Error in mm.

| Method   | Walk | Box  | Avg |
|----------|------|------|-----|
|          | S1   | S2   | S3  |
| MuVS^2   | 18.9 | 19.4 | 20.7|
| MuVS^2,S | 14.6 | 9.6  | 16.6|
| MuVS^2,S,T| 14.1 | 9.3  | 15.9|
| MuVS^3   | 17.6 | 18.6 | 20.6|
| MuVS^3,S | 13.5 | 9.1  | 16.0|
| MuVS^3,S,T| 13.1 | 8.6  | 15.3|

3D body reconstruction, it’s important to estimate the human shape accurately.

**Effect of DCT based temporal prior** As evidenced in the result, adding DCT temporal steadily boost overall performance. As expected its effect diminishes when more views are added, since in this case quite good results can be obtained in the first stage.

5.2. Quantitative comparison

**HumanEva**: We follow the standard practice of evaluating on the “Walking” and “Boxing” sequences of subjects 1, 2 and 3. The same with SMPLify [12], gender of the subject is assumed known and gender-specific model is applied on each motion sequence. The result is shown in Table 2. Here General means the method is trained on the training dataset of HumanEva, instead of separately training model for each specific subject, which is referred to Specific. For the General case, we use the joint regressor shipped with SMPL to get pose prediction, and directly compare the result with ground-truth. For the Specific case, we use the joint regressor trained on HumanEva with MoSh, which is provided in SMPLify [12]. Then the same with Rhodin et al. [39], we compute the displacement between the estimated joint location and ground-truth, then compensate them in
the remaining frames.

As we can see, in the General case, only with 2 views our method is more accurate than all the other methods using all 3 views. And with 3 views we obtain a significant margin from the second best method (55.52 vs 63.25). Our method also achieves the lowest error in the Specific case. And another advantage of our method over the state-of-the-art is we return a highly realistic body mesh together joint skeleton, which facilitates future operation. Though the method proposed by Rhodin et al. [39] also yields blob based 3D mesh, we argue that the underlying SMPL model we use is more realistic. Furthermore, the mesh variation with pose is inherently considered in SMPL model, which is impossible for that of Rhodin et al. [39]. Some qualitative comparison between the meshes are shown in Figure 1. For more please refer to our supplementary materials.

Human3.6M: To further validate the generality and usefulness of MuVS, we also evaluate it on Human3.6M [24]. Human3.6M is the largest public dataset for pose estimation, composed of a wide range of motion types, some of them being very challenging. We use the same parameters trained on HumanEva, then apply MuVS on all the 4 views of subjects S9 and S11. We compare it with SMPLify [12] and other state-of-the-art multi-view based pose estimation methods [33]. The result is shown in Table 4. As we can see, our 3D joint estimation accuracy is quite close to the method proposed in [33], which is concurrent with our work. However they only focus on the 3D joint estimation, while we address the 3D pose and shape estimation at the same time. Our method not only returns accurate 3D joint estimation, but also promising body shape faithful to the subjects. Altogether a realistic and accurate 3D body mesh is obtained, which is ready for later modification and animation. This is not easily achievable for the method proposed in [33].

6. Pose and Shape from Monocular Video

Though we focus on multi-view pose and shape estimation in this work, our method can be easily applied on monocular video sequences without the need of large modification, and still being fully automatic. Note manually initialized pose is required for the method proposed in [39] to work on monocular data.

We compare our method with SMPLify on the first camera view of HumanEva, and the result is shown in Table 5. Of course given only one single video, it’s very hard to apply DCT constraint on depth, since we don’t have any trustable evidence in that dimension. But we empirically find our method can still return quite promising result when the performer doesn’t move so much in depth. We qualitatively evaluate our method on some videos downloaded from Youtube, and show the results for specific instant in Figure 5. And the mesh sequences of one of them is shown in Figure 6. For the full video, please refer to our supplementary materials.

7. Conclusion and Future Work

In this paper we present a new marker-less motion capture system — MuVS, which extends SMPLify in a principled and straightforward way. Our method can not only accurately obtain accurate 3D pose, but also return a realistic and faithful human body mesh as a by-product. Differently from previous work which assumes known silhouette, needs extensive user intervention or applied some requirement about the motion, our algorithm works for the general activities seen in daily life. Extensive evaluation on public benchmark validates the effectiveness and generality of our method. What’s more, MuVS can naturally be applied on monocular video sequences, and achieves promising result.

In the future we would like to further improve our algorithm to handle more complex scenarios, like cluttered background, multiple people, and extreme poses. We will also try to speed up our algorithm to make it more applicable for real-world applications. Other body parts, like face, hand and foot can also be easily combined into our model.

References

[1] Chumpy. http://chumpy.org.
[2] Cmu mocap dataset. http://mocap.cs.cmu.edu.
[3] N. Ahmed, E. De Aguiar, C. Theobalt, M. Magnor, and H.-P. Seidel. Automatic generation of personalized human avatars from multi-view video. In Proceedings of the ACM symposium on Virtual reality software and technology, pages 257–260. ACM, 2005.
[4] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh. Bilinear spatiotemporal basis models. ACM Transactions on Graphics (TOG), 31(2):17, Apr. 2012.
[5] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. Multi-view pictorial structures for 3d human pose estimation. In BMVC. Citeseer, 2013.
[6] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In ACM Transactions on Graphics (TOG), volume 24, pages 408–416. ACM, 2005.
[7] A. O. Bălan and M. J. Black. The naked truth: Estimating body shape under clothing. In European Conference on Computer Vision, pages 15–29. Springer, 2008.
[8] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007.
[9] L. Ballan and G. M. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. 3DPVT, Atlanta, GA, USA, 37, 2008.
[10] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures for multiple
| Method                  | Trained on | Walking | Boxing | Mean | Mean (all) |
|------------------------|------------|---------|--------|------|------------|
|                         |            | S1  | S2  | S3  | S1  | S2  | S3  |
| Rhodin et al. [39]      |            | 74.9 | 59.7 | 67.3 | 59.7 | 67.3 |
| Sigal et al. [43]       |            | 66.0 | 66.0 | 66.0 | 66.0 | 66.0 |
| Belagiannis et al. [10] | General    | 68.3 | 62.7 | 65.5 | 65.5 | 65.5 |
| Elhayek et al. [19]     |            | 66.5 | 60.0 | 63.25| 63.25| 63.25|
| MuVS², S, T             |            | 50.14| 56.11| 79.55| 68.96| 71.73| 78.45|
| MuVS³, S, T             |            | 43.11| 53.37| 73.56| 64.00| 67.94| 71.44| 55.52| 62.23|
| Amin et al. [5]         | Specific   | 54.5 | 47.7 | 51.10| 47.7 | 51.10|
| Rhodin et al. [39]      |            | 54.6 | 35.1 | 44.85| 35.1 | 44.85|
| MuVS³, S, T             |            | 33.72| 36.78| 60.11| 46.85| 49.92| 46.99| 41.82| 45.73|

Table 3: Quantitative comparison on HumanEva-I. 3D joint errors in mm.

| Directions | Discussion | Eating | Greeting | Phoning | Photo | Posing | Purchases | Sit |
|------------|------------|--------|----------|---------|-------|--------|-----------|-----|
| SMPLify [12] |            | 62.0   | 60.2     | 67.8    | 76.5  | 92.1   | 77.0      | 73.0 |
| MuVS², S, T, Sim |      | 35.05  | 39.22    | 38.59   | 37.35 | 59.16  | 46.07     | 40.52| 38.47| 60.07|
| Tekin et al. [47] |        | 102.41 | 147.72   | 88.83   | 125.28| 118.02 | 182.73    | 112.38| 129.17| 138.89|
| MuVS², S, T    |            | 44.32  | 46.99    | 51.75   | 44.99 | 67.68  | 54.56     | 49.25| 48.90| 72.82|
| Pavlakos et al. [33] |    | 41.18  | 49.19    | 42.79   | 43.44 | 55.62  | 46.91     | 40.33| 63.68| 97.56|
| SitDown       |            | 137.3  | 83.4     | 77.3    | 79.7  | 86.8   | 81.7      | 82.3  | 69.3 |
| Smoking       |            | 66.970 | 56.24    | 67.91   | 46.91 | 38.00  | 33.15     | 47.09 | 40.52|
| Waiting       |            | 224.9  | 118.42   | 138.75  | 126.29| 55.07  | 65.76     | 124.97| 125.28|
| WalkDog       |            | 76.51  | 63.70    | 116.24  | 55.44 | 42.94  | 37.24     | 58.22 | 51.75|
| Walk          |            | 119.90 | 52.12    | 42.68   | 51.93 | 41.79  | 39.37     | 56.89 | 46.91|
| WalkTogether  |            |        |         |        |       |       |           |      |     |
| Mean          |            |        |         |        |       |       |           | 69.3  |     |
| Median        |            |        |         |        |       |       |           | 69.3  |     |

Table 4: Qualitative comparison with SMPLify, the methods of Tekin et al. [47] and Pavlakos et al. [33] on H3.6M dataset in mm. The accuracy of our method is comparable with that of the recent method proposed in [47].

| Method          | Walk | Box | Avg |
|-----------------|------|-----|-----|
| SMPLify [12]    | 73.3 | 59.9| 94.4| 82.1| 79.2 | 87.2| 79.9 |
| MuVS³, S, T, Sim | 51.2 | 48.1| 81.6| 61.5| 78.3 | 82.6| 67.2 |

Table 5: Comparison with SMPLify on monocular videos from HumanEva in mm. Here Sim means using Procrustes analysis per frame, as with SMPLify.

References:
[10] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1669–1676, 2014.
[11] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. International Journal of Computer Vision, 87(1):28–52, 2010.
[12] 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 European Conference on Computer Vision, pages 561–578. Springer, 2016.
[13] Y. Chen, T.-K. Kim, and R. Cipolla. Inferring 3d shapes and deformations from single views. In European Conference on Computer Vision, pages 300–313. Springer, 2010.
[14] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In ACM Transactions on Graphics (TOG), volume 27, page 98. ACM, 2008.
[15] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, volume 2, pages 126–133. IEEE, 2000.
[16] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. International Journal of Computer Vision, 61(2):185–205, 2005.
[17] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. Marker-less 3d human motion capture with monocular image sequence and height-maps. In European Conference on Computer Vision, pages 20–36. Springer, 2016.
scenes with a low number of cameras. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3810–3818. IEEE, 2015.

[21] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1746–1753. IEEE, 2009.

[22] N. Hasler, H. Ackermann, B. Rosenhahn, T. Thormählen, and H.-P. Seidel. Multilinear pose and body shape estimation of dressed subjects from image sets. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1823–1830. IEEE, 2010.

[23] S. Ilic and P. Fua. Implicit meshes for surface reconstruction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(2):328–333, 2006.

[24] 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 transactions on pattern analysis and machine intelligence, 36(7):1325–1339, 2014.

[25] A. Jain, T. Thormählen, H.-P. Seidel, and C. Theobalt. Moviereshape: Tracking and reshaping of humans in videos. In ACM Transactions on Graphics (TOG), volume 29, page 148, 2010.

[26] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan. Multiview body part recognition with random forests. In 2013 24th British Machine Vision Conference, BMVC 2013: Bristol; United Kingdom; 9 September 2013 through 13 September 2013. British Machine Vision Association, 2013.
