A Deeper Look into DeepCap

Marc Habermann, MPII, Weipeng Xu, MPII, Michael Zollhoefer, Stanford University, Gerard Pons-Moll, MPII, and Christian Theobalt, MPII

(Invited Paper)

Abstract—Human performance capture is a highly important computer vision problem with many applications in movie production and virtual/augmented reality. Many previous performance capture approaches either required expensive multi-view setups or did not recover dense space-time coherent geometry with frame-to-frame correspondences. We propose a novel deep learning approach for monocular dense human performance capture. Our method is trained in a weakly supervised manner based on multi-view supervision completely removing the need for training data with 3D ground truth annotations. The network architecture is based on two separate networks that disentangle the task into a pose estimation and a non-rigid surface deformation step. Extensive qualitative and quantitative evaluations show that our approach outperforms the state of the art in terms of quality and robustness. This work is an extended version of [1] where we provide more detailed explanations, comparisons and results as well as applications.

Index Terms—Monocular human performance capture, 3D pose estimation, non-rigid surface deformation, human body.

1 INTRODUCTION

Capturing the space-time coherent geometry of entire humans including also their everyday clothing, which is also known as human performance capture, is extensively used nowadays for movie production and game development. With these tools, real humans can be ported into a virtual world and fused into the process of creating virtual content. But also other applications such as virtual try-on, telepresence and virtual and augmented reality applications can strongly benefit from dense capture methods. Especially the latter mentioned applications are mainly used by non-expert users, who do not own dense multi-capture setups. Thus, to democratize these technologies the hardware setup has be as simple and accessible as possible, which means ideally a single color camera is sufficient to capture the entire deforming surface of a human. While there is extensive research regarding monocular skeleton-only tracking, only few works have targeted the capture of the dense deforming geometry of the entire human from a single color stream. Nonetheless, densely capturing entire humans from a single view is still far from being solved while it is indispensable for creating higher fidelity and photo-real characters.

Some works [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] have focused on capturing entire humans from multi-camera setups that typically also include a green screen. While they achieve a superior quality, the hardware requirement makes it almost impossible to use them in other locations like an outdoor film set and restricts the usage to companies that can afford such an expensive setup.

To overcome the shortcomings of multi-view approaches and with the advent of deep learning, some recent approaches [15], [16], [17], [18], [19], [20], [21] focus on predicting the 3D clothed surface of a human from a single color image. In particular, some works leverage implicit surface representations such as 3D voxel grids [16], [22] or pixel aligned implicit surface functions [15], [23]. While such representations can handle topological surface changes as well as the capture of finer geometric details, they suffer from artifacts like missing limbs as the regressed deformations are not explicitly constrained by a discrete surface mesh. Further, these methods are designed to reconstruct a surface per image. Thus, naively applying those techniques to video streams results in individual geometries per frame which are not coherent over time. To this end, another line of work [17], [19], [24], [25] regresses surface deformation with respect to a naked human body model. Here, coherence over time can be ensured and those works do not suffer from missing limbs since the underlying body model constrains the surface deformation. However, one drawback of these approaches is that they do not capture the motion and surface deformations over time.

The state-of-the-art human performance capture methods MonoPerfCap [26] and LiveCap [27] densely track the deforming surface over time. In contrast to the proposed method, they only predict sparse 2D/3D skeletal keypoints from the images and then perform an expensive optimization based pose and surface fitting. By design, their method
suffers from the monocular setting as the deformations can only be constrained by the single view and their pose results suffer from learning forward artifacts which originates from their biased keypoint detections. In contrast, we propose the first learning-based approach that jointly regresses the skeletal pose as well as the non-rigid surface deformation within a single inference pass resulting in a higher accuracy in terms of 3D surface and pose tracking as well as an improved robustness. Specifically, two CNN-based models predict the skeletal joint angles and embedded deformation parameters of a differentiable mesh-based character representation from a single image. By using an explicit mesh representation, our method has the advantage that the surface can be tracked over time which is key-essential for texturing and rendering in graphics. Furthermore, the coarse-to-fine modeling of articulation and surface deformation ensures that the output of our method does not suffer from missing limbs, even when those are occluded or when the actor performs out-of-plane motions.

In contrast to previous work [15], [16], [17], [19], our method does not require any form of 3D supervision for training but instead leverages weak supervision in the form of multi-view videos which can be potentially also sparse. To achieve this, we propose differentiable 3D-to-2D modules which allow us to train our deep architectures in an analysis-by-synthesis manner without using 3D ground truth. At the core, our method leverages a person-specific 3D template of the actor as well as a multi-view video showing the actor while he performs a wide range of motions. Then, our dedicated network modules predict the pose and surface deformation parameters that allow to pose and deform the template. This deformed and posed template is then compared against sparse and dense multi-view observations extracted from the multi-view images in a differentiable manner. Importantly, at test time our method only takes a single image as input and predicts the posed and non-rigidly deformed surface of the entire human including also the clothing. In summary, the main technical contributions of our work are:

- A learning-based 3D human performance capture approach that jointly tracks the skeletal pose and the non-rigid surface deformations from monocular images.
- A new differentiable representation of deforming human surfaces which enables training from multi-view video footage directly.

Our new model achieves high quality dense human performance capture results on our new challenging dataset, demonstrating, qualitatively and quantitatively, the advantages of our approach over previous work. We experimentally show that our method produces reconstructions of higher accuracy and 3D stability, in particular in depth, than related work, also under difficult poses.

This work is an extended version of DeepCap [1] where additional explanations, evaluations, comparisons, applications, and limitations are provided. In particular, the character modeling and processing as well as the specific non-trivial training strategies are explained in more detail. Further, we provide additional evaluations on all 4 subjects of the dataset of [1] and qualitative ablation results. Last, we showcase potential applications and discuss limitations.

2 RELATED WORK

In the following, we focus on related work in the field of dense 3D human performance capture and do not review work on sparse 2D pose estimation.

Capture using Parametric Models. Monocular human performance capture is an ill-posed problem due to its high dimensionality and ambiguity. Low-dimensional parametric models can be employed as shape and deformation prior. First, model-based approaches leverage a set of simple geometric primitives [28], [29], [30], [31]. Recent methods employ detailed statistical models learned from thousands of high-quality 3D scans [32], [33], [34], [35], [36], [37], [38], [39], [40], [41]. Deep learning is widely used to obtain 2D and/or 3D joint detections or 3D vertex positions that can be used to inform model fitting [42], [43], [44], [45], [46]. An alternative is to regress model parameters directly [47], [48], [49]. Beyond body shape and pose, recent models also include facial expressions and hand motion [50], [51], [52], [53] leading to very expressive reconstruction results. Recently, Zhou et al. [54] also predict facial albedo and lighting parameters while even achieving real-time performance. Since parametric body models do not represent garments, variation in clothing cannot be reconstructed, and therefore many methods recover the naked body shape under clothing [55], [56], [57], [58]. The full geometry of the actor can be reconstructed by non-rigidly deforming the base parametric model to better fit the observations [18], [59], [60], [61]. But they can only model tight clothes such as T-shirts and pants, but not loose apparel which has a different topology than the body model, such as skirts. To overcome this problem, ClothCap [14] captures the body and clothing separately, but requires active multi-view setups. Physics-based simulations have recently been leveraged to constrain the surface tracking [62], [63] as well as the pose estimation [64], [65], or to learn a model of clothing on top of SMPL (TailorNet [20]). Instead, our method is based on person-specific templates including clothes and employs deep learning to predict clothing deformation based on monocular video directly.

Depth-based Template-free Capture. Most approaches based on parametric models ignore clothing. The other side of the spectrum are prior-free approaches based on one or multiple depth sensors. Capturing general non-rigidly deforming scenes [66], [67], even at real-time frame rates [67], [68], [69], is feasible, but only works reliably for small, controlled, and slow motions. Higher robustness can be achieved by using higher frame rate sensors [70], [71] or multi-view setups [72], [73], [74], [75], [76]. Techniques that are specifically tailored to humans increase robustness [77], [78], [79] by integrating a skeletal motion prior [72] or a parametric model [78], [80]. HybridFusion [81] additionally incorporates a sparse set of inertial measurement units. These fusion-style volumetric capture techniques [82], [83], [84], [85], [86] achieve impressive results, but do not establish a set of dense correspondences between all frames. In addition, such depth-based methods do not directly generalize to our monocular setting, have a high power consumption, and typically do not work well under sunlight.
Monocular Template-free Capture. Quite recently, fueled by the progress in deep learning, many template-free monocular reconstruction approaches have been proposed. Due to their regular structure, many implicit reconstruction techniques \([16], [87]\) make use of uniform voxel grids. DeepHuman \([16]\) combines a coarse scale volumetric reconstruction with a refinement network to add high-frequency details. Multi-view CNNs can map 2D images to 3D volumetric fields enabling reconstruction of a clothed human body at arbitrary resolution \([88]\). SiCloPe \([89]\) reconstructs a complete textured 3D model, including cloth, from a single image. PIFu \([15]\), its follow-up work \([23]\), and a real-time variant \([90]\) regress an implicit surface representation that locally aligns pixels with the global context of the corresponding 3D object. Similar to our work, ARCH \([91]\) proposes to evaluate the implicit surface function in pose canonical space which makes the learning task easier as the pose and surface deformation can be separated. Unlike voxel-based representations, this implicit per-pixel representation is more memory efficient. These approaches have not been demonstrated to generalize well to strong articulation. Furthermore, implicit approaches do not recover frame-to-frame correspondences which are of paramount importance for downstream applications, e.g., in augmented reality and video editing. In contrast, our method is based on a mesh representation and can explicitly obtain the per-vertex correspondences over time while being slightly less general.

Template-based Capture. An interesting trade-off between being template-free and relying on parametric models are approaches that only employ a template mesh as prior. Historically, template-based human performance capture techniques exploit multi-view geometry to track the motion of a person \([92]\). Some systems also jointly reconstruct and obtain a foreground segmentation \([2], [8], [10], [93]\). Given a sufficient number of multi-view images as input, some approaches \([4], [9], [24]\) align a personalized template model to the observations using non-rigid registration. All the aforementioned methods require expensive multi-view setups and are not practical for consumer use. Depth-based techniques enable template tracking from less cameras \([72]\), \([5]\) and reduced motion models \([5], [10], [11]\) increase tracking robustness. Recently, capturing 3D dense human body deformation just with a single RGB camera has been enabled \([20]\) and real-time performance has been achieved \([27]\). However, their methods rely on expensive optimization leading either to very long per-frame computation times \([25]\) or the need for two graphics cards \([27]\). Similar to them, our approach also employs a person-specific template mesh. But differently, our method directly learns to predict the skeletal pose and the non-rigid surface deformations. As shown by our experimental results, benefiting from our multi-view based self-supervision, our reconstruction accuracy significantly outperforms the existing methods.

3 Method

Given a single RGB video of a moving human in general clothing, our goal is to capture the dense deforming surface of the full body. This is achieved by training a neural network consisting of two components: As illustrated in Fig. 3, our pose network, PoseNet, estimates the skeletal pose of the actor in the form of joint angles from a monocular image (Sec. 3.2). Next, our deformation network, DefNet, regresses the non-rigid deformation of the dense surface, which cannot be modeled by the skeletal motion, in the embedded deformation graph representation (Sec. 3.3). To avoid generating dense 3D ground truth annotation, our network is trained in a weakly supervised manner. To this end, we propose a fully differentiable human deformation and rendering model, which allows us to compare the rendering of the human body model to the 2D image evidence and back-propagate the losses. For training, we first capture a video sequence in a calibrated multi-camera green screen studio (Sec. 3.1). Note that our multi-view video is only used during training. More details about training are provided in Sec. 4. At test time we only require a single RGB video to perform dense non-rigid tracking.

3.1 Template and Data Acquisition

Character Model. Our method relies on a person-specific 3D template model. To this end, we first scan the actor with a 3D scanner \([92]\) to obtain the textured mesh (see Fig. 2). Next, we rig a skeleton onto the template which consists of 23 joints and 21 attached landmarks (17 body and 4 face landmarks) and it is parameterized with 27 joint angles \(\theta \in \mathbb{R}^{27}\), the camera relative rotation \(\alpha \in \mathbb{R}^{3}\) and translation \(t \in \mathbb{R}^{3}\). The landmark placement follows the convention of OpenPose \([97], [98], [99], [100]\). To model the non-rigid surface deformation, we automatically build an embedded deformation graph \(G\) with \(K\) nodes. The connections of a
node $k$ to neighboring nodes are denoted as the set $N_v(k)$. The position of the graph nodes is denoted as $G \in \mathbb{R}^{K \times 3}$ where $G_k$ is the position of node $k$. Finally, the vertex-to-node weights between the graph node $k$ and the template vertex $i$ are defined as $w_{i,k}$ and $N_{vn}(i)$ denotes the set of nodes that influence vertex $i$. The nodes are parameterized with Euler angles $A \in \mathbb{R}^{K \times 3}$ and translations $T \in \mathbb{R}^{K \times 3}$. Similar to [27], we segment the mesh into different non-rigidity classes resulting in per-vertex rigidity weights $s_i$. This allows us to model varying deformation behaviors of different surface materials, e.g. skin deforms less than clothing (see Eq. 13).

**Training Data.** To acquire the training data, we record a multi-view video using $C$ calibrated cameras of the actor doing various actions in a calibrated multi-camera studio with green screen. To provide weak supervision for the training, we first perform 2D pose detection on the sequences using OpenPose [97], [98], [99], [100] and apply temporal filtering. Then, we generate the foreground mask using color keying and compute the corresponding distance transform image $D_{f,c}$ [101], where $f \in [0,F]$ and $c \in [0,C]$ denote the frame index and camera index, respectively. During training, we randomly sample one camera view $c'$ and frame $f'$ for which we crop the recorded image with a bounding box, based on the 2D joint detections. The final training input image $I_{f',c'} \in \mathbb{R}^{256 \times 256 \times 3}$ is obtained by removing the background and augmenting the foreground with random brightness, hue, contrast and saturation changes. For simplicity, we describe the operation on frame $f'$ and omit the subscript $f'$ in following equations.

### 3.2 Pose Network

In our PoseNet, we use ResNet50 [102] pretrained on ImageNet [103] as backbone and modify the last fully connected layer to output a vector containing the joint angles $\theta$ and the camera relative root rotation $\alpha$, given the input image $I_{c'}$. Since generating the ground truth for $\theta$ and $\alpha$ is a non-trivial task, we propose weakly supervised training based on fitting the skeleton to multi-view 2D joint detections.

#### Kinematics Layer.

To this end, we introduce a kinematics layer as the differentiable function that takes the joint angles $\theta$ and the camera relative rotation $\alpha$ and computes the positions $P_{c'} \in \mathbb{R}^{M \times 3}$ of the $M$ 3D landmarks attached to the skeleton (17 body joints and 4 face landmarks). Note that $P_{c'}$ lives in a camera-root-relative coordinate system. In order to project the landmarks to other camera views, we need to transform $P_{c'}$ to the world coordinate system:

$$P_m = R_{c'}^T P_{c',m} + t,$$

where $R_{c'}$ is the rotation matrix of the input camera $c'$ and $t$ is the global translation of the skeleton.

#### Global Alignment Layer.

To obtain the global translation $t$, we propose a global alignment layer that is attached to the kinematics layer. It localizes our skeleton model in the world space, such that the globally rotated landmarks $R_{c'}^T P_{c',m}$ project onto the corresponding detections in all camera views. This is done by minimizing the distance between the rotated landmarks $R_{c'}^T P_{c',m}$ and the corresponding rays cast from the camera origin $o_c$ to the 2D joint detections:

$$\sum_c \sum_m \sigma_{c,m} \| R_{c'}^T P_{c',m} + t - o_c \times d_{c,m} \|^2,$$

where $d_{c,m}$ is the direction of a ray from camera $c$ to the 2D joint detection $p_{c,m}$ corresponding to landmark $m$:

$$d_{c,m} = \frac{(E_{c}^{-1} p_{c,m})_{xyz} - o_c}{\| (E_{c}^{-1} p_{c,m})_{xyz} - o_c \|}.$$

Here, $E_c \in \mathbb{R}^{4 \times 4}$ is the projection matrix of camera $c$ and $p_{c,m} = (p_{c,m}, 1)^T$. Each point-to-line distance is weighted by the joint detection confidence $\sigma_{c,m}$, which is set to zero if below 0.4. The minimization problem of Eq. 2 can be solved in closed form:

$$t = W^{-1} \sum_c \sum_m D_{c,m} (R_{c'}^T P_{c',m} - o_c) + o_c - R_{c'}^T P_{c',m},$$

where

$$W = \sum_c \sum_m I - D_{c,m}.$$
Here, \( \mathbf{I} \) is the \( 3 \times 3 \) identity matrix and \( \mathbf{D}_{c,m} = \mathbf{d}_{c,m} \mathbf{d}_{c,m}^T \). Note that the operation in Eq. 4 is differentiable with respect to the landmark position \( \mathbf{p}_{c,m} \).

**Sparse Keypoint Loss.** Our 2D sparse keypoint loss for the PoseNet can be expressed as

\[
\mathcal{L}_{\text{kp}}(\mathbf{P}) = \sum_{c} \sum_{m} \lambda_{m} \sigma_{c,m} \| \mathbf{p}_c(\mathbf{P}_m) - \mathbf{p}_{c,m} \|^2, \tag{6}
\]

which ensures that each landmark projects onto the corresponding 2D joint detections \( \mathbf{p}_{c,m} \) in all camera views. Here, \( \sigma_c \) is the projection function of camera \( c \) and \( \lambda_{m} \) is the same as in Eq. 2. \( \lambda_m \) is a hierarchical re-weighting factor that varies during training for better convergence. More precisely, for the first one third of the training iterations per training stage (see Sec. 4) for PoseNet, we multiply the keypoint loss with a factor of \( \lambda_{m} = 3 \) for all torso markers and with a factor of \( \lambda_{m} = 2 \) for elbow and knee markers. For all other markers, we set \( \lambda_{m} = 1 \). For the remaining iterations, we set \( \lambda_{m} = 3 \) for all markers. This re-weighting allows us to let the model first focus on the global rotation (by weighting torso markers higher than others). We found that this gives better convergence during training and joint angles overshoot less often, especially at the beginning of training.

**Pose Prior Loss.** To avoid unnatural poses, we impose a pose prior loss on the joint angles

\[
\mathcal{L}_{\text{limit}}(\theta) = \sum_{i=1}^{27} \Psi(\theta_i) \tag{7}
\]

\[
\Psi(x) = \begin{cases} 
(x - \theta_{\max,i})^2, & \text{if } x > \theta_{\max,i} \\
(\theta_{\min,i} - x)^2, & \text{if } x < \theta_{\min,i} \\
0, & \text{otherwise}
\end{cases}. \tag{8}
\]

that encourages that each joint angle \( \theta_i \) stays in a range \([\theta_{\min,i}, \theta_{\max,i}]\) depending on the anatomic constraints.

### 3.3 Deformation Network

With the skeletal pose from PoseNet alone, the non-rigid deformation of the skin and clothes cannot be fully explained. Therefore, we disentangle the non-rigid deformation and the articulated skeletal motion. DefNet takes the input image \( I_{c'} \) and regresses the non-rigid deformation parameterized with rotation angles \( \mathbf{A} \) and translation vectors \( \mathbf{T} \) of the nodes of the embedded deformation graph. DefNet uses the same backbone architecture as PoseNet, while the last fully connected layer outputs a \( 6K \)-dimensional vector reshaped to match the dimensions of \( \mathbf{A} \) and \( \mathbf{T} \). The weights of PoseNet are fixed while training DefNet. Again, we do not use direct supervision on \( \mathbf{A} \) and \( \mathbf{T} \). Instead, we propose a deformation layer with differentiable rendering and use multi-view silhouette-based weak supervision.

**Deformation Layer.** The deformation layer takes \( \mathbf{A} \) and \( \mathbf{T} \) from DefNet as input to non-rigidly deform the surface

\[
\mathbf{Y}_i = \sum_{k \in \mathcal{N}_{vn}(i)} w_{i,k}(R(\mathbf{A}_k)(\mathbf{V}_i - \mathbf{G}_k) + \mathbf{G}_k + \mathbf{T}_k). \tag{9}
\]

Here, \( \mathbf{Y}, \mathbf{V} \in \mathbb{R}^{N \times 3} \) are the vertex positions of the deformed and undeformed template mesh, respectively. \( w_{i,k} \) are vertex-to-node weights, but in contrast to \( \pi_{c} \) we compute them based on geodesic distances. \( \mathbf{G} \in \mathbb{R}^{K \times 3} \) are the node positions of the undeformed graph, \( \mathcal{N}_{vn}(i) \) is the set of nodes that influence vertex \( i \), and \( R(\cdot) \) is a function that converts the Euler angles to rotation matrices. We further apply the skeletal pose on the deformed mesh vertices to obtain the vertex positions in the input camera space

\[
\mathbf{V}_{c',i} = \sum_{k \in \mathcal{N}_{vn}(i)} w_{i,k}(R_{sk,k}(\theta, \alpha)\mathbf{Y}_i + t_{sk,k}(\theta, \alpha)), \tag{10}
\]

where the node rotation \( R_{sk,k} \) and translation \( t_{sk,k} \) are derived from the pose parameters using dual quaternion skinning \cite{105}. Eq. 9 and Eq. 10 are differentiable with respect to pose and graph parameters. Thus, our layer can be integrated in the learning framework and gradients can be propagated to DefNet. So far, \( \mathbf{V}_{c',i} \) is still rotated relative to the camera \( c' \) and located around the origin. To bring them to global space, we apply the inverse camera rotation and the global translation, defined in Eq. 4. \( \mathbf{V}_i = \mathbf{R}_c^T \mathbf{V}_{c',i} + \mathbf{t} \).

**Non-rigid Silhouette Loss.** This loss encourages that the non-rigidly deformed mesh matches the multi-view silhouettes in all camera views. It can be formulated using the distance transform representation \cite{104}

\[
\mathcal{L}_{\text{sil}}(\mathbf{V}) = \sum_{c} \sum_{i \in \mathcal{B}_c} \rho_{c,i} \| D_c(\pi_c(\mathbf{V}_i)) \|^2. \tag{11}
\]

Here, \( \mathcal{B}_c \) is the set of vertices that lie on the boundary when the deformed 3D mesh is projected onto the distance transform image \( D_c \) of camera \( c \). Those vertices are computed by rendering a depth map using a custom CUDA-based rasterizer that can be easily integrated into deep learning architectures as a separate layer. The vertices that project onto a depth discontinuity (background vs. foreground) in the depth map are treated as boundary vertices. \( \rho_{c,i} \) is a directional weighting \cite{27} that guides the gradient in \( D_c \). The silhouette loss ensures that the boundary vertices project onto the zero-set of the distance transform, i.e., the foreground silhouette.

**Sparse Keypoint Graph Loss.** Only using the silhouette loss can lead to wrong mesh-to-image assignments, especially for highly articulated motions. To this end, we use a sparse keypoint loss to constrain the mesh deformation, which is similar to the keypoint loss for PoseNet in Eq. 6

\[
\mathcal{L}_{\text{kp}}(\mathbf{M}) = \sum_{c} \sum_{m} \sigma_{c,m} \| \pi_c(\mathbf{M}_m) - \mathbf{p}_{c,m} \|^2. \tag{12}
\]

Differently from Eq. 6 the deformed and posed landmarks \( \mathbf{M} \) are derived from the embedded deformation graph. To this end, we can deform and pose the canonical landmark positions by attaching them to its closest graph node \( g \) in canonical pose with weight \( \omega_{p,g} = 1.0 \). Landmarks can then be deformed according to Eq. 6 resulting in \( \mathbf{M}_c' \) which is brought to global space via \( \mathbf{M}_c' = \mathbf{R}_c^T \mathbf{M}_c' + \mathbf{t} \).

**As-rigid-as-possible Loss.** To enforce local smoothness of the surface, we impose an as-rigid-as-possible loss \cite{106}

\[
\mathcal{L}_{\arap}(\mathbf{A}, \mathbf{T}) = \sum_{k} \sum_{l \in \mathcal{N}_{vn}(k)} u_{k,l} \| d_{k,l}(\mathbf{A}, \mathbf{T}) \|_1, \tag{13}
\]

where

\[
d_{k,l}(\mathbf{A}, \mathbf{T}) = R(\mathbf{A}_k)(\mathbf{G}_l - \mathbf{G}_k) + \mathbf{T}_k + \mathbf{G}_k - (\mathbf{G}_l + \mathbf{T}_l).
\]
$N_k$ is the set of indices of the neighbors of node $k$. In contrast to [106], we propose weighting factors $u_{k,l}$ that influence the rigidity of respective parts of the graph. We derive $u_{k,l}$ by averaging all per-vertex rigidity weights $s_{i,k,l}$ for all vertices (see Sec. 3.1), which are connected to node $k$ or $l$. Thus, the mesh can deform either less or more depending on the surface material. For example, graph nodes that are mostly connected to vertices on a skirt can deform more freely than nodes that are mainly connected to vertices on the skin. Without this loss, the deformations can strongly drift along the visual hull carved by the silhouette images without receiving any penalty leading to strong visual artifacts.

### 3.4 In-the-wild Domain Adaptation

Since our training set is captured in a green screen studio and our test set is captured in the wild, there is a significant domain gap between them, due to different lighting conditions and camera response functions. To improve the performance of our method on in-the-wild images, we fine-tune our networks on the monocular test images for a small number of iterations using the same 2D keypoint and silhouette losses as before, but only on a single view. This drastically improves the performance at test time as shown in the supplemental material.

### 4 Implementation Details

Both network architectures as well as the GPU-based custom layers are implemented in the Tensorflow framework [107]. We use the Adam optimizer [108] in all our experiments.

**Template Acquisition and Rigging.** To create the textured mesh (see Fig. 2), we capture the person in a static T-pose with an RGB-based scanner [2] which has 134 RGB cameras resulting in 134 images $I_{rec} = \{I_{rec_1}, \cdots, I_{rec_{134}}\}$. The textured 3D geometry is obtained by leveraging a commercial 3D reconstruction software, called Agisoft Metashape [2], that takes as input the images $I_{rec}$ and reconstructs a textured 3D mesh of the person (see Fig. 2). We apply Metashape’s mesh simplification to reduce the number of vertices $N$ and Meshmixer’s [3] remeshing to obtain roughly uniform shaped triangular surfaces. Next, we automatically fit the skeleton (see Fig. 2) to the 3D mesh by fitting the SMPL model [3]. To this end, we first optimize the pose by performing a sparse non-rigid ICP where we use the head, hands and feet as markers (called torso markers). After this, training has to be carefully designed. In general, our strategy first focuses on the torso markers by giving them more weight (see Sec. 3.2). Using this strategy, the global rotation will be roughly correct and joint angles are slowly trained to avoid overshooting of angular values. This is further ensured by our limits term. After several epochs, when the network already learned to fit the poses roughly, we turn off the regularization and let it refine the angles further. More precisely, the training of PoseNet proceeds in three stages. First, we train PoseNet for 120k iterations with a learning rate of $10^{-5}$ and weight $L_{kp}$ with 0.01. $L_{limit}$ has a weight of 1.0 for the first 40k iterations. Between 40k and 60k iterations we re-weight $L_{limit}$ with a factor of 0.1. Finally, we set $L_{limit}$ to zero for the remaining training steps. Second, we train PoseNet for another 120k iterations with a learning rate of $10^{-6}$ and $L_{kp}$ is weighted with a factor of $10^{-4}$. Third, we train PoseNet again 120k iterations with a learning rate of $10^{-6}$ and $L_{kp}$ is weighted with a factor of $10^{-5}$. We always use a batch size of 90.

**Training Strategy for PoseNet.** As we are interested in joint angle regression, one has to note that multiple solutions for the joint angles exist due the fact that every correct solution leads to the same loss value. To this end, training has to be carefully designed. In general, our strategy first focuses on the torso markers by giving them more weight (see Sec. 3.2). Using this strategy, the global rotation will be roughly correct and joint angles are slowly trained to avoid overshooting of angular values. This is further ensured by our limits term. After several epochs, when the network already learned to fit the poses roughly, we turn off the regularization and let it refine the angles further. More precisely, the training of PoseNet proceeds in three stages. First, we train PoseNet for 120k iterations with a learning rate of $10^{-5}$ and weight $L_{kp}$ with 0.01. $L_{limit}$ has a weight of 1.0 for the first 40k iterations. Between 40k and 60k iterations we re-weight $L_{limit}$ with a factor of 0.1. Finally, we set $L_{limit}$ to zero for the remaining training steps. Second, we train PoseNet for another 120k iterations with a learning rate of $10^{-6}$ and $L_{kp}$ is weighted with a factor of $10^{-4}$. Third, we train PoseNet again 120k iterations with a learning rate of $10^{-6}$ and $L_{kp}$ is weighted with a factor of $10^{-5}$. We always use a batch size of 90.

**Training Strategy for DefNet.** We train DefNet for 120k iterations with a batch size of 50. We used a learning rate of $10^{-5}$ and weight $L_{sil}$, $L_{arap}$, and $L_{kp}$ with 1k, 0.05, and 1.5k, respectively.

**Training Strategy for the Domain Adaptation.** To fine-tune the network for in-the-wild monocular test sequences, we train the pre-trained PoseNet and DefNet for 250 iterations, respectively. To this end, we replace the multi-view losses with a single view loss which can be trivially achieved. For PoseNet, we disable $L_{limit}$ and weight $L_{kp}$ with $10^{-6}$. For DefNet, we weight $L_{sil}$, $L_{arap}$, and $L_{kp}$ with 1k, 0.05, and 1.5k respectively. Further, we use a learning rate of $10^{-6}$ and use the same batch sizes as before. This fine-tuning in total takes around 5 minutes. All hyperparameters are empirically determined and fixed across different subjects.

---

1. https://www.treedvs.com/
5 RESULTS

All our experiments were performed on a machine with an NVIDIA Tesla V100 GPU. A forward pass of our method takes around 50ms, which breaks down to 25ms for PoseNet and 25ms for DefNet. During testing, we use the off-the-shelf video segmentation method of [109] to remove the background in the input image. Our method requires OpenPose’s 2D joint detections [27, 98, 99, 100] as input during testing to crop the frames and to obtain the 3D global translation with our global alignment layer. Finally, we temporally smooth the output mesh vertices with a Gaussian kernel of size 5 frames.

Dataset. We evaluate our approach on 4 subjects (S1 to S4) with varying types of apparel. For qualitative evaluation, we recorded 13 in-the-wild sequences in different indoor and outdoor environments shown in Fig. 4. For quantitative evaluation, we captured 4 sequences in a calibrated multi-camera green screen studio (see Fig. 5), for which we computed the ground truth 3D joint locations using the multi-view motion capture software, The Captury [110], and we use a color keying algorithm for ground truth foreground segmentation. All sequences contain a large variety of motions, ranging from simple ones like walking up to more difficult ones like fast dancing or baseball pitching. We will release the dataset for future research.

Qualitative Comparisons. Fig. 3 shows our qualitative results in in-the-wild test sequences with various clothing styles, poses and environments. Our reconstructions not only precisely overlay with the input images, but also look plausible from arbitrary 3D view points. In Fig. 5 and 7, we qualitatively compare our approach to the related human capture and reconstruction methods [15, 16, 27, 47] on our green screen and the in-the-wild sequences, respectively. In terms of the shape representation, our method is most closely related to LiveCap [27] that also uses a person-specific template. Since they non-rigidly fit the template only to the monocular input view, their results do not faithfully depict the deformation in other view points. Further, their pose estimation severely suffers from the monocular ambiguities, whereas our pose results are more robust and accurate (see supplemental video). Comparing to the other three methods [15, 16, 27] that are trained for general subjects, our approach has the following advantages: First, our method recovers the non-rigid deformations of humans in general clothes whereas the parametric model-based approaches [47, 49] only recover naked body shape. Second, our method directly provides surface correspondences over time which is important for AR/VR applications (see supplemental video). In contrast, the results of implicit representation-based methods, PIFu [15] and DeepHuman [16], lack temporal surface correspondences and do not preserve the skeletal structure of the human body, i.e., they often exhibit missing arms and disconnected geometry. Furthermore, DeepHuman [16] only recovers a coarse shape in combination with a normal image of the input view, while our method can recover medium-level detailed geometry that looks plausible from all views. Last but not least, all these existing methods have problems when overlaying their reconstructions on the reference view, even though some of the methods show a very good overlay on the input view. In contrast, our approach reconstructs accurate 3D geometry, and therefore, our results can precisely overlay on the reference views (also see Figs. 5, 8, 9 and 10).

Skeletal Pose Accuracy. We quantitatively compare our pose results (output of PoseNet) to existing pose estimation methods on S1 to S4. To account for different types of apparel, we choose S1 and S2 wearing trousers and a T-shirt or a pullover and S3 and S4 wearing a long and short dress, respectively. We rescale the bone length for all methods to the ground truth and evaluate the following metrics on the 14 commonly used joints [44] for every 10th frame: 1) We evaluate the root joint position error or global localization error (GLE) to measure how good the skeleton is placed in global 3D space. Note that GLE can only be evaluated for LiveCap [27] and ours, since other methods only produce up-to-scale depth. 2) To evaluate the accuracy of the pose estimation, we report the 3D percentage of correct keypoints (3DPCK) with a threshold of 150mm of the root aligned poses and the area under the 3DPCK curve (AUC). 3) To factor out the errors in the global rotation, we also report the mean per joint position error (MPJPE) after Procrustes alignment. We compare our approach against the state-of-the-art pose estimation approaches including VNect [44], HMR [47], HMMR [49], and LiveCap [27]. We also compare to a multi-view baseline approach (MVBL), where we use our differentiable skeleton model in an optimization framework to solve for the pose per frame using the proposed multi-view losses. We can see from Tab. 3 that our approach outperforms the related monocular methods in all metrics by a large margin and is even close to MVBL although our method only takes a single image as input. We further compare to VNect [44] fine-tuned on our training images for S1. To this end, we compute the 3D joint position using The Captury [110] to provide ground truth supervision for VNect. On the evaluation sequence for S1, the fine-tuned VNect achieved 95.66% 3DPCK, 52.13% AUC and 47.16mm MPJPE. This shows our weakly supervised approach yields comparable or better results than supervised methods in the person-specific setting. However, our approach does not require 3D ground truth annotation that is difficult to obtain, even for only sparse keypoints, let alone the dense surfaces. Further note that even for S3 we achieve accurate results even though she wears a long dress such that legs are mostly occluded. On S2, we found that our results are more accurate than MVBL since the classical frame-to-frame optimization can get stuck in local minima, leading to wrong poses.

Surface Reconstruction Accuracy. To evaluate the accuracy of the regressed non-rigid deformations, we compute the intersection over union (IoU) between the ground truth foreground masks and the 2D projection of the estimated shape on S1 and S4 for every 100th frame. We evaluate the IoU on all views, on all views expect the input view, and on the input view which we refer to as AMVloU, RVloU and S1VoI, respectively. To factor out the errors in global localization, we apply the ground truth translation to the reconstructed geometries. For DeepHuman [16] and PIFu [15], we cannot report the AMVloU and RVloU, since we cannot overlay their results on reference views as discussed before. Further, PIFu [15] by design achieves perfect overlay on the input view, since they regress the depth for each foreground
Fig. 4. Qualitative results. Each row shows results for a different person with varying types of apparel. We visualize input frames and our reconstruction overlayed to the corresponding frame. Note that our results precisely overlay to the input. Further, we show our reconstructions from a virtual 3D viewpoint. Note that they also look plausible in 3D.

Fig. 5. Results on our evaluation sequences where input views (IV) and reference views (RV) are available. Note that our reconstruction also precisely overlays on RV even though they are not used for tracking.

Further, we are again close to the multi-view baseline.

Ablation Study. To evaluate the importance of the number of cameras, the number of training images, and our DefNet, we performed an ablation study on S4 in Tab. 3. 1) In the first group of Tab. 3 we train our networks with supervision using 1 to 14 views. We can see that adding more views consistently improves the quality of the estimated poses and deformations. The most significant improvement is from one to two cameras. This is not surprising, since the single camera settings is inherently ambiguous. In Fig. 8, the importance of the number of cameras is also shown qualitatively. 2) In the second group of Tab. 3 and in Fig. 9, we reduce the training data to 1/2 and 1/4. We can see that the more frames with different poses and deformations are seen during training, the better the reconstruction quality is. This is expected since a larger number of frames may better sample the possible space of poses and deformations. 3) In the third group of Tab. 3, we evaluate the AMVloU on the template mesh animated with the results of PoseNet, which we refer to as PoseNet-only. One can see that on average, the AMVloU is improved by around 4%. Since most non-rigid deformations rather happen locally,
Fig. 6. Qualitative comparison to other methods [15], [16], [27], [47] on our green screen evaluation sequences. Note that our results overlay more accurately to the input view and also look more plausible from a reference view that was not used for tracking. Ground truth global translation is used to match the reference view for the results of [27], [47]. Since PIFu [15] and DeepHuman [16] output meshes with varying topology in a canonical volume without an attached root, it is not possible to apply the ground truth translation and therefore we show the reference view without overlay.

Fig. 7. Comparisons to related work [15], [16], [27], [47] on our in-the-wild sequences showing S1 and S4. Our approach can recover the deformations of clothing in contrast to [47] and gives more stable and accurate results in 3D compared to [27]. Moreover, note that in contrast to previous work [15], [16], our method regresses space-time coherent geometry, which follows the structure of the human body.

the difference is visually even more significant as shown in Fig. 11. Especially, the skirt is correctly deformed according to the input image whereas the PoseNet-only result cannot fit the input due to the limitation of skinning. In Fig. 10 we show the PoseNet-only result and our final result on one of our evaluation sequences where a reference view is available. The deformed template also looks plausible from a reference view that was not used for tracking. Importantly, DefNet can correctly regress deformations that are along the camera viewing direction of the input camera (see reference view in second column) and surface parts that are even occluded (see reference view in fourth column). This implies that our weak multi-view supervision during training let the network learn the entire 3D surface deformation of the human body. For more results, we refer to the supplemental video. 4) finally, in Fig. 12 we visually demonstrate the impact of our domain adaptation step. It becomes obvious

<table>
<thead>
<tr>
<th>Method</th>
<th>GLE</th>
<th>3DPCK</th>
<th>AUC</th>
<th>MPJPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNect [44]</td>
<td>66.06</td>
<td>28.02</td>
<td>77.19</td>
<td></td>
</tr>
<tr>
<td>HMR [47]</td>
<td>82.39</td>
<td>43.61</td>
<td>72.61</td>
<td></td>
</tr>
<tr>
<td>HMMR [49]</td>
<td>87.48</td>
<td>45.33</td>
<td>72.40</td>
<td></td>
</tr>
<tr>
<td>LiveCap [27]</td>
<td>317.01</td>
<td>71.13</td>
<td>37.90</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>91.08</td>
<td>98.43</td>
<td>58.71</td>
<td></td>
</tr>
<tr>
<td>MVBL</td>
<td>76.03</td>
<td>99.17</td>
<td>57.79</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 1
Skeletal pose accuracy. Note that we are consistently better than other monocular approaches. Moreover, we are even close to the multi-view baseline.

<table>
<thead>
<tr>
<th>Method</th>
<th>GLE</th>
<th>3DPCK</th>
<th>AUC</th>
<th>MPJPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNect [44]</td>
<td>80.50</td>
<td>39.98</td>
<td>66.96</td>
<td></td>
</tr>
<tr>
<td>HMR [47]</td>
<td>80.02</td>
<td>39.24</td>
<td>71.87</td>
<td></td>
</tr>
<tr>
<td>HMMR [49]</td>
<td>82.08</td>
<td>41.00</td>
<td>74.69</td>
<td></td>
</tr>
<tr>
<td>LiveCap [27]</td>
<td>142.39</td>
<td>79.17</td>
<td>42.59</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>75.79</td>
<td>94.72</td>
<td>54.61</td>
<td></td>
</tr>
<tr>
<td>MVBL</td>
<td>64.12</td>
<td>89.91</td>
<td>45.58</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>GLE</th>
<th>3DPCK</th>
<th>AUC</th>
<th>MPJPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNect [44]</td>
<td>82.06</td>
<td>42.73</td>
<td>72.62</td>
<td></td>
</tr>
<tr>
<td>HMR [47]</td>
<td>86.88</td>
<td>43.91</td>
<td>73.63</td>
<td></td>
</tr>
<tr>
<td>HMMR [49]</td>
<td>82.80</td>
<td>41.18</td>
<td>77.41</td>
<td></td>
</tr>
<tr>
<td>LiveCap [27]</td>
<td>248.67</td>
<td>75.11</td>
<td>37.35</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>96.56</td>
<td>96.74</td>
<td>59.25</td>
<td></td>
</tr>
<tr>
<td>MVBL</td>
<td>75.82</td>
<td>96.20</td>
<td>57.27</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8. Ablation for number of cameras used during training. The most significant improvement happens when adding one additional camera to the monocular setting. But also adding further cameras consistently improves the result as the yellow circles indicate.
### TABLE 2
Surface deformation accuracy. Note that we again outperform all other monocular methods and are close to the multi-view baseline. Further note, that for [16] an evaluation of the multi-view IoU is not possible since their output is always in local image space that cannot be brought to global space.

<table>
<thead>
<tr>
<th>Method</th>
<th>AMVIoU</th>
<th>RVIoU</th>
<th>SVIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMR [47]</td>
<td>62.25</td>
<td>61.7</td>
<td>68.85</td>
</tr>
<tr>
<td>HMMR [49]</td>
<td>65.98</td>
<td>65.58</td>
<td>70.77</td>
</tr>
<tr>
<td>LiveCap [27]</td>
<td>56.02</td>
<td>54.21</td>
<td>77.75</td>
</tr>
<tr>
<td>DeepHuman [16]</td>
<td>-</td>
<td>-</td>
<td>91.57</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>88.07</td>
<td>87.03</td>
<td>89.26</td>
</tr>
<tr>
<td><strong>MVBL</strong></td>
<td>91.74</td>
<td>91.72</td>
<td>92.02</td>
</tr>
</tbody>
</table>

### TABLE 3
Ablation study. We evaluate the number of cameras and the number of frames used during training in terms of the 3DPCK and AMVIoU metrics. Adding more cameras and frames consistently improves the quality of reconstruction. Further, DefNet improves the AMVIoU compared to pure pose estimation.

<table>
<thead>
<tr>
<th>Method</th>
<th>3DPCK↑</th>
<th>AMVIoU↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 camera view</td>
<td>62.11</td>
<td>65.11</td>
</tr>
<tr>
<td>2 camera views</td>
<td>93.52</td>
<td>78.44</td>
</tr>
<tr>
<td>3 camera views</td>
<td>94.70</td>
<td>79.75</td>
</tr>
<tr>
<td>7 camera views</td>
<td>95.95</td>
<td>81.73</td>
</tr>
<tr>
<td>6500 frames</td>
<td>85.19</td>
<td>73.41</td>
</tr>
<tr>
<td>13000 frames</td>
<td>92.25</td>
<td>78.97</td>
</tr>
<tr>
<td>PoseNet-only</td>
<td>96.74</td>
<td>82.53</td>
</tr>
<tr>
<td>Ours(14 views, 26000 frames)</td>
<td>96.74</td>
<td>82.53</td>
</tr>
</tbody>
</table>

### Fig. 9
Ablation for number of frames used during training. The more frames we used during training the better the result becomes as the network can better sample the possible pose and deformation space.

### Fig. 10
Our result from the input view and a reference view that was not used for tracking. Note that our DefNet can even regress deformations along the camera viewing axis of the input camera (second column) and it can correctly deform surface parts that are occluded (fourth column).

### 6 Conclusion

**Limitations.** Conceptually, both representations, pose and the non-rigid deformations, are decoupled. Nevertheless, since the predicted poses during training are not perfect, our DefNet also deforms the graph to account for wrong poses to a certain degree. In our supplemental video, we also tested our method on subjects that were not used for training but which wear the same clothing as the training subject. Although, our method still performs reasonable, the overall accuracy drops as the subjects appearance was never observed during training. Further, our method can fail for extreme poses, e.g. a hand stand, that were not observed during training.

We have presented a learning-based approach for monocular dense human performance capture using only weak multi-view supervision. In contrast to existing methods, our approach directly regresses poses and surface deformations from neural networks, produces temporal sur-
In addition, DefNet also corrects slight errors in the pose and typical skinning artifacts. This work was funded by the ERC Consolidator Grant 681446. In the future, we plan to incorporate hands and the face to pose and non-rigid deformation than existing methods. In our method produces more accurate 3D reconstruction of human body, and can handle loose clothes. Our qualitative and quantitative results in different scenarios show that our method produces more accurate 3D reconstruction of pose and non-rigid deformation than existing methods. In the future, we plan to incorporate hands and the face to our mesh representation to enable joint tracking of body, facial expressions and hand gestures. We are also interested in physically more correct multi-layered representations to model the garments even more realistically.

ACKNOWLEDGMENT
This work was funded by the ERC Consolidator Grant 681446. In the future, we plan to incorporate hands and the face to our mesh representation to enable joint tracking of body, facial expressions and hand gestures. We are also interested in physically more correct multi-layered representations to model the garments even more realistically.

REFERENCES


Marc Habermann works as a PhD student in the Graphics, Vision and Video group at the Max Planck Institute for Informatics. Within his thesis, he explores the modeling and tracking of non-rigid deformations of surfaces, e.g. capturing the performance of humans in their everyday clothing. In his previous works, he showed that this is possible at real-time frame rates and that the 3D performance can be further improved using deep learning techniques. He received the Guenter-Hotz-Medal for the best Master graduates in Computer Science at Saarland University in 2017 and his work, DeepCap, received the CVPR Best Student Paper Honorable Mention in 2020.

Michael Zollhoefer is a Visiting Assistant Professor at Stanford University. His stay at Stanford is funded by a postdoctoral fellowship of the Max Planck Center for Visual Computing and Communication (MPC-VCC), which he received for his work in the fields of computer vision, computer graphics, and machine learning. Before, Michael was a Postdoctoral Researcher in the Graphics, Vision & Video group at the Max Planck Institute for Informatics in Saarbruecken, Germany. He received his PhD in 2014 from the University of Erlangen-Nuremberg for his work on real-time static and dynamic scene reconstruction. His research is focused on teaching computers to reconstruct and analyze our world at frame rate based on visual input. To this end, he develops key technology to invert the image formation models of computer graphics based on data-parallel optimization and state-of-the-art deep learning techniques.

Christian Theobalt is a Professor of Computer Science and the head of the research group “Graphics, Vision, & Video” at the Max-Planck-Institute for Informatics, Saarbruecken, Germany. He is also a professor at Saarland University. His research lies on the boundary between Computer Vision and Computer Graphics. For instance, he works on 4D scene reconstruction, marker-less motion and performance capture, machine learning for graphics and vision, and new sensors for 3D acquisition. Christian received several awards, for instance the Otto Hahn Medal of the Max-Planck Society (2007), the EUROGRAPHICS Young Researcher Award (2009), the German Pattern Recognition Award (2012), an ERC Starting Grant (2013), an ERC Consolidator Grant (2017), and the Eurographics Outstanding Technical Contributions Award (2020). In 2015, he was elected one of Germany’s top 40 innovators under 40 by the magazine Capital. He is a co-founder of the Capture (www.thecapture.com).

Weipeng Xu is a research scientist at Facebook Reality Labs in Pittsburgh. He was a post-doctoral researcher at the Graphic, Vision & Video group of Max Planck Institute for Informatics in Saarbruecken, Germany. He received B.E. and Ph.D. degrees from Beijing Institute of Technology in 2009 and 2016, respectively. He studied as a long-term visiting student at NICTA and Australian National University from 2013 to 2015. His research interests include virtual human character, human pose estimation and machine learning for vision(graphics).