Human Shape Capture and Tracking at Home

Gaurav Mishra[1], Saurabh Saini[1], Kiran Varanasi[2], P J Narayanan[1]

[1]Center for Visual Information Technology, IIIT Hyderabad
[2]DFKI - Kaiserslautern

Figure 1: Left: Input to our system (only depth maps are used), Right: Output of our system, where first three meshes show SMPL model tracked over time and last three meshes show *Consensus meshes*, reposed using SMPL

Abstract

Human body tracking typically requires specialized capture set-ups. Although pose tracking is available in consumer devices like Microsoft Kinect, it is restricted to stick figures visualizing body part detection. In this paper, we propose a method for full 3D human body shape and motion capture of arbitrary movements from the depth channel of a single Kinect, when the subject wears casual clothes. We do not use the RGB channel or an initialization procedure that requires the subject to move around in front of the camera. This makes our method applicable for arbitrary clothing textures and lighting environments, with minimal subject intervention. Our method consists of 3D surface feature detection and articulated motion tracking, which is regularized by a statistical human body model [26]. We also propose the idea of a Consensus Mesh (CMesh) which is the 3D template of a person created from a single view point. We demonstrate tracking results on challenging poses and argue that using CMesh along with statistical body models can improve tracking accuracies. Quantitative evaluation of our dense body tracking shows that our method has very little drift which is improved by the usage of CMesh.

1. Introduction

Human shape and motion capture has been a largely studied topic in the field of computer vision. Recently, computer vision systems achieved 2D and 3D human body tracking from a simple capture setup *e.g.* convolutional neural network (CNN) models can detect body parts in RGB images [37]. However, these methods are not yet applicable for full body shape and motion visualization. Many applications in today's scenario like biomechanical analysis, medical rehabilitation etc. require motion tracking that is broadly accurate not only with respect to the position of the bone joints, but also on the surface of the skin.

Recently Bogo et al. [5] showed results on monocular RGB-D Kinect sequences of freely moving subjects to construct a detailed 3D reconstruction by doing coarse-to-fine processing. Such methods work with subject wearing tight fitting clothes. Models like Loper et al. [26] and Anguelov et al. [2] have been created from a large pool of real-world 3D scans of people to address the problem of human shape and motion capture. These models are accurate but they can only be fitted to a person wearing extremely tight clothes. It is challenging to track people in everyday clothing.

To this end we propose a novel pipeline for tracking people in everyday clothing. Unlike prior methods, we take only the depth channel as input, for the sake of simplicity and independence to illumination conditions and clothing texture. An important distinctive element of our method is that we do not rely on 3D body part detection *e.g.* using the Microsoft Kinect API or a deep neural-network model trained for this purpose. This makes our method a useful baseline method, which can be improved by such features or from alternative information channels such as RGB data.

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00049

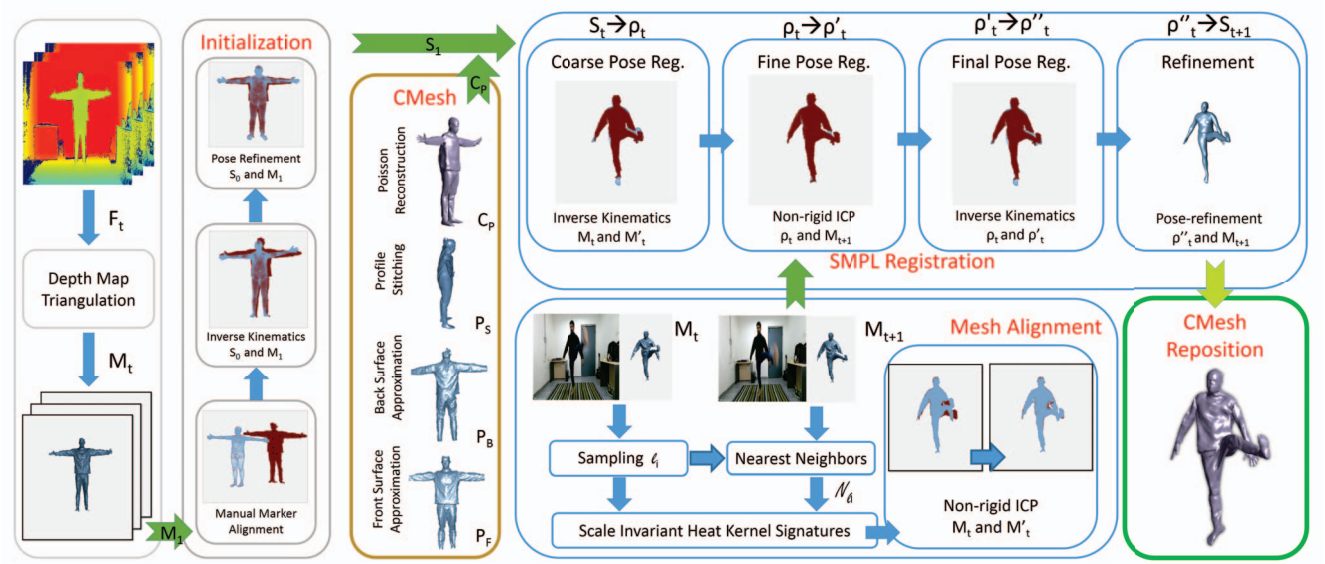

Figure 2: Block Diagram of our proposed method. First we process depth maps to generate triangulated meshes. We use these along with SMPL [26] model for manual initialization for first frame. After that we iterate between the *Mesh Alignment* and *SMPL Registration* stages for each frame. The final tracked, pose aligned and shape adjusted SMPL mesh S_{t+1} is generated as output. We also generate consensus mesh of the person from a 360° sequence which is then reposed using ARAP.

In addition to above we also propose a novel *Consensus Mesh (CMesh)* generation pipeline which refers to creating a 3D template mesh of a subject from 360° sequences captured using a single Kinect. Readers please note that our idea is not to create a high fidelity mesh, but a template mesh which captures the topology of the subject in consideration. Along with qualitative evaluation, we show quantitative evaluation by treating output sequences of De Aguiar et al. [14] and Vlasic et al. [35] as our ground truth. We show that we achieve comparable tracking from just a monocular depth input, with very little drift. We have also shown that using the concept of a *CMesh* we can greatly improve the tracking accuracy w.r.t. naked body models based tracking, simply because of the fact that *CMesh* has a better adherence to topology of the person. We have made several novel contributions :

1. We propose a novel tracking pipeline which consists of 3D HKS (heat-kernel signature) driven non-rigid ICP, articulated skeleton tracking and regularization by a statistical human body model, for fitting a 3D mesh template to a point cloud sequence.

2. We propose a body pose refinement step that uses statistical human body model for correspondence computation, and produces smooth trajectories over time.

3. We believe that we are the first to do temporal tracking on monocular depth input for subjects wearing casual clothes, using a statistical body model. We show tracking results for a variety of clothing styles and challenging poses.

4. We propose a pipeline to create a *Consensus mesh (CMesh)* using a single Kinect. Repositioning of the *CMesh* is shown to reduce quantitative drift and geometric errors in clothed models.

2. Related Work

As our goal is towards building a framework enabling a low-cost, easily-deployable pose tracking system without any restriction on clothing, we will focus only on markerless methods. We also want to estimate unrestricted body motion, so we exclude methods that recognize specific movements or track specific motion cycles from our review. Based on the complexity of data acquisition process, we can split human body motion capture methods into two groups. The first group consists of methods which require complex 3D data capturing setup such as laser scanners, inertial sensors or several multi-view, stereo pairs, depth cameras etc. [14, 18, 32, 42, 36, 17]. The second group consists of methods which use only a monocular RGB camera or a single depth sensor [38, 3, 13, 30, 23, 5]. In this case, as the data is limited and from one perspective only, most of the methods restrict themselves by defining system specific input constraints and priors. The output of this latter group is generally inferior compared to the former but the cost of the system, ease of capturing and setup, makes them applicable for the general public. An important factor in enabling the success of monocular methods is the availability of statistical human body models and datasets. So we first review them.

Human Body Models/Datasets: Several methods used for human body shape modeling, pose estimation and motion tracking use learned parametric human body models for regularization. The data for learning such models is obtained from 3D scans of humans in varying body shapes and poses, that are captured with a high quality multi-camera set-up and registered with each other to a common mesh template. These parametric models can generate new plausible body shapes and poses by interpolating between the data. The SCAPE model was introduced by Anguelov et al. [2] and was learned using several registered scans. It required shape and pose transformation to be applied separately on mesh triangles which sometimes lead to inaccuracies near joints. BlendSCAPE by Hirshberg et al. [19] addressed this issue by approximating triangle rotations as a linear combination of parts' rotations weighted using *blend weights*. Unlike the previous two models, Skinned Multi Person Linear (SMPL) [26] is a *vertex* based model of body shape and pose-dependent shape variations, which uses joint locations of kinematic chain of body parts. Although other complex models which can capture dynamic soft-tissue deformations and give textured outputs also exist (*e.g.* [29, 12, 5]), we choose to use publicly available SMPL for our framework as our main goal is not highly accurate shape reconstruction but motion tracking in complex clothing feasible for a common user. Several datasets associated with the models mentioned above exists for learning and benchmarking. Most of these are static [2, 8, 7, 4] but recently a few large dynamic datasets have also been introduced with captured motion [29, 6]. Although the recent datasets like [29, 4, 6] are better at emulating challenges of the real-world than synthetic datasets like [8, 7], they can not yet be used to represent the output from an inexpensive commodity depth sensor like Kinect which has relatively high noise and low resolution.

Multi-view Systems: Some earlier methods rely on static contours or silhouettes for estimating the topology of the shape but they either assume multi-view acquisition [34, 14] or process binary silhouettes as inputs [24, 16, 31]. Methods like [14, 18, 27, 40, 42, 17] depend on specialized data acquisition stage and hence inaccessible for a common user. Tong et al. [33] generate good quality 3D meshes but use three kinects, and require the person to stand on a rotating turntable while holding the pose. We differ from all of them as our system requires simple set-up of just one Kinect.

Monocular Systems - Dense Surface Reconstruction: Monocular pose estimation, owing to its under-constrained nature, is generally solved using strong priors and multi-stage optimization frameworks. A class of such methods aim specifically at building detailed user model assuming *static* or little motion in the input sequence [20, 27]. In [23], authors present a method for building watertight models of static scenes using only a single Kinect aimed for 3D printing, which requires the subject to rotate around while roughly holding the pose. Recent methods are able to achieve dynamic 3D surface reconstruction, also in real-time for virtual reality and teleconferencing applications [44]. These methods typically require the surface topology to be preserved during motion, such that shape regularization can be applied. Newcombe et al. [28] is able to reason about the canonical shape topology while reconstructing dense motion. Though their method result in a high fidelity surface reconstruction of arbitrary shapes, they show results on slow moving subjects and are not concerned with pose of the person or connecting them with a human body shape. 4D surface reconstruction of complex real world clothing with fast motion remains a challenging problem.

Monocular Systems - Shape and Motion Capture: If we do not require full surface reconstruction, but only human body shape estimation, certain additional assumptions can be placed. Weiss et al. [38] use a single Kinect and SCAPE body model [2] to recover human shape and pose in different configurations. They show results on minimally clothed people and do not attempt tracking. In [39] authors focus on fitting a minimally clothed shape (MCS) under complex clothing using motion cues. Baak et al. [3] use a generative-discriminative hybrid framework, in which they combine inferences from skinned kinematic chain model and retrieved pose from a curated dataset to decide the final pose in each frame. Their solution is data driven and limited by the number of poses in the dataset. Cui et al. [13] build a full 3D human model using a single Kinect for scanning but require the user to maintain a specific pose. In [30] authors focus on virtual *avatar* creation of a person in any pose using four static images from a commodity depth sensor but they do not explicitly model human motion. Efficient human body shape estimation and tracking is demonstrated by [43, 41] on MCS subjects.

Bogo et al. [5] is most similar to ours in motivation as they use *dynamic* monocular RGB-D Kinect sequences of a freely moving object to construct a detailed 3D reconstruction. They do coarse-to-fine processing involving several optimization stages and introduce a new multi-resolution body model based on BlendSCAPE [19]. Although their results have fine details, they do not tackle clothed subjects. Our method differs from theirs in that we do not use the color channel, and rely entirely on non-rigid surface matching on point clouds which is regularized by the statistical human body model. We show good model fitting irrespective of clothing type, additionally we have shown that tracking accuracy can be improved by using a subject specific *consensus mesh* along with statistical human body model.

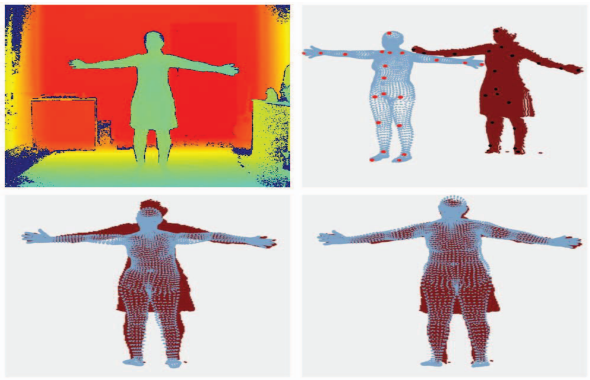

Figure 3: An example initialization. Top left: Depth image from Kinect. Top right: Markers placed on SMPL mesh and point cloud. Bottom left: IK solution. Bottom right: Fine tuned estimate.

3. Subsystem Description

Block diagram of our proposed system is depicted in Figure 2. Our system is divided into two parts. First part focuses on developing a tracking framework for subjects performing challenging actions in everyday clothing. This framework consists of various subsystems like 3D surface feature (SIHKS) driven Non-rigid ICP, Inverse Kinematics and pose refinement based on local energy minimization. Second framework focuses on generating a subject specific *consensus mesh* which is further used to improve tracking. Before explaining our full pipeline we first explain the major subsystems comprising our framework.

3.1. Dataset Acquisition

We used one Kinect V2 sensor for dataset capture, keeping depth and RGB resolution at 512×424 and 1920×1080 respectively. Human segmentation on depth stream was done using Kinect SDK 2.0. The input to our algorithm are segmented depth maps, we have used RGB images for visualization purpose only.

3.2. SMPL Model

As mentioned earlier we use the standard SMPL model [26] for regularizing shape in each frame. The model consists of 6890 vertices and its underlying skeleton contains 24 joints. The parameters for this model are 24×3 joint angles ($\theta_\tau, \tau \in \{1, 2, \dots 72\}$); 10 shape parameters ($\eta \in \{1, 2, \dots 10\}$) and additional 3 translation parameter of the root ($\Lambda \in \{1, 2, 3\}$). We concatenate all the parameters to build a 85 dimensional vector Θ. Using these we can generate SMPL mesh $S(\Theta)$ at any pose for a particular shape.

3.3. Non Rigid ICP

We adapt the registration algorithm by Amberg et al. [1] for non-rigid alignment in our framework by estimating landmarks using Scale Invariant Heat Kernel Signatures (SIHKS [9]). We register meshes by computing the full mesh transformation matrix X_{4n*3} (where n = number of vertices in input mesh). X is formed by vertically concatenating per vertex $4 * 3$ transformation matrices and is computed by minimizing the following equation:

$$E(X) = E_d(X) + \alpha E_s(X) + \beta E_l(X) \qquad (1)$$

Here E_d, E_s, E_l stand for the distance, stiffness and landmark energies respectively, regulated by α, β weight parameters. Both E_d and E_l are of the form $E(X) = \sum_i w_i \left|\left|X_i u_i - v_i\right|\right|^2$, where (u_i, v_i) represent initial correspondence pair. For E_d, v_i is computed using nearest neighbor for all vertices of input mesh. For E_l, (u_i, v_i) represents sparse SIHKS correspondences.

Here w_i is the indicator function, which is 0 for invalid correspondences. We use two constraints to check for the validity of u_i, v_i pairs. First constraint is that the angle between normals at u_i and v_i should be less than $45°$ and second constraint states that u_i must be visible to the camera. We decrease the contribution of the landmark energy term E_l by varying β from 1 to 0 as the algorithm proceeds. This is to capture the increasing confidence of ICP based correspondences compared to SIHKS, over the course of iterations.

Similar to Amberg et al. [1] we define E_s based on differences between transformation matrices assigned to neighboring vertices. To this end we build a node-arc incident matrix (Dekker [15]) by converting the input mesh into a directed graph and following the same process as in Amberg et al. [1]. Just like β we gradually decrease the stiffness factor α from 10 to 0.1 over the course of non-rigid ICP iterations. This models motion over various stiffness scales and hence capture the overall body part movement better.

3.4. Inverse Kinematics

Inverse Kinematics (IK) generates angular updates ($\Delta\theta$) of a kinematic chain given the parameterized initial (e_i) and the final position (e_f) of the end-effectors. Mathematically this can be written in matrix form involving Jacobian, $J(\theta)$ of joint angles [21]:

$$\Delta\theta = J'(JJ' + \lambda^2 I)^{-1}\bar{e}$$

Here $\bar{e} = \bar{e}_f - \bar{e}_i$ and λ is the damping constant. We use *Pseudo-Inverse Damped Least Squares* (PI-DLS [10]) for solving this as it provides well behaved solutions near singularities.

It is known that for an IK problem there are multiple solutions possible when the number of end effectors are sparse *e.g.* a few sensors attached to the free end of a robotic arm. In our case there are multiple such 'end-effectors' which are a few set of points attached to every bone of SMPL's

skeleton ($\sim$ 7-8 markers per bone). This introduces additional constraints in an otherwise under constrained system thereby restricting the search space of incorrect and trivial solutions. We run this algorithm for 50 iterations, ignoring a solution whenever upper (θ_τ^{max}) or lower (θ_τ^{min}) angular bounds of a joint (τ) are breached. These bounds are chosen for each joint in order to restrict the solutions to naturally feasible joint angles. *e.g.* sideways head rotation (along the vertical Y axis) $\theta_\tau^{min} = -90^\circ$ and $\theta_\tau^{max} = 90^\circ$. Please see supplementary material for all such joint limits.

3.5. Pose Refinement

This is a crucial and novel part of our proposed framework. It enables us to fine-tune a coarse SMPL estimate by minimizing a local energy term (E) defined as :

$$E = \alpha E_1 + \beta E_2 + \gamma E_3 \tag{2}$$

The intuition behind the various energy terms is explained below. E_1 penalizes the difference in the visible SMPL mesh vertices and the target point cloud. Let V_i be vertices of point cloud and V_f be its nearest neighbor in visible subset of SMPL mesh, then : $E_1 = \sum_i ||V_f - V_i||^2$.

Using E_1 alone can lead to unnatural human poses. We resolve this issue by defining E_2 using θ^{min} and θ^{max} for each θ in SMPL model s.t. $E_2 = \sum_j ||\theta_j - f(\theta_j)||^2$ where

$$f(\theta) = \begin{cases} \theta^{min}, & \theta < \theta^{min} \\ \theta, & \theta^{min} \leq \theta \leq \theta^{max} \\ \theta^{max}, & \theta > \theta^{max} \end{cases}$$

For temporal smoothing we add E_3 which restricts the current solution θ_{k_t} to be in close vicinity to the solution $\theta_{k_{t-1}}$ from the previous frame. $E_3 = \sum_k \left|\left|\theta_{k_t} - \theta_{k_{t-1}}\right|\right|^2$. It also helps in penalizing abrupt movements and limits jerky perturbations in the results. Hence the final energy can be defined by rewriting Equation 2 as a sum of L_2 terms :

$$E(\Theta) = \alpha \sum_i ||V_f - V_i||^2 + \beta \sum_j ||\theta_j - f(\theta_j)||^2 + \gamma \sum_k \left|\left|\theta_{k_t} - \theta_{k_{t-1}}\right|\right|^2 \tag{3}$$

For minimizing E we use quasi newton gradient descent algorithm (BFGS). During implementation we have used auto differentiation toolbox [25] for computing gradients.

4. Tracking Framework

Here we discuss step-by-step details for our iterative coarse-to-fine tracking framework as shown in Figure 2. Note again that we are only using segmented depth maps from Kinect as input. Before proceeding further, we define some common mathematical notations: Subscript t refers to a time instance in the sequence from 1 to number of frames in the sequence. Each depth map is denoted by F_t. The corresponding triangulated mesh is denoted as $\mathbf{M_t}$ and the SMPL mesh as $\mathbf{S_t}$. The neighbors of a vertex v_i in 3D space are denoted by $\mathcal{N}_{v_i}$ which are estimated using approximate nearest neighbor algorithm.

Depth Map Triangulation ($\mathbf{F_t} \rightarrow \mathbf{M_t}$): Given segmented depth maps, we convert them into triangulated meshes. For this we iterate row wise over the depth maps. We connect a pixel to its right and bottom neighbors if the edge length between them in point cloud space is less then a certain threshold ($<$ 5cm). This generates a good initialization mesh suitable for our purpose.

Initialization ($\mathbf{S_0} \rightarrow \mathbf{S_1}$): For initialization (refer Figure 3) we manually associate 24 markers on the default SMPL mesh S_0 with the corresponding points on $\mathbf{M_1}$. We apply IK to give us a coarse alignment between S_0 and $\mathbf{M_1}$ (subsection 3.4). To further refine the initialization we fine-tune this coarse alignment using Equation 3 which yields the final MCS denoted by S_1. This is the only manual step in our entire framework and needs to be performed only once for a sequence. Although there are no strict restriction regarding the starting pose of the subject but we use a common '**T**' pose for our experiments. For a random shape and pose, automatic initialization is a hard problem but can be solved to a certain extent using techniques mentioned in Bogo et al. [5]. However such an initialization is not the focus of our current work.

Mesh alignment ($\mathbf{M_t} \rightarrow \mathbf{M'_t}$): For mesh alignment between consecutive frames we use non-rigid ICP (subsection 3.3). We define landmarks as randomly sampled vertices near each joint in $\mathbf{M_t}$. We use SIHKS as mesh features which are quite robust but cannot differentiate between symmetric body parts. Furthermore in our case topological difference between $\mathbf{M_t}$ and $\mathbf{M_{t+1}}$ (which might arise due to unrestricted motion and loose clothing) also aggravate the problem. In order to deal with this, we match the landmarks $l_i \subset \mathbf{M_t}$ within a small neighborhood $\mathcal{N}_{l_i} \subset \mathcal{M}_{t+1}$, which yields less noisy correspondences. We denote the resultant aligned mesh as $\mathbf{M'_t}$.

SMPL registration ($\mathbf{S_t} \rightarrow \mathbf{S_{t+1}}$): We register the SMPL mesh $\mathbf{S_t}$ with the point cloud mesh $\mathbf{M_{t+1}}$ using a coarse-fine alignment strategy described below :

1. **Coarse pose registration ($\mathbf{S_t} \rightarrow \rho_\mathbf{t}$):** We apply IK on $\mathbf{S_t}$ using initial end-effectors from $\mathbf{M_t}$ and final from $\mathbf{M'_t}$ using sampling strategy defined in (subsection 3.4).

2. **Fine pose registration ($\rho_\mathbf{t} \rightarrow \rho'_\mathbf{t}$):** As the meshes $\rho_\mathbf{t}$ and $\mathbf{M_{t+1}}$ are relatively close, we use non-rigid ICP without landmarks for fine-tuning the alignment between these two meshes.

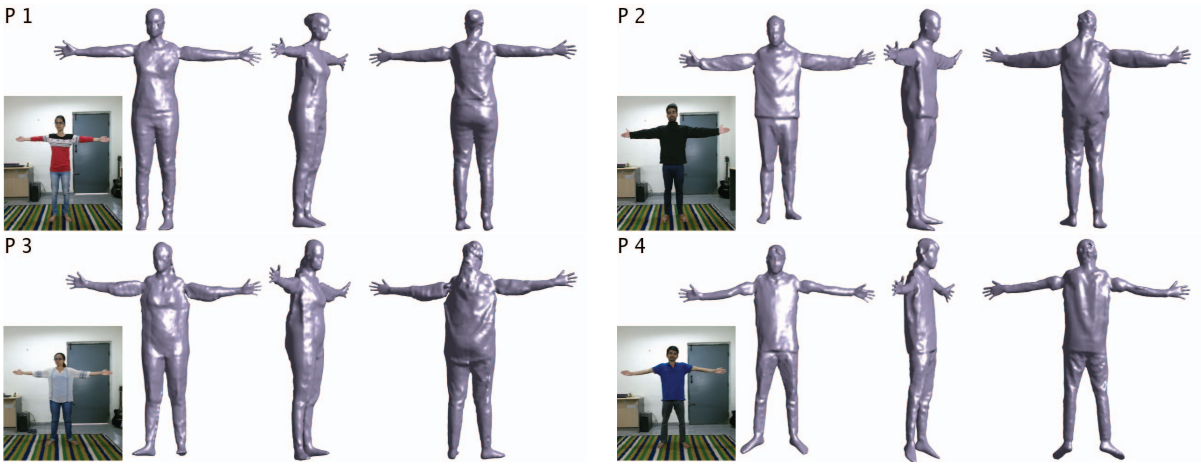

Figure 4: Consensus mesh results. Note how the generated meshes are more faithful to the true geometry compared to the SMPL mesh (Best viewed in color).

3. **Final pose registration ($\rho'_{\mathbf{t}} \rightarrow \rho''_{\mathbf{t}}$):** The result from the last step is pose correct but shape deformed. We perform IK again by choosing initial end effectors from $\rho_{\mathbf{t}}$ and final from $\rho'_{\mathbf{t}}$ which gives $\rho''_{\mathbf{t}}$ which is a refined estimate.

4. **Shape and pose refinement ($\rho''_{\mathbf{t}} \rightarrow \mathbf{S_{t+1}}$):** Finally we apply pose-refinement Equation 3 for fine-tuning the alignment of $\rho''_{\mathbf{t}}$ to give the pose and shape corrected $\mathbf{S_{t+1}}$.

We perform our automatic mesh alignment and SMPL registration steps for all pairs of consecutive frames. We require approximately 3-4 minutes per frame on a 3^{rd} generation Intel processor with 8 GB memory. As we are not aiming for a real time scenario, our framework is currently implemented in Matlab and C++ as a prototype code, which can be improved significantly for computational efficiency.

5. *Consensus mesh* generation

Following paragraphs explain the second major contribution of our work, which is creating a subject specific template mesh ($\mathbf{C_p}$) from a 360° sequence of the person, to assist tracking.

Retrieving candidate frames: We run tracking framework (section 4) on the sequence to get SMPL model parameters (Θ_t). We treat $\Theta_f = \Theta_1$ as our front canonical frame. Rotating the root of Θ_f by 180° we get Θ_b, which gives us back canonical frame. We retrieve $\mu_{\mathbf{f}}$ and $\mu_{\mathbf{b}}$ as the set of closest matching frames based on 3D positions of skeletal joints of Θ_f and Θ_b using nearest neighbor search. During implementation we have kept $|\mu_{\mathbf{f}}| = |\mu_{\mathbf{b}}| = 5$.

Pose alignment: In order to align meshes in $\mu_{\mathbf{f}}$ and $\mu_{\mathbf{b}}$ with their respective canonical frames Θ_f and Θ_b we do pose cancellation w.r.t. frame zero. For this we perform a reverse transformation for each point cloud mesh $\mathbf{M_i} \in \mu_{\mathbf{f}}$ to $\mathbf{M_0}$ (which represents the virtual point cloud mesh corresponding to rest pose of SMPL model ($\mathbf{S_0}$)). We then repose $\mathbf{M_0}$ to $\mathbf{M_f}$ using Θ_f. We perform same operations on $\mathbf{M_i} \in \mu_{\mathbf{b}}$. This gives us pose aligned nearest neighbor set $\mu_{\mathbf{f}'}$ and $\mu_{\mathbf{b}'}$.

Approximating front and back surfaces: Using $\mu_{\mathbf{f}'}$ and $\mu_{\mathbf{b}'}$ as static sequence inputs we execute Kinect fusion to obtain $\mathbf{P_f}$ and $\mathbf{P_b}$ as an approximation of front and back surfaces of the person. This step helps us in removing structured and real world noise due to Kinect, while simultaneously enriching the topology. Pose alignment in the previous step was necessary to remove noise caused due to movement of the person, which is essential for Kinect fusion. We substitute hands and feet from SMPL mesh (S_f) owing to the low resolution of Kinect depth maps in these regions.

Stitching everything together: In order to merge all estimated surfaces, we repose $\mathbf{P_b}$ to $\mathbf{P_f}$ using Θ_f and fill in the missing regions on the left and right profiles by interpolating vertices of SMPL mesh (S_f). Finally we run Poisson reconstruction ([22]) in Meshlab ([11]) to generate the final *consensus mesh* ($\mathbf{C_p}$) Figure 4.

Repositioning of Consensus Mesh: In order to repose the Consensus mesh in each frame according to the tracking, we use a set of highly aligned (as per a certain threshold) vertices between the SMPL and Consensus meshes and perform ARAP. This animates the movements using the Consensus mesh and reduces the tracking error due to better adherence to the true geometry.

Discussion about Consensus Mesh: As explained above, the CMesh ($\mathbf{C_p}$) is a clothed 3D mesh of a person, with a corresponding parametric SMPL model. $\mathbf{C_p}$ adheres better

Figure 5: Qualitative results of our framework on our dataset. Each figure shows the input point cloud (color coded), pose refined SMPL mesh and reposed CMesh. In inset figure we also show corresponding RGB image. Please refer to the supplementary video to notice the various challenging poses, actions, hairstyles and clothing worn by the subjects (Best viewed in color).

to the topology of the loosely-clothed person and the underlying SMPL allows it to be animated in plausible ways as shown before. The combination of CMesh and SMPL can be used for better human tracking, learning body shapes and cloth segmentation. The pair can also be used for pose-related cloth-deformations to retarget body shapes or virtual avatars, captured in more realistic settings.

6. Results and Discussion

For the purpose of evaluating the performance of our algorithm we captured several RGB-D sequences. Our subjects included 4 males and 4 females. Subjects wore challenging everyday clothes like Hoodie, Jeans, T-shirt, Loose top , different hairstyles etc. We recorded 11 sequences per subject (7 common and 4 different actions). Recorded actions included simple exercises, athletic action, Yoga poses etc. To emulate real world setting all sequences were recorded without any special background or body markers. Please refer supplementary for a detailed description of collected dataset. We will be releasing our implementation and the entire dataset to help research in this area.

Qualitative Results: We show qualitative performance of our system in Figure 5. We show color coded point cloud (blue = near, red = far), corresponding tracked SMPL mesh and reposed *consensus mesh* for a few key-frames for some of the captured sequences. Even with a very minimal input, our system is able to tackle challenging cases. Note how our results show correct tracked shape and pose for the following cases : *(i)* complex and fast motion (2c subject turning around, 3c kicking) *(ii)* challenging hairstyles (1b long tied hair, 3a pony tail) *(iii)* loose clothing and complex poses (1c, 2c and 3b Hoodie, 1b and 2b loose top . . .) *(iv)* significant self-occlusion (2c, 3a). This highlights the robustness and generality of our framework.

Quantitative Results: Contemporary methods that are based on monocular input do not tackle 'temporal tracking' with 'statistical body model' fitting specifically for subjects in 'casual clothes'. Lack of implementation resources (codes, complete datasets etc.) make comparisons hard to do. Hence to show objective effectiveness of our system we ran our algorithm on the dataset by De Aguiar et al. [14] and Vlasic et al. [35]. We use their results as our ground truth. To test our system we generate synthetic depth maps from a singular point of view using OpenGL. We additionally created a virtual 360° sequence of the person by rotating few ground truth meshes. We created *consensus mesh* using this sequence.

For quantifying the error of our SMPL tracking w.r.t. ground truth meshes (G_t) we compute mean absolute drift error ϵ_t, as follows: We find correspondences between vertices $a \in S_1$ and $b \in G_1$ by nearest neighbor search to get an ordered set $(a, b) \in C$. Consider a time step $F_t \rightarrow F_{t+1}$. during which $(a_t, b_t) \rightarrow (a_{t+1}, b_{t+1})$. For this transition we define ϵ_t as : $\epsilon_t = d(a_{t+1}, b_{t+1}) - d(a_t, b_t)$. Here $d(x, y)$ is the euclidean distance. ϵ_t measures error w.r.t. motion

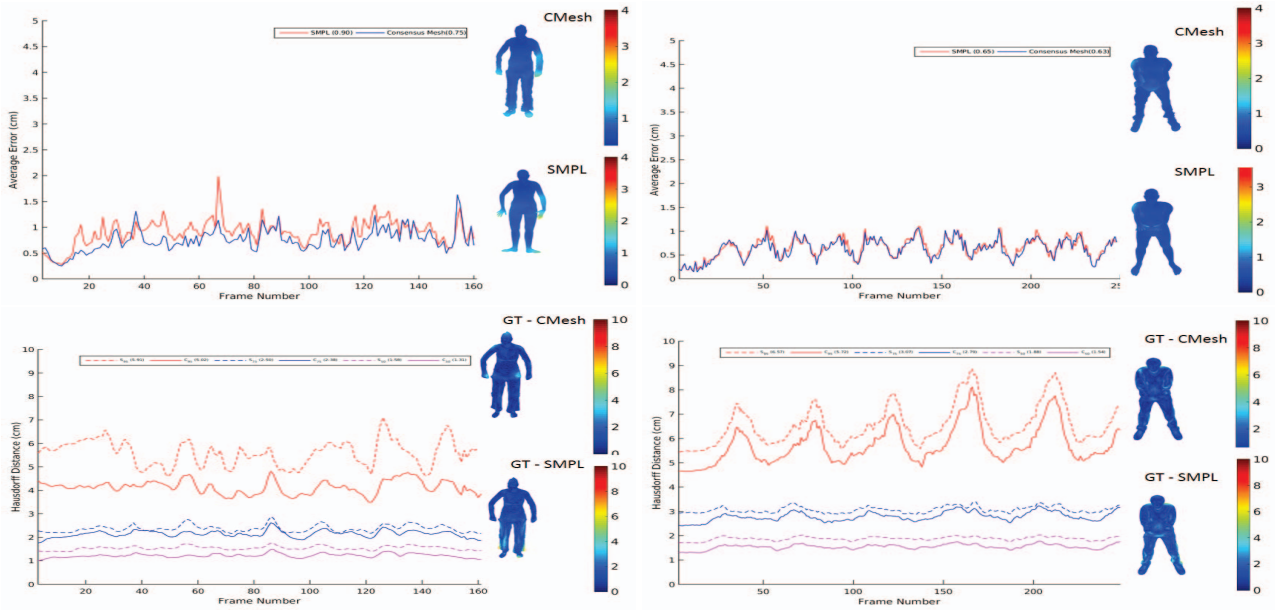

Figure 6: Quantitative results on De Aguiar et al. [14] and Vlasic et al. [35]. Top row shows graphs for mean drift error (ϵ_t) for various frames in the sequence (S31 [14] and I_squat [35]). The color coded CMesh and SMPL mesh represents average (ϵ_t) per vertex over all frames. Bottom row shows percentile based Hausdorff distance (d_l) per frame (for various $l = 95\%$ (red)), 75% (blue), 50% (magenta)). In the graph S stands for SMPL (dotted line) and C stands for Cmesh (solid lines) based errors. Color coded ground truth mesh based on nearest neighbor found w.r.t SMPL and Cmesh is also shown. (Best viewed in colors on screen)

over time but does not tell us anything about how close the geometry of our solution is to the ground truth mesh.

Hausdorff distance d_H is one way to measure the same but since it gives worst possible distance, it is sensitive to outliers. Hence we compute percentile based Hausdorff distance $d_l(P,Q) = \max\left\{\overset{l\%}{\max}_j \min_i ||y_i^p - y_j^q||, \overset{l\%}{\max}_i \min_j ||y_i^p - y_j^q||\right\}$, *e.g.* when $l = 50\%$ we are taking max over medians. We computed the same error metrices for CMesh w.r.t. ground truth Figure 6. Notice how errors for *CMesh* are low as compared to SMPL even for $l = 95\%$. Notice the significant percentage error reduction of repositioned *CMesh* w.r.t. SMPL tracking computed on average of ϵ_t and d_l for all frames (ϵ_t^i and d_l^i) in Table 1.

Sequence	ϵ_t^i	d_{95}^i	d_{75}^i	d_{50}^i
S08	2.40%	5.25%	10.58%	19.55%
S31	16.67%	14.96%	4.56%	17.06%
I_crane	15.45%	9.21%	12.50%	23.78%
I_jumping	12.88%	5.33%	8.04%	14.18%
I_march	16.08%	5.51%	10.25%	20.44%
I_squat	3.07%	12.97%	9.13%	17.97%

Table 1: Percentage error reduction of repositioned CMesh w.r.t. SMPL tracking computed over (ϵ_t^i and d_l^i). Top two sequences are from [14] and bottom four from [35]

Limitations : As we do not explicitly restrict the range of possible human poses, our system sometimes generate unnatural poses. Although our system is capable of handling fairly fast actions, it faces issues in highly challenging cases *e.g.* when fast actions are in conjunction with prominent self-occlusion or profile view. We have also observed that such cases can be corrected if the rate of capturing is fast. We are able to handle large range of casual clothing styles Figure 5 but our method can face issues in extremely challenging cases (*e.g.* Wedding dresses, Saree, Kimono, etc.), which might require explicit cloth modeling.

7. Conclusions

In this paper, we demonstrated a method for full 3D human body shape and motion capture for subjects wearing everyday clothes. Our method has the simple capture set-up of just one depth camera. We show effective articulated motion tracking, by iterating between computation of surface features and performing inverse kinematics regularized by a statistical human body model. Despite the simplicity of the method, our evaluation shows that it can track challenging poses. We also proposed a method for creating *Consensus mesh* of a person which can assist in tracking. In our current work we have shown that animating such a *CMesh* using tracked SMPL models improves tracking accuracy. In future we would like to use *CMesh* in our tracking pipeline itself to improve tracking further.

References

[1] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 4

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, July 2005. ISSN 0730-0301. 1, 3

[3] A. Baak, M. Mller, G. Bharaj, H. P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*, pages 1092–1099, Nov 2011. 2, 3

[4] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *CVPR*, pages 3794 –3801, Columbus, Ohio, USA, June 2014. 3

[5] Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *ICCV*, ICCV '15, pages 2300–2308, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-8391-2. 1, 2, 3, 5

[6] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *CVPR*, July 2017. 3

[7] A. M. Bronstein, M. M. Bronstein, U. Castellani, A. Dubrovina, L. J. Guibas, R. P. Horaud, R. Kimmel, D. Knossow, E. von Lavante, D. Mateus, M. Ovsjanikov, and A. Sharma. Shrec 2010: robust correspondence benchmark. 2010. 3

[8] Alexander Bronstein, Michael Bronstein, and Ron Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008. 3

[9] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *CVPR*, pages 1704–1711, June 2010. 4

[10] Samuel R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, IEEE Journal of Robotics and Automation, 2004. 4

[11] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. 6

[12] James Cownie, John DelSignore, Bronis R. de Supinski, and Karen Warren. Dmpl: An openmp dll debugging interface. In *Proceedings of the OpenMP Applications and Tools 2003 International Conference on OpenMP Shared Memory Parallel Programming*, WOMPAT'03, pages 137–146, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-40435-X. 3

[13] Yan Cui, Will Chang, Tobias Nöll, and Didier Stricker. Kinectavatar: Fully automatic body capture using a single kinect. In *Proceedings of the 11th International Conference on Computer Vision - Volume 2*, ACCV'12, pages 133–147, Berlin, Heidelberg, 2013. Springer-Verlag. 2, 3

[14] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. In *ACM Transactions on Graphics (TOG)*, volume 27, page 98. ACM, 2008. 2, 3, 7, 8

[15] M Dekker. Mathematical programming. *CRC, May*, 1986. 4

[16] Endri Dibra, A. Cengiz Öztireli, Remo Ziegler, and Markus H. Gross. Shape from selfies: Human body shape estimation using cca regression forests. In *ECCV*, 2016. 3

[17] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.*, 35(4):114:1–114:13, July 2016. 2, 3

[18] Juergen Gall, Carsten Stoll, Edilson De Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *CVPR*, 2009. 2, 3

[19] David A. Hirshberg, Matthew Loper, Eric Rachlin, and Michael J. Black. *Coregistration: Simultaneous Alignment and Modeling of Articulated 3D Shape*, pages 242–255. Springer Berlin Heidelberg, 2012. 3

[20] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM. 3

[21] Shuuji Kajita, Hirohisa Hirukawa, Kensuke Harada, and Kazuhito Yokoi. *Introduction to humanoid robotics*, volume 101. Springer, 2014. 4

[22] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):29, 2013. 6

[23] Hao Li, Etienne Vouga, Anton Gudym, Linjie Luo, Jonathan T. Barron, and Gleb Gusev. 3d self-portraits. *ACM Trans. Graph.*, 32(6):187:1–187:9, November 2013. ISSN 0730-0301. 2, 3

[24] Y. Liu, C. Stoll, J. Gall, H. P. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR*, pages 1249–1256, June 2011. 3

[25] M. Loper. Chumpy library. https://pypi.python.org/pypi/chumpy, 2017. 5

[26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015. 1, 2, 3, 4

[27] Alexandros Neophytou and Adrian Hilton. A layered model of human body and garment deformation. In *Proceedings of the 2014 2Nd International Conference on 3D Vision - Volume 01*, 3DV '14, pages 171–178, Washington, DC, USA, 2014. IEEE Computer Society. 3

[28] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, pages 343–352, June 2015. 3

[29] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 34(4):120:1–120:14, August 2015. 3

[30] Ari Shapiro, Andrew Feng, Ruizhe Wang, Hao Li, Mark Bolas, Gerard Medioni, and Evan Suma. Rapid avatar capture and simulation using commodity depth sensors. *Comput. Animat. Virtual Worlds*, 25(3-4):201–211, May 2014. ISSN 1546-4261. 2, 3

[31] Dan Song, Ruofeng Tong, Jian Chang, Xiaosong Yang, Min Tang, and Jian Jun Zhang. 3d body shapes estimation from dressed-human silhouettes. In *Proceedings of the 24th Pacific Conference on Computer Graphics and Applications*, PG '16, pages 147–156, 2016. 3

[32] C. Stoll, N. Hasler, J. Gall, H. P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *ICCV*, pages 951–958, Nov 2011. 2

[33] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. Scanning 3d full human bodies using kinects. *IEEE transactions on visualization and computer graphics*, 18(4):643–650, 2012. 3

[34] Kiran Varanasi, Andrei Zaharescu, Edmond Boyer, and Radu Horaud. *Temporal Surface Tracking Using Mesh Evolution*, pages 30–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. 3

[35] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, volume 27, page 97. ACM, 2008. 2, 7, 8

[36] Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Computer Graphics Forum 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics)*, 2017. 2

[37] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 1

[38] A. Weiss, D. Hirshberg, and M. J. Black. Home 3d body scans from noisy image and range data. In *ICCV*, pages 1951–1958, Nov 2011. 2, 3

[39] Stefanie Wuhrer, Leonid Pishchulin, Alan Brunton, Chang Shu, and Jochen Lang. Estimation of human body shape and posture under clothing. *Comput. Vis. Image Underst.*, 127: 31–42, October 2014. 3

[40] J. Yang, J. S. Franco, F. Hetroy-Wheeler, and S. Wuhrer. Estimation of human body shape in motion with wide clothing. In *ECCV*, 2016. 3

[41] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *CVPR*, pages 2345–2352, 2014. 3

[42] Chao Zhang, Sergi Pujades, Michael Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *CVPR*, 2017. 2, 3

[43] Q. Zhang, B. Fu, M. Ye, and R. Yang. Quality dynamic human body modeling using a single low-cost depth camera. In *CVPR*, pages 676–683, 2014. 3

[44] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Trans. Graph.*, 33(4): 156:1–156:12, July 2014. 3