SimulCap : Single-View Human Performance Capture with Cloth Simulation

Tao Yu$^{1,2}$, Zerong Zheng$^2$, Yuan Zhong$^2$, Jianhui Zhao$^1$, Qionghai Dai$^2$, Gerard Pons-Moll$^3$, Yebin Liu$^2$

$^1$Beihang University, Beijing, China  $^2$Tsinghua University, Beijing, China  $^3$Max-Planck-Institute for Informatics, Saarland Informatics Campus

Abstract

This paper proposes a new method for live free-viewpoint human performance capture with dynamic details (e.g., cloth wrinkles) using a single RGBD camera. Our main contributions are: (i) a multi-layer representation of garments and body, and (ii) a physics-based performance capture procedure. We first digitize the performer using multi-layer surface representation, which includes the undressed body surface and separate clothing meshes. For performance capture, we perform skeleton tracking, cloth simulation, and iterative depth fitting sequentially for the incoming frame. By incorporating cloth simulation into the performance capture pipeline, we can simulate plausible cloth dynamics and cloth-body interactions even in the occluded regions, which was not possible in previous capture methods. Moreover, by formulating depth fitting as a physical process, our system produces cloth tracking results consistent with the depth observation while still maintaining physical constraints. Results and evaluations show the effectiveness of our method. Our method also enables new types of applications such as cloth retargeting, free-viewpoint video rendering and animations.

1. Introduction

Real-time human performance capture using a low budget and an easy setup (e.g., a single depth camera) is a challenging but important task. Fulfilling this goal may enable many applications such as augmented reality, holography telepresence, virtual dressing, etc. Recent advances in single-RGBD 4D reconstruction [43, 71, 23] have enabled capture of geometry, motion, texture and even surface albedo [22] of human performances. However, results are still far from realistic.

The lacking of realism in existing capture methods (in great part) is due to the following limitations. First of all, they all use single piece of geometry for reconstruction (the observed human skin and dressed cloth are connected) so they cannot track and even describe cloth-body interactions, such as layering and sliding. Moreover, the reconstructed results are not editable and animatable, which is very important for many applications like virtual dressing. Second, clothing can not be described by the typically used kinematic chains or sparsely sampled non-rigid deformation node graphs [43, 22, 71], which leads to degraded capture accuracy and over-smoothed results. Third, they can not capture the dynamic deformations of the occluded part of the clothed person using single-view setups.

Cloth simulation methods are at the other side of the spectrum; those can generate plausible cloth dynamics on top of a moving body [60, 54, 27]. The problem here is adjusting the parameters to achieve a desired realistic animation. Furthermore, complex soft-tissue motions, and cloth-body interactions are extremely difficult to formulate using physics—even when allowing for long processing times.

In this work, we introduce SimulCap, a single view RGBD live capture method that combines a layered representation of the human body and clothing with physics based simulation; SimulCap captures cloth motion separately from the body, generates cloth dynamics which satisfy physical constraints, recovers the motion of the occluded parts, and produces a simulation that matches the observed data. SimulCap uses a multi-layer surface representation automatically extracted from the data; this is needed for cloth simulation and to model cloth layering. Our observation is that cloth deformation is caused mostly due to the skeletal motion of the underlying body, which is easier to capture. By simulating the cloth on top of the captured human body, we achieve two goals: (i) we obtain a good
initialization for data fitting of the visible part, and (ii) we can predict the occluded cloth part more accurately than the commonly used surface skinning [6] and non-rigid warping [35]. To capture detail beyond simulation, the observed cloth details, such as wrinkles, can be reconstructed by formulating data fitting as a physical process, which is not only much more efficient, but also preserves physical constraints in the cloth simulation step.

For single-view live capture of human performances, our method can reconstruct realistic results at the visible region and plausible results at the invisible area. SimulCap consists of two stages: multi-layer avatar digitization and physics-based performance capture as shown in Fig.2. We first automatically digitize the subject into a multi-layer avatar, which includes separate surfaces of the undressed body and each of the garments—the subject only needs to turn around one time in front of the camera at the beginning. During the performance capture step, we track both skeleton motion of the undressed body and the detailed non-rigid deformation of the cloth sequentially. By combining cloth simulation with iterative depth fitting, we can achieve realistic performance capture results. We also demonstrate the flexible ability of our system in cloth retargeting application. In summary, SimulCap combines the benefits of capture and simulation, and it constitutes the first live capture method capable of tracking human body motion and cloth separately, while incorporating physical constraints.

2. Related Work

Human Performance Capture. A large body of works require a pre-scanned template to model the body and the clothing using a single surface ([1, 61, 17, 38, 59, 48, 68, 69, 35, 21, 64]). Aguilar et al. [1], Vlasic et al. [61], Gall et al. [17] and Liu et al. [38] demonstrated high quality performance capture from multi-view video input. Ye et al. [69] embedded an articulated deformation model into a Gaussian Mixture Model for skeleton tracking. Li et al. [35] embedded a deformation graph [58] and tracked non-rigid surface motion imposing a local as-rigid-as-possible constraint. Xu et al. [64] combined skeleton motion, non-rigid deformation and 3D pose detection for performance capture from monocular RGB video. Habermann et al. [23] further demonstrates real-time capture from monocular RGB video. Although these methods can achieve very good performance, they require pre-acquisition of a template; furthermore, clothing and body are represented as a single connected surface, which limits their ability to capture detailed clothing deformations and cloth-body interactions.

Parametric body models can be used to bypass the requirement of a pre-scanned templates. Chen et al. [11] adopted SCAPE [5] to track skeleton motion from a single depth camera. Bogo et al. [7] extended SCAPE to capture detailed body shape (without clothing) with appearance during skeleton tracking. Capturing human shape and pose from an RGB image is much more challenging and ill-posed due to the lack of depth cues. Bogo et al. [8] constrains the problem using SMPL [39] to fit predicted 2D joint locations, while [28, 44, 45, 24] estimated shape and pose by integrating SMPL as a layer in a CNN-based framework. Other works focus on estimating body shape under clothing [72, 66, 63]. These works are restricted to the shape space of the body model, which can not represent personalized detail, clothing and hair. Recently, Alldieck et al. [4, 3, 2] reconstruct clothing and hair, represented as displacements on top of SMPL, from an RGB video of a person. The motions are restricted to rotating around the camera. Using an RGBD sensor, DoubleFusion [71] achieved highly robust and accurate capture for a variety of motions by combining SMPL with a voxel representation to represent clothing. None of these methods can separate each of the garments from the body, nor predict cloth deformations for the occluded parts on the RGB/depth video.

Another branch of work focuses on reconstructing the geometry and motion of non-rigid scenes simultaneously. Collet et al. [12] reconstruct high quality 4D sequences using multi-view setup and controlled lighting. Fusion4D [16] and Motion2Fusion [15] set up a rig with several RGBD cameras to capture dynamic scenes with challenging motions in real-time. DynamicFusion like approaches [43, 25, 22, 55, 70, 56, 34] reconstructed geometry and non-rigid motion simultaneously form a single-view and in real-time. The aforementioned methods either require multi-camera setups or can not capture the occluded regions.

Cloth Simulation and Capture. Works in this category model or capture clothing more explicitly. Simulation of clothing has been investigated for more than 30 years. Some works focus on super realistic cloth simulation results using millions of triangles [60, 54, 27], while the others concentrate on improving the fidelity for real-time simulation [51, 41, 40, 19, 18, 29, 31, 62, 14, 20, 65, 54]. Physics-based mass-spring models [51] or position based dynamics [41, 40] are commonly used for simulation of cloth. Realism, controllability and speed remain open problems for simulation methods. Example-based methods [31, 62, 14, 65, 20] learn from offline animations to achieve real time performance, but generalization to novel motions, shapes and fabrics is challenging.

Static cloth capture has been demonstrated, to some degree, from single images [73, 13] or RGBD [10]. Reconstruction typically requires manual intervention, or learning for a specific set of garments. Dynamic cloth reconstruction [9, 49] typically requires multi-view studios, and is restricted to capturing a single garment, and not the person wearing it. Both simulation and capture can be leveraged by estimating the physics parameters of cloth [53, 57] or soft-
Figure 2. The pipeline of our system. The first step is to get a multi-layer avatar using Double-Layer Surface Reconstruction and Multi-Layer Avatar Generation. Then we capture human performances by performing Body Tracking and Cloth Tracking, which include cloth simulation and iterative depth fitting, sequentially for each new RGBD frame. The final capture process is: turn around in front of a camera, wait several seconds for multi-layer avatar digitization and then start physics-based human performance capture in real-time.

tissue [32] from multi-view or 4D captured results, with the goal of driving simulation for new motions. Instead, we use simulation during capture to reconstruct occluded deformations from a single camera. Data-driven models are alternative to simulation; they learn how the clothing deforms on top of the body [42, 67, 33]. Although this is a promising direction, the models cannot separate the garments from the body [42, 67], or require garment specific learning [33].

The most relevant here is the work of Pons-Moll et al. [46](ClothCap). Similar to us, they jointly estimate body shape, pose and cloth deformation by using separate meshes for garments and body. However, they only demonstrate results using 4D scans as input, which do not suffer from occlusion. We address the more challenging monocular RGBD scenario, and show how physics-based simulation can help to capture the non-visible parts.

3. Multi-Layer Avatar Digitization

Multi-layer avatar digitization consists of two steps: double-layer surface reconstruction and multi-layer digitization. Double-layer surface comprises the surface of the dressed body and the undressed body as shown in Fig. 2.

We obtain a double-layer surface using DoubleFusion [71], which is a single-view, real-time method, which reconstructs the dressed body and undressed body surface at the same time. People only need to turn around once in front of a depth camera. To obtain a complete dressed surface without holes, we perform Poisson surface reconstruction [30] and remeshing [26]. This ensures a complete manifold for later segmentation and cloth simulation steps.

For multi-layer avatar generation, we parse and segment different cloth from the dressed body surface. However, cloth segmentation is a difficult task even for 2D images, so we require the colors of garment to be sufficiently different. By combining a learning based image parsing and volumetric fusion together we can get robust 3D cloth parsing and segmentation results efficiently and automatically.

To initialize the segmentation step, we use the state-of-the-art learning-based human parsing algorithm [37] to get cloth segmentation at the first frame as shown in Fig. 2. Then we estimate cloth colors using K-means and use it to segment the rest input rgb frames. We fuse all the color segmentation results into a parsing volume (A volume has the same resolution and size as the TSDF volume) which is not only robust to noisy 2D segmentation, but also very convenient to segment 3D clothing under the volume representation. Specifically, for each parsing voxel inside the truncated band of the TSDF volume, we first project it onto each input RGB image according to the tracked non-rigid motions, and store the corresponding pixel segmentation labels. The value saved in each parsing voxel is an array of label frequencies. Note that we only consider 3 labels (upper cloth, lower cloth and skin) in order to simplify the segmentation. In addition to label frequencies, we also fuse color into the parsing volume for subsequent segmentation on the surface using MRF. After cloth segmentation, we smooth the noisy boundaries of the segmented cloth pieces and handle the occlusion between multiple clothes by assuming that upper cloth is always outside pants.

In order to augment the realism of the reconstructed avatar, we enhance the head of the undressed body by deforming it to fit the fused head on the dressed surface.

4. Physics-Based Performance Capture

The physics-based human performance capture contains 2 steps: body tracking and cloth tracking. In the first step, we track the motion of the undressed body. In the second step, detailed cloth motion is tracked based on both, cloth simulation, and current depth input.
4.1. Body Tracking

The challenge of body tracking in our system is that we have to track accurate skeleton motion of the undressed body only given the depth of the dressed body. Moreover, the body-depth interpenetration during skeleton tracking, which is not considered in previous methods, is a very important factor in our system. This is because severe interpenetration will deteriorate the subsequent cloth tracing step as shown in Fig. 3.

Iterative closest point algorithm (ICP) is used for skeleton tracking. To eliminate the ambiguity between undressed body and dressed-depth-input, we leverage the reconstructed double layer surface (in Sec. 3) for constructing the tracking data term as in [71]. Moreover, we construct another interpenetration term to limit body-depth interpenetration. The energy function of body tracking is:

\[ E_{\text{skel}} = \lambda_{\text{data}} E_{\text{data}} + \lambda_{\text{inter}} E_{\text{inter}} + \lambda_{\text{pri}} E_{\text{pri}}, \]  

where the \( E_{\text{data}} \) measures the fitting between the skinned double layer and the input depth point cloud, please refer to [71] for detailed formulation; \( E_{\text{inter}} \) measures body-depth interpenetration; \( E_{\text{pri}} \) is human pose prior in [8] for penalizing unnatural poses.

The interpenetration term is defined as:

\[ E_{\text{inter}} = \sum_{(v_b, u_c) \in Q} |v_b - (u_c - n_{uc}\sigma)|^2, \]  

where \( Q \) is the correspondence set of all the interpenetrated body vertices \( v_b \) and its nearest cloth depth point \( u_c \); \( n_{uc} \) is the normal of \( u_c \); \(-n_{uc}\sigma\) represent a shift along the inverse normal direction to make sure the target position of \( v_b \) is behind the depth observation. By incorporating interpenetration term, we can get better body tracking and cloth tracking results as shown in Fig. 3.

4.2. Cloth Tracking

Given the undressed body with its motion, we can simulate plausible cloth dynamics for both visible and invisible regions. The simulated cloth provides very good initial status which facilitate fitting the cloth to the depth input. However, the depth fitting process remains non-trivial. It is difficult for previous methods like Laplacian Deformation and non-rigid registration to achieve wrinkle-level detailed deformations under a real-time budget and locally as-rigid-as-possible constraints. Moreover, stretching the simulated cloth to the depth input directly may generate many artifacts: First, since we can only get partial observation under single-view setup, there must be a gap between the simulated region and the depth fitting region on the cloth, which will lead to spatial discontinuity on the final reconstructed cloth mesh as shown in Fig. 5(b)(depth boundary region); Second, the direct depth fitting method may even break the consistency of the internal physical constraints and generate non-physical fitting results as shown in Fig. 5(b)(chest region), which will also lead to unexpected simulation results in the next frame. So we propose a method that performs depth fitting iteratively as a physical process, which can not only achieve efficient&realistic depth fitting, but also maintain the physical constraints in the simulation step.

4.2.1 Cloth Simulation

For cloth simulation, although a lot of advanced cloth simulation methods have been proposed in recent years, we extend classical Force-Based Mass-Spring method [51] due to its efficiency and simplicity. This method models cloth as a mass-spring system in which each vertex has a mass and all the vertices are connected by springs. Force-based methods calculate the resultant force for each vertex (mass) explicitly. The simulation steps can be concluded as:

- **Initialization:** Calculate the initial status of the mass-spring system (assign mass for each vertex and calculate rest status for all the springs);
- **Simulation:** For each vertex in each time step:
  1. **Resultant Force Calculation.** Calculate the sum of all the internal and external forces for each vertex, and then calculate vertex acceleration according to the fundamental law of dynamics. The internal forces are generated by different types of springs (internal constraints) while the external forces include omnipresent loads (e.g., gravity) and other specified external forces.
  2. **Vertex Position Update.** Using Explicit Euler Integration to update the position of each vertex according to the resultant vertex acceleration.
  3. **Collision Handling.** Detect and handle different types of collision for cloth and body.

We regard all the edges on the triangle mesh as stretching springs and they provide in-plane constraints. And to constrain the bending of the triangle mesh, we add an additional torsion spring on the common edges of two connected

Figure 4. Simulation forces in our system. (a) Resultant external force $F_e$ on vertex $p_1$, which include gravity $G$ and other external force $F_m$; (b) Stretching force $F_s$ between vertex $p_1$ and $p_2$, where $d$ is the rest length of the stretching spring and $\Delta x$ is half of the length difference between rest length and current length; (c) The rest status and delta angle $\Delta \alpha$ of current status of torsion spring. (d) Torsion force on the 4 vertices ($p_1$, $p_2$, $p_3$ and $p_4$) of the two connected triangles and the updated position.

Figure 5. Illustration of iterative depth fitting. From left to right: depth input (a), result of direct depth fitting (b), iterative depth fitting without (c) and with (d) smooth blending.

Figure 6. Illustration of depth fitting force $F_d$.

By adjusting the stiffness of different springs, we can approximate different types of cloth materials in the real world. We illustrate all the forces of our system below.

The resultant external force can be calculated as the summation of all the external forces, $F_e = G + F_m$, as shown in Fig 4(a). Note that we use the inverse normal direction of the floor (which we detected at the first frame) as the direction of gravity in our system.

We suppose all vertex have the same mass. The stretching force $F_s$ on the two vertices of stretching spring are equal and can be calculated as: $F_s = k \cdot \Delta x$, where $k$ is the stiffness of the stretching spring and $\Delta x$ is the half of length difference between current length and rest length of the stretching spring (Fig. 4(b)).

The moment of torsion spring is defined as $M = w \cdot \Delta \alpha$, where $w$ is the torsion coefficient and $\Delta \alpha$ is the angle difference between current angle and rest angle of the torsion spring (Fig. 4(c)). The angle of the torsion spring was calculated by the angle between the normal vectors of the two connected triangles. We calculate torsion force $F_t$ on each vertex (Fig. 4(d)) according to the bending constraints projection method in [41].

We suppose all the stretching spring has the same stretching coefficient $k$ while all the torsion spring has the same stiffness coefficient $w$ for simplicity. We implement the simulation method on GPU. Kernel merge and warp shuffle techniques are used to further improve performance. The time step is set to 0.00033s to avoid the overshooting and unstable problems of explicit Euler integration as illustrated in [51]. We perform explicit Euler integration 100 times between two rendering frames and linearly interpolate the position of each inner body vertex for body-cloth collision handling. We use static collision handling scheme for body-cloth collision while use continuous collision handling scheme for cloth-cloth collision as in [50].

4.2.2 Iterative Depth Fitting

Although the simulated cloth is plausible and consistent in both visible and invisible regions, it by no means exactly the same with the real-world cloth because real physical world always have much more factors that is hard to simulate, for example, the weave and structure of the cloth or even soft tissue motions of the undressed body etc. However, for performance capture systems, the goal is to capture real world performances efficiently, so the captured results should fit the real observations. Therefore, we need to fit the visible area of the simulated cloth to current depth input for generating more realistic cloth tracking results.

Inspired by the physics-based simulation algorithms, we formulate the depth fitting process as a physical process, in which the input depth point has attraction to the cloth. Thus, a new force should be defined and has positive correlation to the distance between cloth and depth. We name the new force as depth fitting force which only affects cloth vertices.
The force is defined as:

\[
F_d = \psi(p) \cdot \sum_{u^i \in N(u_c)} \eta \cdot \tau(u_c) \cdot e^\gamma |u^i - P| \cdot \frac{u^i - P}{|u^i - P|}.
\]

where \(P\) is a visible cloth vertex; \(u_c\) is the projective depth point of \(P\); \(u^i\) is the \(i\)th neighbor vertex in the 1-ring neighbor set \(N(u_c)\) of \(u_c\); \(\tau(u^i)\) is a 2D Gaussian kernel defined on \(u_c\) for blending the fitting forces \(F_d\) generated from all the neighbor points as shown in Fig. 6. \(\psi(P)\) is the smooth blending weight for depth fitting which we described in Fig. 7; \(\gamma\) and \(\eta\) are scaling factors to keep the force in a valid range and we set it to 240 and 0.34 respectively.

Iterative depth fitting is then achieved by performing cloth simulation again, in which we consider the driving force of the moving body and the depth fitting force as external force for iterative depth fitting as in the cloth simulation step. This is the reason why we have to split cloth simulation with body enhancement step takes 800 ms, with body tracking 0.5 ms, iterative depth fitting 20 ms and all the rest steps takes 5 ms.

Even we “simulate” the depth fitting process, we may still have spatial discontinuities around the depth boundaries as shown in Fig. 5(c). The reasons are two folds: On one hand, the simulated cloth cannot perfect align with the depth boundaries due to the non-perfect body tracking and cloth simulation results; On the other hand, the depth boundaries always contains much more noise than the central areas which makes situation even worse. To get a smooth transition between simulated region and depth fitting region around depth boundaries, we first generate a smooth blending mask using the 2D silhouette of the simulated cloth. Then scaling the depth fitting force for each cloth vertex according to the value of the smooth blending mask. The more a vertex is close to the boundary, the less it will be forced to fit depth observation. We illustrate the generation of the smooth blending mask in Fig. 7. The result of smooth blending is shown in Fig. 5(d). We iterate 100 times for iterative depth fitting as in the cloth simulation step.

Note that we have to perform cloth simulation first for getting a good approximation of current depth input and then perform iterative depth fitting. Other than that, the driving force of the moving body and the depth fitting force may conflict with each other and generate unnatural results. This is the reason why we have to split cloth simulation with iterative depth fitting.

5. Results

We demonstrate our results in Fig. 8. Note the faithful cloth dynamics that we reconstructed.

SimuCap is implemented on one NVIDIA TITAN Xp GPU. An efficient Gauss-Newton solver was implemented for body motion optimization. We perform cloth simulation in parallel and use kernel merge techniques to further improve the performance. The double-layer surface reconstruction takes 4 – 6 s for different self-turning around motions. The multi-layer avatar generation step takes 10 s, with post-geometry-processing 7.9 (poisson reconstruction 5.5 s and remeshing 2.4 s), cloth segmentation 1.5 s and body enhancement 0.5 s. Note that we perform parsing fusion along with the TSDF fusion step in double-layer surface reconstruction to improve efficiency. And the physics-based performance capture step takes 56 ms, with body tracking 17 ms, cloth simulation 14 ms, iterative depth fitting 20 ms and all the rest steps takes 5 ms.

For poisson reconstruction, we set the depth of octree as 8 and the final edge length of each triangle after remeshing is about 2 cm. In body tracking, we perform 6 ICP iterations, in which \(\lambda_{data} = 1.0, \lambda_{prior} = 0.01\) and we set \(\lambda_{inter} = 1.0\) for the first 3 iterations and \(\lambda_{inter} = 10.0\) for the rest iterations. For cloth simulation, we specify the stretching \(k\) and bending \(w\) parameters for each cloth at the beginning: The velocity damping coefficient in the simulation solver is set to 0.1 for more stable energy dissipation; The mass of each vertex is set to 0.01. We classify cloth material into 3 types: soft \((k = 300, w = 0.01)\), middle \((k = 800, w = 0.05)\) and hard \((k = 1300, w = 0.1)\). For each sequence, we choose a type of material parameters according to the real cloth material.

5.1. Evaluation

In this section, we first evaluate our method by comparing with state-of-the-art methods. Then we evaluate the reconstruction in the invisible region in detail, which is a very important improvement of our method. Finally, we evaluate the proposed iterative depth fitting method quantitatively.

Since we are the first single-view method for live cap-
Figure 8. Reconstruction results of our system.

Figure 9. Comparison with DoubleFusion and the multi-layer baseline method. (a) reference color image (not used during performance capture); (b) depth input; (c) our results; (d) results of DoubleFusion; (e) results of the multi-layer baseline method.

ture of human performances based on multi-layer surface representation, we compare with DoubleFusion, which is the state-of-the-art for single-view real-time human performance capture based on double-layer representation, to demonstrate the effectiveness of our method. Moreover, we also implement a baseline method for multi-layer performance capture, which leverages a multi-layer surface and perform ICP-based non-rigid tracking for each cloth independently. There are 4 typical improvements of our method:

First, our method achieves much more realistic cloth-body interactions (cloth sliding wrt the body and/or cloth leave the body) as shown in Fig. 9(c)(up). DoubleFusion uses a single piece of geometry for representing the outer surface, which means cloth and body are on the same piece of geometry, so they cannot handle naturally separations between cloth and body Fig. 9(d)(up). For the multi-layer baseline method, it is still very difficult for ICP-based non-rigid tracking methods to track such challenging interactive motions because of the limited observations (single-view setup) and real-time budget Fig. 9(e)(up).

Second, our method captures much more realistic cloth dynamics compared with the others benefiting from cloth simulation and the iterative depth fitting scheme Fig. 9(c)(middle). The Degree of Freedom (DOF) of non-rigid tracking in DoubleFusion and the multi-layer baseline method is limited due to the real-time budget, so they cannot track detailed cloth dynamics Fig. 9(d,e)(middle). Note that in DoubleFusion, it keeps fusing all the depth observation onto the surface, so it may capture cloth details when the subject keep relative still in front of the camera, but it will suffer from delay and fast motion (fast movements will smooth the fused surface details) because it is a temporal fusion process. For the multi-layer baseline method, the geometric details on the cloth, which correspond to the initial static template, do not change over time and, thus, not physically plausible. Moreover, the occluded surface areas is transformed according to skinning/warping alone in DoubleFusion and the multi-layer baseline method, which models mostly articulated deformations.

Third, a typical artifacts around armpits of DoubleFusion and the multi-layer baseline method Fig. 9(d,e)(bottom) can be eliminate by our method. The reason for such artifacts are erroneous surface fusion results, inaccurate skeleton embedding and smooth warping weight around such regions. Our method can generate more plausible results around such regions benefitting from the “divide-and-conquer” scheme as shown in Fig. 9(c)(bottom).

Finally, our method can infer plausible cloth dynamics even in the invisible region Fig. 10(2nd row)(please note the faithful direction and density of the wrinkles we reconstructed), which cannot be achieved by previous methods Fig. 10(3rd and 4th row).

We evaluate the proposed iterative depth fitting method quantitatively in Fig. 11. As shown in the figure, with iterative depth fitting, the reconstructed cloth dynamics is more accurate thus much more consistent to the depth input.
and the cloth meshes are simulation-ready, so the avatar can be easily incorporated into 3D engines for rendering new free viewpoint sequences. Moreover, much more realistic rendering (with dynamic shading effects) can be achieved given intrinsic texture on the cloth.

6. Conclusion

We proposed the first method that marries physics-based simulation with performance capture and is capable of tracking people and their clothing using a multi-layer surface. We demonstrated very realistic live capture results using a single-view RGBD camera. Higher realism is achieved because our forward model for tracking is closer to how bodies and clothing deform in the real world: Skeletal motion deforms the body, which in turn deforms the clothing layered on top of it. Modelling this process allows us track cloth-body interactions and hallucinate the surface in the occluded regions. We have also demonstrated that this allows to retarget captured clothing to different bodies. In summary, SimulCap demonstrates that modelling the physical process—even using a simple computationally efficient model—allows to capture performances from partial observations. We believe that this new direction for capture will enable the generation of photorealistic fully-animatable multi-layer avatars for analysis and synthesis, and will open many applications in VR/AR, virtual try-on and tele-presence.

Limitations and Future Work. Although we can reconstruct plausible cloth dynamics even for relatively loose clothing (e.g., skirts), the achieved realism in the occluded regions is limited by the quality of the simulator, and tracking of very thick clothing (e.g., sweaters) remains challenging. Incorporating more advanced cloth simulators and take into account the sewing patterns might increase the achieved realism. Moreover, capturing the natural interactions between hands/arms and cloth requires more accurate physics-based collision models. Finally, topology changes, face, hands and soft-tissue cannot be faithfully reconstructed using SimulCap, which remains challenging even for multi-view offline methods such as [46]. Fortunately, capturing clothing and body separately makes it straightforward to integrate new models of faces [36], hands [52] and soft-tissue [47]. Other potential future directions include: Incorporating human soft-tissue models (e.g., [47]) to faithfully capture cloth-body interactions, “learning” a data-driven clothing deformation model from captured results, and inferring material properties.

Acknowledgements This work is supported by the NSF of China No.61827805, No.61522111, No.61531014, No.61233005; Changjiang Scholars and Innovative Research Team in University, No.IRT_16R02; Gerard Pons-Moll is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—409792180.
References


[34] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 8


[38] Matthias Müller. Hierarchical position based dynamics. 2008. 2


[66] Jinlong Yang, Jean-Sébastien Franco, Franck Hétoy-Wheeler, and Stefanie Wuhrer. Estimation of Human Body Shape in Motion with Wide Clothing. In European Conf. on Computer Vision, Amsterdam, Netherlands, 2016. 2


