Realistic Virtual Humans from Smartphone Videos

Stephan Wenninger
Computer Graphics Group
TU Dortmund University

Jascha Achenbach
Computer Graphics Group
Bielefeld University

Andrea Bartl
HCI Group
Würzburg University

Marc Erich Latoschik
HCI Group
Würzburg University

Mario Botsch
Computer Graphics Group
TU Dortmund University

Figure 1: From monocular smartphone videos we generate realistic virtual humans that can readily be used in game engines.

ABSTRACT

This paper introduces an automated 3D-reconstruction method for generating high-quality virtual humans from monocular smartphone cameras. The input of our approach are two video clips, one capturing the whole body and the other providing detailed close-ups of head and face. Optical flow analysis and sharpness estimation select individual frames, from which two dense point clouds for the body and head are computed using multi-view reconstruction. Automatically detected landmarks guide the fitting of a virtual human body template to these point clouds, thereby reconstructing the geometry. A graph-cut stitching approach reconstructs a detailed texture. Our results are compared to existing low-cost monocular approaches as well as to expensive multi-camera scan rigs. We achieve visually convincing reconstructions that are almost on par with complex camera rigs while surpassing similar low-cost approaches. The generated high-quality avatars are ready to be processed, animated, and rendered by standard XR simulation and game engines such as Unreal or Unity.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
VRST '20, November 1–4, 2020, Virtual Event, Canada
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7619-8/20/11.
https://doi.org/10.1145/3385956.3418940

CCS CONCEPTS

• **Computing methodologies** → **Mesh geometry models**.

KEYWORDS

3D Reconstruction, Virtual Reality, Avatars

ACM Reference Format:
Stephan Wenninger, Jascha Achenbach, Andrea Bartl, Marc Erich Latoschik, and Mario Botsch. 2020. Realistic Virtual Humans from Smartphone Videos. In *26th ACM Symposium on Virtual Reality Software and Technology (VRST '20), November 1–4, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3385956.3418940

1 INTRODUCTION

Virtual humans provide promising applications in digital and interactive media, from entertainment, e.g., movies and gaming, to computer-mediated social interaction, or serious applications in simulation or medical areas in Virtual, Augmented and Mixed Reality (VR, AR, and MR; in short XR). An important aspect of several of these use cases is the believability of virtual humans. This seems obvious for synthetic actors used in movies, e.g., in *Locker* [22], *Rendez-vous in Montreal* [45] or the late *Star Wars* saga [25]. However, it also was demonstrated to be important for the general perception of virtual humans [60] either as virtual agents or avatars (see for instance [11]).

Believability of virtual humans encompasses three distinct levels of modeling: (i) realistic appearance, (ii) realistic motion, and (iii) realistic high-level behaviors [46]. While virtual agents are simulated and controlled by algorithms or pre-recorded animations, avatars, as digital alter-egos of users in virtual worlds, are interactively controlled by the users. Hence (i), realistic appearance modeling, is important for both, realistic virtual agents as well as for avatars. It is central, though, for realistic avatars of one self [59] as well as of others' [41] given a direct avatar control scheme by a reliable, i.e., accurate and fast full-body motion tracking system [56] to cover levels (ii) and (iii).

Several approaches for modeling realistic appearances of virtual humans exist today, from hand-crafted optimizations to 3D-reconstructions of real humans based on laser scanning, structured light scanning, or multi-view stereo. Reconstructed shapes have to be combined with high-quality textures as well as with suitable skeletal rigs and blendshapes for animation or tracking [56]. The results vary in terms of time, effort, and faithfulness of reproduction and depend on sensor accuracy, reconstruction principle, and degree of automatism.

Recent approaches based on an automated template matching (e.g., [2, 29]) achieve good reproduction results within 10–20 minutes. However, they depend on elaborated RGB camera rigs consisting of multiple dozens to a hundred of interconnected and synchronized camera devices, which results in complex and expensive setups. This paper presents a fully automated pipeline for creating highly detailed, animation-ready 3D avatars from a low-cost setup employing only a smartphone camera. The faithfulness of our virtual humans is largely comparable to reconstructions from more elaborated setups, and they are compatible with standard game and/or XR engines and frameworks. Since the overall complexity of the sensor equipment as well as the overall costs are drastically reduced, our approach opens up many more of the use cases of virtual humans in digital and interactive media applications.

2 RELATED WORK

As an alternative to reconstructing avatar models, one can record, transmit, and render streams of depth images from RGBD cameras, which creates believable reproductions of recorded users [43]. However, the quality of reproduction crucially depends on a sufficient resolution in both the spatial, color, *and* temporal domain of the employed RGBD cameras, which, as of today, still are significantly lower compared to dedicated high-quality sensors. Some performance capture approaches fuse RGBD streams from one or multiple sensors into a volumetric representation from which a textured mesh is extracted [23, 32]. These methods are template-free, i.e., they do not include a prior of human performances, and thus allow realtime reconstruction of challenging scenes of people interacting with objects. However, these approaches are restricted to mere reproductions of human performances, whereas full 3D virtual humans allow for more flexibility due to their separation of static geometry and appearance from dynamic animation.

Virtual human models can be reconstructed at a high degree of realism by utilizing sensors that are optimized for spatial and color resolution, such as multi-view stereo images, RGB video streams, or laser scans. High temporal resolution for capturing dynamic performances can then be delivered by dedicated motion tracking solutions. Reconstructions of virtual humans have to recombine geometric shape and accurate textures for high-quality appearance, as well as skeletal rigs and facial blendshapes for animation or tracking [56].

Today, such approaches usually exploit template models to guide the reconstruction process, see, e.g., Egger et al. [26] and Zollhöfer et al. [61] for face reconstructions. Similarly, human body models, such as the SCAPE model [9], have been used as template models for full body reconstruction (see, e.g., [52]). Later models like SMPL [44], SMPL-H [55], or SMPL-X [49] provide additional features like linear blend skinning, hand and finger movements, or facial expressions.

Highest quality avatar reconstructions are achieved using elaborated multi-camera rigs with high-quality image sensors, which often consist of dozens to over a hundred DSLR cameras. Through multi-view stereo these approaches accurately reconstruct both geometry and texture (see, e.g., [44, 53]). The virtual humans of Feng et al. [29] and Achenbach et al. [2] are reconstructed from such camera rigs (in 20 and 10 minutes, respectively) and feature skeleton-based body and hand animation as well as blendshape-based facial expressions. However, their complex hardware setup restricts the availability (and hence applicability) of their approaches.

Template-based human body models can also be generated from consumer-level RGBD sensors (e.g., [13]), but the low spatial resolution and limited image quality leads to rather low-quality reconstructions. Malleson et al. [47] therefore use an RGBD sensor in combination with a stereo RGB camera pair, but their avatars are still of rather low quality, lack facial details, and reconstruct the body in a stylized manner only. Lowering hardware requirements to the extreme, several learning-based techniques reconstruct 3D body models from a single RGB/RGBD input image or sequence of video frames [14, 27, 38, 39, 48]. However, these methods optimize parameters of a low-dimensional body model only, without considering fine-scale per-vertex displacements, which inherently limits the accuracy of the shape reconstruction. Moreover, they all do not consider texture reconstruction, which is crucial for realistic avatar appearance. Alldieck et al. [8] reconstruct textured avatars from a single image, by synthesizing normal/displacement maps from a partial texture calculated through DensePose [31] and mapping them onto the SMPL model. Limiting the input to a single image, however, inevitably restricts the faithfulness of the reconstruction.

Alldieck et al. [5, 6] therefore reconstruct a textured and animatable avatar from a monocular RGB video that captures a subject turning 360 degrees in A-pose. Their model is based on SMPL, which is fitted to the subjects silhouettes, extracted by CNN-based semantic segmentation, in a subset of the video frames. The shape is further refined using shape-from-shading techniques and an albedo texture is generated via a per-texel graph cut optimization with a semantic prior [6]. In follow-up work, Alldieck et al. [7] estimate the SMPL parameters from only 1–8 input images, based on a neural network that incorporates semantic segmentation and estimated 2D landmarks. The texture is again generated via [6]. While their approaches reconstruct full avatars from consumer-level input, we show that our approach leads to higher accuracy and realism. Our approach is inspired by Ichim et al. [36], who generate a quite accurate personalized head model from a smartphone selfie video. From this video they reconstruct a dense point cloud, to which

they fit a parametric template model. We extend their ideas to the challenging case of full-body avatars with detailed hands and faces.

Template-based performance capture methods employ an actor-specific model for tracking the movements of a person. For instance, Habermann et al. [33] generate this model by capturing an RGB video of the actor in static pose, extracting around 70 frames, reconstructing a textured mesh through photogrammetry, manually embedding a skeleton, and computing rigging weights using Blender. Our approach can act as a fully automatic alternative to their preprocessing stage. Besides providing more geometric details and animation controllers (fingers, facial expressions), it has the advantage that all actor models share the connectivity of the template mesh, allowing for statistical regularization.

The discussed reconstruction methods differ significantly in the faithfulness of their resulting models and in their costs, including hardware requirements and the amount of manual intervention needed. High-quality results with few manual intervention is offered by complex multi-camera rigs, like the one used in Achenbach et al. [2]. In contrast, the approaches of Alldieck et al. [5, 6, 7] require a single affordable camera only, but the quality of their reconstructions is considerably lower than the one achieved by multi-camera rigs. In this paper we describe a method that combines the advantages of both approaches, generating high-quality fully animatable virtual humans from video sequences captured by a consumer-level monocular smartphone camera.

3 METHOD

Our avatar generation is inspired by the smartphone-based head scanning of [36] and builds on our previous work on *Fast Generation of Realistic Virtual Humans* (abbreviated as *FGVH*) [2]. We combine and closely follow these two approaches, but extend them in several important aspects in order to enable full-body avatar reconstructions from simple monocular smartphone videos.

In FGVH [2] we scanned people using two custom-built single-shot multi-camera rigs: a full-body scanner and a face scanner, consisting of 40 and 8 DSLR cameras, respectively. Given the camera images, a multi-view stereo reconstruction computes two high-quality point clouds for body and face, to which a human body template is fitted using nonrigid (or deformable) registration. Since the holistic template model features a detailed skeleton for body and hands as well as eyes, teeth, and facial blendshapes, the reconstructed virtual humans are ready for animation in XR simulation and game engines. The main drawback of FGVH is the extensive hardware setup, an issue shared by several character reconstruction/tracking methods [29, 37, 44].

In order to make 3D-scanning and avatar generation available to a wider range of people, we considerably lower the hardware requirements and employ a consumer smartphone camera only. We take two video clips of a person, the first one capturing the full body, the second one capturing the head of the subject. From these video clips we automatically select individual frames using optical flow analysis and sharpness estimation (Section 3.1) and compute two dense point clouds for the body and head using multi-view stereo reconstruction (Section 3.2), thereby resembling the body and face scan of FGVH. We then pose and deform a human body template to closely fit the body and face point clouds (Section 3.4) and guide this process by a couple of feature landmarks. In contrast to FGVH, where landmarks in the point clouds are manually picked, our landmark detection is fully automatic (Section 3.3). When reconstructing the model's texture from the input frames, we cannot rely on standard multi-view reconstruction because of imperfections in our input data. Instead, we employ a graph cut texture stitching approach, which yields visually superior results (Section 3.5).

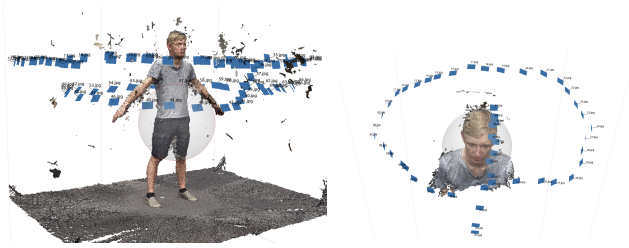

Figure 2: Camera locations for the full-body scan, consisting of two orbits around the scanned subject (left), and head scan, taking a close-up of the head/face region (right).

3.1 Input Data

Previous works on monocular reconstruction [6, 7] facilitate avatar creation from low-cost setups by taking one video that captures the full body of the person. However, we noticed (analogous to FGVH) that a separate head scan greatly improves the quality and detail of the avatar's head region. One approach for acquiring a close-up scan of the head would be to simply include it in the video for the full-body scan. However, since we employ a multi-view-stereo approach, we rely on the person holding as still as possible during the capture process. Increasing the length of the video by including a detailed scan of the head would imply more motion of the scanning subject and stronger violate the multi-view-stereo assumption.

Instead, we take two videos of the person, the first one capturing the full body in A-pose from a slight distance and the second one capturing the head in a close-up fashion. For the full-body video, the smartphone camera is moved (by a second person) in two circular paths around the scanned subject: The first camera path captures the upper body (head, torso, arms), the second one the lower body (hips, legs, feet). The head scan consists of one circular camera motion around the subject's head and additionally films the top of the head and the region under the chin (Figure 2).

Our input videos are shot at 4 k resolution (3840×2160) and 30 Hz frequency on a Google Pixel 3. Experiments with other smartphones capable of capturing 4 k videos gave similar results. The full-body video takes about 80 s and the head video about 30 s. The scanned subjects cannot hold perfectly still for this long, but we found that we could still employ a multi-view stereo approach and produce point clouds of sufficient quality.

To this end we first select N frames of the input video, which are then processed by the multi-view stereo reconstruction (Agisoft Metashape [4] in our case) in order to compute the point clouds for the subsequent template fitting pipeline. Using all frames of the input video would rapidly exceed the capabilities of the photogrammetry software. Our experiments revealed that extracting $N = 75$

images from the full-body video and $N = 50$ images from the head video is a good trade-off between computation time and resulting point cloud quality.

Simply extracting every n-th video frame would not account for any non-uniform camera movement by the person performing the scan. To simplify the capturing process while ensuring a uniform coverage of the scanned subject, we instead extract frames based on a uniform inter-frame movement, which we estimate through optical flow analysis using the implementation of Farnebäck [28] in OpenCV [18]. This yields a dense 2D flow field $\mathbf{f}_i$ representing the movement between frames i and $i + 1$, from which we estimate the *amount* of movement f_i as the average length of the 2D flow vectors in $\mathbf{f}_i$. We treat the resulting inter-frame movements f_i as a noisy 1D signal and smooth it by convolution with a Gaussian kernel ($\sigma = 2$) to compute filtered movement estimates $\tilde{f}_i$. We then iterate through the video and select a new frame once the *accumulated* movement between it and the previously selected frame reaches the threshold $\frac{1}{N}\sum_i \tilde{f}_i$. This defines a set of frames with uniform movement in between them.

We noted, however, that frames selected by the above procedure might be blurry either due to motion blur or the camera being in the process of adjusting the focus. We eliminate this problem by finding the sharpest frame in the k-neighborhood $\mathcal{N}_k$ of each selected frame ($k = 5$ in our experiments). We estimate sharpness as the variance of the Laplacian of the input image [50] and select the frame in $\mathcal{N}_k(i)$ with the highest value. We finally change the orientation of the selected frames according to the EXIF metadata of the video and pass the selected frames $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$ to the photogrammetry reconstruction.

3.2 Point Cloud Generation

The photogrammetry software Agisoft Metashape [4] proceeds in several steps: First, feature points are detected and matched in between individual input images. Based on these sparse (but reliable) points, the intrinsic and extrinsic camera parameters are computed for each input image. Finally, given the camera calibration, the dense point cloud is computed.

For the last step, the software allows to restrict the computation of the dense point cloud to an oriented bounding box. This will speed up not only the photogrammetry algorithm, but also all subsequent steps of our pipeline, because the resulting point cloud consists of fewer points. In contrast to FGVH, we cannot rely on a pre-calibrated camera setup and, thus, cannot rely on a constant scanning volume of interest.

However, we know that the camera positions provided by the extrinsic camera calibration enclose the scanned subject, hence we can use them to estimate the bounding box. We first determine an oriented box through PCA of the camera positions, where by design of our camera trajectory (see Figure 2) the first two principal directions $\mathbf{e}_1$ and $\mathbf{e}_2$ span the least-squares fitting plane through the camera locations, and $\mathbf{e}_3$ corresponds to its normal, i.e., the up-direction. From the camera box extent in directions $\mathbf{e}_1$ and $\mathbf{e}_2$ we can estimate the subject's arm span and, since the arm span of humans roughly corresponds to their height, also the height of the bounding box. The bounding box of the head scan is determined in a very similar manner.

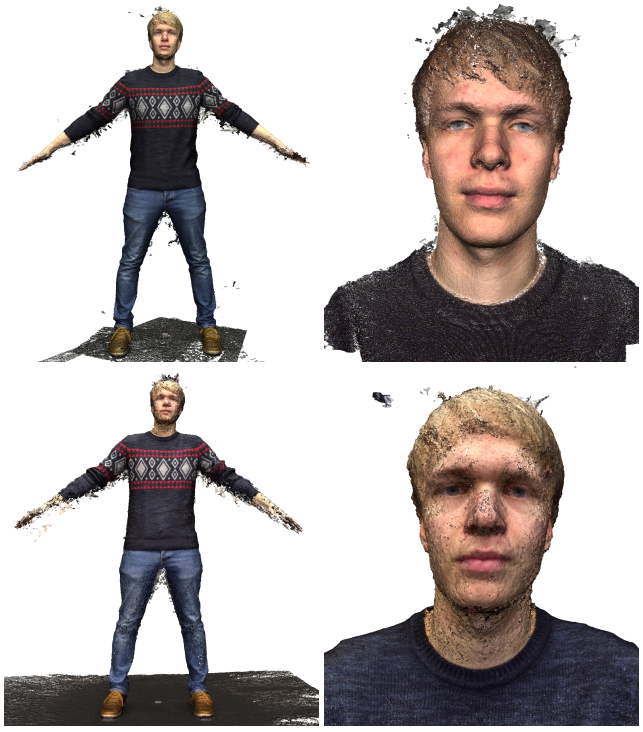

Figure 3: Comparison of full-body (left column) and face/head (right column) point clouds between FGVH (top row) and our approach (bottom row). Our point clouds are more noisy, less detailed, and more likely to have missing data (e.g., in the arm region).

After specifying the two bounding boxes, Agisoft Metashape computes dense point clouds from the selected input camera images, leading to a point cloud $\mathcal{P}_B$ for the full body scan (ca. 2.8 M points) and a point cloud $\mathcal{P}_H$ for the head scan (ca. 1.6 M points). Due to the lower resolution of our smartphone camera and the inevitable slight motion of the scanned subject during the capture process, our point clouds are more noisy and more likely to have missing data than the point clouds in FGVH (see Figure 3 for a comparison).

3.3 Landmark Detection

The template fitting procedure (Section 3.4) is boot-strapped and guided by feature landmarks on the point clouds $\mathcal{P}_B$ and $\mathcal{P}_H$. While in FGVH landmarks in the reconstructed point clouds are manually selected, we propose a fully automatic landmark detection.

In order to automatically detect these landmarks, we employ the landmark estimation of OpenPose [21] to all input images, which gives us for each image up to 135 landmarks (including confidence values c_i): 25 full-body landmarks defining a 2D skeleton, 21 landmarks per hand, and 68 facial landmarks. The detected landmarks have to be filtered in order to deal with unreliable detections. We address this issue by discarding all detections that belong to skeletons with less than 4 detected bones or which exhibit a maximum confidence value lower than 0.5.

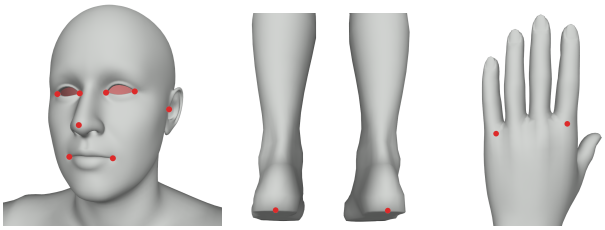

Figure 4: We use 15 landmarks on the full-body scan for guiding the template fitting process. The location of these landmarks is visualized here on the template mesh.

For the full-body point cloud $\mathcal{P}_B$ we only use a small subset of 15 landmarks (shown in Figure 4), since not all of the 135 landmarks can be reliably back-projected from their 2D image location onto the 3D point cloud (using the camera calibration data from the photogrammetry reconstruction). However, this subset turned out to be fully sufficient for guiding the full-body template fitting process.

Since each of the required 15 landmarks might have been detected in several input images, we have to select the one that allows for the most robust back-projection from 2D image coordinates onto the 3D point cloud $\mathcal{P}_B$. We choose the most suitable image based on the following heuristics: Hand and ear landmarks should be back-projected from images orthogonal to the sagittal plane, while heel, nose, mouth, and eye landmarks should be back-projected from images orthogonal to the frontal plane of the human body.

Images orthogonal to the sagittal plane are found by examining the shoulder landmarks. These exhibit a small lateral distance in suitable images (see Figure 5). The distinction between the left and right side of the sagittal plane is done based on the confidence values for the left and right finger and ear landmarks.

For finding suitable images for the heel landmark projection, we look for several characteristics: For one, the left and right heel landmarks have to be located on the left and right side of the image, respectively. However, OpenPose mislabels left and right legs in some cases, so we additionally use the fact that in suitable images the toe landmarks always have to be above the heel landmarks. In images orthogonal to the frontal plane the heel landmarks should also approximately be located at the same height in the input image.

Projecting the nose, mouth, and eye landmarks should ideally be done from the most frontal image, which we find through a combined measure of horizontal and vertical frontality. Horizontal frontality is computed in terms of the symmetry of the landmarks around the center line of the face, where a higher symmetry score indicates higher frontality. Vertical frontality is computed as the ratio between eye height and eye width. The bigger this ratio, the more orthogonal the viewing vector is to the frontal plane of the human body. From all images we assume the image with the highest sum of these measures to be the most frontal image.

The landmarks in the selected images are then back-projected onto the point cloud $\mathcal{P}_B$ using the camera calibration provided by the photogrammetry software. The same procedure is repeated for the head scan: We find the most frontal image and project the 68 facial landmarks onto $\mathcal{P}_H$.

Figure 5: Result of the automatic landmark detection. We heuristically find the best image for each landmark. Nose, mouth, and eye landmarks are projected from frontal images (left), hand and ear landmarks from lateral images (center) and heel landmarks from dorsal images (right).

The landmark detection of OpenPose in combination with our filtering and back-projection yields—in a fully automatic manner—3D landmark positions (15 for the full-body scan, 68 for the head scan), which guide the subsequent template fitting procedure. Note that the back-projection might fail in case of missing data in low-quality point clouds; in this rare case we prompt the user to manually select the corresponding 3D landmark (see Figure 13).

3.4 Template Fitting

Reconstructing a high-quality avatar mesh from medium-quality scanner data is a challenging problem because of noise, outliers, and holes in the input data. Like FGVH we exploit prior knowledge (that we are scanning humans) by fitting a statistical human body model to the scanner point cloud(s) by optimizing the template's position, orientation, scaling, PCA parameters, and fine-scale per-vertex deformation. In this way the template mesh regularizes the fitting procedure and fills up regions of missing data. Our template fitting approach closely follows the nonrigid registration of FGVH, but extends it at several places in order to deal with our lower-quality data. Due to space constraints we can only briefly recap FGVH and therefore mainly point out where our method differs.

We use the same template character from the Autodesk Character Generator [10], which is fully rigged and capable of body, hand, and face animations. The template mesh consists of $n \approx 21\,\text{k}$ vertices with positions $\mathcal{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$. In order to incorporate a statistical prior on human body shapes, we fit this template model to about 1700 human scans from the CAESAR database [54] and compute a 30-dimensional PCA subspace from the resulting data. This yields a more expressive statistical model—and hence a more robust fitting process—than FGVH, where a 10-dimensional PCA is computed from about 200 scans from mixed sources [10, 15, 34], including non-realistic ones [10].

Following FGVH we uniformly down-sample the two point clouds to twice the vertex density of the template mesh in order to speed up the fitting process, resulting in ca. 150 k points each for the body scan and the head scan.

In the first step we coarsely fit the template model to the body point cloud $\mathcal{P}_B$ by optimizing the alignment, pose, and coarse shape (in the 30-dim. PCA space) of the template model. To this end, we minimize the squared distances between the 15 automatically

detected landmarks $\mathcal{L}$ in the point cloud $\mathcal{P}_B$ and their pre-selected counterparts on the template model by alternatingly (i) computing the optimal scaling, rotation, and translation [35], (ii) optimizing joint angles through inverse kinematics [20], and (iii) optimizing the PCA shape parameters (linear least squares problem). After convergence, we further improve alignment, pose, and PCA shape by, in addition to landmarks $\mathcal{L}$, also minimizing squared distances between points in $\mathcal{P}_B$ and their closest points on the template mesh.

Let $\bar{\mathcal{X}} = (\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n)$ be the vertices resulting from the coarse-alignment phase. We then perform a fine-scale nonrigid registration by minimizing the nonlinear objective function

$$
\begin{aligned}
E_{\text{body}}(\mathcal{X}) &= \lambda_{\text{lm}} E_{\text{lm}}(\mathcal{X}) + \lambda_{\text{close}} E_{\text{close}}(\mathcal{X}) + \lambda_{\text{reg}} E_{\text{reg}}(\mathcal{X}), \\
E_{\text{lm}}(\mathcal{X}) &= \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \left\| \mathbf{x}_l - \mathbf{p}_l \right\|^2, \\
E_{\text{close}}(\mathcal{X}) &= \frac{1}{\sum_{c \in C} w_c} \sum_{c \in C} w_c \left\| \mathbf{x}_c - \mathbf{p}_c \right\|^2, \\
E_{\text{reg}}(\mathcal{X}) &= \frac{1}{\sum_e A_e} \sum_{e \in \mathcal{E}} A_e \left\| \Delta^e \mathbf{x}(e) - \mathbf{R}_e \Delta^e \bar{\mathbf{x}}(e) \right\|^2.
\end{aligned}
\tag{1}
$$

The *landmark term* E_{lm} penalizes the squared distance between the 15 automatically detected landmarks $\mathbf{p}_l$, $l \in \mathcal{L}$, in the point cloud and their corresponding vertices $\mathbf{x}_l$ on the template mesh (Figure 4). Similarly, the *closeness term* E_{close} penalizes the squared distance between corresponding closest points $\mathbf{p}_c$ in the point cloud and $\mathbf{x}_c$ on the template, where C is the set of these closest point correspondences. The closest points $\mathbf{x}_c$ are located on the mesh surface and represented via barycentric coordinates. Using $w_c \in [0, 1]$ we weight-down correspondences in hand and head regions, since the former are typically unreliable and the latter will be replaced by the head scan. The *regularization term* E_{reg} penalizes the geometric distortion from the undeformed state $\bar{\mathcal{X}}$ to the deformed state $\mathcal{X}$. In particular, it measures, for each edge $e \in \mathcal{E}$, the squared deviation of deformed edge-Laplacian $\Delta^e \mathbf{x}(e)$ and rotated undeformed Laplacians $\mathbf{R}_e \Delta^e \bar{\mathbf{x}}(e)$, weighted by edge area A_e (see [3] for details). This nonlinear least-squares problem is solved using an alternating optimization for vertex positions and edge rotations (repeated block-coordinate descent), where we set λ_{close} to 1 and gradually decrease λ_{lm} from 0.1 to 10^{-4} and λ_{reg} from 1 to 10^{-7}.

Having deformed the template model to the full-body scan, we further refine the geometry of the head region by fitting it to the head scan $\mathcal{P}_H$. In order to align the template model to the head scan, we find optimal scaling, rotation, and translation by minimizing squared distances between the detected 68 facial landmarks and their corresponding landmarks on the template model [35]. Afterwards, we further refine scaling, rotation, and translation through ICP [12]. In contrast to FGVH and due to our more noisy point clouds, we regularize the head fit by a 30-dimensional statistical head model derived from the publicly available data of [1]. After this coarse registration, we add fine-scale geometric detail by performing a nonrigid deformation that minimizes the objective function (1) restricted to the head region. The regularization term E_{reg} is the same as before. However, this time λ_{reg} is initially weighted by 1 and gradually decreased to 10^{-8}. We again solve the nonlinear least-squares problem using repeated block-coordinate descent.

After the fine-scale nonrigid registration, we adjust the skeletal joint positions through mean value coordinates and put the model into T-pose [2]. Finally, we add facial details (eyes and teeth) and reconstruct blendshapes. Following FGVH we adjust the template's teeth and eyes by optimizing for scaling, rotation, and translation based on the deformation of the mouth and eye region. To resolve occasional penetrations of eyes and eyelids we non-rigidly deform the eyelids to fit the transformed eye geometry. For reconstructing blendshapes, we map all blendshapes from the template mesh to the fitted model using deformation transfer [57].

3.5 Texture Generation

Given the camera images and the reconstructed avatar mesh, FGVH computes textures for the full-body scan and the face scan using Agisoft Metashape and blends them using Poisson Image Editing [51]. In our case, this approach leads to noticeable artifacts because of inaccuracies in the geometry reconstruction caused by inevitable motion during the capture process, as shown in Figure 8. We avoid these problems by computing the texture image through a graph cut optimization [17].

Using the fitted avatar mesh (Section 3.4) and the camera calibration data of Agisoft Metashape (Section 3.2), we generate partial textures by rendering the avatar mesh from each camera position. The projection from 3D world coordinates to the respective camera's image plane is modeled as a standard pinhole camera with Brown's distortion model [19], whose parameters are provided by Metashape's intrinsic and extrinsic camera calibration.

This projection is used to generate a partial texture $\mathbf{T}_i$ from each input image $\mathbf{I}_i$ in a two-pass rendering process implemented via OpenGL. In the first pass, all mesh triangles are projected onto the image plane of camera $\mathbf{c}_i$ and the resulting depth buffer is stored as $\mathbf{D}_i$. The second render pass generates the partial texture $\mathbf{T}_i$ by rendering the mesh onto the uv-layout of the template character. The fragment shader then (i) discards all fragments that do not pass the depth test $\mathbf{D}_i$ (i.e., that are not visible from camera $\mathbf{c}_i$) and (ii) computes the color value by accessing the camera image $\mathbf{I}_i$ at the texture coordinate $\mathbf{u}_j$ defined by projecting the interpolated surface point $\mathbf{p}_j$ onto the image plane of camera $\mathbf{c}_i$. We additionally compute the angle α between the surface normal and the viewing ray and additionally discard all fragments where α exceeds a threshold of 45° in order to rule out foreshortening effects. Color information for the remaining fragments is then written to the corresponding texture coordinate at $\mathbf{T}_i$. This rendering procedure results in a partial texture $\mathbf{T}_i$ and visibility map $\mathbf{V}_i$ (storing $\cos(\alpha)$ for each pixel) for every input image (see Figure 6).

Stitching the partial textures together could be done by simply performing a "best view" selection, i.e., coloring each texel from the partial texture where the corresponding surface patch was most orthogonal to the viewing vector. However, because of the motion during the scanning procedure, the reconstructed geometry is not accurate enough, and thus the partial textures do not align perfectly. Performing a best view selection would thus lead to noticeable seams between surface patches.

Graph cut methods [17] have been used successfully to seamlessly stitch together images or textures. We take inspiration from various works [6, 30, 42] and formulate our texture stitching as a

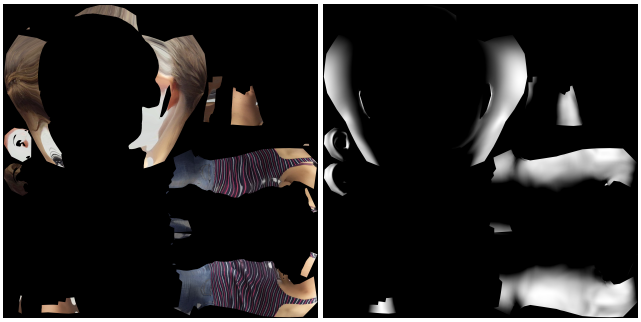

Figure 6: Partial texture (left) and visibility map (right).

Figure 7: The patches induced by the best view selection (left) and by our graph cut optimization (right). The latter leads to larger patches and fewer seams.

combinatorial optimization: Each of the F faces of the mesh is to be textured by one of the partial textures. This can be described by an index set $\mathcal{I} = \{l_1, \ldots, l_F\}$ with $l_i \in \{1, \ldots, N\}$, which labels each face with a partial texture index. The graph cut optimization then minimizes the error function

$$
\begin{aligned}
E_{\text{tex}}(\mathcal{I}) &= \sum_{i=1}^{F} D(f_i, l_i) \;+\; \sum_{i,j=1}^{F} S(f_i, f_j, l_i, l_j)\,, \\
D(f_i, l_i) &= \frac{1}{|\,\mathcal{U}(f_i)\,|} \sum_{\mathbf{u} \in \mathcal{U}(f_i)} \big(1 - \mathbf{V}_{l_i}(\mathbf{u})\big)\,, \\
S(f_i, f_j, l_i, l_j) &= \frac{1}{|\,\mathcal{U}(f_i, f_j)\,|} \sum_{\mathbf{u} \in \mathcal{U}(f_i, f_j)} \Big\| \mathbf{T}_{l_i}(\mathbf{u}) - \mathbf{T}_{l_j}(\mathbf{u}) \Big\|,
\end{aligned}
\tag{2}
$$

with a data term $D(f_i, l_i)$ and a smoothness term $S(f_i, f_j, l_i, l_j)$. The *data term* prefers to texture faces from input images where the face normal is parallel to the viewing vector, summing up the visibility map $\mathbf{V}_{l_i}$ for partial texture $\mathbf{T}_{l_i}$ over the set $\mathcal{U}(f_i)$ of texels of face f_i in uv-coordinates. The *smoothness term* ensures that neighboring faces are textured from images that avoid visible seams, by penalizing color differences on the texels of their shared edge $\mathcal{U}(f_i, f_j) = \mathcal{U}(f_i) \cap \mathcal{U}(f_j)$ in uv-coordinates.

We treat the objective function (2) as a multi-label graph cut optimization problem [16, 17, 40]. This defines a Markov Random Field that we optimize with the implementation provided by Szeliski et al. [58]. We initialize the optimization with the best view selection, which is a good starting point since it is equal to the minimum of the data term.

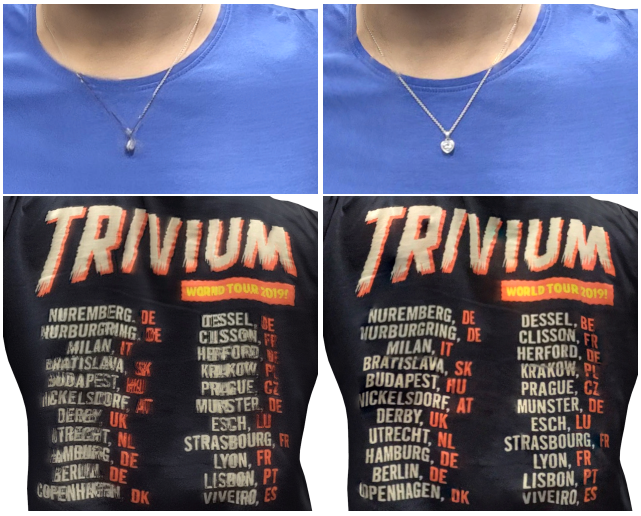

Figure 8: Texture generation of Agisoft Metashape (left) and our graph cut optimization (right), the latter yielding more detail on the necklace and the letters on the shirt.

The resulting labeling $\mathcal{I}$ defines which patches of the final texture are colored from which partial texture. Figure 7 shows the optimized labeling in comparison to the best view selection. Note that bigger parts of the texture are now textured from the same input image, which naturally reduces the amount of visible seams. There are, however, still some luminosity differences at the patch boundaries, which we eliminate by blending the patches using Poisson image editing [51]. Texture regions belonging to areas on the model that were not seen (e.g., the crotch region) are automatically filled by harmonic color interpolation.

This texture generation process is performed for both the full-body scan and the head scan. The head texture is then injected into the full-body texture using Poisson image editing in order to cope with illumination differences between the two scans. Since hands and eyes are typically not well scanned, their texture information is taken from the template texture and adapted to the scanned subject using histogram matching in CIELAB space [36].

As can be observed in Figure 8, the textures generated by our graph cut approach have more detail and are sharper compared to the textures generated by Agisoft Metashape.

4 RESULTS

Our avatar reconstruction takes about 20 minutes, measured on a desktop PC with 12 × 3.6 GHz Intel Xeon CPU and a Nvidia GTX 1080 Ti GPU, and consists of the following steps: capturing and transferring the videos (4 min), processing videos and generating point clouds (7 min), landmark detection and template fitting (2 min), and texture generation and merging (7 min).

In the following we provide quantitative comparisons with FGVH, which due to its extensive setup acts as an (approximate) ground truth, as well as qualitative comparisons to the monocular reconstructions of Alldieck et al. [5, 7].

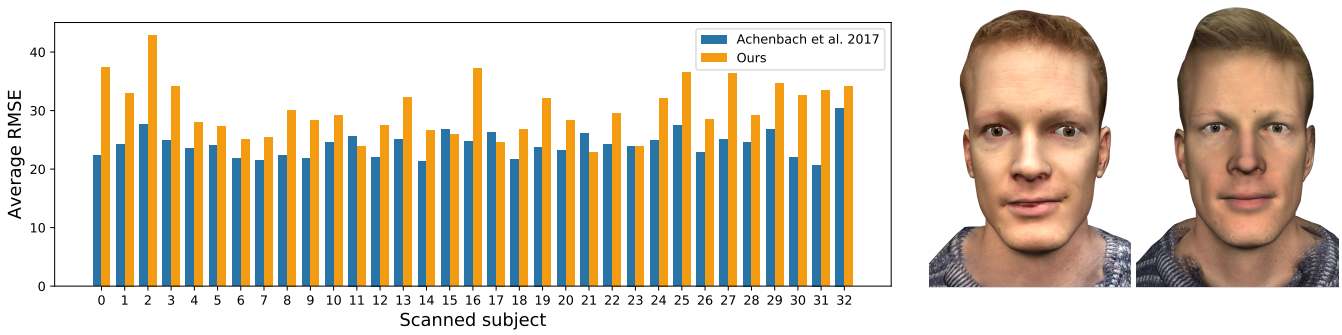

Figure 9: Root-mean-square reprojection errors of FGVH and our method over 33 reconstructed avatars. The close-up on the right shows the subject for which our method performs the worst compared to FGVH (Subject 2).

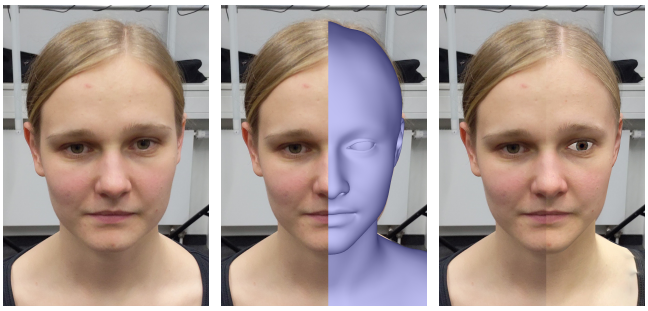

Figure 10: We evaluate the reprojection error in image space by rendering (without lighting) the reconstructed avatars from all input camera locations onto the input images.

Figure 11: Our avatars feature eyes, teeth, and facial blendshapes, and can thus be animated out-of-the-box, e.g., through real-time facial motion capturing.

In order to quantitatively compare our low-cost reconstruction with the multi-camera reconstruction of FGVH, we scanned and reconstructed 34 people with their method and ours. We had to discard the scan of one person, where the point cloud reconstruction failed due to dark clothing. For the remaining 33 scans we compare the reprojection error, which we compute by rendering for each input image the textured avatar from the corresponding camera location (see Figure 10) and computing the root-mean-square error (RMSE) over all rendered pixels in the CIELAB color space. Averaging the RMSE over all input images yields the reprojection error for one avatar reconstruction, which effectively measures reconstruction accuracy in both geometry and texture. Figure 9 shows the reprojection errors for all scanned subjects. Not surprisingly, the expensive camera rig of FGVH yields lower errors thanks to more accurate point clouds (cf. Figure 3). Although their RMSE ($\mu = 24.2$, $\sigma = 2.15$) is 20% lower than ours ($\mu = 30.3$, $\sigma = 4.77$), our hardware costs (about 600 EUR) are only 1% of theirs (about 60 kEUR). We note that Agisoft's texture generation leads to a slightly lower RMSE ($\mu = 29.6$, $\sigma = 4.47$), but our graph cut optimization yields perceptually superior results (Figure 8). As a purely geometric measure, the two-sided Hausdorff distance [24] between our reconstructions and the (approximate) ground truth given by FGVH is 7.1 mm on average, confirming that our avatars are quite accurate despite the low hardware requirements.

We qualitatively compare our method to the monocular avatar generation approaches of Alldieck and colleagues. The first method [5] reconstructs avatars from a video of a person turning around 360° in A-pose (taking around 2 h). The second method [7] requires only eight images of this 360° movement and generates the texture using the stitching technique of [6] (taking around 5 min). The input videos/images were taken using the same Google Pixel 3 to provide comparable input data. We used the original implementations provided by the authors, but doubled the default number of pose and shape estimation steps in Alldieck et al. [7] to achieve better results, as suggested to us by the authors. Figure 12 compares avatars reconstructed with Alldieck et al. [5], Alldieck et al. [7], FGVH, and our method, showing that our results are superior to Alldieck et al. and comparable to FGVH. Note that the avatars reconstructed by Alldieck et al. [5, 7] lack articulated hands, eyes, teeth, and facial blendshapes.

Our reconstructed avatars provide these facial animation controllers, as demonstrated in Figure 11 and the accompanying video. More results and comparisons, including dynamic skeletal and facial animations, can be found in the accompanying video.

Our method still has several limitations, as shown in Figure 13. First, the photogrammetry software cannot deal with very dark

Figure 12: Avatars of the same two persons reconstructed from different methods. From top to bottom row: [5], [7], our method and the expensive multi-camera setup of [2]. Note that our reconstruction improves on previous low-cost avatar generation pipelines in both geometry and texture.

Figure 13: Limitations of our method. Dark clothing (left) and movement during the capture process (center) is challenging for the multi-view stereo reconstruction. This leads to errors in geometry and texture. The landmark back-projection fails for point clouds with missing data (right).

clothing. Second, the point cloud quality degrades for body parts exhibiting noticeable movement during the capturing process. This is especially true for the arms, leading to a lower accuracy in geometry and texture reconstruction. Third, while the automatic 2D landmark detection worked robustly in all cases, the back-projection to the 3D failed for four subjects due to missing data in the point cloud. In these cases, the user was asked to manually select the landmarks. Finally, glasses, hair, and accessories are challenging for all photogrammetric approaches, including ours.

5 CONCLUSION

In this paper we presented a fully automated pipeline for generating high-quality virtual humans from monocular videos taken with a consumer smartphone, taking just about 20 minutes in total. Comparisons with both hardware-intensive and low-cost approaches show our virtual humans to be almost on par with the former while surpassing the latter. Our avatars are ready to be used in XR applications, as they allow skeletal and facial animation and are compatible with standard engines used in this field. This opens up the ability for the research community to work on high-quality avatars without extensive hardware setups.

For future work, we want to make our approach more robust to movements by segmenting the extracted video frames either into foreground/background or into semantic parts (e.g., torso, arms, legs, and head), which could potentially improve the quality of the multi-view stereo reconstruction. Furthermore, we plan to exploit the capabilities of smartphone APIs to build a designated application for controlling the capture process and gaining access to the intrinsic camera calibration. Another interesting direction is to scan challenging areas like the arms separately, i.e., to divide the capture process into more than two videos.

ACKNOWLEDGMENTS

The authors are very grateful to all scanned subjects. Additionally, we thank Niklas Krome for his work on the face tracking demo and Timo Menzel for his implementation of the texture histogram matching. This research was supported by the German Federal Ministry of Education and Research (BMBF) through the project ViTraS (ID 16SV8225)

REFERENCES

[1] Jascha Achenbach, Robert Brylka, Thomas Gietzen, Katja zum Hebel, Elmar Schömer, Ralf Schulze, Mario Botsch, and Ulrich Schwanecke. 2018. A Multilinear Model for Bidirectional Craniofacial Reconstruction. In *Proc. of Eurographics Workshop on Visual Computing for Biology and Medicine*. 67–76.

[2] Jascha Achenbach, Thomas Waltemate, Marc Erich Latoschik, and Mario Botsch. 2017. Fast Generation of Realistic Virtual Humans. In *Proc. of ACM Symposium on Virtual Reality Software and Technology*. 10.

[3] Jascha Achenbach, Eduard Zell, and Mario Botsch. 2015. Accurate Face Reconstruction through Anisotropic Fitting and Eye Correction. In *Proc. of Vision, Modeling & Visualization*. 1–8.

[4] Agisoft. 2020. Metashape Pro. http://www.agisoft.com/.

[5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018. Video Based Reconstruction of 3D People Models. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 8387–8397.

[6] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018. Detailed Human Avatars from Monocular Video. In *Proc. of International Conference on 3D Vision*. 98–109.

[7] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. 2019. Learning to Reconstruct People in Clothing from a Single RGB Camera. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 1175–1186.

[8] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. 2019. Tex2Shape: Detailed Full Human Body Geometry From a Single Image. In *Proc. of IEEE International Conference on Computer Vision*. 2293–2303.

[9] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: Shape Completion and Animation of People. *ACM Transactions on Computer Graphics* 24, 3 (2005), 408–416.

[10] Autodesk. 2014. Character Generator. https://charactergenerator.autodesk.com/.

[11] Amy L. Baylor. 2009. Promoting motivation with virtual agents and avatars: role of visual presence and appearance. *Philosophical Transactions of the Royal Society B: Biological Sciences* 364, 1535 (2009), 3559–3565.

[12] Paul J. Besl and Neil D. McKay. 1992. A Method for Registration of 3-D Shapes. *ACM Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256.

[13] Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. 2015. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *Proc. of IEEE International Conference on Computer Vision*. 2300–2308.

[14] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. 2016. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *Proc. of European Conference on Computer Vision (Lecture Notes in Computer Science)*. Springer International Publishing, 561–578.

[15] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. 2014. FAUST: Dataset and Evaluation for 3D Mesh Registration. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 3794–3801.

[16] Y. Boykov and Vladimir Kolmogorov. 2004. Experimental Comparison of Min-Cut/Max-Flow Algorithms for An Energy Minimization in Vision. *ACM Transactions on Pattern Analysis and Machine Intelligence* 26, 9 (2004), 1124–1137.

[17] Y. Boykov, O. Veksler, and R. Zabih. 2001. Fast Approximate Energy Minimization via Graph Cuts. *ACM Transactions on Pattern Analysis and Machine Intelligence* 23, 11 (2001), 1222–1239.

[18] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* 25 (2000), 120–125.

[19] Duane C. Brown. 1971. Close-range camera calibration. *Photogrammetric Engineering* 37, 8 (1971), 855–866.

[20] Samuel R. Buss. 2004. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation* 17 (2004).

[21] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[22] Michael Crichton. 1981. Looker. Movie.

[23] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. 2017. Motion2fusion: Real-Time Volumetric Performance Capture. *ACM Transactions on Computer Graphics* 36, 6 (2017), 246:1–246:16.

[24] M.-P. Dubuisson and A. K. Jain. 1994. A modified Hausdorff distance for object matching. In *Proceedings of International Conference on Pattern Recognition*. 566–568.

[25] Gareth Edwards. 2016. Rogue One: A Star Wars Story. Movie.

[26] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 2020. 3D Morphable Face Models – Past, Present and Future. *ACM Transactions on Computer Graphics* 39, 5 (2020).

[27] Xianyong Fang, Jikui Yang, Jie Rao, Linbo Wang, and Zhigang Deng. 2019. Single RGB-D Fitting: Total Human Modeling with an RGB-D Shot. In *Proc. of ACM Symposium on Virtual Reality Software and Technology*. 24:1–24:11.

[28] Gunnar Farnebäck. 2003. Two-Frame Motion Estimation Based on Polynomial Expansion. In *Image Analysis*, Josef Bigun and Tomas Gustavsson (Eds.). Springer, 363–370.

[29] Andrew Feng, Evan Suma Rosenberg, and Ari Shapiro. 2017. Just-in-time, viable, 3-D avatars from scans. *Computer Animation and Virtual Worlds* 28 (2017), 3–4.

[30] Ran Gal, Yonatan Wexler, Eyal Ofek, Hugues Hoppe, and Daniel Cohen-Or. 2010. Seamless Montage for Texturing Models. *Computer Graphics Forum* 29, 2 (2010), 479–486.

[31] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. DensePose: Dense Human Pose Estimation in the Wild. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 7297–7306.

[32] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. 2017. Real-Time Geometry, Albedo, and Motion Reconstruction Using a Single RGB-D Camera. *ACM Transactions on Computer Graphics* 36, 4 (2017), 44:1–44:13.

[33] Marc Habermann, Weipeng Xu, Michael Zollhöfer, Gerard Pons-Moll, and Christian Theobalt. 2019. LiveCap: Real-Time Human Performance Capture From Monocular Video. *ACM Transactions on Computer Graphics* 38, 2 (2019), 14:1–14:17.

[34] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and Hans-Peter Seidel. 2009. A statistical model of human pose and body shape. *Computer Graphics Forum* 28, 2 (2009), 337–346.

[35] Berthold K. P. Horn. 1987. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* 4, 4 (1987), 629–642.

[36] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM Transactions on Computer Graphics* 34, 4 (2015), 45:1–45:14.

[37] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. 2018. Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 8320–8329.

[38] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-end Recovery of Human Shape and Pose. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 7122–7131.

[39] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. 2020. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 5253–5263.

[40] V. Kolmogorov and R. Zabih. 2004. What Energy Functions can be Minimized via Graph Cuts? *ACM Transactions on Pattern Analysis and Machine Intelligence* 26, 2 (2004), 147–159.

[41] Marc Erich Latoschik, Daniel Roth, Dominik Gall, Jascha Achenbach, Thomas Waltemate, and Mario Botsch. 2017. The Effect of Avatar Realism in Immersive Social Virtual Realities. In *Proc. of ACM Symposium on Virtual Reality Software and Technology*. 39:1–39:10.

[42] Victor S. Lempitsky and Denis V. Ivanov. 2007. Seamless Mosaicing of Image-Based Texture Maps. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 1–6.

[43] Yunpeng Liu, Stephan Beck, Renfang Wang, Jin Li, Huixia Xu, Shijie Yao, Xiaopeng Tong, and Bernd Froehlich. 2015. Hybrid Lossless-Lossy Compression for Real-Time Depth-Sensor Streams in 3D Telepresence Applications. In *Advances in Multimedia Information Processing – PCM 2015*. Springer International Publishing, 442–452.

[44] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-person Linear Model. *ACM Transactions on Computer Graphics* 34, 6 (2015), 248:1–248:16.

[45] Nadia Magnenat-Thalmann and Daniel Thalmann. 1987. The Direction of Synthetic Actors in the film Rendez-vous à Montréal. *IEEE Computer Graphics and applications* 7, 12 (1987), 9–19.

[46] Nadia Magnenat-Thalmann and Daniel Thalmann. 2005. Virtual humans: thirty years of research, what next? *The Visual Computer* 21, 12 (2005), 997–1015.

[47] Charles Malleson, Maggie Kosek, Martin Klaudiny, Ivan Huerta, Jean-Charles Bazin, Alexander Sorkine-Hornung, Mark Mine, and Kenny Mitchell. 2017. Rapid one-shot acquisition of dynamic VR avatars. In *Proc. of IEEE Virtual Reality*. 131–140.

[48] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. 2018. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *Proc. of International Conference on 3D Vision*. 484–494.

[49] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body From a Single Image. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 10975–10985.

[50] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martínez, and Joaquín Fernández-Valdivia. 2000. Diatom Autofocusing in Brightfield Microscopy: a Comparative Study. In *Proc. of International Conference on Pattern Recognition*. 3318–3321.

[51] Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson Image Editing. *ACM Transactions on Computer Graphics* 22, 3 (2003), 313–318.

[52] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. 2017. Building Statistical Shape Spaces for 3D Human Modeling. *Pattern Recognition* 67, C (2017), 276–286.

[53] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. 2015. Dyna: A Model of Dynamic Human Shape in Motion. *ACM Transactions on Computer Graphics* 34, 4 (2015), 14.

[54] K. M. Robinette, H. Daanen, and E. Paquet. 1999. The Caesar Project: A 3-D Surface Anthropometry Survey. In *Proc. of the 2nd International Conference on 3-D Digital Imaging and Modeling*. IEEE Computer Society, 380–386.

[55] Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Computer Graphics* 36, 6 (2017), 245:1–245:17.

[56] Daniel Roth, Jan-Philipp Stauffert, and Marc Erich Latoschik. 2019. *VR Developer Gems*. Vol. 1. Springer US, Chapter Avatar Embodiment, Behavior Replication, and Kinematics in Virtual Reality, 321–348.

[57] Robert W. Sumner and Jovan Popović. 2004. Deformation Transfer for Triangle Meshes. *ACM Transactions on Computer Graphics* 23, 3 (2004), 399–405.

[58] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. 2006. A Comparative Study of Energy Minimization Methods for Markov Random Fields. In *Proc. of European Conference on Computer Vision*. 16–29.

[59] Thomas Waltemate, Dominik Gall, Daniel Roth, Mario Botsch, and Marc Erich Latoschik. 2018. The Impact of Avatar Personalization and Immersion on Virtual Body Ownership, Presence, and Emotional Response. *IEEE Transaction on Visualization and Computer Graphics* 24, 4 (2018), 1643–1652.

[60] Katja Zibrek, Seán Martin, and Rachel McDonnell. 2019. Is Photorealism Important for Perception of Expressive Virtual Humans in Virtual Reality? *ACM Transactions on Applied Perception* 16 (2019), 1–19.

[61] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. 2018. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. *Computer Graphics Forum* 37, 2 (2018), 523–550.