Fast Generation of Realistic Virtual Humans

Jascha Achenbach
Computer Graphics Group, Bielefeld University

Thomas Waltemate
Computer Graphics Group, Bielefeld University

Marc Erich Latoschik
HCI Group, Würzburg University

Mario Botsch
Computer Graphics Group, Bielefeld University

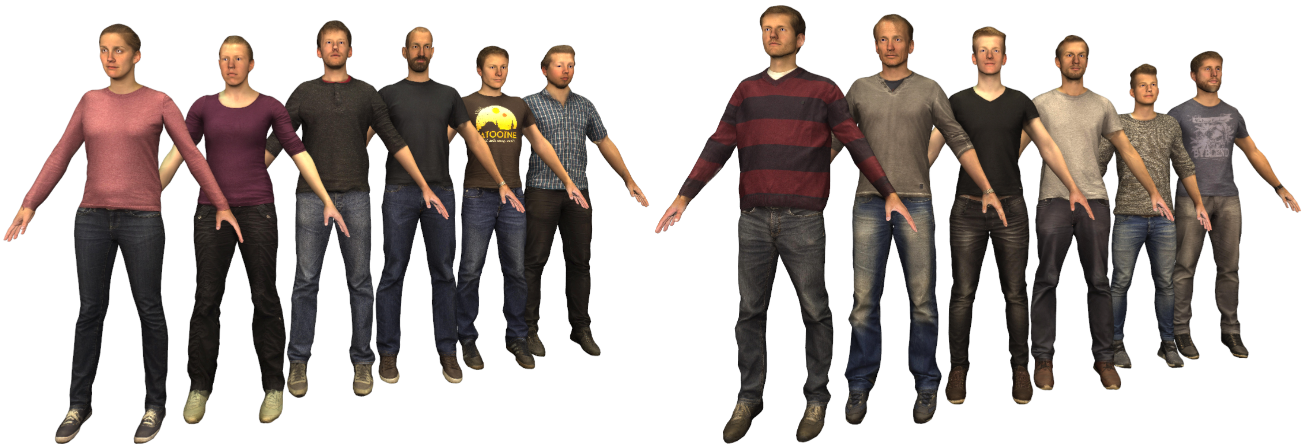

Figure 1: We generate realistic virtual humans from real persons through multi-view stereo scanning. The resulting characters are ready to be animated through a skeletal rig and facial blendshapes, and are compatible with standard graphics and VR engines. The whole reconstruction process requires only minimal user input and takes less than ten minutes.

ABSTRACT

In this paper we present a complete pipeline to create ready-to-animate virtual humans by fitting a template character to a point set obtained by scanning a real person using multi-view stereo reconstruction. Our virtual humans are built upon a holistic character model and feature a detailed skeleton, fingers, eyes, teeth, and a rich set of facial blendshapes. Furthermore, due to the careful selection of techniques and technology, our reconstructed humans are quite realistic in terms of both geometry and texture. Since we represent our models as single-layer triangle meshes and animate them through standard skeleton-based skinning and facial blendshapes, our characters can be used in standard VR engines out of the box. By optimizing for computation time and minimizing manual intervention, our reconstruction pipeline is capable of processing whole characters in less than ten minutes.

CCS CONCEPTS

• **Computing methodologies** → **Mesh geometry models**; *Appearance and texture representations*;

KEYWORDS

virtual characters, virtual humans, avatars, 3D scanning

ACM Reference format:
Jascha Achenbach, Thomas Waltemate, Marc Erich Latoschik, and Mario Botsch. 2017. Fast Generation of Realistic Virtual Humans. In *Proceedings of VRST'17, Gothenburg, Sweden, November 8–10, 2017,* 10 pages.
https://doi.org/10.1145/3139131.3139154

1 INTRODUCTION

Today, virtual characters are widely used for applications ranging from computer games, special effects in movies, virtual try-on, to medical surgery planning and virtual assistance. Virtual characters are especially important for Virtual Reality (VR) for both virtual agents simulated by artificial intelligence as well as avatars, the digital alter-egos of the users in the virtual worlds. Immersive embodied scenarios provide ample possibilities to study psychophysical effects caused by modifying avatar appearance and hence, e.g., altering self-perception or body ownership [Banakou and Slater 2014; González-Franco et al. 2010; Latoschik et al. 2017, 2016; Lugrin et al. 2015; Peck et al. 2013; Roth et al. 2016; Slater et al. 2010] are therefore a common and interesting topic in VR research.

Today, 3D-scanning of real humans is a prominent technique to generate virtual humans. Striving for realism and human-like appearance requires geometrically accurate meshes and detailed textures, and the application of the resulting models in interactive scenarios requires them to be animated: Their full-body posture,

VRST'17, November 8–10, 2017, Gothenburg, Sweden

ACM ISBN 978-1-4503-5548-3/17/11.
https://doi.org/10.1145/3139131.3139154

hand posture, eye gaze, and facial expressions have to be controllable through suitable skeletal rigs and blendshapes, respectively. To be widely employable, the resulting character models should be compatible with standard game engines or VR frameworks, and the overall avatar creation should ideally be fast enough to be performed during rapid prototyping or empirical studies.

However, despite the increasing availability of scanning technologies and the large body of research on 3D-scanning and mesh reconstruction in both computer vision and computer graphics, creating believable and ready-to-animate virtual humans in a short amount of time is still a challenging problem. Existing approaches reconstruct static full-body "selfies" [Li et al. 2013] without animation controls, or full-body models without controls for hands or facial expressions [Bogo et al. 2015, 2014], or head models for facial puppetry without a full-body [Cao et al. 2014a; Weise et al. 2011]. Approaches for the fast generation of characters with all required animation controls are mostly lacking. In addition, many approaches focus on geometry reconstruction only, and neglect the generation of high quality textures from scanner input.

In this paper we present a complete character generation pipeline that is able to digitally clone a real person into a realistic high-quality virtual human, which can then be used for animation and visualization in any standard graphics or VR engine. The whole reconstruction process requires only a minimum amount of user interaction and takes less than ten minutes on a desktop PC.

For 3D-scanning we employ a custom-built camera rig with 40 cameras for the body and 8 cameras for the face, and compute dense point clouds through multi-view stereo reconstruction. In order to robustly deal with noise and missing data, and to avoid character rigging in a post-process, we fit a generic human body model to the user's scanner data. In particular, we build upon the template model from Autodesk's Character Generator, which is already equipped with a detailed skeleton and skinning weights, a rich set of blendshapes, as well as eyes and teeth. This template model is further enriched by statistical data on human body shapes, which yields a prior for the template fitting process. By fitting the template geometry to the scanner data and transferring eyes, teeth, skeleton, and blendshapes to the morphed template, our reconstructed models are ready to be animated.

By construction, all our reconstructed characters share the tessellation of the template model. Hence they are in dense one-to-one correspondence, which allows to transfer properties between models. As one application example, we exploit this fact by scanning subjects with and without clothing, and then storing the clothes, i.e., the difference between the two models. This allows us to easily and seamlessly transfer clothing from one character to another. This largely reduces potential confounds caused by different cloths of different avatars used, e.g., in perception studies. To keep our models simple and compatible to any standard rendering engine, and to enable highly efficient character animation, we represent our characters by a single-layer mesh and employ standard skinning and blendshapes for body and face animation, respectively.

Overall, our contributions enable the generation of realistic and fully animatable virtual humans in just a couple of minutes, which makes them accessible to a wide range of VR experiments, where they can be used as avatars or conversational agents.

2 RELATED WORK

Due to the increasing availability of 3D-scanning solutions and the growing demand for virtual human models, there is a huge body of literature on scanning, reconstructing, and animating virtual characters. Due to space constraints we restrict to the approaches most relevant to ours, beginning with techniques for reconstructing full body models, followed by face capturing methods, and finally discussing approaches for reconstructing animatable VR characters.

2.1 Full-Body Reconstruction

Several methods employ affordable RGBD sensors (e.g., Kinect) for scanning and reconstructing human bodies [Feng et al. 2014; Li et al. 2013; Sturm et al. 2013; Tong et al. 2012]. However, due to the coarse and noisy data delivered by these sensors, their character reconstructions are bound to a rather low quality. Since our goal is reconstructing realistic high quality virtual humans, we instead base our framework on a multi-camera rig that can capture a subject in a fraction of a second. Using multi-view stereo we then reconstruct a dense point cloud from the camera data.

This point cloud could then be fed into a surface reconstruction method, followed by an auto-rigging process for embedding a control skeleton and defining skinning weights [Baran and Popović 2007; Feng et al. 2015]. However, the surface reconstruction might fail to faithfully capture delicate features (e.g., fingers), causing the auto-rigging to fail. We therefore use a fully-rigged template model that we fit to the scanner data using non-rigid registration.

Fitting a template model to a large amount of training data allows to build a statistical model, which can act as prior when fitting the template to scanner data. The SCAPE model [Anguelov et al. 2005] is one of the first, most prominent, and most frequently employed human body models. It has been extended in many ways [Bogo et al. 2014; Hirshberg et al. 2012; Pishchulin et al. 2017; Sigal et al. 2007; Straka et al. 2012], which have been applied in different scenarios ranging from breathing animation [Tsoli et al. 2014], over soft-tissue animation [Loper et al. 2014], to estimation of shape and posture from either a single image [Guan et al. 2009] or from RGBD sequences [Bogo et al. 2015; Weiss et al. 2011].

Many other statistical human body models have been proposed [Allen et al. 2003, 2006; Hasler et al. 2009; Loper et al. 2015; Wuhrer et al. 2014], which can be roughly classified as triangle-based or vertex-based methods, depending on how they model posture articulation and fine-scale deformation. Triangle-based methods have to solve a linear Poisson system to compute the deformed vertex positions, and are therefore incompatible to standard graphics engines. In contrast, models based on per-vertex linear blend skinning, such as, e.g., SMPL [Loper et al. 2015] or S-SCAPE [Pishchulin et al. 2017], can readily be used in such engines. We therefore also base our model on vertex-based linear blend skinning. However, in comparison to SMPL and S-SCAPE our model has a higher geometric resolution and provides fine-scale details, such as fingers, eyes, and teeth. Furthermore, it is equipped with a more detailed skeleton and allows for hand and face animation.

In order to place the skeleton within the model shape, SMPL learns a joint regressor from a large amount of data, which then represents joint positions as a linear function of the model's shape.

Since our skeleton is more detailed than that of SMPL and the training data is not available, we cannot use their regressor. Instead, we follow Feng et al. [2015] and represent the joint positions as generalized barycentric combinations of the template's vertex positions, which also is a linear function.

While the above methods work well for reconstructing the *geometry* of human bodies, they mostly neglect the texture reconstruction, which however is crucial for VR applications. In contrast, we reconstruct a high-quality texture from the reconstructed geometry and the individual camera images of our scanner.

2.2 Face Reconstruction

There is a lot of work dedicated to face reconstruction from images, video, RGBD data, laser scans, or multi-view stereo. The pioneering work of Blanz and Vetter [1999] first proposed a PCA-based statistical face model for reconstructing face models from 3D scanner data or 2D photographs. Similar to body reconstruction, most approaches employ a statistical model as a regularizing prior.

Many approaches use an RGBD sensor to reconstruct face models [Cao et al. 2014b; Liang et al. 2014] and/or to animate them based on captured performance data [Bouaziz et al. 2013; Hsieh et al. 2015; Thies et al. 2015]. However, their face reconstructions suffer from low quality in geometry and texture, due to the inherent limitations of current RGBD sensors. High quality face reconstructions can be achieved through multi-camera rigs and multi-view stereo reconstruction [Achenbach et al. 2015; Beeler et al. 2010; Ghosh et al. 2011]. However, these approaches aim at a *static* high quality reconstruction and do not provide ready-to-animate models. Other works use video input to generate *dynamic* face models, which are subsequently animated based on the video stream [Garrido et al. 2016; Shi et al. 2014; Thies et al. 2016; Wu et al. 2016]. Ichim et al. [2015] proposed a method for creating a textured 3D face rig from picture and video input taken on a cell-phone. Since we aim at high quality geometry and texture, but also at short acquisition time, we employ multi-view face scanning based on [Achenbach et al. 2015]. We take the deformed template model, which was previously fit to the full-body scan, and refine its face region by fitting it to the point cloud resulting from the face scan.

Dynamic facial animations are crucial for VR characters, e.g. for speech animation or emotional facial expressions. With the industry standard being linear blendshape models [Lewis et al. 2014], the character generation pipeline also has to construct the required set of FACS blendshapes [Ekman and Friesen 1978]. For high quality production without time constraints, these blendshapes are often created manually by artists or reconstructed by scanning real actors performing these expressions [Alexander et al. 2009]. A faster process is enabled by example-based facial rigging [Li et al. 2010], which generates personalized facial blendshapes from a small set of example expressions. Since we want to keep acquisition and processing time low, we scan the actor in neutral expression only, and generate the full set of FACS blendshapes by adjusting the template's generic blendshapes to the deformed model using deformation transfer [Sumner and Popović 2004]. If acquisition and processing time is not that critical, reconstructing a few additional expressions and using example-based facial rigging would be a good compromise.

2.3 Avatar Reconstruction

While there are many approaches for reconstructing human body shapes *or* human faces *or* human hands, only few previous works aim at reconstructing a complete virtual human featuring animatable body, face, *and* hands.

Malleson et al. [2017] present a single snapshot system for rapid acquisition of animatable, full-body avatars based on an RGBD sensors. While the total processing time is in the order of seconds, the body is a stylized astronaut character that roughly fits the body dimensions only. Albeit face shape and texture are also considered, the results are of rather low quality and lack facial details, as only a low-dimensional face space is considered for fitting.

Feng et al. [2017] present a system for generating virtual characters by scanning a human subject. Their model is equipped with a full-body skeleton rig and is capable of facial expressions and finger movements. In direct comparison, their reconstruction process takes about twice as long as ours and requires more manual effort. Blendshapes are generated by explicitly scanning the actor in five different expressions, restricting the model to a few, but nicely personalized blendshapes. In contrast, our method reconstructs the full set of FACS blendshapes from a single face scan in neutral pose, and thus is compatible with standard animation packages. On the downside, our blendshapes are more generic and not as actor-specific. The biggest drawback of Feng's method is that by construction each model has a different tessellation, which prevents statistical analysis and detail/cloth transfer between models. In contrast, all our models share the tessellation of the initial template mesh.

3 INPUT DATA

Our 3D-scanning setup is based on multi-view stereo reconstruction using a single-shot multi-camera rig, since this minimizes acquisition time to a fraction of a second, while at the same time providing high quality results in terms of geometry and texture. We built a full-body scanner and a separate face scanner, consisting of 40 and 8 DSLR cameras, respectively, as shown in Figure 2. The cameras of each scanner are triggered simultaneously and the resulting pictures are subsequently downloaded from the cameras. We decided for a separate face scanner, in contrast to augmenting the full-body scanner with more cameras aiming at the face region, since otherwise the face cameras had to be manually adjusted to the individual subjects' heights.

The images of the 40 body cameras and of the 8 face cameras are automatically passed to the commercial software Agisoft Photoscan Pro, which computes two high-resolution point sets $\mathcal{P}_B$ of the body and $\mathcal{P}_F$ of the face, as well as camera calibration data. Face scans usually consist of about 1M points, body scans of about 3M points. Since the template mesh has a limited resolution of 21k vertices, we uniformly sub-sample the two point sets to 40k and 80k points, respectively. This sampling resolution is chosen such that the resulting point density is still about twice as high as the vertex density of the template mesh. This speeds up the fitting process significantly without noticeably sacrificing geometric fidelity. When it is clear from the context we omit the index B and F, and just write $\mathcal{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_N)$. Note that each point $\mathbf{p}_j$ is equipped with a normal vector $\mathbf{n}_j$ and RGB colors $\mathbf{c}_j$.

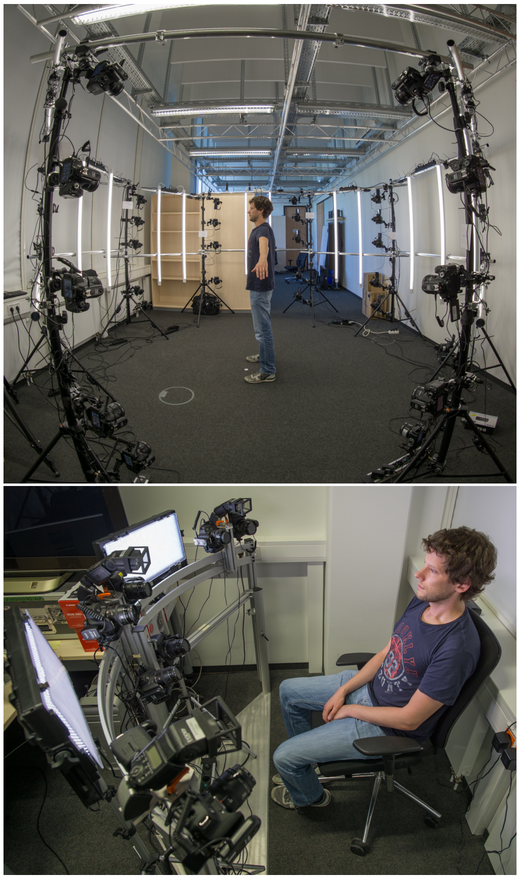

Figure 2: Our custom-built full-body scanner (left) and face scanner (right) are both based on multi-view stereo and consist of 40 and 8 DSLR cameras, respectively.

Since the bottom of the feet is not visible for the full-body scanner, these regions cannot be captured properly. The missing points below the feet can easily result in an erroneous fitting of the feet regions. In contrast, the floor around the feet is usually scanned quite well. We exploit this by detecting the floor plane and removing its points from the point cloud $\mathcal{P}_B$. We then we uniformly sample the detected floor plane underneath the feet region. This proved to be effective to capture the real extent of the feet and keep the feet on the floor during fitting without special treatment.

As a template model we picked a character from Autodesk's Character Generator [Autodesk 2014], because these characters are already equipped with facial blendshapes, eyes and teeth, and a skeleton with corresponding skinning weights. However, any other template model with skeleton and blendshapes would work as well. The template mesh consists of $n \approx 21$k vertices with positions $\mathcal{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$. A bar denotes vertex positions in the undeformed state: $\bar{\mathcal{X}} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

In order to incorporate prior knowledge on human body shapes into the reconstruction process, we integrate shapes from multiple data bases by fitting our template character to their registered body models. From the FAUST database [Bogo et al. 2014] we used 10 scans of different subjects standing in A-pose, and we included 111 scans from [Hasler et al. 2009]. Moreover, we added 82 synthetic models with different shapes from Autodesk's Character Generator. After fitting our template model to these models, they all share the same tessellation, allowing us to compute a ten-dimensional PCA subspace based on vertex positions of posture-normalized characters in T-pose. This PCA will act as initialization and regularization for the body fitting described in the next section.

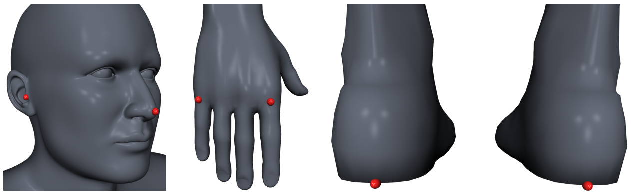

Figure 3: Nine landmarks are selected manually on the full-body point set, whose (pre-selected) counterpart vertices on the template model are shown here.

4 BODY RECONSTRUCTION

After computing and post-processing the point cloud $\mathcal{P}_B$ of the full-body scan, the next step is to align and fit the template model to this point set. As in most template fitting approaches, this fit is robustly performed in several steps: In the initialization phase, we optimize the alignment (scaling, rotation, translation), pose (skeleton joint angles), and PCA parameters for the ten-dimensional shape space. Afterwards, a fine-scale deformation fits the model to the data. Once the geometry fit is done, we have to compute texture, correct joint positions, and pose-normalize the model.

4.1 Initialization

Initially, the point set $\mathcal{P}_B$ and the template are in different coordinate systems and have different poses, since the template is in T-pose and the body scan is performed in A-pose. To bootstrap the template fitting procedure, we manually select nine landmarks $\mathcal{L}$ on the point-set $\mathcal{P}_B$, whose corresponding vertices on the template model have been pre-selected (see Figure 3). The landmarks have been chosen to ensure that important body parts like head, hands, and feet are fitted properly.

In the first step we optimize the alignment and pose of the template model in order to minimize the squared distances between these nine landmarks on the template model and their corresponding landmarks in the point set. To this end, we alternatingly compute (a) the optimal scaling, rotation, and translation [Horn 1987] and (b) optimize the joint angles using inverse kinematics based on linear blend skinning [Buss 2004]. This procedure is iterated until the relative change of the squared distances falls below 0.05. This initialization process is depicted in Figure 4, (a) and (b).

The landmark-based fit gives us a good estimate of scaling, rotation, translation, and joint angles. We further optimize these variables by additionally taking closest point correspondences into account, which are computed by finding, for each point $\mathcal{P}_B$, its closest point on the template. We prefer these scan-to-model correspondences over model-to-scan correspondences, since they were

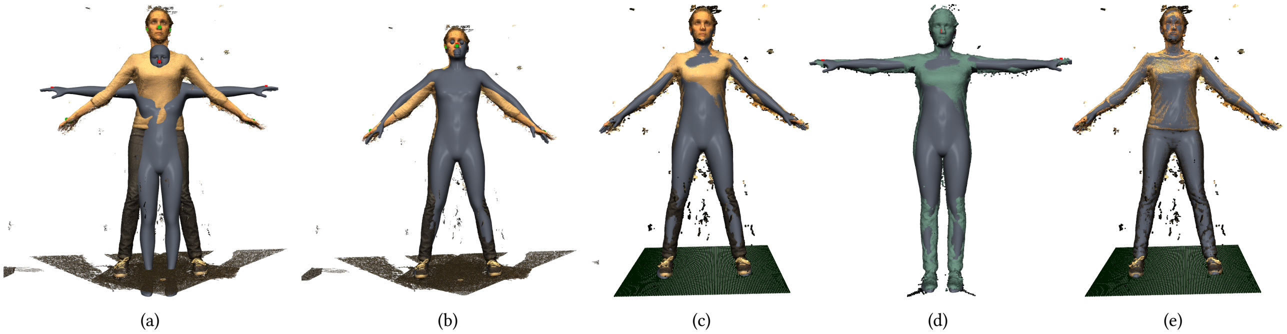

Figure 4: We first optimize alignment (scaling, rotation, translation: (a)) and pose (joint angles: (b)) based on nine manually selected landmarks. This fit is refined by incorporating closest point correspondences (c) and by alternating with PCA regularization in T-pose (d). After this initialization, we perform a fine-scale deformation to the point set (e).

shown to yield more accurate fits [Achenbach et al. 2015]. As usually done in ICP-based registrations, we prune unreliable correspondences based on distances and normal deviations. We employ the same alternating optimization as before to optimize alignment and pose, this time minimizing squared distances of landmarks and of correspondences (Figure 4(c)).

After convergence of the alignment and pose optimization, we add the PCA weights to the active variables and thereby optimize the geometric shape in the ten-dimensional PCA space, again by minimizing squared distances between landmarks and correspondences. As our PCA model is pose-normalized in T-pose, the PCA-fitting is performed in T-pose (see Figure 4(d)), and is alternated with alignment and pose optimization. The shape change caused by adjusting PCA parameters requires adjusting the skeleton's joint positions. To this end, we represent joint positions by mean value coordinates [Ju et al. 2005] with respect to the vertex positions of the template mesh. Joint positions are then a linear function of vertex positions, and hence also a linear function of PCA parameters. Two iterations of this procedure are usually sufficient for a good initial fit of shape and pose.

4.2 Deformable Registration

With the point set and template model in good initial alignment we perform a fine-scale non-rigid registration, following the approach of [Achenbach et al. 2015]. To this end, we minimize the energy

$$E_{\text{body}}(\mathcal{X}) = \lambda_{\text{lm}} E_{\text{lm}}(\mathcal{X}) + \lambda_{\text{fit}} E_{\text{fit}}(\mathcal{X}) + \lambda_{\text{reg}} E_{\text{reg}}(\mathcal{X}, \bar{\mathcal{X}}) , \qquad (1)$$

where the three energy terms are explained below.

The *landmark term* E_{lm} penalizes the squared distance between the nine manually selected landmarks $\mathbf{p}_l$, $l \in \mathcal{L}$, in the point set and their counterpart vertices $\mathbf{x}_l$ on the template model

$$E_{\text{lm}}(\mathcal{X}) = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \|\mathbf{x}_l - \mathbf{p}_l\|^2 . \qquad (2)$$

The *fitting term* E_{fit} penalizes the squared distance between corresponding points $\mathbf{x}_c$ and $\mathbf{p}_c$

$$E_{\text{fit}}(\mathcal{X}) = \frac{1}{\sum_{c \in C} w_c} \sum_{c \in C} w_c \|\mathbf{x}_c - \mathbf{p}_c\|^2 , \qquad (3)$$

where C is the set of closest point correspondences and w_c are per-vertex weights as discussed below. The closest points $\mathbf{x}_c$ are expressed as barycentric combinations of the template vertices $\mathbf{x}_i$.

The *regularization term* E_{reg} penalizes the geometric distortion from the undeformed model $\bar{\mathcal{X}}$ (the result of the initialization phase of Section 4.1) to the deformed state $\mathcal{X}$, measured by the squared deviation of the per-edge Laplacians

$$E_{\text{reg}}(\mathcal{X}, \bar{\mathcal{X}}) = \frac{1}{\sum_e A_e} \sum_{e \in \mathcal{E}} A_e \left\| \Delta^e \mathbf{x}(e) - \mathbf{R}_e \Delta^e \bar{\mathbf{x}}(e) \right\|^2 . \qquad (4)$$

Here, A_e is the area associated to edge e, and $\mathbf{R}_e$ are per-edge rotations to best-fit deformed and undeformed Laplacians (see [Achenbach et al. 2015] for details). We prefer the edge-based Laplacian over the standard vertex-based Laplacian, since in our experiments it converges slightly faster to very similar results.

The three coefficients λ_{lm}, λ_{fit}, and λ_{reg} are used to guide the iterative fitting procedure, where the surface stiffness is controlled by λ_{reg}. In the beginning, only the manually specified (hence quite reliable) landmarks are taken into account, using $\lambda_{\text{reg}} = 1$, $\lambda_{\text{lm}} = 1$ and $\lambda_{\text{fit}} = 0$. We then gradually decrease λ_{reg} after each iteration until $\lambda_{\text{reg}} = 10^{-5}$. After these iterations, the template is sufficiently well aligned to yield reliable closest point correspondences. We therefore continue with $\lambda_{\text{reg}} = 10^{-5}$ and $\lambda_{\text{lm}} = 1$, but additionally set $\lambda_{\text{fit}} = 1$ to also consider E_{fit}. Then, both λ_{lm} and λ_{reg} are gradually decreased until $\lambda_{\text{reg}} = 10^{-9}$.

During the fitting procedure we weight down parts of the template using the per-vertex weights w_c in E_{fit} in order to prevent unreliably scanned regions from being fitted to strongly (see Figure 5). We weight down the hands, since they are usually not scanned well, and the face region to allow us to add more detail when combining with the face scan in Section 5.

The nonlinear objective function (1) is minimized by solving for vertex positions $\mathbf{x}_i$ and per-edge rotations $\mathbf{R}_e$ using alternating optimization (a.k.a. block coordinate descent) [Achenbach et al. 2015; Bouaziz et al. 2014]. Figure 4(e) shows the final result of the body fitting procedure.

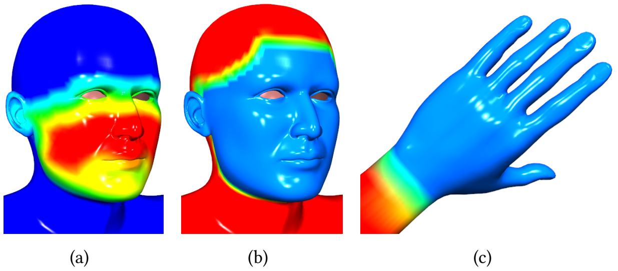

Figure 5: Per-vertex weights in the fitting energy allow to fit only the face region (a) or only the body (b), and to down-weight (typically poorly scanned) hands (c).

4.3 Texture Reconstruction

After the coarse scale initialization and the fine-scale non-rigid registration, the template has been accurately aligned and deformed to fit the point cloud of the body scan. We pass the deformed template model to Agisoft Photoscan Pro, which makes use of the existing texture layout from Autodesk's Character Generator and computes a high-quality 4k×4k texture based on the 40 camera images and their calibration data (see Figure 6(a)).

Since the camera images typically do not provide meaningful texture information for eyes and teeth, we use a pre-selected image mask to preserve the corresponding texture regions, i.e., to use eye and teeth texture from the generic template texture.

Due to occlusions and delicate geometric structures, scanning artifacts can easily occur for the fingers, which can result in an inaccurate template fit and then to misaligned textures for the fingers. We reconstruct a plausible hand texture by searching for the best-matching hand texture in Autodesk's Character Generator and using this hand texture instead. We identify the best-matching texture based on the Euclidean distance between RGB values of the back of both hands, the Autodesk texture and the one of the scanned subject (the latter is fitted reliable due to the manually selected landmark on the hands). Here, it turned out beneficial to distinguish between male and female hand textures. The found hand texture area is then seamlessly merged into the reconstructed full-body texture using Poisson image editing [Pérez et al. 2003].

Finally, the texture area below the armpits is typically corrupt as these are not sufficiently visible from our cameras. We smoothly fill these texture regions by harmonic color interpolation, which we compute by solving a sparse linear Laplace system with suitable Dirichlet color boundary constraints, similar in concept to Poisson image editing [Pérez et al. 2003].

4.4 Pose Normalization

Due to the non-rigid shape deformation the template's joints are not at their correct positions anymore. We again adjust the joint positions based on the precomputed mean value coordinates, this time representing the joint positions as a linear function of vertex positions (instead of PCA parameters). Employing mean value coordinates for this mapping ensures that joints are placed at meaningful positions even for strong shape deformations.

After mapping the skeleton to the deformed template (in scan pose), we use it to undo the pose fitting, i.e., to put the model into T-pose, as it is usually required by animation tools. In particular for character animation via motion capturing this is an important step, since these systems usually rely on a standardized T-pose as initialization. To make sure that both feet of the resulting character are standing exactly on the floor after pose-normalization, we first rigidly translate the model to put the (pre-selected) sole vertices onto the floor and then non-rigidly deform them onto the floor plane, while allowing only the feet to slightly deform, regularized by the Laplacian energy (4).

5 FACE RECONSTRUCTION

After fitting the template model to the full-body scan $\mathcal{P}_B$, we now improve the geometry and texture of its facial region by fitting it to the face scan $\mathcal{P}_F$ and exploiting its eight close-up camera images. We closely follow the face reconstruction approach of [Achenbach et al. 2015], but adjust it to the combined body-and-face reconstruction setup and extend it by blendshape reconstruction.

5.1 Initialization

Since the face scan and the body scan are not aligned to each other, the template model is not aligned to the face scan either.

Following [Achenbach et al. 2015], we automatically detect facial landmarks in the input camera images using [Asthana et al. 2013], which are then mapped to 3D points in $\mathcal{P}_F$ using the camera calibration data. We transform the template to the face scan by finding optimal scale, rotation, and translation to minimize squared distances between the detected 3D facial landmarks and their (pre-selected) counterparts on the template model. Afterwards, we refine scale, rotation, and translation by iteratively finding closest point correspondences and computing the optimal similarity transformation in the usual ICP manner [Horn 1987]. Note that we transform the whole full-body template, but based on landmarks and correspondences of the face scan $\mathcal{P}_F$ only.

5.2 Deformable Registration

After the initialization the template model and the facial point set $\mathcal{P}_F$ are sufficiently well aligned to start the fine-scale non-rigid deformation. To this end we minimize the energy

$$\begin{aligned} E_{\text{face}}(\mathcal{X}) \;=\; & \lambda_{\text{lm}} E_{\text{lm}}(\mathcal{X}) + \lambda_{\text{fit}} E_{\text{fit}}(\mathcal{X}) + \\ & \lambda_{\text{reg}} E_{\text{reg}}(\mathcal{X}, \bar{\mathcal{X}}) + \lambda_{\text{mouth}} E_{\text{mouth}}(\mathcal{X}) \,. \end{aligned} \tag{5}$$

Here E_{fit} again represents closest point correspondences and is weighted by $\lambda_{\text{fit}} = 1$. We again employ per-vertex weighting in the fitting term E_{fit}, such that only the face and ear vertices are dragged toward the face scan (see Figure 5(a)).

E_{lm} represents a landmark term, weighted by $\lambda_{\text{lm}} = 1$, and includes three types of landmarks: Besides the automatically detected facial features, we manually pick two landmarks on each ear to more precisely fit the ears. Furthermore, we manually pick seven contour points for each eye in the frontal face picture and compute landmarks for eye lid reconstruction (see [Achenbach et al. 2015]).

The regularization term E_{reg} is the same as for the body fitting. It is initially weighted by $\lambda_{\text{reg}} = 1$ and is gradually decreased to $\lambda_{\text{reg}} = 10^{-9}$ during the iterative fitting procedure.

We observed that during fitting it is not guaranteed that the mouth stays closed, and therefore add an energy term preventing contour points on the upper/lower lip to diverge

$$E_{\text{mouth}}(\mathcal{X}) = \frac{1}{M}\sum_{i=1}^{M} \|\mathbf{x}_i^{(u)} - \mathbf{x}_i^{(l)}\|^2, \tag{6}$$

where $\left\{\mathbf{x}_i^{(u)}, \mathbf{x}_i^{(l)}\right\}$ are $M = 11$ pairs from upper and lower lip, respectively, which are pre-selected on the template mesh. This energy term is weighted by $\lambda_{\text{mouth}} = 0.5$.

Note that at this stage we optimize the vertices of the head region only, while keeping all other vertices fixed by removing them from the linear systems. Analogous to the body fitting step we solve the nonlinear optimization using alternating optimization for vertex positions and edge rotations. Note that we do not employ the anisotropic bending model of [Achenbach et al. 2015], since the template's face region is too coarse to benefit from the (computationally more expensive) anisotropic wrinkle reconstruction.

5.3 Facial Details and Blendshapes

Similar to [Ichim et al. 2015], we adjust the template's teeth by optimizing for anisotropic scaling, rotation, and translation, based on the deformation of the mouth region from the undeformed template to the deformed and fitted mesh. We also transform the eyes by optimizing for isotropic scaling, rotation, and translation for each eye individually, again based on the deformation of the individual eye region from the undeformed to the deformed mesh.

Face animation requires a suitable set of blendshapes, which represent the face in different expressions, typically consisting of the FACS blendshapes [Ekman and Friesen 1978] and of visemes for speech animation. Since we only scan the actor in neutral facial expression, we have to "invent" a proper set of blendshapes. Since facial expression are similar across different individuals, we transfer all blendshapes from our generic template model to the fitted model using deformation transfer [Sumner and Popović 2004], similar to [Weise et al. 2011]. This transfers the deformation from the template model (generic neutral ↦ generic expression) to the target model.

Note that our blendshapes are rather generic, since they transfer the template's expression to the scanned person. Feng et al. [2017] instead scan additional expressions and use those as (highly personalized) blendshapes, but they do not generate additional ones. A good compromise would be to add a small number of scanned example expressions to the deformation transfer process, as done by example-based facial rigging [Li et al. 2010]. This, however, increase the acquisition time.

5.4 Texture Reconstruction

Analogous to the body fitting step, we generate a 4k×4k texture from the eight camera images of the face scanning session using Agisoft Photoscan. This yields an accurate high-quality texture, but only for the face region, which we therefore extract using a pre-selected image mask and then seamlessly copy it into the full-body texture using Poisson image editing [Pérez et al. 2003] (see Figure 6). As mentioned before, we keep the texture for eyes and teeth from the original texture. The luminance of these regions are adjusted, such that their mean luminance coincides with the mean luminance of the face. This adapts the texture of teeth and eyes to the lighting conditions of the scan.

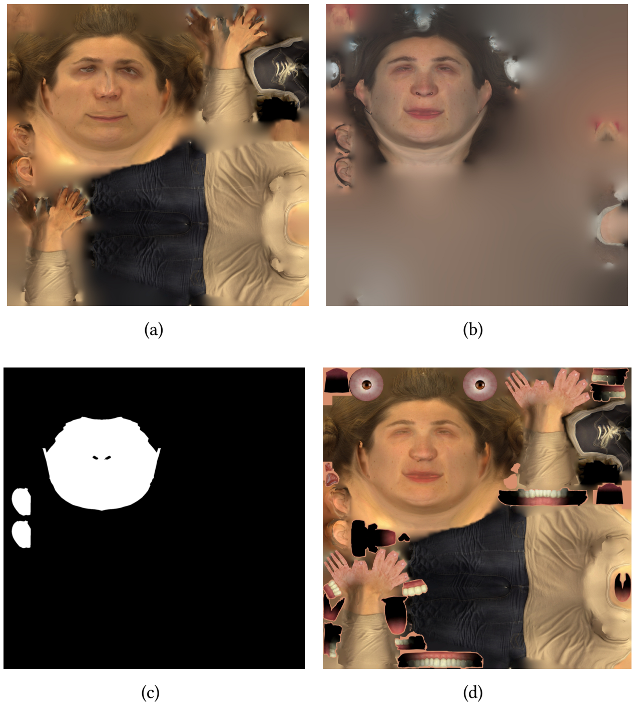

Figure 6: Textures computed from the camera images of the body scan (a) and the face scan (b). Since the face region is more accurately represented in the latter, it is extracted using a pre-computed image mask (c) and seamlessly copied into the body texture through Poisson image editing (d).

6 RESULTS

We tested our virtual human generation pipeline on a large set of subjects, and our approach reliably produced convincing results for all of them. A representative subset can be seen in Figure 1 and in the accompanying video.

The use of multi-view stereo reconstruction allows us to reconstruct both accurate geometries as well as high quality textures. As can be seen in Figure 7, additionally incorporating our dedicated face scanner significantly improves the visual quality of the face region, since it was scanned at higher resolution. A comparison of a captured image from the body scanning session with the personalized virtual human is depicted in Figure 8.

Our reconstructed characters can readily be animated in any standard graphics or VR engine, since they feature a standard skeleton for full-body and hand animation as well as a standard set of blendshapes for face animation. The accompanying video demonstrates that our characters can efficiently be animated and rendered in a real-time scenario. Figure 9 and the accompanying video show one of our scanned characters used as a conversational virtual agent, where face and body animation are crucial to enable the agent to talk, perform gestures, and show facial expressions.

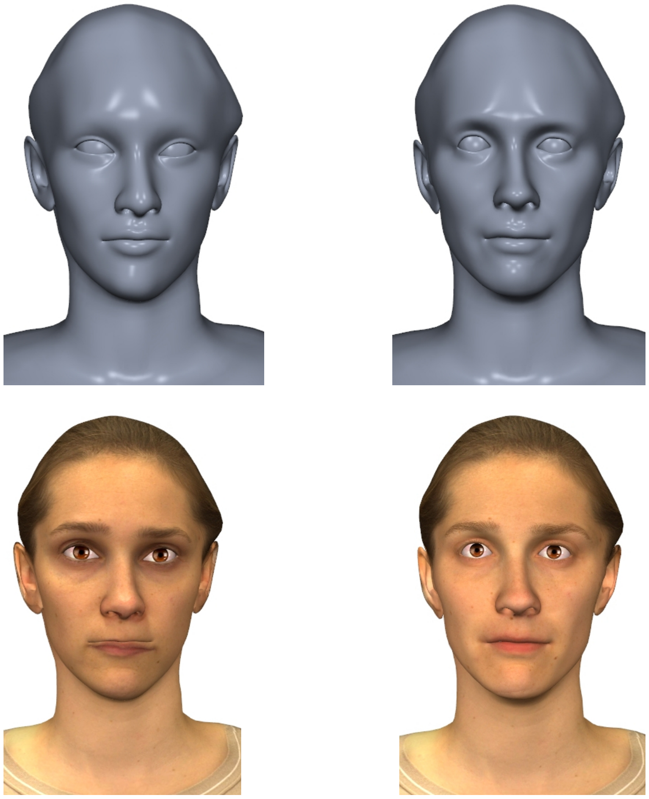

Figure 7: Comparison of the face region reconstructed from the full-body scan only (left) and by additionally incorporating the dedicated face scanning (right).

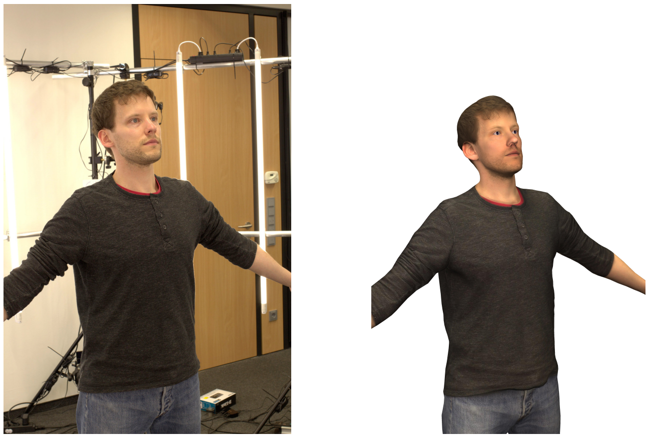

Figure 8: Comparison of a photo from the body scanning session with a rendering from the generated virtual human.

Our method also has some limitations: Texture artifacts may still occur in regions that are not visible from more than one camera, as is the case for all photogrammetry approaches. The most critical areas are the armpits and the hands, but also the crotch and the inner parts of the arms can be problematic. These issues can be overcome by using more cameras, which will lead to a better coverage for texture data at the expense of longer computation times. Furthermore, we do not remove the scene lighting during scanning from the albedo textures, as done, e.g., in [Bogo et al. 2014].

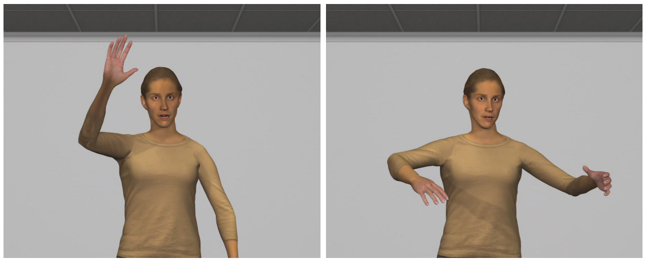

Figure 9: Our virtual humans can be directly used as expressive conversational agents, since they are able to gesture, talk, and to show facial expressions and emotions.

6.1 Performance

On average the processing of a single character takes about ten minutes from scan to a complete ready-to-animate avatar. See Table 1 for detailed information about computation times of our subprocesses, where the timings were taken on a desktop PC with Intel$^{©}$ Xeon$^{©}$ CPU (6×3.5 GHz) and a Nvidia$^{©}$ GTX 980 GPU.

The computationally most expensive part of our template fitting procedure is the computation of the closest point correspondences in each fitting iteration. While this can be accelerated by using a kD-tree or a similar space partitioning technique, we found that a simple linear search implemented on the GPU provides a much higher speed-up for the model complexities in our application. In comparison to a CPU-based kD-tree, our straightforward GPU implementation of a brute-force search is about 12 times faster. A GPU-based implementation of a spatial hierarchy would probably lead to an even higher speed-up, but would also require a considerably more complex implementation.

Table 1: Time needed for the sub-processes of our pipeline.

Process	Approx. time
Face scanning	1/10 s
Transfer images from face scanner	15 s
Full-body scanning	1/10 s
Transfer images from body scanner	80 s
Compute face point set $\mathcal{P}_F$	15 s
Compute body point set $\mathcal{P}_B$	75 s
Manual selection of landmarks	120 s
Automatic selection of facial features	60 s
Fit face geometry	20 s
Fit body geometry	35 s
Compute face texture	45 s
Compute and merge body texture	100 s
Compute facial blendshapes	5 s
Overall	$\sim$ 10 min

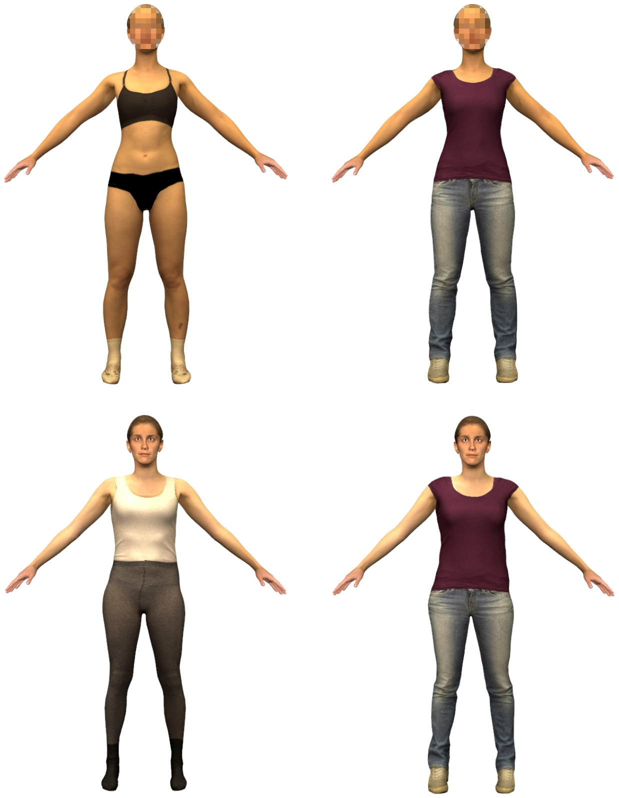

Figure 10: After reconstructing Subject A with both minimal clothing (top left) and clothing of interest (top right), we transfer this clothing to another Subject B with minimal clothing (bottom left) in order to get Subject B with the clothing from Subject A (bottom right).

6.2 Clothing Transfer

Due to their construction by fitting the same generic template model to scanner data, all our models share the same tessellation and hence are in one-to-one correspondence. This allows to transfer arbitrary per-vertex or per-texel properties between models, which we exploit for transferring clothing.

Similar to [Pons-Moll et al. 2017] we extract and store clothing as the difference (in geometry and texture) between a character wearing minimal clothing and the same character wearing a desired set of clothes. This clothing can then be transferred to another character, as shown in Figure 10. In our pipeline we segment clothing either manually or automatically by wearing a green suit.

Note that in contrast to [Pons-Moll et al. 2017] we still represent our character models as single-layer meshes, i.e., we bake the clothing into the model's geometry and texture. While this leads to less realistic cloth animations, it preserves the computational efficiency and compatibility with standard graphics engines.

While being a comparatively simple application, the ability to control the clothing of virtual humans is crucial in experiments with scanned virtual characters, as it allows to factor out perceptional effects caused by different clothing styles of the scanned subjects.

7 CONCLUSION

In this paper we presented a fast and reliable pipeline to digitally clone real persons into realistic virtual humans. For 3D-scanning we employ a custom-built camera rig with 40 cameras for the body and 8 cameras for the face, and compute dense point clouds through multi-view stereo reconstruction. In order to robustly deal with noise and missing data, we fit a generic human body model to the user's scanner data. By also transferring the skeleton, blendshapes, and eyes of the generic template to the model, our reconstructed virtual humans are ready-to-animate in standard game engines and VR frameworks. Furthermore, we demonstrated how to easily and seamlessly transfer clothing from one character to another, while still being compatible to standard rendering engines.

Our character generation requires only a minimum amount of user interaction and takes less than ten minutes on a desktop PC. It is therefore fast enough to be performed at the beginning of each session in a VR experimental study.

While our pipeline produced convincing results with all tested subjects, some inherent limitations remain. Due to scanning subjects in A-pose, some areas are not visible from enough cameras and thus are not reconstructed well. While missing data can be compensated by template data during geometry reconstruction, these regions still suffer from texture artifacts.

As future work we plan on using the proposed pipeline to generate characters for experiments with virtual mirrors or preference studies for personalized virtual agents. Another interesting direction for future work is the realistic modeling of clothing motion. Moreover, we will work on further speeding up the whole pipeline and making it fully automatic.

ACKNOWLEDGMENTS

We thank all scanned subjects. This work was supported by the Cluster of Excellence Cognitive Interaction Technology "CITEC" (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG), and through the project "KogniHome" funded by the Federal Ministry of Education and Research (BMBF).

REFERENCES

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.

Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. 2009. The Digital Emily Project: Photoreal Facial Modeling and Animation. In *SIGGRAPH 2009 Courses.* ACM, 1–15.

Brett Allen, Brian Curless, and Zoran Popović. 2003. The Space of Human Body Shapes: Reconstruction and Parameterization from Range Scans. *ACM Transactions on Graphics* 22, 3 (2003), 587–594.

Brett Allen, Brian Curless, Zoran Popović, and Aaron Hertzmann. 2006. Learning a Correlated Model of Identity and Pose-Dependent Body Shape Variation for Real-Time Synthesis. In *Proc. of Eurographics Symposium on Computer Animation.* 147–156.

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

Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. 2013. Robust discriminative response map fitting with constrained local models. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition.* 3444–3451.

Autodesk. 2014. Character Generator. https://charactergenerator.autodesk.com/. (2014).

Domna Banakou and Mel Slater. 2014. Body ownership causes illusory self-attribution of speaking and influences subsequent real speaking. *Proceedings of the National Academy of Sciences* 111, 49 (2014), 17678–17683.

Ilya Baran and Jovan Popović. 2007. Automatic Rigging and Animation of 3D Characters. *ACM Transactions on Graphics* 26, 3, Article 72 (2007), 8 pages.

Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. 2010. High-quality Single-shot Capture of Facial Geometry. *ACM Transactions on Graphics* 29, 4 (2010), 1–9.

Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proc. of SIGGRAPH*. 187–194.

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.

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.

Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. 2014. Dynamic 2D/3D Registration. In *Eurographics Tutorials*.

Sofien Bouaziz, Yangang Wang, and Mark Pauly. 2013. Online Modeling for Realtime Facial Animation. *ACM Transactions on Graphics* 32, 4, Article 40 (2013), 10 pages.

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), 1–19.

Chen Cao, Qiming Hou, and Kun Zhou. 2014a. Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation. *ACM Transactions on Graphics* 33, 4 (2014), 1–10.

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014b. FaceWarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2014), 413–425.

P. Ekman and W. Friesen. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press.

Andrew Feng, Dan Casas, and Ari Shapiro. 2015. Avatar Reshaping and Automatic Rigging Using a Deformable Model. In *Proc. of ACM Motion in Games*. 57–64.

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.

Andrew Feng, Ari Shapiro, Wang Ruizhe, Mark Bolas, Gerard Medioni, and Evan Suma. 2014. Rapid Avatar Capture and Simulation Using Commodity Depth Sensors. In *SIGGRAPH 2014 Talks*. ACM.

Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM Transactions on Graphics* 35, 3, Article 28 (2016), 15 pages.

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. 2011. Multiview Face Capture Using Polarized Spherical Gradient Illumination. *ACM Transactions on Graphics* 30, 6, Article 129 (2011), 10 pages.

Mar González-Franco, Daniel Perez-Marcos, Bernhard Spanlang, and Mel Slater. 2010. The contribution of real-time mirror reflections of motor actions on virtual body ownership in an immersive virtual environment. In *Proc. of IEEE Virtual Reality Conference*. 111–114.

P. Guan, A. Weiss, A. Balan, and M. J. Black. 2009. Estimating human shape and pose from a single image. In *Proc. of International Conference on Computer Vision*. 1381–1388.

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

David A. Hirshberg, Matthew Loper, Eric Rachlin, and Michael J. Black. 2012. Coregistration: Simultaneous Alignment and Modeling of Articulated 3D Shape. In *Proc. of European Conference on Computer Vision*. 242–255.

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.

Pei-Lun Hsieh, Chongyang Ma, Jihun Yu, and Hao Li. 2015. Unconstrained Realtime Facial Performance Capture. In *Proc. of Computer Vision and Pattern Recognition*. 1675–1683.

Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM Transactions on Graphics* 34, 4, Article 45 (2015), 14 pages.

Tao Ju, Scott Schaefer, and Joe Warren. 2005. Mean Value Coordinates for Closed Triangular Meshes. *ACM Transactions on Graphics* 24, 3 (2005), 561–566.

Marc 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*. to appear.

Marc Erich Latoschik, Jean-Luc Lugrin, and Daniel Roth. 2016. FakeMi: a fake mirror system for avatar embodiment studies. In *Proc. of ACM Virtual Reality Software and Technology*. 73–76.

J. P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Fred Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. In *Eurographics 2014 - State of the Art Reports*.

Hao Li, Etienne Vouga, Anton Gudym, Linjie Luo, Jonathan T. Barron, and Gleb Gusev. 2013. 3D Self-portraits. *ACM Transactions on Graphics* 32, 6, Article 187 (2013), 9 pages.

Hao Li, Thibaut Weise, and Mark Pauly. 2010. Example-based Facial Rigging. *ACM Transactions on Graphics* 29, 4, Article 32 (2010), 6 pages.

Shu Liang, Ira Kemelmacher-Shlizerman, and Linda G. Shapiro. 2014. 3D Face Hallucination from a Single Depth Frame. In *Proc. of International Conference on 3D Vision*. 31–38.

Matthew Loper, Naureen Mahmood, and Michael J. Black. 2014. MoSh: Motion and Shape Capture from Sparse Markers. *ACM Transactions on Graphics* 33, 6, Article 220 (2014), 13 pages.

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

Jean-Luc Lugrin, Johanna Latt, and Marc Erich Latoschik. 2015. Anthropomorphism and illusion of virtual body ownership. In *Proc. of the 25th International Conference on Artificial Reality and Telexistence and 20th Eurographics Symposium on Virtual Environments*. 1–8.

C. Malleson, M. Kosek, M. Klaudiny, I. Huerta, J. C. Bazin, A. Sorkine-Hornung, M. Mine, and K. Mitchell. 2017. Rapid one-shot acquisition of dynamic VR avatars. In *Proc. of IEEE Virtual Reality Conference*. 131–140.

Tabitha C Peck, Sofia Seinfeld, Salvatore M Aglioti, and Mel Slater. 2013. Putting yourself in the skin of a black avatar reduces implicit racial bias. *Consciousness and cognition* 22, 3 (2013), 779–787.

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

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

Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael Black. 2017. ClothCap: Seamless 4D Clothing Capture and Retargeting. *ACM Transactions on Graphics* 36, 4, Article 73 (2017), 15 pages.

Daniel Roth, Kristoffer Waldow, Felix Stetter, Gary Bente, Marc Erich Latoschik, and Arnulph Fuhrmann. 2016. SIAMC: a socially immersive avatar mediated communication platform. In *Proc. of ACM Virtual Reality Software and Technology*. 357–358.

Fuhao Shi, Hsiang-Tao Wu, Xin Tong, and Jinxiang Chai. 2014. Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos. *ACM Transactions on Graphics* 33, 6, Article 222 (2014), 13 pages.

Leonid Sigal, Alexandru O. Balan, and Michael J. Black. 2007. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Proc. of International Conference on Neural Information Processing Systems*. 1337–1344.

Mel Slater, Bernhard Spanlang, Maria V Sanchez-Vives, and Olaf Blanke. 2010. First person experience of body transfer in virtual reality. *PloS one* 5, 5 (2010).

Matthias Straka, Stefan Hauswiesner, Matthias Ruther, and Horst Bischof. 2012. Rapid Skin: Estimating the 3D Human Pose and Shape in Real-Time. In *Proc. of International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*. 41–48.

Jürgen Sturm, Erik Bylow, Fredrik Kahl, and Daniel Cremers. 2013. CopyMe3D: Scanning and Printing Persons in 3D. In *Proc. of German Conference on Pattern Recognition*. 405–414.

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

Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. 2015. Real-time Expression Transfer for Facial Reenactment. *ACM Transactions on Graphics* 34, 6, Article 183 (2015), 14 pages.

J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. 2016. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. of IEEE Computer Vision and Pattern Recognition*. 2387–2395.

Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. 2012. Scanning 3D Full Human Bodies Using Kinects. *IEEE Transactions on Visualization and Computer Graphics* 18, 4 (2012), 643–650.

Aggeliki Tsoli, Naureen Mahmood, and Michael J. Black. 2014. Breathing Life into Shape: Capturing, Modeling and Animating 3D Human Breathing. *ACM Transactions on Graphics* 33, 4, Article 52 (2014), 11 pages.

Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. 2011. Realtime Performance-based Facial Animation. *ACM Transactions on Graphics* 30, 4, Article 77 (2011), 10 pages.

Alexander Weiss, David Hirshberg, and Michael J. Black. 2011. Home 3D Body Scans from Noisy Image and Range Data. In *Proc. of IEEE International Conference on Computer Vision*. 1951–1958.

Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. 2016. An Anatomically-Constrained Local Deformation Model for Monocular Face Capture. *ACM Transactions on Graphics* 35, 4, Article 115 (2016), 12 pages.

Stefanie Wuhrer, Leonid Pishchulin, Alan Brunton, Chang Shu, and Jochen Lang. 2014. Estimation of Human Body Shape and Posture Under Clothing. *Computer Vision and Image Understanding* 127 (2014), 31–42.