EUROGRAPHICS 2013 / I. Navazo, P. Poulin
(Guest Editors)

Volume 32 (2013), Number 2

Pose Space Image Based Rendering

A. Hilsmann[1,2], P. Fechteler[2] and P. Eisert[1,2]

[1]Humboldt University of Berlin, [2] Fraunhofer HHI

Figure 1: *Our approach synthesizes images of clothes from a database of images by interpolating image warps as well as intensities in pose space.*

Abstract

This paper introduces a new image-based rendering approach for articulated objects with complex pose-dependent appearance, such as clothes. Our approach combines body-pose-dependent appearance and geometry to synthesize images of new poses from a database of examples. A geometric model allows animation and view interpolation, while small details as well as complex shading and reflection properties are modeled by pose-dependent appearance examples in a database. Correspondences between the images are represented as mesh-based warps, both in the spatial and intensity domain. For rendering, these warps are interpolated in pose space*, i.e. the space of body poses, using scattered data interpolation methods. Warp estimation as well as geometry reconstruction is performed in an offline procedure, thus shifting computational complexity to an a-priori training phase.*

Categories and Subject Descriptors (according to ACM CCS): I.4 [Computer Graphics]: Image Processing and Computer Vision—

1. Introduction

In this paper, we address the task of visualizing articulated objects with complex pose-dependent appearance, such as clothes. One subtle but very characteristic effect of clothes is wrinkling. Garments that roughly follow a person's shape, e.g. shirts or pants, typically exhibit fine wrinkles and buckling patterns, especially near joints. For such type of clothing, it is a feasible assumption that wrinkling is pose-dependent [WHRO10]. Traditionally, the visualization of clothes relies on a textured 3D computer graphics model while foldings and dynamics of the cloth are synthesized. Current cloth simulation techniques can produce highly realistic results for animated clothing with detailed wrinkling patterns but require complex physical simulation of the cloth characteristics [VMTF09, WHRO10]. An alternative to physical synthesis is observation of appearance through a number of images. In image-based rendering approaches, a database of previously recorded images of an object is used to generate new viewpoint images by appropriate interpolation and merging [LH96, BBM*01].

In our method, we approach the task of clothing visualization in an image-based manner, using as much information from real images of a piece of clothing as possible - providing *examples* of the clothing's appearance. We assume that clothing appearance (wrinkling and shading) is body pose-dependent, concentrating on clothes that roughly fit the shape of a human body. An image-based approach exploits the fact that all characteristics, such as texture deformation and shading properties, are captured by the example images.

This information can be extracted as spatial and photometric warps between the images and exploited to synthesize images of new pose configurations. The pose-dependency assumption allows a mapping of the extracted information onto a lower dimensional space, the *pose space*, i.e. the space of body poses, to synthesize new images as a function of body pose. For the interpolation of warps and image intensities, we utilize scattered data interpolation methods that have already been successfully used in pose space deformation (PSD) methods for animation [LCF00, SRC01].
Our approach is driven by the following assumptions:

- We are rather aiming at a plausible photo-realistic and perceptually correct visualization of clothes than at physically accurate and correct reconstruction.
- Texture and shading represent strong cues for the perception of shape such that fine details can be modeled by images (appearance). Hence, rough shape is modeled in a geometric model to allow animation while small but subtle details are represented by appearance.
- Wrinkles, creases, and appearance of clothes that roughly follow a person's shape are mainly influenced by the person's pose. Although external forces might also affect wrinkling behavior, appearance especially at fine wrinkles is mainly affected by nearby joints [WHRO10].
- Wrinkling behavior is primarily affected by the *nearest* joints. Hence, we can split up the pose space into subspaces of body parts with lower degrees of freedom, thus reducing the dimensionality of the interpolation domain as well as the number of required examples.

This paper is organized as follows. Sec. 2 reviews related work and states the contribution of our work. Sec. 3 describes the definition and construction of a database for pose space image-based rendering (PS-IBR). Sec. 4 describes how the extracted information in the database is used to synthesize images for new poses. Sec. 5 discusses results and describes possible application scenarios.

2. Related Work and Contributions

Image-based Rendering and Interpolation. Our intention of generating new images by warping and merging existing images from a database is closely related to image-based rendering and image interpolation. In image-based rendering (IBR), view-dependent appearance of an object is presented by a number of images. New viewpoint images are synthesized by appropriate filtering and interpolating these images [LH96, BBM*01]. IBR techniques can be classified based on how much geometry information about the scene is exploited [SK00]. Basically, the amount of images and geometry is a trade-off between photo-realism and compression of the model. While purely image-based representations have the advantage of photo-realistic rendering, they have high costs of data acquisition and storage requirements such that geometry is often exploited to reduce the number of images [DTM96, BBM*01].

Image interpolation transforms two images of the same scene (separated in space and/or time) to generate in-between images with a seamless transition [LLB*10, SLW*11]. This is generally done by determining image correspondences and gradually warping both images onto each other. A space-time interpolation method of complex real-world scenes was recently proposed in [LLB*10]. The captured images are parameterized in a space-time interpolation domain and a tetrahedralization of that space is used to interpolate correspondences between the images. Our proposed approach is related to their method, as we, too, interpolate warps between images. However, our interpolation domain is the high-dimensional space of body poses in contrast to the 3-dimensional space/time-domain, making animations possible. Furthermore, we extract not only spatial warps between images but also photometric ones to capture shading differences, especially at wrinkles with complex shading patterns.

Pose Dependency. Little research has been done in body-pose-dependent image-based rendering. One of the earliest work that approached the problem of modeling pose-dependent appearance was presented in [Dar98] who learned the silhouette appearance of an articulated arm as a function of its end position. Later, methods were proposed for textured non-rigid scenes, which separated the scene into several rigid scene parts and treated each part separately [CYJ02]. Recently, methods for image-based rendering of persons or clothes have been proposed that account for pose-dependency of appearance [XLS*11, HSR11]. These approaches search a texture in a database of images (based on pose [XLS*11] or silhouette information [HSR11]) to be mapped onto a 3D model. Both approaches search the best texture based on pose-related information but the synthesis procedure is not pose-dependent. In [HE12], an image-based animation method is proposed for a reduced scenario of an arm bending sequence with one degree of freedom and a linear interpolation between different poses, resulting in a 4D interpolation space, similar to time/space-interpolation.

While to our knowledge, a pose-dependent synthesis is new to image-based rendering, interpolation in pose space has been previously applied to geometry deformation in animation techniques, called pose space deformation (PSD) [LCF00, SRC01, WSLG07]. These methods provide examples of pose-dependent shape of an animated object for a number of example poses. Typically, the examples are modeled by an animator but body scans [ACZP02] or examples from computationally demanding simulations [WHRO10] have also been used. These examples guide the geometric deformation (e.g. muscle bulging) during animation which pure skinning methods like skeletal subspace deformation (SSD, details in [LCF00]) cannot model. PSD methods calculate vertex displacements between an animated reference shape and a provided example shape for predefined example poses. These example poses are located at scattered positions in *pose space*, e.g. the space of articulated body poses

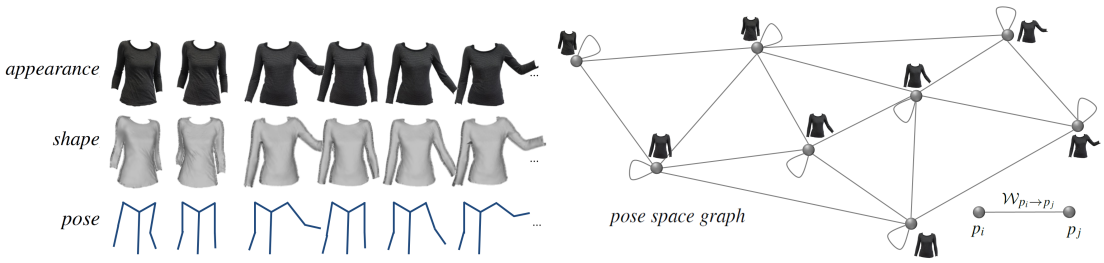

Figure 2: *The base representation consists of appearance (images and alpha masks) and rough shape information for several body poses and views. Each image is associated with a skeleton model, parameterizing the images in pose space. Additionally, parameters for spatio-intensity image warps are stored between images that lie close to each other in pose space. In a pose graph, nodes represent images and connections represent estimated image warps between them.*

[LCF00, WSLG07, WHRO10]. To generate the shape for a new position in this space, the vertex displacements are interpolated by scattered data interpolation methods, e.g. radial basis functions (RBF) [LCF00, SRC01] or k-nearest neighbor interpolation [ACZP02].

Recently, methods for learning the pose-dependent shape of clothing have been proposed in [WHRO10, GRH*12]. These methods learn pose- and body-shape-depended cloth wrinkling and drapery from high accurate physical clothing simulation to dress virtual characters. While these methods are purely shape-based, we combine pose-dependency with image-based rendering techniques.

Contributions. In contrast to classical IBR which is restricted to viewpoint interpolation, we have developed a body pose-dependent method that generates new images based on pose parameters by interpolating and merging images of clothes from a database in *pose space*. The concept of interpolating sample body poses is transferred and modified from PSD methods in animation [LCF00, SRC01, WSLG07] to IBR and we address related issues, such as blending and photo-consistency. The high dimensionality of the pose space is addressed by splitting up this space into subspaces of body parts to reduce the dimensionality of the interpolation domain as well as the number of required examples. This allows for a modeling of different body parts from different database images which are finally merged based on influence fields per body part.

3. Database Definition and Creation

Input to our visualization method is a set of skeleton joint angles or a comparable parameterization of a human pose. This pose parameterization defines the position in *pose space* to be interpolated from the database. The main idea of our approach is to capture a database of images in an a-priori training phase from different calibrated viewpoints showing different body poses. The poses parameterize the images in pose space, thereby providing scattered examples of pose-dependent appearance. To render an image for an arbitrary pose, a subset of images is selected from the database based on pose similarity. These images are warped and interpolated to the new pose and finally fused to one result by blending. In the following, we describe the definition of our image-based representation as well as the construction of a database. Sec. 4 concentrates on the synthesis by scattered data interpolation in pose space.

3.1. Base Model Representation

Our database (Fig. 2) consists of a set of calibrated images $\mathcal{I}_{p,v}$ (*appearance*) for various poses p and views v. Each image is associated with an alpha mask and a view-dependent 3D mesh $\mathcal{M}^3_{p,v} = \{\mathbf{V}^3_{p,v}, \mathcal{F}_{p,v}\}$ (*shape*), with vertices $\mathbf{V}^3_{p,v} = [\mathbf{v}_1 \dots \mathbf{v}_M]$ and topology $\mathcal{F}_{p,v}$, representing coarse geometry. For calibration and reconstruction of the coarse geometry, we use Bundler [SSS08] and refine the resulting depth maps/disparities by registering two neighboring stereo pairs onto each other. To do so, we triangulate the alpha masks using [She96] and estimate vertex correspondences between the images using the intensity-based image registration method of [HSE10]. From the estimated correspondences and calibration information, we can assign a depth to each vertex and store a depth map represented as a mesh $\mathcal{M}^3_{p,v} = \{\mathbf{V}^3_{p,v}, \mathcal{F}_{p,v}\}$ for each image.

For animation and parameterization purposes, each image is associated with a skeleton, representing the body pose p, e.g. as a vector of joint angles $\mathbf{q}_p$, which positions the image in *pose space*. For pose estimation, we fit a generic body model with skeleton information to the coarse geometry information of each image [FHE12]. Skinning weights between the meshes and the skeletons for SSD animation are calculated using the method of [BP07].

3.2. Image Warps

In addition to the base model, we store parameters for spatio-intensity image warps between images in the database. The information which images are connected by warps is organized in a graph structure which we call *pose graph* (Fig. 2). In this section, we explain the concept of pose graphs and

mesh-based spatio-intensity warps used for our representation. For simplicity reasons, we omit the subscript v, indicating the viewpoint, in the following. In fact, different viewpoints can be treated as different poses and interpolation between view points is akin to pose interpolation.

Pose Space and Pose Graph. Each image is associated with a pose parameterization $\mathbf{q}_p$ which defines its position in pose space. We use an axis-angle representation of each skeleton joint (the axes are also represented by two angles) and concatenate the angles in $\mathbf{q}_p$. Any other pose representation like joint quaternions or relative joint positions can also be used. As distance measure between two poses in pose space we use the L_2-norm:

$$d_{ij} = d(\mathbf{q}_{p_i}, \mathbf{q}_{p_j}) = \left\| \mathbf{q}_{p_i} - \mathbf{q}_{p_j} \right\|^2 \tag{1}$$

Image warps (see below) are estimated between *close* images in pose space, i.e. image pairs with a distance d_{ij} below a predefined threshold. In the pose graph (Fig. 2), we store the information which images are connected by warps, enabling smooth transitions between the poses. In this graph, nodes represent images (plus shape, silhouette and pose) and edges represent warps (note, that each image is also connected to itself). We define the neighborhood of image $\mathcal{I}_{p_i}$, i.e. all images that are connected to $\mathcal{I}_{p_i}$ by an edge, by $\mathcal{N}_{p_i}$. In other words, $\mathcal{N}_{p_i}$ defines the set of images with associated warps from and to $\mathcal{I}_{p_i}$.

General Definition of Mesh-based Warps. A spatial image warp $\mathcal{W}_s(\mathbf{x})$ from a source image $\mathcal{I}_i$ to a target image $\mathcal{I}_j$ moves the pixels $\mathbf{x}$ to a new location such that the warped image $\mathcal{I}_i(\mathcal{W}_s(\mathbf{x}))$ best resembles $\mathcal{I}_j$:

$$\mathcal{I}_i(\mathcal{W}_s(\mathbf{x})) \approx \mathcal{I}_j(\mathbf{x})$$

A mesh-based spatial warp between $\mathcal{I}_i$ and $\mathcal{I}_j$ can be defined by a 2D mesh $\mathcal{M}_i^2 = \{\mathbf{V}_i, \mathcal{F}_i\}$ on the source image, with vertices $\mathbf{V}_i = [\mathbf{v}_{i_1} \dots \mathbf{v}_{i_K}]$ and topology $\mathcal{F}_i$, and additional vertex displacements $\Delta\mathbf{V}_{i\to j}$, defining corresponding vertex positions $\mathbf{V}_i + \Delta\mathbf{V}_{i\to j}$ in the target image (Fig. 3):

$$\mathcal{W}_s : \{\mathbf{V}_i, \mathcal{F}_i, \Delta\mathbf{V}_{i\to j}\} \tag{2}$$

Such a warp moves the pixels with barycentric interpolation between vertex positions:

$$\mathcal{W}_s(\mathbf{x}) = \mathbf{x} + \sum_{t=1}^{3} \beta_t \Delta\mathbf{v}_{t_{i\to j}} \tag{3}$$

where $\Delta\mathbf{v}_t$ are the vertex displacements of the triangle surrounding $\mathbf{x}$ and β_t are the corresponding barycentric coordinates. However, differences between images cannot always be modeled by pure spatial warps due to e.g. varying lighting or shading patterns. In these cases, a photometric warp $\mathcal{W}_p(\mathcal{I}(\mathbf{x}))$ can be used to modify the intensities in the image, such that $\mathcal{W}_p(\mathcal{I}_i(\mathcal{W}_s(\mathbf{x})))$ best resembles $\mathcal{I}_j$ [HSE10]:

$$\mathcal{W}_p(\mathcal{I}_i(\mathcal{W}_s(\mathbf{x}))) \approx \mathcal{I}_j(\mathbf{x})$$

Often, such a photometric warp is modeled as an intensity scale field multiplied to the image (Fig. 3 and Fig. 4):

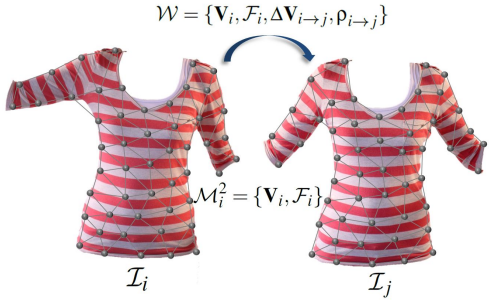

Figure 3: *A spatio-intensity image warp between $\mathcal{I}_i$ and $\mathcal{I}_j$ can be defined by a mesh $\mathcal{M}_i^2$ on $\mathcal{I}_i$ and vertex displacements into $\mathcal{I}_j$ as well as one intensity scale parameter per vertex.*

Figure 4: *A shading map generated from the photometric warps accounts for local shading at small wrinkles (left) as well as more global lighting changes (right).*

$$\mathcal{W}_p(\mathcal{I}(\mathbf{x})) \equiv \mathcal{W}_p(\mathbf{x}) \cdot \mathcal{I}(\mathbf{x}) \tag{4}$$

A mesh-based photometric warp can be defined by one intensity scale parameter ρ per vertex and the pixel intensity scale field is interpolated using barycentric coordinates:

$$\mathcal{W}_p(\mathbf{x}) = \sum_{t=1}^{3} \beta_t \rho_{t_{i\to j}} \tag{5}$$

A joint spatio-intensity warp between $\mathcal{I}_i$ and $\mathcal{I}_j$ can then be defined by a mesh in the source image plus vertex displacements and a vector of the intensity scale parameters:

$$\mathcal{W}_{i\to j} = \{\mathbf{V}_i, \mathcal{F}_i, \Delta\mathbf{V}_{i\to j}, \rho_{i\to j}\} \tag{6}$$

Creating Pose Graph Warps. We estimate and store joint spatio-intensity warps as defined above for each edge in the pose graph. For later interpolation, we split up the warps between the database images into a coarse image warp, induced by animating the associated 3D shape and subsequent projection into the desired view, and additional fine scale warps.

To animate a pose image $\mathcal{I}_{p_i}$ to an arbitrary pose p_a, we use skeleton subspace deformation (SSD) of the associated 3D mesh $\mathcal{M}_{p_i}^3$ and project the animated mesh into the new virtual camera. The established 2D vertex correspondences can then be used to warp the image to the new pose and view. We call these warps induced by SSD-animation and projection *SSD-guided warp*:

$$\mathcal{W}_{p_i\to p_a}^{\mathcal{SSD}} = \{\mathbf{V}_{p_i}, \mathcal{F}_{p_i}, \Delta\mathbf{V}_{p_i\to p_a}^{\mathcal{SSD}}, \mathbf{1}\} \tag{7}$$

Figure 5: *Fine-scale warp estimation. From left to right: Source and target poses overlaid, SSD-warped source pose image (note the deformation at the arm) and final warped source pose image. Note, how lighting is adapted in the fine-scale warped image (right).*

Figure 6: *Fine wrinkling patterns are modeled by the intensity warp: spatially warped image without intensity adaptation, spatio-intensity warped image and target image.*

where $\mathbf{V}_{p_i}$ denotes the original vertices of $\mathcal{M}^3_{p_i}$ projected into the camera of $\mathcal{I}_{p_i}$ and $\Delta\mathbf{V}^{\mathcal{SSD}}_{p_i\to p_a}$ denotes the vertex displacements induced by SSD animation to the desired pose and projection into the desired view.

The SSD-guided warps do not model real deformations, e.g. cloth wrinkling when an arm is bent. Hence, using SSD-guided image warps alone would result in implausible image warps. To compensate for the insufficiencies in the SSD-guided warp and to assure photo-consistency between poses, we estimate additional example fine-scale warps $\mathcal{W}^{fine}_{p_i\to p_n}$ on top of the SSD-guided warps between pose images $\mathcal{I}_{p_i}$ and $\mathcal{I}_{p_n}, p_n \in \mathcal{N}_{p_i}$ connected by an edge in the pose graph. The concatenated warp registers the images accurately both spatially and photometrically (Fig. 5 and 6). Formally, the concatenation $\oplus$ of these warps can be expressed as:

$$\begin{aligned}\mathcal{W}_{p_i\to p_n} &= \mathcal{W}^{\mathcal{SSD}}_{p_i\to p_n} \oplus \mathcal{W}^{fine}_{p_i\to p_n} \\ &= \{\mathbf{V}_{p_i}, \mathcal{F}_{p_i}, \Delta\mathbf{V}^{\mathcal{SSD}}_{p_i\to p_n} + \Delta\mathbf{V}^{fine}_{p_i\to p_n}, \rho^{fine}_{p_i\to p_n}\}\end{aligned} \tag{8}$$

where $\mathbf{V}_{p_i}$ and $\Delta\mathbf{V}^{\mathcal{SSD}}_{p_i\to p_n}$ are defined as above and $\Delta\mathbf{V}^{fine}_{p_i\to p_n}$ and $\rho^{fine}_{p_i\to p_n}$ are the parameters of the fine scale warps. To estimate these parameters between a SSD-warped source pose image $\mathcal{I}_{p_i}(\mathcal{W}^{\mathcal{SSD}}_{p_i\to p_n}(\mathbf{x}))$ and a target pose image $\mathcal{I}_{p_n}(\mathbf{x}), p_n \in \mathcal{N}_{p_i}$, we use the method of [HSE10].

We now have created a database representation which defines pose-dependent appearance at scattered positions in pose space by assigning a pose parameterization to each image. Furthermore, for each image, we have estimated warp parameters to other images at scattered positions in pose space. Both warp parameters as well as image intensities represent our example data which will be interpolated during rendering (Sec. 4).

4. Pose Space Image-Based Rendering

To synthesize an image for a new pose p_a, we select the *closest n* images $\mathcal{I}_{p_i}$ to p_a in pose space. For clarity reasons, we will first explain how we interpolate warps from the selected images to the new pose p_a before we explain how these images are selected (Sec. 4.2). Sec. 4.3 explains the blending of the selected images. Sec. 4.4 focuses on a separation of the pose space in subspaces to allow for a larger variety of poses and to reduce the number of required examples.

4.1. Warp Interpolation

For each of the selected n images, a warp to the new pose p_a is unknown and needs to be interpolated from the stored example image warps $\mathcal{W}_{p_i\to p_n}, p_n \in \mathcal{N}_{p_i}$. Recall, that the database warps are concatenated warps, consisting of an SSD-guided warp as well as a fine scale warp. The SSD-guided warps are fully defined by the geometry and the skeleton information and we only need to interpolate the fine scale warp parameters on top of the SSD-guided warp. Let $\tilde{\mathcal{W}}_{p_i\to p_a}$ denote the interpolated warp from pose p_i to a new pose p_a:

$$\tilde{\mathcal{W}}_{p_i\to p_a} = \mathcal{W}^{\mathcal{SSD}}_{p_i\to p_a} \oplus \tilde{\mathcal{W}}^{fine}_{p_i\to p_a} \tag{9}$$

where the interpolated fine scale warp $\tilde{\mathcal{W}}^{fine}_{p_i\to p_a}$ is represented by the interpolated vertex displacements and intensity scale parameters $\Delta\tilde{\mathbf{V}}^{fine}_{p_i\to p_a}$ and $\Delta\tilde{\rho}^{fine}_{p_i\to p_a}$.

The poses $p_n \in \mathcal{N}_{p_i}$ define example poses where a warp from $\mathcal{I}_{p_i}$ is known to. These poses are located at scattered positions in pose space and the interpolation problem is thus a problem of scattered data interpolation. One interpolation strategy also used in some PSD approaches [ACZP02, SRC01], is to find a smooth weighting function per example such that the interpolated warp parameters are a linear combination of the stored parameters from p_i to $p_n \in \mathcal{N}_{p_i}$:

$$\begin{aligned}\Delta\tilde{\mathbf{V}}^{fine}_{p_i\to p_a} &= \sum_{p_n\in\mathcal{N}_{p_i}} w_n(\mathbf{q}_{p_a}) \cdot \Delta\mathbf{V}^{fine}_{p_i\to p_n} \\ \tilde{\rho}^{fine}_{p_i\to p_a} &= \sum_{p_n\in\mathcal{N}_{p_i}} w_n(\mathbf{q}_{p_a}) \cdot \rho^{fine}_{p_i\to p_n}\end{aligned} \tag{10}$$

This means that we need to define weight functions w_n to all

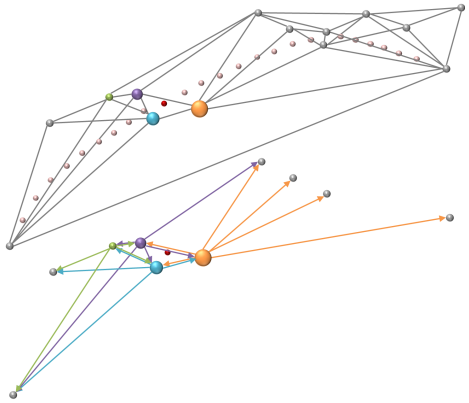

Figure 7: *Illustration of PS-IBR. Each database image is positioned in pose space by a skeleton parameterization (graph nodes). For a new pose (red dot), the closest n images are selected (colored dots, size reflects blending weight), for which we interpolate a warp to the new position from the database warps (graph edges, colored arrows in the lower image).*

poses $p_n \in \mathcal{N}_{p_i}$. These weight functions should fulfill three constraints [ACZP02]:

- The weights should sum to one:

$$\sum_{p_n \in \mathcal{N}_{p_i}} w_n(\mathbf{q}_{p_a}) = 1 \tag{11}$$

- If p_a falls onto a pose $p_n \in \mathcal{N}_{p_i}$ connected to p_i in the pose graph, the weight for that sample must be one, and all other weights must be zero:

$$w_n(\mathbf{q}_{p_a}) = 1 \;\text{ if } p_n = p_a \tag{12}$$

- The weights should be continuous so that the transitions are smooth during animation.

Several interpolation methods have been proposed in the PSD literature. Most approaches use radial basis functions [LCF00, SRC01, WSLG07]. However, one problem with radial basis functions is that they can give negative weights which can lead to exaggerated warps [ACZP02]. An alternative interpolation scheme is k-nearest neighbors (kNN) interpolation which was proposed by Buehler et al. [BBM*01] for view point interpolation in image-based rendering and also used by Allen et al. [ACZP02] in PSD. It selects k closest example points (in our case, $k = |\mathcal{N}_{p_i}|$) and assigns each of them a weight based on their inverse distance:

$$f_n(\mathbf{q}_{p_a}) = \frac{1}{d(\mathbf{q}_{p_a}, \mathbf{q}_{p_n})} - \frac{1}{d(\mathbf{q}_{p_a}, \mathbf{q}_{p_k})} \tag{13}$$

where $d(\mathbf{q}_{p_a}, \mathbf{q}_{p_n})$ is the distance between the new pose and the example pose parameterization $\mathbf{q}_{p_n}$ as given in Eq. (1) and p_k is the k^{th} closest pose in $\mathcal{N}_{p_i}$. Finally, the weights are normalized such that they sum to one:

$$w_n(\mathbf{q}_{p_a}) = \frac{f_n(\mathbf{q}_{p_a})}{\sum_{p_t \in \mathcal{N}_{p_i}} f_t(\mathbf{q}_{p_a})} \tag{14}$$

We interpolate a warp $\tilde{\mathcal{W}}_{p_i \to p_a}$ for each of the selected images from their stored example warps using kNN-interpolation. Next, the warped images are blended as described in Sec. 4.3.

4.2. Selection of Images

With the interpolation scheme introduced above, a pose p_a for which a warp can be interpolated from a sample pose p_i is constrained by

$$\mathbf{q}_a = \sum_{p_n \in \mathcal{N}_{p_i}} w_n \mathbf{q}_{p_n} \;, 0 \le w_n \le 1. \tag{15}$$

Hence, the possible warps depend on the number and position of each image's neighbors in the pose graph and it is not enough to simply search for the n closest pose images by minimizing e.g. Eq. (1). We rather have to take into account the pose graph neighbors forming the set of sample warps:

$$e_{ai} = \|\mathbf{q}_{p_a} - \sum_{p_n \in \mathcal{N}_{p_i}} w_n \mathbf{q}_{p_n}\|^2 . \tag{16}$$

Such a measure reflects how well a position in pose space $\mathbf{q}_{p_a}$ can be interpolated from the stored warps. As the distance between the selected images and the new position in pose space should not be neglected completely, we search the n pose images for which the following combined distance measure is minimized:

$$d'_{ai} = \alpha d_{ai} + \gamma e_{ai} \tag{17}$$

where d_{ai} is a distance between the poses p_i and p_a according to Eq. (1). In our experiments, we used $\alpha = \gamma = 0.5$.

4.3. Image Blending

After we have interpolated a warp from each selected image $\mathcal{I}_{p_i}$ to the new position in pose space, we blend the warped images to synthesize an image for the new pose p_a by

$$\mathcal{I}_{p_a} = \sum_{i=1}^{n} w^b_{p_i}(\mathbf{q}_{p_a}) \mathcal{W}_{p_i \to p_a}(\mathcal{I}_{p_i}) \tag{18}$$

where $w^b_{p_i}$ is the blending weight for image $\mathcal{I}_{p_i}$ in contrast to the weight functions for warp interpolation. Again, we use the kNN method to calculate the blending weights $w^b_{p_i}$.

In contrast to PSD, the warps are interpolated from different reference poses. If we select n images for warping and blending, one pose space warp interpolation is performed for each of these images separately. For each image, the set of example warps is defined by the edges in the pose graph. This means, that for each image, we interpolate from a different set of examples. While for each selected image the kNN interpolation scheme results in a smooth animation sequence when traveling through the pose graph, the images to be blended might differ slightly. This is a consequence of different reference positions for each of the interpolation problems. To assure photo-consistency during blending, we

Figure 8: *Left to right: Influence weights for the right body part (red ≈ 1, blue ≈ 0) and a synthetic image generated by local warping and blending (n = 3 per body part); selected images from the database for each subspace with blending weights; the two pose graphs and positions of the selected images as well as the new image in the pose graphs.*

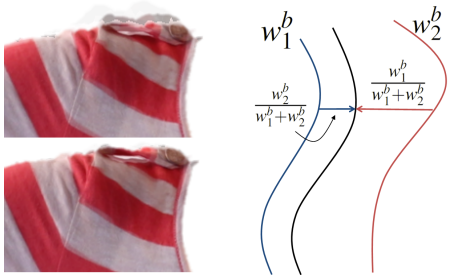

Figure 9: *Left: Blending without and with silhouette clean-up; right: calculation of mean silhouette from the two images with the highest blending weights.*

adjust the silhouettes of the warped images onto a weighted mean silhouette of the two images with the highest blending weights (weighted according to the blending weights, see Fig. 9), gradually warping between the silhouettes during animation. We do this instead of warping onto the silhouette of the image with the highest blending weight to avoid jumps during animation, if the image with the highest weight changes and the silhouettes of the images in the database differ a lot (e.g. for sparse datasets with very different example poses or for loose clothing). The registration scheme of the silhouettes is very simple and fast. We sample the silhouettes and perform a non-rigid Iterative Closest Point (ICP) algorithm. As the images are already very close to each other, the ICP converges in less than 4 iterations in most cases.

4.4. Definition of Sub Pose Spaces

With the method described above, we can synthesize images of poses that lie in the convex hull of the captured body poses and can be expressed as a linear combination of sample poses connected by warps according to Eq. (15). Hence, the variety of possible pose images depends on the number of example images and warps per example as well as their position in pose space. Due to the high dimensionality of the pose space it is therefore difficult to synthesize arbitrary pose images with a limited set of images and warps. Under the assumption that wrinkling is mostly affected by the nearest joints (e.g. the left elbow does not influence deformations at the right arm), we can split up the pose space into subspaces, related to body parts, and proceed as above for each subspace and finally blend and merge the images locally. This allows for a larger variety of possible poses, without the need of more example images.

To split up the pose space into subspaces related to body parts, we define influence fields, indicating how much a vertex is influenced by that body part. In our experiments, we split up the upper body (torso, sternum, shoulder and elbow joints), as well as the lower body (torso, hip and knee joints), into two parts each, related to the left/right side of the body. The influence fields are smooth and overlapping (Fig. 8) and used for local warping as well as local blending. Each image is now positioned in two pose spaces and in each pose space, we search for the nearest n images. Local blending weights are determined based on the distance in each pose space. For all selected images (regardless in which subspace they were selected), we interpolate a warp as explained in Sec. 4.1 where Eq. (10) is extended to

$$\begin{aligned}\tilde{\Delta\mathbf{v}}^{fine}_{m,p_i\to p_a} &= \sum_s \sum_{p_n\in\mathcal{N}^s_{p_i}} w^s_n(\mathbf{q}_{p_a})\cdot\omega^s_m\cdot\Delta\mathbf{v}^{fine}_{m,p_i\to p_n}\\ \tilde{\rho}^{fine}_{m,p_i\to p_a} &= \sum_s \sum_{p_n\in\mathcal{N}^s_{p_i}} w^s_n(\mathbf{q}_{p_a})\cdot\omega^s_m\cdot\rho^{fine}_{m,p_i\to p_n}\end{aligned} \tag{19}$$

where $\sum_s$ describes the summation over the different subspaces and ω^s_m is the influence weight of vertex m for subspace s. $\mathcal{N}^s_{p_i}$ denotes the neighborhood of pose p_i in the pose graph of subspace s (note, that the pose graphs of the different body parts can be different). After all images have been warped to the new pose, the images are blended locally based on the influence fields and blending weights per subspace. Fig. 8 shows an example with local warping and blending. The right body part is composed of two images with blending weights 0.9 and 0.1, while the left body part is composed of two different images with blending weights 0.58 and 0.42. Another example is shown in Fig. 10.

5. Results, Discussions and Applications

We have created different databases of upper and lower body clothing. The number of example images varies between 40 and 240 (viewpoints and poses). In all our experiments, we used two pose spaces as explained in Sec. 4.4 and $n = 3$ images per pose space. Results of our method are best evaluated

Figure 10: *Synthesis of a completely new pose (right) from four database images (left).*

visually in the accompanying video showing different pose animations and view interpolations of synthesized poses (example frames are shown in Fig. 1 and 14). The use of real images and warp interpolation yields realistic movements of wrinkles (e.g. wrinkling at the elbows in the jacket sequ. or at the knee in the jeans sequ.), very fine details (such as an epaulette in the shirt sequ.) as well as shading during animation (e.g. shadow casting of the arm on the body in the jacket sequ.). Even for loose clothing (such as the blouse) strong wrinkling at the torso induced by arm movements can be modeled.

Fig. 10 shows an example of a completely new synthetic pose generated from four database images, depicted on the left. No similar pose has been captured in the database and without splitting the pose space into subspaces, more example poses would have been necessary to synthesize this pose due to the high dimensionality of the pose space.

Besides interpolation, we also performed experiments on extrapolation (Fig. 13 and accompanying video). The details in Fig. 13 show that the kNN interpolation yields plausible results for moderate extrapolation, as it produces non-negative weights which sum to one and warps are not exaggerated. Hence, realistic wrinkling and shading can even be modeled for extrapolation (note e.g. the realistic wrinkling at the upper legs in the jeans examples). However, the farther the extrapolated pose lies outside the convex hull, the more blurring artifacts can appear due to misalignment of the images as no warp examples are known outside the convex hull (see details in Fig 13). Coarsely, these misalignments are adjusted by our silhouette clean-up such that they are not very obvious during animation. Future work will concentrate on more sophisticated texture clean up for extrapolation, e.g. using additional texture correspondences, similar to *Floating Textures* [EDM*08]. Self-occlusions in the extrapolated pose that have not been captured in other poses can cause *overlapping* artifacts (see knee detail in Fig. 13). Future work will include visibility maps to detect these cases during synthesis.

For further evaluation, we followed a Leave-one-out procedure. We took one of the pose images (with pose information, geometry and warps from and to that pose) out of the database and synthetically generated an image for that pose with our method. Following, we reduced the dataset by half and proceeded as before. Fig.12 shows two examples. The differences between the ground truth and the synthetic image increase if the datasets are reduced as i) images and warps are interpolated from larger distances and ii) during the analysis procedure the warps need to be estimated between more distant example images (compare Fig. 11). Nevertheless, the synthetic images, especially the pose-dependent characteristics, are still visually correct as long as plausible views can be estimated between the example images (see also the video for a comparison of visualization results with different numbers of example images). Recall, that our objective is the realistic and plausible visualization of pose-dependent characteristics and not accurate reconstruction.

Figure 11: *If the source and the target poses differ too much, differences between the images cannot be completely aligned by image registration, especially in cases of self-occlusions in the source image. Left to right: source image, target image, warped source image onto target pose.*

Figure 12: *Comparison to ground truth. Left to right: ground truth image, synthetic image produced with full data set and synthetic image produced with half data set. Third row: differences between the ground truth and the synthetic images for the lower example. Although differences can be seen in the difference images, the synthetic images are plausible and visually correct for both datasets.*

Figure 13: *Examples of extrapolation: In each example, on the left depicted is the nearest interpolated pose (outlined) overlaid with the two extrapolated images depicted on the right. Details are shown in the lower right, see also accompanying video.*

Figure 14: *Example frames and details of synthetic animation sequences (see also accompanying video). Fine details, such as wrinkling at the elbow, shading and the complex movements, e.g. of an epaulette, can be modeled with our approach.*

Limitations. Like in any example-based or image-based method, the variety of poses that can be synthesized by our method depends on the number of images in the database and their distribution/density in pose space. Generally, a dense sampling of the pose space facilitates warp estimation as well as image synthesis. If the images in the database differ too much, differences between the images might not be fully aligned by image registration and warp estimation is more challenging, e.g. for loose clothing which performs strong deformations, or self-occlusions, i.e. missing texture information, in the source pose (e.g. Fig. 11). The larger the distances between poses (for interpolation as well as extrapolation), the stronger are the artifacts introduced by SSD animation (e.g. candy wrap, Fig. 15), making correction by fine scale warps more difficult. However, similar to PSD, our method is not constrained to SSD as the underlying animation technique but can be combined with any other more sophisticated animation method.

Cost. The cost for visualization mainly depends on the number of pose subspaces and selected images per pose space (number of image warps and silhouette clean-ups). In contrast to radial basis function interpolation used in many PSD approaches, the kNN interpolation method is cheap and the only time-consuming part is the warping of the images which can be done in a few milliseconds. The size of the database only influences the search for the nearest n images per pose space.

Application. The main application we are targeting at is

Figure 15: *Candy-wrap artifact induced by SSD for strong extrapolation.*

the visualization of clothes in augmented reality applications such as virtual try-on of clothes. In such an environment, a user is tracked by one or more cameras and visualized on a mirror-like display showing him/her augmented with new clothes. Based on the estimated pose of the user, an image of the selected clothes can be synthesized from the database and mapped onto the image of the user.

6. Conclusions and Discussions

We presented a body pose dependent image-based rendering technique which interpolates image warps and intensities in pose space for the synthesis of images of clothes for new body poses. To allow for a larger variety of poses, we split up the body into different body parts and warp and blend locally based on influence fields per body part. The use of real images results in a realistic visualization of the clothes especially at fine details. Generally, our method is not limited to the visualization of clothes but can rather be applied to any articulated objects where the assumptions of Sec. 1 hold.

Acknowledgements

We thank Johannes Furch and Benjamin Prestele for their support during data capturing and video production. This work is partially funded by the German Science Foundation, DFG EI524/2-1 and by the European Commission, FP7-287723 REVERIE.

References

[ACZP02] ALLEN B., CURLESS B., Z. POPOVIĆ Z.: Articulated Body Deformation from Range Scan Data. *ACM Trans. Graph. 21*, 3 (2002), 612–619. 2, 3, 5, 6

[BBM*01] BUEHLER C., BOSSE M., MCMILLAN L., GORTLER S., COHEN M.: Unstructured Lumigraph Rendering. In *Proc. ACM SIGGRAPH* (2001). 1, 2, 6

[BP07] BARAN I., POPOVIĆ J.: Automatic Rigging and Animation of 3D Characters. *ACM Trans. Graph. 26* (2007). 3

[CYJ02] COBZAS D., YEREX K., JÄGERSAND M.: Dynamic Textures for Image-based Rendering of Fine-Scale 3D Structure and Animation of Non-rigid Motion. In *In Proc. of Eurographics* (2002), pp. 1067–7055. 2

[Dar98] DARRELL T.: Example Based Image Synthesis of Articulated Figures. In *In Advances in Neural Information Processing Systems* (1998), MIT Press. 2

[DTM96] DEBEVEC P. E., TAYLOR C. J., MALIK J.: Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-based Approach. In *Proc. ACM SIGGRAPH* (1996). 2

[EDM*08] EISEMANN M., DECKER B. D., MAGNOR M., BEKAERT P., DE AGUIAR E., AHMED N., THEOBALT C., SELLENT A.: Floating Textures. *Computer Graphics Forum 27*, 2 (2008), 409–418. 8

[FHE12] FECHTELER P., HILSMANN A., EISERT P.: Kinematic icp for articulated template fitting. In *Vision, Modeling, and Visualization Workshop 2012* (2012). 3

[GRH*12] GUAN P., REISS L., HIRSHBERG D., WEISS A., BLACK M. J.: DRAPE: DRessing Any PErson. In *Proc. ACM SIGGRAPH)* (2012). 3

[HE12] HILSMANN A., EISERT P.: Image-based animation of clothes. In *Eurographics 2012 Short Papers* (2012). 2

[HSE10] HILSMANN A., SCHNEIDER D. C., EISERT P.: Realistic Cloth Augmentation in Single View under Occlusion. *Computers & Graphics 34*, 5 (2010), 567–574. 3, 4, 5

[HSR11] HAUSWIESNER S., STRAKA M., REITMAYR G.: Image-based Clothes Transfer. In *Proc. Int. Symp. Mixed and Augmented Reality* (2011), pp. 169–172. 2

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *Proc. ACM SIGGRAPH* (2000), pp. 165–172. 2, 3, 6

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proc. ACM SIGGRAPH* (1996). 1, 2

[LLB*10] LIPSKI C., LINZ C., BERGER K., SELLENT A., MAGNOR M.: Virtual Video Camera: Image-Based Viewpoint Navigation Through Space and Time. *Computer Graphics Forum 29*, 8 (2010), 2555–2568. 2

[She96] SHEWCHUK J. R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, vol. 1148 of *Lecture Notes in Computer Science*. 1996, pp. 203–222. 3

[SK00] SHUM H. Y., KANG S. B.: A Review of Image-Based Rendering Techniques. In *Proc. of SPIE Visual Communications and Image Processing (VCIP 2000)* (Perth, Australia, 2000). 2

[SLW*11] STICH T., LINZ C., WALLRAVEN C., CUNNINGHAM D., MAGNOR M.: Perception-Motivated Interpolation of Image Sequences. *ACM Trans. Appl. Perception 8*, 2 (2011), 1–25. 2

[SRC01] SLOAN P.-P. J., ROSE C. F., COHEN M. F.: Shape by Example. In *Proc. Symp.on Interactive 3D Graphics* (2001), pp. 135–143. 2, 3, 5, 6

[SSS08] SNAVELY N., SEITZ S. M., SZELISKI R.: Modeling the World from Internet Photo Collections. *Int. J. Comput. Vision 80*, 2 (2008), 189–210. 3

[VMTF09] VOLINO P., MAGNENAT-THALMANN N., FAURE F.: A Simple Approach to Nonlinear Tensile Stiffness for Accurate Cloth Simulation. *ACM Trans. Graph. 28*, 4 (2009). 1

[WHRO10] WANG H., HECHT F., RAMAMOORTHI R., O'BRIEN J. F.: Example-based Wrinkle Synthesis for Clothing Animation. *ACM Trans. Graph. 29*, 4 (2010). 1, 2, 3

[WSLG07] WEBER O., SORKINE O., LIPMAN Y., GOTSMAN C.: Context-Aware Skeletal Shape Deformation. *Computer Graphics Forum 26*, 3 (2007), 265–273. 2, 3, 6

[XLS*11] XU F., LIU Y., STOLL C., TOMPKIN J., BHARAJ G., DAI Q., SEIDEL H.-P., KAUTZ J., THEOBALT C.: Video-based Characters - Creating New Human Performances from a Multi-View Video Database. *ACM Trans. Graph. 30*, 4 (2011). 2