Virtual Try-On Through Image-based Rendering

Stefan Hauswiesner, *Student Member, IEEE,* Matthias Straka, and Gerhard Reitmayr, *Member, IEEE*

Abstract—Virtual try-on applications have become popular because they allow users to watch themselves wearing different clothes without the effort of changing them physically. This helps users to make quick buying decisions and thus improves the sales efficiency of retailers. Previous solutions usually involve motion capture, 3D reconstruction or modeling, which are time consuming and not robust for all body poses. Our method avoids these steps by combining image-based renderings of the user and previously recorded garments. It transfers the appearance of a garment recorded from one user to another by matching input and recorded frames, image-based visual hull rendering and online registration methods. Using images of real garments allows for a realistic rendering quality with high performance. It is suitable for a wide range of clothes and complex appearances, allows arbitrary viewing angles and requires only little manual input. Our system is particularly useful for virtual try-on applications as well as interactive games.

Index Terms—mixed reality, augmented reality, image-based rendering, virtual try-on

1 Introduction

Virtual try-on applications have become popular in recent years because they allow users to see themselves wearing different clothes without the effort of changing them physically. This helps users to quickly judge whether they like a garment or not, which in turn allows retail shops to sell more in less time. Web shops employ virtual try-on applications to reduce product return rates [1]. Moreover, digital entertainment applications and games also aim at creating an image or a video of a user wearing different clothes. Digital content creation has become a major challenge in recent years as virtual worlds increase in size and complexity.

In our system, users standing inside a room equipped with cameras can see themselves on a large TV screen which shows them wearing different clothes. Users are allowed to move freely inside the room and can see themselves from arbitrary viewpoints. The system therefore needs to capture and render the user at interactive rates while augmenting his or her body with garments.

Some previous solutions display garment pictures as static overlays or retexture images from a single camera. These approaches are not directly capable of producing arbitrary viewing angles. Moreover, garments do not adapt their size and shape to the user due to their static nature. Other solutions address these issues by reconstructing 3D clothes models. These approaches can produce the desired output, but require motion capture to track the user's position over time. Motion capture is a challenging task that we wanted to avoid.

We propose an approach where a user can be displayed wearing previously recorded garments. We achieve this by creating a garment database that stores the appearance of a worn garment over time. These recordings are not required to be from the same user and are performed using the same multi-camera device that is used for augmentation. The database can be queried by silhouette images. At runtime, the best fitting frame is picked for rendering. The image-based visual hull (IBVH) algorithm is used to render users and clothes from arbitrary viewing angles. To fit the transferred clothes to the user and adapt the garment's shape to its new body, rigid and non-rigid registration is performed. Both registration tasks are formulated as optimization problems.

Our approach is suitable for a wide range of clothing, and multiple garments can be combined. By using images of real garments instead of virtual models, a realistic rendering quality can be achieved. The complex appearance of clothes that comes from anisotropic materials and non-rigid deformations is already captured by the images and can therefore be reproduced with high performance. It does not require any physics simulation, because the effect of gravity and cloth stretch is also present in the images. Due to image-based rendering, our method does not need an explicit 3D reconstruction of the garments or the user to produce views from all sides. For the same reason, it does not require manual 3D modeling by digital artists and it avoids motion capture. The whole process does not need manual interaction other than an initial garment segmentation.

In this work we improve all stages of our previous augmentation pipeline [2]. We contribute efficient methods for GPU-based silhouette matching by line-based and area-based silhouette sampling and evaluate their performance. Moreover, a novel method to extend the pose space by combining multiple matches is introduced. We introduce an extended non-rigid registration formulation that adapts the garment shape by silhouette and depth data terms. The problem domain of the registration process is analyzed to derive an efficient non-linear conjugate gradient solver with proper initialization that exploits frame-to-frame coherence to improve quality and performance. Coherent visual quality over time is also enforced by a weighted infinite response filter that is capable of suppressing artifacts inherent

• *The authors are with the Graz University of Technology*
E-mail: hauswiesner—straka—reitmayr@icg.tugraz.at

(a) (b) (c) (d)

Fig. 1. Results of the proposed method: a user is dressed with the clothing of a previously recorded user. Image of the user and the garment before augmentation are shown for illustration.

to a discrete pose space. We evaluated all improvements for their impact on the visual quality by comparing them to ground truth images.

2 Related work

Interactive virtual try-on applications need to track the user's position and pose to augment him or her with garments. The garments themselves can be reconstructed from camera images. Virtual- and mixed reality dressing applications that combine these aspects have been suggested.

2.1 Motion capture

Motion capture, or human pose tracking, is the task of determining the user's body pose. It usually involves a pose and shape model that is fitted to sensor data and therefore comprises a model-based tracking problem. We only consider optical, marker-less pose tracking, because we do not want users to wear markers.

Pose tracking from multiple video streams [3], [4], [5] was used for animating and rendering people. Recent GPU-based implementations adapt pose and shape in real time [6]. A system for markerless human motion transfer [7] was suggested to transfer motions from one person to another. Another system can modify the user's body shape [8] in a virtual mirror. Recent developments in sensor technologies have enabled the acquisition of depth images in real-time, which opened up new possibilities for pose tracking with a single camera. [9] have shown how to track full body motions using a time-of-flight camera. The more recent Microsoft Kinect device allows for real-time recording of color and depth images at a very low cost, as well as high-quality real-time human pose estimation [10].

2.2 Clothes reconstruction

Many virtual dressing applications draw a textured clothes mesh over a camera image. Obtaining that mesh is a key aspect of such systems. Some approaches use CAD models [11], which are labor intensive to create.

Clothes can also be reconstructed from images. Reconstructing the garment from a video sequence is a hard task, especially because of occlusions and the non-rigid shape of cloth. Many approaches use markers on the cloth for capturing, which makes them less suitable for our method. More recent approaches do not require markers [12], [13]. They usually use a shape-from-stereo approach and apply complex processing to the data to account for occlusions. However, all approaches that rely on point correspondences that are computed from the image data assume a certain texturedness of the garment. By using the light dome of [14] or a laser scanner [15] this limitation can be removed, but such hardware is expensive and processing can not be performed in real time.

Once the shape of a garment is digitized it needs to be fitted to the user's body model. This is a complex problem [16] that is usually not handled in real time.

2.3 Virtual try-on

Previous methods [17], [18], [19] work by finding the best matching dataset in a previously recorded database that contains all possible poses of the user. These systems first learn and then search a database of poses by using a pose similarity metric. The best match is used to deform a texture to fit the user. However, like many other retexturing approaches they operate in 2D and therefore do not allow the user to view him- or herself from arbitrary viewpoints.

The *Virtual Try-On* project [1] offers a set of applications for various tailoring, modeling and simulation tools. 3D scans of real garments are acquired by color-coded cloth. Cloth surface properties are measured from real samples. *MIRACloth* is a clothes modeling application [20] which can create garments, fit them to avatars and simulate them. However, both Virtual Try-On and MIRACloth do not include a mixed reality component that allows users to see realistic clothing on themselves immediately.

Kinect-based body scanning [21] enables virtual try-on applications at low costs but systems with a single sensor require multiple views. A system that uses manually modeled garment meshes was introduced [22]. Similar to our system, it performs a non-rigid registration to align garments and user. We use a non-linear formulation instead of Laplacian surface editing to avoid finding correspondences.

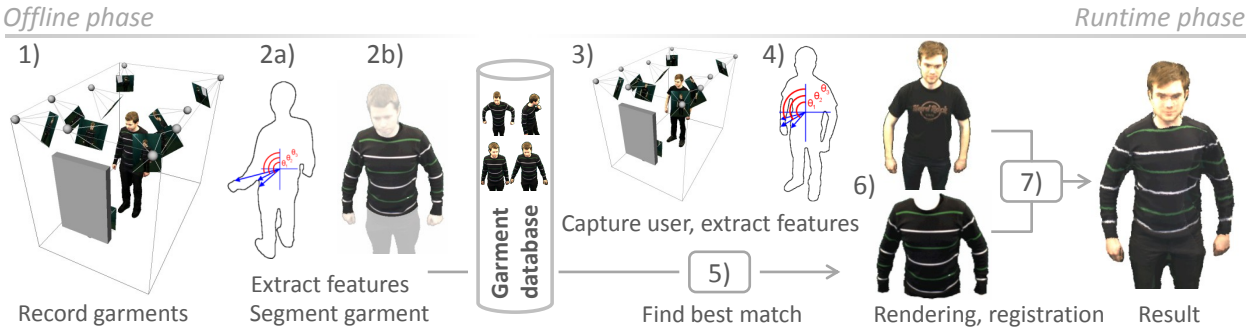


Fig. 2. Overview of our pipeline. In the offline phase, garments are recorded and processed to create a database. At runtime, the database can be queried for a record that matches the current user's pose. The recorded data is used to augment the user with a garment.

Transferring garment meshes from one human model to another is an important step for many virtual try-on applications. It requires a shape and pose adaption process. Volumetric Laplacian deformation [23] can achieve this.

Motion capture, reconstruction and retexturing is used to render dressed people [24]. Other virtual mirrors are restricted to specific tasks, like augmenting logos or shoes [25], [26] and even faces [27], [28].

2.4 Prerequisites

The virtual dressing room that is used for this work consists of a 2x3 meter footprint cabin with green walls [29]. Ten cameras are mounted on the walls: two in the back, two on the sides and six in the front. The cameras are synchronized and focused at the center of the cabin, where the user is allowed to move freely inside a certain volume (see Figure 2 left). All cameras are calibrated intrinsically and extrinsically and connected to a single PC. Most processing is performed on the graphics processing unit (GPU). The output device is a 42" TV that is mounted to the front wall in a portrait orientation. In such a setup, silhouettes can be extracted from the camera images quickly and robustly by background subtraction. Silhouettes make novel view synthesis very efficient. The image-based visual hull (IBVH) algorithm creates a depth map for arbitrary viewpoints [30]. Such a depth map can be textured with the camera images for realistic image-based rendering.

3 The augmentation process

Similar to [18] our clothes augmentation process has an offline phase for recording garments and an online phase where users can be augmented. The stages of the recording process (see Figure 2) are:

1) A user wears a garment which should be used for future augmentations. He or she enters the dressing room and performs a short series of different poses while being recorded.
2) Garments are segmented and features are extracted from the recorded video streams. Results are stored in a garment database.

This phase can be controlled: it is possible to recapture the scene when incomplete, or switch segmentation strategies. From now on, users who enter the dressing room can be augmented with previously recorded garments. We call this the runtime phase:

3) Users can move freely inside the room while being captured.
4) Features from the captured images are extracted.
5) The best fitting pose from the garment database is selected.
6) The selected pose of the desired garment and the captured user are rendered from the same viewpoint using image-based rendering.
7) Small pose mismatches are compensated for by rigid and non-rigid registration.

This process results in a composite mirror image, showing the user wearing a different item or items of clothing. A pipeline that enables such an approach has several key stages: first, the garment database needs to be created such that it can be accessed very efficiently at runtime (section 4). Second, at runtime the garment and the user need to be aligned roughly before rendering (section 5.1). Third, quick image-based rendering is required (section 5.2). Finally, remaining mismatches between garment and user need to be resolved on a per-pixel level (section 5.3).

4 Offline: garment database construction

A user puts on one or multiple garments which should be transferred to other users. The other clothing should be selected to allow for easy segmentation. In this paper we call this user the *model-user* to emphasize the difference to an end-user. We assume that the model-user can be instructed to perform all the desired motions, whereas the end-user just consumes the product as he or she wishes.

The model-user enters the dressing room and performs a series of different poses while being recorded. Recordings consist of synchronized videos from the ten cameras. Each garment database contains a single recorded sequence and therefore a single piece of clothing. When multiple garments or different sizes should be provided, each item needs its own recording and database.

4.1 Preprocessing

This stage takes the camera videos as input. Videos are recorded at 15 Hz, so usually the user's relative motions between two successive frames are small. Similar looking frames are of limited use for the later stages, because minor pose offsets are compensated for by registration. To reduce the database size, many frames can therefore be skipped. Currently, we only pick every second frame to be included in the garment database. Moreover, before insertion every new frame is checked for silhouette pixel overlap against the frames in the database to skip frames that are too similar.

Next, each camera image that is not skipped is segmented separately. First, a previously recorded static background image is subtracted to remove the background. Then, the garment is segmented: we use color keying and a graph cut tool. Defining the color key or a graph-cut seed is the only manual interaction that is needed to create the garment database from an image sequence. At runtime after the registration process, the segmented pixels are culled to remove the unwanted parts from the output.

4.2 Extracting features

At runtime, the frame which contains the model-user's pose that is most similar to the current user's pose has to be found. This can be achieved by matching colors, silhouettes or higher level features like motion capture positions. We did not find color information to be particularly useful, because garments of the model-user and the current user are likely to be entirely different. Motion capture may fail for cases where limbs are close to the body or form loops. We therefore focus on silhouettes, which are well defined as long as the user stays inside the viewing frustums of the cameras. To find the best fitting pose based on silhouettes, a metric to calculate silhouette similarity is required. When silhouette similarity or difference can be measured, standard algorithms can be used to search the resulting space.

Our approach extends the work of [17]. We work on more than one silhouette image. This additional input data is required to obtain a descriptive feature vector even when main features of the user, like his or her arms and legs, are located in front of the body and thus not visible in one or more silhouette images. During our evaluation we found that four cameras spaced around the user were usually sufficient to see the user's limbs for all body orientations. Figure 4 shows the spatial arrangement of the four cameras in our setup.

Required invariances

It is beneficial for a matching algorithm to be invariant to at least the scale and translation of the user's pose. For example, it is very likely that the model-user has a different body size than the end-user. Moreover, it is common that users do not stand on exactly the same spot in the room. Rotational invariance is not desired for our system, because we use a non-uniform distribution of cameras in the cabin.

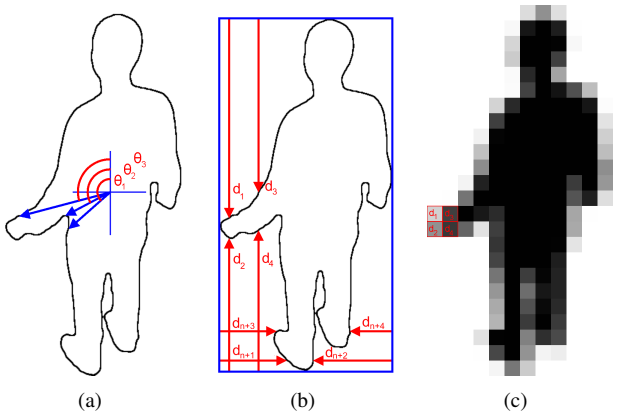


Fig. 3. Silhouette features comparison. (a) the distance between center of mass and silhouette edge at regularly spaced angles may provide insufficient sampling of the arms. (b) axis aligned sampling of the bounding box alleviates the problem. (c) area-based sampling computes fill-rates of grid cells and therefore also represents the silhouette's interior.

We therefore do not want to match poses that are oriented differently to avoid undersampling of important body parts.

To extract features, we compared three different silhouette sampling approaches. The first approach uses a radial pattern.

Radial line sampling

To extract features, the center of mass of each silhouette image is computed [2]. From the center of mass, 360 regularly spaced directions are sampled to find the closest and second closest silhouette exit edges (see Figure 3(a)). The closest edges usually describe the central body shape. The second closest edges describe the location of arms and legs. The distance between center of mass and edge is stored. When there is no second closest edge its distance is set to the distance of the closest edge. All distances are normalized per image to be independent of scale. Invariance to translation is given by relating all distances to the center of mass.

Axis-aligned line sampling

The second approach uses an axis-aligned sampling pattern. We chose this sampling pattern over a radial pattern, because silhouettes of people frequently have the property that arms and legs are aligned parallel to the radial sampling lines. Figure 3(a) illustrates this problem. Such an alignment causes an underrepresentation of arms and legs in the feature space, which makes it hard to match these body parts correctly.

First, the minimum bounding rectangle of each silhouette image is computed. Along the x and y intervals of the rectangle, regularly spaced lines are sampled (see Figure 3(b)). Along each line, the minimum distances from the ends of the line to the silhouette are stored as a vector. The concatenated description vectors of all four silhouette images form the silhouette matching space. Similar to

previous approaches [18], [2], all distances are normalized to unit length to allow matching across different scales. Invariance to translation is given, because all measurements are taken at positions relative to the bounding box.

Area-based sampling

The third approach samples the silhouettes densely. The minimum bounding rectangle of each silhouette is split into grid cells, which are sampled to obtain the fill rate of each cell (see Figure 3(c)). By aligning the sampling grid with the bounding box, this method is invariant to translation and scale. This method is more robust against a noisy segmentation than line sampling, because fill rates vary rather smoothly at silhouette edges and also the interior of the silhouette is explicitly contained in the feature vector. However, because the feature vector's components only contain overlap information instead of location information, the feature vectors are less descriptive than their line sampled counterparts. As a consequence, the search space can not be reduced as effectively as with line sampling methods (Figure 12(a)).

Section 7 evaluates all three sampling strategies with the result that area-based and axis-aligned sampling performed better than radial in terms of matching precision. However, area-based sampling requires more memory access operations and a larger search space.

4.3 Reducing the dimensionality

Depending on the sampling resolution and pattern, the resulting feature space can have several thousand dimensions, which describe the pose of the model-user at one moment in time. All frames of the recorded videos are processed in that manner, resulting in a large data matrix of pose descriptions. Using principle component analysis (PCA), we were able to reduce this to around 50 dimensions for line sampling, or 200 dimensions for area sampling without losing descriptiveness. Both vector spaces are small enough to query the database at runtime by a simple search. Other silhouette matching approaches [18] create data structures to improve search performance. In our system, however, exhaustive searching is sufficiently fast and only of minor importance to the overall performance (see Figure 15).

4.4 Extending the pose space

The garment database may not always contain a satisfying pose. This is the case when the model-user forgets to cover certain poses, or when the temporal resolution of the recording is not sufficient. To alleviate this problem, it is possible to extend the pose space at runtime by combining several poses into a single output. See Figure 4 and 5 for an illustration of the process.

To achieve this, we propose the following procedure. First, the best fitting pose is queried as described above. Then the matching error is examined. If it does not exceed a certain threshold then the match is considered good enough and only a single output is generated. However, if the threshold is exceeded, we split the silhouette images that

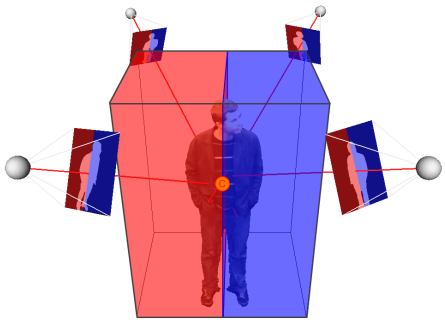


Fig. 4. The pose space is split at the X coordinate of the projected center of mass in each silhouette image. This approximates a 3D splitting plane that would be slower to compute.

are used for matching vertically at the X screen coordinate of the center of the user. Each half is then transformed to its PCA subspace and matched against the database separately. Therefore, during the offline phase the principle components of the garment database need to be computed for each half separately too. This method is an approximation of splitting the space in 3D, which to compute precisely would require more computations, in particular the creation of depth maps for each silhouette image. Images can also be split horizontally to allow for different poses in the upper/lower body half. This approximation assumes that the body segments are clearly distinguishable in the camera images that are used for matching. In practice this means that cameras mounted at the sides or straight above the user can not be used for pose matching.

Split segments are joined when they contain the same best match in order to decrease rendering time when possible. Finally, a blending kernel is launched that stitches the segments and employs a linear blending filter to cross-fade between them. To allow for free viewpoint motion, the blending operation can not be performed in image space, because when viewing from the sides the occlusion needs to be resolved correctly. Instead, the visual hull points of all split segments are clipped against the axis-aligned planes that are assumed by the matching. Figure 5 shows an extended pose space rendering result. This method can improve the output quality considerably, which can be measured. See Section 7 for evaluation data.

The output of this stage is the garment database. Each entry consists of features extracted for later matching, background-subtracted color images and segmentation information.

5 At runtime: clothes augmentation

Once one or more garment databases have been created, users can enter the dressing room and watch themselves wearing different clothes on the display in front of them.

First, the same features as in the offline phase are extracted from the current camera images. These features are transformed to PCA space by applying the precomputed

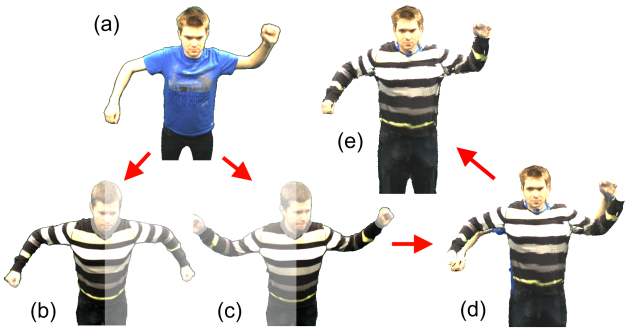


Fig. 5. Extending the pose space by combining two poses into a new pose. (a) shows the current user's pose. (b) and (c) show the best fitting poses for each image half. The result is a rendered image of the two halves without (d) and with non-rigid registration (e).

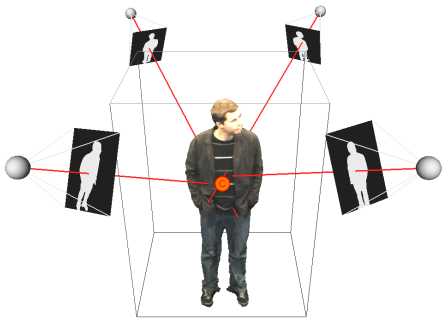


Fig. 6. This illustration shows how the 3D center of mass point C is approximated. The 2D centers of mass from the silhouette images are extruded and a 3D intersection is approximated.

transformation matrix from the offline stage. The result is used to find the garment database entry where the model-user's pose is closest to the pose of the current user. We use a simple Euclidean distance measure to find the closest feature vector during the search. The camera images associated with this entry show the best match for the desired garment. Most likely the found images show the model-user standing in a slightly different position than the current user. To compensate for this offset, a rigid registration is performed.

5.1 Rigid registration

The goal of this stage is to find a translation vector and a scale factor that aligns the rendering output of the user and the garment model. At this stage no rendering is performed yet, which means that no 3D data is available to determine the transformation by standard methods. Instead, we approximate the translation and scale by utilizing the silhouette images and the projection matrices of the corresponding cameras.

To achieve this, we need two or more cameras that see the user entirely. While our system utilizes 10 cameras in total, four of them are mounted in the upper corners of the cabin (see Figure 6). Due to their distance to the user, the frustums of these cameras is sufficiently large to see the whole user and can be used for registration.

First, the 2D center of mass is extracted from each of the silhouette images of the cameras. By transforming these points with their corresponding inverted projection matrices we obtain a 3D ray for each camera. Such a ray emanates from a camera center and runs through the 2D center of mass on the corresponding image plane. These rays do not necessarily intersect, so we compute the point with the least squared distance to the rays. The result is an approximation of the 3D center of mass, which otherwise would require a 3D reconstruction to determine. To compute the translation, we subtract the 3D center of mass points of the garment model and the current user's model.

The 3D center of mass does not necessarily lie on the rays that are cast through the 2D center of masses due to the perspective projection of the cameras. However, garment and user are subject to a similar error and as a result the rigid registration offsets were satisfying during our experiments.

A very similar operation is performed with the topmost 2D point of each silhouette image. We assume that the 3D intersection of the corresponding rays is a good approximation of the top of the head of the user. The Z-coordinate of this 3D point describes the height above the floor. To determine scale, we simply compare the body heights of the user and the model-user. This of course only compensates for different body heights. We leave the other dimensions to be compensated by the non-rigid registration, which can also handle different body shapes more precisely.

We do not compensate for rotation, because the matching phase is only invariant to translation and scale. This means that the matching phase already found a correctly rotated pose at this stage. This behavior is especially useful because it is hard to determine the rotational registration from silhouette images alone.

During evaluation, these approximations proved to be sufficiently stable. With the computed translation and scale factors, the current viewing matrix is updated. This way, the subsequent garment rendering pass produces its output at the desired location and scale in 3D space.

5.2 Image-based rendering

In this phase, the matched garment dataset and the current user are rendered. Both rendering tasks need to quickly generate an output image from a set of camera images with silhouette information.

We employ the image-based visual hull (IBVH) [31] algorithm to compute depth maps of novel viewpoints directly from the segmented camera images. It bypasses the computation of an explicit representation, such as a voxel grid or a mesh. Extracting such a representation and rendering it are two separate tasks which contain some redundancy. Therefore, it is beneficial to the overall performance to directly derive an output image from the camera images.

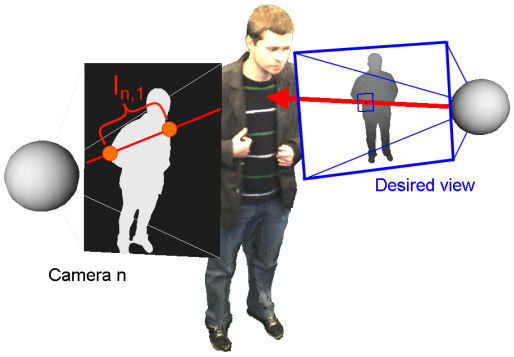


Fig. 7. Illustrating the concept of image-based visual hull rendering. For every pixel its viewing ray (red arrow) is projected onto each camera plane (red line). All interior intervals $I_{n,1..m}$ are found and intersected. The front most interval starts with the output depth value.

The IBVH algorithm works by first defining a viewing ray for every output pixel. Each viewing ray is projected onto all of the camera images. The resulting lines are searched for intersections with the user's silhouette. Pairs of intersections build intervals that describe where the user is located along the viewing ray. These intervals need to be intersected to find the subintervals where all cameras agree. The beginning of the front-most interval describes the depth value at each pixel. See Figure 7 for an illustration. Currently we use the IBVH rendering implementation of [30], which exploits the temporal coherence in the camera images to improve the overall performance.

After the depth map is extracted, it has to be textured using the camera images. We use a view-dependent texture mapping [32] scheme that utilizes the viewing direction of the cameras and surface normals to find suitable color samples for every output pixel.

The garment and the user are rendered to different buffers from the same viewpoint. The garment's position and scale were modified by the rigid registration process to obtain a rough alignment. In addition to the small shape and pose differences, the rendered buffers differ from the desired final output: parts of the model-user are still visible. For example, the head, hands and all items of clothing that should not be augmented. To remove these parts later on, the segmentation labels that were determined in the offline phase are used during rendering to write the alpha channel. After non-rigid registration the alpha values are used to remove any remaining parts of the model-user from the output.

5.3 Non-rigid registration of the output images

The last remaining difference to the desired output image is usually a small pose and shape inconsistency. Large offsets and differences in scale have already been compensated by the rigid registration. But the garment database does not cover all possible poses at an infinitely dense resolution: it only contains discrete samples of the pose space. Moreover, the body shapes of the current user and the model-user do not necessarily match. The non-rigid registration is used to compensate for this minor lack of overlap. See Figure 8 for an illustration.

We identified two main methods to achieve a better overlap: by interpolating adjacent poses, or by moving the pixels of one adjacent pose. Interpolating is rather difficult, because silhouette-based pose spaces are high-dimensional and therefore many adjacent datasets have to be found and interpolated. Additionally, a high quality interpolation requires corresponding features in color or silhouette space.

We therefore apply an optimization procedure that translates the pixels of the rendered garment to account for small deviations of pose. In addition to reducing pose inconsistencies, the optimization allows the garment shape to adapt to the current user's body shape.

The domain of the problem can be formulated as an energy function that should be minimized. We use pixel locations as the data elements to allow the silhouette to shrink or grow and to retain the spatial arrangement of pixels. Previous approaches align garment models in 3D [23], which is accurate but slow due to the larger problem domain. For producing a correct appearance, however, it is sufficient to align the rendered images in 2D. This reduces the dimensionality of the optimization problem considerably. Non-rigid deformations are often formulated as sets of linear equations [33]. However, linear approaches require corresponding features to guide the deformation. For aligning a garment with the user, however, it may be hard to find such correspondences, because both shape and appearance can be different. We therefore formulate the problem as a non-linear energy function, which allows us to directly specify the desired objective: overlap between garment and user.

The energy function E_{total} that is minimized is the sum of all per-pixel energies $E(x, y)$. Every pixel (x_{ij}, y_{ij}) of the garment silhouette is a particle and can move freely.

$$E_{total} = \sum_i \sum_j E(x_{ij}, y_{ij}) \tag{1}$$

The optimization procedure is initialized with the garment pixel's locations:

$$\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \begin{pmatrix} i \\ j \end{pmatrix} \tag{2}$$

Energies consist of a data term and a neighbor-term for regularization.

$$E(x_{ij}, y_{ij}) = D(x_{ij}, y_{ij}) + \alpha \cdot N(x_{ij}, y_{ij}) \tag{3}$$

$$D(x_{ij}, y_{ij}) = (I_{garment}(i, j) - I_{user}(x_{ij}, y_{ij}))^2 + \gamma \cdot (M_{garment}(i, j) - M_{user}(x_{ij}, y_{ij}))^2$$

$$I_{garment}(i, j) = \begin{cases} 1, & \text{if } (i, j) \text{ belongs to garment} \\ 0, & \text{otherwise} \end{cases}$$

$$I_{user}(x, y) = \begin{cases} 1, & \text{if } (x, y) \text{ belongs to user} \\ 0, & \text{otherwise} \end{cases}$$

(a) (b)

Fig. 8. (a) illustrates how the non-rigid registration adapts the garment to the user's shape and pose. (b) shows the influence of the depth data term on the non-rigid registration: the garment adapts to the user's hand.

The data term D of Equation 3 describes the overlap between garment and user by computing the L2-norm between silhouette and depth value differences. The I-terms are 1 for foreground pixels and 0 for background and thus describe a silhouette. The M-terms denote the depth maps of the garment and the rendered user. During rendering the depth map values are normed to unit range according to the near- and far clipping planes. These plane settings should be equal when rendering the user and the garment to avoid unwanted distortions.

During optimization, the I-terms push garment pixels towards the user's body and thus align the shape outlines. In addition, the M-terms move pixels within the shapes to align regions with similar depth and depth patterns. For example, misplaced sleeves are moved towards the user's arms even when the user holds his arms in front of his body (see Figure 8 for an illustration). The I and M-terms are non-linear with respect to their two spatial dimensions. As a consequence, we need a solver for non-linear equations.

The regularization term N of Equation 3 tries to retain the spatial arrangement of the cloth. We use the direct neighbors of each garment pixel as well as neighbors that are further away, but using a smaller weight. One such neighbor is N_{uv}.

$$N_{uv}(x_{ij}, y_{ij}) = \delta \cdot C_{uv}(x_{ij}, y_{ij}) + \cdot S_{uv}(x_{ij}, y_{ij})$$

$$C_{uv}(x_{ij}, y_{ij}) = (min(\sqrt{(x_{ij} - x_{uv})^2 + (y_{ij} - y_{uv})^2}, d) - d)^2 \quad (4)$$

$$\text{with } d = \sqrt{(u - i)^2 + (v - j)^2}$$

$$S_{uv}(x_{ij}, y_{ij}) = (u' - x_{uv})^2 + (v' - y_{uv})^2$$

$$\text{with } u' = x_{ij} + (u - i) \text{ and } v' = y_{ij} + (v - j)$$

x_{uv} and y_{uv} are the new x and y coordinates of the neighbor, and u' and v' are its initial coordinates relative to x_{ij} and y_{ij}. The compression term C_{uv} tries to keep the neighbor at least at its initial distance d, while the stretch term S_{uv} tries to keep it at its relative position. This regularization is important to retain features of the garment, like stripes and printed logos.

The factor α weighs the regularization versus the data term. It therefore can be seen as the *stiffness* of the garment. Moreover, it indirectly controls the convergence speed of the optimization: stiffer garments converge faster because the influence of the data terms propagate quicker. However, when the stiffness is set too high, the performance decreases again. For such cases it is more desirable to use a larger neighborhood for regularization.

γ weighs the depth data- against the silhouette term. To achieve a registration effect within the silhouette like in Figure 8(b), the depth term needs to be emphasized.

δ scales the compression- against the stretch penalty. Both are good for regularization, but there are slight differences that justify using both at the same time. The compression term is particularly useful for reducing pixel collisions when the optimization result is converted back to discrete pixel locations. Moreover, a relatively strong penalization of compression can be used to retain the screen-space area of loose garments. The stretch term on the other hand is more suitable for retaining the spatial arrangement of a neighborhood, because it also penalizes rotations.

5.3.1 The solver

We assume that the pose differences between garment and user are small, which makes the structure of the problem not prone to local minima. The energy function is minimized hierarchically, but not in a globally optimal way. In contrast to our previous approach [2], we use the nonlinear conjugate gradient (NCG) method. We observed that the proposed energy function contains many narrow valleys formed by opposing gradients of the data term and the regularization term. Figure 9 illustrates the error function in a space with reduced dimensionality: only two particles are considered, and the image space is reduced to a single dimension. The path to the minimum is formed by a diagonal ridge that is caused by the $C_{uv}(x, y)$ term, the quadratic slopes (red) caused by the $S_{uv}(x, y)$ term and the data term that decreases in positive x-direction. For such problems, NCG converges faster than gradient descent

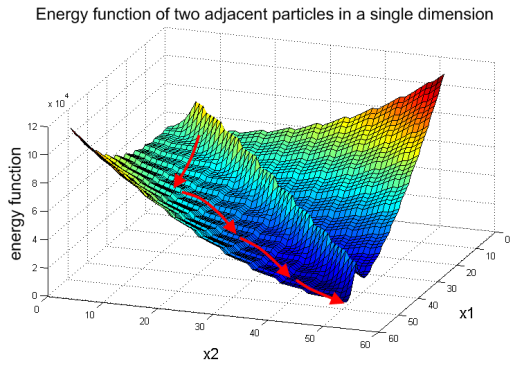


Fig. 9. Plot of the energy function E_{total}. For illustration purposes, the problem is reduced to two dimensions. The shallow valley that is marked with red arrows indicates a quick path to the minimum that a conjugate gradient method can take.

because it can follow the valleys. Our solver computes the β factor according to Polak−Ribière and uses an automatic reset mechanism $\beta = max(0, \beta^{PR})$ [34]. The output of Figure 8, for example, can be computed by 50 iterations of gradient descent, or 15 iterations of the nonlinear conjugate gradient method with the same fixed step length. The additional cost of computing the β factor every iteration is negligible compared to this speed up. We assume the derivative of the *min*-function in Equation 4 to be 0 where it is not differentiable.

The registration starts by converting the IBVH-rendered color image of the current user to a binary image that only shows the user's silhouette. Then, low resolution copies of the silhouette and the IBVH-rendered depth map are computed and both are smoothed by convolution with a Gauss kernel. The low resolution level usually has only $\frac{1}{16}$th of the output resolution. For both the low- and the high resolution levels, the x and y gradients are precomputed because these terms are static during the optimization. The optimization procedure itself consists of a repeated execution of three steps: computing the derivatives at each garment pixel location in x and y direction, computing β^{PR} and the conjugate gradient and finally updating the pixel locations. We use a fixed step length instead of a line search, which we found empirically to reduce execution time. First, 70 iterations are executed on the low resolution buffers. Then, the optimization state is magnified to full resolution by linear interpolation. Now it is possible to compute several iterations at the full resolution, but we observed that the difference is usually not perceivable.

The vector between the new, optimized pixel locations and their initial positions yields an offset buffer that describes how garment pixels need to move in order to maximize overlap with the user. This buffer is morphologically closed and smoothed by a 20×20 box filter to suppress noise. Finally, the garment pixels are moved according to the offset buffer. All of these steps are implemented as CUDA kernels, which allows such a high number of operations to be performed for every frame. Section 7 evaluates the positive impact on the output quality of the non-rigid registration.

5.3.2 Initialization and temporal coherence

The described registration process is very sensitive to the current user's output shape, especially when rather large step lengths are chosen for fast convergence. This can lead to a perceivable swinging and jumping of the garment image over time even when the same database frame is used. To remove this behavior, temporal coherence can be enforced for successive output images that are computed from the same database frame. An effective way to achieve this goal is to initialize the optimization procedure with the last frame's state.

This way, only the user's relative motion between the last frame and the current frame needs to be compensated for. As a result, pixels of body parts that have not moved are already in a converged state and therefore stay at a coherent location over time. Moreover, the last frame's state is usually closer to the desired result than a newly initialized state and therefore the optimization converges faster.

However, the last frame's state can not be reused by simply copying, because between the frames the viewpoint might have changed. Therefore, the state needs to be transformed between the different coordinate systems by image warping [35]. Now the initial state defined in Equation 2 can be translated by the offsets from the last frame.

As a result, the garment does not swing or jump between successive frames as long as the same database frame is used. Moreover, the non-rigid registration compensates for larger pose mismatches with the same number of iterations, or, requires less iterations for the same distance because the optimization converges across frames.

5.3.3 Cross-fading during frame transitions

For the other case, where two successive output images are created from different garment database frames, there may be a perceivable *jump* between the frames. This is mostly due to garment deformations that are too small for the registration algorithms to capture. To alleviate this problem, we cross-fade the output image with the previous output, thus creating an infinite impulse response (IIR) filter. The lack of coherence is especially visible in image regions that should stay static from one image to the next. Such regions usually show body parts of the user that do not move. Other image regions that show moving body parts can not be cross-faded without producing ghosting artifacts. Therefore, the IIR filter needs to be weighted by a map that describes the user's relative motion between two successive output frames. We compute each pixel's value of this weight map as the normed average length of the previous and current non-rigid registration offsets (see Figure 10). This effectively suppresses the filter response at image regions that are subject to strong user motion. To reuse the previous frame's output color and registration offsets correctly even during viewpoint motion, image warping needs to be applied. As a result, garment frame transitions triggered by user motion are less obvious.

(a) (b)

Fig. 10. (a) illustrates the weight map that is used for cross-fading while arms and legs of the user moved. Red indicates motion. (b) is a screenshot that shows how multiple overlapping garments can be combined.

5.3.4 *Detecting convergence*

So far we used a fixed number of iterations. However, a satisfying state may be reached earlier. Usually convergence is detected by checking the gradient length or evaluating the error function every other iteration. However, in our scenario the error function is rather noisy and flat due to the strong neighbor terms. Therefore, detecting convergence is more robust by only evaluating the data term. The algorithm stops iterating when the summed data term values do not improve for a certain number of iterations.

Detecting convergence is especially useful in combination with reusing the last frame's state for initialization: on top of frame coherence, the system's performance automatically increases when pose mismatches are small while still maintaining the ability to cover larger mismatches when necessary. In combination with the coherent initialization, we observed that the execution time of the optimization process can drop to 60% when the previous garment frame is reused.

5.3.5 *Loose clothing*

Loosely fitting garments break the assumption of silhouette similarity between the garment data worn by the model-user and the current user. The non-rigid registration may have problems with loose clothing: the algorithm tries to move clothes pixels towards the target silhouette. We addressed this issue by increasing the stiffness term of the energy function. This effectively preserves the cloth structure even for dresses or coats, see Figure 1(a).

The garment database lookup also assumes silhouette similarity between garment and user. For example, a long skirt results in large silhouette-matching errors in the leg regions. During our experiments with a long dress and a lab coat this did not break the matching algorithm, because all frames of the database suffered from the same error. Thus, the relative matching order between frames stayed valid.

Fig. 11. Mixed reality try-on: a user is augmented with a dress and rendered on a virtual bridge.

However, we can not generally guarantee that silhouette-based matching works for all existing garments.

5.4 Composing and display

Parts of the model-user are still visible in the garment buffer, but with alpha values that indicate which pixels have to be removed. These regions were required as a good optimization target during the non-rigid registration procedure. After optimization the unwanted pixels are removed.

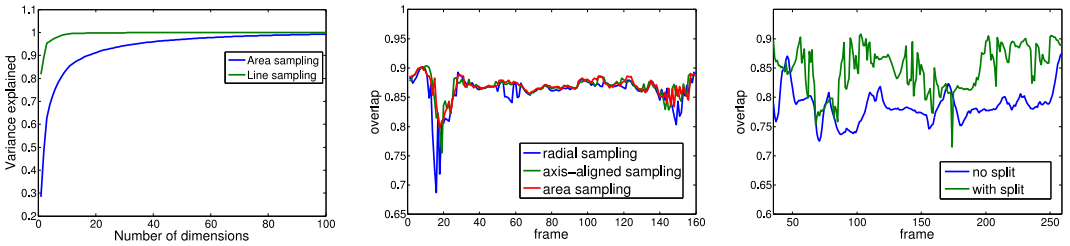
In a final step, the garment buffer and the current user's buffer are composed. Depth buffering can resolve the overlap. However, this can lead to unwanted effects when the user's body shape is wider than the garment. In such cases, the user's body is visible where it should not be. We allow for clothes to float on top of the user to avoid such wrong or noisy occlusions. As a result, the augmented garment is visible except in regions where the model user was removed. In these regions, the current user's body replaces the model user. This occlusion effect is correct, because the model-user occluded the same regions.

When multiple garments are augmented, the items are consecutively composed with the previous result. Figure 10(b) illustrates how a shirt and an open jacket can be combined despite the overlap. In this case, the composing starts by merging the user and the shirt. The result is used as the new image of the user and is composed with the jacket. The graphics API is used to display the result and enables a combination with conventionally rendered objects. Figure 11 shows such a mixed reality scenario.

6 Hardware and Implementation

Our setup includes ten color cameras that are connected to a single PC with three IEEE 1394b cards and transmit images at 15 Hz with a resolution of 640×480 pixels. At the PC the images are immediately loaded onto the GPU (an Nvidia GeForce GTX 480) where most of the subsequent processing is performed.

Our algorithms are designed to be executed on a single PC with a single GPU. This is important to avoid network latency and inter-GPU data transfers at runtime. We observed that the latency between the user's motions and display output can be irritating. The runtime stages are implemented as Nvidia CUDA kernels, which enabled us to formulate the computations without any graphics pipeline overhead.



(a) (b) (c)

Fig. 12. Different feature sampling schemes for silhouette-based matching. (a) shows how much variance in the data can be explained with a certain number of dimensions. In (b) quality is evaluated by pixel overlap. (c) shows the positive effect of extending the pose space for the dataset of Figure 5.

Storage of the garment database

For every database entry, all ten camera images are written to the hard disk using the PNG format with alpha channel for segmentation. The extracted features are also stored on the hard disk. Before the runtime phase starts, the images are copied onto a RAM-Disk, which maps a file handle to chunks of main memory to speed up the file access. Moreover, our system is capable of caching parts of the garment database on the GPU for further speedup.

7 Evaluation

We evaluated our methods in terms of visual quality and performance and show the improvement over our previous work [2]. The output resolution was fixed to 1000×800 pixels for all tests and the viewpoint set such that the rendered user is maximized on the screen. The chosen resolution is higher than the camera resolutions, because some cameras are closer to the user than the virtual viewpoint.

7.1 Sampling patterns for silhouette matching

In our first test, we evaluated different silhouette features in terms of the resulting matching quality. Features are extracted using radial, axis-aligned and area-based sampling. For a recorded sweater sequence, we measured the pixel overlap between the current user and the garment in each output image from a frontal viewpoint. The pixel overlap factor is computed as the fraction of user pixels that are covered with garment pixels. Figure 12 shows the results. Area-based sampling provides a 0.8% better average overlap than radial sampling. This rather small percentage corresponds to an approximate average of 900 pixels per frame, which is a clearly visible difference. However, while quality is better with area-based sampling, the feature vectors are less descriptive.

The dimensionality reducing effect of the PCA is therefore less effective for area-based sampling, which requires the subspace in which silhouettes are matched to have more dimensions than with the line sampling (radial or axis-aligned) approaches. Figure 12(a) shows that area-based sampling requires 100 search dimensions to become equally descriptive. During our experiments, we found that the number of dimensions that are used for searching must account for almost all (at least 99.9%) of the variance in the data to achieve a satisfying matching quality. To guarantee this we use 50 dimensions for the two line sampling approaches and 200 dimensions for area sampling. Due to its lower requirements, we favor the axis-aligned line sampling approach. It performs almost as well as area-based sampling (less than 0.1% average difference in this evaluation) at lower computational cost and therefore is a noticable improvement over our previous work [2].

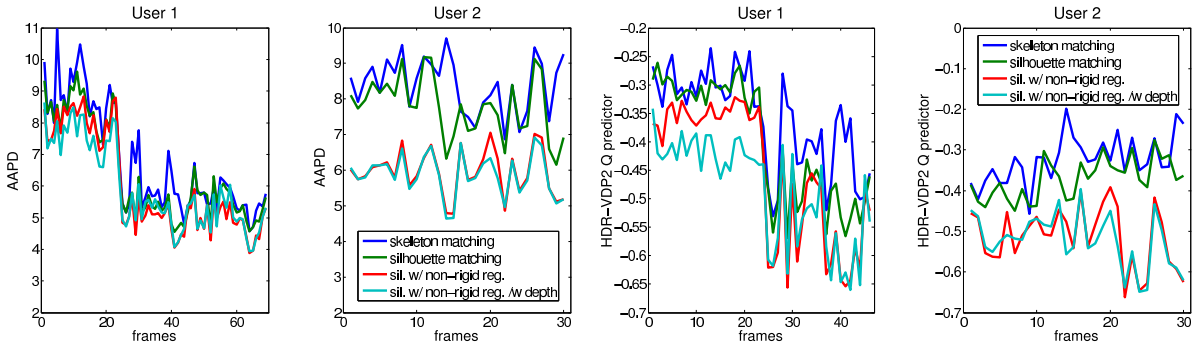
7.2 Pose space extension

To evaluate the usefulness of extending the pose space by splitting the silhouette image domains, we applied the same overlap factor metric as in the first test. For recorded garment data that has a sufficient collection of poses, this method of course does not improve the result. However, for the dataset of Figure 5, where poses are missing, we measured an average overlap improvement of 8.25%. This corresponds to an approximate average of 9000 pixels per frame, which is a considerable improvement over our previous work [2]. Figure 12(c) shows the overlap factor for the whole sequence. This proves the viability of the pose space extension method. However, there is an impact on the performance that amounts to an additional garment.

7.3 Overall quality and non-rigid registration

To assess the quality of the overall clothes augmentation system, we first created two ground truth datasets. We recorded two people with different body sizes: one male, 1.86 meters tall (user 1), and one female, 1.62 meters tall (user 2). Both performed the same body poses with and without the garment (a striped sweater) that is later subject to image-based clothes transfer. To obtain ground truth images, we manually labeled a set of frames from these videos where body poses match exactly. We also recorded the model-user who was 1.80 meters tall and wore the same sweater to generate the garment database.

Figure 13 shows the result of the evaluation for both people. We compared the rendered garment frames to the corresponding ground truth frames for two viewpoints: a frontal view and a side view. Two quality metrics are used: the average absolute pixel difference and the HDR-VDP2 quality predictor [36] that also accounts for the human



(a) (b) (c) (d)

Fig. 13. End-to-end quality evaluation of our approach for two users. A set of frames created by clothes transfer are compared to ground truth by means of average absolute pixel differences (AAPD) in (a) and (b) and the HDR-VDP2 quality metric in (c) and (d). Small values are better.

visual system and perception. We first compared silhouette-based matching to its major alternative: motion capture. For motion capture, we used the OpenNI skeleton tracking API[1], which can be used to track poses from depth sensors. To make the approaches comparable, we compute suitable depth maps from the visual hull. The motion capture results in the form of joint positions were normalized and matched using Euclidean distances just like the silhouette feature vectors. Results differ depending on the user's body pose: when arms and legs can be identified easily, the motion capture delivers similar results as the silhouette matching. However, for poses with arms fully touching the body, the tracking fails. See Figure 14(a) for an example. This proves the suitability of our approach. In addition, the plots show how the non-rigid registration improves the output quality. By adding the depth data term, we were able to further improve it over our previous formulation.

7.4 Performance

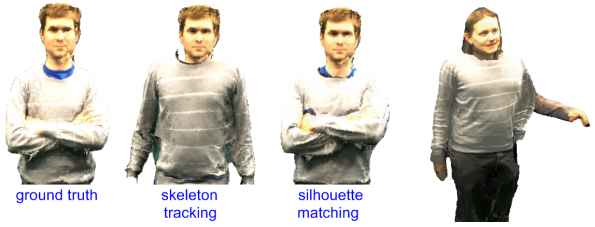
We measured the execution times of the runtime processes. One garment (a sweater) is transferred from one user to another. The garment database contains 275 frames, which in turn consist of 10 camera images. The evaluation system was equipped with a GeForce GTX 480 from Nvidia. Other PC components only have minor impact on the performance. The timings in Figure 15 are picked from a representative frame that required a reload of all camera images of the current user, and a reload of a frame from the garment database. The garment database was fully cached in GPU memory. Our implementation is not perfectly optimized, but runs at interactive rates for a single garment. We assume that the next hardware generation will be suitable for handling more than one item of clothing interactively.

8 Limitations and Future Work

The suggested clothes augmentation pipeline transfers the appearance of garments from one user to another. It uses captured images for rendering, which inherently contain features like realistic light-cloth interaction and wrinkles. It is suitable for a wide range of garments, like shirts, sweaters, trousers, dresses and accessories like scarfs, sunglasses etc. Moreover, recording the garment data is quick and cheap compared to manual 3D modeling. However, the proposed pipeline stages also have limitations that can reduce the visual quality that the system achieves.

We utilize IBVH rendering for image synthesis from silhouette images. It is an approximative surface reconstruction method that can not capture all concavities and requires a relatively high number of cameras to produce a satisfying output quality. As a result, artifacts might appear, especially when the user's arms cover other body parts. At the same time, IBVH rendering is a very quick method even at high geometry/camera resolutions, which makes it particularly suitable for interactive reconstruction systems. Moreover, the visual hull is a good geometric proxy for projective texturing, because it consists of large, smooth surface patches that can be textured with little distortions.

Both IBVH rendering and garment augmentation rely on image segmentation methods. Errors in the segmentation process lead to missing image regions or artifacts. However, due to the controlled background in our setup and control over what the model-user wears, this was not a problem during our experiments.



(a) (b)

Fig. 14. (a) shows a frame from the evaluation data set of user 1 of Figure 13, where skeleton- and silhouette-based matching strongly differ. (b) shows the visual impact of a missing pose in the garment database.

1. http://www.openni.org

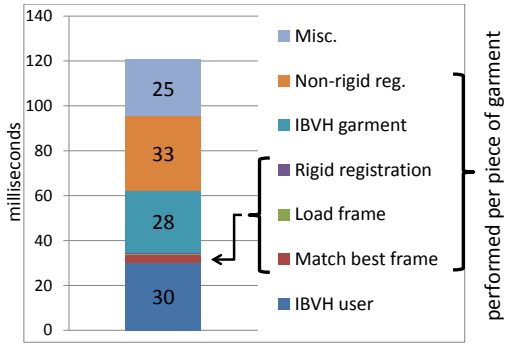


Fig. 15. Performance of the runtime processes.

Matching algorithms that are based on silhouette shapes assume a certain shape similarity between images of the model- and the end user. This is not always given. For example, when the desired garment is a dress and the current user is wearing trousers, then the shapes are not similar in the leg region. While silhouette matching is quite reliable for small dissimilarities, it loses precision with, for example, longer dresses. In practice, we achieved good matching results even for dresses and coats (see Figure 1(a)), but we can not generally guarantee that it works for arbitrary garment and user shapes.

Visual artifacts can originate from missing poses in the garment database, see Figure 14(b) for an example. However, such problems are alleviated by our method for extending the pose space, and can be further reduced by carefully instructing the model-user. In practice it takes only a few minutes to create a sufficient pose space record.

The suggested composing method allows the garment to float on top, while the user is only visible in image regions where the model-user was removed. It is not robust to bad matching or registration quality. In such cases, parts of the user might be covered by garment where they should not be. In future work, a robust segmentation method, like graph-cut, could combine foreground, color and depth information to find smooth borders between garment and user.

The scene lighting in the virtual try-on room should be constant to allow the recorded garments' appearances to fit to the current user. But even under static lighting, there might be subtle coherence issues. Figure 14(b) shows an example: the rendered sweater looks rather masculine, which is due to the fact that the model-user was male. The non-rigid registration morphs the shape to match the user, but does not modify the colors. Shape information that is conveyed through shading is therefore still present in the augmentation. Future work should preprocess the garment database with a deshading algorithm, and perform shading computations during augmentation to suppress unwanted shape cues. In addition, such an extension would alleviate the limitation to a constant scene lighting.

The suggested image-based approach is not suitable for simulating physical effects of user motion. For example, dresses can not swing realistically. On the other hand we do not require any physical simulation to produce nicely fitting clothes with realistic wrinkles and lighting, and avoid potentially unstable computations.

9 Conclusions

We introduced a method that allows users to see themselves wearing different clothes from arbitrary viewpoints while being able to move freely. The process is interactive and can be used for a virtual dressing room application. This work contributed efficient methods for silhouette matching, rigid- and non-rigid registration. Moreover, the clothes transfer pipeline was extended to reduce most of the shortcomings of previous approaches. We proposed to extend the pose space by combining several garment frames to improve the output quality. Moreover, loose or overlapping clothing can be handled. An efficient solver for the optimization problem of non-rigid registration was described and evaluated.

In contrast to prior work, we use low-level image processing algorithms. This allows us to circumvent critical parts like motion capture and 3D reconstruction. Our approach is suitable for a wide range of clothing and even combinations of garments. Moreover, our system achieves realism by using image-based rendering, which makes it a good alternative to manually modeled garments. Manual interaction is only required once per garment to define its segmentation mask, which is a particularly fast task given modern segmentation algorithms. Recording garments is also faster and cheaper than modeling, which makes us confident that in the future such a system can be deployed in a retail environment.

Acknowledgments

This work was supported by the Austrian Research Promotion Agency (FFG) under the BRIDGE program, project #822702 (NARKISSOS).

References

[1] A. Divivier, R. Trieb, and A. e. a. Ebert, "Virtual try-on: Topics in realistic, individualized dressing in virtual reality," in *Proc. of Virtual and Augmented Reality Status Conference*, Leipzig, Germany, 2004.

[2] S. Hauswiesner, M. Straka, and G. Reitmayr, "Image-based clothes transfer," in *Proc. of the 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Basel, Switzerland, 2011.

[3] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *CVPR*, 2009.

[4] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker, "Detailed human shape and pose from images," *Computer Vision and Pattern Recognition, IEEE Computer Society*, vol. 0, 2007.

[5] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–9, 2008.

[6] M. Straka, S. Hauswiesner, M. Ruther, and H. Bischof, "Rapid skin: Estimating the 3d human pose and shape in real-time," in *Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, 2012, pp. 41 –48.

[7] G. Cheung, S. Baker, J. Hodgins, and T. Kanade, "Markerless human motion transfer," in *ACM SIGGRAPH 2004 Sketches*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 31–. [Online]. Available: http://doi.acm.org/10.1145/1186223.1186262

[8] M. Richter, K. Varanasi, N. Hasler, and C. Theobalt, "Real-time reshaping of humans," in *Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission (3DIMPVT)*, 2012, pp. 340–347.

[9] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *CVPR*, 2010.

[10] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011.

[11] E. de Aguiar, L. Sigal, A. Treuille, and J. K. Hodgins, "Stable spaces for real-time clothing," in *ACM SIGGRAPH 2010 papers*. New York, NY, USA: ACM, 2010, pp. 106:1–106:9. [Online]. Available: http://doi.acm.org/10.1145/1833349.1778843

[12] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur, "Markerless garment capture," in *ACM SIGGRAPH 2008 papers*, ser. SIGGRAPH '08. New York, NY, USA: ACM, 2008, pp. 99:1–99:9. [Online]. Available: http://doi.acm.org/10.1145/1399504.1360698

[13] M. Salzmann, J. Pilet, S. Ilic, and P. Fua, "Surface deformation models for nonrigid 3d shape recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1481–1487, August 2007. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2007.1080

[14] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik, "Dynamic shape capture using multi-view photometric stereo," in *ACM SIGGRAPH Asia 2009 papers*. New York, NY, USA: ACM, 2009, pp. 174:1–174:11. [Online]. Available: http://doi.acm.org/10.1145/1661412.1618520

[15] C. Stoll, J. Gall, E. de Aguiar, S. Thrun, and C. Theobalt, "Video-based reconstruction of animatable human characters," in *ACM SIGGRAPH Asia 2010 papers*, ser. SIGGRAPH ASIA '10. New York, NY, USA: ACM, 2010, pp. 139:1–139:10. [Online]. Available: http://doi.acm.org/10.1145/1866158.1866161

[16] J. Li, J. Ye, Y. Wang, L. Bai, and G. Lu, "Technical section: Fitting 3d garment models onto individual human models," *Comput. Graph.*, vol. 34, pp. 742–755, December 2010. [Online]. Available: http://dx.doi.org/10.1016/j.cag.2010.07.008

[17] J. Ehara and H. Saito, "Texture overlay for virtual clothing based on pca of silhouettes," in *Proceedings of the 5th International Symposium on Mixed and Augmented Reality*, ser. ISMAR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 139–142. [Online]. Available: http://dx.doi.org/10.1109/ISMAR.2006.297805

[18] H. Tanaka and H. Saito, "Texture overlay onto flexible object with pca of silhouettes and k-means method for search into database," in *Proceedings of the IAPR Conference on Machine Vision Applications*, Yokohama, JAPAN, 2009.

[19] Z. Zhou, B. Shu, S. Zhuo, X. Deng, P. Tan, and S. Lin, "Image-based clothes animation for virtual fitting," in *SIGGRAPH Asia 2012 Technical Briefs*, ser. SA '12. New York, NY, USA: ACM, 2012, pp. 33:1–33:4. [Online]. Available: http://doi.acm.org/10.1145/2407746.2407779

[20] P. Volino and N. Magnenat-Thalmann, "Accurate garment prototyping and simulation," *Computer-Aided Design & Applications, CAD Solutions, Vol. 2, No. 5*, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.1310

[21] A. Weiss, D. Hirshberg, and M. Black, "Home 3d body scans from noisy image and range data," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 1951 –1958.

[22] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE Virtual Reality)*, 2012.

[23] J. Li and G. Lu, "Customizing 3d garments based on volumetric deformation," *Computers in Industry*, vol. 62, no. 7, pp. 693 – 707, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166361511000443

[24] A. Hilton, D. Beresford, T. Gentils, R. Smith, and W. Sun, "Virtual people: Capturing human models to populate virtual worlds," in *Proceedings of the Computer Animation*, ser. CA '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 174–. [Online]. Available: http://portal.acm.org/citation.cfm?id=791217.791567

[25] A. Hilsmann and P. Eisert, "Tracking and retexturing cloth for real-time virtual clothing applications," in *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques*, ser. MIRAGE '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 94–105. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01811-4_9

[26] P. Eisert, P. Fechteler, and J. Rurainsky, "3-d tracking of shoes for virtual mirror applications," in *IEEE Conference on Computer Vision and Pattern Recognition.*, June 2008, pp. 1 –6.

[27] I. Kemelmacher-Shlizerman, A. Sankar, E. Shechtman, and S. M. Seitz, "Being john malkovich," in *11th European Conference on Computer Vision (ECCV)*, 2010, pp. 341–353.

[28] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, "Video face replacement," *ACM Trans. Graph.*, vol. 30, no. 6, p. 130, 2011.

[29] M. Straka, S. Hauswiesner, M. Ruether, and H. Bischof, "A free-viewpoint virtual mirror with marker-less user interaction," in *Proc. of the 17th Scandinavian Conference on Image Analysis*, 2011.

[30] S. Hauswiesner, M. Straka, and G. Reitmayr, "Coherent image-based rendering of real-world objects," in *Symposium on Interactive 3D Graphics and Games*. New York, USA: ACM, 2011, pp. 183–190. [Online]. Available: http://doi.acm.org/10.1145/1944745.1944776

[31] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, "Image-based visual hulls," in *SIGGRAPH '00 proceedings*. New York, NY, USA: ACM Press/Addison-Wesley, 2000, pp. 369–374.

[32] P. Debevec, Y. Yu, and G. Boshokov, "Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping," University of California at Berkeley, Berkeley, CA, USA, Tech. Rep., 1998. [Online]. Available: http://portal.acm.org/citation.cfm?id=893689

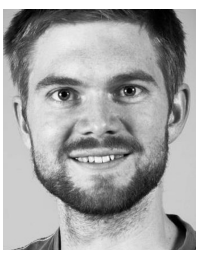
[33] M. Botsch and O. Sorkine, "On linear variational surface deformation methods," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 1, pp. 213 –230, jan.-feb. 2008.

[34] M. J. D. Powell, "Convergence properties of algorithms for nonlinear optimization," *SIAM Review*, vol. 28, no. 4, pp. pp. 487–500, 1986. [Online]. Available: http://www.jstor.org/stable/2031100

[35] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[36] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, "Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions," in *ACM SIGGRAPH*, New York, NY, USA, 2011, pp. 40:1–40:14. [Online]. Available: http://doi.acm.org/10.1145/1964921.1964935

Stefan Hauswiesner received his master's degree (2009) and PhD degree (2013) from the Graz University of Technology. He is research and teaching assistant at the Institute for Computer Graphics and Vision, Graz University of Technology. His research interests include image-based rendering and image processing in the context of mixed reality applications.



Matthias Straka received his bachelor's degree in 2007 and his master's degree in 2009 from the Graz University of Technology. He is currently working towards his PhD degree at the Institute for Computer Graphics and Vision, Graz University of Technology. His research interests include interactive 3D human shape and pose estimation from multi-view images with focus on applications for virtual dressing rooms and visual body measurements.

Gerhard Reitmayr is professor for Augmented Reality at the Graz University of Technology. He received his Dipl.-Ing. (2000) and Dr. techn. (2004) degrees from Vienna University of Technology. He worked as a research associate at the Department of Engineering at the University of Cambridge, UK until May 2009 where he was researcher and principal investigator. Research interests include the development of augmented reality user interfaces, wearable computing, ubiquitous computing environments and the integration of these. Research directions include computer vision techniques for localisation and tracking and interaction methods.