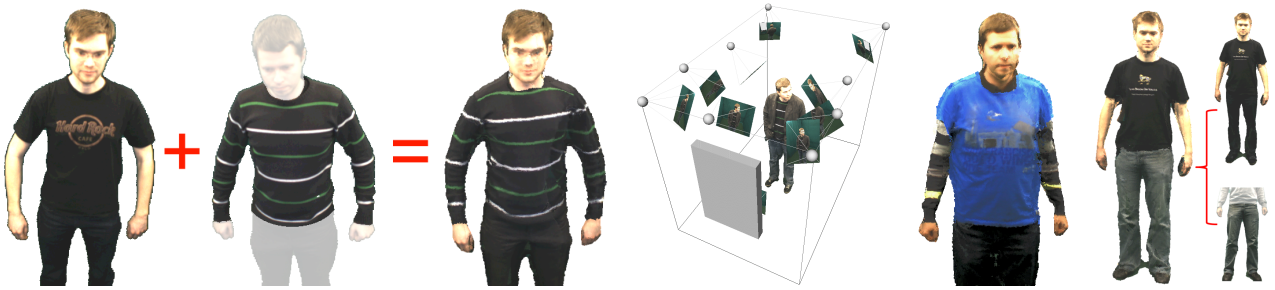
Image-Based Clothes Transfer

Stefan Hauswiesner*
Graz University of Technology

Matthias Straka†
Graz University of Technology

Gerhard Reitmayr‡
Graz University of Technology



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

Figure 1: (a) shows the concept of image-based clothes transfer: a user is dressed with the clothing of a previously recorded user. (b) shows an illustration of the virtual dressing room in which a user is captured and at the same time can see himself wearing different garments. (c) and (d) show results for a t-shirt over a sweater and jeans.

Abstract

Virtual dressing rooms for the fashion industry and digital entertainment applications aim at creating an image or a video of a user in which he or she wears different garments than in the real world. Such images can be displayed, for example, in a *magic mirror* shopping application or in games and movies. Current solutions involve the error-prone task of body pose tracking. We suggest an approach that allows users who are captured by a set of cameras to be virtually dressed with previously recorded garments in 3D. By using image-based algorithms, we can bypass critical components of other systems, especially tracking based on skeleton models. We rather transfer the appearance of a garment from one user to another by image processing and image-based rendering. Using images of real garments allows for photo-realistic rendering quality with high performance.

Keywords: augmented reality, image-based rendering, visual hull, virtual dressing room, CUDA

Index Terms: I.3.3 [Computing Methodologies]: COMPUTER GRAPHICS—Picture/Image Generation; I.3.6 [Computing Methodologies]: COMPUTER GRAPHICS—Methodology and Techniques

1 Introduction

In our mixed reality scenario, users inside a room equipped with cameras can see themselves on a large TV screen, which shows them wearing different clothing. Users are allowed to move freely inside the room and can watch themselves from arbitrary viewpoints. The system therefore needs to capture and render the user at interactive rates, and also augment his body with garments. Current solutions are either restricted to two dimensions, or rely on the complex process of 3D model-based tracking. Tracking humans is still a challenging task, and it likely fails when limbs are not visible or in unusual positions.

We propose an approach where a user can be displayed wearing previously recorded garments. These recordings can show a different person. By using image-based algorithms, we can bypass critical components of other systems, especially 3D reconstruction of garment models and pose tracking based on skeleton models. We achieve this by creating and querying a garment database. The database stores the appearance of a recorded garment and can be used to transfer it to the current user. Image-based visual hull rendering (IBVH) is used to render users and clothes from arbitrary viewing angles. By using images of real garments instead of virtual models, a photo-realistic rendering quality can be achieved with high performance. The approach is suitable for all kinds of clothing, and multiple pieces of garment can be combined. It does not need any physics simulation, because the effect of gravity and cloth stretch is already present in the images.

The contributions of this paper are a process for transferring clothes from one user to another. Section 3 gives an overview. It describes a very efficient method for GPU-based silhouette matching (section 5). Novel rigid and non-rigid registration methods (sections 6.1 and 6.3) are applied to multiple image-based renderings to fit the transferred clothes to the user.

2 Related work

Human pose tracking and clothes reconstruction from images have been studied extensively and comprise the main components of conventional virtual dressing applications. In addition to that, silhouette similarity computation and matching is an important component of the suggested approach.

Pose tracking and clothes reconstruction

We only consider optical, marker-less pose tracking, because any sort of markers is too obtrusive for virtual dressing. Pose tracking from multiple video streams [2] was used for animating and rendering persons. Tracking is difficult for certain poses, particularly human poses are likely to be ambiguous, even when many cameras are used.

Many virtual dressing applications draw a reconstructed garment mesh over a camera image. Recent reconstruction approaches do

*e-mail: hauswiesner@icg.tugraz.at
†e-mail: straka@icg.tugraz.at
‡e-mail: reitmayr@icg.tugraz.at

not require markers [1, 5]. They usually use a shape-from-stereo approach and apply complex processing to the data to account for occlusions.

Clothing and retexturing

An alternative to reconstruction and physical simulation is observation. Such systems work by finding the best matching dataset in a previously recorded database that contains all possible poses of the user [11]. However, like many other retexturing approaches it operates in 2D and therefore does not allow the user to view himself from arbitrary viewpoints.

The *Virtual Try-On* project [3] offers a set of applications for various tailoring, modeling and simulation tools. 3D scans of real garments are acquired by color-coded cloth. *MIRACloth* is a modeling application [12] which allows to create garments, fit them to avatars and simulate them. Both Virtual Try-On and MIRACloth do not include a mixed reality component that allows users to see realistic clothing on themselves immediately. Other virtual mirrors are restricted to specific tasks, like replacing logos or shoes [7, 4].

Selection of motion capture data based on multi-view silhouettes is shown in [9]. Other silhouette similarity metrics and acceleration structures [8] exist.

3 Approach

Our approach circumvents many of the problems above. It utilizes image-based rendering techniques and low-level image features to transfer clothes from one user to another. Similar to [11] the process has an offline and a runtime phase. The core stages of the process are:

1. A user wears a garment which should be transferred to other users. He enters a room equipped with cameras 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 database.
3. From now on, every user who enters the dressing room, can wear the recorded garment. 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 queried.
6. The selected pose and the captured user are rendered from the same viewpoint using image-based rendering.
7. Small pose mismatches are compensated by registration.

This process results in a composite mirror image, showing the user wearing a different piece (or pieces) of cloth. Our algorithms are designed to be executed on a single PC with a single GPU. This is important to avoid the latency of network and inter-GPU data transfers during runtime. We observed that the latency between the user's motions and display output can easily cause simulator sickness.

4 Offline: creating the garment database

A user puts on a piece or multiple pieces of garment 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. The virtual dressing room that is used for this project consists of a 2 by 3 meters cabin with green walls [10] (see Figure 1(b)). Ten cameras are mounted at the walls: two at the back and eight at the front and the sides. All cameras are focused at the center of the cabin, where the user is allowed to move freely inside a certain volume.

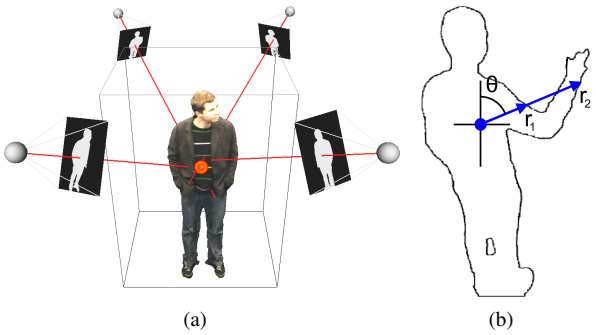


Figure 2: (a) shows how the 3D center of mass point C is approximated by unprojecting the 2D center of masses from the silhouette images. (b) silhouette features: for a set of angles, the distance between center-of-mass and silhouette edge are found.

Multiple cameras are required to allow arbitrary viewpoints during runtime. The model-user enters the dressing room and performs a series of different poses while being recorded. In each video frame, foreground regions can be easily segmented by background subtraction. Similar looking frames are of limited use for the later stages, because minor pose offsets are compensated by registration. To reduce the database size, many frames can therefore be skipped. Before insertion every new frame is checked for silhouette pixel overlap against the frames in the database to skip frames that are too similar.

All pixels that are not part of the desired piece (or pieces) of garment are segmented for later removal. The segmentation algorithm should be chosen depending on the expected garment colors. Currently, we use color keying and graph cut. While segmentation may fail, we have the advantage of being in an offline stage: we can try different parameters, or even perform a manually guided segmentation when necessary.

5 Matching recorded garment data

During runtime, the time-frame which contains the model-user's pose that is most similar to the current user's pose has to be found. To establish a match, we compare silhouette features, which is more robust than model-based tracking. The key factor for this technique is the silhouette similarity metric. When silhouette similarity, or distance, can be measured, standard algorithms can be used to search the resulting space. Our approach extends the work of [11]. We work on more than one silhouette image: usually four are sufficient. This additional input data is required to obtain a descriptive feature vector even when main features of the user, like his arms and legs, are located in front of the body and thus not visible in one or more silhouette images. Moreover, we use two silhouette intersection points instead of one to match arms and legs better. Figure 2 (b) gives an illustration of the process. Figure 2 (a) shows the spatial arrangement of the four most descriptive cameras.

To extract features, first the center of mass of each silhouette image is computed. From the center of mass, 360 regularly spaced directions are sampled to find the closest and second closest silhouette exit edges. 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. Four camera images with $360 * 2$ values each result in a feature vector of 2880 dimensions, which describe the pose of the model-user at a time-frame. All time-frames of the recorded videos are processed in that manner, resulting in a large database of pose descriptions. Using principle component analysis

(PCA), the space can be reduced to about 50 dimensions without losing distinctness.

All distances are normalized per image to be independent of scale factors. Invariance to translation is given by relating all distances to the center of mass. Rotational invariance on the other hand is not desired for our system, because we use a non-uniform distribution of cameras in the cabin. We therefore do not want to match poses that are oriented differently to avoid undersampling of important body parts.

6 Runtime phase

The database is stored on a RAM-Disk for quick access and can be cached in GPU memory. During runtime, features are extracted from the current silhouette images as described above. These features are transformed to PCA space by applying the transformation matrix from the offline stage. The pose entry with the smallest euclidean distance in the subspace is selected from the database. Since this space has a fairly small number of dimensions, a simple linear search is a sufficient search strategy.

6.1 Rigid registration

Most likely the matched time-frame shows the model-user in a slightly different position as the current user. The goal of this stage is to find a translation and a scale vector that aligns the rendering output of the user and the garment model. No reconstruction has been performed up to this stage, which means that no 3D data is available to determine the transformation by standard methods. We rather approximate translation and scale by extruding 3D rays from the 2D center of masses of the silhouette images and intersecting them.

While our system utilizes 10 cameras in total, four of them are mounted in the upper corners of the cabin (see Figure 2 (a)). Due to their distance to the user, the frustums of these cameras is sufficiently large to see the whole user during runtime. First, the 2D center of mass is extracted from each of the silhouette images of these four cameras. By transforming these 2D points with their corresponding inverted projection matrices we obtain a 3D ray for each camera. We assume that these rays intersect the 3D center of mass at some depth. To determine this depth, we intersect all four rays. The resulting point is a good approximation to the center of mass, which to determine otherwise would require a 3D reconstruction. To compute the translation, we subtract the 3D center of mass points of the garment and the current user.

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 to 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.

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

6.2 Rendering

In this phase, the matched garment dataset and the current user are rendered. Our system avoids explicit model representations, such as voxel grids or meshes. The image-based visual hull (IBVH) algorithm derives output images directly from the input silhouettes. We use the IBVH rendering implementation of [6] which also employs view-dependent texture mapping to find suitable color values for every output pixel.

6.3 Non-rigid registration

The last remaining difference to the desired output image is usually a small pose inconsistency. Large inconsistencies have already been removed 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. To compensate for this minor lack of overlap, the non-rigid registration is started.

The image that is rendered from the garment database and the rendered image of the user are input to an optimization procedure, which aligns their silhouettes for final output (Figure 3 (a)). The domain of the problem can be formulated as an energy function that should be minimized. We use pixel locations as input to allow the garment to shrink or grow and to retain the spatial arrangement of pixels.

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

$$E_{total} = \sum E(x,y)$$

Energies consist of a data term, which is 0 when the particle is located on the target silhouette, and > 0 otherwise. A neighbor-term keeps particles roughly in their initial spatial arrangement.

$$E(x,y) = \alpha * E_{data}(x,y) + \beta * E_{neighbors}(x,y)$$

$$E_{data}(x,y) = (1 - I_{target}(x,y))^2$$

The data term E_{data} describes the overlap between garment and user. Its derivative therefore pushes garment pixels towards the user's body. The smoothness term $E_{neighbors}$ 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 $E_{neighbor_{u,v}}$.

$$E_{neighbor_{u,v}}(x,y) = \gamma * E_{compress_{u,v}}(x,y) + \eta * E_{stretch_{u,v}}(x,y)$$

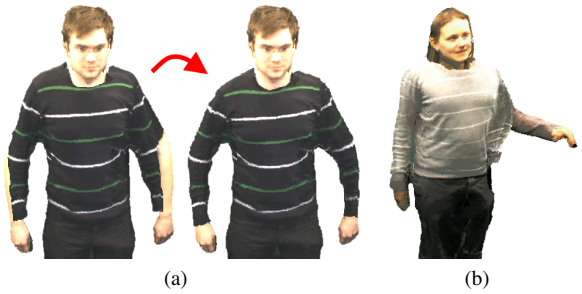
$$E_{compress_{u,v}}(x,y) = (min(\sqrt{(x-u')^2 + (y-v')^2}, d) - d)^2$$

$$E_{stretch_{u,v}}(x,y) = (u-u')^2 + (v-v')^2$$

u' and v' are the current x and y coordinates of the neighbor, and u and v are its initial coordinates relative to x and y. The term $E_{compress_{u,v}}$ tries to keep the neighbor at a minimum distance d, while the term $E_{stretch_{u,v}}$ tries to keep it at its initial relative position. This regularization is important to retain features of the garment, like stripes and printed logos. The weight terms α and β control the ratio between *stiffness* of the garment output and convergence speed. γ and η scale the penalties for compression and stretch of the rendered garment.

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. The used optimization method should allow for a large amount of dimensions, which prohibits any non-sparse methods. We currently use gradient descent because it is simple, well parallelizable and has a low memory footprint.

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, a low resolution copy of the silhouette is computed and both are smoothed by convolution with a Gauss kernel. The low resolution level has only $\frac{1}{16}$th of the output resolution. On both levels, the x and y gradients are precomputed because these



(a) (b)

Figure 3: (a) shows the effect of non-rigid registration: the garment adapts to the user's shape and pose. (b) shows the visual impact of a missing pose in the garment database: the current user's left arm can not be covered by the transferred sweater.

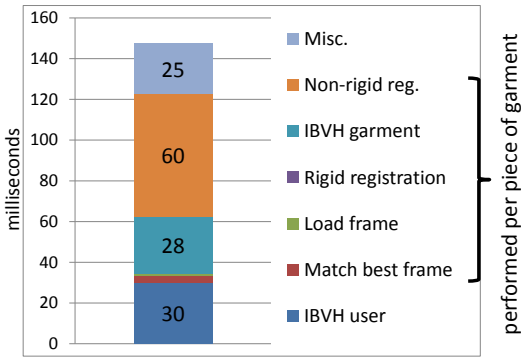


Figure 4: Performance of the suggested runtime processes.

terms are static during the optimization. The optimization procedure itself consists of a repeated execution of two steps: computing the derivatives at each garment pixel location in x and y direction, and updating the pixel location in direction of its derivatives. First we execute 400 iterations with large step lengths on the low resolution target silhouette. Then, 20 iterations with decreasing step lengths are executed in full resolution to remove overlapping pixels. The difference 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 20x20 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 every frame.

6.4 Final display

After non-rigid registration the segmentation information from the garment database is used to remove pixels of the model-user from the output. These regions were required during optimization as part of the data term. In a final step, the garment buffer and the current user's buffer are blended. We allow for clothes to float on top of the user to avoid wrong or noisy occlusions. The resulting occlusion effect is correct for most viewing angles, because the model-user occluded the same regions. The blended buffer is passed to the graphics API for display.

7 Results

We measured the runtime of the online phase for an output resolution of 1000x900 pixels and for a single garment on a system equipped with a GeForce GTX 480 from Nvidia. While our system is not perfectly optimized, it achieves interactive frame rates. See Figure 4 for timings. The suggested approach offers a visual quality that is close to the original video data (see Figure 3 (a)). However, it also has disadvantages: poses may be missing in the garment database (see Figure 3 (b)) and physical simulation, for example, a swinging dress, is not possible.

8 Conclusions and Future Work

The proposed approach uses low-level image processing algorithms and image-based rendering to produce appealing images in which a user can see himself wearing different clothes. This is achieved by matching previously recorded garment data to the current user's pose. By avoiding model-based tracking, the suggested approach is robust, interactive and suitable for all kinds of garments. A combination of algorithms is required during both the offline and the runtime processes. This paper contributes efficient methods for silhouette matching, rigid- and non-rigid registration.

Future work on the subject should primarily focus on increasing the output quality systematically. The approach would also benefit from a way of separating a person's shape from its dressed shape, but without the robustness issues of model-based tracking.

Acknowledgements

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

References

[1] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 99:1–99:9, New York, NY, USA, 2008. ACM.

[2] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Trans. Graph.*, 22(3):569–577, 2003.

[3] 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.

[4] P. Eisert, P. Fechteler, and J. Rurainsky. 3-d tracking of shoes for virtual mirror applications. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –6, 2008.

[5] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. *Int. J. Comput. Vision*, 81:53–67, January 2009.

[6] S. Hauswiesner, M. Straka, and G. Reitmayr. Coherent image-based rendering of real-world objects. In *Symposium on Interactive 3D Graphics and Games*, pages 183–190, New York, USA, 2011. ACM.

[7] 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 Collaboration Techniques*, MIRAGE '09, pages 94–105, Berlin, Heidelberg, 2009. Springer-Verlag.

[8] D. Mohr and G. Zachmann. Silhouette area based similarity measure for template matching in constant time. In *Proceedings of the 6th international conference on Articulated motion and deformable objects*, AMDO'10, pages 43–54, Berlin, Heidelberg, 2010. Springer-Verlag.

[9] L. Ren, G. Shakhnarovich, J. K. Hodgins, H. Pfister, and P. Viola. Learning silhouette features for control of human motion. *ACM Trans. Graph.*, 24:1303–1331, October 2005.

[10] 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 (SCIA)*, 2011.

[11] 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.

[12] P. Volino and N. Magnenat-Thalmann. *Virtual clothing: theory and practice*. Number v. 1. Springer, 2000.