Free Viewpoint Virtual Try-On With Commodity Depth Cameras

Stefan Hauswiesner∗ Grazer University of Technology
Matthias Straka† Grazer University of Technology
Gerhard Reitmayr‡ Grazer University of Technology

Abstract

We present a system that allows users to interactively control a 3D model of themselves at home using a commodity depth camera. It augments the model with virtual clothes that can be downloaded. As a result, users can enjoy a private, virtual try-on experience in their own homes. As a prerequisite, the user needs to enter or pass through a multi-camera setup that captures him or her in a fraction of a second. From the captured data, a 3D model is created. The model is transmitted to the user’s home system to serve as a realistic avatar for the virtual try-on application. The system provides free-viewpoint high quality rendering with smooth animations and correct occlusion, and therefore improves the state of the art in terms of quality. It utilizes cheap hardware and therefore is affordable for and accessible to a wide audience.

1 Introduction

Due to widely available computing power and fast internet access, online garment shopping has become popular. However, web shops that only provide simple apparel viewing mechanisms do not give customers sufficient information to evaluate size, fit and color of garments. This leaves the customers unsatisfied and as a consequence product return rates are high [Cordier et al. 2003]. Virtual try-on applications attempt to alleviate this problem by giving users the opportunity to examine garments more thoroughly in an interactive, virtual environment.

Inside such a virtual environment, a 3D model of a person wears the desired garment. Realistic models can be obtained by 3D reconstruction from calibrated camera images. However, such systems often require complex and expensive system hardware and the result can not be consumed interactively at the user’s home PC.

We suggest a system that reconstructs users within seconds. It only requires them to walk through a multi-camera system, which can be placed, for example, at a shop entrance. Then, the user can take the reconstructed model - his virtual avatar - home and interact with it on his computer. Users can select and download garment models from a website, or take it with them after their shop visit. These models can be used to augment the avatar, while the avatar can be controlled with tracking data from the depth camera. This allows users to augment themselves with new garment inside their home environment. Users can move in front of the camera, and turn the virtual viewpoint to see themselves from arbitrary viewing angles.

Users are only required to have a standard PC and a depth camera, such as the Microsoft Kinect. It is easy for users to even improve the experience by attaching a video projector or a large TV to their PC. The required hardware at the client’s home is cheap and therefore widely available. Our approach does not require the depth camera to be calibrated in any way - it can be used out of the box. Therefore, no special technical skills are required. Moreover, our system allows to render virtual clothes, which are only available as 3D models. It can combine virtual and reconstructed garments seamlessly.

The contribution of this work is an affordable system that allows free-viewpoint image-based rendering of a personalized avatar with clothes augmentation to be deployed at users’ homes. It builds on state-of-the-art technology and introduces a novel combination of a multi-camera system with widely available depth cameras. It is suitable for all kinds of clothes, and has several advantages for the user: it is private, it is time-efficient when trying on a number of different clothes, it utilizes cheap hardware at the client side and is easy to set up. It offers free viewpoint navigation in contrast to a conventional mirror and is easily expendable by attaching a TV or projector. Compared to conventional web-shops, users can get a better impression of the garments.

Figure 1: An illustration of the concept of our system. A user is recorded at a shop entrance. He can take his avatar home, and view and interact with it on his own computer equipped with a depth camera. The user’s avatar wears augmented, reconstructed or virtual garments to create a comfortable, private try-on experience.
2 Related work

Human pose tracking and clothes reconstruction from images have been studied extensively and comprise the main components of most virtual try-on applications.

Pose tracking

Human pose tracking is the task of determining a user’s articulated pose. It usually involves a pose and shape model that is fit to sensor data and therefore comprises a model-based tracking problem. We only consider optical, marker-less pose tracking, because any sort of markers is too obtrusive for try-on applications.

Pose tracking from multiple video streams [Gall et al. 2009; Balan et al. 2007; Vlasic et al. 2008] is used for animating and rendering persons. 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. [Ganapathi et al. 2010] have shown how to track full body motions using a time-of-flight camera. The more recent Microsoft Kinect camera 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 [Shotton et al. 2011].

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 existing CAD models [Protopsaltou et al. 2002].

When models are not available they have to be reconstructed. Capturing clothes from a video sequence is a hard task, especially because of occlusions and the non-rigidity of cloth. Many approaches use markers on the cloth for capturing, which makes them less suitable for our application. More recent approaches do not require markers [Bradley et al. 2008; Hasler et al. 2006; Salzmann et al. 2007; Gay-Bellile et al. 2007; Furukawa and Ponce 2009; de Aguiar et al. 2008; Furukawa et al. 2000; Scholz and Magnor 2004]. 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 [Vlasic et al. 2009] or a laser scanner [Stoll et al. 2010] this limitation can be removed, but such hardware is expensive and processing cannot 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 and computationally demanding problem [Li et al. 2010; Meng et al. 2011].

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. One such system is [Ehara and Saito 2006]. It searches a database of poses by using a silhouette similarity metric and uses the best match to deform a texture to fit the user. 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 [Divivier et al. 2004] 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 cloth modeling application [Volino and Magnenat-Thalmann 2005] which allows to create garments, fit them to avatars and simulate them. A similar system is shown in [Meng et al. 2010]. Physical simulation of virtual clothes is also described in [Vassilev 2000]. These approaches 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 placing logos or shoes [Hilsmann and Eisert 2009; Eisert et al. 2008]. Model-based tracking, reconstruction and retexturing is used to render dressed people [Hilton et al. 1999]. However, this system does not allow to change clothes.

Web-based systems

Web-based systems allow users to watch and interact with garment models at home by using web-browsers. Even interactive cloth physics simulation can be deployed through the browser [Chittaro and Corvaglia 2003]. However, displaying garments on virtual avatars that do not resemble the user’s appearance are of limited use. Adapting a virtual avatar’s shape and color to the user’s appearance shows great potential to improve this technology [Cordier et al. 2001; Cordier et al. 2003; Magnenat-Thalmann et al. 2004]. Our approach extends this concept and provides the user with a completely textured 3D copy of himself that can be controlled by his own body.

Proprietary systems

Imagine That1 shows a simple overlay of virtual clothes over a video stream, similar to the Try On Bathing Suit2. Metaio3 also developed a static overlay system without tracking. These systems assume that the user stands in a fixed position in front of the camera, and therefore severely limit the usefulness. AITech4 developed a magic mirror with overlays that adapt to the user, but are only 2D.

The companies AR Door5 and Topshop use a TV screen and a Microsoft Kinect for tracking, but do not reconstruct the user to let him use the application at home. Another system is called Swivel from FaceCake6, which has similar limitations. Zugara’s Webcam Social Shopper7 uses markers and static overlay, which is not suitable for our approach.

3 Approach

Our approach consists of two major parts (see Figure 2). First, the user needs to be captured by a set of surrounding cameras in order to generate a virtual avatar that can be rendered from any viewpoint (including the back of the user). Such a multi-camera setup can be placed, for example, at a clothing store. Reconstruction only needs to be performed once. After the user has been captured by all cameras, a human mesh model is deformed non-rigidly to fit the camera images. Garments are captured using the same technology: a person wears the desired garment while being captured by the same hardware that is used for user reconstruction. However, after reconstruction, all parts of the model other than the desired clothing are segmented and removed. This is necessary to avoid that the user’s body parts will be overlapped by parts of the person who

\[1\] http://www.imaginethattechnologies.com
\[2\] http://www.tryonbathingsuit.com
\[3\] http://www.metaio.com
\[4\] http://www.magic-mirror.es
\[5\] http://ar-door.com
\[6\] http://www.facecake.com
\[7\] http://www.webcamsocialshopper.com
wore the garment. Note, that the person recording the garment does not have to be the user himself. Rather, one garment recording can be re-used for all users. Section 4 describes this part in detail. Both the user’s avatar and a number of garment models can be transferred to the user’s personal computer.

The second part of our system is located at the user’s home. It consists of the user’s PC equipped with a depth camera and our software. The depth camera data is used to track the user’s body. The tracking data is used to deform the user’s reconstructed 3D avatar such that it reflects the current user’s body pose. The avatar can then be rendered and augmented with additional clothes. Section 5 describes how garments are generated and displayed.

4 Reconstruction

This section describes the reconstruction algorithms that generate the user’s 3D avatar out of a set of images from calibrated cameras.

4.1 The 3D avatar

For rendering an image of the user and augmenting him or her with different clothes, a virtual avatar is needed. Since the system should be executable on a low-budget platform, the most natural choice is a textured triangle mesh which can be adapted to the current pose of the user. We propose to use a mesh from the SCAPE database [Anguelov et al. 2005] which consists of laser-scans of human bodies. The SCAPE database only provides static meshes and therefore does not allow animations with smooth transitions between poses. A human body is usually animated by transforming its underlying skeleton, which is a graph that consists of nodes (joints) and edges (bones). We apply a rigging algorithm [Baran and Popović 2007] on the mesh which calculates the skeleton and skinning weights for all vertices automatically.

In order to modify the pose of the avatar, we change the underlying skeleton by transforming the joint positions. We determine the current joint positions of the user by a human pose tracking algorithm, such as provided by the OpenNI framework [OpenNI 2011]. The skeleton of the avatar is then transformed: For each joint, we determine the transformation \( T_i \) that aligns the SCAPE skeleton with tracking data. Note that the skeleton models are defined slightly differently, which requires a mapping between the OpenNI and SCAPE skeleton. By applying linear blend skinning [Baran and Popović 2007], we transform each vertex of the surface mesh from its initial position \( V_i \) to the current pose of the user:

\[
V'_i = \sum_i w_i \cdot T_i \cdot V_i
\]  

where \( w_i \) is the skinning weight for bone \( i \). The whole process of skeleton transformation and skinning is illustrated in Figure 4 where human pose tracking is performed using a depth camera.

4.2 Adapting the avatar to the user

As mentioned in the previous section, we apply a skinning algorithm to a static mesh and transform its skeleton so that it has the same pose as the user. This allows for a realistic animation where the user can control the avatar as if it was his own mirror image. However, changing only the pose of the avatar is not sufficient to generate a realistic representation of the user. For this reason, we deform the SCAPE mesh so that it adapts to the shape of the user. This could be achieved by a full body scanner such as a laser scanner or a structured light system [D’Apuzzo 2007]. But such systems have the disadvantage that the scanning process takes up to several seconds where the user is requested to stand perfectly still. Multi-camera setups such as [Straka et al. 2011] have the advantage that multiple views of the user are captured at the same time, which enables a body scan within fractions of a second. Such a body scanning process only needs to be performed once and the scanner can be placed in a clothing store, for example.

Our multi-camera setup consists of a 2 × 3 meter cabin where ten cameras are mounted on the walls: two at the back and eight at the front and the sides. All cameras are pointed towards the center of the cabin, where the user is allowed to move freely inside a certain volume (see Figure 1). The cameras are connected to a single PC via IEEE 1394b and capture images with a resolution of 640×480 pixels. All cameras are calibrated intrinsically and externally so that their projection matrices \( P_i \) are known. Due to a controlled green background, it is easy to segment the user in all views by simple color background subtraction. The resulting silhouette images are used to extract the shape of the user as an 3D mesh. Color information is also used to generate a texture for the virtual avatar.

The process of body scanning (the reconstruction) is equivalent to deforming the mesh so that its projection into all camera images produce an optimal overlap with the silhouette of the user (see Figure 3). Since the SCAPE database consists only of static poses, we need to first transform the position, size and pose of the skinned SCAPE mesh to match the current pose of the user inside the scanner. To determine the current pose, we use the pose estimation algorithm included in the OpenNI framework [OpenNI 2011]. Since this algorithm requires a depth map as input, we cannot use the multi-camera images directly. We propose an intermediate step that does not require a dedicated depth camera in order to estimate the human pose within our multi-camera setup. We simulate a virtual Microsoft Kinect camera by rendering a depth map of the visual hull of the user, seen from a viewpoint that would be expected from
a real Kinect setup, and apply the skeleton tracking algorithm on this data. This has the advantage that the calibration of the virtual camera is completely known and no additional hardware is needed. The OpenNI pose estimation algorithm estimates the joint positions of all limbs of the user with great precision. This data allows us to perform an initial alignment of the skinned SCAPE mesh with the true pose of the user.

The aligned SCAPE mesh is then deformed using the Laplacian deformation framework [Botsch and Sorkine 2008]. Similar to [Gall et al. 2009; Vlasic et al. 2008] we project the mesh into all camera views and move the rim vertices (vertices that should project onto the silhouette contour) towards the closest silhouette contour of the corresponding camera view (see Figure 3). Since we can not determine the 3D position of these vertices in 2D silhouette images directly, we allow rim vertices to move freely along the viewing rays from the camera’s projection center through silhouette contour pixels. The Laplacian deformation framework allows to globally optimize the positions of the deformed vertices \( V \) using all constraints in a single linear least-squares system:

\[
V' = \arg\min_V \left( \|L V\|^2 + w_{C} \|C_{uv} V - q_{uv}\|^2 \right)
\]  

where \( L \) is the Laplacian matrix of the mesh which enforces that the mesh maintains its shape (we use cotangent weights). It is modified so that local rotations and scale changes of the mesh are handled implicitly during optimization [Botsch and Sorkine 2008]. The matrix \( C_{uv} \) and the vector \( q_{uv} \) encode the constraints that force rim vertices to lie on viewing rays of the corresponding camera. Each such constraint is represented through two linear equations:

\[
(N^c_i - v_{i,u}N^c_i) V_i = -T^c_i + v_{i,v}T^c_j \quad (3)
\]

\[
(N^c_i - v_{i,v}N^c_i) V_i = -T^c_j + v_{i,v}T^c_j \quad (4)
\]

In the equations above the \( 3 \times 4 \) projection matrix \( P \) of camera \( c \) is split into its translational vector \( T^c \) and remaining \( 3 \times 3 \) rotational part \( N^c \) \((P = [N^c|T^c])\). \( N^c_i \) denotes the first row of matrix \( N^c \). \( v_i = (v_{i,u}, v_{i,v}) \) represents the screen space coordinate of the closest silhouette contour pixel to vertex \( V_i \) when projected into image \( c \). We only consider vertex-contour pairs that have a similar normal vector in image space and that are at most 20 pixels apart. Changing the weighting parameter \( w_{C} \) allows to balance the amount of deformation allowed on the mesh and the regularization. When \( w_{C} \) is chosen too strong, the mesh over-adapts to silhouette data and therefore becomes similar to the visual hull. This results in unnatural looking bodies. Similar to [Gall et al. 2009] we propose to use multiple iterations of the Laplacian deformation procedure with smaller weights and to recalculate vertex-silhouette correspondences at each iteration.

By solving equation (2), we obtain new vertex positions \( V' \) for the mesh. Note that even though the system of equations can become quite large (larger than three times the number of vertices squared), the resulting matrix is very sparse and an optimal solution can be found efficiently.

The output of this stage is an adapted SCAPE mesh that fits the image data and therefore resembles the user. Since we only modify the position of vertices slightly, the associated skinning weights remain valid and the resulting mesh can be used to generate new poses by modifying the underlying skeleton. It can therefore be used at the user’s personal computer to render and texture the avatar of the user as well as pieces of clothing.

5 Virtual try-on at home

During the virtual try-on session at the user’s home, the expensive multi-camera setup is no longer needed. A depth camera, in our case a Microsoft Kinect, delivers depth maps at a resolution of 320 × 240 pixels at 30 Hz. From these depth maps, the user’s pose can be determined by using the OpenNI library [OpenNI 2011] or [Shotton et al. 2011]. Using the tracked joint positions, we transform the adapted virtual avatar (see Section 4.2) into the current pose of the user at each frame using the skinning method described in Section 4.1. See Figure 4 for an illustration of this process. In addition, one or multiple garment models that the user has selected are transformed in a similar way. Finally, the avatar is rendered together with the garment.

5.1 Garment generation

Garments are reconstructed from images of our multi-camera installation by using the same methods as described above: a person, for example, an employee of the clothing store, enters the multi-camera space while wearing the desired piece of clothing. The user’s pose and shape are reconstructed by deforming a SCAPE mesh to fit the images. Each garment therefore consists of a mesh and a set of camera images for texturing.

However, since we do not want to see the user who was wearing the garment for reconstruction, we need to segment the desired garment. Currently, we perform this segmentation either manually,
or by simple chroma keying of the camera images. The per-pixel segmentation is stored in the alpha channel of the images. During rendering, these images are bound as textures, so the segmentation can be queried and fragments that are not part of the garment can be discarded. Moreover, all triangles of the mesh that do not contain garment texels are removed from the mesh. While these triangles are invisible during rendering anyway, they would interfere with the physics simulation of the cloth.

5.2 Garment selection

Each garment consists of a mesh and a set of images. The files can be grouped by compressing them to a single archive and can be made available through, for example, a website or a web shop. A rendered preview can help the user to pick garments. We currently do not make any assumptions on how the files are distributed - the involved decisions depend on the desired business model. At the user’s PC the unpacked files can be loaded by our application.

5.3 Garment adaptation

The reconstructed garment model and the user’s avatar model are in full correspondence: all vertices represent the same body regions in both models. However, the models differ in shape and pose. The user wearing the garment for reconstruction, and the end-user are generally not the same person and are not required to perform the same poses during reconstruction. See Figure 5 for an example.

During user interaction, the models are deformed to the same pose (the pose of the current user that is tracked) by skinning the garment model to the user’s skeleton. This way, the overall scale and translation of the models match. However, shapes do not match yet. Parts of the garment may be hidden below the surface of the user, because the garment model was reconstructed while being worn by a different person. We do not want to compensate for the shape mismatch completely, because this would distort the garment’s appearance. Instead, we only want to compensate for rough differences. To do so, we suggest a non-rigid deformation of the garment mesh. Garment vertices are translated in direction of the vertex normal until they are on top of the user’s surface. The resulting per-vertex offsets are computed at the program start and applied to the garment mesh during runtime after skinning.

5.4 Physics simulation

In addition to linear-blend skinning, our system also features a physics simulation that is based on the Nvidia PhysX® SDK. Due to hardware acceleration, it is suitable for simulating cloth at interactive rates. To achieve correct collision detection between pieces of garment and the user’s body, we extract collision proxies from the SCAPE mesh. At the program start, all vertices that belong to a body part are grouped. For the vertices of each body part, a convex hull is computed. The convex hull is then passed to PhysX as a convex collision geometry. During runtime, the collision check for convex meshes can be performed more efficiently than for general meshes. Since body parts are mostly convex by nature, this does not reduce the quality of the simulation.

In our system, we usually employ skinning to deform the mesh of the user itself and meshes of tight fitting clothes. Physics simulation is usually applied to loosely fitting cloth, such as dresses or long skirts. The resulting, deformed triangle meshes are input to the next stage: the rendering.

5.5 Rendering and texturing

The triangle meshes of the user and the garments are rendered using standard OpenGL calls. However, the texturing step is performed by custom shader programs.

Our system has two separate coordinate systems: the one of the multi-camera capturing device at the shop, and one at the user’s home defined by the depth camera hardware and configuration. We do not require the user to calibrate the camera coordinate system, so we can not safely make use of it. We rather rely on mesh correspondences: each vertex and each triangle of the user’s avatar mesh describes the same body feature as the corresponding vertex or triangle in the garment mesh. To obtain correct position and coloring,

Figure 5: (a) shows the user. (b) illustrates the problem of an unadapted garment. (c) incorporates adaption.

we need to pass the vertex positions and normal vectors of both coordinate systems to the shader programs.

The vertex shader projects each vertex position defined in the depth camera coordinate system by using the current model-view and projection matrix. This allows the user to freely watch his avatar from any direction, and allows for arbitrary body poses of the avatar. At the same time, the vertex program passes the untransformed positions and normals of the multi-camera capturing device to the fragment program. Moreover, the camera images from the multi-camera setup are bound as textures.

For each fragment, the passed vertex position is projected into the image planes of two cameras that have a viewing angle most similar to the inverted normal. This projection does not redirect the location of the output, it is rather just used to determine the color at the fragment location. The color values at the projected locations are queried, blended and returned as output, thus resulting in a projective texture mapping scheme [Debevec et al. 1998].

5.6 Alternative modes of operation

The common use case of our system utilizes both the multi-camera capturing device and a depth camera at the user’s site. However, our system also supports alternative modes.

For users who have not been captured by the multi-camera system, we can provide virtual avatars that are fully configurable. To achieve this, our system can import the MakeHuman® MHX file format. The freely available MakeHuman application allows users to create and modify human avatars by specifying a wide range of parameters, like gender, body part proportions, facial features etc. Figure 6 (b) shows a virtual avatar.

Users who do not own a depth camera can also use our system. Body tracking can be entirely replaced by replaying motion capture data. To achieve this, our system can either replay the captured data from the user when he was inside the multi-camera system, or load ASF/AMC files. These files are a common motion capture data exchange format, and many tracks can be downloaded from the Carnegie Mellon University Motion Capture Database[9]. Also for users who own a depth camera this might be an entertaining feature, because some of the available motion tracks are highly athletic.

Moreover, our virtual try-on system can also run entirely at the shop, or show the user who is currently captured a preview on what is possible at his home PC. Our multi-camera setup is equipped with a 42” TV screen, which can be used to show the user’s mirror image immediately [Straka et al. 2011] from arbitrary viewpoints. See Figure 6 (a) for an example. After reconstructing the 3D avatar of the user, we can use the cameras to generate a depth map by image-based visual hull rendering [Hauswiesner et al. 2011]. This depth map can be used to track the user by utilizing the OpenNI SDK. The tracked user can be augmented with garments in the same way as described above.

Another extension of the original concept are garment models that exist only virtually, for example, as meshes from CAD models. Similar to reconstructed clothes, virtual clothes can be used for augmentation. Figure 6 (b) shows a virtual avatar wearing virtual clothes. They consist of textured meshes, but do not have semantic information on how to drape them to a human body. To build this information, we align the meshes with the SCAPE template model by using a mesh editing tool. Then, the skinning weights of each garment vertex is derived from the closest vertices in the template mesh. These weights are stored in a file that is part of the garment model, and is therefore also delivered to the user. During runtime at the user’s PC, the mesh is skinned exactly like the reconstructed meshes described above.

6. User feedback

We created a survey to gauge the interest in a virtual try-on system and to validate our approach for a virtual try-on for home use. The survey was web-based and asked participants about their interest in fashion, online shopping and augmented reality, and about their prior experiences with these topics. Pictures and videos of the suggested system were shown to the participants, before they were asked specific questions that give use feedback on the acceptance of the approach.

The survey was answered by 41 users aged 23 to 44 years. 8 participants were female, 33 were male. All but one participants were familiar with computers in general, and 32 of them had prior knowledge in augmented reality. From these demographics we can not make conclusions about the general population. However, 28 persons stated an average or above average interest in fashion, so we assume that the results are representative for technology aware early adopters of the suggested approach.

6.1 Questions and results

First, the participants were asked to rate their interests according to a 5-point Likert scale with 1 = very interested and 5 = not interested. See Figure 7 for detailed questions and answers.

Then, the participants were asked how important privacy during shopping was to them. Over 87 percent of the participants valued privacy averagely or higher. 68 percent already had experience with shopping clothes at home (via mail, telephone or online).

The participants were shown a picture of a real person (like a model in a catalog) and a virtual avatar who both wore the same piece of clothing. Then, the participants had to rate how helpful the representations were. We used a unipolar Likert scale ranging from 1 = very helpful to 5 = not helpful. The answers clearly showed that participants preferred real persons (average rating of 1.98) over virtual ones (average rating of 3.44). This difference is statistically significant: the ratings were compared using a paired T-test ($T_{40} = -6.172, p < .01$). However, almost all participants (40 out of 41) agreed that a virtual avatar, whose appearance resembled their own appearance, was more helpful than a random virtual avatar. Participants were shown a rendered screenshot of our system.

http://www.makehuman.org
http://mocap.cs.cmu.edu
in comparison to a random virtual avatar along with this question (see Figure 8).

Next, the participants had to watch a demo video that showed how a user who interacted with the suggested system using a depth camera, and in comparison a demo video that showed a user who just watched a preset animation. Figure 9 shows screenshots of the video. Over 68 percent of the participants rated the interactive system as more helpful than the one with a preset animation. Finally, participants watched a demo video that visualized free-viewpoint navigation in contrast to a fixed viewpoint. Over 90 percent of the participants assigned an above average importance to the possibility to view themselves from all sides while wearing virtual clothes.

We conclude that home shopping is an important channel for buying and selling clothes, and that privacy is one of the reasons for it. Participants had a strong affinity towards real models, but also valued the possibilities of virtual avatars, like interaction and free-viewpoint navigation. Also, the possibility to have the virtual avatar look like themselves was perceived well. Therefore, we conclude that virtual try-on applications have to deliver great realism in both the visual quality and the interaction techniques to be accepted by users. The suggested approach tries to fulfill all these aspects by providing a system that can be used in private, offers realism by using reconstructed models and is interactive by using commodity depth cameras.

7 Conclusions and future work

This paper described an efficient and affordable method for virtual try-on at home. The method consists of two major components: a 3D reconstruction room with multiple calibrated cameras, and a user’s desktop computer equipped with a commodity depth camera. Users who entered the reconstruction space can download their 3D avatars at home and control them with their own bodies. We achieve this by tracking the user’s body with the depth camera. Users can download clothes in order to augment their avatars. We also gathered user feedback to verify the general interest in the topic as well as the specific approach that we suggest.

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

References


Chittaro, L., and Corvaglia, D. 2003. 3d virtual clothing: from garment design to web3d visualization and simulation. In Pro-


