Real-time Reshaping Of Humans

Michal Richter
mrichter@mpi-inf.mpg.org

Kiran Varanasi
varanasi@mpi-inf.mpg.org

Nils Hasler
hasler@mpi-inf.mpg.org

Christian Theobalt
Max-Planck-Institut Informatik, Saarbruecken
theobalt@mpi-inf.mpg.org

Abstract

We present a system for real-time deformation of the shape and appearance of people who are standing in front of a depth+RGB camera, such as the Microsoft Kinect. Our system allows manipulating human body shape parameters such as height, muscularity, weight, waist girth and leg length. The manipulated appearance is displayed in real-time. Thus, instead of posing in front a real mirror and visualizing their appearance, users can pose in front of a 'virtual mirror' and visualize themselves in different body shapes. Our system is made possible by a morphable model of 3D human shape that was learnt from a large database of 3D scans of people in various body shapes and poses. In an initialization step, which lasts a couple of seconds, this model is fit to the 3D shape parameters of the people as observed in the depth data. Then, a succession of pose tracking, body segmentation, shape deformation and image warping steps are performed – in real-time and independently for multiple people. We present a variety of results in the paper and the video, showing the interactive virtual mirror cabinet experience.

1. Introduction

Artists have long employed the technique of morphing the visual appearance of humans for comical and caricature purposes, and to improve the appearance. With the advent of digital technology, such editing has become commonplace through tools such as Adobe Photoshop. However, such editing was cumbersome and required intensive manual work by the artist in order to produce a plausible image. Quite recently, certain methods allowed for automatic morphing of shape and appearance of people in photographs [13] and in videos [8]. This is achieved by a morphable human body model that is built from a statistical analysis of human body shapes, obtained from a large database of 3D scans. During an initialization step, the shape parameters of the model are fit to a given person. Then, these parameters are varied to obtain the structure of the person in a deformed shape, but in the same pose. To show this deformation in a picture or a video, an image-warping step needs to be performed that is driven by positional constraints on certain selected pixels. Each of these steps might involve manual input that makes the task harder for reshaping humans in videos [8], since the editing has to be performed in a holistic way throughout the video. In this paper, we take this challenge much further, by presenting a completely automatic system that is capable of real-time manipulation of human body shape and appearance.

Achieving the shape-morphing effect in real-time opens up a new range of personalized augmented reality applications that could be interresting especially for fairgrounds and theme parks. Indeed, carnivals have often employed mirrors with optical distortions such that people experience the effect of seeing themselves thinner, shorter, fatter or taller. Mirrors can be designed to deflect the light just so that these effects are visible to people standing at a certain position and distance. However, the suspension of disbelief that is possible through mirror optics alone is rather narrow. The same can be said about simple image distortion methods that can be applied over digital images. In this paper, we present an approach for semantically meaningful human shape distortion in real-time. The person need not stand in a specific place or pose, and the range of possible shape deformations is far wider – e.g., the muscularity or the BMI of the person can be changed. Thus, our system can also serve as a *Virtual Gym* showing how people will look if they loose weight or work out regularly. This is possible, because the shape deformations we provide can be interpreted semantically and look sufficiently realistic.

In order to achieve such deformations, a semantically meaningful knowledge about the human shape is required, and that shape has to be fit properly to the image, before the image is warped. The works of Zhou et al. [13] and Jain et al. [8] perform this off-line by statistically modeling

978-0-7695-4873-9/12 $26.00 © 2012 IEEE
DOI 10.1109/3DIMPVT.2012.81

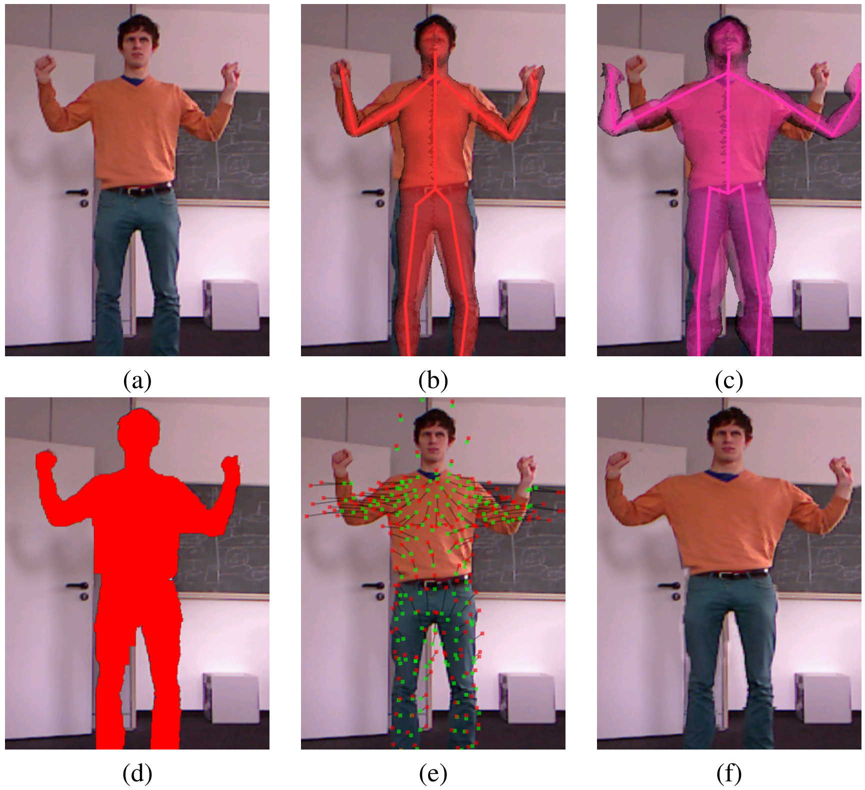


Figure 1. Overview of the system workflow: (a) Input video (b) Registered 3D shape overlay (c) Modified 3D shape (d) Foreground mask (e) Image warping constraints – red points correspond to target locations and green points to source locations (f) Output warped image

the shape semantics from a database of 3D scans of people (both works rely on the database collected by Hasler et al.[6]). This problem, however, becomes far more challenging when trying to achieve real-time performance. In this paper, we solve a variety of theoretical and engineering challenges to present a *virtual mirror cabinet* experience for the users. In real-time, the users can visualize semantically-meaningful deformations of their shape and appearance. For example, a person can appear taller or more muscular, and this appearance is maintained as the person moves and changes pose. The shape of the person can also be continuously changed, in real-time over the course of motion. Different shape deformations can be applied to different people, which allows for an entertaining and fun experience.

Our main contributions in this paper, differentiating it from related works [13] [8], are as follows.

1. A fast and completely automatic shape initialization procedure, where a set of human body shape parameters are rapidly estimated.
2. A real-time pose tracking, pose transfer and shape deformation system for articulated 3D motion.
3. A real-time image foreground segmentation that refines the Kinect SDK foreground mask using the RGB images of the Kinect sensor.
4. A real-time implementation of image warping improving on the Movie Reshape [8] implementation. In contrast to Movie Reshape, our implementation handles foreground and background separately to avoid any distortions in the background.

2. Previous work

Our work falls in the broad range of techniques dealing with real-time augmented reality. Augmented reality services which partly alter the incoming video streams are being used increasingly in computer games [11], television advertising and in video-chats. However, the kind of possible edits are often limited to a rigid shape that is overlayed onto the surface of the object being imaged. For example, a pair of spectacles can be added to the face of a consumer [10], such that the new look can be visualized before a purchase is made. Hilsmann et al. [7] have demonstrated an application where a virtual print is added to the T-shirt of a person being imaged. An entire virtual dress can also be constructed, and added to a person's frame [14]. Many in this gamut of applications have been termed as '*virtual mir-*

ror' applications. However, each application is often limited to a specific scenario. Various artifacts may affect the quality of the results owing to misalignment in shape registration over images. Simulating virtual non-rigid shapes such as dresses or human bodies in a perceptually pleasing manner is also hard, and may result in visual artifacts if done improperly. Various research efforts have been addressing some of these issues. But so far, modifying the actual body shape of the persons in real-time has not been attempted. Our work enlarges the scope of *virtual mirror* applications by addressing this important challenge.

We now discuss a category of works from computer vision whose objective is to estimate human body shape parameters from input photographs or video. Such modeling of human body shape is necessary to effect any semantically meaningful body shape modification. Anguelov et al. [1] present the SCAPE body model as a means for separately modeling the statistical variation of human body shapes and poses from a database of 3D scans - with several people standing in the same pose and one person standing in different poses. Hasler et al. [6] present a different statistical body model by simultaneously modeling both body shape and pose from a database of 3D scans consisting of several people in various poses. Such statistical human body models have a variety of applications. Balan and Black [2] use the SCAPE model to present an application for estimating the 3D 'naked' body shape of a person underneath clothes from a set of photographs. Weiss et al. [12] present an application for obtaining a more accurate 3D body scan of the person from the depth data given by the Kinect sensor, and regularizing the noisy depth information through the SCAPE body priors. However, this approach requires higher computation time and multiple scans. In contrast, we present an approach for rapid shape estimation with real-time applications. Our work takes direct inspiration from the works of Zhou et al. [13] and Jain et al. [8] who present interactive systems for reshaping human body shape in photographs and videos respectively. We extend the scope of these works and target real-time reshaping of humans to allow for a virtual mirror experience.

3. Overview

We first describe the input and output sensor modalities, and the underlying mathematical model for shape and pose deformation. Afterwards, the real-time image deformation method is detailed.

System Input and Output As input to our system, we take the RGB video stream provided by the Microsoft Kinect sensor at a resolution of 640×480 pixels, with the frame rate of 30 fps. We assume that there are 1 - 2 people (referred to as *actors*) in front of the Kinect camera, and take as input a set of shape modification attributes from the user. The output from our system is an altered RGB video stream where the required shape modifications are performed (as illustrated in Fig. 1). To drive our method, we take two further inputs from the Kinect sensor (i) a depth data stream at a resolution of 320×240 pixels with the depth range between 1.2 and 3.5 meters (ii) the joint positions of 20 skeletal joints (denoted henceforth by $\{K_i\}$) for $1-2$ actors given by the Kinect SDK. However, please note that these input skeleton estimates are not temporally coherent and 'bone lengths' vary over time. Also, no joint angle parameters are provided[1].

Statistical human shape model Our system relies on a mathematical model for human body shape that models variations across several men and women. We build on top of the model developed by Hasler et al. [6], that was built from 3D laser scans of over 100 people of varying age and gender. Our model can be represented as $B^{\boldsymbol{\alpha}}_{\boldsymbol{\Theta}} = (M^{\boldsymbol{\alpha}}, S_{\boldsymbol{\Theta}}, W)$ where $M^{\boldsymbol{\alpha}}$ is a 3D surface mesh consisting of $nm = 6500$ vertices in $\mathbb{R}^3$ controlled by shape parameters $\boldsymbol{\alpha}$ (explained in the following), $S_{\boldsymbol{\Theta}}$ is a kinematic skeleton of joints $\{J_i\}$, which is controlled by pose parameters $\boldsymbol{\Theta}$, and W is a set of skinning weights that map the surface vertices to a linear combination of rigid transformations affecting various joints. We design the skeleton such that the joint positions J_i correspond to the locations K_i, where Kinect SDK places the skeletal joints. The model is built on top of an average human body model $B^0_0 = (M^0, S_0, W)$, specifying the average surface mesh M^0 of the persons in the database and the rigged kinematic skeleton S_0 in the default pose. Henceforth, we call this the *default body*. Underlying this model are a set of $n = 20$ shape deformation vectors $\{^n_1\mathbf{v}_i\}$ that provide a linear basis spanning the principal variations of human body shape, as observed in the database collected by Hasler et al. [6]. When the user specifies a set of input human shape parameters $\boldsymbol{\alpha} = \{^n_1\alpha_i\}$, a corresponding 3D surface mesh (in the default pose) can be obtained as

$$M^{\boldsymbol{\alpha}} = M^0 + \sum_{i=1}^{n} \alpha_i * \mathbf{v}_i \tag{1}$$

This body shape can be retargeted to a new pose specified by joint transformations $\boldsymbol{\Theta} = \{\theta_i\}$. We use a set of $nj = 52$ joint parameters, where $\theta_1, \theta_2, \theta_3$ correspond to the absolute position of the root joint, and the parameters $\theta_4, \ldots, \theta_{nj}$ correspond to rotation angles for the rest of the joints. The body model in the new pose $\boldsymbol{\Theta}$ with shape parameters $\boldsymbol{\alpha}$ is denoted as $B^{\boldsymbol{\alpha}}_{\boldsymbol{\Theta}}$. Mesh deformation with respect to these joint transformations is modelled by a stan-

[1]The Kinect for Windows SDK 1.5 released in May 2012 already provides rotation parameters for joints. However, our project was developed before its release.

dard surface skinning approach driven by skinning weights W.

Initialization When the actor appears in the scene for the first time, an initialization step needs to be performed in order to estimate the body shape parameters $\boldsymbol{\alpha}^S = \{\alpha_i\}$ that map to the principal shape variations given by the basis deformation vectors $\{\mathbf{v}_i\}$. We refer to the output of this initialization step, i.e. the estimated source model of the actor with the embedded skeleton in the default pose as the '*source body*' B_0^S.

Real-time deformation Based on this initialized shape deformation model, we describe a system for reshaping the appearance of humans in real-time. Fig. 1 provides a conceptual overview, which consists of the following 4 principal steps

1. **Pose deformation** At each frame, pose $\boldsymbol{\Theta}$ is estimated by minimizing the distances between the joints of $B_{\boldsymbol{\Theta}}^S$ and corresponding Kinect joints K_i (Fig. 1b).
2. **Shape deformation and pose transfer** The model of desired actor appearance *target body* B_0^T is obtained from B_0^S through a shape deformation driven by user provided shape parameters $\boldsymbol{\alpha}^T$, which correspond to semantically meaningful attributes such as height, weight, muscularity, etc. The pose $\boldsymbol{\Theta}$ is transfered to *target body*,so that $B_{\boldsymbol{\Theta}}^T$ models the desired target appearance (Fig. 1c).
3. **Foreground mask estimation** Foreground region corresponding to the actor is estimated. This step takes the Kinect SDK foreground mask and refines it by means of *Graph-cut* segmentation [3] to obtain the final foreground mask (Fig. 1d).
4. **Image warping** Based on the constraints obtained from correspondences between the image projections of $B_{\boldsymbol{\Theta}}^S$ and $B_{\boldsymbol{\Theta}}^T$ (Fig. 1e), the foreground region of the image is deformed by an image-warping step, which achieves the desired actor appearance modification. Finally, the output image is composed by overlaying morphed foreground over the background image (Fig. 1f).

In the following, the initialization strategy is detailed in Section 4. In Section 5, the method for real-time shape deformation is explained. Results of our approach are presented and discussed in Section 6, and directions to future work are pointed out. Section 7 concludes the paper.

4. Initialization

We propose a novel and completely automatic initialization procedure to estimate the body shape parameters of the actor from the Kinect sensor. The objective of this method is to provide a simple and fast strategy (taking only a couple of seconds) for initialization without any manual intervention or markers. The actor appears in front of the sensor in an arbitrary static pose. Using the point cloud from the depth image and the landmarks of joint positions given by the Kinect SDK, the source body shape B^S of the actor is estimated. This consists of the following steps.

Transformation of point cloud into default pose Let $P_{in} = \{_1^{np} p_i \in \mathbb{R}^3\}$ be the 3D point cloud given by the Kinect depth sensor. The objective of this step is to transform P_{in} into the default pose P_0 such that body shape parameters can be optimised. In order to achieve this transformation, we first need to estimate correspondences between points in the depth image and the vertices of B_0^0. We first construct an intermediate body shape B_0^D from B_0^0 by scaling the bone lengths of S_0 in order to fit bone lengths given by the Kinect skeleton $\{K_i\}$, and correspondingly scaling the offsets of vertices from joints in the direction along the bones.

Then we solve the inverse kinematics (IK) problem in order to estimate pose parameters $\boldsymbol{\Theta}_{in}$, such that the skeleton of B_{in}^D exactly fits the joint positions K_i given by the Kinect SDK. We adopt a standard IK-solve step by estimating the joint rotations down the kinematic chains (starting from the root joint), in a manner similar to that described by Bregler et al. [4]. This procedure gives us a body shape model B_{in}^D that is reasonably close to P_{in}. We take advantage of this proximity to estimate correspondences between P_{in} and the vertices in the body shape model B_{in}^D, by simply assigning correspondences to the closest neighboring vertex for each point in P_{in}. We use the skinning weights W of corresponding mesh vertices as the skinning weights for their matches in P_{in} and transfer the point cloud to the default pose P_0. We do this by applying the inverse of the pose transformations $\boldsymbol{\Theta}_{in}^{-1}$ along the kinematic chain and blending the joint transformations with linear blend skinning.

Fitting statistical body model to point cloud data Here, we estimate the optimal set of parameters $\hat{\boldsymbol{\alpha}} = \{_1^n \hat{\alpha}_i\}$ such that the mesh $\hat{M} = M^{\hat{\boldsymbol{\alpha}}}$ matches the shape of the actor. Given P_0 from the earlier step, we perform a gradient descent optimization of the shape parameters $\alpha_1, \ldots, \alpha_n$ in order to minimize the distances between points in P_0 and their corresponding mesh vertices. i.e. given the point cloud $P_0 = (p_1, ..., p_{np})$, mesh $M = (v_1, ..., v_{nm})$ and correspondence function $k(i) = j$, where v_j is the closest mesh vertex to the point p_i), we minimize the following energy with respect to the parameters $\boldsymbol{\alpha}$

$$E(\boldsymbol{\alpha}) = \sum_{i=1}^{np} (p_i - v_{k(i)}^{\boldsymbol{\alpha}})^2 \tag{2}$$

Subsequently, we perform this shape initialization step at several frames and compute the optimal set of shape parameters for a given person by averaging the results from individual frames. Since the Kinect depth sensor captures at 30 fps and our optimization procedure is very fast, this only takes $2-3$ seconds to initialize the model. We refer to the body model with optimized shape parameters $\hat{\boldsymbol{\alpha}}$ as the *source body* $B_0^{\hat{\boldsymbol{\alpha}}} = B^S$.

Skeleton refitting The earlier step provides us with the surface mesh M_0^S of the source body B_0^S but not yet the underlying skeleton. We obtain the positions of skeletal joints utilizing the approach used in [5].

For each joint J_i, we associate a set of mesh vertices $\{_1^{ni}v_a\}$ that are close to the joint. We denote the positions of these vertices in the default body as $v_1^0, ..., v_{ni}^0$ and their positions in the source body as $v_1^S, \ldots, v_{ni}^S$. Let J_i^0 be the position of the joint in the default body B_0^0. Then the position J_i^S in the source body B_0^S can be computed as

$$J_i^S = J_i^0 + \frac{\sum_{a=1}^{ni}(v_a^S - v_a^0)}{n} \tag{3}$$

After this initialization procedure, we have the source body B_0^S i.e, a 3D body model for the actor with a 3D surface mesh and a rigged kinematic skeleton. We use this to track the actor and apply various body shape deformations.

5. real-time shape deformation

5.1. Shape deformation

To deploy the modified body appearance, we take a set of semantically meaningful shape deformation attributes and deform the source body shape B_0^S to a target body shape B_0^T. The shape database of Hasler et al. [6] is recorded along with a set of semantic attributes - height, weight, breast girth, waist girth, hips girth, legs girth, muscularity. Following the Movie-Reshape project [8], which uses the same database, we adopt a linear map between the space of principal shape variations $\{\mathbf{v}_i\}$ as defined earlier, and the space of semantically meaningful shape offsets $\{\mathbf{w}_i\}$ (please note that unlike $\{\mathbf{v}_i\}$, the semantic shape offset vectors $\{\mathbf{w}_i\}$ might be correlated with each other). We take as input from the user a set of semantically meaningful shape modification parameters $\lambda_1, \ldots, \lambda_6$ and morph the source body B^S into the target body B^T as follows.

$$M^T = M^S + \sum_{i=1}^{6} \lambda_i * \mathbf{w}_i \tag{4}$$

This shape deformation step does not depend on the sensory input from the Kinect, but only needs to be performed whenever the parameters λ_i are changed by the user. In our

(a) (b)

Figure 2. Image segmentation: Results for person shortening (a) Kinect SDK foreground (b) *Graph-cut* refined foreground.

virtual mirror application, we provide for a gradual and continuous change of shape variations which is more pleasing to the user.

5.2. Pose deformation

In this step, we deform the pose of source and target bodies to suit to the observations of joint positions $\{K_i\}$ given by the Kinect SDK. For this, we take the skeleton S_0 of the source body B_0^S in the default pose, and estimate the skeletal pose $\boldsymbol{\Theta}$ such that the distance between joints J_i of $S_{\boldsymbol{\Theta}}$ and the corresponding Kinect joints K_i in the current frame is minimized. Again, we use the standard IK-solve procedure by consecutively estimating the root position and joint rotation angles down the kinematic chains. The pose adjusted models $B_{\boldsymbol{\Theta}}^S$ and $B_{\boldsymbol{\Theta}}^T$ with meshes skinned according to the pose $\boldsymbol{\Theta}$ and their projections into the video image provide the necessary spatial constraints for image warping step.

5.3. Foreground estimation

Prior to the image-warping, we need to separate out the scene background from the image foreground over which the actor appears. We take advantage of the foreground segmentation mask in the depth image that is provided by the Kinect SDK, but this is jittery at the edges. In this step, we propose a real-time procedure that refines this mask and extracts a clean foreground. We build up on work of Boykov and Jolly [3], who formulate image segmentation as a graph partitioning problem. An unordered graph is constructed between image pixels such that every pair of neighboring pixels is connected by an edge whose weight is given by color similarity, and two extra terminal nodes are added to denote foreground and background regions. Source and sink weights are given by similarity of the pixels to known foreground and background regions. The image is then segmented into foreground and background regions by a fast *min-cut* algorithm which cuts along the edges with minimum weight. In order to achieve this in real-time, we apply this method only on a narrow band of pixels around the boundary of the foreground mask given the Kinect SDK (those pixels that are likely to be assigned incorrectly).

5.4. Image warping

In this step, we take the input image and a set of deformation constraints in order to produce a warped output image. The deformation constraints are given by source and target 2D locations for a set of pixels. This sparse set of constraints are smoothly interpolated over the whole image domain. In our case, the constraints come from the projections of a sparse subset of vertices of $B^S_{\mathbf{\Theta}}$ and $B^T_{\mathbf{\Theta}}$ - the source and target bodies in the new pose $\mathbf{\Theta}$. Let the 2D vertex locations in the source and target bodies be $s_1, \ldots, s_n$ and $t_1, \ldots, t_n$ respectively. Similar to the [8], we use the Moving Least Squares (MLS) method for image warping, through a parallelized GPU implementation for real-time performance. For each pixel x, MLS finds the optimal 2D transformation M_x that transfers the pixel to its new location $x^{'} = M_x(x)$. Following the minimization strategy of Müller et al. [9], M_x is obtained as

$$M_x = argmin_M \sum_{i=1}^{n} \frac{1}{|x - s_i|^2} (M(s_i) - t_i)^2 \qquad (5)$$

In our GPU implementation of MLS, we compute the transformation M_x only for a subset of pixels in a uniform 160×120 pixel grid. The warping field of the whole image domain is obtained from the grid by linear interpolation. This optimization exploits the fact that the warping field varies smoothly, so the sparse spatial sampling does not have significant impact on results. With this technique, we achieve 16× speed-up when warping an image of size 640×480 pixels, which enables us to accomplish real-time performance even with low end integrated GPUs such as 16 cores NVIDIA NVS 3100M.

Please note that the image warping method in our paper is different from the one proposed in the Movie-Reshape paper [8], which does not separate foreground from background. This direct warping results in background distortions that are negligible for minor shape deformations. However, they become disturbing for exaggerated shape deformations, which are often used in the context of our application (see Fig. 3 for comparison).

5.5. Output image composition

In order to produce the illusion of a virtual mirror cabinet, we compose the warped image foreground with an unwarped image background. Holes may appear if the foreground region is shrunk (e.g, when the person is deformed to be shorter or thinner). We fill these holes through inpainting from a static background image, which is recorded first during initialization. This background image is subsequently updated over time such that pixels marked as background with high confidence (i.e, sufficiently far from pixels estimated as foreground) replace the older pixels.

(a) (b)

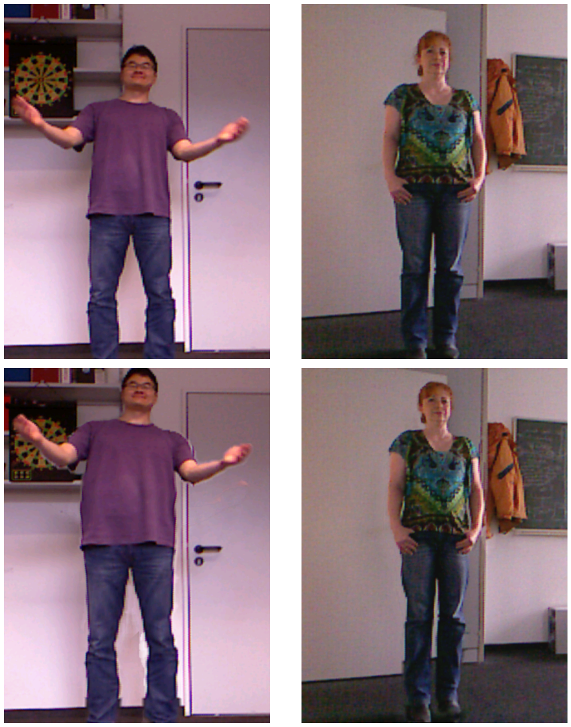
Figure 3. Image warping: Results for shortening and muscularity increase (a) Movie-Reshape warping strategy (b) our method.

(a) (b)

Figure 4. Result of shape modification: Top row - input images. Bottom row - (a) increased muscularity (b) decreased muscularity

6. Results

We evaluate the system on a number of users of various body proportions, and wearing a variety of clothing. In



(a) (b)

Figure 5. Result of shape modification: Top row shows the input images. Bottom row shows (a) increased weight and height (b) effect of making thinner with longer legs.

Fig. 4 and Fig. 5, we present the resulting images of various body shape modifications. These results are best viewed in the accompanying video, which shows the interactive nature of our system and how it achieves temporally consistent smooth body modifications in the video. Fig. 4 shows the results of increasing and decreasing of muscularity of the actor. The foreground shape deformations resemble caricature depiction due to the extreme change of the muscularity parameter. Distortions in background areas are avoided because the deformation is restricted to the foreground. The high accuracy of the segmentation leads to very plausible results. Fig. 5 shows examples of moderate shape deformations. The resulting shape deformations look realistic and give the idea of how the person will look if certain body shape parameters change. The achieved level of realism of minor shape edits opens up possibilities of personalized augmented reality applications such as the "Virtual Gym" - showing users how will they look if they diet or go to a gym regularly.

The importance of graph cut segmentation refinement is shown in figure Fig. 2 as well as in the acompanying video. The refined masks yield more accurate results and reduce the artifact of jittering near the actor's boundary.

The advantages of our image warping strategy over the Movie-reshape image warping strategy in case of significant shape deformations are illustrated by Fig. 3 and the acompanying video.

The system is capable of producing an output video with a frame-rate of 12 fps on a regular laptop[2] and 18 fps on a high-end desktop computer[3] - foreground estimation takes 50% and image warping takes 40% of computation time approximately. It is possible to track and deform two persons simultaneously. This makes it suitable for deployment in interactive demos and virtual chat rooms.

Limitations and Future Work Visual artifacts might result from errors in the steps *Shape Initialization*, *Pose Tracking*, *Segmentation*, or *Image Warping*. These artifacts can be broadly divided into two categories. The first group are visually displeasing artifacts that are harmful for the virtual mirror illusion. The second group are artifacts that are not visually displeasing, but where the output appearance of the person is different from the desired result. We show some of these artifacts in Fig. 6.

We first discuss the visually unpleasant artifacts. Certain poses of the person might be difficult to register for the Kinect-based pose tracking algorithm, for example, when some of the limbs are occluded by other parts of the body. Since we rely on the joint location detection provided by the Kinect SDK, which is not temporally coherent and unstable for such difficult poses, this may result in jumps and shaky movement in the output video. Such artifacts can be removed by incorporating temporal priors in pose tracking.

Detecting occlusions and handling missing data is also an important area for future work. When parts of the actor are not visible in the input image but are required to be visible after the shape modification, our method fails to reproduce them since it warps each frame of the video independently. More intelligent object inpainting algorithms are required to handle such cases. Minor artifacts can also appear when filling holes in the final image with pixels from the static (inpainted) background. This is mostly caused by the fact that the lighting condition may have changed between the current frame and the frames where pixels used for inpainting originate from.

Imperfect segmentation of the body shape near the boundaries of the foreground mask might produce flickering. Inaccurate segmentation also leads to some warping of the pixels that are originally in the background but marked incorrectly as belonging to the actor. Better foreground segmentation can be achieved by incorporating more accurate appearance and shape priors, than simple color similarity.

The second class of artifacts where the output appearance of the person differs from the desired result, happen

[2]Dell Latitude E6410, Intel CORE i5, NVIDIA NVS 3100M

[3]HP Elite 7300 Microtower, Intel Core i7 2600 Sandy Bridge, NVIDIA GeForce GT545 3GB

(a) (b) (c)

Figure 6. Examples of artifacts: (a) shows the artifact in the hair region due to segmentation failure, (b) shows unnatural discontinuities in the background caused by the fact that the shadow casted by the person in not properly represented in the static background image, (c) shows unnatural legs cut due to missing image information when the person is made taller.

due to imperfect alignment between the tracked model and the actor's image caused either by poor model initialization or pose tracking. In such cases, the warping constraints are not placed at the correct locations and the shape deformation effect can be weakened. Another problem results from the fact that the warping constraints act globally on the actor's foreground image, which means that constraints on one body-part influence another body-part when both come close. For example, the person's arm and chest are warped together when they are in close proximity in the input image. This problem can be alleviated by accurate body-part segmentation in the input images, which poses a challenging problem for future work.

7. Conclusion

In this paper, we have presented a system for real-time shape deformation of people. To achieve this, we solve several challenging vision problems in real-time: shape initialization, human pose-tracking, shape segmentation and image warping. We build on existing state-of-the-art systems for depth imaging and body-part detection, and statistical human body modeling. On top of this, we solve several crucial problems to achieve real-time performance. Our work presents the first system that is capable of producing human body-shape modification effects in real-time.

References

[1] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, July 2005.

[2] A. Balan and M. Black. The naked truth : Estimating 3d human body shape under clothing. In *ECCV*, pages 15–29, 2008.

[3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, pages 105–112, 2001.

[4] C. Bregler and J. Malik. Tracking people with twists and exponential maps. page 8, Washington, DC, USA, 1998. IEEE Computer Society.

[5] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi. Markerless motion capture through visual hull, articulated icp and subject specific model generation. *Int. J. Comput. Vision*, 87(1-2):156–169, Mar. 2010.

[6] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. In P. Dutr'e and M. Stamminger, editors, *Computer Graphics Forum (Proc. Eurographics 2008)*, volume 2, Munich, Germany, Mar. 2009.

[7] A. Hilsmann and P. Eisert. Tracking and retexturing cloth for real-time virtual clothing appliations. In *Mirage 2009 - Computer Vision/Computer Graphics Collaboration Techniques and Applications*, May 2009.

[8] A. Jain, T. Thormählen, H.-P. Seidel, and C. Theobalt. Moviereshape: Tracking and reshaping of humans in videos. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2010)*, 29(5), 2010.

[9] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3):471–478, July 2005.

[10] Rayban. Virtual mirror, 2012.

[11] Sony. Wonderbook: Book of spells for sony playstation 3, 2012.

[12] A. Weiss, D. Hirschberg, and M. Black. Home 3d body scans from noisy image and range data. In *ICCV*, pages 1951–1958, 2011.

[13] S. Zhou, H. Fu, L. Liu, D. Cohen-Or, and X. Han. Parametric reshaping of human bodies in images. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 126:1–126:10, New York, NY, USA, 2010. ACM.

[14] Zugara. The webcam social shopper : Virtual dressing room, 2011.