Garment Personalization via Identity Transfer

Roy Shilkrot ▪ *MIT*

Daniel Cohen-Or ▪ *Tel Aviv University*

Ariel Shamir ▪ *Interdisciplinary Center, Herzilya*

Ligang Liu ▪ *University of Science and Technology of China*

This system creates a virtual experience akin to trying on clothing. It clones the user's photographic image into a catalog of images of models wearing the desired garments. The process takes into account the user's skin color and body dimensions.

There's a pressing need to provide a more personalized experience for online shopping than just browsing through catalogs. This is even more critical for clothes shopping because different people have different physical characteristics and preferences. Researchers have created technologies such as virtual mirrors and video fitting using VR for "trying on" outfits online. However, these technologies haven't provided comfortable user interaction or achieved an emotional response in which users can visualize themselves wearing or using the products in a natural environment.

We aim to create a more precise, natural clothing fit for users. We concentrate on a single image, striving for high-quality results that create the experience of an identity transfer. The input to our system comprises a picture of the system's user, called the *user image*, and a reference picture of a human model from a clothing catalog, called the *catalog image*. Figure 1 shows an example of our system transferring a subject's identity onto garments from the catalog. Our system produces a real-time photo album depicting how users might look if they wore the clothes and posed for a camera.

One of our goals was to design a system that unskilled users could operate, in which preprocessing of the user image and system training require only quick, simple interaction. Toward that end, we combined techniques from computer graphics, computer vision, and machine learning. Our system is based on a recently developed body-reshaping process,[1] skin detection and recoloring, and, most notably, a novel image-space procedure to extract and transfer human heads in images. Unlike other research in face replacement (see the "Related Work in Identity Transfer" sidebar), we extract and clone an entire human head, including the hair and neck.

System Overview

Because humans are especially sensitive regarding images of themselves, seamless, easy, and quick head transferring in images is especially challenging. In particular, our system must

- accurately segment the head with minimal (unskilled) user assistance,
- allow adjustment of the input pose and scaling of the head and shoulders, and
- seamlessly clone the source head and place it on the target body.

To do this, our system has two main components: offline semiautomatic image preprocessing and

0272-1716/13/$31.00 © 2013 IEEE

learning, and online automatic identity transfer (see Figure 2).

The system starts with an input image of the user clearly showing the head, neck, and hair. Input images can come from a photo album or a webcam. During preprocessing, the system also takes as input simple measurements of the user's body shape (height, weight, girth, and so on). We can either define the measurements once and store them for future use or chose from a set of characteristic body shapes. The system then performs head extraction based on curve fitting, segmentation, and position training.

Online, the system seamlessly reattaches the extracted head image to the body in the catalog image. It then adjusts the catalog image's skin tone to match the user's and warps the catalog image to the subject's dimensions.

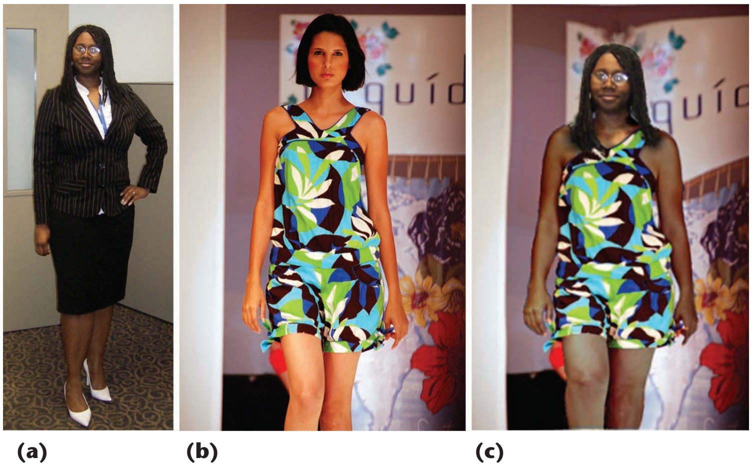


Figure 1. Given (a) an input picture of the user and (b) a picture of a human model from a clothing catalog, our system (c) transfers the user's identity onto the catalog image. This provides a precise, natural "try on" experience for the user.

Head Extraction

To extract the image of the user's head and transfer it onto the catalog image, we employ a novel graph-cut-based technique. For the examples in this article, we employ a similar procedure to the catalog images, but we could have used other methods to prepare the catalog.

Segmenting a foreground object from its background is challenging. In this case, the head region isn't uniform and contains different elements such as the face, hair, and exposed skin on the neck So, the traditional two-kernel model (head and background) doesn't work well.

For more robust head extraction, we developed a three-kernel model based on textons, which are fundamental local structures of texture in natural images. We segment the image into the face, hair, and background. To initialize the segmentation, we estimate these parts' locations in the image using a parametric polycurve template. The system determines the template models' statistical parameters through learning. We use these parameters to find the best shape and position for the user image.

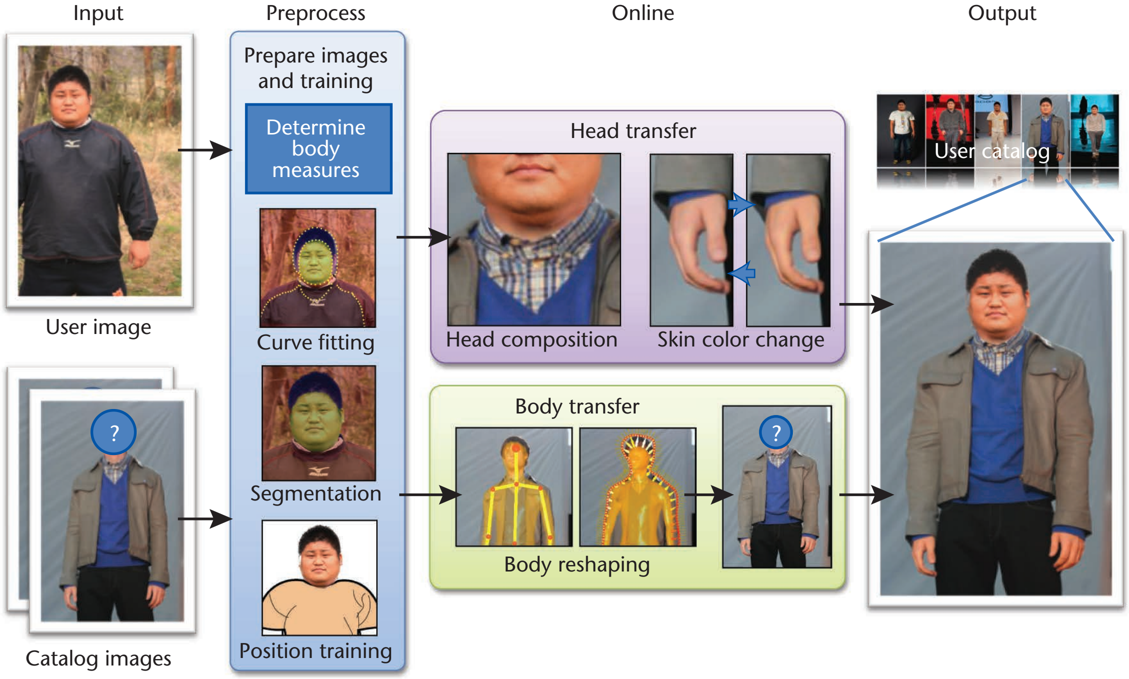


Figure 2. Identity transfer for an experience of virtually trying on clothing. Our system has two main components: offline semiautomatic image preprocessing and learning, and online automatic identity transfer.

Related Work in Identity Transfer

Automatic transfer of human visual properties between images has received much attention in the computer graphics community in recent years. Either by means of 3D or 2D models, or using pixel-based methods, the common goal is to change the appearance of a human in one image to look like a different person.

Face Replacement

Volker Blanz and Thomas Vetter presented a method for reconstructing 3D facial models as well as creating and simulating unknown views.[1] This method inspired others to use morphable 3D facial models to alter faces in images. Such was the case in Shahzad Malik's research[2] and the Digital Emily Project.[3]

Blanz and his colleagues also showed how to fit a morphable model of 3D faces to both the source and target face images and how to estimate the images' shape, pose, and lighting parameters.[4] They exchanged faces by rendering the face reconstructed from the source image with the rendering parameters estimated from the target image.

Dmitri Bitouk and his colleagues' system automatically replaced faces across images with different poses, lighting, facial expressions, and skin tones, without 3D reconstruction.[5] It used a large library of face images to find good candidate target images and applied various recoloring and relighting techniques to the new face image.

Instead of using a 3D model, the other main approach for face replacement copies pixels, assisted by canonical blending methods in 2D computer graphics. Elaine Newton and her colleagues used a variant of this approach for face de-identification;[6] Neel Joshi and his colleagues used this variant to enhance personal photos.[7] Kevin Dale and his colleagues presented another variant for processing two simultaneous video streams in which they transferred the face of a person in one video onto a person in the other.[8]

Complete Head Segmentation

Not satisfied by face segmentation alone, several researchers have extracted full human heads from images. Yaser Yacoob and Larry Davis proposed hair segmentation bootstrapped by a parametric model of a human face.[9] This method continued the segmentation by region growing based on RGB color value statistics.

Kuang-chih Lee and his colleagues proposed an iterative method for complete face segmentation.[10] They used a Markov random field and a comparison between a graph-cut solver and a loopy belief propagation solver. The initializer was a set of six template masks of hair of varying lengths learned from a training dataset.

Image Composition

We can use seamless image composition approaches to address the head transferring we describe in the main article. Patrick Pérez and his colleagues introduced a technique for compositing images seamlessly by combining image gradients.[11] Gradient-based techniques have become the standard for seamless stitching and composition. For example, Aseem Agarwala and his colleagues developed the Photomontage system, which lets users interactively create a composite image by combining different parts of several source photographs.[12] In our research, we adopted the gradient-domain technique to stitch the head and body while also considering skin color.

References

1. V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces," *Proc. Siggraph, ACM*, 1999, pp. 187–194.
2. S. Malik, *Digital Face Replacement in Photographs*, tech. report, Computer Science Dept., Univ. Toronto, 2003.
3. O. Alexander, "The Digital Emily Project: Photoreal Facial Modeling and Animation," *ACM Siggraph 2009 Courses*, ACM, 2009, article 12.
4. V. Blanz et al., "Exchanging Faces in Images," *Computer Graphics Forum*, vol. 23, no. 3, 2004, pp. 669–676.
5. D. Bitouk et al., "Face Swapping: Automatically Replacing Faces in Photographs," *ACM Trans. Graphics*, vol. 27, no. 3, 2008, article 39.
6. E. Newton, L. Sweeney, and B. Malin, "Preserving Privacy by De-identifying Face Images," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 2, 2005, pp. 232–243.
7. N. Joshi et al., "Personal Photo Enhancement Using Example Images," *ACM Trans. Graphics*, vol. 29, no. 2, 2010, article 12.
8. K. Dale, et al. "Video Face Replacement," *ACM Trans. Graphics*, vol. 30, no. 6, 2011, article 130.
9. Y. Yacoob and L. Davis, "Detection and Analysis of Hair," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, 2006, pp. 1164–1169.
10. K.-C. Lee et al., "Markov Random Field Models for Hair and Face Segmentation," *Proc. 8th IEEE Int'l Conf. Automatic Face and Gesture Recognition* (FG 08), IEEE, 2008, pp. 1–6.
11. P. Pérez, M. Gangnet, and A. Blake, "Poisson Image Editing," *ACM Trans. Graphics*, vol. 22, no. 3, 2003, pp. 313–318.
12. A. Agarwala et al., "Interactive Digital Photomontage," *ACM Trans. Graphics*, vol. 23, no. 2, 2004, pp. 294–302.

Template Model Fitting

The segmentation first estimates the locations of the head parts and background. To do this, we fit a parametric template polycurve describing the human head's general shape (see Figure 3a) to the user image. The polygons derived from the template curves define the final three or four regions of interest. These regions become the initial guess for the graph-cut segmentation.

The template model is based on a dataset of 190

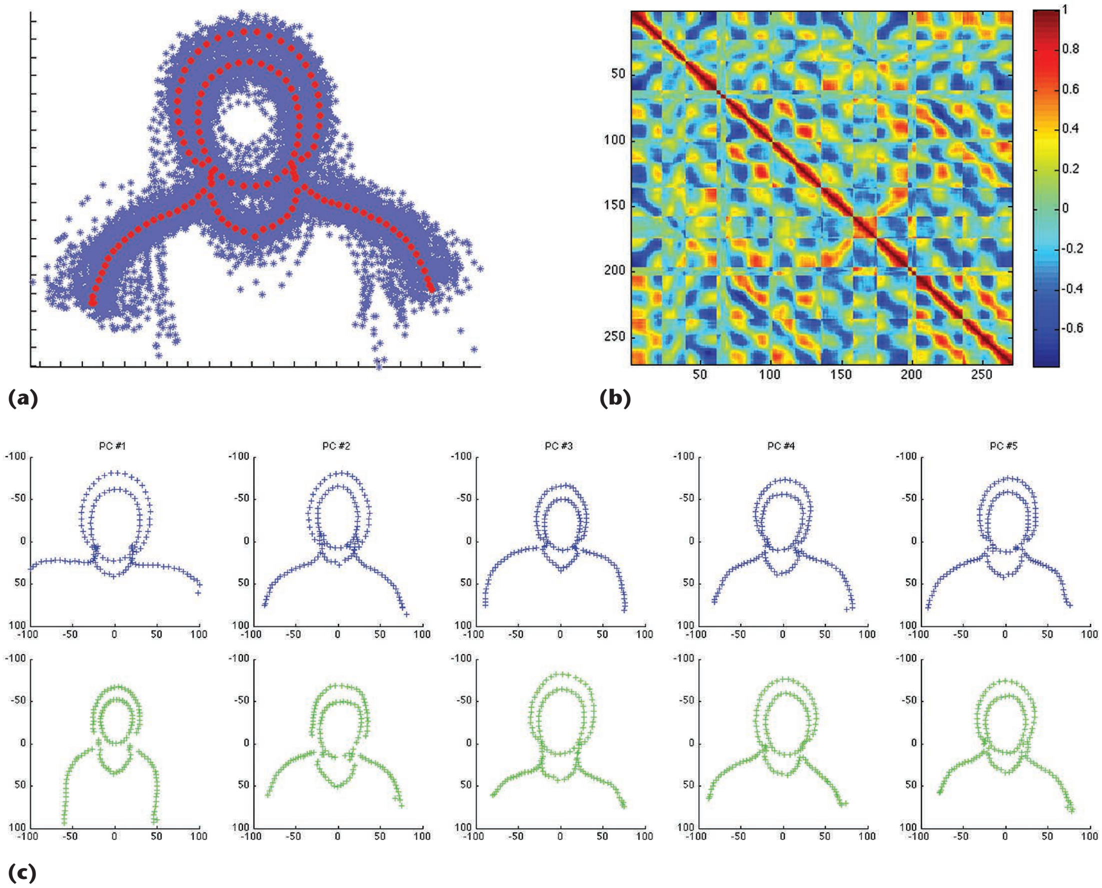


Figure 3. Learning-based head extraction. (a) The mean template head curve on top of the sample distribution. (b) The samples' correlation matrix. (c) Variations along the first five principal components. Blue depicts the mean shape from adding $\sigma\sqrt{\phi_i}$ and green from adding $-\sigma\sqrt{\phi_i}$, where σ is the variance (in this case, 3) and ϕ_i is the *i*th principal component. This shows the variation in the length of the neck, broadness of the shoulders, deepness of the collar, and so on.

polycurves with 135 nodes. An interactive application manually fits these to various human-head images in multiple positions. During data gathering, the system presents the base polycurve template, positioned using the midpoint between the eyes, and asks the user to fit it to a given image. The allowed operations include global scaling, translating, and local deformations to the curves for accurate fitting.

We align the resulting polycurves using the Procrustes metric, following Mikkel Stegmann's process.[2] We use 120 curves as the training set and hold the rest for testing. First, we align all the training shapes to the first shape by finding the translation, rotation, and scale, which minimizes the sum of the squared error between the shapes' nodes. Then, we find the mean shape by finding the average point positions and realign all the shapes to the mean. The parametric template model uses principal-component-analysis construction on the aligned training set. Figure 3b shows the correlation matrix between node coordinates in the 270-dimension sample space; Figure 3c shows variations along the first five principal components.

The fitting first aligns the mean shape to a starting position in the user images. Then, simulated annealing converges toward the template's final shape and position. A fitness function measures the closeness of all 135 nodes to an edge pixel in the image via a Euclidean distance transformation on Canny edge detection of the input image.

A random sampling of the model parameters and potential locations helps guide the annealing. We create the random set of the potential locations by picking *N* positions from a 2D normal distribution with $\sigma = 5s$ around the current position, where σ is the variance and s is a scale factor that depends on the image size. We create the random set of deformations using a normal distribution around one of the *M* highest-ranking principal components. The new location and deformation are chosen if they present better fitness than the current location.

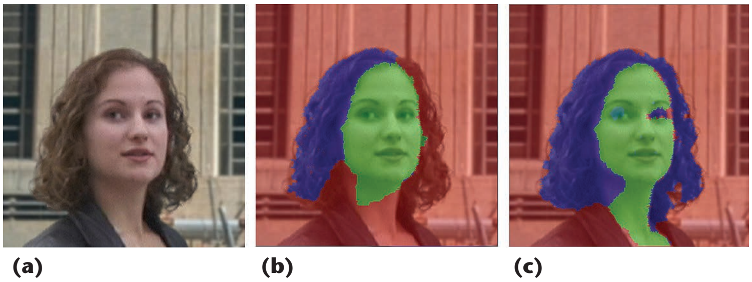


Figure 4. Head segmentation using the color space and cluster space: (a) the input image, (b) the segmentation result with three 50^3-bin 3D histograms over the RGB color space, and (c) the segmentation results with 50 texton clusters. Color-based segmentation fails to segment the skin region owing to color variation, although the skin region has similar texture features. Moreover, although the hair on the left of the face is colored differently from the hair on the right, its texture is similar. Cluster-based segmentation segments the hair correctly.

This process repeats L times. The values we use for M, N, and L in our results are 2, 10, and 20, respectively. We determined these values by evaluating our algorithm on the ground-truth test set. Through the final template position and shape, we derive either three or four regions and use them as an initial guess for the segmentation algorithm. The background, hair, and face are the first three regions; the shoulders and body constitute the fourth.

The Feature Space

We don't consider only gray-level or RGB intensity values as features but use a cluster feature space. Similar to the notion of textons, we create a filter bank of Gabor and difference-of-Gaussian filters of different rotations and variances. Each pixel p is initially represented by a vector $\mathbf{H}_p$, which is a combination of the RGB intensities and the responses to the filters in the bank.

We then cluster the vectors describing each pixel into k clusters, which define the cluster space. Each pixel gets a discrete value in the range of [1, k]. The parameter k has great importance in images with a high variance of textures. We set k = 20 as the default, which worked well in all our experiments. However, users can adjust k when dealing with complex images.

Because we aren't taking into account any spatial information, this clustering doesn't create continuous region segmentation. However, it exposes the latent connection between color and texture. The texton-based cluster space can efficiently distinguish texture features in different regions such as the face, hair, and background. Figure 4 compares using the cluster space and color space to extract the head from the background.

Graph-Cut Optimization

We formulate the segmentation problem as an iterative pixel-labeling problem in terms of energy minimization. The input is a set of pixels X represented in the feature space [1, k] and an initial guess of a label A_x^0 for each pixel. A_x^0 can be {FACE, HAIR, BACKGROUND} and can optionally contain a fourth label, BODY. The goal in this missing data problem is to find the most likely labeling A, given the observations X, by employing an energy function:

$$E\left(A^t\right) = \sum_{x\ \ X} E_p\left(A_x^t, x\right) + \sum_{\substack{x\ \ X \\ x_i\ \ N(x)}} E_n\left(A_x^t, x, A_{x_i}^t, x_i\right), \qquad (1)$$

where the data term, E_p, depicts the penalty of assigning a hypothesized label A_x^t to pixel x at iteration t, and the smoothness term, E_n, describes the penalty for assigning different labels to neighboring pixels. Via graph-cut optimization, the key to effective segmentation lies in defining these two energy terms.

The data term. Previous methods have used a probabilistic model to estimate E_p—for example, histograms or an estimation of the distribution function using Gaussian Mixture Models (GMMs).[2,3] However, we found these methods didn't work well for head extraction on out test set. Consequently, we use a co-occurrence model for E_p. We use not only the estimated probability of label $A^t{}_x$ being assigned to pixel x but also the probabilities of observing the surrounding neighboring pixels. So, we define E_p as

$$E_p\left(A_x^t, x\right) = L\left(A_x^t; x, N(x), \mu\right),$$

where $N(x)$ are x's neighbors and μ is the model parameters. To evaluate the likelihood of the initial label assignment, we set the hypothesis A^t to be the initial guess A^0. We assume a weak naive Bayes model of conditional independence between the neighbors. Therefore, the likelihood function is derived as follows:

$$\begin{aligned} L\left(A_x^t; x, N(x), \mu\right) &= P\left(A_x^t \middle| x, N(x), \mu\right) \\ &\quad P\left(x, N(x), \mu \mid A_x^t\right) P\left(A_x^t\right) \\ &= P\left(x \mid A_x^t\right) P\left(N(x) \mid x, A_x^t\right) P\left(A_x^t\right) \\ &= P\left(x \mid A_x^t\right) P\left(A_x^t\right) \prod_{x_i\ \ N(x)} P\left(x_i \middle| x, A_x^t\right). \end{aligned}$$

We use the log likelihood:[4]

$$\log L\left(A_x^t; x, N(x), \mu\right) = \log\left(P\left(x \middle| A_x^t\right)\right) + \log\left(P\left(A_x^t\right)\right) + \sum_{x_i\ \ N(x)} \log\left(P\left(x_i \middle| x, A_x^t\right)\right).$$

Generally, the system empirically learns probabilities from the image, using normalized histograms of K-Means cluster-number frequencies. $P(x|A_x^t)$ is the posterior probability for observing x under the condition of observing A_x, which is simply the normalized magnitude of A_x, from the appropriate bin in the histogram. We use the co-occurrence measure for $P(x_i|x, A^t{}_x)$, following research in statistical reasoning and probabilistic texture analysis (see the sidebar "The Gray-Level Co-occurrence Matrix"). However, instead of a gray-level co-occurrence matrix, we use a cluster-space co-occurrence matrix (CSCM). We calculate this matrix in four directions and from a one-pixel distance. The estimated conditional probability is

$$P\left(x_i \middle| x, A_x^t\right) = \frac{p\left(x_i x \middle| A_x^t\right)}{P\left(x \middle| A_x^t\right)} = \frac{CSCM_{A_x^t}\left(H_x, H_{x_i}\right)}{\sum_{h\ H} CSCM_{A_x^t}\left(H_x, h\right)},$$

where H is the cluster space of pixels [1, k] and H_p is the cluster-space value of pixel x. The co-occurrence measure considers the effects of neighbors in the data term. This improves the segmentation results on images in which the hair and background are highly similar.

The smoothness term. The literature suggests the smoothness term is simply

$$E_n\left(A_x^t, x, A_{x_i}^t, x_i\right) = \begin{array}{ll} \exp -\frac{\left(I_x - I_{x_i}\right)^2}{2\sigma^2} & A_x^t \neq A_{x_i}^t \\ 0 & A_x^t = A_{x_i}^t \end{array},$$

where I_x is the gray-scale intensity value of x and $\sigma = 3$. We can form a more meaningful term using the texton vectors:

$$E_n\left(A_x^t, x, A_{x_i}^t, x_i\right) = \begin{array}{ll} \exp -\frac{\left\|Tex(x) - Tex(x_i)\right\|}{2\sigma^2} & A_x^t \neq A_{x_i}^t, \\ 0 & A_x^t = A_{x_i}^t \end{array}$$

where σ is effectively the weight of the smoothness term. In our experiments we used $\sigma = 0.1$, and $Tex(x)$ is the texton descriptor—that is, a vector in the feature space that's the corresponding K-Means cluster center assigned for x.

Having defined the two energy terms for a hypothesized labeling of the image, we can minimize the energy in Equation 1 using iterative graph cuts in each iteration setting $A^{t+1} = A^t$.[4]

The Gray-Level Co-occurrence Matrix

The gray-level co-occurrence matrix (GLCM) is a statistical tool for describing texture, set forth by Robert Haralick and his colleagues in 1973.[1] Researchers have used it to develop prominent methods for image retrieval[2] and classification.[3] A GLCM contains the frequencies with which each intensity value appears in occurrence with another intensity value in an image, separated by a given distance and direction. GLCMs are relatively simple to compute and provide high-order information about image textures.

References

1. R. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Systems, Man and Cybernetics*, vol. 3, no. 6, 1973, pp. 610–621.
2. P. Howarth and S. Rüger, "Evaluation of Texture Features for Content-Based Image Retrieval," *Image and Video Retrieval*, LNCS 3115, Springer, 2004, pp. 2134–2135.
3. M. Partio et al., "Rock Texture Retrieval Using Gray Level Co-occurrence Matrix," *Proc. 5th Nordic Signal Processing Symp.*, IEEE, 2002, pp. 8–14.

Finalization and Discussion

To improve the results, we adopt Bayesian matting in the final stage,[5] working on a five-pixel-wide band around the segmentation's boundary. Figure 5 shows two examples that compare different modeling approaches. The three-kernel model gave better results than both the Grabcut method[3] and two-kernel model. We also experimented with a four-kernel model to account for the neck and chest appearing in the image. However, the results for that model didn't show significantly better segmentation of the face and hair.

Because our algorithm runs in nearly real time (an average of 35 ms for a 500 × 500 image), user interaction is possible. Like a paint program, our application lets users scribble on the image, manually marking one of the semantic labels. We treat the scribbles as hard constraints for the graph-cutting by setting the pixel region probability to $P(x|A_{\text{scribble}}) = 1$ for the appropriate region and $P(x|A_{\text{not-scribble}}) = 0$ for the rest.

Head Transfer

Creating a visually appealing composition of a head with a different body is also a challenge. The human eye is highly sensitive to small imperfections in a human figure, especially if discrepancies exist between the head and the torso. The challenge is twofold. First, we must correctly determine the position and scale of the extracted input head. Second, the stitching between the head and the catalog body should be seamless.

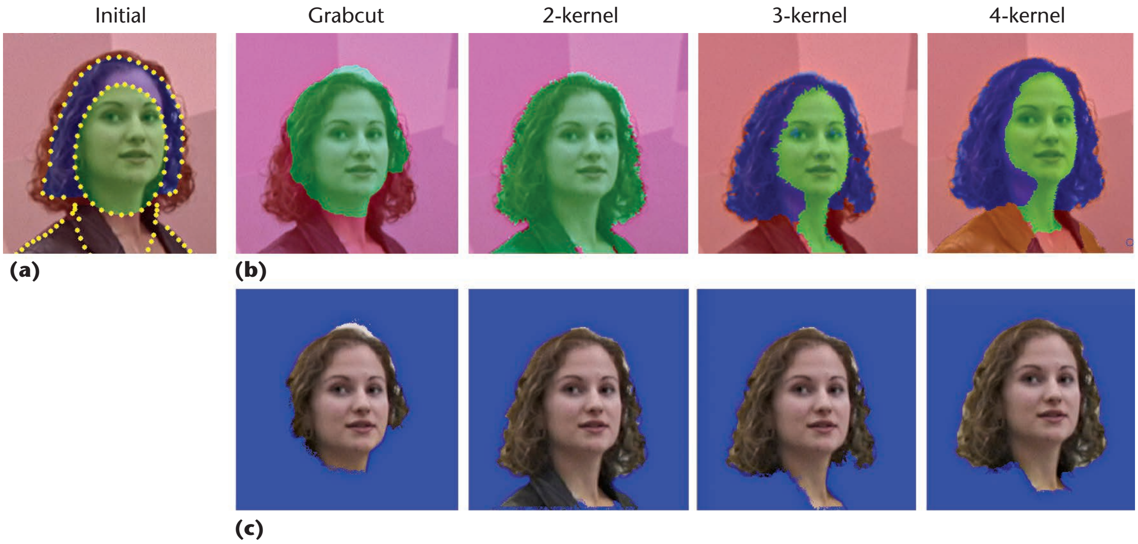


Figure 5. Head extraction using different models. (a) The initial image. (b) The segmentation results. (c) The final extraction results. The three-kernel model gave better results than both the Grabcut method and two-kernel model. However, the four-kernel model didn't show significantly better segmentations of the face and hair.

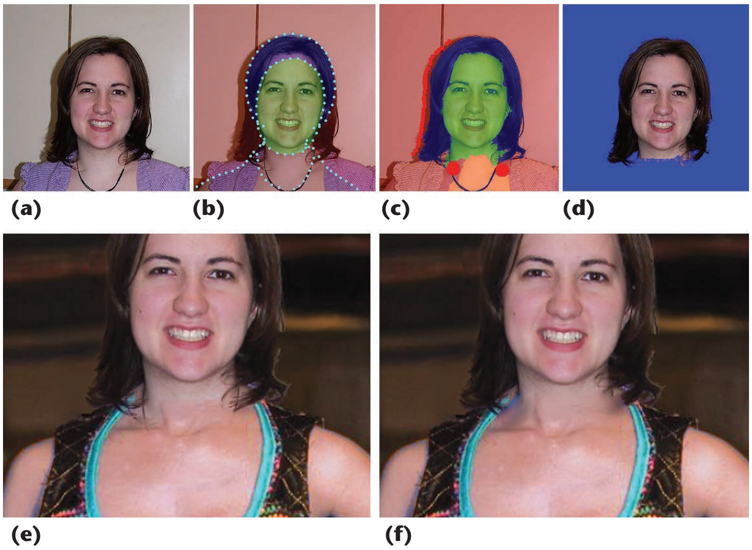


Figure 6. Transferring a head. (a) We start with the input image. (b) The system determines the fitted curve. (c) Segmentation then occurs, with possible user corrections. (d) The system produces the extracted head. (e) If the head is transferred onto the catalog image's body without Poisson blending, an artificial seam is visible. (f) If the head is transferred with Poisson blending, the stitching is seamless.

Head Position and Scale

To determine the best location and scale for a given user head, we learn the image's characteristics during preprocessing. The user trains the system by fitting the head image over two illustrative models of different sizes and poses. The system records characteristic information and uses this to automatically fit the image online.

The system premarks each illustrative model with two anchors, a_{left} and a_{right}, at the base of the neck. After the user fits his or her head over the illustration, the system records v, the 2D direction vector from a_{left} to the left eye. This serves as the head's relative translation. The distance between the eyes is normalized by the distance between the anchors so that the measurements are invariant with the illustration's size: $\hat{v} = v/d_{\text{anchors}}$. The system also records the ratio of the distance between the anchors and the distance between the eyes: $r_{\text{anchor-eye}} = d_{\text{eyes}}/d_{\text{anchors}}$. Finally, the system averages these two measurements over all the fittings of the user's head.

During online head transfer, the transfer algorithm is provided with

- the average measurements we just discussed,
- the distance between the eyes in the user image, and
- the position of a_{left} and a_{right} in the catalog image.

To position the head, we multiply $\hat{v}$ by the anchor distance and recover the hypothesized location of the left eye: $e_{\text{left}} = a_{\text{left}} + \hat{v} \ d_{\text{anchors}}$. We define the scale factor for the head as $\hat{d}_{\text{eyes}} = r_{\text{anchor-eye}} \ d_{\text{anchors}}$.

Composition

To generate a seamless composition, we use Poisson image editing.[6] We create a narrow band around the input head mask's boundary; this constitutes the unknown region Ω. The wider the band is, the better we preserve background features.

However, we might also blur some features such as strands of hair. So, we use the Laplacian of the background image as a guidance field for the Poisson equations. This way, we preserve features of the background model, such as the two prominent sternal head muscles, for a more convincing blend.

Figure 6 illustrates head transfer.

Body Transfer

We must adjust the catalog image to finalize the identity transfer. This involves recoloring the exposed skin to fit the user's skin color, relighting the head, and adjusting the body shape.

Recoloring

Recoloring the catalog image is critical for a convincing identity transfer. First, the catalog skin colors must match the user's skin color. Second, we must adjust the composed head's illumination to fit the catalog photo. We employ two-way recoloring to retain the user's skin color and the catalog photo's scene illumination.

First, preprocessing segments any exposed skin in the catalog image, including arms and legs, so that they can be recolored online. Next, the system learns a statistical model of the skin's color channels for both the catalog image's skin parts and the segmented image of the user's face. We model the color of the skin areas with a GMM in the LAB color space.

The system transfers the user's skin color model to the target catalog skin parts by finding the best pairing of the two sets of Gaussians. Because the number of Gaussians is small, the system selects the best of all possible pairings using symmetric Kullback-Leibler divergence as the distance metric. We match the target image's pixel colors to the source image by calculating a new color:

$$P_{\text{new}} = \sum_i Pr(i \mid P_{\text{old}}) \Sigma_{s_i} (\Sigma_{t_i})^{-1} (P_{\text{old}} - \mu_{t_i}) + \mu_{s_i},$$

where Σ_{s_i} and Σ_{t_i} are the covariance matrices and μ_{t_i} and μ_{s_i} are the means for the *i*th Gaussian of the source and target models, respectively. We obtain $Pr(i|P_{\text{old}})$ by predicting the probability for Gaussian i seeing pixel P_{old}. Figure 7 shows how we change the original skin tone to a lighter or darker tone.

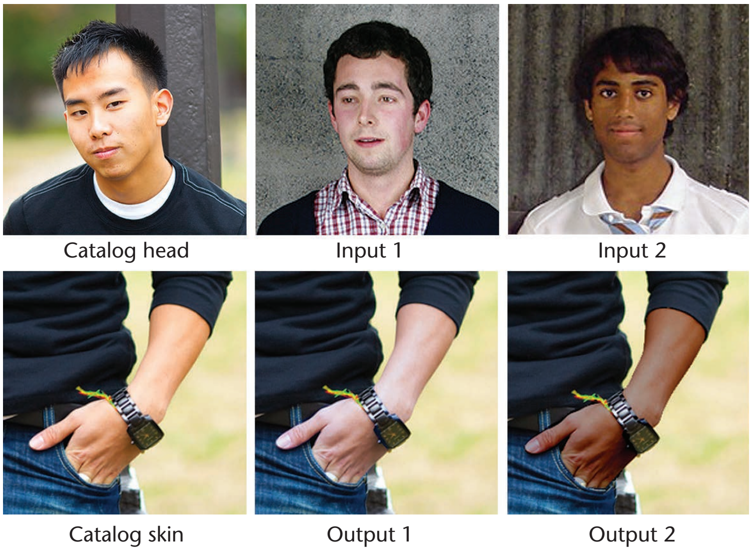


Figure 7. Employing users' face color to recolor the catalog image's skin. Input 1 caused a lighter skin tone; input 2 caused a darker skin tone.

We've experimented with using 1, 3, and 5 Gaussians. Using only one produced the best skin color transfer because the skin has a specific color range. Conversely, the system can learn the luminance channel of the catalog image's skin area and apply it to the user's face, enhancing the resulting image's overall coherence.

Relighting

Often, user images are lit nonuniformly by several directional lights. This can create unwanted discrepancies when the user places a head that's lit from a certain direction into a scene lit from another direction.

We implement relighting for the head that uses an adapted spherical-harmonics model.[7] We fit a canonical 3D model of a human face to the user image to approximate the normal at each pixel. We then solve an overdetermined system of simultaneous linear equations:

$$I(i,j) = \rho(i,j)\,\mathrm{Har}\left(\vec{n}(i,j)\right)^t l\,, \qquad (2)$$

where ρ is the face albedo, l is the unknown coefficients vector of the nine highest-order spherical-harmonic bases, and Har is the response of each corresponding normal to those bases.

To solve for l, we use a method based on singular value decomposition:

1. Fix ρ to a constant calculated directly as the mean intensity value.
2. Solve for an approximate value for l.

Then, we calculate ρ using l, simply by rewriting Equation 2:

$$\rho(i,j) = \frac{I(i,j)}{\mathrm{Har}\left(\vec{n}(i,j)\right)^t l}\,.$$

Using the recovered ρ, we recalculate l. We use l to calculate the final albedo, which serves as the de-lit image of the face.[7]

This process is highly suited for parallel computation and usually runs in less than 9 ms for a 250 × 250 image on a quad-core CPU. Real-time computation lets users interactively adjust the fitting of the 3D model to the image until the results are satisfactory.

This method, however, recovers only the albedo for the face area, which is insufficient for relighting the entire head. To create the complete relighted head, we use gradient domain blending. First, we

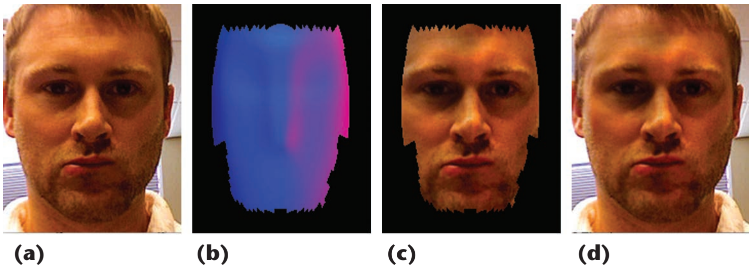


Figure 8. Relighting using spherical harmonics. (a) The original image. (b) The normal map. (c) The albedo. (d) The relighted image. To get a natural result, we use Poisson blending and then Laplacian pyramid blending to blend the relighted face area into the existing face.

use Poisson blending and then Laplacian pyramid blending to blend the relighted face area into the existing face to get a natural result. Figure 8 shows how we relight a head.

Body Reshaping

For body reshaping, we employ Shizhe Zhou and his colleagues' method.[1] Offline, we match the contour of the body in the catalog image to the contour of a general 3D human model. We then use the given measurements of the user's body to reshape this 3D model's height, shoulders, waist, and approximate weight (slim, medium, or heavy).

Because a user's measurements don't tend to change dramatically over time, they can be stored once during setup for each user. The system can then automatically warp any catalog image to match the required user measurements.

Implementation and Evaluation

We implemented our system using C++ on an Intel 2.8-GHz CPU with 2 Gbytes of RAM. For an input image with 800 × 600 resolution, the system typically takes approximately 30 seconds to extract the head and one second to complete the composition, excluding user interaction. The complete process, including user interaction, usually takes less than three minutes and produces a five-image catalog. Figure 9 shows some results.

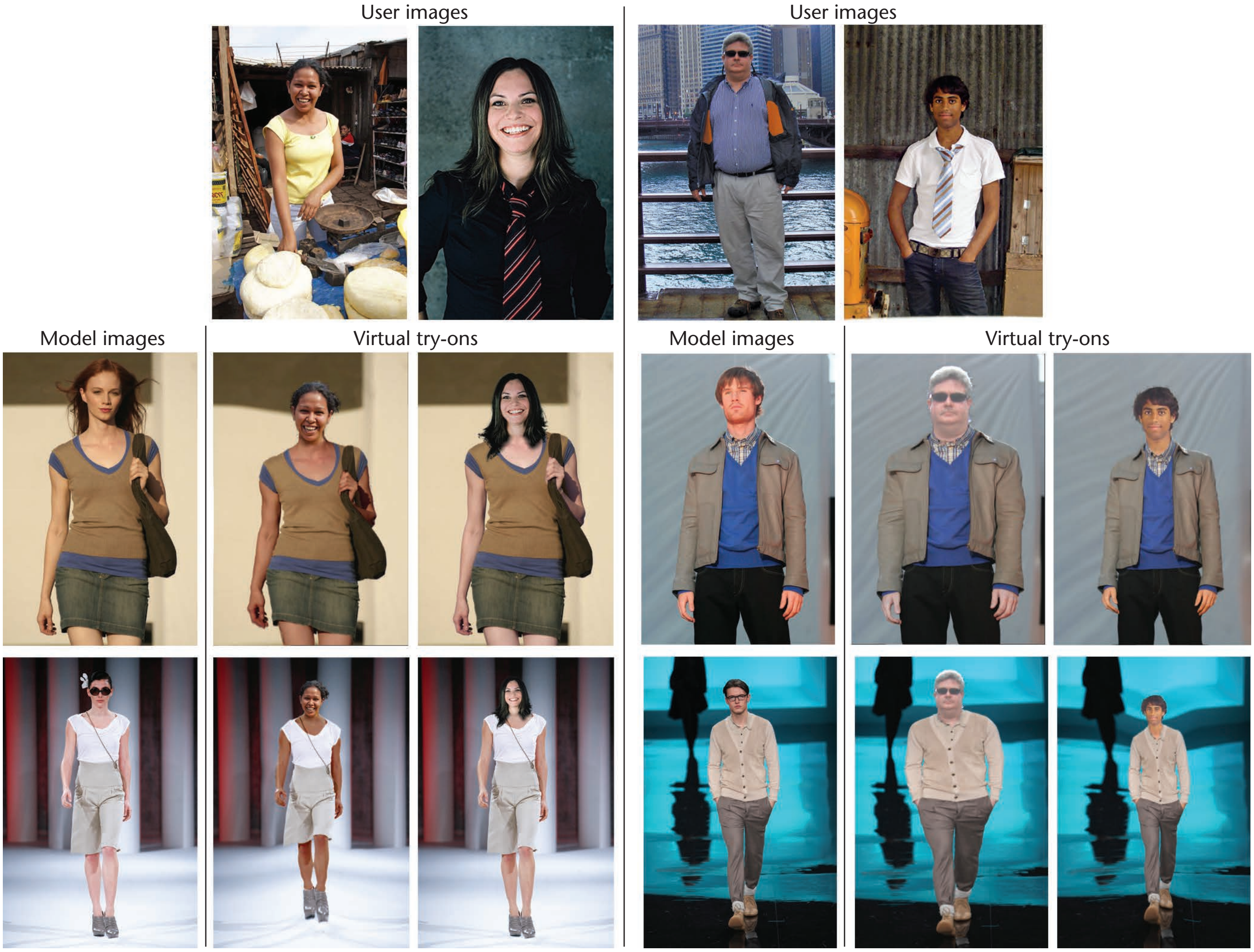


Figure 9. The results of transferring the users' faces and characteristics to the catalog images. The complete process, including user interaction, usually takes less than three minutes.

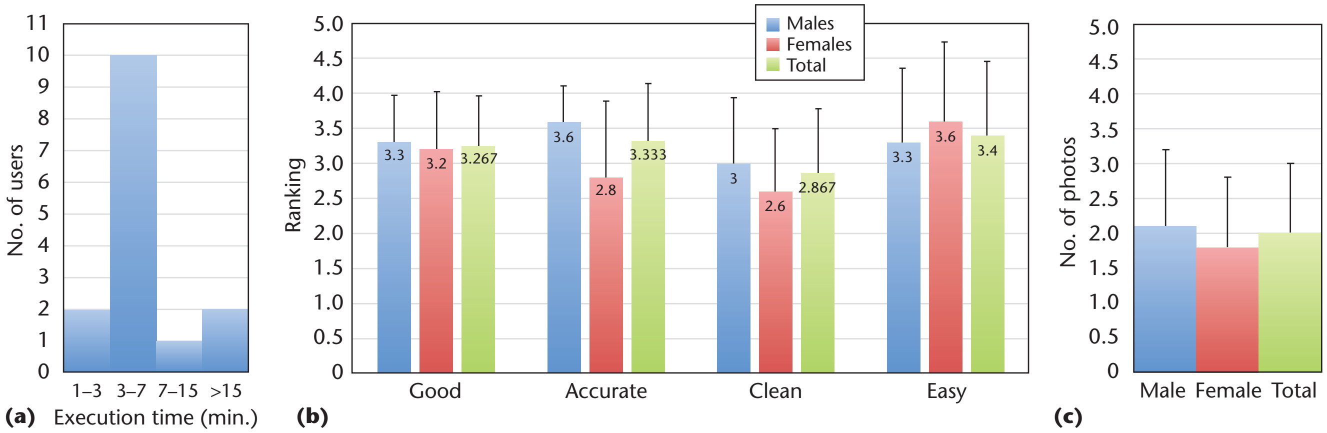


Figure 10. The results for user study A. (a) The execution time. (b) The measures of quality (1 is the worst ranking; 5 is the best). (c) An estimation of how many photos a stranger would consider real. For an explanation of the measures of quality, see the section "User Study A" in the main article.

To evaluate our system, we performed two user studies.

User Study A

By email, we recruited 15 participants (five females and 10 males) ranging from 20 to 35 years old. Each participant watched a short instructional video before using the system to create a personalized catalog. Figure 10a shows the execution times.

After viewing their catalog, they answered four general questions:

- How good are the results? (How much did they appreciate the results?)
- How clean are the results? (Did they see any visible artifacts?)
- How accurate are the results? (Was the head positioned correctly?)
- How easy was the interaction?

The answers consisted of rankings ranging from 1 (worst) to 5 (best). Figure 10b shows the results.

Finally, the participants estimated how many of their five final results would look real in the eyes of a stranger (see Figure 10c). On average, they estimated that at least two of their personalized images would seem real. The male users were more satisfied with the results' accuracy. We believe this is because most male users have short hair, which is easier to segment and therefore more accurate.

User Study B

This study involved more than 100 randomly selected participants. They viewed 21 images and indicated which ones seemed real (untouched) and which ones seemed fake (created by our system). Two of the images served as training for the users, seven were real, and 12 were created by our method. Table 1 shows that the participants discerned the real images from the fake ones almost as many times as they labeled the real images as fake.

Limitations

Our method suffers from various limitations during extraction and composition. First, the reshaping of the body distorts some of the background and some of the natural flow of clothing. To cope with background distortion, we can first segment the catalog model from its background and create two layers:

- the foreground model, whose identity will be changed, and
- the background, where the image should simply be pasted on.

This involves some composition challenges, which we leave for future research.

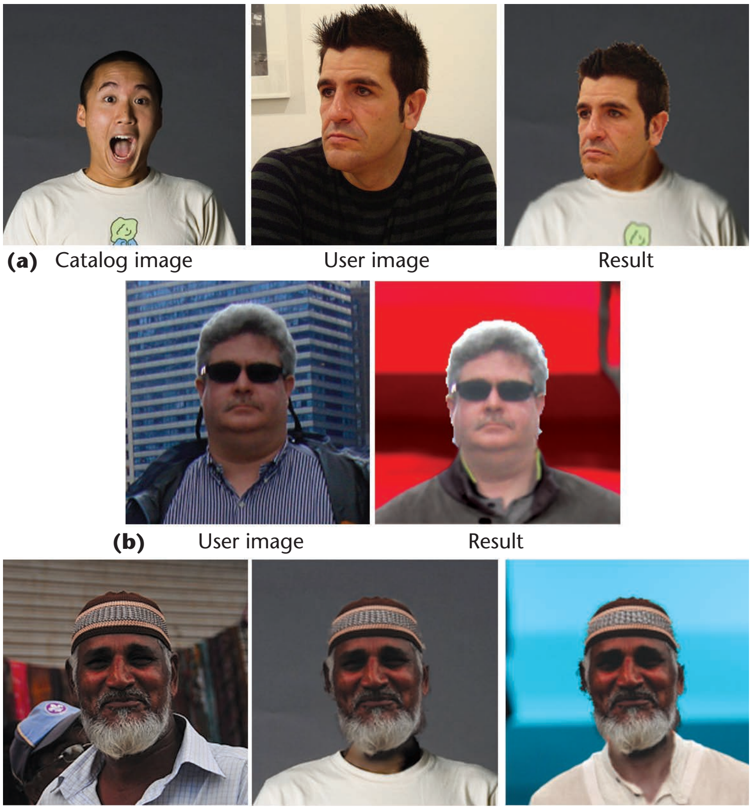
We also assume that both the catalog and user images involve relatively simple poses and don't contain occlusions or extreme conditions. Extreme differences in head poses can generate unrealistic results (see Figure 11a), and input images taken in extreme lighting conditions can cause unrealistic skin recoloring. One visible artifact is washout, which is caused by moving the distribution of the luminosity channel from dark to bright (see Figure 11b). Some hairstyles also prove difficult to segment and can impair neck estimation. Long hairstyles that have many strands or facial hair that occludes parts of the neck are usually more difficult to segment (see Figure 11c).

Our method is an alternative to graphics designers' labor-intensive work of transferring human heads in images. It also gives new insight into automatic, nonparametric human-image segmentation. In addition, we believe it will benefit the

Table 1. The confusion matrix for user study B.*

	Actual	
Observed	**Real**	**Fake**
Real	346	483
Fake	235	513

"Observed" refers to how the participants labeled the images; "Actual" refers to the images' true values. For example, 483 participants mislabeled fake images as real. (Fake images were created by our system; real images were untouched.)


(a) Catalog image User image Result

(b) User image Result

(c) User image Result 1 Result 2

Figure 11. Some limitations of our system. (a) Extremely different poses can produce awkward, unrealistic results. (b) Extreme lighting can also create unrealistic results—for example, light skin in a heavy shade can cause washout. (c) Facial hair can produce unwanted blurring in the seam region around the neck.

online user experience in situations such as social interaction via avatars and personalized games. Our method can also be ported easily to mobile devices for use in real-time manipulation of portrait photographs or offered as a Web service.

Future research will involve generating garment-fitting results for arbitrary input images while removing some of the limitations on subject pose and image orientation. Furthermore, we believe that the ultimate goal for a personalized catalog is to provide a full video experience. Perhaps the new modalities in end-user imaging technology, such as depth images and stereoscopic photography, could be employed to reach that goal.

References

1. S. Zhou et al., "Parametric Reshaping of Human Bodies in Images," *ACM Trans. Graphics*, vol. 29, no. 4, 2010, article 126.
2. M.B. Stegmann, "Active Appearance Models: Theory, Extensions and Cases," master's thesis, Dept. of Mathematical Modelling, Technical Univ. Denmark, 2000.
3. C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 309–314.
4. Y. Boykov and M. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," *Proc. 8th IEEE Int'l Conf. Computer Vision* (ICCV 01), IEEE, 2001, pp. 105–112.
5. Y.-Y. Chuang et al., "A Bayesian Approach to Digital Matting," *Proc. 2001 IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition* (CVPR 01), IEEE CS, 2001, pp. 264–271.
6. P. Pérez, M. Gangnet, and A. Blake, "Poisson Image Editing," *ACM Trans. Graphics*, vol. 22, no. 3, 2003, pp. 313–318.
7. Y. Wang et al., "Face Re-lighting from a Single Image under Harsh Lighting Conditions," *Proc. 2007 IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 07), IEEE CS, 2007, pp. 1–8.

***Roy Shilkrot** is a PhD candidate and research assistant at the MIT Media Lab. His research interests include computer graphics and vision, with applications in augmented reality. Shilkrot received an MSc in computer science from Tel Aviv University. Contact him at roys@media.mit.edu.*

***Daniel Cohen-Or** is a professor at Tel Aviv University's school of computer science. His research interests include shape analysis and synthesis, shape modeling, surface reconstruction, and computer graphics, particularly modeling and synthesis. Cohen-Or received a PhD in computer science from the State University of New York at Stony Brook. Contact him at dcor@tau.ac.il.*

***Ariel Shamir** is an associate professor at the Efi Arazi School of Computer Science at the Interdisciplinary Center, Herzliya. His research interests include computer graphics, geometric modeling, image processing, and machine learning. Shamir received a PhD in computer science from the Hebrew University of Jerusalem. Contact him at arik@idc.ac.il.*

***Ligang Liu** is a professor at the School of Mathematical Sciences at the University of Science and Technology of China. His research interests include digital geometric processing, computer graphics, and image processing. Liu received a PhD in computer-aided geometric design and computer graphics from Zhejiang University. Contact him at lgliu@ustc.edu.cn.*