Image-based Shaving

Minh Hoai Nguyen, Jean-Francois Lalonde, Alexei A. Efros and Fernando De la Torre

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

Abstract

Many categories of objects, such as human faces, can be naturally viewed as a composition of several different layers. For example, a bearded face with glasses can be decomposed into three layers: a layer for glasses, a layer for the beard and a layer for other permanent facial features. While modeling such a face with a linear subspace model could be very difficult, layer separation allows for easy modeling and modification of some certain structures while leaving others unchanged. In this paper, we present a method for automatic layer extraction and its applications to face synthesis and editing. Layers are automatically extracted by utilizing the differences between subspaces and modeled separately. We show that our method can be used for tasks such beard removal (virtual shaving), beard synthesis, and beard transfer, among others.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation

1. Introduction

"But how would I look without a beard?" – this agonizing question must be familiar to any long-term beard wearer considering the momentous decision of shaving it all off. Indeed, unlike other, more continuous facial transformations, such as change in expression or aging, the presence or absence of a beard changes a person's appearance dramatically and has a huge effect on our ability to recognize him. No wonder than that growing or shaving a beard is a favorite form of disguise. One of the goals of this paper is to help indecisive bearded men by showing what they might look like clean-shaven (Figure 1).

Of course, predicting the appearance of a person without a beard given only a photograph of that person with a beard is an ill-posed problem. Particular individual characteristics, moles, scars, a double chin, would be impossible to reconstruct if they are not visible in the photograph. Therefore, our aim is only to synthesize a plausible version of what the occluded parts of a person's face might look like. The idea is to exploit the statistical redundancies in facial appearance: just as the left half of a face is an extremely good predictor for what the right half might look like, we believe that the upper part of the face should provide enough information to generate a good guess for the appearance of the lower part. One way to approach this problem is with a pure machine learning solution: given enough beard/no beard training image pairs, it should be possible to learn a regressor that estimates one given the other. However, this approach requires a large amount of training data - pairs of beard/no beard images of the same person, in similar pose, under similar lighting, etc. This is something that would be very hard

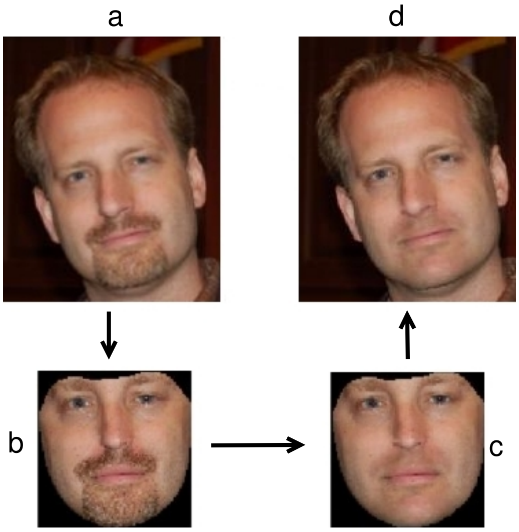

Figure 1: *Beard removal process. (a): a bearded face; (b): canonical face obtained from (a); (c): canonical face with the beard removed; (d): final result of beard removal.*

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

to obtain in large quantities. Instead, we would like our approach to work given just a large labeled collection of photographs of faces, some with beards, some not. The set of non-beard photographs together would provide a face model while bearded images would serve as the structural deviation from that model.

So, how can we model faces from a set of photographs? Appearance Models (AMs) such as Morphable Models or Active Appearance Models (AAMs) have been extensively used for parameterizing the space of human faces [BV99,CET98,Pig99,BSVS04,dlTB03b,VP97,TP91, PV07, WLS*04]. AMs are very appealing modeling tools because they are based on a well-understood mathematical framework and because of their ability to produce photo-realistic images. However, direct application of standard AMs to the problem of beard removal is not likely to produce satisfactory results due to several reasons. First, a single holistic AM is unlikely to capture the differences between bearded and non-bearded faces. Second, AMs do not allow modification of some structures while leaving others unchanged. To alleviate such problems, part-based and modular AMs have been proposed. Pentland *et. al.* [PMS94, MP97] used modular eigenspaces and showed an increase in recognition performance. Pighin [Pig99] manually selected regions of the face and showed improvements in tracking and synthesis. Recently, Jones & Soatto [JS05] presented a framework to build layered subspaces using PCA with missing data. However, previous work requires manually defining/labeling parts or layers which are tedious and time-consuming.

In this paper, we propose a method for automatic layer extraction and modeling from a set of images. Our method is generic and should be applicable not just to beards but to other problems where a layer presentation might be helpful (e.g. faces with glasses). For the sake of simple explanation, however, we will describe our method using the beard removal example throughout this paper.

Our algorithm can be summarized as follows. We start by constructing two subspaces, a beard subspace from a set of bearded faces, and a non-beard subspace from a set of non-bearded faces. For each bearded face, we compute a rough estimate for the corresponding non-bearded face by finding the robust reconstruction in the non-beard subspace. We then build a subspace for the differences between the bearded faces and their corresponding reconstructed non-bearded versions. The resulting subspace is the beard layer subspace that characterizes the differences between beard and non-beard. Now, every face can be decomposed into three parts. One part can be explained by the non-beard subspace, another can be explained by the beard layer subspace, and the last part is the "noise" which cannot be explained by either beard or non-beard subspaces. Given the decomposition, one can synthesize images by editing the part contributed by the beard layer subspace. To generate final renderings, we perform a post-processing step for refinement. The post-processing step does beard segmentation via Graph Cut [BVZ01] exploiting the spatial relationships among beard pixels.

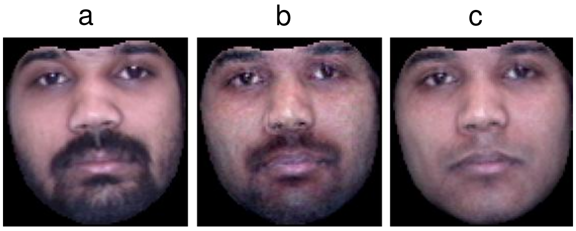

Figure 2: *Subspace fitting, (a): original image, (b): fitting using linear projection, (c): fitting using iteratively reweighted least squares.*

The rest of the paper is structured as follows. Section 2 describes how robust statistics can be used for rough estimation of non-bearded faces from bearded ones. The next section explains how to model the differences between two subspaces. Section 4 discusses the use of Graph Cut for beard segmentation. Section 5 shows another application of our method: beard addition. Information about the database and other implementation details are provided in Section 6.

2. Removing Beards with Robust Statistics

In this section, we show how robust statistics on subspaces can be used to detect and remove beards in faces. Let us first describe some notations used in this paper. Bold upper-case letters denote matrices; bold lower-case letters denote column vectors; non-bold letters represent scalar variables. $\mathbf{a}_i$ and $\tilde{\mathbf{a}}_i^T$ are the i^{th} column and row of matrix $\mathbf{A}$ respectively, a_{ij} is the entry in i^{th} row and j^{th} column of $\mathbf{A}$; u_i is the i^{th} element of column vector $\mathbf{u}$. $||\mathbf{u}||_2^2$ denotes the squared L_2 norm of $\mathbf{u}$, $||\mathbf{u}||_2^2 = \mathbf{u}^T\mathbf{u}$.

Let $\mathbf{V} \in \Re^{d\times n}$ be a matrix of which each column is a vectorized image of a face without a beard . d denotes the number of pixels of each image and n the number of samples (in our experiments, $d = 92*94 = 8648$, and $n = 738$; see Section 6 for more details). Let $\mathbf{x}$ be a face image with a beard and $\mathbf{x}^*$ the same image with the beard removed. A naïve approach to remove the beard would be to reconstruct the face $\mathbf{x}$ in the non-beard subspace $\mathbf{V}$. That is,

$$\mathbf{x}^* = \mathbf{V}\hat{\mathbf{c}}$$
$$\text{with} \quad \hat{\mathbf{c}} = \arg\min_{\mathbf{c}} ||\mathbf{x} - \mathbf{V}\mathbf{c}||_2^2 = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{x}$$

Unfortunately, the beard typically is a significant part of the face and can strongly bias the estimate of the coefficients $\mathbf{c}$. Figure 2a shows a bearded face, Figure 2b is the projection of Figure 2a into the non-beard subspace. The projection makes the beard regions lighter while darkening the other regions, but it does not effectively remove the beard.

To address this problem, the beards can be treated as outliers of the non-beard subspace $\mathbf{V}$. Using an M-estimator [Hub81], we can remove the influence of the outliers from the projection [BJ98, dlTB03a]. The M-estimator uses a robust function (e.g. Geman-McClure [GM87]) rather than a quadratic one and minimizes:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \sum_{i=1}^{d} \rho(x_i - \tilde{\mathbf{v}}_i^T \mathbf{c}, \sigma) \qquad (1)$$

Here, $\rho(x,\sigma) = \frac{x^2}{x^2+\sigma^2}$ is the Geman-McClure function, and $\tilde{\mathbf{v}}_i^T$ denotes the i^{th} row of $\mathbf{V}$. The above optimization function can be solved approximately using Iteratively Re-weighted Least Square (IRLS) method [BT74, HW77, dlTB03a]. We review the IRLS procedure, an approximate and iterative algorithm to solve an M-estimation problem, originally proposed by Beaton and Turkey [BT74] and extended by [HW77, Li85]. An algorithm to solve Equation 1 with fixed σ is equivalent to minimizing a weighted least squares problem iteratively [Li85]. At the k^{th} iteration, we solve a weighted least squares problem in which the importance of pixels are weighted differently using a diagonal weight matrix $\mathbf{W} \in \Re^{d\times d}$:

$$\begin{aligned} \mathbf{c}^{(k)} &= \arg\min_{\mathbf{c}} ||\mathbf{W}(\mathbf{x} - \mathbf{V}\mathbf{c})||_2^2 \\ &= (\mathbf{V}^T\mathbf{W}^T\mathbf{W}\mathbf{V})^{-1}\mathbf{V}^T\mathbf{W}^T\mathbf{W}\mathbf{x} \end{aligned}$$

The matrix $\mathbf{W}$ is calculated at each iteration as a function of the previous residual $\mathbf{e} = \mathbf{x} - \mathbf{V}\mathbf{c}^{(k-1)}$ and is related to the "influence" [HRRS86] of pixels on the solution. Each element, w_{ii}, of $\mathbf{W}$ will be equal to

$$w_{ii} = \frac{1}{2e_i}\frac{\partial\rho(e_i,\sigma)}{\partial e_i} = \frac{\sigma^2}{(e_i^2+\sigma^2)^2}$$

The parameter σ of the robust function is also updated at every iteration: $\sigma := 1.4826 * \text{median}(\{|x_i - x_i^*| : i = \overline{1,d}\})$ [Hub81]. Pixels that belong to beard regions would not be reconstructed well by the non-beard subspace. The weights of those pixels and their influence on the fitting process will decrease as more and more iterations are run. The non-bearded face $\mathbf{x}^*$ is taken to be $\mathbf{V}\hat{\mathbf{c}}$ with $\hat{\mathbf{c}}$ is the limit of the sequence $\mathbf{c}^{(k)}, k = 1,2,\ldots$.

Figure 2c is the non-bearded reconstruction of Figure 2a after convergence. As can be seen, this method produces substantially better result than the result of naïve approach given Figure 2b.

3. Factorizing Layered Spaces

While beards are outliers of the non-beard subspace, the converse is not necessary true. In other words, not all outliers of the non-beard subspace are beards. In general, robust statistical methods discussed in the previous section often do not provide visually satisfactory results as characteristic moles and scars are also removed. To overcome this problem, in this section, we propose an algorithm to factorize layers of spatial support given two training sets $\mathbf{U} \in \Re^{d\times n_1}$ and $\mathbf{V} \in \Re^{d\times n_2}$ into common and non-common layer subspaces (e.g. beard region and non-beard region). That is, given $\mathbf{U}$ and $\mathbf{V}$, we will factorize the spaces into a common (face) and non-common subspace (beard). Although our method is generic, we will illustrate the procedure by continuing working with beard and non-beard subspaces.

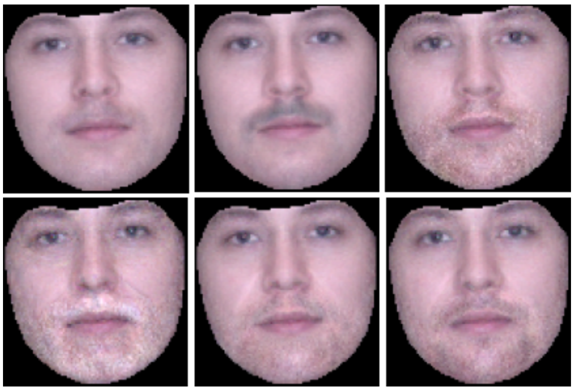

Figure 3: *The first six principal components of* $\mathbf{B}$*, the subspace for the beard layer. The principal components are scaled (for visibility) and imposed on the mean image of non-beard subspace.*

To obtain the non-common subspace between the subspaces of faces with and without beards, we first need to compute the difference between every bearded face and its corresponding non-bearded version. In other words, using the procedure from the previous section, for every bearded face $\mathbf{u}_i$, we compute $\mathbf{u}_i^*$, the robust reconstruction in the non-beard subspace. Let $\mathbf{D}$ denote $[(\mathbf{u}_1 - \mathbf{u}_1^*)\ldots(\mathbf{u}_n - \mathbf{u}_n^*)]$, $\mathbf{D}$ defines the outlier subspace of the non-beard subspace. Beards are considered outliers of the non-beard subspace, but they are not outliers of the beard subspace. As a result, we can perform Principal Component Analysis (PCA) [Jol02] on $\mathbf{D}$ to filter out the outliers that are common to both subspaces, retaining only the components for the beard layer. Let $\mathbf{B}$ denote the principal components obtained using PCA retaining 95% energy. $\mathbf{B}$ defines a beard layer subspace that characterizes the differences between beard and non-beard subspaces. Figure 3 shows the first six principal components of $\mathbf{B}$ superimposed on the mean image of the non-beard subspace.

Now given any face $\mathbf{d}$ ($\mathbf{d}$ could be a bearded or non-bearded face, and it is not necessary part of the training data), we can find coefficients $\hat{\alpha}, \hat{\beta}$ such that:

$$\hat{\alpha}, \hat{\beta} = \arg\min_{\alpha,\beta} ||\mathbf{d} - \mathbf{V}\alpha - \mathbf{B}\beta||_2^2$$

Note that $\hat{\alpha}, \hat{\beta}$ can be found by solving a system of linear equations:

$$\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \left(\begin{bmatrix} \mathbf{V}^T \\ \mathbf{B}^T \end{bmatrix} \begin{bmatrix} \mathbf{V} & \mathbf{B} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{V}^T \\ \mathbf{B}^T \end{bmatrix} \mathbf{d}\,.$$

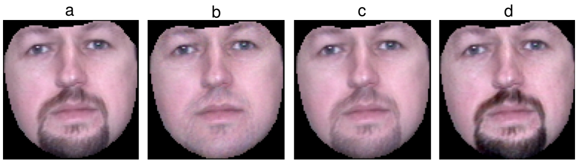

Figure 4: *Synthesis faces generated by manipulating the contribution of subspace* **B***. (a): an original face, (b): the same face with the beard removed, (c): the face with the beard part is reduced by half, (d): the face with the beard part is enhanced 50%.*

Here $\left(\begin{bmatrix} \mathbf{V}^T \\ \mathbf{B}^T \end{bmatrix} [\,\mathbf{V}\ \mathbf{B}\,]\right)^{-1} \begin{bmatrix} \mathbf{V}^T \\ \mathbf{B}^T \end{bmatrix}$ can be replaced by the pseudo-inverse of $[\,\mathbf{V}\ \mathbf{B}\,]$ if $\left(\begin{bmatrix} \mathbf{V}^T \\ \mathbf{B}^T \end{bmatrix} [\,\mathbf{V}\ \mathbf{B}\,]\right)^{-1}$ does not exist.

Now let $\bar{\mathbf{d}}$ denote $\mathbf{d} - \mathbf{V}\hat{\alpha} - \mathbf{B}\hat{\beta}$, we have:

$$\mathbf{d} = \mathbf{V}\hat{\alpha} + \mathbf{B}\hat{\beta} + \bar{\mathbf{d}} \tag{2}$$

Thus any face can be decomposed into three parts, the non-beard part $\mathbf{V}\hat{\alpha}$, the difference between beard and non-beard $\mathbf{B}\hat{\beta}$ and the residual $\bar{\mathbf{d}}$. The residual is the part that cannot be explained by either the non-beard subspace nor the beard subspace. The residual exists due to several reasons such as insufficiency of training data, existence of noise in the input image $\mathbf{d}$, or existence of characteristic moles or scars.

Given the above decomposition, we can create synthetic faces. For example, consider a family of new faces $\mathbf{d}(\lambda)$ generated by $\mathbf{d}(\lambda) = \mathbf{V}\hat{\alpha} + \lambda\mathbf{B}\hat{\beta} + \bar{\mathbf{d}}$, where λ is a scalar variable. We can remove the beard, reduce the beard or enhance the beard by setting λ to appropriate values. Figure 4a shows an original image and Figure 4b, 4c, 4d are generated faces with λ are set to 0, 0.5, and 1.5 respectively.

Figure 5 shows some results of beard removal using this approach for beard removal ($\lambda = 0$). The results are surprisingly good considering the limited amount of training data and the wide variability of illumination. However, this method is not perfect (see Figure 6); it often fails if the amount of beard exceeds some certain threshold which is the break-down point of robust fitting. Another reason is the insufficiency of training data. We believe the results will be significantly better if more data is provided.

4. Beard Mask Segmentation Using Graph Cuts

So far, our method for layer extraction has been generic and did not rely on any spatial assumptions about the layers. However, for tasks such as beard removal, there are several spatial cues that can provide additional information. In this section, we propose a post-processing step that exploits one of such spatial cues: a pixel is likely to be a beard pixel if most of its neighbors are beard pixels, and vise versa. What this post-processing step does is to segment out the contiguous beard regions.

To understand the benefits of beard segmentation, let us reconsider the current beard removal method. Recall that an image $\mathbf{d}$ can be decomposed into $\mathbf{V}\hat{\alpha} + \mathbf{d}^B + \bar{\mathbf{d}}$, where $\mathbf{d}^B$ denotes $\mathbf{B}\hat{\beta}$ which is the contribution of the beard layer. Currently, the corresponding non-bearded face $\mathbf{d}^*$ of $\mathbf{d}$ is taken to be $\mathbf{d} - \mathbf{d}^B$. Ideally, one would like the entries of $\mathbf{d}^B$ that correspond to non-beard pixels to vanish. In practice, however, this is not usually the case because we are working with real world data. Entries of $\mathbf{d}^B$ corresponding to non-beard pixels usually have small magnitudes but not exactly zero. Performing beard segmentation using the aforementioned spatial constraint will help to reduce the effect of this problem.

We formulate the beard segmentation task as a graph labeling problem. For a face image $\mathbf{d}$, construct a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is the set of nodes corresponding to the pixels of $\mathbf{d}$ and $\mathcal{E}$ is the set of edges representing the 4-connectivity neighboring relationships of the pixels. The labeling problem is to assign a label $l_i (\in \{1(\text{beard})\ ,0(\text{non-beard})\})$ for each node $i \in \mathcal{V}$. We would like to find a labeling $L = \{l_i | i \in \mathcal{V}\}$ that minimizes a Gibbs energy function $E(L)$:

$$E(L) = \sum_{i \in \mathcal{V}} E_i^1(l_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}^2(l_i, l_j) \tag{3}$$

Here $E_i^1(l_i)$ is the unary potential function for node i to receive label l_i and $E_{ij}^2(l_i, l_j)$ is the binary potential function for labels of adjacent nodes. In our experiment, we define $E^1(.), E^2(.,.)$ based on $\mathbf{d}^B$, the contribution of the beard layer. Let $\mathbf{d}_i^B$ denote the entry of $\mathbf{d}^B$ that corresponds to node i. Define $E^1(.), E^2(.,.)$ as follows:

$$E_i^1(l_i) = \begin{cases} 0 & \text{if } l_i = 1 \\ |\mathbf{d}_i^B| - a & \text{if } l_i = 0 \ \& \ ||\mathbf{d}_i^B| - a| \le b \\ b & \text{if } l_i = 0 \ \& \ |\mathbf{d}_i^B| - a > b \\ -b & \text{if } l_i = 0 \ \& \ |\mathbf{d}_i^B| - a < -b \end{cases}$$

$$E_{ij}^2(l_i, l_j) = \begin{cases} 0 & \text{if } l_i = l_j \\ \frac{b}{2} & \text{if } l_i \ne l_j \end{cases}$$

The unary and binary potential functions are defined intuitively. A beard pixel is a pixel that can be explained well by the beard subspace but not by the non-beard subspace. As a result, one would expect $|\mathbf{d}_i^B|$ is large if pixel i is a beard pixel. The unary potential function is defined to favor the beard label if $|\mathbf{d}_i^B|$ is high ($> a$). Conversely, the unary potential favors the non-beard label if $|\mathbf{d}_i^B|$ is small ($< a$). To limit the influence of individual pixel, we limit the range of the unary potential function from $-b$ to b. Beard is not necessary formed by one blob of connected pixels; however, a pixel is more likely to be beard if most of their neighbors are. We design the binary potential function to encode that preference. In our implementation, a, b are chosen empirically as 8 and 4 respectively.

The exact global optimum solution of the optimization problem in Eq.3 can be found efficiently using graph

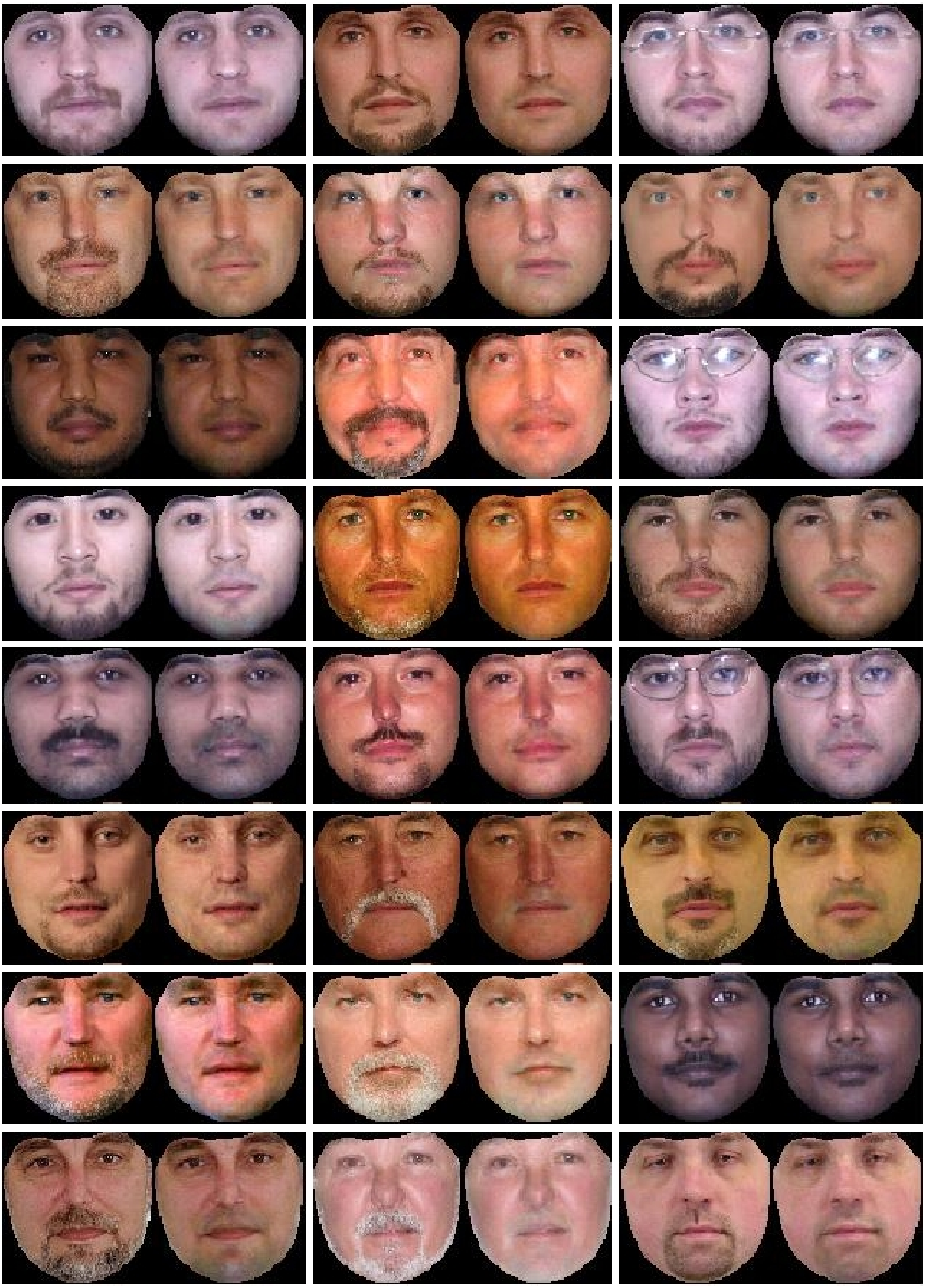

Figure 5: *Results of beard removal using layer subtraction. This figure displays 24 pairs of images. The right image of each pair is the result of beard removal of the left image.*

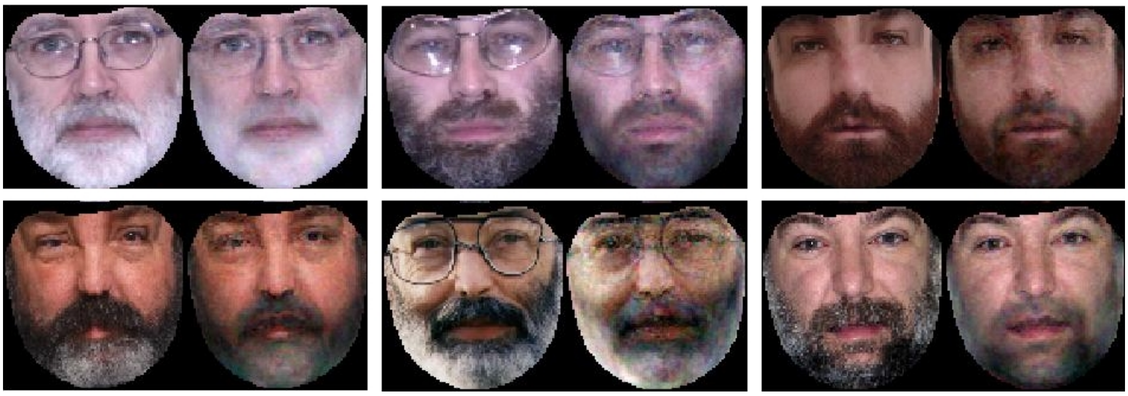

Figure 6: *Beard removal results: six failure cases. The right image of each pair is the beard removal result of the left image.*

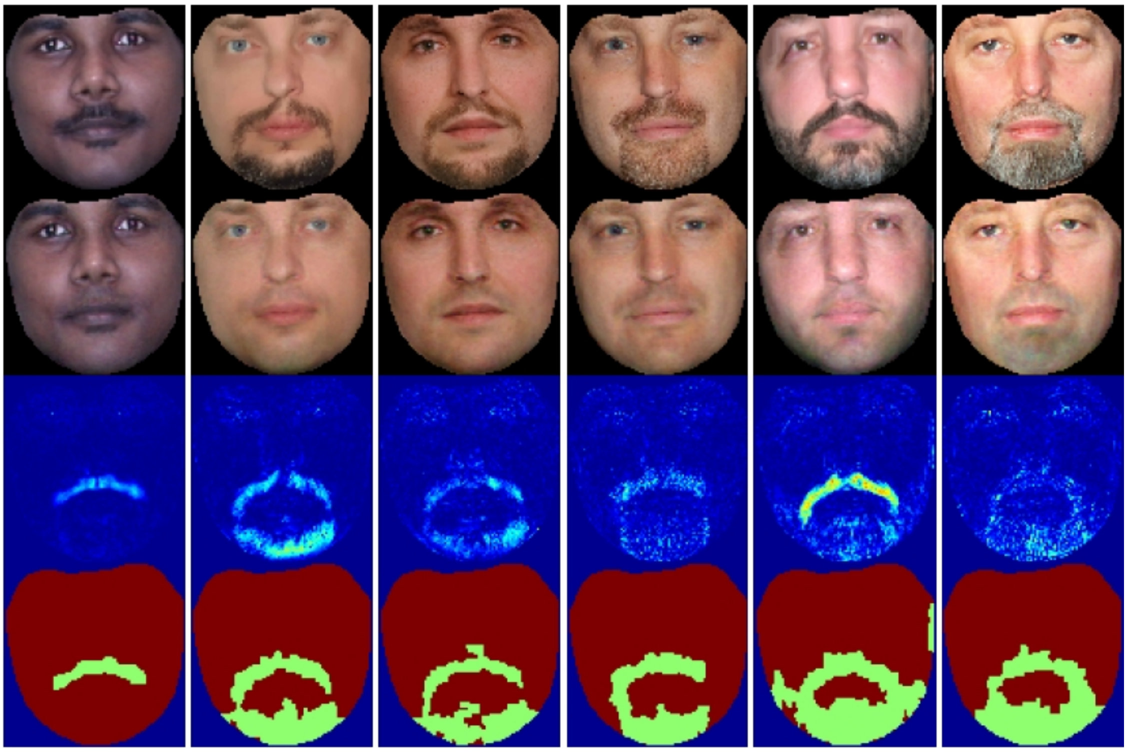

Figure 7: *Beard segmentation using Graph Cut. Row 1: original bearded faces; row 2: corresponding non-bearded faces; row 3: difference between bearded and non-bearded faces, brighter pixel means higher magnitude; row 4: resulting beard masks are shown in green.*

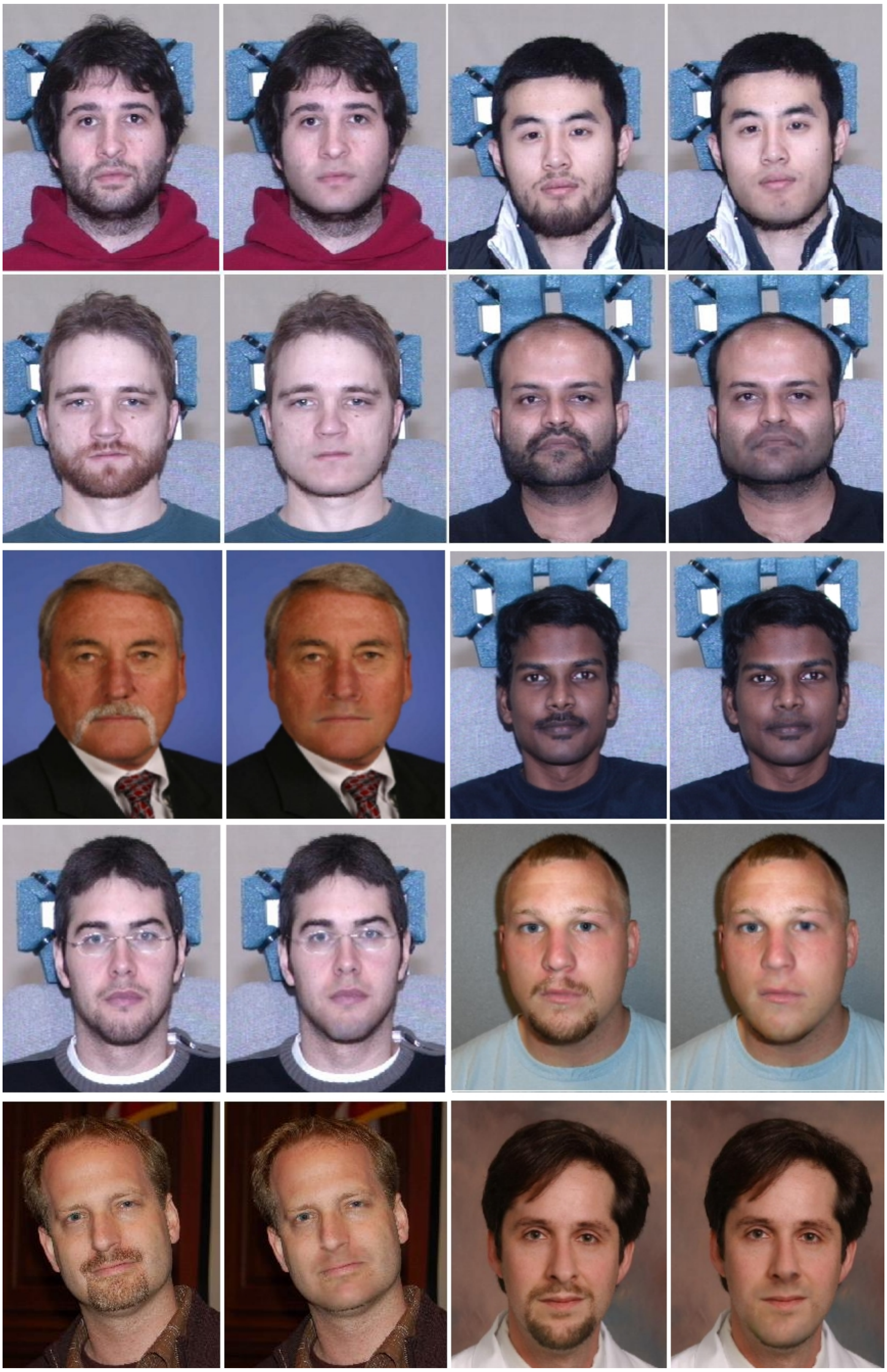

Figure 8: *Beard removal results after beard segmentation.*

cuts [BVZ01] (for binary partition problems, solutions produced by graph cuts are exact). Figure 7 shows the results of beard segmentation using this technique. The first row shows original bearded faces. The second row displays the corresponding non-bearded faces. The third row shows $|\mathbf{d}^B|$'s, the beard layers. The last row are the results of beard segmentation using the proposed method.

Once the beard regions are determined, we can refine the beard layer by zeroing out the entries of $\mathbf{d}^B$ that do not belong to the beard regions. Performing this refinement step together with unwarping the canonical non-bearded faces yield the resulting images shown in Figure 8.

5. Beard Addition

Another interesting question would be "How would I look with a beard?". This question is much more ambiguous than the question in Section 1. This is because there is only one non-bearded face corresponding to a bearded face (at least theoretically). On the contrary, there can be many bearded faces that correspond to one non-bearded face. A less ambiguous question would be to predict the appearance of a face with a beard from another person. Unfortunately, because of differences in skin color and lighting conditions, a simple overlay of the beard regions onto another face will result in noticeable seams. In order to generate a seamless composite, we propose to transfer the beard layer instead and perform an additional blending step to remove the sharp transition.

To remove the sharp transitions, we represent each pixel by the weighted sum of its corresponding foreground (beard) and background (face) values. The weight is proportional to the distance to the closest pixel outside the beard mask, determined by computing the distance transform. This technique is very fast and yields satisfying results such as the ones shown in Figure 9.

6. Database and Other Implementation Details

We use a set of images taken from CMU Multi-PIE database [GMC*07] and from the web. There are 1140 images of 336 unique subjects from the CMU Multi-PIE database and they are neutral, frontal faces. Among them, 319 images have some facial hairs while the others come from female subjects or carefully-shaved male subjects. Because the number of hairy faces from the Multi-PIE database is small, we downloaded 100 additional beard faces of 100 different subjects from the web. Thus we have 419 samples for the beard subspace in total and 738 samples for the non-beard subspace. We make sure no subject has images in both subspaces.

As in active appearance models [CET98], face alignment require a set of landmarks for each face. We use 68 hand-labeled landmarks around the face, eyes, eyebrows, nose and lips. Unlike the Layered Active Appearance Models [JS05], we do not have landmarks for the beard regions. Faces are aligned using triangular warping which is a warping method that groups landmarks into triangles and warps triangles to canonical ones. The size of canonical faces (for e.g. see Figure 2) is 94×92.

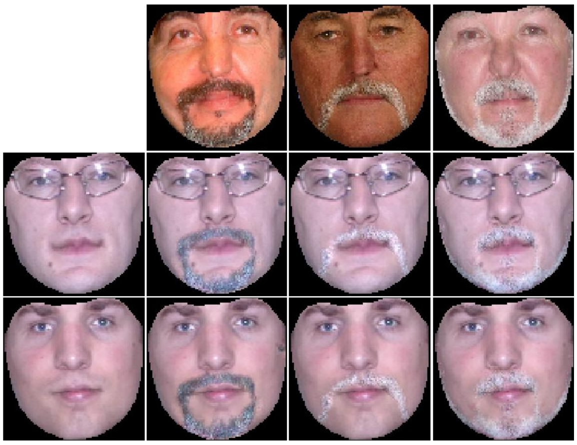

Figure 9: *Results of transferring beard layers. Beard layers of three images in the top row are extracted and transfered to the faces in the first column.*

In our implementation, robust subspace fitting is done independently for three color channels Red, Green, and Blue. We only combine the three channels when computing the beard masks. In particular, the combined $\mathbf{d}^B$ of three channels are taken as $(|\mathbf{d}^B_R| + |\mathbf{d}^B_G| + |\mathbf{d}^B_B|)/3$ where $\mathbf{d}^B_R, \mathbf{d}^B_G, \mathbf{d}^B_B$ are $\mathbf{d}^B$ of Red, Green, and Blue channels respectively.

7. Discussion and Future Work

In this paper, we have presented a method for automatic layer extraction and its applications to face synthesis and editing. Our method works by exploiting the differences between two subspaces. To the best of our knowledge, this is the first automatic method for constructing layers from sets of weakly labeled images. The results of our method are surprisingly good despite the limited amount of training data. It should also be noted that our method is generic and applicable not just for beards. Figure 10 shows some preliminary results on the removal of glasses.

We are currently working on several directions that might lead to further improvements. First, we believe our method can perform significantly better by simply enlarging the set of training images. Second, a recursive algorithm utilizing interactive user feedback would probably result in a better layer subspace. For example, obtaining the users' judgment on the robustly reconstructed non-bearded faces can help to filter out bad inputs for the creation of the beard layer. Obtaining such user feedback is definitely less tedious and time-consuming than manually labeling the layers.

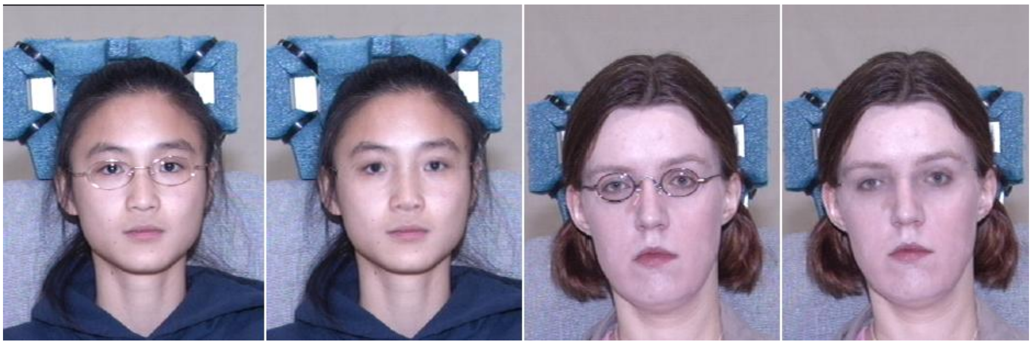

Figure 10: *Preliminary results for removal of glasses.*

References

[BJ98] BLACK M., JEPSON A.: Eigen-tracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision 36*, 2 (1998), 101–130.

[BSVS04] BLANZ V., SCHERBAUM K., VETTER T., SEIDEL H.: Exchanging faces in images. In *EUROGRAPHICS* (2004).

[BT74] BEATON A. E., TURKEY J. W.: The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics 16*, 2 (1974), 147–185.

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3D faces. In *SIGGRAPH* (1999).

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1222–1239.

[CET98] COOTES T., EDWARDS G., TAYLOR C.: Active appearance models. In *Proceedings of European Conference on Computer Vision* (1998), vol. 2, pp. 484–498.

[dlTB03a] DE LA TORRE F., BLACK M. J.: A framework for robust subspace learning. *International Journal of Computer Vision. 54* (2003), 117–142.

[dlTB03b] DE LA TORRE F., BLACK M. J.: Robust parameterized component analysis: theory and applications to 2D facial appearance models. *Computer Vision and Image Understanding 91* (2003), 53 – 71.

[GM87] GEMAN S., MCCLURE D.: Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute LII-4:5* (1987).

[GMC*07] GROSS R., MATTHEWS I., COHN J., KANADE T., BAKER S.: *The CMU Multi-pose, Illumination, and Expression (Multi-PIE) Face Database*. Tech. rep., Robotics Institute, Carnegie Mellon University, 2007. TR-07-08.

[HRRS86] HAMPEL F., RONCHETTI E., ROUSSEEUW P., STAHEL W.: *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York., 1986.

[Hub81] HUBER P. J.: *Robust Statistics*. New York ; Wiley, 1981.

[HW77] HOLLAND P. W., WELSCH R. E.: Robust regression using iteratively reweighted least-squares. *Communications in Statistics*, A6 (1977), 813–827.

[Jol02] JOLLIFFE I.: *Principal Component Analysis*, 2 ed. Springer-Verlag, New York, 2002.

[JS05] JONES E., SOATTO S.: Layered active appearance models. In *Proceedings of the IEEE International Conference on Computer Vision* (2005), vol. 2, pp. 1097–1102.

[Li85] LI G.: Robust regression. In *Exploring Data, Tables, Trends and Shapes* (1985), Hoaglin D. C., Mosteller F., Tukey J. W., (Eds.), John Wiley & Sons.

[MP97] MOGHADDAM B., PENTLAND A.: Probabilistic visual learning for object representation. *Pattern Analysis and Machine Intelligence 19*, 7 (July 1997), 137–143.

[Pig99] PIGHIN F.: *Modeling and Animating Realistic Faces from Images*. PhD thesis, University of Washington, 1999.

[PMS94] PENTLAND A., MOGHADDAM B., STARNER T.: View-based and modular eigenspaces for face recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (1994), pp. 84–91.

[PV07] PIERRARD J.-S., VETTER T.: Skin detail analysis for face recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[TP91] TURK M., PENTLAND A.: Eigenfaces for recognition. *Journal Cognitive Neuroscience 3*, 1 (1991), 71–86.

[VP97] VETTER T., POGGIO T.: Linear object classes and image synthesis from a single example image. *Pattern Analysis and Machine Intelligence 19*, 7 (1997), 733–741.

[WLS*04] WU C., LIU C., SHUM H., XY Y., ZHANG Z.: Automatic eyeglasses removal from face images. *Pattern Analysis and Machine Intelligence 26*, 3 (2004), 322–336.