Visio-lization: Generating Novel Facial Images

Umar Mohammed Simon J.D. Prince Jan Kautz

University College London

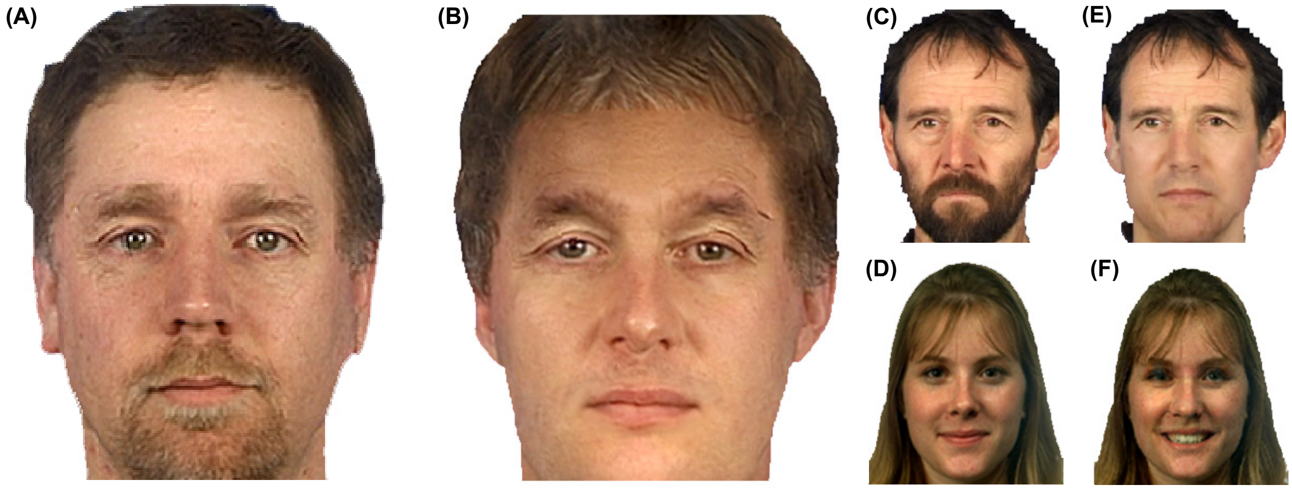

Figure 1: *We aim to learn a model of facial images (including hair, eyes, beards etc.) and use this to generate new samples (A and B). The results do not resemble any of the training faces, but are realistic and incorporate variation in sex, age, pose, illumination, hairstyle and other factors. We also describe methods to edit real faces (C and D) by inpainting large regions (E) or changing expression (F).*

Abstract

Our goal is to generate novel realistic images of faces using a model trained from real examples. This model consists of two components: First we consider face images as samples from a texture with spatially varying statistics and describe this texture with a local non-parametric model. Second, we learn a parametric global model of all of the pixel values. To generate realistic faces, we combine the strengths of both approaches and condition the local non-parametric model on the global parametric model. We demonstrate that with appropriate choice of local and global models it is possible to reliably generate new realistic face images that do not correspond to any individual in the training data. We extend the model to cope with considerable intra-class variation (pose and illumination). Finally, we apply our model to editing real facial images: we demonstrate image in-painting, interactive techniques for improving synthesized images and modifying facial expressions.

CR Categories: I.3.8 [Computing Methodologies]: Computer Graphics—Applications;

Keywords: face, texture synthesis, non-parametric sampling

1 Introduction

It has recently become possible to render almost any image at near photo-realistic quality. However, creating and editing realistic content remains time-consuming and requires considerable expertise. In particular, human faces pose a formidable challenge as they exhibit considerable variation due to identity, pose, lighting, hairstyle, expression, and other factors. Moreover, humans have extensive visual experience of faces and may be particularly sensitive to errors.

Nevertheless, synthesis of novel faces has many applications including creating criminal photofits or anonymizing faces in existing footage such as Google Street View. Such a technique would also be a step towards automatically creating realistic humanoid actors and avatars: considerable effort is currently expended in creating human characters for games and movies. Additional editing techniques would allow a degree of human control: e.g. we might require existing characters to change hairstyle or grow a mustache. Similarly editing real photos is useful for visualizing the likely results of plastic surgery, or improving portrait photos by changing expression, replacing blinking eyes or removing glasses.

In this paper we present an algorithm for synthesizing novel human faces including hair, eyes and beards (see Figure 1). Our method is statistical in that it learns a probabilistic model from a set of training faces. It generates new images that are both realistic (they obey the structural constraints of the face and have plausible texture) and novel (the identity is not the same as any of the training images). We demonstrate that our model can generate completely novel faces, add or remove facial features such as mustaches, fill in large obscured parts of a face, turn unrealistic renderings of face parts into realistic faces, and even change expression.

1.1 Related Work

There is a large body of work concerning modeling human faces. Linear models describing the pixel intensity across face images of multiple different individuals were first developed for face recognition [Turk and Pentland 1991] but have found application in graphics. Blanz and Vetter [1999] extended the linear approach to model both face texture and 3D shape and combined it with an explicit lighting scheme. They fitted this model to 2D face images and used it to relight, repose and morph the original image. The same model was also modified to allow users to edit face models to match internal mental images [Blanz et al. 2006]. A related multi-linear approach [Vlasic et al. 2005] was used to transfer video performances of one individual to animations of another. The above methods work well for fitting to real world images and modifying subsets of characteristics. However, they are limited in their ability to describe fine textures such as hair, eyebrows, and beards: the synthesized images are weighted sums of training images and these details tend to get averaged out. Moreover, linear models are not well suited to synthesizing completely new faces as they assign significant probability to implausible face configurations.

Liu et al. [2007] addressed some of these deficiencies by hallucinating high-frequency details that agreed with the prediction of the linear model. The results are superior to the linear model alone but are still not fully realistic (see Section 5). There have also been numerous linear and non-linear models for editing faces, but most are targeted at specific applications such as beard removal [Nguyen et al. 2008], removing blemishes [Brand and Pletscher 2008] or super-resolution [Dedeoglu et al. 2004; Liu et al. 2005].

Other work has modelled the faces of particular individuals. Such models have been used to create new videos of the same person mouthing words that they did not originally speak [Bregler et al. 1997; Ezzat et al. 2002]. These models produce very realistic results but do not describe the between-individual variation and hence cannot be used for our purpose. Similarly Weyrich et al. [2006] built near-photorealistic face models by measuring the geometry, reflectance and subsurface scattering of each individual face. They show how facial detail (e.g. freckles) can be learnt and transferred between individuals, but cannot model larger structures in this way. In conclusion, no existing face model is suited to our goal.

However, two recent strands of work in image synthesis provide inspiration. The first is non-parametric *texture synthesis* [Efros and Leung 1999; Wei and Levoy 2000; Efros and Freeman 2001; Kwatra et al. 2003]. Given a small sample of texture the goal is to generate a larger output texture. These methods synthesize novel textures by pasting pixels, patches or regions from the original sample into the new image such that they are in local agreement. Unfortunately, these methods were designed for stochastic textures with stationary statistics and only have knowledge of the local Markov structure. They can have no notion, for example, that a facial image must contain a plausible configuration of eyes, nose and mouth. Secondly, and at the opposite end of the spectrum are *photo synthesis* methods. These insert entire visual objects into the image at once for the purpose of inpainting [Hays and Efros 2007; Diakopoulos et al. 2004] or augmenting existing images [Lalonde et al. 2007]. Such methods are suited to replacing an entire face in an image [Bitouk et al. 2008], but are unsuited to generating *novel* faces.

In this paper we propose a system for generating face images which we term “visio-lization” after the Latin *visio* for face. It lies between the extremes of texture- and image-synthesis. In common with both, we create new images by copying parts of existing images. As in texture synthesis, we build new images by considering only small regions at any one time, which allows us to induce randomness. However, we do so using a model that is non-stationary, and has a notion of the global form of the face. In Section 2 we describe a non-stationary, non-parametric method for generating faces with local consistency. In Section 3 we describe a method for generating faces that have the correct global structure but poor local texture. In Section 4 we combine the local and global models and synthesize realistic faces. The rest of the paper explores extensions and applications of this technique.

2 Local Non-Parametric Model

2.1 Image Quilting

We first review the ‘image quilting’ method of Efros and Freeman [2001]. Image quilting synthesizes a new texture given an input texture sample. The first step is to extract all possible patches of a given size from the input texture to form a ‘patch library’. The synthesized image will consist of a regular grid of these library patches such that each overlaps its neighbors by a few pixels. A new texture is synthesized starting in the top-left of this grid and proceeding to the bottom-right. At each position, a library patch is chosen such that it is visually consistent with the patches that have previously been placed above and to the left. The new patches can then be blended together using a variety of techniques.

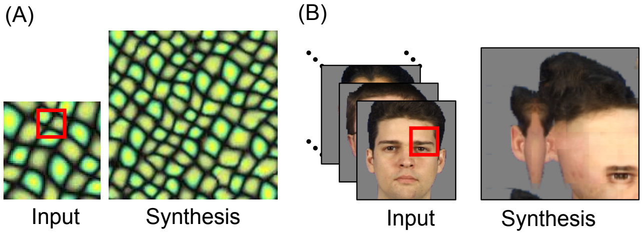

Figure 2: *(A) The image quilting algorithm applied to a texture. A library of overlapping ‘blocks’ is built from the input texture. A new texture is synthesized from top-left to bottom-right by copying patches from the input texture such that they match in the overlapping regions. (B) Image Quilting applied to face images. The input texture is a set of face images. The output does not resemble a face.*

An example of an input texture and the synthesized result using ‘image quilting’ is shown in Figure 2A. What happens if we directly apply this method to face images? In Figure 2B we show the results of building a patch library from a set of weakly registered frontal face images from the XM2VTS database [Messer et al. 1999] and synthesizing a new image. The result contains some facial parts but fails to capture the overall structure of the face. This is unsurprising since the image quilting technique is designed for *stationary* textures. However, the statistics of frontal faces are clearly not stationary. The joint distribution of nearby pixel values depends on the position in the image: the top of the image always contains hair, the center contains the nose and so on.

2.2 Non-Stationary Image Quilting

We adapt the image quilting method to take account of the non-stationary statistics of faces. We divide the training images into the same regular grid of overlapping patches as the output image. We now extract a separate library of patches at each location (see Figure 3A). Once more we synthesize a new image from top-left to bottom right choosing at every position a patch that ensures visual consistency with existing neighbors (Figure 3B). However, now each patch is taken from the appropriate library for that position, giving the resulting images the desired non-stationary statistics.

We implement this model by hand-marking 12-68 points on each of 2000 library faces and affine warp to a standard template shape. Each face image is divided into a regular grid of 9 × 9 overlapping RGB patches where the overlapping region is one quarter of the patch size. At each image location we build a library of 24000 patches by choosing patches from the appropriate region of the library images under a variety of small 2D rotations and translations.

In synthesis, we choose the first patch (in the top-left of the image) randomly. For subsequent patches, we find the N patches that are most visually consistent in the overlap region. Visual consistency was quantified using the sum of the square differences in the overlap region across the three color channels. We randomly select one of these N patches. This randomness prevents the algorithm from exactly recreating one of the images in the library. For all experiments in this paper N was of the order of 100 patches. Having chosen all the patches we blend them together seamlessly using a gradient domain method which will be discussed in Section 4.

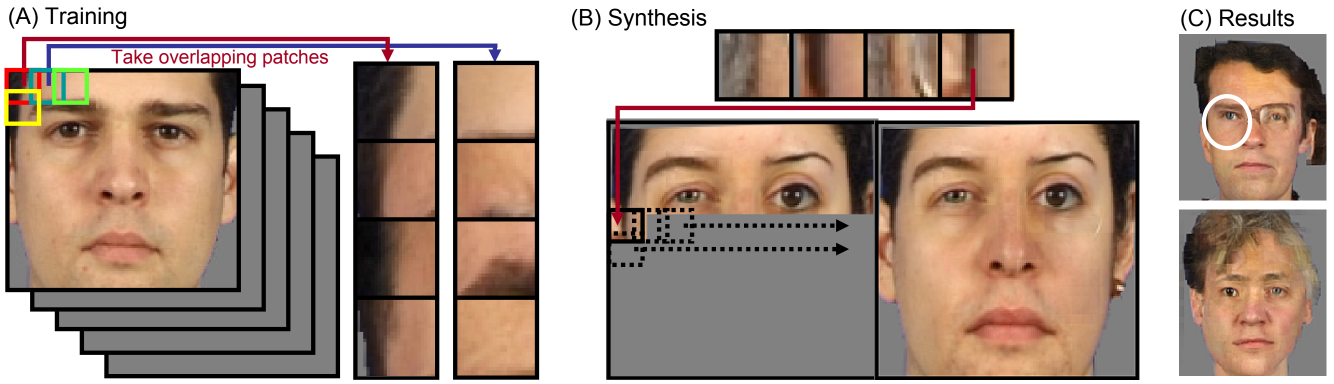

Figure 3: *Local non parametric model for face generation. (A) Learning the model involves building a separate patch library for each position in the image. (B) Synthesis proceeds from top-left to bottom-right. At each step, we search the library associated with the current position for a patch that is visually consistent with the previously synthesized patches above and to the left. (C) Results from this model are locally consistent: viewed through any small window, the image looks correct. However, they lack global consistency.*

Images synthesized from this model are shown in Figure 3C. Although they are an improvement over image quilting, they remain unrealistic. Within any small window, the image looks correct. However, the model only considers the local Markov structure of the image and contains nothing to enforce global constraints on the face. This deficit leads to "Frankenstein" images where the characteristics of the face (e.g. race, gender, hair color) change gradually across the image. To solve this problem we must ensure that the later patches are consistent with the previously pasted ones even when they are not adjacent.

3 Global Parametric Model

To resolve the remaining problems with our non-stationary image quilting technique, we consider a second parametric model that has complementary properties. Simple linear subspace models of faces (e.g. [Turk and Pentland 1991]) explicitly model the covariance of all of the pixels and hence have a good understanding of the global structure of the image. However, they are poor at modeling local textures. In this section we describe a linear subspace model which we refer to as a global parametric model. In Section 4 we show how to combine this with our non-parametric image quilting method.

The global parametric model describes face data using a factor analyzer [Bishop 2006]. This model is similar to principal component analysis, but is fully probabilistic. The vectorized pixel data $\mathbf{x}_i$ from the i'th training image is modeled as:

$$\mathbf{x}_i = \mu + \mathbf{F}\mathbf{h}_i + \varepsilon_i. \quad (1)$$

Each face $\mathbf{x}_i$ is assumed to consist of an additive mixture of (i) a mean μ, (ii) a per pixel noise component ε_i with mean zero and diagonal covariance Σ, and (iii) a weighted linear combination of basis faces or factors. These weights are held in the factor loading vector $\mathbf{h}$. The factors themselves occupy the columns of the factor matrix $\mathbf{F}$. The factor analyzer can alternately be written as:

$$\begin{aligned} Pr(\mathbf{x}_i|\mathbf{h}_i) &= \mathcal{G}_{\mathbf{x}_i}[\mu + \mathbf{F}\mathbf{h}, \Sigma] \quad (2)\\ Pr(\mathbf{h}) &= \mathcal{G}_{\mathbf{h}}[\mathbf{0}, \mathbf{I}] \quad (3) \end{aligned}$$

where $\mathcal{G}[\mathbf{a}, \mathbf{B}]$ denotes a Gaussian distribution with mean $\mathbf{a}$ and covariance $\mathbf{B}$. Note that the zero mean, identity covariance prior over the factor loadings $\mathbf{h}$ resolves the ambiguity over the scale of $\mathbf{F}$. We learn the parameters of the model $\theta = \{\mathbf{F}, \Sigma, \mu\}$ using 40 iterations of the expectation-maximization (EM) algorithm [Dempster et al. 1977]. We trained the factor analysis model with 1500 70 × 70 face images using 8 factors. To generate a new face we:

- randomly sample factor loadings $\mathbf{h}$ from the prior,
- weight the factor images by these loadings and sum,
- add the mean face component, μ.
- Note that we do *not* add the stochastic noise component ε_i.

Example generated images are shown in Figure 4. They are globally coherent (look like a single individual), but are blurry and fail to reproduce realistic local texture. With more factors the blurriness is reduced, but at the cost of introducing high frequency artifacts.

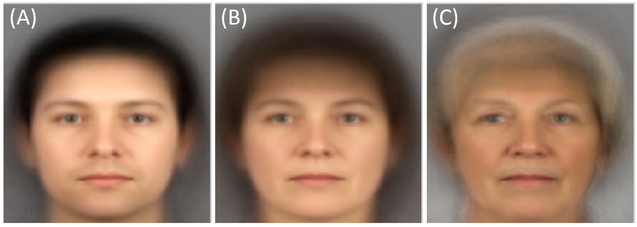

Figure 4: *Example results from the global model. They resemble faces, but are blurry and contain significant artifacts. However, unlike the model presented in Section 2 they are globally consistent. The identity of the face does not vary with position.*

4 Combining Local and Global Models

In Sections 2 and 3 we presented two models for faces with complementary properties. In this section we exploit the best points of both to generate more realistic images. We first generate an image from the global parametric model. This creates a blurry image of the type found in Figure 4. We then synthesize an image using the local non-parametric model that is consistent with this target. In probabilistic terms, we condition the local model on the result of the global parametric model. This is similar to the texture transfer approach of Efros and Freeman [2001] and the method of Ashikhmin [2001], who both conditioned texture synthesis on an underlying image. However, here texture synthesis is non-stationary and the conditioning image was stochastically generated.

In practice this conditioning is implemented as follows. As before, patches are chosen such that they are visually consistent (in terms of squared difference) with the patches above and to the left. However, we also require visual consistency with the results of the global synthesis. Patch choice is now determined using a weighted sum of these two constraints. As before, we randomly choose from the N best matching patches. For frontal faces, we enforce symmetry by constraining the choice of patches horizontally opposite each other to come from the same individual. This prevents small

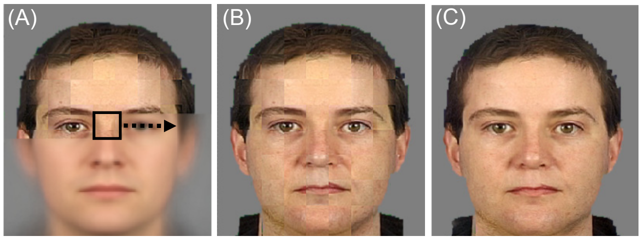

Figure 5: *Combining local and global models. (A) As previously we synthesize images from top-left to bottom right by choosing patches that are visually consistent with those above and to the left. We now also ensure that the patches are consistent with the target global image (visualized here as being underneath the synthesized image). (B) This ensures that we get a globally consistent result. (C) After blending together patches in the gradient domain.*

but noticeable asymmetries, particularly in the color and size of the eyes. Figure 5A-B illustrates this process.

We post-process the results in two ways: first we use Poisson image editing [Perez et al. 2003] to remove artifacts due to slight differences in skin tone between patches. As we synthesize the image, we store the indices of the patches used. We then create x- and y-gradient domain images by assembling together the gradients of the chosen patches. In overlap regions, we average the gradient images. We solve a Poisson equation to find an image that has gradients as close to the synthesized gradients as possible and that exactly obey the boundary conditions at the edge of the image (determined by the output of the global parametric model). The results are shown in Figure 5C. Finally, we un-warp the image using the inverse of a randomly chosen transformation from the training data. In principle more complex transformation families could be considered and this geometric warp could be jointly modeled with the global intensity model, but in practice we find this unnecessary.

The methods described above are sufficient to synthesize frontal images. If we train the global and local models with libraries of profile faces, we can similarly generate novel faces in profile. However, this method cannot synthesize both frontal and profile faces simultaneously (see Figure 6B-E). The global target face often contains a linear combination of frontal and non-frontal images and the synthesized face is correspondingly unrealistic.

Multimodal datasets of this sort fail because the global factor analysis model assumes that the data is unimodal: consequently, if both frontal and profile faces are assigned high probability it is inevitable that mixtures of the two will also be likely (see Figure 6A). To resolve this problem, we use a multi-modal global model: we learn a mixture of factor analyzers (MoFA) model using the EM algorithm (see Ghahramani and Hinton [1997]). This model performs an unsupervised clustering of the data into K linear subspace models with cluster weights π_k and parameters $\{\mu_{1...K}, \mathbf{F}_{1...K}, \Sigma_{1...K}\}$. Generation from the MoFA model proceeds as follows:

- choose 1 of K factor analyzers from discrete distribution $\{\pi_1 \ldots \pi_K\}$,
- choose factor loadings $\mathbf{h}$ from a normal distribution with zero mean and identity covariance,
- weight the k'th factor images $\mathbf{F}_k$ by these loadings and sum,
- add a k'th mean face component μ_k.

To demonstrate this idea we learnt a mixture of two factor analyzers each containing 8 factors using a database consisting of both frontal and profile faces. Results from this global model are shown in Figure 6G-H. To synthesize realistic images, we constrain the pasted patches to come from training images that were primarily associated with the simulated cluster. We now have a single model that synthesizes both profile and frontal faces (Figure 6I-J). This clustering approach also improves results when frontal faces alone are used and was used to generate the results in Figures 1 and 7.

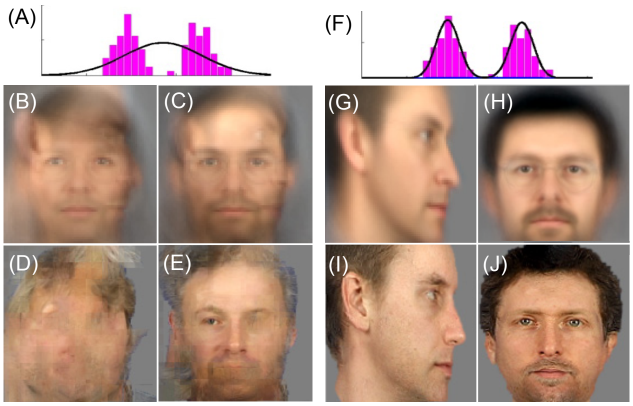

Figure 6: *Synthesis of both frontal and profile faces. The original global model describes both clusters with a single Gaussian (A). Draws from this distribution (B-C) often lie between the two clusters and result in poor synthesis results (D-E). Using a more suitable clustered model (F) results in sensible global models (G-H) and synthesized images (I-J).*

4.1 Efficient Implementation

Our implementation depends on pre-computation and storage of the sum of squared difference between all possible combinations of patches in each overlapping region. For every possible patch at position (x,y) we store a file containing the indices of the 5000 patches at (x+1,y) that agree most closely and the associated overlap errors. When we synthesize a new patch at position (x+1,y) we simply read the file associated with the particular patch to the left. By preparing a second set of files describing the vertical relations between patches we can also load in a file associated with the synthesized patch above at position (x+1,y-1) We intersect these two lists to find possible candidate patches for the current position.

We compare these candidate patches to the global model. This comparison can be made efficient by (i) pre-computing the distance d_1 from the patch to the subspace defined by the global model and (ii) pre-computing the position of each patch within this low-dimensional subspace. At run-time we calculate the squared distance d_2 between the global image and the candidate patches within this subspace. We calculate the total error between the global model and the subspace using Pythagoras' theorem.

This pre-computation takes several weeks (single CPU), but the result is that we can synthesize new images in 1-2 seconds. When we only replace part of an image (see Section 6), it is even faster.

5 Results

In Figure 7 we show several faces generated using our method. The system can create a wide variety of different looking faces which vary in age, gender, hairstyle and other factors. We can also generate images under different lighting conditions or with different poses by training with the appropriate library. We also demonstrate common failure modes: occasionally small errors occur where the chosen patch does not closely agree with its predecessor. For example, the right ear in the top row of Figure 7F is flawed, as is the chin on the middle panel. It is possible to remove such problems by inpainting as shown in Section 6. We also occasionally generate unrealistic and blurry hairstyles as in the lower panel.

These problems aside, it is usually possible to synthesize realistic face images with this method. Informal experiments with a 2 second presentation suggest that human observers find it hard to discriminate our best synthesized examples from real faces. However, realism alone is not a sufficient evaluation criterion: the synthesized faces also need to be novel. Figure 8A-C shows three examples of

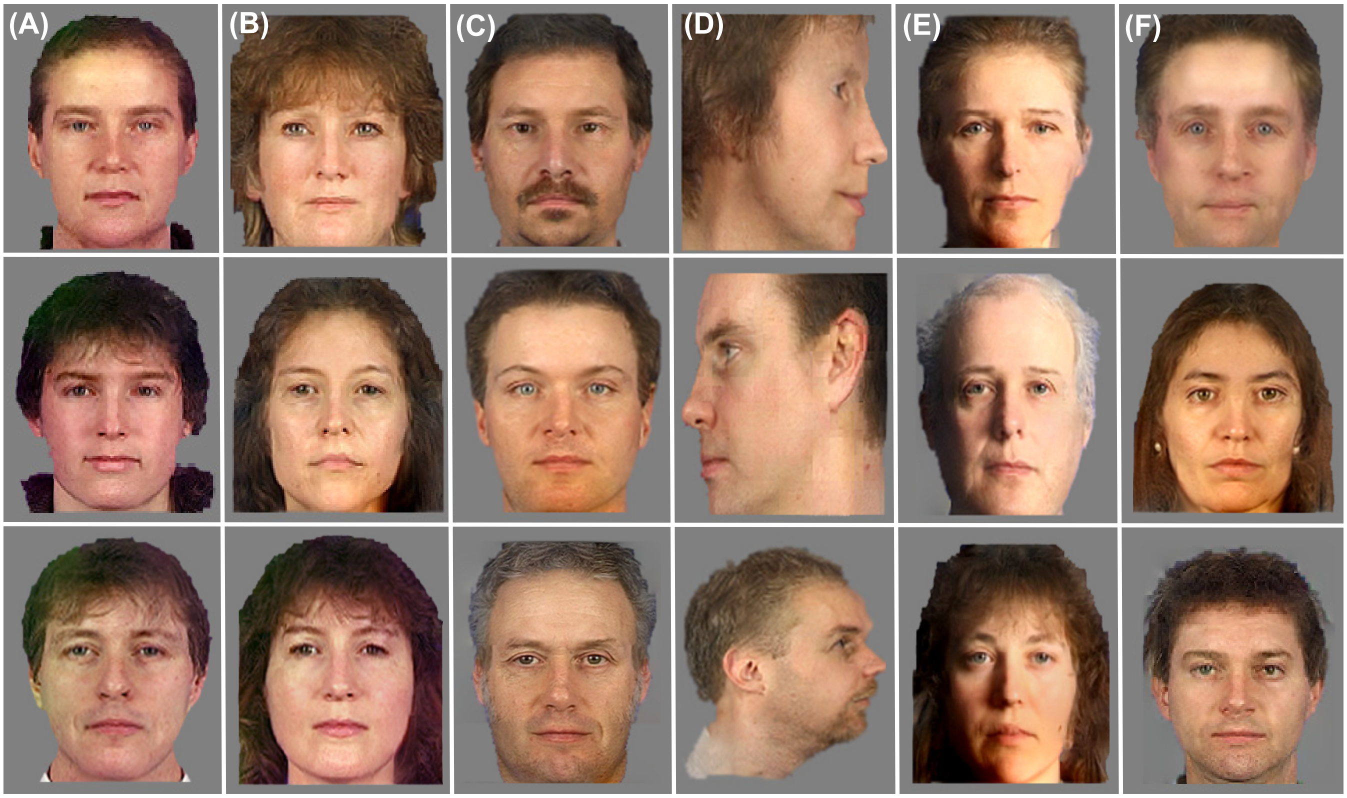

Figure 7: *Results. (A) Synthesized men (B) Women. (C) Men are sometimes generated with facial hair and sideburns. (D) Profile images (E) Side-lit images (F) Failure modes: Patch errors on right ear (top) and chin (middle). Unrealistic and blurry hair (bottom).*

synthesized images. Further insight is provided by Figure 8D-F. Patches are colored by the identity of the training individual they were sampled from. All three synthesized faces are genuine hybrids comprising parts of many different people. Note that even the individual facial features (mouth, eyes etc.) are not necessarily copied in their entirety from a single individual, but are synthesized piecemeal from several different people. We have also investigated the closest training face (in the least squares sense) to the generated faces. These do not resemble the generated face (Figure 8G-I). We conclude that we are successfully generating novel images.

In Figure 9 we show that our results compare favorably to previous attempts to generate random faces. The results of Liu et al. [2007] are relatively blurry and do not produce realistic hair textures. Their method also induces randomness using a parametric global model. It differs from ours in that they hallucinate high-frequency detail having learnt the relation between low- and high-frequency image patches in a training set. The final result is a weighted combination of the low resolution global model and the confabulated high-frequency information. We note however that this model was intended primarily for super-resolution, where we would not expect our model to perform well. We also show randomly generated results from FaceGen (www.facegen.com). Although their results are not fully realistic, it should be noted that these are 2D projections of 3D models so they are more flexible. We have not shown a comparison with Blanz and Vetter [1999] as their model does not have an explicit method for randomly generating examples, although our results would also compare favorably.

6 Editing Real Images

Until now this paper has been concerned with generating novel faces. The rest of the paper concerns the application of the same model to editing existing faces. These might be real photos or facial images synthesized using the described method.

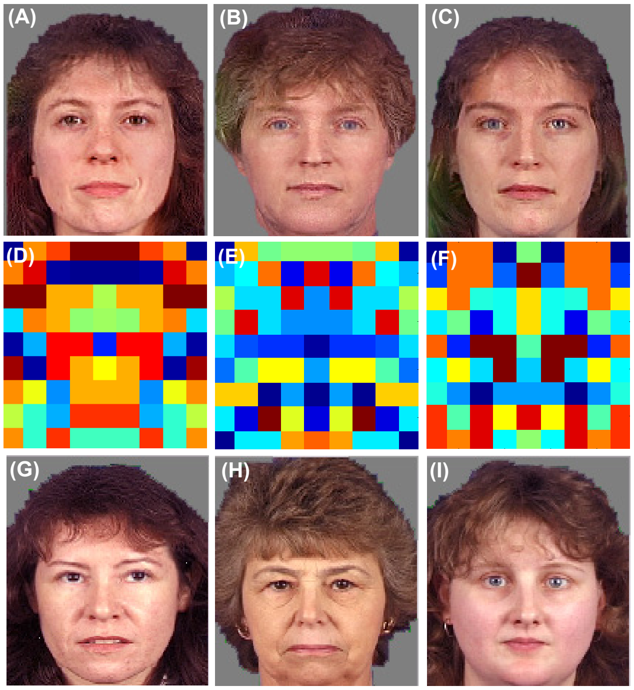

Figure 8: *Analysis of synthesized faces. (A-C) Three examples (D-F) Colors indicate origin of synthesized patch. Generated faces are hybrids of many individuals. (G-I) Closest training face.*

6.1 Image Inpainting

In this section we describe an approach to inpainting of faces. Here part of the face is missing and we wish to complete the image in

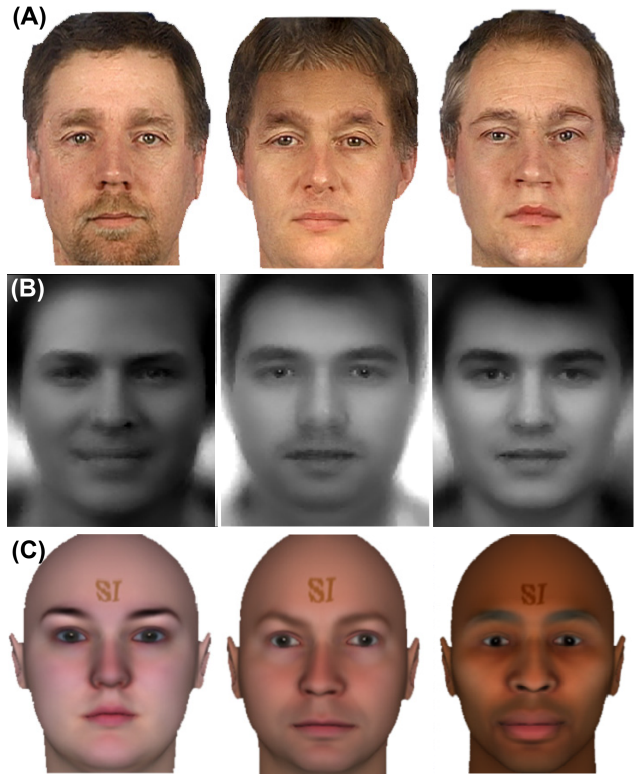

Figure 9: *Comparison with other methods. (A) Our results. (B) Results of Liu et al. [2007]. (C) 2D projections of random 3D models generated by FaceGen.*

such a way that it looks natural. Our approach (Figure 10A-D) is as follows: first we generate a 'global target image' from the parametric factor analysis model that is compatible with the observed part of the face. Then we synthesize texture over the missing region that agrees with this observed image and matches at the boundaries.

We denote the observed part of the image vector $\mathbf{x}$ by $\mathbf{x}_o$ and the missing part by $\mathbf{x}_m$. To generate the global target image we:

- estimate factor loadings $\mathbf{h}^*$ that best describe the observed part of the image $\mathbf{x}_o$,
- weight the factor images $\mathbf{F}$ by these loadings and sum,
- add the mean face component, μ,
- extract missing dimensions $\mathbf{x}_m$ from the resulting global image.

In order to find the factor loadings $\mathbf{h}^*$ that best describe an image we apply Bayes' rule:

$$Pr(\mathbf{h}|\mathbf{x}) = \frac{Pr(\mathbf{x}|\mathbf{h})Pr(\mathbf{h})}{\int Pr(\mathbf{x}|\mathbf{h})Pr(\mathbf{h})dh} \tag{4}$$

where the terms in the numerator of the right hand side were defined in Equations 2 and 3. The posterior probability $Pr(\mathbf{h}|\mathbf{x})$ can be calculated in closed form and is Gaussian with moments:

$$E[\mathbf{h}] = (\mathbf{F}^T\Sigma^{-1}\mathbf{F})^{-1}\mathbf{F}^T\Sigma^{-1}(\mathbf{x}-\mu) \tag{5}$$

$$\mathrm{Cov}[\mathbf{h}] = (\mathbf{F}^T\Sigma^{-1}\mathbf{F})^{-1} \tag{6}$$

See Bishop [2006] for details of this calculation. When part of the image is missing we substitute only the observed dimensions of $\mathbf{F}$, Σ, μ and $\mathbf{x}$ into this calculation. The most likely factor loadings $\mathbf{h}^*$ are at the mean of the posterior (Equation 5).

Four more examples of the inpainting procedure are shown in Figures 10(E-H). We can also generate multiple hypotheses, by drawing several possible values of the factor loadings $\mathbf{h}^*$ from the posterior distribution defined by Equations 5 and 6. This produces several possible target global models, each of which induces a different final result. An example of this can be seen in Figure 10I. This could be applied to helping create photofits of criminals.

This technique is related to that of Agarwala et al. [2004], who created hybrid faces by specifying both source and destination regions for pixel copying. Our method requires only that we specify an area to be replaced. It is also related to the face swapping technique of Bitouk et al. [2008]. However, our system synthesizes novel content rather than verbatim copying from a library face.

6.2 Interactivity

We can exploit the speed of our system to allow interactive techniques. One possible use of this is to use the inpainting method described above to repair patch errors in synthesized faces such as those found in Figure 7F. For example, the synthesized face in Figure 11A has a flaw on the cheek. We can select this region, and generate several new versions of the region by inpainting. We then choose one that is visually pleasing (Figure 11B). Note that none of the other images in this paper have been manipulated in this way: they were all generated without user interaction.

We have also investigated making manual edits of images and using the result as a target for our non-parametric model as shown in Figure 11(C-E). We add a mustache to a real face using a standard paint program. We then use the modified region to guide a non-parametric texture synthesis. The result is a realistic mustache.

6.3 Changing Facial Characteristics

Finally, we investigate editing larger scale characteristics of faces such as expression. This requires a global model that separates the content of the face (the identity) from the style (smiling or not smiling). We employ an asymmetric bilinear model [Tenenbaum and Freeman 2000] which describes the generative process as:

$$\mathbf{x}_{ij} = \mu_j + \mathbf{F}_j\mathbf{h}_i + \varepsilon_{ij}, \tag{7}$$

where $\mathbf{x}_{ij}$ represents the i'th face in the j'th style. The factor loadings $\mathbf{h}_i$ are constant for an individual. The basis functions $\mathbf{F}$, mean μ and noise Σ vary depending on whether the style is normal (j=1) or smiling (j=2). We train this model from images of 700 individuals, each of which is seen in both style conditions. The parameters of this model $\theta = \{\mu_1, \mu_2, \mathbf{F}_1, \mathbf{F}_2, \Sigma_1, \Sigma_2\}$ were learnt using the method described in Prince et al. [2008].

To generate a global image in style 2, given a style 1 face $\mathbf{x}_1$, we:

- estimate factor loadings $\mathbf{h}^*$ that best describe the image $\mathbf{x}_1$,
- weight the factor images $\mathbf{F}_2$ by these loadings and sum,
- add the mean face component, μ_2.

We then use this to guide the non-parametric model which now pastes down only patches from images seen in style 2. As in Section 6, the factor loadings can be calculated via Bayes' rule. The most likely loadings are the mean of the posterior distribution:

$$h^* = (\mathbf{F}_1^T\Sigma_1^{-1}\mathbf{F}_1)^{-1}\mathbf{F}_1^T\Sigma_1^{-1}(\mathbf{x}_1 - \mu_1) \tag{8}$$

Figure 12A shows a section of an original face in style 1. This is used to calculate factor loadings $\mathbf{h}^*$. Figure 12B and C show the predictions of these factor loadings in style 1 and 2 respectively. Figure 12D shows the effect of applying non-parametric texture synthesis. Further results are given in panels E-H.

7 Discussion and Conclusions

We have presented methods for synthesizing random realistic face images. Our method generates an approximate target image which is globally coherent and then synthesizes texture over the top with a non-stationary image quilting model. We have shown that the same

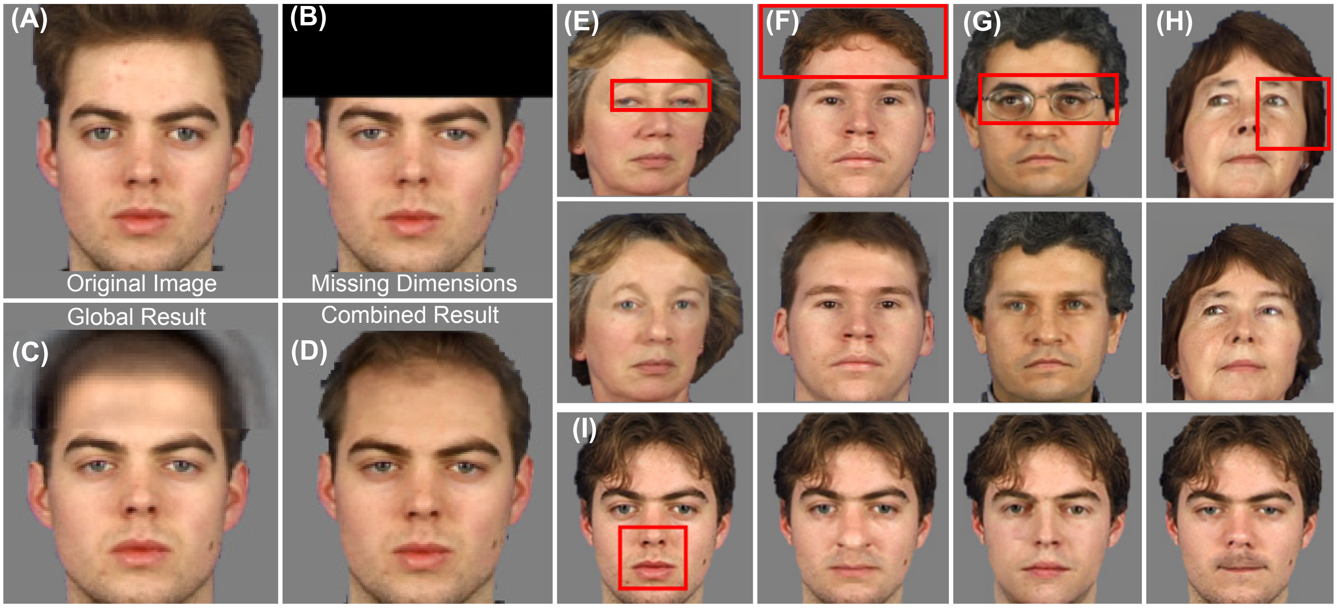

Figure 10: *Inpainting faces. We take an original face (A) and remove the top part (B). We fill the missing part with the most likely pixel values based on conditioning the factor model on the remaining observed image (C). We then synthesize new texture over the top to replace the missing region (D). (E-H) More examples. (I) Generating multiple hypotheses.*

model can be used to edit and inpaint existing faces. Our method is simple and works well: the synthesized faces look very realistic.

In this paper, we have generated frontal and profile faces. However, our method can also synthesize faces at intermediate poses. It is still necessary to preserve facial symmetry, but there is no longer a one-to-one mapping between patches in the left- and right-hand sides of the face. Fortunately, it is sufficient to define a subset of patches from the left and right sides that are linked. The parts of the face that they describe should overlap but need not correspond exactly: symmetry is effectively propagated to the intervening patches as they are forced to agree with their constrained neighbours.

Our results are not entirely without flaws. Possible improvements include (i) using larger databases of faces (ii) creating more patches with small affine transformations of the RGB colors so that we are more likely to find appropriate matches (iii) employing more sophisticated texture synthesis methods such as the graph cut textures method of Kwatra et al. [2003]. This would however, result in a drastic decrease of speed.

The method as it stands has some limitations: we cannot synthesize Asian/African faces or faces with glasses as there are too few examples in our database. We are also limited to poses and lighting conditions found in our training databases. However, our methods could be extended by (i) using a much larger set of faces and (ii) adapting the methods of [Bitouk et al. 2008] for filtering candidate patches and recoloring and relighting.

Our technique is closely related to work in super-resolution of faces. For example [Dedeoglu et al. 2004] and [Liu et al. 2005] also used patch-based representations to hallucinate realistic faces from low resolution images. However, these methods were not designed for synthesizing novel faces or editing real high-resolution images. Consequently they do not have mechanisms to induce randomness, encourage global coherence, or predict missing regions of the face.

This work opens several new avenues of research. One possibility is to synthesize multiple images that are perceived to have the same identity. Solving this problem is related to face recognition and would allow us to synthesize videos of faces. We might also aim to synthesize more complex object classes. Faces are relatively easy in that the constituent parts (eyes, nose etc.) are always present. This is not the case for houses or chairs for instance.

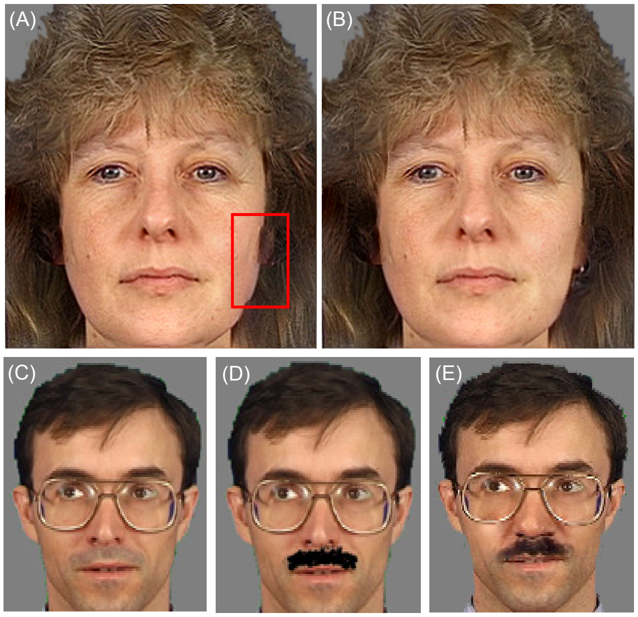

Figure 11: *The synthesized face in (A) has a flaw on the cheek. We can manually select this region and inpaint to generate a flawless example (B). We edit the real face (C) in a paint program by drawing on a mustache (D). We use the edit as the global target image to give a more realistic result (E).*

Acknowledgements

We would like to thank our reviewers for suggestions and comments, and Andrew Fitzgibbon, Gabriel Brostow, Tim Weyrich and Marcus Magnor for discussions and reading an early draft of the paper. We thank Ce Liu for allowing us to use his images. We are grateful to the members of the Vision and Imaging Science group as well as the Virtual Environments and Computer Graphics group at UCL who helped labeling images. This work was supported by a Microsoft Research PhD scholarship.

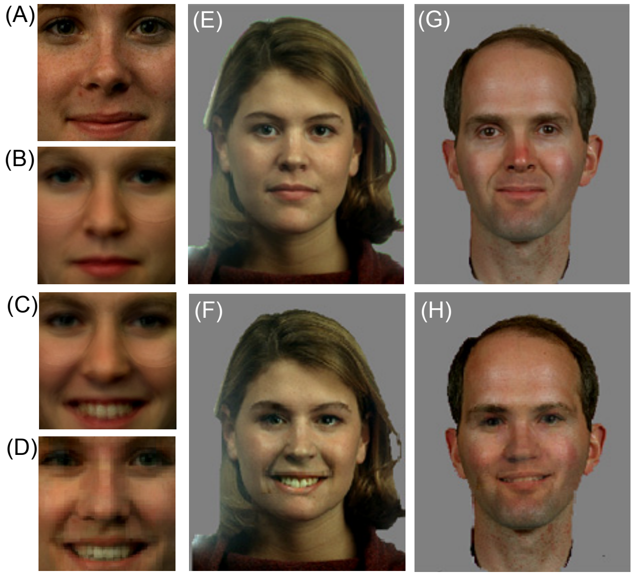

Figure 12: *Changing expression. (A) original region (B) approximation of original with bilinear model (C) prediction of smiling face from bilinear model (D) after synthesis. (E-H) Two real faces adapted in which the expression has been changed.*

References

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. 2004. Interactive digital photomontage. *ACM Transactions on Graphics (Proc. SIGGRAPH) 23*, 3, 294–302.

Ashikhmin, M. 2001. Synthesizing natural textures. *In Proc. ACM Symposium on Interactive 3D Graphics*, 217–226.

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. Springer.

Bitouk, D., Kumar, N., Dhillon, S., Belhumeur, P., and Nayar, S. K. 2008. Face swapping: automatically replacing faces in photographs. *ACM Trans. Graph. 27*, 3, 1–8.

Blanz, V., and Vetter, T. 1999. A morphable model for the synthesis of 3d faces. In *Proceedings of ACM SIGGRAPH 99*, 187–194.

Blanz, V., Albrecht, I., Haber, J., and Seidel, H.-P. 2006. Creating face models from vague mental images. *Computer Graphics Forum 25*, 3, 645–654.

Brand, M., and Pletscher, P. 2008. A conditional random field for photo editing. In *Proceedings of CVPR*, 187–194.

Bregler, C., Covell, M., and Slaney, M. 1997. Video rewrite: Driving visual speech with audio. In *Proceedings of ACM SIGGRAPH 97*, 353–360.

Dedeoglu, G., Kanade, T., and August, J. 2004. High-zoom video hallucination by exploiting spatio-temporal regularities. In *Proceedings of CVPR*, 151–158.

Dempster, A. P., Laird, N. M., and Rubin, D. B. 1977. Maximum likelihood for incomplete data via the EM algorithm. *Journal of the Royal Statistical Society 39 (B)*, 1, 1–38.

Diakopoulos, N., Essa, I., and Jain, R. 2004. Content based image synthesis. In *CIVR 04*, 299–307.

Efros, A. A., and Freeman, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of ACM SIGGRAPH*, 341–346.

Efros, A. A., and Leung, T. K. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of ICCV*, vol. 2, 1033–1038.

Ezzat, T., Geiger, G., and Poggio, T. 2002. Trainable video-realistic speech animation. In *Proceedings of ACM SIGGRAPH 2002*, 388–398.

Ghahramani, Z., and Hinton, G. E. 1997. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Dept. of Computer Science, University of Toronto, Canada.

Hays, J., and Efros, A. A. 2007. Scene completion using millions of photographs. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3, 4:1–4:7.

Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A. 2003. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics 22*, 3, 277–286.

Lalonde, J., Hoiem, D., Efros, A. A., Rother, C., Winn, J., and Criminisi, A. 2007. Photo clip art. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3, 3:1–3:10.

Liu, W., Lin, D., and Tang, X. 2005. Hallucinating faces: Tensorpatch super-resolution and coupled residue compensation. In *Proceedings of CVPR*, 478–484.

Liu, C., Shum, H., and Freeman, W. 2007. Face hallucination: theory and practice. *International Journal of Computer Vision 75*, 1, 115–134.

Messer, K., Matas, J., Kittler, J., Luettin, J., and Maitre, G. 1999. XM2VTSbd: The extended MTVTS database. In *Proceedings 2nd Conference on Audio and Videobase Biometric Personal Verification (AVBPA99)*, 72–77.

Nguyen, M., Lalonde, J., Efros, A., and la Torre, F. D. 2008. Image-based shaving. *Computer Graphics Forum (Eurographics) 27*, 2, 627–635.

Perez, P., Gangnet, M., and Blake, A. 2003. Poisson image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3, 313–318.

Prince, S., Elder, J., Warrell, J., and Felisberti, F. 2008. Tied factor analysis for face recognition across large pose differences. *IEEE Pattern Recognition and Machine Intelligence 30*, 6, 970–984.

Tenenbaum, J., and Freeman, W. 2000. Separating style and content with bilinear models. *Neural Computation 12*, 6, 1247–1283.

Turk, M. A., and Pentland, A. P. 1991. Face recognition using eigenfaces. In *Proceedings of CVPR*, 586–591.

Vlasic, D., Brand, M., Pfister, H., and Popovic, J. 2005. Face transfer with multiliner models. *ACM Transactions on Graphics (Proc. SIGGRAPH) 24*, 3, 426–433.

Wei, L., and Levoy, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of ACM SIGGRAPH 2000*, 479–488.

Weyrich, T., Matusik, W., Pfister, H., Bickel, B., Donner, C., Tu, C., McAndless, J., Lee, J., Ngan, A., Jensen, H., and Gross, M. 2006. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics (Proc. SIGGRAPH) 25*, 3, 1013–1024.