Example-Based Modeling of Facial Texture from Deficient Data

Arnaud Dessein, William A. P. Smith, Richard C. Wilson, Edwin R. Hancock

Example-Based Modeling of Facial Texture from Deficient Data

Arnaud Dessein[1] William A. P. Smith[2] Richard C. Wilson[2] Edwin R. Hancock[2]
[1] IMB / LaBRI, Université de Bordeaux, France
[2] Department of Computer Science, University of York, UK

Abstract

We present an approach to modeling ear-to-ear, high-quality texture from one or more partial views of a face with possibly poor resolution and noise. Our approach is example-based in that we reconstruct texture with patches from a database composed of previously seen faces. A 3D morphable model is used to establish shape correspondence between the observed data across views and training faces. The database is built on the mesh surface by segmenting it into uniform overlapping patches. Texture patches are selected by belief propagation so as to be consistent with neighbors and with observations in an appropriate image formation model. We also develop a variant that is insensitive to light and camera parameters, and incorporate soft symmetry constraints. We obtain textures of higher quality for degraded views as small as 10 pixels wide, than a standard model fitted to non-degraded data. We further show applications to super-resolution where we substantially improve quality compared to a state-of-the-art algorithm, and to texture completion where we fill in missing regions and remove facial clutter in a photorealistic manner.

1. Introduction

The problem of modeling complete and high-quality facial texture from input data that is “deficient” (i.e., low-resolution, noisy and/or incomplete) has many potential applications. These include facial super-resolution, face texture completion, pre-processing for manual or automatic face recognition, creation of fully textured avatars from a single low-quality view, estimation of high-resolution frontal views from low-resolution CCTV images of non-frontal faces, or prediction of clean-shaven appearance.

Besides typical noise sources, deficient data arise in a number of ways. The complete ear-to-ear texture of a face is never fully visible in a single image. This is particularly problematic for images in a non-frontal pose where as much as half of the texture may be missing. A similar problem occurs when parts of a face are occluded by scene clutter, other people, glasses or facial hair. When a face image is captured by a cheap sensor, at large distance, or with heavy compression, then the resolution may be insufficient for further processing. These deficiencies make many face processing problems more difficult when dealing with real-world data.

The predominant approach to face modeling problems has been statistical [6, 8, 9, 20]. Namely, learning the common modes of variation in face appearance from training data. Such models can be used to constrain many face processing problems. However, in the context of deficient data, statistical models suffer from a number of drawbacks (see Fig. 1). The state-of-the-art in face capture [4, 14] allows measurement of very high-resolution texture and shape information that can be used for photorealistic rendering (left). On the other hand, face modeling has failed to keep pace with the quality of data that can be captured from real faces (middle). The most critical weakness of existing statistical face models is their inability to accurately approximate unseen faces. This leads to models that fail to capture distinguishing, fine-scale details of a face. This situation is exacerbated further when such models are fitted to 2D images. Here, not only local details but global properties of identity such as gender and ethnicity are lost, even when the input data is of relatively high quality (right).

We address the issues caused by deficient data with a novel example-based approach to face texture modeling. We fit a 3D morphable model for shape correspondence between training faces and input images. The 3D model is segmented into uniform overlapping patches via farthest-point sampling. We build a database for texture on the segmented model, and estimate the optimal patch combination by loopy belief propagation so that the reconstructed textures best explain the data whilst ensuring local consistency.

To assemble these known ingredients in a unified framework, we also introduce several technical contributions. First, farthest-point sampling is enhanced with an original patch growing scheme based on geodesic projections, so as to create uniform patches that overlap. Second, loopy belief propagation is formulated on the 3D mesh rather than 2D images via an appropriate model of image formation, where we explicitly simulate how vertices project to the observed pixel grid. Third, we propose a texture normalization step to make our methods insensitive to camera and lighting

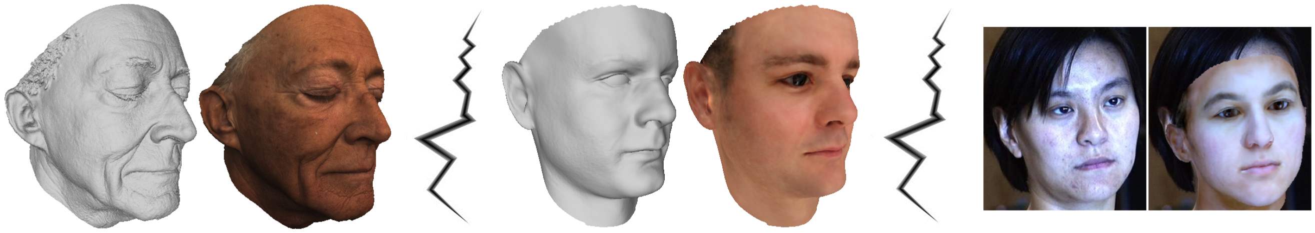

Figure 1. There is a dramatic gap between the realism afforded by state-of-the-art face capture [4] (left) and face modeling [20] (middle). This gap is larger still when a model like [20] is fitted to an image, even if the image is of relatively high quality (right). Zoom in for detail.

parameters, show the coherence of this step with our image formation model, and derive new updates of normalized patch compatibilities for tractability. Fourth, we incorporate soft symmetry constraints on the model, which requires revisiting both segmentation and loopy belief propagation.

In addition, our methods have a number of appealing properties. First, the pipeline is fully intrinsic to the 3D model, so that we avoid undesired distortions due to extrinsic processing such as 2D flattening or 3D Euclidean operations. Second, the patch database is independent of the degradation factor, just needs to be trained once, and can be updated incrementally with any new data. Third, we explicitly account for issues of occlusions and of pose changes between multiple views. Fourth, we can handle varying appearance caused by changes in camera or illumination between training and testing, or even between input images.

We assume that the morphable model parameters for each input image are provided by a separate algorithm. It is not the topic of this paper to estimate these parameters, and there are many algorithms for fitting a 3D morphable model to 2D images [1, 2, 5, 6, 20, 23, 28], including low-resolution input [17]. We here use the well-known model [20] fitted to ground-truth data. This gives an oracle upper bound on texture reconstruction performance, since it is likely that shape will be less accurate when estimated on degraded images. We thus also evaluate the effect of perturbations in the parameters to simulate fitting errors or spatial noise, by considering the extreme case of the mean shape only as a rough estimate. This way, we obtain an oracle lower bound on performance since shape is then the worst possible. Pose is also not perfect because we do not reestimate it based on the mean shape, meaning only the gross positioning and orientation of the face are correct. This is relevant since the gross pose can be estimated relatively well given a couple of manually or automatically detected keypoints (e.g., eye centers, mouth corners, nose tip).

2. Related Work

Face modeling Statistical face models such as active appearance models [8, 9] and 3D morphable models [6, 20] assume that there is a dense correspondence between any pair of faces, which transforms faces into a vector space where any convex combination yields a valid face. Specifically, a 3D morphable model is a deformable mesh $\mathcal{M}(\boldsymbol{\alpha}) = (\mathcal{K}, \mathbf{S}(\boldsymbol{\alpha}))$. The connectivity is given by the simplicial complex $\mathcal{K}$, a set whose elements can be vertices $\{i\}$, edges $\{i, j\}$ or faces $\{i, j, k\}$, with indices $i, j, k \in \{1, \ldots, N\}$. The shape $\mathbf{S}(\boldsymbol{\alpha})$ of the N vertices depends on a vector $\boldsymbol{\alpha} \in \mathbb{R}^S$ which contains the parameters of a low-dimensional, linear model learned from training data via PCA:

$$\mathbf{S}(\boldsymbol{\alpha}) = \bar{\mathbf{S}} + \sum_{m=1}^{S} \alpha_m \mathbf{S}_m \ , \tag{1}$$

where $\bar{\mathbf{S}} \in \mathbb{R}^{3\times N}$ is the mean shape, and $\mathbf{S}_m \in \mathbb{R}^{3\times N}$ are S shape modes. Texture is also commonly described similarly, and since PCA implicitly acts as a low-pass filter, these models fail to capture high-frequency detail.

Mohammed *et al.* [15] tackle this problem for 2D morphable models in the context of 2D face synthesis. Instead of fitting to image data, they randomly generate parameters of a statistical face model before adding high-frequency detail by importing real patches with local consistency and that agree with the global face image. Patches are selected in raster scan order using a greedy algorithm with partially-stochastic decisions to improve variation in the synthesis.

Our model can be viewed as a 3D extension of [15], with some important differences. First, our model and patch database are built on a deformable 3D mesh rather than in 2D image space. Second, we require a model of image formation to relate appearance. Third, we use belief propagation for a better approximation to the optimal choice of patches. Fourth, we account for color transformations between the model and images, enabling a wider range of images to be analyzed. Fifth, we do not use a global statistical texture model but condition results on observed data only.

Super-resolution Facial texture modeling from images of poor resolution shares similarities with super-resolution problems [7, 19, 22, 25]. However, unlike classical 2D super-resolution, multiple images of a face are unlikely to differ only by sub-pixel translations and rotations in the image plane. Hence, the majority of face-specific approaches to super-resolution work with a single image and are more accurately described as face hallucination (aka recogstruction as a neologism for reconstruction by recognition).

A pioneering work in this context was done by Baker and Kanade [3]. They incorporated a database for a class of objects (text or faces) into previous probabilistic frameworks for super-resolution. Freeman *et al*. [11] also used a database (generic though can be restricted to faces) of corresponding high- and low-resolution texture patches. They form a Markov random field over the pixel lattice, and apply belief propagation to select high-resolution patches whose low frequencies match those observed, while ensuring local consistency via neighboring patch compatibilities. They also perform color normalization in local contrast, though this lacks of a physical interpretation compared to our texture normalization. Liu *et al*. [12] developed a face-specific version of this by using a non-stationary patch database and a global statistical model trained on face images. Mortazavian *et al*. [16] further extended the example-based approach by exploiting a 3D morphable face model. However, the 3D model is used solely to establish correspondence between a single input image and the training data. Super-resolution is then performed in a flattened texture space, at the cost of distortions inherent to the parameterization, with the frequency-domain approach of previous 2D works. Hence, they have no explicit model of image formation.

We emphasize that our face texture modeling approach is more general than super-resolution. Indeed, we reconstruct a full texture on a 3D model that can be rendered in any pose, whereas classical super-resolution methods reconstruct a partial texture on a 2D image in the exact same pose as observed. When applied to super-resolution problems, our approach is also the first to handle multiple views with complex 3D pose changes involving possibly large out-of-plane rotations and translations as well as occlusions.

Moreover, existing example-based methods to super-resolution suffer from a number of limitations. First, they consider a fixed degradation factor (often a power of 2), either because they build a specific dictionary for a given degradation factor, or because the frequency decomposition makes assumptions about the frequency range that has been lost and must be reconstructed. Second, there is no obvious way to integrate information from multiple images, particularly when the 3D pose changes. Third, they assume that the model and image can be directly compared without accounting for appearance variation due to camera or lighting. Our proposed model overcomes each of these weaknesses. Lastly, we handle faces as small as 10 pixels wide whereas typical super-resolution algorithms require over 30 pixels.

Texture completion Statistical models can be used to explain missing data by fitting a complete model to partial data. In 2D, view-based active appearance models [9] can predict frontal appearance from a profile view. Similarly, a multilinear model with a mode for pose variation, such as [26], can also predict any viewpoint given a single view.

Lüthi *et al*. [13] proposed a 3D extension, with a probabilistic approach for fitting to partial data which provides the subspace of all solutions consistent with the observed data. Like 2D methods, it suffers however from the fundamental limitation of statistical approaches. They can only describe a face in terms of its most common aspects of appearance. This leads to a loss of detail and an inability to meaningfully reconstruct deficient data.

3. Overlapping Mesh Segmentation

In this section, we discuss the segmentation of the mean face shape into uniform overlapping patches. We first sample the mesh uniformly to create patches via the underlying tessellation. We then grow the patches so that they overlap.

3.1. Uniform Sampling

We process sampling with a greedy farthest-point strategy similar to [21]. We denote by $\mathcal{I}_l = \{i_1^\star, \dots, i_l^\star\} \subset \mathcal{K}$ the set of first l selected samples. We also define D_l as the geodesic distance map to $\mathcal{I}_l$. The next sample $i_{l+1}^\star$ is selected as the vertex which is the farthest from all samples:

$$i_{l+1}^\star = \underset{i \in \mathcal{K}}{\arg\max}\ D_l(i) \ . \tag{2}$$

For the next iteration, D_{l+1} is updated as the minimum between D_l and the distance map to the new sample:

$$D_{l+1}(i) = \min\bigl\{D_l(i), D(i, i_{l+1}^\star)\bigr\} \ . \tag{3}$$

We continue the process until the desired number M of vertices are sampled. In the end, patches $\mathcal{P}_1, \dots, \mathcal{P}_M$ are obtained via the geodesic Voronoi tessellation:

$$\mathcal{P}_m = \left\{ i \in \mathcal{K} \colon D(i, i_m^\star) = \min_{i^\star \in \mathcal{I}_M} D(i, i^\star) \right\} \ . \tag{4}$$

3.2. Patch Growing

We denote by $\mathcal{E}$ the set of pairs (m, n) such that patches $\mathcal{P}_m$ and $\mathcal{P}_n$ share an edge $\{i, j\} \in \mathcal{K}$. To grow patch $\mathcal{P}_m$, we consider separately each of its neighbors $\mathcal{P}_n$ with $(m, n) \in \mathcal{E}$, and define thresholds d_{mn} as follows:

$$d_{mn} = \rho \times D(i_m^\star, i_n^\star) \ , \tag{5}$$

where $\rho \geq 0$ is an overlap ratio set by the user. The overlap $\mathcal{O}_{mn}$ of $\mathcal{P}_m$ onto $\mathcal{P}_n$ is then constructed by projections:

$$\mathcal{O}_{mn} = \left\{ i \in \mathcal{P}_n \colon \min_{j \in \mathcal{P}_m} D(i, j) \leq d_{mn} \right\} \ . \tag{6}$$

Eventually, a grown patch $\mathcal{Q}_m$ is obtained by concatenation:

$$\mathcal{Q}_m = \mathcal{P}_m \cup \bigcup_{n | (m,n) \in \mathcal{E}} \mathcal{O}_{mn} \ . \tag{7}$$

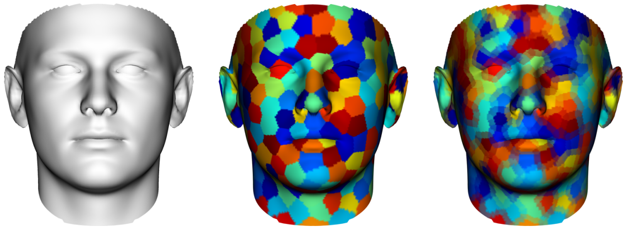

Figure 2. Segmentation results. The mean face (left) is segmented by first sampling the mesh uniformly to create patches (middle), and then growing the patches so that they overlap (right).

3.3. Segmentation Results

We segmented the mean face of the morphable model into $M = 200$ patches, so that they are about the size of the eyes. We visualize the obtained patches by assigning different colors and averaging them in overlaps (see Fig. 2). Geodesic distances were computed by fast marching as in [21]. Sampling was seeded with eye centers to avoid stochastic indeterminacy due to random initialization. Patches were grown using a small overlap ratio $\rho = 0.2$ to keep results visually interpretable. In subsequent experiments, this ratio was increased to $\rho = 0.5$ for a better merging of solutions, which corresponds to a natural overlap of about half the patch size. The whole process took a couple of minutes on a standard computer, but needs only to be computed once beforehand and can then be stored.

Overall, the results show that the segmentation produces roughly uniform patches over the face, and hence transfers the notion of a regular overlapping patch structure from 2D images to 3D meshes. We also notice that the seeding allows single patches to span the eyes, which might be desirable to avoid that different parts of the iris are assigned different colors during texture reconstruction.

4. Texture Patch Selection

In this section, we present our methods to select texture patches from the training database. We define observation error functions and patch overlap compatibilities on the mesh via an image formation model. The optimal patch combination is then estimated by loopy belief propagation.

4.1. Image Formation Model

It is not straightforward to measure plausibility between an observed image and training textures that have supposedly produced this image. Doing it in the image domain does not seem relevant for two main reasons. Indeed, this requires rendering all training faces on the observed image grid, which gets demanding as the database size increases. Moreover, errors computed in the image domain are not directly related and scaled to compatibilities of training texture patches pasted on the mesh for reconstruction.

We thus compute errors directly on the mesh after texture sampling. When the observed images are not degraded, we simply sample them on the mesh by back-projection of color and bilinear interpolation in the pixel grid. Textures at visibility boundaries are filled in by nearest-neighbor interpolation instead, and missing data are left untextured. However, since back-projection requires a sufficiently fine grid, this model reveals insufficient in case of poor resolution.

To overcome this, we simulate the formation of a low-resolution image with pixel colors $\mathbf{c}_r$ as produced by a mesh with high-resolution textures $\mathbf{t}_i^+$. We assume that the mesh vertices are uniformly and finely sampled compared to the low-resolution pixel size. The first assumption is reasonable since the fitted shape involves local deformations of the mean shape which is almost uniformly sampled. The second one holds too since we consider degraded images of 10 to 40 pixels wide which is negligible compared to more than 50,000 vertices in the model.

Hence, the color $\mathbf{c}_r$ of a given pixel r can be approximated by averaging the contributions of the visible vertices that project onto that pixel:

$$\mathbf{c}_r = \frac{1}{|\mathcal{V}_r|} \sum_{i \in \mathcal{V}_r} \mathbf{t}_i^+ \quad , \tag{8}$$

where $\mathcal{V}_r$ is the set of visible vertices that project onto pixel r. This partitions the visible vertices via an equivalence relation where i and j belong to the same class if they are in the same set $\mathcal{V}_r$. We denote the equivalence class of a visible vertex i as $\overline{\mathcal{V}}_i$, and by $r(i)$ the index of the pixel onto which i is projected. We can thus compute errors on the mesh by sampling a low-resolution observed texture via nearest-neighbor interpolation $\mathbf{t}_i^- = \mathbf{c}_{r(i)}$, while resampling the high-resolution training textures $\mathbf{t}_{ip}^+$ into low-resolution textures $\mathbf{t}_{ip}^-$ by local averaging in projected pixels:

$$\mathbf{t}_{ip}^- = \frac{1}{|\overline{\mathcal{V}}_i|} \sum_{j \in \overline{\mathcal{V}}_i} \mathbf{t}_{jp}^+ \quad . \tag{9}$$

Finally, to prevent from sampling incorrect textures because of background corruption along the projected outline, we exclude border pixels from sampling so that vertices falling in such pixels are considered as occluded. This can be done together with vertex visibility testing and texture sampling, by using a depth-buffer on a refined pixel grid. Typically, this grid is obtained by simply rescaling the coarse grid by the super-resolution factor, which allows the buffer to be reused when rendering the reconstructed image.

4.2. Observation Error Function

We model the likelihood of training patches given the low-resolution observed textures via a Gaussian potential:

$$\phi_{mp} = \exp\left\{-\frac{E_{mp}}{2\sigma_E^2}\right\} \quad , \tag{10}$$

where σ_E^2 is a variance parameter, and E_{mp} is the Euclidean error in patch $\mathcal{Q}_m$ between observed and training textures:

$$E_{mp} = \sum_{i \in \mathcal{Q}_m} \left\| \mathbf{t}_{ip}^- - \mathbf{t}_i^- \right\|_2^2 \ . \tag{11}$$

This implicitly accounts for a color degradation by Gaussian noise, though other kinds of noise (e.g., Poisson, salt-and-pepper) could be handled as well by plugging other discrepancy measures. To account for missing data, occluded vertices can simply be assigned a null error. When having multiple input images, we sum up the errors across views.

4.3. Patch Overlap Compatibility

We model the plausibility of high-resolution training patches given their neighbors through a Gaussian potential:

$$\psi_{mnpq} = \exp\left\{ -\frac{C_{mnpq}}{2\sigma_C^2} \right\} \ , \tag{12}$$

where σ_C^2 is another variance parameter, and C_{mnpq} is the Euclidean error in overlap $\mathcal{O}_{mn}$ between training textures of neighbor patches:

$$C_{mnpq} = \sum_{i \in \mathcal{O}_{mn}} \left\| \mathbf{t}_{ip}^+ - \mathbf{t}_{iq}^+ \right\|_2^2 \ . \tag{13}$$

Practically, some neighbor patches may not overlap, either because of using a too small overlap ratio ρ or of the patch before growing being composed of its center only (which happens for big triangles filling the interior of the mouth). Such neighbors can simply be ignored and removed from the edge structure $\mathcal{E}$ since they would always be fully compatible. Compatibilities are also multiplied by the number of views to maintain a similar influence compared to errors.

4.4. Loopy Belief Propagation

The underlying probabilistic model can be cast into a graphical model, as a factor graph where nodes represent patches $\mathcal{P}_m, \mathcal{P}_n, \ldots$ and edges represent neighboring relations between overlapping patches. This network has a structure of Markov random field where dependencies reduce to pairwise interactions only (see Fig 3).

Solving for the most likely realization of patches from training faces $\mathbf{t}_{ip}^+, \mathbf{t}_{iq}^+, \ldots$ in this model is an expensive combinatorial problem. Indeed, the probability of training patches not only depends on their similarity with observed data $\mathbf{t}_i^-$, but also on the neighboring compatibility between selected patches. An exact brute-force approach rapidly becomes intractable as the database size grows. Instead, we follow an iterative algorithm for approximate inference known as loopy belief propagation [18], which is quadratic in the number of training faces. Specifically, we employ the min-sum variant which rather works on the log-likelihood for better numerical stability, due to the high number of vertices involved in exponentiated sums of ϕ_{mp}, ψ_{mnpq}.

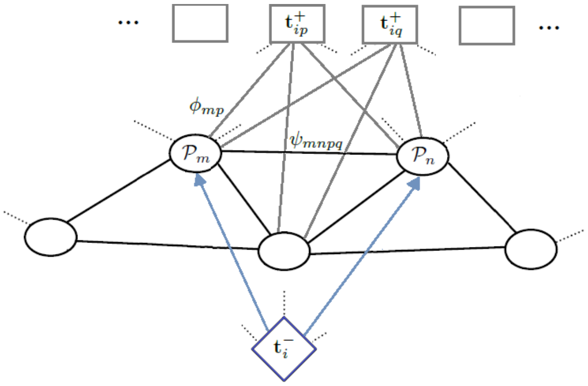

Figure 3. Network structure. The probabilistic model is a Markov random field over patches $\mathcal{P}_m, \mathcal{P}_n, \ldots$ where the training data $\mathbf{t}_{ip}^+, \mathbf{t}_{iq}^+, \ldots$ are selected per patch to be consistent with observations $\mathbf{t}_i^-$ and with neighbors according to potentials ϕ_{mp}, ψ_{mnpq}.

Once patches are selected, the final texture is obtained by averaging the textures in overlap regions. We could also employ Poisson blending here, but it is quite demanding and did not seem to improve systematically results in our experiments, certainly because of patch overlap compatibilities ensuring local consistency. The whole process of error computations and belief propagation takes a couple of seconds. The patch compatibilities can be precomputed once for the whole database, which rather takes several minutes.

As a final remark, the algorithm, apart from the number of iterations I, depends on a single parameter, the ratio $\alpha = \sigma_E^2 / \sigma_C^2$ of the two variance parameters. This variance ratio controls the trade-off between reconstructing the observations as faithfully as possible and ensuring that neighbor patches are as compatible as possible. In all our experiments, we fixed $\alpha = 1$ to intuitively give the same influence between error and compatibility terms. A number of iterations $I = 10$ was also found sufficient for convergence.

5. Texture Normalization

In this section, we describe a variant to perform texture patch selection in a normalized space that is insensitive to camera and lighting parameters. We notably introduce a texture appearance model, and modify observation errors as well as patch compatibilities accordingly.

5.1. Texture Appearance Model

We assume that the color channels are independent, and thus consider a single channel while omitting its index in the sequel. We account for illumination with ambient light plus a directed light source. The ambient lighting simply multiplies the intrinsic texture $\tilde{t}_i$. The directed lighting can be modeled according to the well-known dichromatic model of reflectance [29]. Hence, it comprises additive diffuse and specular terms that apply an affine transform to the intrinsic texture (aka albedo). We also account for acquisition settings via a basic camera model with unknown color gain and offset per color channel when recording appearance.

We thus obtain an affine model for texture appearance:

$$t_i = a_i \tilde{t}_i + b_i \ \ , \tag{14}$$

where a_i encodes the camera gain, ambient lighting and diffusion, while b_i encodes the camera offset and specularity. We note that only diffusion and specularity, due to directed lighting, actually depend on vertex i, via angles between local surface normals and light source or viewer directions.

5.2. Normalized Observation Error Function

We normalize textures before computing the observation errors according to the above model for texture appearance. Since the high-resolution observations are hidden, we do so in the low-resolution domain, by normalizing observed textures t_i^- via statistical standardization in mean and variance as $\hat{t}_i^- = (t_i^- - \mu)/\sigma$, where we use the mean μ and standard deviation σ for textures t_i^- of visible vertices.

Similarly, we normalize the training textures as $\hat{t}_{ip}^- = (t_{ip}^- - \mu_p)/\sigma_p$, where we use now the mean μ_p and standard deviation σ_p for the textures t_{ip}^- corresponding to observed vertices. We use these exact same parameters to normalize the high-resolution training textures as $\hat{t}_{ip}^+ = (t_{ip}^+ - \mu_p)/\sigma_p$.

To compare training texture patches with observed textures, we now compute the error function in the low-resolution space after normalization:

$$e_{mp} = \sum_{i \in \mathcal{Q}_m} \left(\hat{t}_{ip}^- - \hat{t}_i^-\right)^2 \ \ . \tag{15}$$

This allows training patches to be used for reconstruction even if they were captured with another camera and under a different pose or illumination. In more detail, the error function is now invariant under affine transformations of textures, and hence under arbitrary changes in camera calibration via color gain and offset, as well as arbitrary scalings of ambient and directed light intensities. Going further, we may assume that the training database samples a fine enough subset of all possible angles between the light source and viewer directions, or at least the angles we expect to observe. Hence, the process is also robust, if not invariant, against changes in such angles up to neglecting deviations of local surface normals between the observed face and training faces.

5.3. Reconstructed Patch Overlap Compatibility

On the contrary to observation errors, compatibilities need to be computed based on the effective appearance in the final reconstructed space. In other words, we must simulate how neighbor patches would interact in overlaps if they were chosen for reconstruction. When normalized training textures are selected, we need to insert back the normalization parameters from the observed textures for reconstruction. It leads to reconstruction as $\check{t}_{ip}^+ = \sigma \hat{t}_{ip}^+ + \mu$, and $\check{t}_{ip}^- = \sigma \hat{t}_{ip}^- + \mu$, in the high- and low-resolution spaces respectively. For coherence of our methods, we would expect the degradation of the high-resolution reconstruction to equal the low-resolution reconstruction under our models for texture degradation, normalization and reconstruction:

$$\check{t}_{ip}^- = \frac{1}{|\overline{\mathcal{V}}_i|} \sum_{j \in \overline{\mathcal{V}}_i} \check{t}_{ip}^+ \ \ . \tag{16}$$

This is indeed true as shown in the supplementary material.

The patch overlap compatibilities are thus now defined in the reconstructed high-resolution space:

$$c_{mnpq} = \sum_{i \in \mathcal{O}_{mn}} \left(\check{t}_{ip}^+ - \check{t}_{iq}^+\right)^2 \ \ . \tag{17}$$

These compatibilities cannot, unfortunately, be precomputed since we do not know the reconstructed textures $\check{t}_{ip}^+, \check{t}_{iq}^+$ in advance. Given the computational burden of calculating these compatibilities, it is unrealistic to compute them again from scratch for each new observed face. Nonetheless, it is possible to precompute several related quantities so as to update the compatibilities on the fly. Even if still quadratic in the number of training faces, this makes the computation much faster during testing, typically a couple of seconds instead of several minutes. These updates are developed in the supplementary material.

6. Experimental Results

We now present experiments. We explain how the patch database is built, show texture modeling results as well as applications to super-resolution and texture completion.

6.1. Patch Database

We build a patch database with images from the CMU PIE face database [24]. The coefficients of the fitted morphable model and camera parameters provide a 3D model for each subject and a dense correspondence to each image. To obtain a complete texture over the face, we combine the 3 available views of the same subject under the same illumination by stitching them and filling unobserved data with Poisson blending [10]. We sample 5 different lighting conditions, ambient light only, and ambient plus one flash light among front, top, left and right. As the morphable model does not account for glasses, beards or mustaches, we exclude corresponding subjects from the database, leading to a total of 175 face models representing 35 different subjects under 5 lighting conditions. We also exclude temporarily the subject under testing for a fair out-of-sample evaluation.

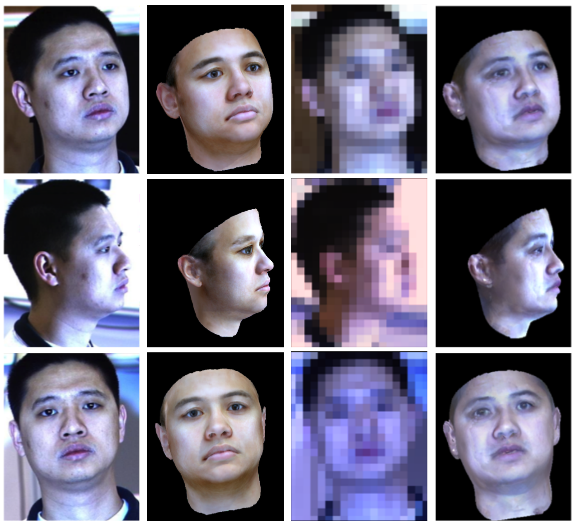

Figure 4. Texture modeling from multiple low-quality. Col. 1: High-quality ground-truth. 2: Morphable model fit to high-quality ground-truth [20]. 3: Low-quality, color-transformed input images. 4: High-quality output of our method.

6.2. Texture Modeling

We here assess our methods for facial texture modeling from several incomplete poses. To account for possibly varying illumination conditions and acquisition settings, we employ the normalized variant of our algorithm.

In Fig. 4, we show results on reconstructing a complete face model for a given subject. Three reference views (Col. 1) are fitted to a state-of-the-art morphable model [20]. Even with high-quality input, the model is incapable of generalizing to the fine details of the face and unseen lighting conditions (Col. 2). Low-quality input images are obtained after downsampling by a factor of 16 to produce partial views of about 12 pixels wide (Col. 3). In the second and third rows, a color transformation is also applied to the input. We solve for a single high-quality model that can be rendered in any pose and color space (Col. 4). Since we solve for normalization parameters per image, we are able to correct the significant color differences between views.

6.3. Super-Resolution

We now apply our approach to classical super-resolution. This can easily be done by rendering the output of our face model in the same pose as observed. For visualization, we also display the non-modeled parts of the image after simple bicubic interpolation to match the output resolution.

In Fig. 5, we show super-resolution results for four subjects in nearly frontal poses. The high-resolution images (Col. 1) are artificially degraded, so that we have a ground-truth to compute error metrics, and then reconstructed. The state-of-the-art morphable model [20] fitted to the original images is again incapable of generalizing to the fine details (Col. 2). The images are obtained after downsampling by a factor of 16 (Col. 3). From this deficient input, our method yields faces with a high level of plausible detail, matching the ground-truth well (Col. 4). On the other hand, a state-of-the-art example-based super-resolution algorithm [27], trained with exactly the same images as our model, fails to reconstruct any meaningful detail (Col. 5).

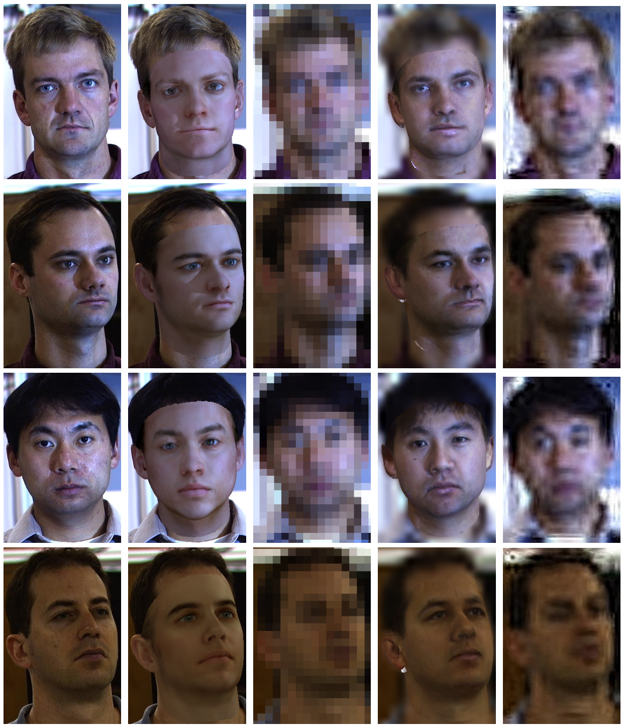

Figure 5. Super-resolution. Col. 1: High-resolution ground-truth. 2: Morphable model fit to high-resolution ground-truth [20]. 3: Low-resolution input. 4: High-resolution output of our method. 5: Face-trained example-based super-resolution [27].

In Table 1, we present results of a leave-one-out evaluation on the full database. We use the root mean square normalized by the number of rendered pixels as an error metric (scores are given in percentage error). We consider degradations δ of 4, 8, 16, 32, the 5 illumination conditions, and average results across subjects. Given the poor visual results and computational burden (about an hour per image) of [27], we exclude it from this quantitative evaluation.

Overall, applying our methods with the same shape fit as the morphable model provides a better texture, independently of lighting conditions, and even for the coarsest degradation (Rows 2 to 5). Although using a rough shape fit via the mean shape degrades results, our methods are still competitive with the morphable model fit to high-resolution input, and compare favorably except for the coarsest degradation (Rows 6 to 9). This is thanks to the image formation model, which allows to cope with misalignment of up

Table 1. Super-resolution evaluation (basic/normalized variants).

	δ	Ambient	Front	Left	Right	Top
[20]	1	06.0	16.9	12.9	13.7	17.0
Fit	4	04.8/04.0	12.1/09.9	09.8/08.4	10.5/08.6	12.1/10.0
	8	05.1/04.3	12.5/10.4	10.2/08.9	10.9/09.1	12.5/10.6
	16	05.3/04.7	13.5/11.7	10.8/10.0	11.6/10.1	13.5/12.0
	32	05.9/05.5	15.5/14.9	12.6/12.6	13.7/12.6	16.0/15.5
Mean	4	05.9/05.1	13.1/10.8	09.8/09.4	11.6/09.7	13.2/11.0
	8	06.2/05.4	13.5/11.4	11.2/09.9	12.0/10.2	13.6/11.7
	16	06.5/06.0	14.7/13.0	12.1/11.2	12.8/11.4	14.8/13.3
	32	07.2/06.9	16.9/16.4	14.0/14.1	14.8/14.0	17.3/16.9

to one low-resolution pixel because of averaging contributions regardless of their distance to the pixel center. The results also reveal that the normalized variant systematically outperforms the basic version. This is because normalization allows more expressiveness in the reconstruction process by adapting patch textures according to the input texture parameters. As we would expect, the reconstruction errors also increase with the degradation factor. They are also greater when directed light is present compared to ambient lighting only. We believe this results from both variants being unable to cope fully with nonlinearities in appearance due to shadowing and specularity.

6.4. Texture Completion

We finally consider texture completion problems. We set the degradation factor to 1, and discard the pixel aggregation step from the image formation model. We also reinsert the observed data in the reconstructed texture by Poisson blending at boundaries. Lastly, we constrain the face model with soft symmetry to ensure the plausibility of important symmetric features such as the eyes. Details on soft symmetry constraints are given in the supplementary material.

In Fig. 6, we demonstrate an example of texture completion for a face in a non-frontal pose. The non-frontal input only shows part of the face texture (Col. 1), which is clearly visible when sampling the view to texture the morphable model fit and rotate it to a frontal pose (Col. 2). Our methods reconstruct a complete and photorealistic texture (Col. 3) which, when projected into a frontal image (Col. 4), closely matches a real frontal view of the face (Col. 5).

In Fig. 7, we show an example of removing extraneous features, or clutter. The input images contain respectively a mustache and glasses (Col. 1). Such features are not modeled by the morphable model so that a pixel-wise binary outlier mask can be calculated via the fitting error (Col. 2). The morphable model fit is then textured by sampling the masked image (Col. 3). Performing texture completion over the occluded regions and blending with the original textures provides an ear-to-ear, high-quality texture (Col. 4). By rendering in the original image, we obtain photorealistic results where the extraneous features are removed (Col. 5).

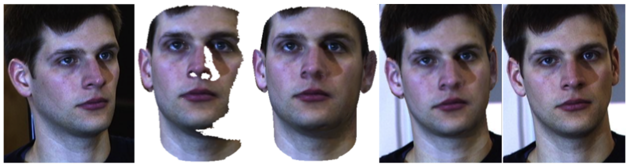

Figure 6. Texture completion for a non-frontal pose. Col. 1: Non-frontal input. 2: Morphable model fit with sampled texture. 3: Completed texture. 4: Frontal output. 5: Frontal ground-truth.

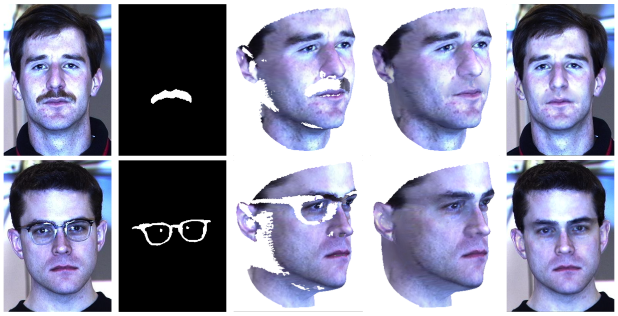

Figure 7. Texture completion for removal of extraneous features. Col 1: Cluttered input. 2: Outlier mask. 3:. Morphable model fit with sampled texture. 4:. Completed texture. 5: Cleaned output.

7. Conclusion

We presented an example-based approach to face texture modeling from deficient views. We obtained promising results which are encouraging for further developments.

Firstly, dense correspondence restricts synthesis to examples of features in the same location as previously observed. Allowing patches to move onto the mesh surface is, however, not easy to deal with and should be investigated.

Another direction is to improve the texture appearance model with a full mixing matrix to capture spectral overlaps between light contents and between camera responses. We could also normalize per patch to account for multiple light sources or nonlinearities due to shadowing and specularity.

Lastly, we would like to build a model using intrinsic texture rather than appearance under representative lighting conditions. This way, much less training data would be needed. Nonetheless, error functions would require a model of illumination and patch selection would be more complex.

As for applications, we plan to evaluate our approach for face recognition in the wild. However, this requires designing our own shape fitting algorithm. Finally, tuning the three main design parameters (patch number, overlap factor, variance ratio) beyond the intuitive values used here might improve performance depending on the task at hand.

Acknowledgment

This work was supported by grant DSTLX1000070369 from the Defence Science & Technology Laboratory.

References

[1] O. Aldrian and W. A. P. Smith. Inverse rendering of faces with a 3D morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(5):1080–1093, May 2013. 2

[2] B. Amberg, A. Blake, A. Fitzgibbon, S. Romdhani, and T. Vetter. Reconstructing high quality face-surfaces using model based stereo. In *IEEE Int. Conf. on Computer Vision*, pages 1–8, Rio de Janeiro, Brazil, Oct. 2007. 2

[3] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1167–1183, Sept. 2002. 3

[4] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graph. (SIGGRAPH)*, 29(4):40:1–40:9, July 2010. 1, 2

[5] V. Blanz, A. Mehl, T. Vetter, and H.-P. Seidel. A statistical method for robust 3D surface reconstruction from sparse data. In *Int. Symp. on3D Data Processing, Visualization and Transmission*, pages 293–300, Thessaloniki, Greece, Sept. 2004. 2

[6] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, Los Angeles, CA, USA, Aug. 1999. 1, 2

[7] S. Borman and R. L. Stevenson. Super-resolution from image sequences – a review. In *Midwest Symp. on Circuits and Systems*, pages 374–378, Notre Dame, IN, USA, Aug. 1998. 2

[8] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Computer Vision — ECCV'98*, volume 1407 of *LNCS*, pages 484–498. Springer, Berlin Heidelberg, 1998. 1, 2

[9] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. View-based active appearance models. *Image Vision Comput.*, 20(9–10):657–664, Aug. 2002. 1, 2, 3

[10] A. Dessein, W. A. P. Smith, R. C. Wilson, and E. R. Hancock. Seamless texture stitching on a 3D mesh by Poisson blending in patches. In *Int. Conf. on Image Processing*, pages 2031–2035, Paris, France, October 2014. 6

[11] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Comput. Graph. Appl.*, 22(2):56–65, Mar./Apr. 2002. 3

[12] W. Liu, D. Lin, and X. Tang. Neighbor combination and transformation for hallucinating faces. In *IEEE Int. Conf. on Multimedia and Expo*, pages 145–148, Amsterdam, The Netherlands, July 2005. 3

[13] M. Lüthi, T. Albrecht, and T. Vetter. Probabilistic modeling and visualization of the flexibility in morphable models. In *Mathematics of Surfaces XIII*, volume 5654 of *LNCS*, pages 251–264. Springer, Berlin Heidelberg, 2009. 3

[14] W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, M. Weiss, and P. Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Eurographics Symp. on Rendering*, pages 183–194, Grenoble, France, June 2007. 1

[15] U. Mohammed, S. J. D. Prince, and J. Kautz. Visio-lization: Generating novel facial images. *ACM Trans. Graph. (SIGGRAPH)*, 28(3):57:1–57:8, Aug. 2009. 2

[16] P. Mortazavian, J. Kittler, and W. Christmas. A 3-D assisted generative model for facial texture super-resolution. In *IEEE Int. Conf. on Biometrics: Theory, Applications, and Systems*, pages 452–458, Washigton, DC, USA, Sept. 2009. 3

[17] P. Mortazavian, J. Kittler, and W. Christmas. 3D morphable model fitting for low-resolution facial images. In *IAPR Int. Conf. on Biometrics*, pages 132–138, New Delhi, India, Mar./Apr. 2012. 2

[18] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Conf. on Uncertainty in Artificial Intelligence*, pages 467–475, Stockholm, Sweden, July/Aug. 1999. 5

[19] S. C. Park, M. K. Park, and G. K. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Process. Mag.*, 20(3):21–36, May 2003. 2

[20] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, pages 296–301, Genova, Italy, Sept. 2009. 1, 2, 7, 8

[21] G. Peyré and L. D. Cohen. Geodesic remeshing using front propagation. *Int. J. Comput. Vis.*, 69(1):145–156, Aug. 2006. 3, 4

[22] C. Riedinger, M. N. Khemakhem, and G. Chollet. A study of some super resolution techniques in video sequence. In *Int. Conf. on Sciences of Electronics, Technologies of Information and Telecommunications*, pages 386–392, Sousse, Tunisia, Mar. 2012. 2

[23] S. Romdhani and T. Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 986–993, San Diego, CA, USA, June 2005. 2

[24] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression (PIE) database. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 46–51, Washington, DC, USA, May 2002. 6

[25] M. E. Tipping and C. M. Bishop. Bayesian image super-resolution. In *Adv. Neural Inf. Process. Syst.*, volume 15 of *NIPS*, pages 1279–1286. MIT Press, Cambridge, 2003. 2

[26] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In *Computer Vision — ECCV 2002*, volume 2359 of *LNCS*, pages 447–460. Springer, Berlin Heidelberg, 2002. 3

[27] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Process.*, 19(11):2861–2873, Nov. 2010. 7

[28] L. Zhang and D. Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):351–363, Mar. 2006. 2

[29] T. Zickler, S. P. Mallick, D. J. Kriegman, and P. N. Belhumeur. Color subspaces as photometric invariants. *Int. J. Comput. Vision*, 79(1):13–30, Aug. 2008. 6