High Fidelity Facial Hair Capture
ICT Technical Report No. ICT-TR-01-2012.
August, 2012.

Graham Fyffe
USC Institute for Creative Technologies *

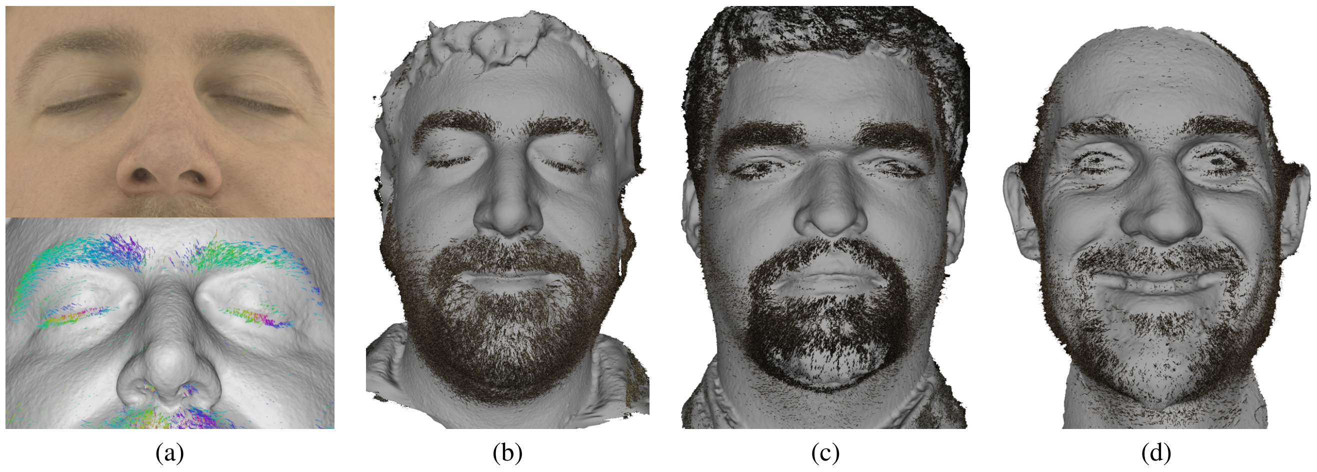

Figure 1: (a) Renderings of reconstructed facial geometry and hair particles with texture (top) and false color illustrating hair orientation (bottom). (b-d) Gray renderings of three different subjects with recovered facial geometry and hair particles.

Abstract

We propose an extension to multi-view face capture that reconstructs high quality facial hair automatically. Multi-view stereo is well known for producing high quality smooth surfaces and meshes, but fails on fine structure such as hair. We *exploit* this failure, and automatically detect the hairs on a face by careful analysis of the pixel reconstruction error of the multi-view stereo result. Central to our work is a novel stereo matching cost function, which we call *equalized cross correlation*, that properly accounts for both camera sensor noise and pixel sampling variance. In contrast to previous works that treat hair modeling as a synthesis problem based on image cues, we reconstruct facial hair to explain the same high-resolution input photographs used for face reconstruction, producing a result with higher fidelity to the input photographs.

1 Introduction

Modeling human hair from photographs is a topic of ongoing interest to the graphics community. Yet, the literature is predominantly concerned with the hair volume on the scalp, and it remains difficult to capture digital characters with interesting facial hair. Recent stereo-vision-based facial capture systems (e.g. [Furukawa and Ponce 2010][Beeler et al. 2010]) are capable of capturing fine skin detail from high resolution photographs, but any facial hair present on the subject is reconstructed as a blobby mass. To create convincing digital characters, an artist typically has to remove the offending geometry, and re-model the hair using artistic tools. Our primary goal in this work is to automate the creation of facial hair for a face captured using multi-view stereo, with high fidelity to the input photographs. Prior work in facial hair photo-modeling is based on learned priors and image cues (e.g. [Herrera et al. 2010]), and does not reconstruct the individual hairs belonging uniquely to the subject. We propose a method for capturing the three dimensional shape of complex, multi-colored facial hair from a small number of photographs taken simultaneously under uniform illumination. This includes the hairs that make up the eyebrows, eyelashes, and any other relatively trim facial hair present.

We analyze the pixel reconstruction errors in multi-view stereo, and observe that errors are high on pixels where the subject is not well represented by a smooth mesh, most notably around the hairs on the face. This arises from the typical assumption in multi-view stereo, that the face may be reconstructed as a smooth polygonal mesh with a texture map. We also note that the resolution obtained using readily available camera equipment is capable of imaging individual facial hairs. This motivates our approach, which is to reconstruct hairs that explain the pixels that could not be explained using a smooth mesh. We also re-texture the parts of the mesh that lie beneath the hair, obtaining a more complete model of the human face including facial hair that results in a lower overall pixel reconstruction error.

Besides hair detection and reconstruction, we derive a novel stereo matching cost function, which we call *equalized cross correlation* (Section 4.1). It is related to normalized cross correlation but accounts for camera sensor noise and pixel sampling variance. This allows us to discriminate between reconstruction error caused by violation of the smooth mesh assumption versus that caused by noise.

2 Related Work

Several works address the issue of hair generation in some form or another based on photographs. Most of these are concerned with modeling the hair volume on the head. Wei et al. [Wei et al. 2005] compute 2D hair orientation fields, and grow 3D fibers that follow

*e-mail: fyffe@ict.usc.edu

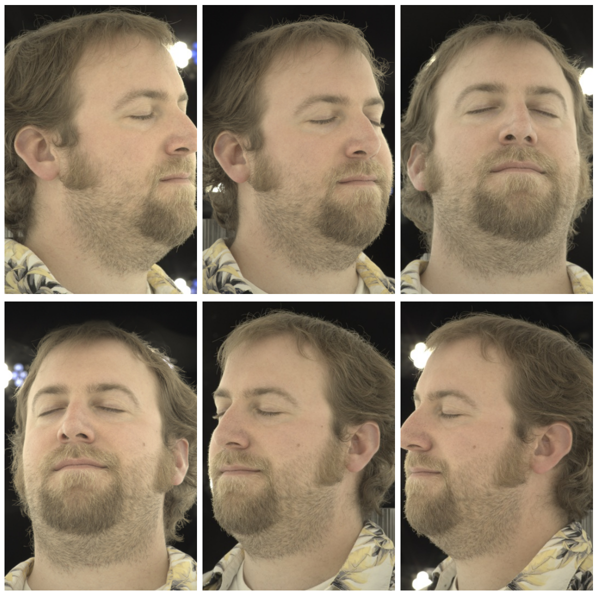
Figure 2: The arrangement of photographs used in our method is typical of multi-view stereo face capture. Our examples use six photographs under uniform illumination, such as those shown here.

the fields using orientation triangulation. Paris et al. [Paris et al.] use many photos and synthesize long hair on the head, also by tracing particles along orientation fields. Luo et al. [Luo et al. 2011] consider a hair volume in motion. While these are similar in some ways to our work, we locate and reconstruct individual hairs, instead of using cues such as gross hair orientation. In that sense the work of Jakob et al. [Jakob et al. 2009] is more similar to ours, but uses a focus sweep scheme to capture a hair assembly instead of multi-view stereo. Others address facial hair specifically. Herrera et al. [Herrera et al. 2010] model facial hair based on statistical analysis of photographs, not directly reconstructing the individual hairs of the subject. These related works inspire our method in several respects. Contemporary with this work is the method of [Beeler et al. 2012], which produces high quality facial hair using passive multi-view stereo and has the advantage of producing connected piecewise linear hair primitives instead of particles. However, the method relies on additional close-up photographs of the facial hair regions, and requires significant post-processing and smoothing steps to produce plausible hairs.

3 Overview of Method

We base our method on multi-view passive stereo reconstruction using high-resolution photographs. In our examples, we use six photographs at 2600×3908 pixels captured simultaneously from six digital SLR cameras surrounding the face. The face is illuminated by a uniform field of white LED light, though any relatively uniform illumination could be used. Optionally, specular reflection may be canceled by the use of polarization filters on the lights and cameras, as in [Ghosh et al. 2011]. Such photographs are typical of passive stereo facial capture, requiring no special setup for hair (see Figure 2). We aim to explain the set of input photographs using a textured face mesh and facial hair particle primitives. We first discuss facial geometry reconstruction, followed by facial hair detection and removal, and finally facial hair reconstruction.

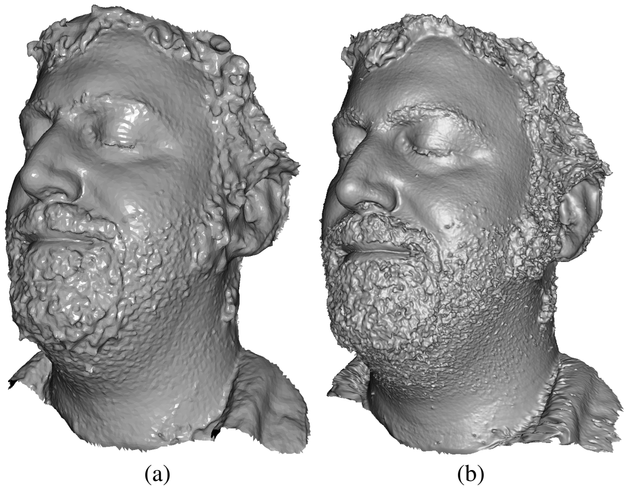
Figure 3: (a) An example base mesh, obtained using the method of [Furukawa and Ponce 2010]. (b) Our initial refined mesh obtained with the method in Section 4.

4 Facial Geometry Reconstruction

Reconstruction begins with the method of [Furukawa and Ponce 2010], up to the point of Poisson meshing [Kazhdan et al. 2006] to construct a low-resolution base mesh of the face. Our method differs in the way the mesh is further refined. We smooth the base mesh and parameterize it into a UV texture coordinate space defined by an artist (although many works exist on automatic mesh parameterization). All further processing occurs in the UV space. Our refinement scheme solves for a displacement map in the UV space which displaces the base mesh vertices along their normals. In some ways this is similar to the mesh refinement in [Furukawa and Ponce 2010] or [Beeler et al. 2010], however we perform the refinement using a multi-view stereo reconstruction algorithm that is not restricted to the local minimum surrounding the base mesh. Consequently, the base mesh may be of relatively low resolution. (Figure 3 shows an example base mesh and refined mesh used in our work.) In all computations, we ignore any unused pixels in the UV space. We express the refinement problem as the minimization of the following pairwise energy function defined on a graph coincident with the UV texture grid, with $\mathcal{V}$ representing vertices, and edges $\mathcal{E}$ between all horizontal and vertical neighbors, which we solve using the TRW-SAD algorithm [Ghosh et al. 2011]:

$$E(X) = \sum_{s\in\mathcal{V}} D_s(x_s) + \sum_{(s,t)\in\mathcal{E}} S_{st}(x_s, x_t) \tag{1}$$

Here, D_s is a data term and S_{st} is a smoothing term. The data term is as follows:

$$D_s(x_s) = \frac{\sum_{(i,j)\in\text{stereocam}_s} w_{is} w_{js} C_{ij}(\text{pos}_s(x_s))}{\sum_{(i,j)\in\text{stereocam}_s} w_{is} w_{js}} \tag{2}$$

where w_{is} is the dot product of the view vector and the surface normal of the base mesh at site s. The smoothing term is:

$$S_{st}(x_s, x_t) = \lambda_{\text{smooth}} \frac{((\text{pos}_s(x_s) - \text{pos}_t(x_t)) \cdot n_{st})^2}{|\text{pos}_s(x_s) - \text{pos}_t(x_t)|^2} \tag{3}$$

where pos_s is 3D position given a displacement, stereocam_s is the set of neighboring camera pairs in which $\text{pos}_s(x_s)$ is visible (visibility information is updated after every iteration of the algorithm), C is the matching cost function, n_{st} is the estimated surface normal between sites s and t, and λ_{smooth} is the smoothing term weight. ($\lambda_{\text{smooth}} = 1$ in all figures in this work.) The matching cost function is evaluated over a 5×5 pixel grid set tangent to the surface and centered at the 3D point under consideration. The grid spacing is scaled in both dimensions to be as close as possible to the pixel spacing in the two images being compared. This construction avoids any dependence on the UV parameterization of the mesh, and hence the matching is unaffected by any unevenness in the parameterization. The matching cost function is similar to normalized cross correlation (NCC), but modified to take into account the noise in the image signal, as detailed in Section 4.1. We discuss the choice of n_{st} in Section 4.2.

4.1 Equalized Cross Correlation (ECC)

Typical stereo matching cost functions make use of normalized cross correlation (NCC) over a small patch of pixels as a measure of pixel similarity. NCC is advantageous as it is invariant to scaling or offsetting the pixel values. However, NCC has the disadvantage that it makes no consideration of the noise present in the image signals. If a patch of pixels has low contrast, then after normalization the patch will be dominated by noise. Such noisy patches may cause spurious stereo matches, which may frustrate good matches in neighboring high-contrast patches. To remedy this shortcoming, we modify NCC to account for noise in the image. We call this equalized cross correlation (ECC). The derivation of ECC follows. We model the values in n×n-pixel image patches $\mathcal{I}$ (and $\mathcal{J}$ likewise) as drawn from a normal distribution (for pixel coordinate p) based on some unknown idealized image patch μ:

$$\mathcal{I}_p \sim \mathcal{N}(\bar{\mathcal{I}} + \breve{\mathcal{I}}\mu_p, \sigma^2_{\mathcal{I}p} + \tau^2_{\mathcal{I}}), \tag{4}$$

where $\bar{\mathcal{I}}$ and $\breve{\mathcal{I}}$ define a linear transform on the idealized patch, $\sigma^2_{\mathcal{I}p}$ models pixel noise and $\tau^2_{\mathcal{I}}$ models the variance due to pixel sampling of the idealized patch. Following [Healey and Kondepudy 1994], we empirically fit a pixel noise model to our cameras:

$$\sigma^2_{\mathcal{I}p} = a\mu_{\mathcal{I}p} + b \quad \approx \quad a\mathcal{I}_p + b, \tag{5}$$

with $a = 0.01, b = 0.001$ for the red channel, $a = 0.008, b = 0.001$ for green and $a = 0.01, b = 0.001$ for blue, and $\mu_{\mathcal{I}p}$ is the (unknown) mean of the Normal distribution $\mathcal{N}(\mu_{\mathcal{I}p}, \sigma_{\mathcal{I}p})$ representing the pixel noise model. We assume a camera sensor with linear radiometric response. Nonlinear pixel data would have to be linearized before proceeding. Assuming the idealized patch μ has zero mean and unit variance, we estimate:

$$\bar{\mathcal{I}} \approx \tfrac{1}{n^2}\sum_p \mathcal{I}_p, \qquad \breve{\mathcal{I}} \approx \sqrt{\tfrac{1}{n^2}\sum_p (\mathcal{I}_p - \bar{\mathcal{I}})^2}, \tag{6}$$

and we approximate the pixel sampling variance $\tau^2_{\mathcal{I}} \approx \frac{1}{4}\breve{\mathcal{I}}^2$, which is a heuristic estimate equal to the variance introduced if the pixels were resampled using bilinear interpolation with random sub-pixel offsets. Next we define the *normalized* patch $\widehat{\mathcal{I}}$, similar to NCC:

$$\widehat{\mathcal{I}}_p = \breve{\mathcal{I}}^{-1}(\mathcal{I}_p - \bar{\mathcal{I}}) \quad \sim \mathcal{N}(\mu_p, \breve{\mathcal{I}}^{-2}(\sigma^2_{\mathcal{I}p} + \tau^2_{\mathcal{I}})). \tag{7}$$

The matching cost L is simply the negative log likelihood of $\widehat{\mathcal{I}} - \widehat{\mathcal{J}}$:

$$L = \tfrac{1}{n^2}\sum_p \frac{(\widehat{\mathcal{I}}_p - \widehat{\mathcal{J}}_p)^2}{2(\breve{\mathcal{I}}^{-2}(\sigma^2_{\mathcal{I}p} + \tau^2_{\mathcal{I}}) + \breve{\mathcal{J}}^{-2}(\sigma^2_{\mathcal{J}p} + \tau^2_{\mathcal{J}}))}. \tag{8}$$

The $\frac{1}{n^2}$ term averages out the matching cost over the patch, which is necessary to avoid over-counting when the costs of all the patches are summed together in (1). (Each patch is also over- or under-counted by some factor due to summing over vertices instead of pixels, however the same factor applies to the smoothing term and hence the terms balance.) For exposition, we define the *equalized* patch $\widetilde{\mathcal{I}}$ (and $\widetilde{\mathcal{J}}$ similarly), also expanding and simplifying on τ:

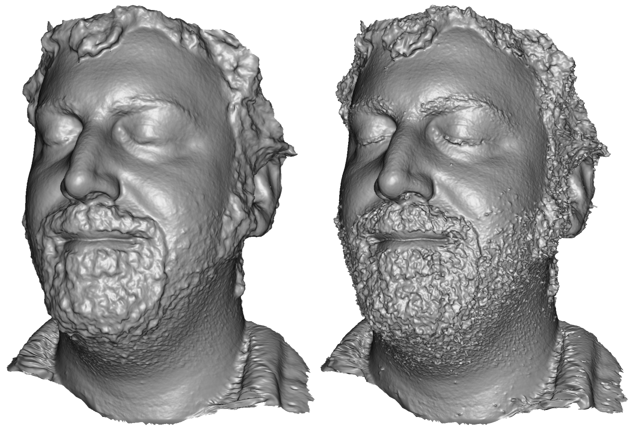
(a) (b)

Figure 4: Recovered facial geometry of the subject in Figure 2. (a) Using NCC photometric cost. (b) Using ECC photometric cost. Note how the NCC result exhibits uniform smoothing over the entire face, whereas the ECC result exhibits spatially varying smoothness correlated to the actual smoothness of the subject.

$$\widetilde{\mathcal{I}}_p = \frac{\widehat{\mathcal{I}}_p}{\sqrt{2n^2(\breve{\mathcal{I}}^{-2}\sigma^2_{\mathcal{I}p} + \breve{\mathcal{J}}^{-2}\sigma^2_{\mathcal{J}p} + \frac{1}{2})}}, \tag{9}$$

which leads to the simplified expression for ECC:

$$L = \sum_p (\widetilde{\mathcal{I}}_p - \widetilde{\mathcal{J}}_p)^2. \tag{10}$$

It is critical to note here that the sum of squared differences over equalized patches, as shown above, properly models the error in the reconstruction of the input images. However, we may not use it directly in our energy formulation, since low-contrast patches will unfairly dominate high-contrast patches. Instead we must normalize the cost function based on the patch norms:

$$\frac{L}{|\widetilde{\mathcal{I}}|^2 + |\widetilde{\mathcal{J}}|^2} = 1 - \frac{2\sum_p \widetilde{\mathcal{I}}_p \widetilde{\mathcal{J}}_p}{|\widetilde{\mathcal{I}}|^2 + |\widetilde{\mathcal{J}}|^2}. \tag{11}$$

Indeed, if $\sigma_{\mathcal{I}p}, \sigma_{\mathcal{J}p}$ are zero, then the above reduces to NCC. Unfortunately by restoring fairness in the energy formulation, we lose the equivalence to reconstruction error. To mitigate this effect, we weight the cost function by the denominator from the previous iteration's solution, yielding the revised ECC function:

$$C = L \times \frac{(|\widetilde{\mathcal{I}}|^2 + |\widetilde{\mathcal{J}}|^2)_{\text{previous}}}{|\widetilde{\mathcal{I}}|^2 + |\widetilde{\mathcal{J}}|^2}. \tag{12}$$

So far we have assumed the image patches have only a single channel of data. For color images, we may combine the costs as follows:

$$C_{\text{rgb}} = \frac{\sum_{c\in\{r,g,b\}} L_c}{3} \times \frac{\left(\sum_{c\in\{r,g,b\}} |\widetilde{\mathcal{I}}_c|^2 + |\widetilde{\mathcal{J}}_c|^2\right)_{\text{previous}}}{\sum_{c\in\{r,g,b\}} |\widetilde{\mathcal{I}}_c|^2 + |\widetilde{\mathcal{J}}_c|^2}. \tag{13}$$

The division by 3 follows from the assumption that the color channels are highly correlated within a patch. Figure 4 compares geometry recovered using the NCC photoconsistency cost to that using

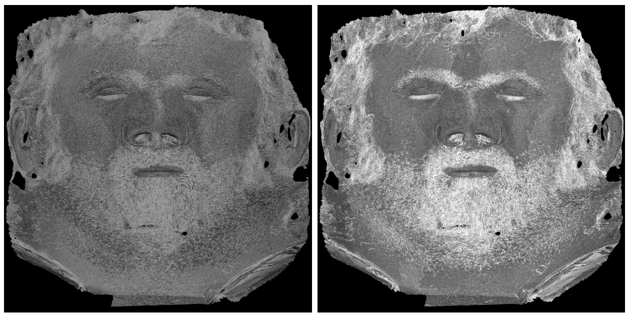

Figure 5: Optimized photoconsistency cost of the subject in Figure 2. (a) Using NCC. (b) Using ECC. Note how the higher cost regions in ECC correspond to non-smooth regions of the face, particularly hair, and smooth regions of the face have relatively uniform cost. NCC is often confused by texture detail, such as skin pores and eyelid creases, as pixel sampling variance is not taken into account.

the ECC photoconsistency cost. The ECC result exhibits sharper features without relying on any spatially varying smoothing term heuristics. Figure 5 visualizes the optimized NCC cost versus ECC. ECC accounts for pixel noise and sampling variance, leaving only the cost associated with violations of the smooth mesh assumption.

4.2 Smoothing Term

For a suitably smooth base mesh that's not too far from the real mesh, setting n_{st} equal to the surface normal of the base mesh is not a bad choice. This favors surfaces with a similar surface normal as the smooth base mesh, and is used for all the figures in this work. We also experimented with an approximation to the second-order prior in [Woodford et al. 2009], by locally fitting a plane to the previous iteration's displacement map in the neighborhood around the sites s and t, then setting n_{st} to the normal of the plane, but we found no clear advantage to this method over the base mesh prior. If photometric surface normals are available, then n_{st} may be chosen based on the photometric surface normals, as in [Ghosh et al. 2011]. Alternatively, n_{st} may be chosen as a function of the gradient of texture intensity, as in [Beeler et al. 2010].

4.3 Multi-View Texture Blending

We reconstruct a color texture map of the face in the same UV space used for the parameterization of the mesh. Hence, each mesh vertex corresponds to exactly one pixel in the texture map. We reconstruct the color of a vertex by projecting its 3D position onto all of the cameras, and blending the pixel values from the input images. We omit views in which the vertex is not visible. Since the pixel intensities in the different cameras may not match exactly and we wish to avoid seams in the reconstructed texture, we formulate a linear least squares problem over all vertices in terms of not only their absolute texture values, but also their texture *gradients*. For a given vertex s, the absolute texture value estimate T_s is:

$$T_s = \frac{\sum_{i \in \mathrm{cam}_s} w_{is}^2 \sigma_{\mathcal{I}s}^{-2} \mathcal{I}_s}{\sum_{i \in \mathrm{cam}_s} w_{is}^2 \sigma_{\mathcal{I}s}^{-2}}, \tag{14}$$

where $\mathcal{I}_s$ is the pixel value in view i at the projected position of $\mathrm{pos}_s(x_s)$.The texture gradient estimate G_{st} between s and t is:

$$G_{st} = \frac{\sum_{i \in \mathrm{cam}_{st}} w_{is} w_{it} (\sigma_{\mathcal{I}s}^2 + \sigma_{\mathcal{I}t}^2)^{-1} (\mathcal{I}_t - \mathcal{I}_s)}{\omega_{\mathrm{prior}} + \sum_{i \in \mathrm{cam}_{st}} w_{is} w_{it} (\sigma_{\mathcal{I}s}^2 + \sigma_{\mathcal{I}t}^2)^{-1}} \tag{15}$$

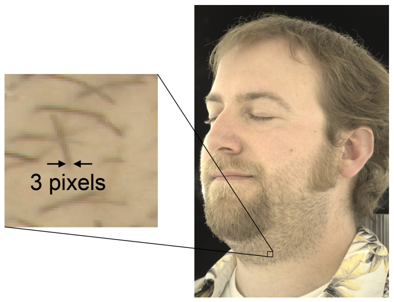

Figure 6: Using consumer digital SLR cameras to photograph a face, a single facial hair is typically two to four pixels wide.

where $\mathrm{cam}_{st} = \mathrm{cam}_s \cap \mathrm{cam}_t$, and $\omega_{\mathrm{prior}} = 0.01$ serves to smooth out regions with no information. We then formulate the sparse linear least squares problem:

$$E(\mathrm{T}) = \sum_{s \in \mathcal{V}} (\mathrm{t}_s - T_s)^2 + \lambda_G \sum_{(s,t) \in \mathcal{E}} (\mathrm{t}_t - \mathrm{t}_s - G_{st})^2 \tag{16}$$

where t_s is the texture value at site s, and λ_G is a user-supplied weight parameter that controls the smoothness of the texture blending between cameras. All of the results in this work use $\lambda_G = 10$. We minimize (16) using the TRW-S algorithm [Kolmogorov 2006] with Gaussian (quadratic) closed-form message passing, which acts as a robust sparse linear least squares solver. Figure 9(a) shows a result of the texture blending.

5 Facial Hair Detection and Removal

The reconstructed facial geometry and texture from Section 4 contains a poor representation of facial hair, which must be removed before synthesizing hair particles. We detect which mesh vertices correspond to hair using a combination of the ECC photoconsistency cost in UV space, and more traditional orientation filters in image space. These two hair indicators and the hair removal method are described in the sections following.

5.1 Photoconsistency Hair Indicator

We claim that our ECC matching cost (in Section 4.1) accurately models the pixel reconstruction error not associated with pixel noise or pixel sampling variance, and is therefore due to the subject being poorly represented by a smooth mesh. On a human face, this indicates where the hair is, since hair is not a smooth surface. We may therefore compute a binary hair indicator value α^D for each pixel site s in UV space (Figure 9(b)), by comparing the value of the data term (2) to a threshold D_{thresh} (where $D_{\mathrm{thresh}} = \frac{1}{2}$ in this work):

$$\alpha_s^D = \left(\frac{D_s(x_s)}{D_{\mathrm{thresh}}} > 1 \right). \tag{17}$$

5.2 Orientation Filter Hair Indicator

Having photographed the subject at a resolution of 2600×3908 pixels, the individual facial hairs in the photographs are typically between two and four pixels wide (see Figure 6), and can be detected as line segments. We detect lines and their orientations using the 9-tap G_2, H_2 separable steerable filter quadrature pair from [Freeman and Adelson 1991], which suits the hair width in our photographs. We divide the detected orientation strength by its standard deviation as predicted by the camera noise model, to normalize it

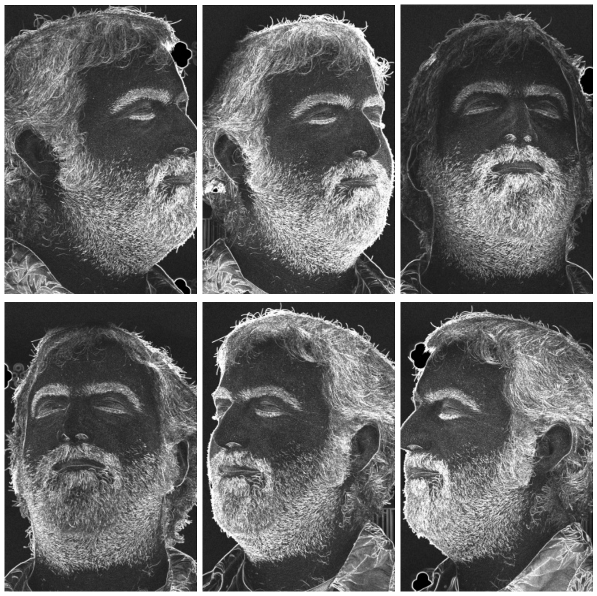

Figure 7: Orientation strength detected in the six input photographs, normalized with respect to surface albedo.

with respect to surface albedo. Figure 7 visualizes the orientation strength for the subject in Figure 2. We next determine which pixels contain a local maximum orientation strength, indicating a detected feature. Figure 8 illustrates the process on a patch of facial hair. We use quadratic interpolation to locate the local maximum in either the vertical or horizontal 3-neighborhood centered at each pixel, whichever is most perpendicular to the detected orientation. Unlike previous work, we locate the maxima of both the total orientation strength $|G_2^\theta|^2 + |H_2^\theta|^2$ and also of the line detector strength $|G_2^\theta|^2$. Instead of testing the G_2, H_2 phase to determine whether a feature is a line or an edge, we test if the local maximum of the line detector strength is greater than half the local maximum of the total orientation strength, i.e. $\lceil |G_2^\theta|^2 \rceil > \frac{1}{2} \lceil |G_2^\theta|^2 + |H_2^\theta|^2 \rceil$, indicating a line feature. This greatly improves the localization of line features, which we have found may differ from the local maximum of the total orientation strength by more than one pixel. For later use during hair reconstruction, we store a 2D oriented line segment (at most one per pixel in each camera view) if the local maximum line detector strength is above a threshold value G_{thresh}. ($G_{\text{thresh}} = 50$ in this work.) We finally compute the hair indicator function α^G:

$$\alpha_s^G = \left(\frac{1}{|\text{cam}_s|} \sum_{i \in \text{cam}_s} F_{p_i} \frac{|G_2^\theta|_{p_i}^2}{G_{\text{thresh}}} > 1 \right), \tag{18}$$

where p_i is the projection of $\text{pos}_s(x_s)$ onto camera i, and $F_p = 1$ if a line feature is detected within the bounds of a pixel p (0 otherwise). Figure 9(c) visualizes this hair indicator function.

5.3 Combined Hair Indicator

The indicator functions α^D and α^G contain noise consisting of false positives and false negatives. Helpfully, the noise differs between the two indicator functions, as they are constructed from different sources of information. Thus, we may construct an improved hair indicator function α by blending the two indicator functions:

$$\alpha_s = \left(\frac{D_s(x_s)}{D_{\text{thresh}}} + \frac{1}{|\text{cam}_s|} \sum_{i \in \text{cam}_s} F_{p_i} \frac{|G_2^\theta|_{p_i}^2}{G_{\text{thresh}}} > 2 \right). \tag{19}$$

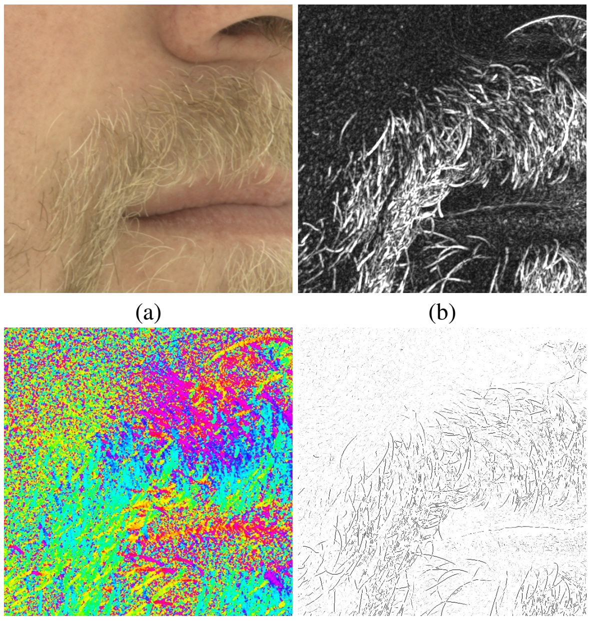

Figure 8: (a) Photograph cropped to show facial hair detail. Note that some hairs are lighter than the surrounding skin, while others are darker. (b) Detected orientation strength. (c) Detected orientation, rendered as hue. (d) Extracted oriented line segments, shaded according to strength of G_2 response.

Figure 9 compares α^D, α^G and α. Finally, we dilate the true values in the hair indicator map by one pixel to avoid classifying the fringes of hairs as skin. Guided by the hair indicator map, we hole-fill the non-skin regions of the texture map to produce a skin map using a simple smoothness assumption, again using Gaussian TRW-S message passing to efficiently solve the hole-filling problem. Our goal is to produce a texture map with the facial hair removed, to provide a plausible skin texture to underly the facial hair that we reconstruct later. Additionally, we smooth the skin geometry in these regions to reduce skin bumpiness beneath the hair. Figure 10 shows the result of hair geometry smoothing and hair texture removal.

6 Facial Hair Reconstruction

We reconstruct facial hairs as a set of 3D oriented hair particles. We first correct any view-dependent color variation in the input images using the face geometry and texture computed in Section 5.3. We then construct (up to) one 3D oriented hair particle for every 2D line segment detected in the input images in Section 5, determining its most likely 3D position by examining the 2D line segments in other views. The steps are described in the following.

6.1 Correcting View Dependence

The input photographs of the face exhibit some view-dependent effects due to skin reflectance and variation in the cameras. These variations would frustrate our subsequent hair reconstruction efforts, so we correct the skin tones of each input photograph in turn to match the blended texture result using the following scheme. First, we compute the RGB ratio at each pixel in UV space between the blended texture and the color contributed by the input photograph. Ratios computed for occluded pixels and pixels with hair are unreliable, so we ignore the values for any occluded pixel, and any pixel that is not skin according to the skin indicator map

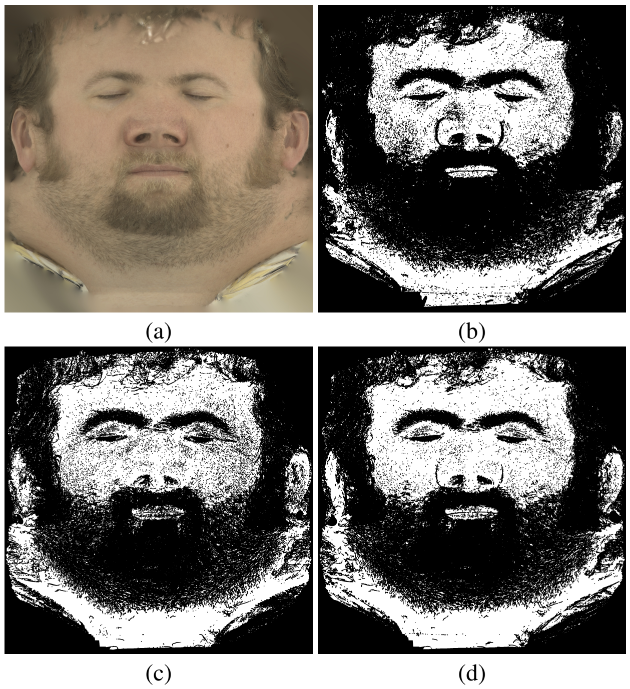

Figure 9: (a) Multi-View blended texture in UV space. (b) Hair indicator map α^D estimated from data term. (c) Hair indicator map α^G estimated from orientation strength. (d) Hair indicator map α combining data term with orientation strength.

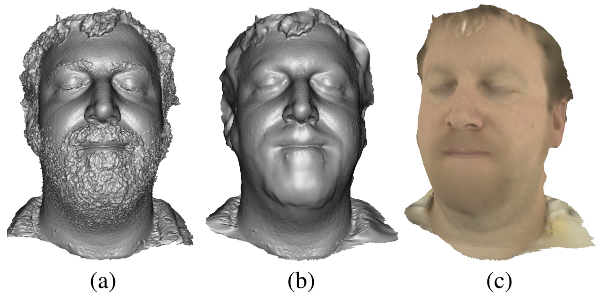

Figure 10: (a) Initial facial geometry. (b) Geometry with smoothed hair regions. (c) Geometry from (b) with texture after hair removal.

computed in Section 5.3. The ignored pixels are filled and the entire map is slightly blurred using (again) Gaussian TRW-S message passing, yielding a smooth, complete ratio map in UV space. The ratio map is then mapped onto the geometry and rasterized into the camera view over a white $(1, 1, 1)$ background. We soften the rasterized ratio image using a 2-pixel gaussian blur to eliminate any hard edges, and finally we multiply the input photograph by the ratio image to obtain an image that is nearly free of view-dependent color variation. Figure 11 illustrates this process.

6.2 Particle Triangulation

Any 2D line segment detected in the input images (in Section 5.2) is presumably the projection of some 3D line segment onto the image plane. Therefore, for each 2D line segment l_i in any view v_i, we search for the most likely 3D line segment that could have produced it (rejecting it if no satisfactory 3D line segment is found, as described later). We traverse a set of candidate 3D line segments

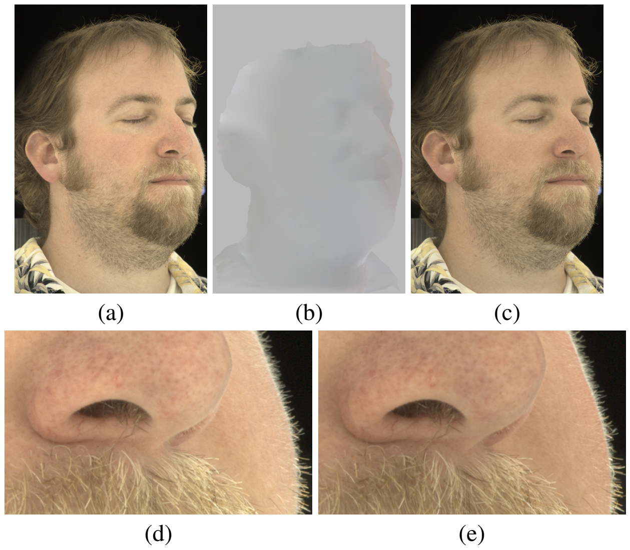

Figure 11: (a) Input photograph before view dependence correction. (b) Ratio image obtained from rasterizing the model with a smooth ratio map, based on the ratio of blended texture skin pixel values to input pixel values. Constructing the ratio map in UV space allows it to be smoothed without consideration of discontinuities in the camera view. (c) Input photograph multiplied by ratio image. (d) Zoom of (a). (e) Zoom of (b). Note the supression of view-dependent reflectance, especially near grazing angles.

by projecting l_i into each other view v_j (we rasterize into v_j a line corresponding to the ray through the center of l_i in view v_i) and considering each line segment l_j that it intersects. The intersection of 2D line segments l_i in view v_i and l_j in view v_j uniquely determines a 3D line segment l_{ij} via a 3D orientation triangulation similar to that in [Wei et al. 2005]. We call this 3D oriented line segment a *hair particle*. We then project the center of l_{ij} onto the facial surface along surface normals to determine a UV coordinate. We reject l_{ij} if the hair indicator map from Section 5.3 is false at that UV coordinate, or if the center of l_{ij} is too far from the surface (more than 1cm). We next compute a color for l_{ij} by averaging the pixel colors of the particle's projected position in unoccluded views (using the view-dependence-corrected pixel values from Section 6.1). We reject l_{ij} if its color is too similar to the underlying texture, based on a user defined threshold. We gauge the likelihood of l_{ij} by counting how many 2D oriented line segments agree with it in all unoccluded camera views (called the *supporting* line segments). We search for supporting line segments in a 5×5 pixel window around the 2D projected position in each view where l_{ij} is unoccluded by the facial geometry. A line segment supports the particle if the following condition holds, evaluated in image space:

$$((p - c) \cdot n)^2 + 16(d \cdot n)^2 < 3, \tag{20}$$

where p is the projected particle position, d is the projected particle orientation, c is the center of the 2D line segment and n is the 2D normal of the 2D line segment. This is a threshold heuristic that penalizes the projected distance from particles to 2D line segments, and penalizes the difference between projected particle orientation and 2D line segment orientation with greater weight. The coefficients were determined by user-defined tweaking. We reject l_{ij} if supporting line segments are found in fewer than three views. Out of all the particles l_{ij} visited and not rejected, we keep the one with the most supporting line segments and associate it with l_i. This scheme produces a dense set of 3D hair particles with a degree of redundancy, as particles seeded from different views may overlap significantly. However, the number of particles produced is not prohibitive, as it is bounded by the number of 2D line segments detected in the input images.

6.3 Skin Texture Revision

The hair maps computed in Section 5 contain some amount of false positives, which result in overly aggressive smoothing of the underlying skin texture in some areas. After the 3D hair particles are reconstructed (Section 6.2), a new underlying skin texture may be computed by rasterizing the hair particles into all views, then reprojecting the skin texture from the views, ignoring pixels that are occluded by hair. Skin texture areas that are occluded in all views are smoothed over and blending is performed using the same gradient-based scheme as in Section 4.3. Figure 12 illustrates the result of this revised texture projection.

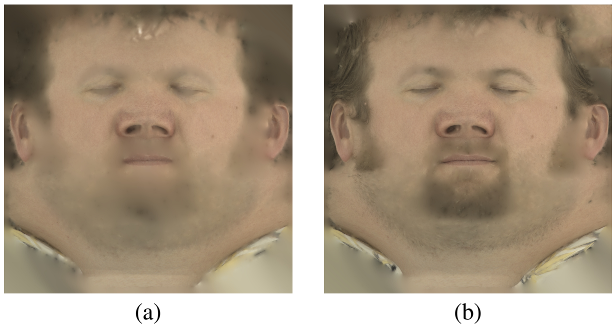

Figure 12: (a) Skin texture with overly aggressive smoothing due to false positives in hair indicator map. (b) Revised skin texture using the reconstructed 3D hair particles to evaluate visibility.

7 Results

We show facial geometry and hair particle reconstruction results for three different subjects, two with light colored hair and one with dark colored hair. The hairs are rendered using a simple rasterization algorithm using z buffering to handle occlusion. Figure 1(b-d) shows gray renderings of the results for the three subjects, with Figure 1(a) showing a detail region of the first subject with skin texture and colored particles, as well as a false color rendering showing the fidelity of recovered hair orientation. Figure 13 shows

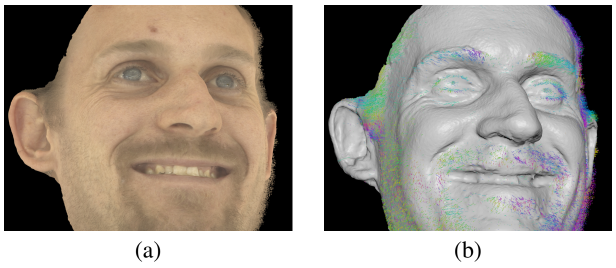

Figure 13: (a) Face model rendered with shaved skin texture map and colored hair particles. (b) False colored rendering to show geometric detail and hair particle orientation.

textured and false colored renderings of the second subject. A variety of hair types are reconstructed, including eyebrows, eyelashes, and stubble. Some false positives appear in the eyes due to specular reflections of the small lights used to illuminate the subject. Figure 14 shows textured renderings of the third subject, highlighting the recovered detail in the eyelashes and eyebrows. We also show results for faces without facial hair reconstruction, to compare our equalized cross correlation photoconsistency term to the more common normalized cross correlation. Figure 15 shows results for two subjects, illustrating the increased detail obtained from ECC without sacrificing smoothness, as observed earlier in Figure 4 for a hairy face. Using NCC, if the smoothing strength is increased to obtain similar smoothness in the cheeks compared to ECC, then finer details suffer (e.g. teeth, eyes). Likewise, if the smoothing strength is decreased to obtain similar sharpness in fine features compared to ECC, then smooth areas such as the cheeks become noisy. In previous works, similar attributes have been obtained using ad-hoc smoothing term heuristics. In contrast, ECC is derived from a principled model of image noise and variance, and the geometric smoothing term is not modified.

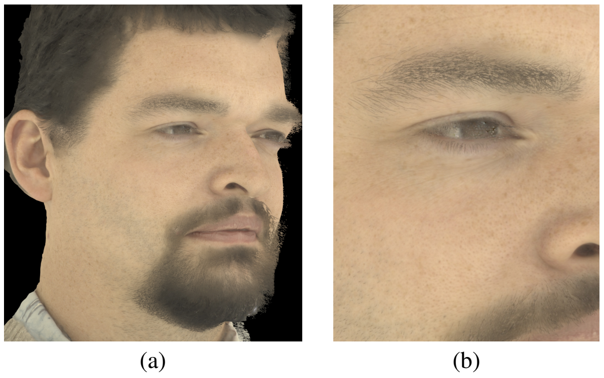

Figure 14: (a) Face model rendered with shaved skin texture map and colored hair particles. (b) Zoom of (a) to show detail.

8 Conclusion and Future Work

We have proposed a method for facial hair capture that makes use of existing data from multi-view stereo face capture, requiring no changes to typical passive stereo capture set-ups. The method is automatic, and generates a hairless face model and skin texture, and a particle-based hair model. We introduced the equalized cross correlation cost function (ECC), which serves as an improved photoconsistency measure useful to stereo geometry reconstruction in general. We leverage the meaningful residual error in ECC-based stereo reconstruction to help identify areas containing facial hair. We also described methods for removing facial hair from reconstructed facial geometry and texture, and a method for reconstructing the facial hair as a set of 3D oriented particles.

Three notable areas remain ripe for future work. First, we would like to explore improved reflectance modeling of the hair, enabling photorealistic rendering. Second, many people have extremely fine facial hairs that are invisible from a frontal view (the so-called "peach fuzz"). It may be possible to reconstruct peach fuzz by careful analysis of the silhouettes of the face. Third, we would like to combine our method with 3D hair growth strategies that model hairs as connected curves rooted to the face (such as [Beeler et al. 2012]). This might reduce false positives, such as the "floaters" visible over some areas, and may enable extending the method to handle longer facial hair such as long beards, and possibly the unification with solutions for reconstructing hair on the scalp.

Acknowledgements

We thank the following people for their support and assistance: Andrew Jones, Daniel Buck, Lynn Fyffe, Paul Graham, Abhijeet Ghosh, Jay Busch, Xueming Yu, Paul Debevec, Bill Swartout, Randall Hill and Randolph Hall.

References

BEELER, T., BICKEL, B., BEARDSLEY, P., SUMNER, B., AND GROSS, M. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH) 29*, 3, 40:1–40:9. 1, 2, 4

BEELER, T., BICKEL, B., NORIS, G., MARSCHNER, S., BEARDSLEY, P., SUMNER, R. W., AND GROSS, M. 2012. Coupled 3d reconstruction of sparse facial hair and skin. *ACM Trans. Graph. 31* (August), 117:1–117:10. 2, 7

FREEMAN, W. T., AND ADELSON, E. H. 1991. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 891–906. 4

FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 8, 1362–1376. 1, 2

GHOSH, A., FYFFE, G., TUNWATTANAPONG, B., BUSCH, J., YU, X., AND DEBEVEC, P. 2011. Multiview face capture using polarized spherical gradient illumination. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, ACM, New York, NY, USA, SA '11, 129:1–129:10. 2, 4

HEALEY, G., AND KONDEPUDY, R. 1994. Radiometric ccd camera calibration and noise estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 16*, 3 (mar), 267 –276. 3

HERRERA, T. L., ZINKE, A., WEBER, A., AND VETTER, T. 2010. Toward image-based facial hair modeling. In *Proceedings of the 26th Spring Conference on Computer Graphics*, ACM, New York, NY, USA, SCCG '10, 93–100. 1, 2

JAKOB, W., MOON, J. T., AND MARSCHNER, S. 2009. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph. 28*, 5. 2

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '06, 61–70. 2

KOLMOGOROV, V. 2006. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 10, 1568–1583. 4

LUO, L., LI, H., WEISE, T., PARIS, S., PAULY, M., AND RUSINKIEWICZ, S. 2011. Dynamic hair capture. 2

PARIS, S., CHANG, W., KOZHUSHNYAN, O. I., JAROSZ, W., MATUSIK, W., ZWICKER, M., AND DURAND, F. Hair photobooth: Geometric and photometric acquisition of real hairstyles. 2

WEI, Y., OFEK, E., QUAN, L., AND SHUM, H.-Y. 2005. Modeling hair from multiple views. *ACM Trans. Graph. 24*, 3, 816–820. 1, 6

WOODFORD, O., TORR, P., REID, I., AND FITZGIBBON, A. 2009. Global stereo reconstruction under second-order smoothness priors. 2115–2128. 4

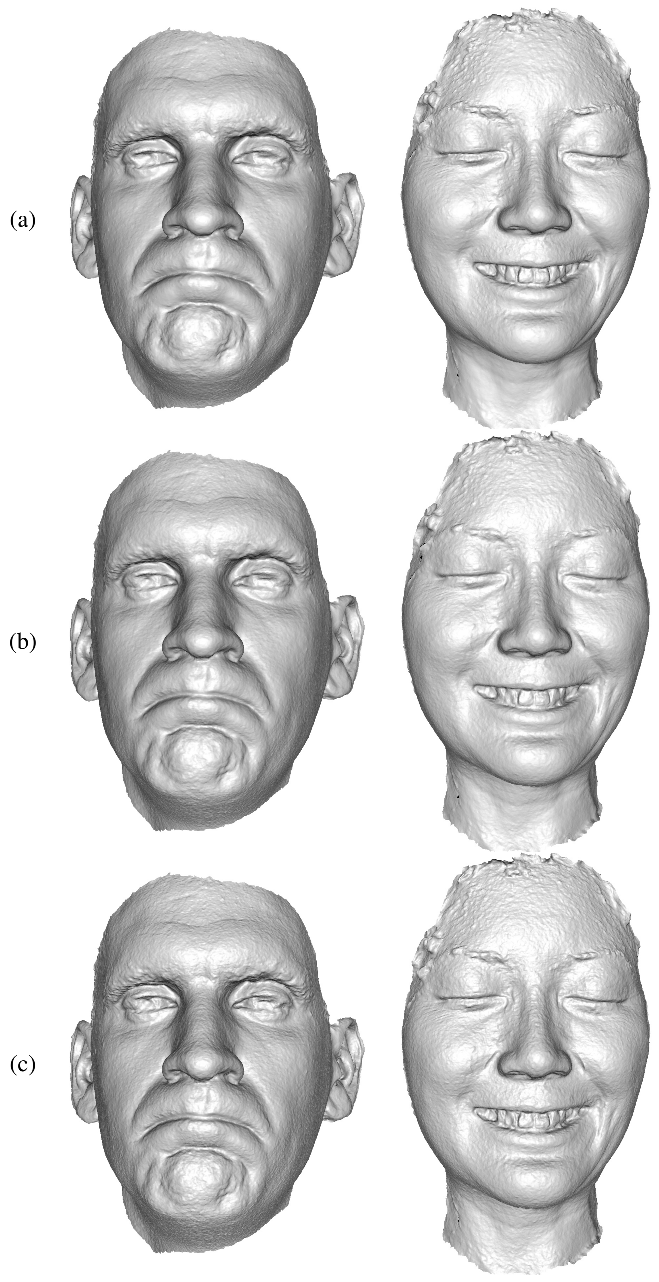

Figure 15: Two subjects reconstructed without facial hair, comparing ECC photoconsistency to NCC. (a) ECC. (b) NCC with more smoothing. (c) NCC with less smoothing.