Style Transfer for Headshot Portraits

YiChang Shih[1,2] Sylvain Paris[2] Connelly Barnes [2,3] William T. Freeman[1] Frédo Durand[1]

[1]MIT CSAIL [2]Adobe [3]University of Virginia

(a) ***Input****: a casual face photo* **(b)** ***Outputs****: new headshots with the styles transferred from the examples. The insets show the examples.*

Figure 1: *We transfer the styles from the example portraits in the insets in (b) to the input in (a). Our transfer technique is local and multi-scale, and tailored for headshot portraits. First, we establish a dense correspondence between the input and the example. Then, we match the local statistics in all different scales to create the output. Examples from left to right: image courtesy of Kelly Castro, Martin Schoeller, and Platon.*

Abstract

Headshot portraits are a popular subject in photography but to achieve a compelling visual style requires advanced skills that a casual photographer will not have. Further, algorithms that automate or assist the stylization of generic photographs do not perform well on headshots due to the feature-specific, local retouching that a professional photographer typically applies to generate such portraits. We introduce a technique to transfer the style of an example headshot photo onto a new one. This can allow one to easily reproduce the look of renowned artists. At the core of our approach is a new multiscale technique to robustly transfer the local statistics of an example portrait onto a new one. This technique matches properties such as the local contrast and the overall lighting direction while being tolerant to the unavoidable differences between the faces of two different people. Additionally, because artists sometimes produce entire headshot collections in a common style, we show how to automatically find a good example to use as a reference for a given portrait, enabling style transfer without the user having to search for a suitable example for each input. We demonstrate our approach on data taken in a controlled environment as well as on a large set of photos downloaded from the Internet. We show that we can successfully handle styles by a variety of different artists.

CR Categories: I.4.3 [Computing Methodologies]: Image Processing and Computer Vision—Enhancement

Keywords: Headshot portraits, style transfer, example-based enhancement

Links: DL PDF

1 Introduction

Headshot portraits are a popular subject in photography. Professional photographers spend a great amount of time and effort to edit headshot photos and achieve a compelling style. Different styles will elicit different moods. A high-contrast, black-and-white portrait may convey gravity while a bright, colorful portrait will evoke a lighter atmosphere. However, the editing process to create such renditions requires advanced skills because features such as the eyes, the eyebrows, the skin, the mouth, and the hair all require specific treatment. Further, the tolerance for errors is low, and one bad adjustment can quickly turn a great headshot into an uncanny image. To add to the difficulty, many compelling looks require maintaining a visually pleasing appearance while applying extreme adjustments. Producing such renditions requires advanced editing skills beyond the abilities of most casual photographers. This observation motivates our work: we introduce a technique to transfer the visual style of an example portrait made by an artist onto another headshot. Users provide an input portrait photo and an example stylized portrait, and our algorithm processes the input to give it same visual look as the example. The output headshot that we seek to achieve is the input subject, but as if taken under the same lighting and retouched in the same way as the example. We also support the case in which an artist has produced a collection of portraits in a consistent style. In this case, our algorithm automatically picks a suitable example among the collection, e.g., matching beardless examples to beardless inputs.

This enables the stylization of a large set of input faces without having to select an example for each one manually.

From a technical perspective, editing headshots is challenging because edits are made locally — hair does not receive the same treatment as skin, and even skin may be treated differently over the forehead, cheeks, and chin. Further, lighting is critical to the face's appearance: point light sources generate very different appearance from area lights and similarly for front versus side lighting. For these reasons, algorithms that automate the editing of generic photographs often perform poorly on headshots because they are global, or ignore the specificities of headshot retouching. For example, we show in the results section the limitations of the global style-transfer approach of Bae et al. [2006] when applied to headshots.

We address these challenges with an approach specific to faces. First, we precisely align the input and example faces using a three-step process. Then, motivated by the artists' use of brushes and filters with different radii to manipulate contrast in different scales, we introduce a new multiscale approach to transfer the local statistics of the example onto the input. Matching the local statistics over multiple scales enables the precise copying of critical style characteristics such as the skin texture, the hair rendition, and the local contrast of the facial features. All these elements exhibit sophisticated spatial frequency profiles, and we shall see that our multiscale algorithm performs better than single- and two-scale methods. We designed our algorithm to be tolerant to the differences that inevitably exist, even after alignment, between the input and example faces. Another important feature of the algorithm is its ability to exploit a mask to transfer only the face statistics while ignoring those of the background. We produce the final result by transferring the eye highlights and matching the example background. When a series of consistently stylized headshots is available, we can automatically estimate the success of this transfer procedure and select the highest ranked example, thereby automatically selecting a suitable reference portrait among the many available.

Contributions This paper introduces the following contributions:

▷ Given an input unprocessed headshot and a model headshot by an artist, we describe an automatic algorithm to transfer the visual style of the model onto the input.

▷ We introduce a multiscale technique to transfer the local statistics of an image. We explain how to focus the transfer on a region of interest and how to cope with outliers.

▷ We describe an automatic algorithm to select a suitable example among a collection of consistently stylized headshots.

2 Related Work

Global Transfer Transferring global statistics from one image to another can successfully mimic a visual look for cases such as still lifes and landscapes, e.g., [Reinhard et al. 2001; Pitié et al. 2005; Tai et al. 2005; Bae et al. 2006; Sunkavalli et al. 2010; HaCohen et al. 2011]. But, as discussed above, portraits can require a different treatment for each spatial region. That said, our approach shares some characteristics with those style-transfer algorithms. Like Reinhard et al. [2001], Pitié et al. [2005], and Tai et al. [2005], we transfer the color palette. Like Sunkavalli et al. [2010], we use a multiscale image decomposition [Burt and Adelson 1983; Oliva et al. 2006]. We rely on a dense correspondence between the input and example akin to HaCohen et al. [2011] and, like Bae et al. [2006], explicitly focus on photographic style. We transfer the image local contrast, as do Bae et al., but introduce a fully multiscale approach instead of using their two-scale method. For portrait stylization, this local and spatially varying approach matches the desired style much better.

Local Transfer Other authors have applied local stylistic changes in different contexts. Cohen-Or et al. [2006] locally change image colors to produce images with a more harmonious color palette, Wang et al. locally apply color schemes [2010] and transfer the look of specific cameras [2011], and Shih et al. [2013] locally remap image colors to render outdoor scenes at a new time of day. All these methods aim for a different application than ours.

Example-based Face Enhancement Joshi et al. [2010] and An and Pellacini [2010] successfully transfer color balance and overall exposure. Tong et al. [2007] and Guo et al. [2009] transfer make-up. Brand and Pletscher [2008] and Leyvand et al. [2008] improve face appearance. In comparison, we focus specifically on photographic style transfer, including aspects such as skin texture, local contrast, and light properties.

Face Synthesis Our work is related to face synthesis [Liu et al. 2007; Mohammed et al. 2009] in that we generate a portrait that can differ dramatically from the input. However, unlike Mohammed et al. [2009], we seek to retain the identity of the person shown in the input photo. Liu et al. [2007] do that, but consider the different problem of resolution enhancement.

Face Relighting Altering the illumination on a face is a common operation for face recognition and video editing, e.g., [Wen et al. 2003; Zhang et al. 2005; Peers et al. 2007]. In comparison, we focus on photographic style. While this may involve some illumination change, it is not a primary objective of our work and we do not claim a contribution in this area.

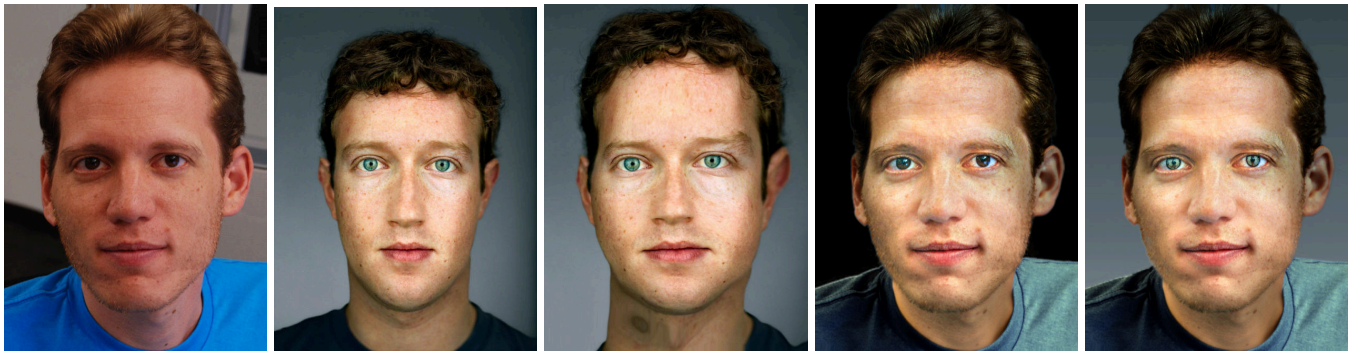
3 Multiscale Local Transfer

Our goal is to match the appearance of the input subject to the example. In this work, the styles that we target are typically achieved by local operations on image intensity, e.g., recolor and contrast, and some amount of illumination and defocus, but do not include changes of expression, pose, shape, perspective, or focal length.

Figure 2 shows the intermediate results of each step in our method. We start from an untouched input face photo, typically taken by an untrained user under arbitrary lighting conditions, and a stylized headshot as the example, typically taken by a professional under studio lighting and retouched. We assume that the input and example have approximately similar poses and facial expressions. We first establish a dense correspondence between the input and the model, that is, each input pixel is put in correspondence with a pixel of the model (§ 3.1). Then, we transfer the local statistics of model onto the input (§ 3.2) — this is the core of our approach. Finally, we transfer the eye highlights and the background (§ 3.3).

3.1 Dense Correspondence

To obtain correspondences between the input and reference images, we take a coarse-to-fine approach, using a series of off-the-shelf tools. We detect the facial landmarks using a template [Saragih et al. 2009]. This gives us 66 facial landmarks as well as a triangulated face template. First, we roughly align the eyes and mouth of the example with those of the input using an affine transform akin to Joshi et al. [2010]. Then, we morph the example to the input using the segments on the face template [Beier and Neely 1992]. This initial estimation often successfully aligns the eyes and mouth, but



(a) *Input* **(b)** *Example* **(c)** *Compute the dense correspondence between (a) and (b)* **(d)** *Locally match the power map between (a) and (b). This is the key of our work.* **(e)** *Transfer eye highlights and background*

Figure 2: *Overview of our approach: given an input (a) and an example (b), we start by finding the dense correspondence field between them. In (c), we visualize the correspondence by warping (b) to (a). Then we decompose (a) and (b) into Laplacian stacks, and match the local statistics in each scale. We aggregate the new stack to create (d). Finally, we transfer the eye highlights and the background (e).*

.

misses important edges such as the face contour and mouth. The final step is to refine the correspondence using SIFT Flow [Liu et al. 2011]. Figure 2 (b) and (c) shows an example headshot before and after alignment. The facial features — eyes, nose, mouth, hair — are put in correspondence with the input but the identity remains that of the example photo, i.e., the warped example is not the result that we seek. The next section explains how to transfer the local properties of the warped example while preserving the identity of the input.

3.2 Multiscale Transfer of Local Contrast

In this section, we seek to transfer the local contrast of the example onto the input. Our goal is to match the visual style of the example without changing the identity of the input subject. That is, we want the output to represent the same person as the input with the same pose and expression, but with the color and texture distribution and overall lighting matching the example. We perform this operation at multiple scales to deal with the wide range of appearances that a face exhibits, from the fine-grain skin texture to the larger signal variations induced by the eyes, lips, and nose. Further, working at multiple scales allows us to better capture the frequency profile of these elements, akin to the work of Heeger and Bergen [1995]. Our technique builds upon the notion of power maps [Malik and Perona 1990; Su et al. 2005; Li et al. 2005; Bae et al. 2006] to estimate the local energy in each image frequency subband. Similarly to Li et al. [2005], to prevent aliasing problems, we do not downsample the subbands. The rest of this section details our technique. For clarity's sake, we first assume grayscale images and that the region of interest is the entire image. We later explain how to adapt our algorithm to deal with colors and to use a mask.

Multiscale Decomposition As illustrated in Figure 3, the first step of our algorithm is to decompose the input and example images into multiscale Laplacian stacks. We describe the procedure for the input image I; the same procedure applies for the example image E. The construction uses a 2D normalized Gaussian kernel $G(\sigma)$ of standard deviation σ. Using $\otimes$ as the convolution operator, we define the level L_ℓ at scale $\ell \geq 0$ of the input Laplacian stack as:

$$L_\ell[I] = \begin{cases} I - I \otimes G(2) & \text{if } \ell = 0 \\ I \otimes G(2^\ell) - I \otimes G(2^{\ell+1}) & \text{if } \ell > 0 \end{cases} \quad (1)$$

and for a stack with $n \geq 0$ levels, we define the residual as:

$$R[I] = I \otimes G(2^n) \quad (2)$$

Local Energy Inspired by power maps [Malik and Perona 1990; Su et al. 2005; Li et al. 2005; Bae et al. 2006], we estimate the local energy S in each subband by the local average of the square of subband coefficients. Intuitively, this estimates how much the signal locally varies at a given scale. Concretely, since we do not downsample the Laplacian layers, we adapt the size over which we average the coefficients to match the scale of the processed subband. For the ℓ^{th} subband, this gives:

$$S_\ell[I] = L_\ell^2[I] \otimes G(2^{\ell+1}) \quad (3)$$

For the example image E, we account for the correspondence field that we have computed previously (§ 3.1). Using $W(\cdot)$ for the warping operator defined by this field, we compute:

$$\tilde{S}_\ell[E] = W(S_\ell[E]) \quad (4)$$

where we compute $S_\ell[E]$ with Equation 3. Estimating the energy before warping the data avoids potential perturbations due to distortion and resampling.

Robust Transfer Using these two estimates (Eq. 3 and 4), we modify the input subbands so that they get the same energy distribution as the example subbands. Letting O be the output image, we formulate a first version of our transfer operator as:

$$L_\ell[O] = L_\ell[I] \times \text{Gain} \quad (5a)$$

$$\text{with Gain} = \sqrt{\frac{\tilde{S}_\ell[E]}{S_\ell[I] + \epsilon}} \quad (5b)$$

where ϵ is a small number to avoid division by zero ($\epsilon = 0.01^2$, I is between [0,1]) and the square root compensates for the square used to define the energy in Equation 3. The gain maps (Eq. 5b) in Figure 3 show how they vary over space to capture local contrast. Intuitively, gain values below 1 mean a decrease of local contrast, and conversely, values greater than 1 mean an increase. While this version works well overall, it can introduce artifacts where I and E

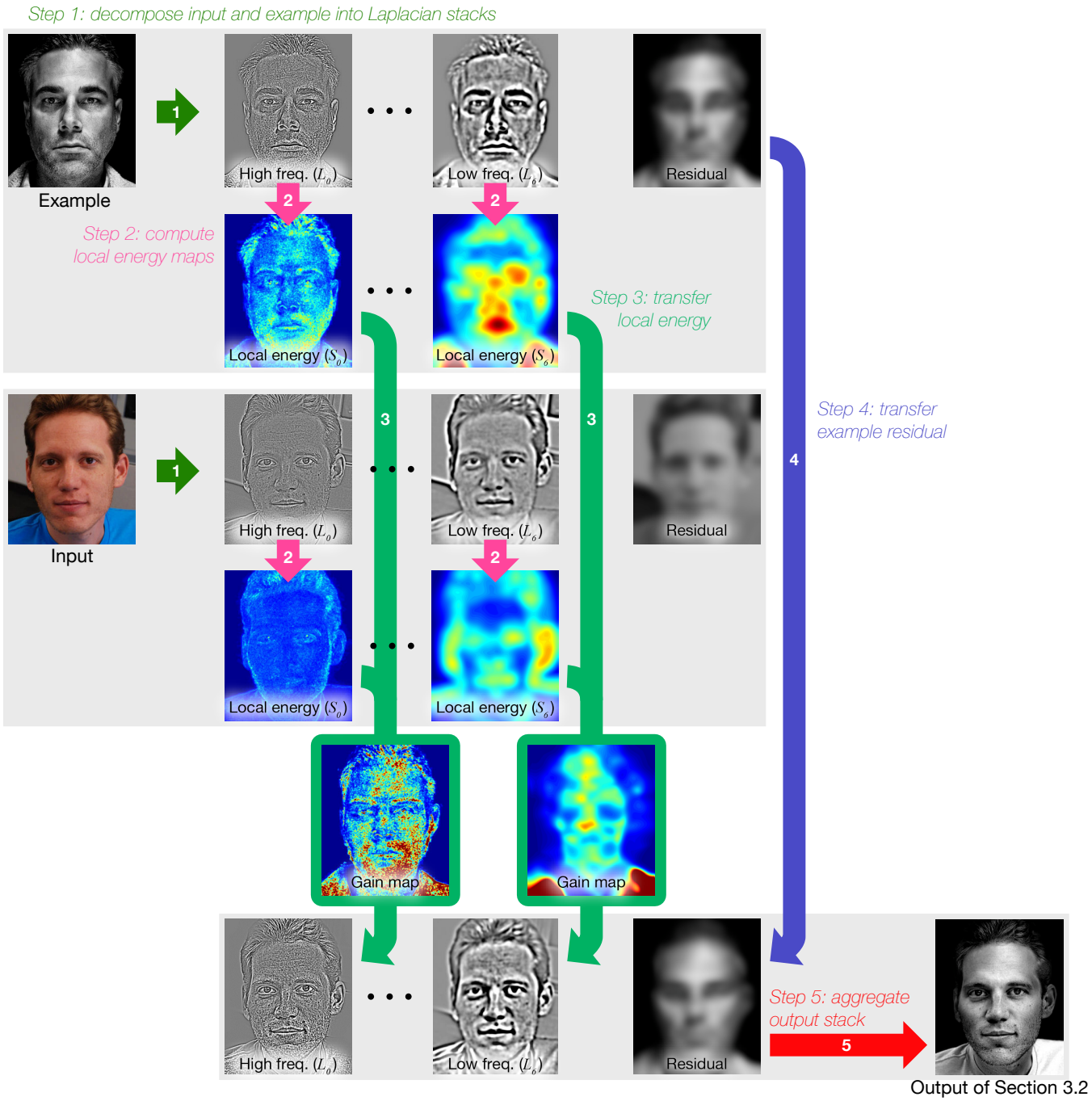


Figure 3: *Our transfer algorithm starts by (1) decomposing the input and example into Laplacian stacks. (2) In each band, we compute the local energy map by averaging the coefficients' L_2 norm in a neighborhood. (3) Using the energy maps, we match the local statistics to create a new stack, and (4) transfer the input residual to this new stack. (5) Finally, we aggregate the new stack to create the output. Gain maps capture spatially-variant local information over different scales. At the finer level, the gain map captures beard and skin. At the coarser level, it captures larger textures, e.g., eyes, nose, and some amount of highlight.*

mismatch. For instance, if the example has a mole and not the input, the gain map (Eq. 5b) will spike at the mole location, generating a mole in the output that does not exist on the input. Another common case is an input with glasses and an example without. The gain map (Eq. 5b) has low values along the glasses, which produces unsightly phantom glasses in the output. Figure 4 illustrates these two cases. These problems correspond to outliers in the gain map. We address this issue by defining a robust gain map that clamps high and low values, and smooths the gains:

$$\text{RobustGain} = \max(\min(\text{Gain}, \theta_h), \theta_l) \otimes G(\beta\, 2^\ell) \qquad (6)$$

We use $\theta_h = 2.8$, $\theta_l = 0.9$, $\beta = 3$, and $n = 6$ for the Laplacian stack in all our examples. Finally, for the output residual, we directly copy the warped example residual, i.e.: $R[O] = W(R[E])$. This

step captures the overall lighting configuration on the face as shown by Wen et al. [2003]

Discussion The choice of the neighborhood size that we use to estimate the local energy is critical. Neighborhoods that are too large make the transfer almost global and poorly reproduce the desired style. Neighborhoods that are too small make the transfer similar to a direct copy of the example values — while this faithfully transfers the example style, it also copies the identity of the example subject, which is not acceptable. The size that we use in Equation 3 strikes a balance and transfers the example style while preserving the input identity. Figure 5 illustrates this trade-off.

Dealing with Colors We work in the CIE-Lab color space because it approximates human perception, and process each channel independently using the algorithm that we just described. We use the fact that the human visual system is less sensitive to chrominance variations to not process the a and b high frequencies; in practice, we skip the first three subbands.

Using a Mask We extend our transfer algorithm to use a mask defining a region of interest. Intuitively, we truncate the Gaussian convolutions so that they only consider values within the mask. In practice, we replace each Gaussian convolution (Eq. 1, 2, 3, and 6) as follows:

$$\text{Image} \otimes G \rightsquigarrow \frac{(\text{Image} \times \text{Mask}) \otimes G}{\text{Mask} \otimes G} \tag{7}$$

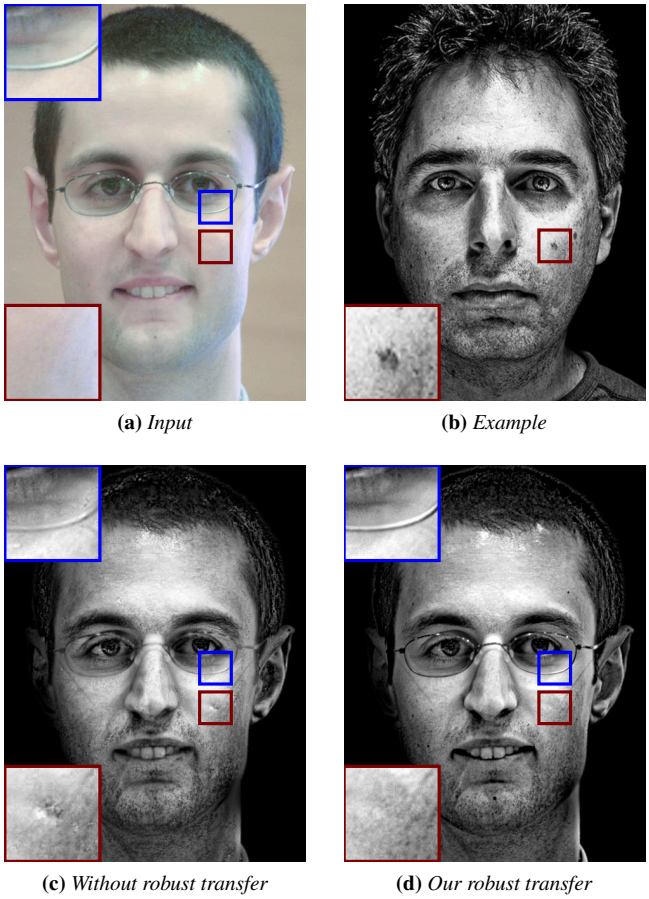
This operation can also be interpreted as convolving premultiplied alphas and unpremultiplying the result.

In practice, we run GrabCut [Rother et al. 2004] initialized with a face detection result to find a binary mask that we refine using the Matting Laplacian [Levin et al. 2008]. As shown in Figure 6, without a mask, the large differences that may exist in the background region perturb the transfer algorithm near the face contour, and using a mask solves this problem.

3.3 Additional Postprocessing

The multiscale transfer algorithm that we have just described matches the local contrast, the color distribution, and the overall lighting direction of the example headshot. In this section, we add two additional effects: matching the eye highlights and the background.

Eye Highlights The reflection of the strobes in the eyes is often an important factor of a headshot style. To transfer the eye highlights from the example onto the input, we separate the specular reflection from the example eyeball and copy that onto the input's eyes. On the example, we first locate the iris using circular arc detection around the position given by the face template [Daugman 1993]. Then, we create an approximate segmentation in to iris, highlight, and pupil by running a k-means algorithm on the pixel colors with $k = 3$. We refine the reflection mask using alpha matting [Levin et al. 2008] (Fig. 7). On the input, we detect the existing highlights as the brightest pixels in the iris region. In practice, we use a threshold of 60 on the L channel of CIE-Lab colorspace. Then, we erase the detected pixels and fill in the hole using inpainting. We used the `griddata` Matlab function that was sufficient in our experiments. One could use a more sophisticated algorithm, e.g., [Barnes et al. 2009], to further improve the results if needed. Finally, we compose the example highlights on top of the input eyes. We center them using the pupils as reference, and scale them in proportion of the iris radii.



(a) *Input* **(b)** *Example*

(c) *Without robust transfer* **(d)** *Our robust transfer*

Figure 4: *Images (a) and (b) do not match on the glasses and the moles (blue and red boxes). Without robust transfer, a simple gain map leads to over-smoothing on the glass frames and artifacts on the skin. Our robust transfer in (d) avoids the problem.*

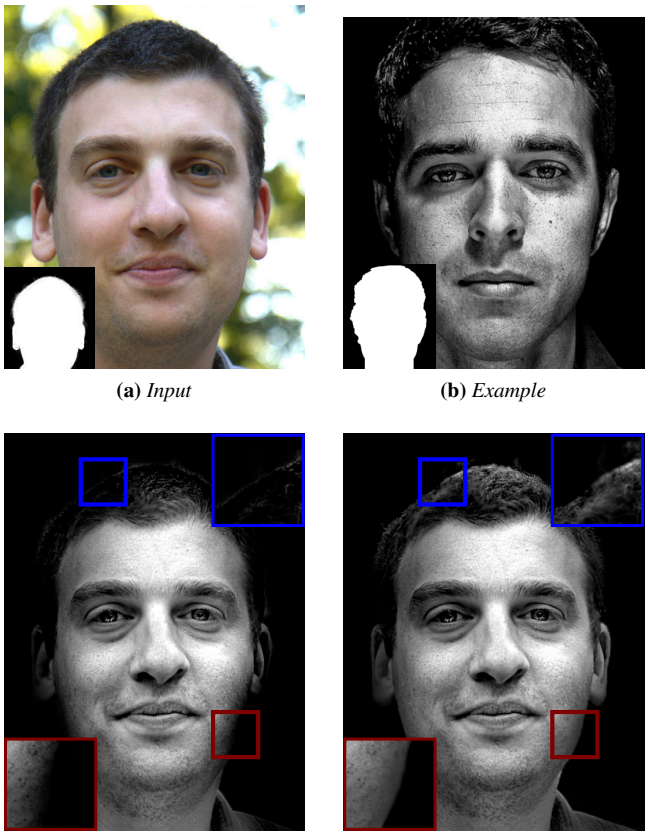
Background The background also contributes to the mood of a portrait [Phillips 2004, § 2]. For this purpose, we directly replace the input background with the example background. We use the previously computed masks to extract the example background and replace the input background with it. If needed, we extrapolate the missing data using inpainting — we use the `griddata` Matlab function in practice. Figure 2e illustrates this process.

3.4 Automatic Selection of the Example in a Collection

Many photographers produce collections of headshots with a consistent and recognizable style. For such cases, we provide an algorithm for the automatic selection of the best example for a given input. A good candidate has similar facial characteristics to the input, such as both having beards. Inspired by research in face retrieval [Tuzel et al. 2006; Ahonen et al. 2006], we use the local energy S in Equation 3 as the face feature vector, and look for the candidate with the closest distance to the input in feature space. We concatenate S_ℓ over all scales to get the feature vector representing a face image, and use the normalized cross correlation between the two feature vectors as the similarity function. We found this choice more robust to image retouching than the L_2 distance. For computational efficiency, we do not warp the example image in the searching step.

(a) *Input* **(b)** *Example* **(c)** *Neighborhood too small* **(d)** *Neighborhood too large* **(e)** *Our choice*

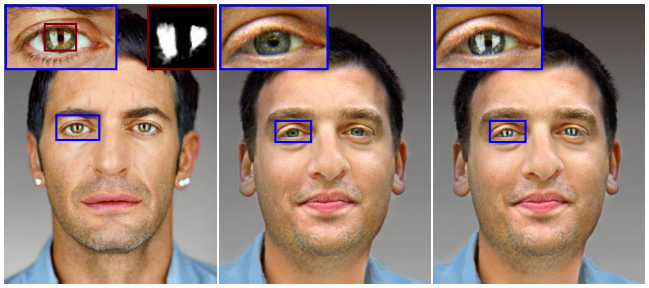
Figure 5: *We tested a few different neighborhood sizes for computing local energy. Result (c) uses a neighborhood size that is too small, so the result's identity does not look like input subject. Result (d) uses a neighborhood that is too large, so the result fails to capture the local contrast. (e) Our choice in Eq. 3 produces a successful transfer.*



(a) *Input* **(b)** *Example*

(c) *Without adapting to face mask, the face contour disappeared* **(d)** *Our Laplacian that is adapted to face mask*

Figure 6: *(a) shows an input where the face and background have very different colors. Without using a face mask, the hair and face contour disappear in the background, as shown in the blue and red insets in (c). We restrict the Laplacian to use the pixels within the face region, defined by the masks in the insets in (a) and (b). The result in (d) shows the hair and face contour are better preserved.*



(a) *Input* **(b)** *Without eye highlights* **(c)** *Adding eye highlights*

Figure 7: *Taking the input in Fig. 6, we transfer the eye highlight from the example (a) by alpha matting. We show the extracted alpha map in the red box in (a). (b) and (c) show the effect of adding eye highlights.*

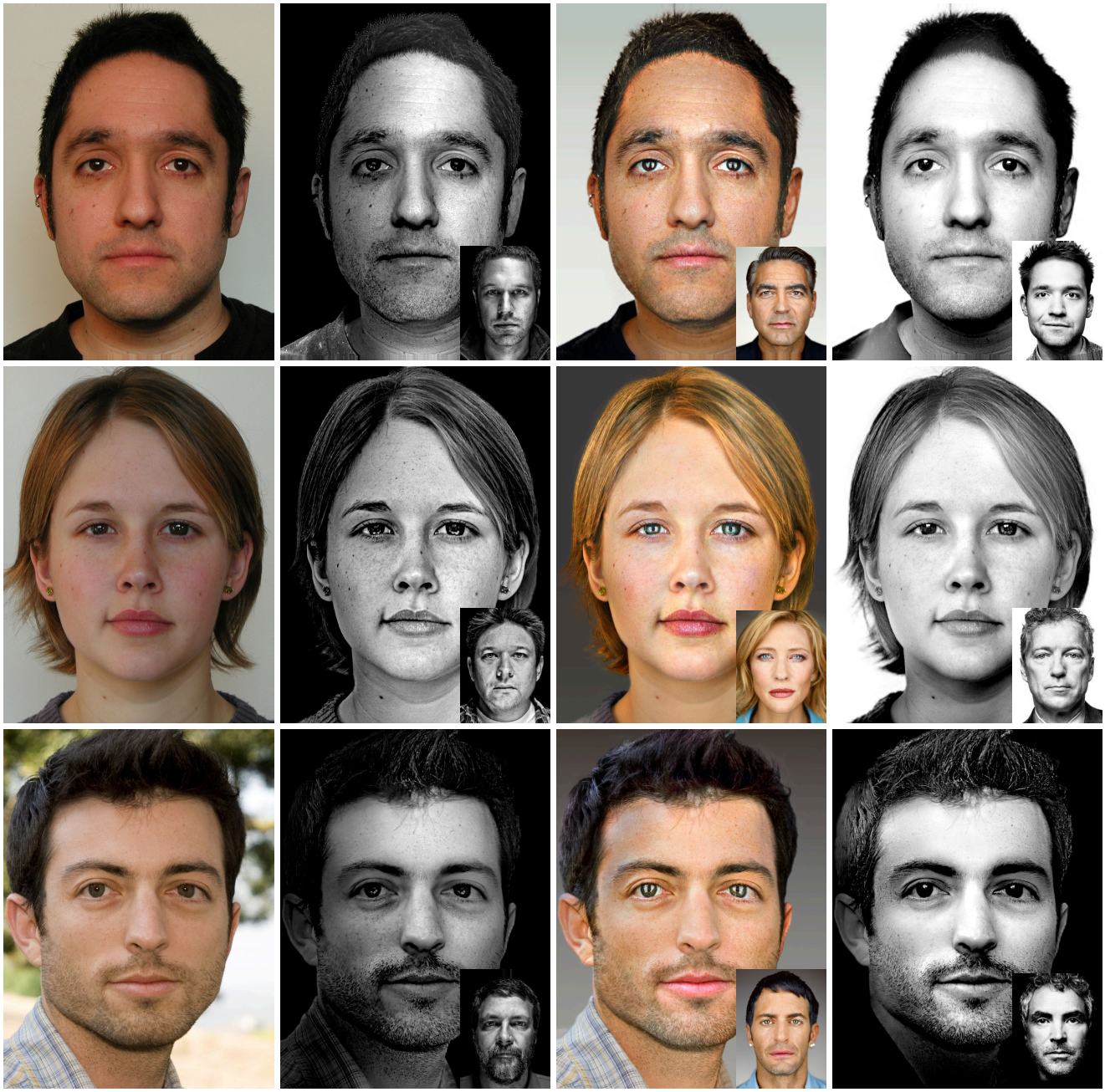
For our experiments, we use a portrait collections database by 3 different photographers who are unaffiliated to us. Each collection has on the order of 50–75 example headshots. In all our results, we use the example selected automatically unless otherwise specified.

4 Results

Figure 8 shows our style transfer for compelling styles created by three different photographers. The examples are selected by our automatic algorithm as the best candidate for each input. We selected these three styles because they are widely different from each other, with black-and-white and colors, low key (i.e., dark) and high key (i.e., bright), soft and detailed. Further, they also differ significantly from the inputs. Our method successfully transfers the tone and details, for input photos under indoor and outdoor lighting conditions, and subjects of both genders.

Figure 9 shows how different styles generate different gain maps. The low-key and highly contrasted style emphasizes the details on the entire face. The warm color and soft lighting style preserves most of the details, and slightly emphasizes the forehead. The high contrast black-and-white style sharpens the face borderline but smoothes the cheeks.

To verify the robustness of our method, we also tested it on 94 photos collected from the photography website Flickr. The results are shown in the supplemental material. To collect the dataset, we searched for photos with the keywords "headshot" and "portrait," then automat-



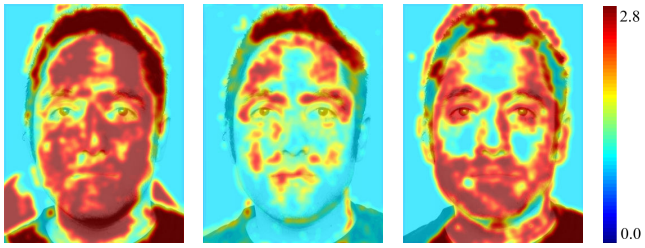
Input

Outputs of our method, using examples in the inset at bottom right

Figure 8: *We transfer the examples in the insets to the inputs in column (a). The examples in each column in (b) are from one photographer. From left to right, the three styles are low-key and high contrast, warm and soft lighting, and high-contrast black-and-white. We test on indoor male, female, and outdoor subject.*

ically filtered out profile faces using a face detector, and removed faces whose eye distance is below 150 pixels to ensure sufficient resolution. The dataset contains a large variety of casual headshot photos with various facial features such as beard, accessories, and glasses, as well as people of different gender, age, skin color, and facial expression. The photos are taken under a variety of uncontrolled lighting conditions, both indoors and outdoors. This dataset is challenging because some photos are noisy due to low-light conditions, and the background can be cluttered which makes matting hard. The full results on this dataset are can be found at `people.csail.mit.edu/yichangshih/portrait_web/`

Figure 10 shows results with diverse success levels. The output quality depends on the input data; our method works best on well-lit and in-focus photos in which facial details such as skin pores are visible. Figure 11 shows that our method captures some amount of illumination change when the lighting setups in the input and example are different. However, our method is not specifically designed for face relighting and we claim no contribution in this field.



(a) *Gain map at $\ell = 2$ for the low-key style, (row 1, col 2) in Figure 8*

(b) *color style, (row 1, col 3)*

(c) *nearly all-black-and-white style, (row 1, col 4)*

Figure 9: *We overlay the gain maps of the first row in Figure 8 on the input to show that the three styles manipulate the details in different ways: (a) emphasizes the details on the entire face, (b) emphasizes the details on the forehead and near the center, and (c) emphasizes the beard but smoothes the cheeks.*

Comparison to the reference image Figure 12 compares our style transfer to a reference portrait of the same subject made by the photographer who created the example. Even though the example and the reference are different subjects, our method successfully transfers a look that is visually similar to the reference, including mimicking the highlights, shadows, and the high contrast style. In the supplemental material, we provide the result of using the reference in Figure 12 as the example. This is to test the ideal scenario that the database is sufficiently large so that we can find an example almost identical to the input.

Automatic example selection Figure 13 shows the transfer using the top three examples selected by our automatic algorithm. Given an input with beard and sideburns, our algorithm successfully retrieves the examples that match the beard on the input. Further, while the transferred results vary because of using different examples, they are nonetheless all plausible and have similar tone and details. In the supplemental material, we provide the transferred results using the top four candidates on all the three styles.

Global dynamic range In a few cases, our local statistic matching does not reproduce the example global dynamic range, as shown in Figure 14. A naive solution is to transfer the histogram from the example but this may lose facial details when the example has wide dynamic range with nearly saturated regions. Instead, we suggest to balance the local details and global range by averaging the local statistic matching result with and without histogram transfer applied as a post-process.

Manual correction Figure 15 shows examples of manual corrections applied to correct failures of the automatic method. These are the only results in the paper with manual intervention, all the others are generated automatically. Our correction includes correcting correspondence, face mask, and eye locations. Out of 94 results in Flickr dataset, we corrected the correspondence 5 times, the face mask twice, and eye locations 4 times.

Running times With an unoptimized MATLAB implementation, the main algorithm of our style transfer takes about 12 seconds: 7 seconds in dense matching and 5 seconds in the multi-scale local transfer. The images we test are about 1300×900 pixels, with about 300 pixels between the two eyes.

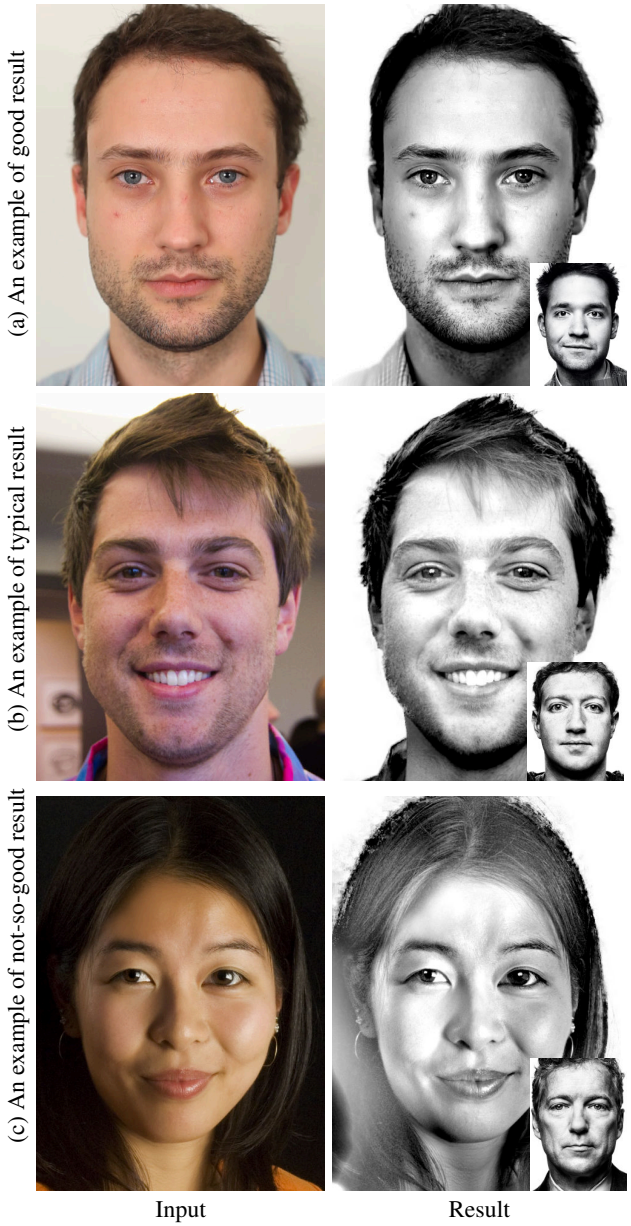


Figure 10: *We show examples of (a) good, (b) typical and (c) poor results. (a) Our method achieves good results when input is clean and uniformly lit. (b) A typical input usually contains some amount of noise, which remains on the output. (c) In this input, the hair textures almost disappear in the background, which results in poor performance on the output.*

4.1 Comparison with Related Work

Figure 16 compares our method with related work in multiscale transfer and color transfer. For Sunkavalli et al. [2010], we used the code provided by the authors. They adapted the code to style transfer by using multiscale histogram matching and disabling the Poisson editing. We also show comparisons on global color transfer based on histogram and linear color mapping [Reinhard et al. 2001; Pitié et al. 2005]. Our result captures details more faithfully because our method is local.

(a) *Input* **(b)** *Example* **(c)** *Our result*

Figure 11: *Our method captures some amount of lighting change in the case that the lighting in the example (b) is different from the input (a).*

Figure 17 shows comparisons to Bae et al. [2006] that is designed for black-and-white images. We used the implementation from the author. We also compare to Sunkavalli et al. because their method also works in a multi-scale manner. For fair comparisons, we adapted their methods to incorporate the face mask by replacing the input background with the example background. Note that without modification, any global method fails in this case, because the input has brighter background than face, but vice versa on the example. In the supplemental material, we provide black-and-white comparisons to other related methods.

We also attempted to compare with HaCohen et al. [2011], because their method uses local correspondences. However, they solve a different problem of finding repeated image contents such as the same person in a different pose. Our goal is to match across different persons and styles, so their matching often does not work for our style transfer. In particular, their implementation reports empty matches on the face regions of the three styles in Figure 8. In the supplemental material, we compare to their result using an example with similar appearance, so that they can find good matches.

In all fairness, all these related methods are designed for general image content, while our method is tailored for face portraits. Our advantage comes from the dense local matching, which captures the spatially varying details and lighting on the face. Some of the related methods can be adapted for our problem by restricting the transfer within the face region. In the supplemental, we provide comparisons using the adapted methods, as well as the comparisons on all three styles used in Figure 8.

4.2 Extensions

4.2.1 Style transfer on video

Our method can be extended to videos of frontal faces with moderate motion, such as videos of news anchors or public speeches. Independently transferring the example to each frame results in flickering due to lack of temporal coherence. This is because the dense matching is often unreliable when the facial expressions in the example and video frame are very different. To ensure temporal coherence, we avoid directly computing a dense matching from each frame to the example. Instead, we leverage the optical flow [Brox et al. 2004] within the video. First, we choose the exemplar that best matches the first video frame, using the automatic selection described in § 3.4. Then among all the frames in the video, we pick the candidate that best matches this exemplar, using the same automatic selection. Next, for each frame, we compute the correspondence to the best candidate frame by aggregating the optical flow between adjacent frames. Finally, we transfer the style to the best candidate, and propagate the style representation, i.e. multi-band gains, to the rest of the frames by using the correspondences to the best candidate frame.

Figure 18 shows that we successfully transfer the style to the input video, even with the frames of very different facial expressions. Our video result in the accompanying supplemental video shows good temporal coherence in the presence of extreme facial expressions.

Facial makeup transfer Figure 20 shows that our method can transfer facial makeup including the skin foundation, lip color and eye shadow. In the original method, the green color on the eye shadow is bled to the sclera (the white region of the eye). We fix this by automatically replacing the transferred output with the original sclera. The sclera is segmented by GrabCut around the eye region given by the face detector. In the supplemental material, we show a comparison with the state-of-the-art works [Tong et al. 2007; Guo and Sim 2009].

5 Discussion and Conclusion

The main novelty of our paper is a style transfer algorithm that is local and multiscale. Compared to generic style transfer, our approach is tailored for headshot portraits. First, it is local to capture spatially-variant image processing typical in portrait editing. Second, it is multiscale to handle facial textures in different scales. We validate the method using a large dataset of images from the Internet, and extend the method to videos of frontal faces.

Limitation While our method works on the bulk of the inputs that we collected online, we found the result quality is often limited by the quality of the matting mask. Also, our method may magnify the input noise.

It is important to select an example that matches well. Figure 19 shows that matching people of different skin color creates an unnatural look. In general, we require the input and the example to have similar facial attributes, e.g., beard, skin color, age, and hair style. Further, our method cannot remove hard shadows, nor can we create them from the example. In some rare cases, part of the identity of the example may be transferred on the input and causes artifacts. We also tested on profile headshots, but they failed because the face detector is unable to locate the landmarks. Styles of non-photorealistic rendering are beyond our scope. For example, cartoon portraits or paintings.

In some cases, the highlight transfer may fail because the input and example have very different eye color. Disabling eye highlight transfer is better for these cases.

Future work We are interested in style transfer from multiple examples. For instance, using different face regions from different people to better match the input face. This perhaps can increase the effective database size, by allowing for multiple matches in cases where there is no single good match.

Acknowledgements

We thank photographers Kelly Castro, Martin Schoeller, and Platon for allowing us to use their photographs in the paper. We also thank Kelly Castro for discussing with us how he works and for his feedback, Michael Gharbi and Krzysztof Templin for being our portrait models. We acknowledge the funding from Quanta Computer and Adobe.

(a) *Input* **(b)** *Example* **(c)** *Our result* **(d)** *Reference*

Figure 12: *We compare the transfer result in (c) to a reference portrait (d) made by the same photographer who created the example in (b). While our transfer is not exactly identical, it looks visually similar.*

(a) *Input* **(b)** *Transferred outputs using the top three retrieved examples, shown in the insets.*

Figure 13: *We use our automatic example selection algorithm to retrieve the top three examples and show the transferred results . All of our examples correctly match the beard on the input. Even with the variation within the three results, the transferred results are all plausible and have similar tone and details.*

References

AHONEN, T., HADID, A., AND PIETIKAINEN, M. 2006. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*.

AN, X., AND PELLACINI, F. 2010. User-controllable color transfer. In *Computer Graphics Forum*, vol. 29, 263–271.

BAE, S., PARIS, S., AND DURAND, F. 2006. Two-scale tone management for photographic look. In *ACM Trans. Graphics*.

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graphics*.

BEIER, T., AND NEELY, S. 1992. Feature-based image metamorphosis. In *ACM Trans. Graphics*, vol. 26.

BRAND, M., AND PLETSCHER, P. 2008. A conditional random field for automatic photo editing. In *IEEE Conf. Computer Vision and Pattern Recognition*.

BROX, T., BRUHN, A., PAPENBERG, N., AND WEICKERT, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*. 25–36.

BURT, P., AND ADELSON, E. 1983. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications 31*, 4, 532–540.

COHEN-OR, D., SORKINE, O., GAL, R., LEYVAND, T., AND XU, Y.-Q. 2006. Color harmonization. *ACM Trans. Graphics 25*.

DAUGMAN, J. G. 1993. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Analysis and Machine Intelligence 15*, 11, 1148–1161.

GUO, D., AND SIM, T. 2009. Digital face makeup by example. In *IEEE Conf. Computer Vision and Pattern Recognition*.

HACOHEN, Y., SHECHTMAN, E., GOLDMAN, D. B., AND LISCHINSKI, D. 2011. Non-rigid dense correspondence with applications for image enhancement. In *ACM Trans. Graphics*, vol. 30.

(a) *Input* **(b)** *Example* **(c)** *Only local matching* **(d)** *Transferring histogram from (b) to (c)* **(e)** *Averaging (c) and (d)*

Figure 14: *In a few cases, our local matching (c) does not match the global dynamic range of the example (b). (d) Transferring the histogram from (b) to (c) may lose important facial details, such as pores on the skin. (e) In practice, we suggest to balance the local details and global range by averaging (c) and (d).*

HEEGER, D. J., AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *ACM Trans. Graphics*, ACM, 229–238.

JOSHI, N., MATUSIK, W., ADELSON, E. H., AND KRIEGMAN, D. J. 2010. Personal photo enhancement using example images. *ACM Transaction on Graphics (TOG) 29*, 2, 1–15.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2008. A closed-form solution to natural image matting. *IEEE Trans. Pattern Analysis and Machine Intelligence 30*, 2, 228–242.

LEYVAND, T., COHEN-OR, D., DROR, G., AND LISCHINSKI, D. 2008. Data-driven enhancement of facial attractiveness. In *ACM Transactions on Graphics (TOG)*, vol. 27, ACM, 38.

LI, Y., SHARAN, L., AND ADELSON, E. H. 2005. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. Graphics 24*.

LIU, C., SHUM, H.-Y., AND FREEMAN, W. T. 2007. Face hallucination: Theory and practice. *International Journal of Computer Vision 75*, 1, 115–134.

LIU, C., YUEN, J., AND TORRALBA, A. 2011. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 5.

MALIK, J., AND PERONA, P. 1990. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A 7*.

MOHAMMED, U., PRINCE, S. J., AND KAUTZ, J. 2009. Visiolization: generating novel facial images. In *ACM Transactions on Graphics (TOG)*, vol. 28, 57.

NARS, F. 2004. Makeup your mind. *PowerHouse Books*.

OLIVA, A., TORRALBA, A., AND SCHYNS, P. G. 2006. Hybrid images. In *ACM Transactions on Graphics (TOG)*, vol. 25, 527–532.

PEERS, P., TAMURA, N., MATUSIK, W., AND DEBEVEC, P. 2007. Post-production facial performance relighting using reflectance transfer. *ACM Transactions on Graphics (TOG) 26*, 3, 52.

PHILLIPS, N. 2004. Lighting techniques for low key portrait photography. *Amherst Media*, 12–16.

PITIÉ, F., KOKARAM, A. C., AND DAHYOT, R. 2005. N-dimensional probability density function transfer and its application to color transfer. In *IEEE Conference on Computer Vision*.

REINHARD, E., ADHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Computer Graphics and Applications 21*, 5, 34–41.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, vol. 23, 309–314.

SARAGIH, J. M., LUCEY, S., AND COHN, J. F. 2009. Face alignment through subspace constrained mean-shifts. In *IEEE Conference on Computer Vision*, 1034–1041.

SHIH, Y., PARIS, S., DURAND, F., AND FREEMAN, W. T. 2013. Data-driven hallucination for different times of day from a single outdoor photo. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*.

SU, S., DURAND, F., AND AGRAWALA, M. 2005. De-emphasis of distracting image regions using texture power maps. In *Proc. of ICCV Workshop on Texture Analysis and Synthesis*.

SUNKAVALLI, K., JOHNSON, M. K., MATUSIK, W., AND PFISTER, H. 2010. Multi-scale image harmonization. *ACM Trans. Graphics 29*, 4, 125.

TAI, Y.-W., JIA, J., AND TANG, C.-K. 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1.

TONG, W.-S., TANG, C.-K., BROWN, M. S., AND XU, Y.-Q. 2007. Example-based cosmetic transfer. In *IEEE Pacific Graphics*.

TUZEL, O., PORIKLI, F., AND MEER, P. 2006. Region covariance: A fast descriptor for detection and classification. In *European Conference on Computer Vision*. 589–600.

WANG, B., YU, Y., WONG, T., CHEN, C., AND XU, Y. 2010. Data-driven image color theme enhancement. In *ACM Transactions on Graphics*, vol. 29, 146.

WANG, B., YU, Y., AND XU, Y.-Q. 2011. Example-based image color and tone style enhancement. *ACM Transactions on Graphics 30*, 4.

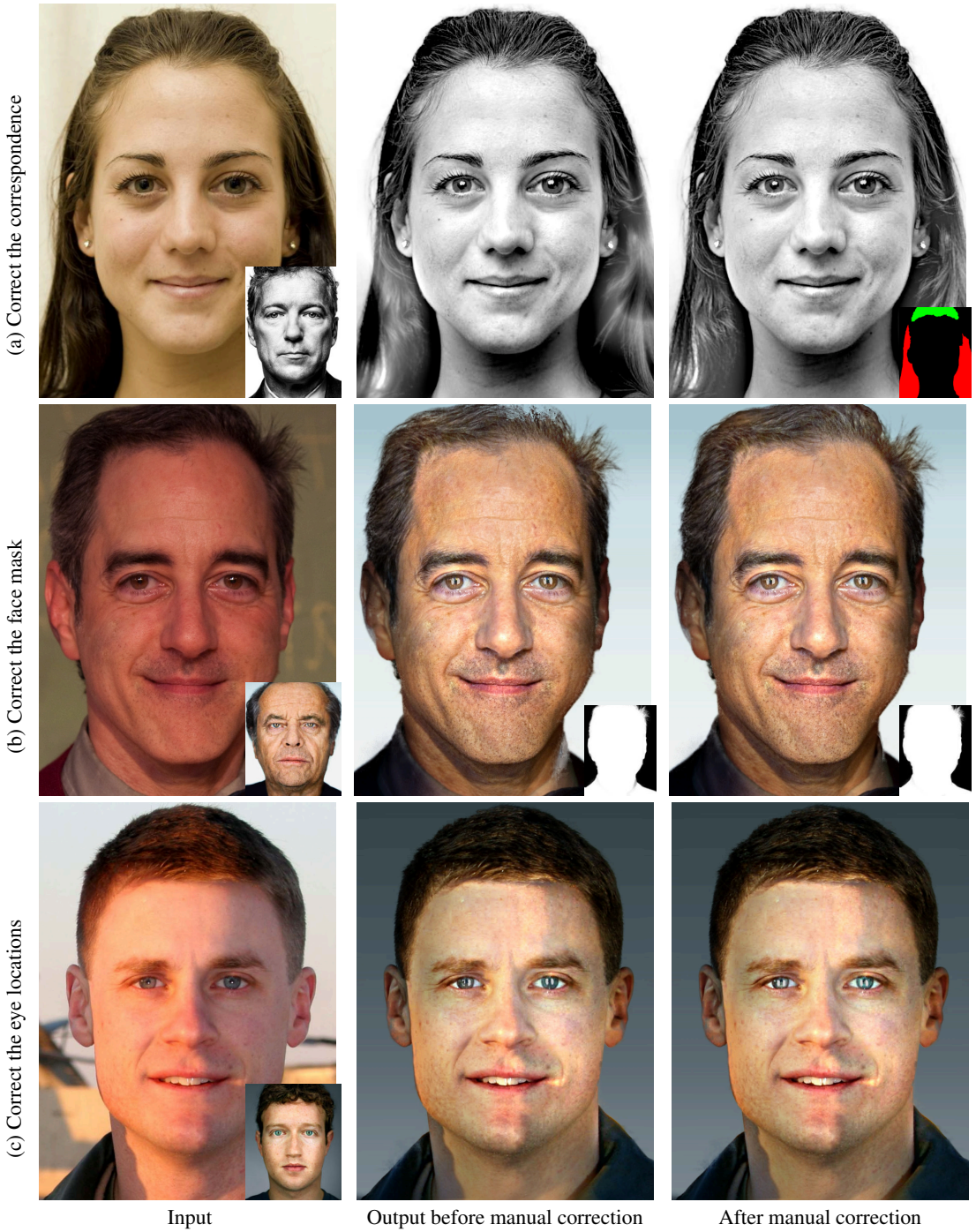
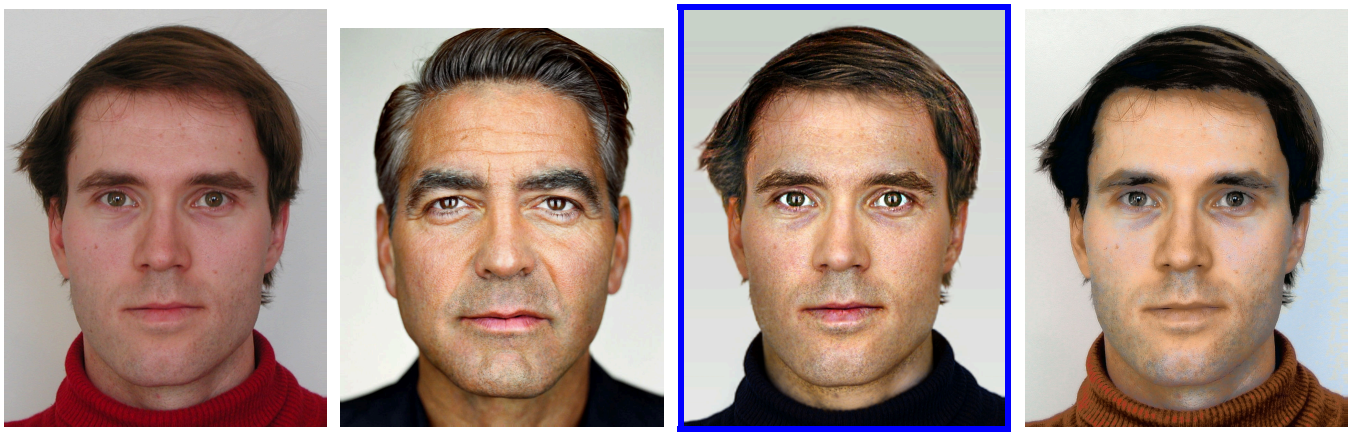


Figure 15: *We propose manual corrections to fix the rare failure cases of the automatic method. (a) The mismatching between the input hair and the example (in the bottom right inset) results in artifacts on the output. We correct the correspondence through a user-provided map shown in the inset in the output. This map constrains the gain on the red regions to be the same as the green region. (b) We correct the hair on the top by correcting the face mask. The automatic one and the corrected one are shown in the insets of middle and right column. (c) We correct the right eye location for highlight transfer.*



(a) *Input* **(b)** *Example* **(c)** *Our result* **(d)** *Sunkavalli et al. 2010*

(e) *Histogram transfer on RGB channel separately* **(f)** *Reinhard et al. 2001* **(g)** *Pitié et al. 2007* **(h)** *Photoshop Match Color*

Figure 16: *We compare to related methods on color transfer and multi-scale transfer. Our result is closer to the example. The readers are encouraged to zoom in to see the details. Because the backgrounds are of similar color we did not adapt related work here to use the face mask.*

(a) *Example* **(b)** *Our result* **(c)** *Bae et al. 2006* **(d)** *Sunkavalli et al. 2011*

Figure 17: *We compare with Bae et al. that works on tonal (black-and-white) transfer, as well as the multi-scale transfer of Sunkavalli et al. 2011]. These methods have been adapted to use the face mask because the input and example have different background colors.*

(a) *Input frames* **(b)** *Example*

(c) *Output frames*

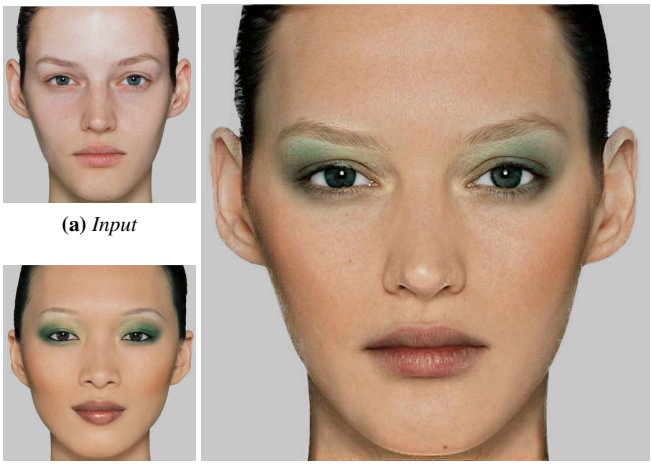
Figure 18: *We show style transfer on an input sequence in (a), using the example in (b). Our results in (c) show that we can handle frames with very different facial expressions. Please see the accompanying supplemental video for more results.*

(a) *Input* **(b)** *Failed output*

Figure 19: *A failure case: matching a white male to an African male in the inset creates an unrealistic result.*

WEN, Z., LIU, Z., AND HUANG, T. S. 2003. Face relighting with radiance environment maps. In *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, II–158.

ZHANG, L., WANG, S., AND SAMARAS, D. 2005. Face synthesis and recognition from a single image under arbitrary unknown lighting using a spherical harmonic basis morphable model. In *IEEE Conf. Computer Vision and Pattern Recognition*.



(a) *Input*

(b) *Example* **(c)** *Our makeup transfer result*

Figure 20: *We extend our method to makeup transfer with minor modification. We transfer the example makeup in (b), taken from a professional makeup book [Nars 2004]. The result in (c) shows that skin foundation, lip color and eye shadow are successfully transferred to the input (a).*