Style Transfer for Headshot Portraits

Vishal Batchu, Sri Aurobindo Munagala

November 16 2016

1 Abstract

The aim of the project is to transfer the style of a professionally taken headshot portrait to that of an amateur one. Headshot portraits, professionally taken, carry visual styles that require advanced photography skills. In this project, we attempt to try and transfer style from such artistically styled headshots, to amateur photos taken. Essentially, given the Input image and an Example image (from which the style should be transferred), local statistics of the Example image are propagated to the Input image.

Figure 1: Illumination, texture and facial feature matching and transfer from input example image to input content image.

2 Introduction

Headshot portraits are commonly taken in professional settings, and a lot of time and effort is spent in styling these photographs, through various editing methods. Each feature - eyes, skin, eyebrows, mouth, hair, all require specific treatment. This project uses a combination of various techniques to style an image using other, professionally taken styled portraits. Users can input an input headshot photo and an example photo, and the styled input is returned - comprising of the same lighting, retouches, etc.

Figure 2: An example for photographic illumination transfer

We have a four-part algorithm. In the first part, we detect facial feature points on the Input and Example images, and morph the Example to the Input. In the second, we segment both the Input and morphed Example images, from their respective backgrounds, and extrapolate the background of the morphed Example to find background color throughout the image, to blend in to the final image later on. In the third part, we do a multi-scale transfer of local statistics, using pyramid-based energy transfer on the foregrounds. In the final part, we transfer the background we computed earlier, to the output image, to get the final image. We then perform additional post-processing, applying a high-boost filter to boost individual details, and bilateral filtering for smoothing the image.

3 Algorithm

3.1 Facial Feature Point Detection

We used code implemented from the paper Saragih et al.[2009], to find 66 facial feature points on the faces of the input and example. We mirror the facial points corresponding to the jawline, and compress them, to approximately find forehead points. This heuristic works in most cases since a human face generally satisfies this property. Portraits where the person is not facing the camera directly but at an angle also work since the same heuristic works there as well. We also add feature points along the borders of the image so the entire image stays intact after morphing.

Figure 3: Points 1

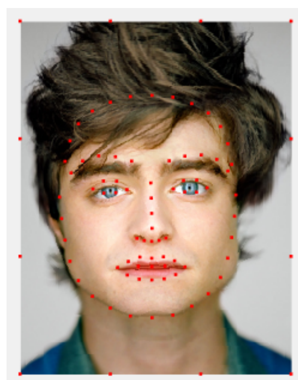

Figure 4: Points 2

Figure 5: Points 3

3.2 Morphing using Delaunay Triangulation

Using these sets of points, we find the Delaunay triangulation for each, and use a geometric morphing algorithm on these triangles, to morph the Example to the Input.

Figure 6: Image on the left gets morphed to the image on the right

Figure 7: Another example of morphing

3.3 Foreground segmentation

We use the Grabcut algorithm from Rother et al.[2004], to segment out foreground from the background, in the Input, as well as the morphed Example image. The foreground for each would consist of the head, neck, and shoulders. We create masks according to this segmentation, and store the morphed Example's background for later use. The bit-mask has ones for the foreground and zeros for the background.

Figure 8: Using GrabCut we segment out the image into foreground and background so that transfer could be done on the foreground and background could be copied over from the example image

Some amount of smoothing is done after the segmentation so that it does not create any artifacts during style transfer.

3.4 Pyramid Based energy transfer for foreground

We begin by constructing a Laplacian pyramid of the image, without down sampling as we proceed through the levels:

$$L_l[I] = \begin{cases} I - I \otimes G(x) & if\ l = 0 \\ I \otimes G(x^l) - I \otimes G(x^{l+1}) & if\ l > 0 \end{cases}$$

For our purposes we used $x = 1.2$ in our implementation. The last layer of the pyramid is the residual:

$$L_{lmax}[I] = I \otimes G(x^{lmax})$$

We define the energy of an image at a certain level in the Laplacian pyramid , as the square of the Laplacian at that level convolved with a Gaussian filter:

$$S_l[I] = L_l^2[I] \bigotimes G(x^{l+1})$$

At each level, we calculate the energies of both the images, and then calculate the *gain* between the energy maps of the morphed example image and the input image. Gain is defined as:

$$gain_l = \sqrt{\frac{S_l[E]}{(S_l[I]+\varepsilon)}}$$

We then set the output image to be the input image, multiplied with the gain for all layers except the residual whereas for the residual layer we directly use the style image residual:

$$L_l[out] = L_l[content] \times gain$$

$$R_{lmax}[out] = R_{lmax}[example]$$

At times, there are issues with the gain maps, so we clamp their limits to between 0.1 and 2.9, and then apply a Gaussian averaging filter, to smoothen it. This is defined as the robust gain:

$$RobustGain = max(min(Gain, \Theta_h), \Theta_l) \otimes G(x^l)$$

We then reconstruct the final output image from the new Laplacian pyramid $L_l[out]$ for l = 0 to lmax-1 and $R_{lmax}[out]$ for the residual.

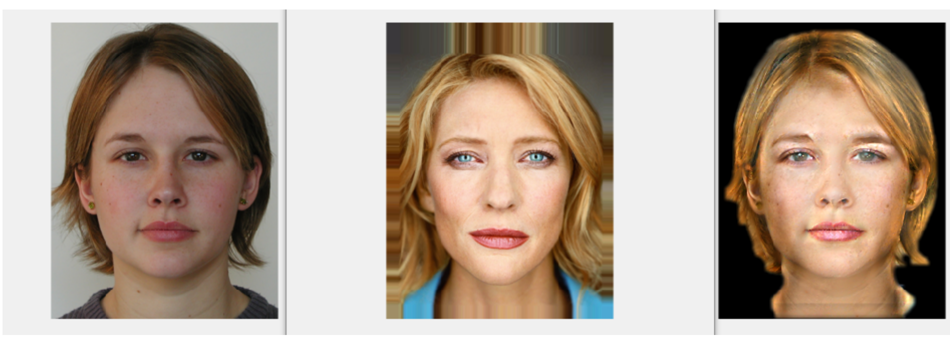

Figure 9: Performing multi-scale style transfer from the foreground of the example image to the foreground of the content image

We use the CIE-Lab color space, as it approximates human perception, and work on each channel independently for each of the steps above. We assume each image to have a double-representation (0.0 to 1.0 range for each pixel).

Figure 10: In our initial results, using RGB images to perform multi scale transfer does not lead to the best results since a lot of the style is not transferred and the style of the original image is retained.

3.5 Background transfer

We take the background of the morphed Example image, using the mask we computed earlier (using grabcut), and extrapolate it to find the likely background colors at intermediate points. Currently we perform multiple iterations of max filtering and mean filtering over the background to ensure the entire image is covered. We then merge this background into the segmented foreground of the Input image, by blending them, to achieve background transfer.

4 Finishing Touches

To the final image we get, we apply high-boost filtering to ensure that individual details are boosted, and also apply bilateral filtering for image smoothness.

Figure 11: For a lot of images increasing the sharpness by performing high boost filtering yields good results where facial hair etc are given a lot more detail.

5 Results and Conclusions

5.1 Final Results

Figure 12: Result 1

Figure 13: Result 2

Figure 14: Result 3

Figure 15: Result 4

Figure 16: Result 5

Figure 17: Result 6

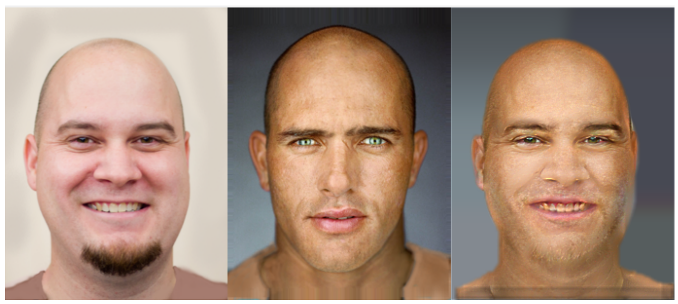

Figure 18: Result 7

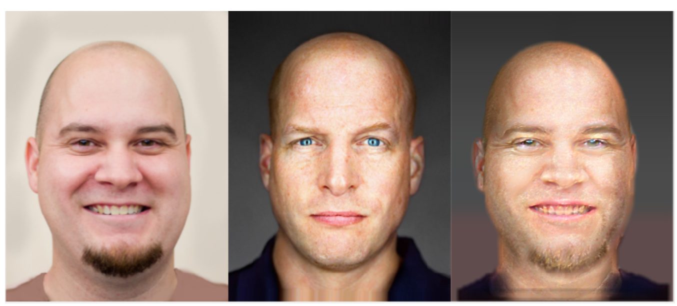

Figure 19: Result 8

5.2 Some Failure Cases

5.2.1 Issues with Glasses

Figure 20: When glasses are present, those parts get transferred as well, so we end up getting black circles around the eyes as we see in this image. This is an issue only when the example image has large glasses.

5.2.2 Incorrect Hair Morphing

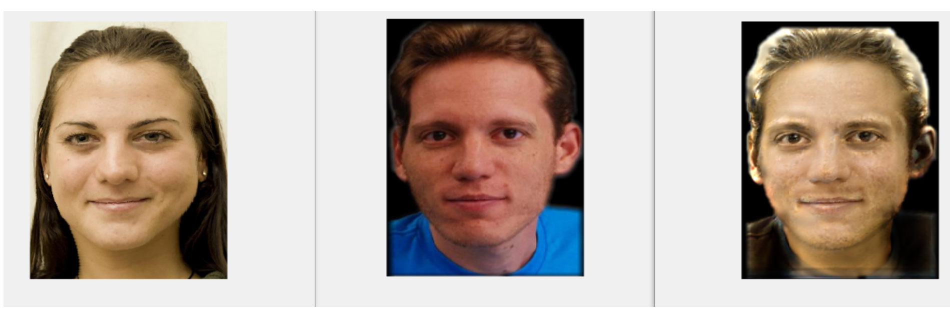

Figure 21: When the example image hair does not get aligned well with the content image hair, the parts where its missing, the background is copied as we see here. This is an issue with the morphing that occurs when examples have a small amount of hair on the top of their head.

5.2.3 Large Illumination Changes

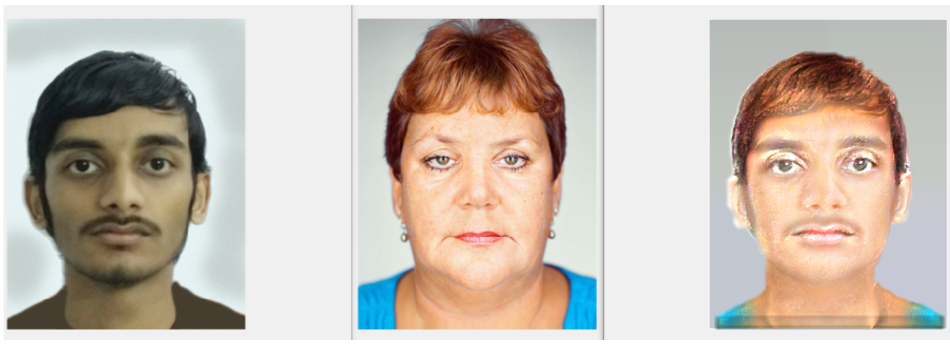

Figure 22: When the example images have high illuminance, the transfer does not take place well. We end up giving low importance to a lot of the important features from the content image.

5.2.4 Incorrect Morphing

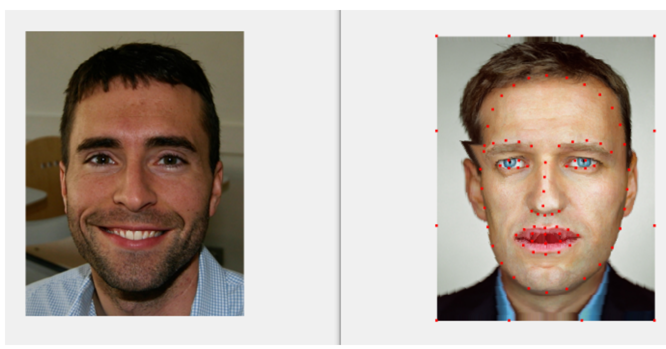

Figure 23: At times, the feature points detected end up occuring very close to each other and the morphing performed on these points does not give us good results.

5.2.5 Grab-cut T Shirt Selection

Figure 24: At times grab-cut ignores the T-Shirt and sets it as a part of the background rather than the foreground. When this happens we don't transfer the style from the T-Shirt and we end up ignoring it in entirety. This is a problem for some portraits but in general, its not a huge issue.

6 Further extensions

6.1 Consider more feature points

Once we have the output of grab cut, we can add in more feature points along the boundary in both the images, this would ensure that the ears and the hair get selected as features as well so the morphing would align them better. This would solve a lot of the failures where we are unable to transfer color well since either the hair is missing or the ears are not present in the same locations in both the images etc.

6.2 In-painting of background

Instead of using max and mean filters to propagate the background to the blank parts of the segmented image, we can use in-painting which will automatically fill in the blank regions with the nearest match. This would give better results in most cases.

6.3 Extension to videos and real-time

The components which take the most time, in this algorithm, are the segmentation (grab-cut) parts and morphing. In videos, since each frame is a slight change from the previous, we can re-use segmentation results (computing flows near the boundaries. Instead of transferring the style from the example to the input in each successive frame, we transfer style to the first frame, and propagate that style to successive frames, and 'reset' by transferring from example, every 12-24 frames.

References

[1] Carsten Rother, Vladimir Kolmogorov, Andrew Blake. "GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts. *SIGGRAPH (2004)*.

[2] Thaddeus Beier, Shawn Neely - Feature-Based Image Metamorphosis. *SIGGRAPH (1992)*.

[3] Ce Liu, Jenny Yuen, Antonio Torralba - SIFT Flow: Dense Correspondence across Scenes and its Applications *IEEE (2011)*

[4] Jitendra Malik, Pietro Perona - Preattentative texture discrimination with early vision mechanisms. *(1989)*

[5] Anat Levin, Dani Lischinski, Yair Weiss - A Closed Form Solution to Natural Image Matting. *CVPR (2006)*

[6] Jason M. Saragih, Simon Lucey, Jeffrey F. Cohn - Face Alignment through Subspace Constrained Mean-Shifts *ICCV (2009)*