# Style Transfer for Headshot Portraits

## Summary

Style Transfer is a technique that allows you to transfer the style of one image to another.

In this paper they present a method for transferring the style of a headshot portrait to a target image by transferring properties such as local contrast and lighting while being tolerant to differences in faces.

It also proposes a method to automatically find a good example photo to use as a reference for portrait from a set.

## Methodology (Multiscale Local Transfer)

- First dense correspondence is established between the input and the model: each input pixel is put in correspondence.
- Then local statistics of example are transferred onto the input
- Lastly transfer eye highlights and the background

### Dense Correspondence

- Obtain Correspondences between input and reference image using `Coarse-to-fine` approach.
- Find 66 facial landmarks using a template {Saragih et al. 2009}
- Align eyes and mouth using `affine transform` similar to {Joshi et al. 2010}
- Morph example to input using segments of face template obtrained {Beier and Neely 1992}. This initial esitmation often successfully aligns the eyes and mouth, but misses important edges such as the face contour and mouth.
- Correspondence is refined using SIFT Flow {Lie et al. 2011}

### Multiscale Transfer of Local Contrast

### Multiscale Decomposition

- First step is to decompose input and example images to multiscale `Laplacian stacks`. Laplacian stack is a stack of images where each image is the difference between the original image and its Gaussian blurred version.

- This decomposition allows us to transfer local contrast and lighting independently.

- The consruction uses a `2D normalized Guassian Kernel` $G(\sigma)$ of standard deviation $\sigma$.

- Using *convolution operation* ($\otimes$) on input image $I$, **level** $L_l$ of input Laplacian stack is:

  - $L_l[I] = I - I \times G(2)$ i $l = 0$
  - $L_l[I] = I \times G(2^l) - I \times G(2^{l+1})$ if $l \geq 0$

We define **residual** as: - $R[I] = I \otimes G(2^n)$

### Local Energy

- Inspired from power maps {Malid and Perona 1990l Su et al. 2005; Li et al. 2005; Bae et al. 2006}
- Local energy $S$ in each `subband` is estimated by `local average` of the square of subband coefficients.

- We take the `L2 Norm` average of the neighbourhood for finding local average
- Do not `downsample` Laplacian Layers
- Instead adapt the size over which we average the coefficients to match the scale of processed subband.
- $S_l[I] = L_l^2[I] \otimes G(2^{l+1})$, where $I$ is input image
- $\tilde{S}_l[E] = W(S_l[E])$, where $W$ is warping operator and $E$ is example image

#### Robust Transfer

- We modify the input subbands so that they get te same energy distribution as the example subbands.
- $L_l[O] = L_l[I] \times \text{Gain}$, where $O$ is output image.
- $\text{Gain} = \sqrt{\dfrac{\tilde{S}_l[E]}{S_l[I] + \eta}}$. where $\eta$ is small number to avoid division by $0$.
- To address issues of artifacts where $I$ and $E$ mismatch, `Robust Gain Map` is used.
- $\text{RobustGain} = max(min(\text{Gain}, \theta_h), \theta_t) \otimes G(\beta 2^l)$
- **Values used**: $\theta_h = 2.8$, $beta = 3$, $n = 6$ for the Laplacian stacks
- For `output residual`, directly copy the `warped example residual`: $R[O] = W(R[E])$
- **Choice of neighbourhood size** is critical

#### Dealing with Colors

- Transformations are done in `CIE-Lab` color space, because it approximates human perception and process each channel independently.
- In practice we, **skip first 3 subbands**

#### Using a Mask

- Not completely necessary
- But for extending this transfer algorithm to use a `mask` defining a *ROI*, we truncate the Guassian convolutions so that they only consider values within the mask.
- Basically replace each Guassian Convolution as:
  - $\text{Image} \otimes G \rightarrow \dfrac{(\text{Image} \times \text{Mask}) \otimes G}{\text{Mask} \otimes G}$
- We run `GrabCur` {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}
- Without mask large differences may exist in the background region perturb the transfer algorithm near the face contour, and using a mask solves this problem.

#### Additional Post Processing

#### Transfer of Eye Highlights

- Separate the `specular reflection` from example eyeball and copy that onto the input's eyes
- On example, we first locate the iris using `circular arc detection` around the positions given by **face template**.

- Thn create an approximate segmentation of iris, highlight and pupil using `k-means` algorithm on the pixel colors with `k=3`
- Refine Reflection mask using `alpha matting` {Levinet al. 2008}
- Detect the existing highlights in the input using `thresholding.`
- In practice, a threshold of `60` on the `L channel` of CIE-Lab colors space
- Then erase the detected pixels and fill the holes using `inpainting`
- In practice, using `griddata` Matlab function was sufficient.
- Not we compose the example highlights on top of the input eyes, then center them using the `pupils` as reference, and then scale them proportion of the `iris radii`.

**Background**

- Directly replace the input background with the example background
- Use previous computed masks to extract the example background and replace the input background with it.
- We extrapolate the missing data using `inpaiting` using `griddata` Matlab function.

**Automatic Selection of the Example**

- Use the local energy `S`, as the face featuer vector
- Look for the candidate with the colsed distance to the input in feature space
- Concatenate $S_l$ over all scales to get the feature vector representing a face image.
- Use `normalized coress-correlation` to find the two feature vectors as the similarity function.
- It's more robust to image retouching than the `L2 norm` distance.
- For computational efficiency, do not wrap the exmaple image in the searching step.