SIGGRAPH2014

Style Transfer for Headshot Portraits

YiChang Shih
MIT CSAIL

Sylvain Paris
Adobe

Connelly Barnes
University of Virginia

William T. Freeman
MIT CSAIL

Frédo Durand
MIT CSAIL

CSAIL

Adobe

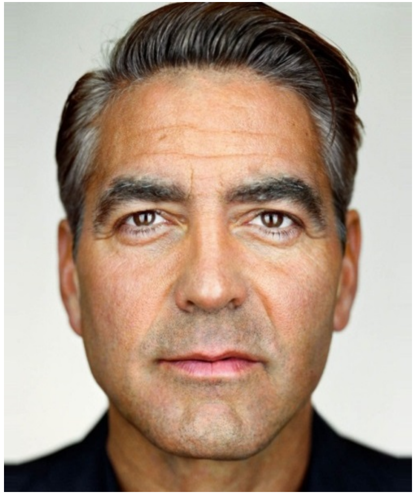
UNIVERSITY of VIRGINIA

Professional portraits look better

Ordinary photo

Professional photo

The goal: make good portraits easy

- Make  look like 

Ordinary photo

Professional photo

- Transfer the style from the example photo
- Automatic

We work on headshots

- What we match: **retouching, texture, lighting**
- What we do not match: pose, expression, clothing, focal length, aperture

Preview our result

Input

Example

Output

Hard problem: color transfer is not sufficient

- Humans are intolerant to artifacts on faces

Input

Example

Our method

[HaCohen et al. 2010]
(lighting and details
are missing)

Related work: global transfer

[Bae et al. 2006, Sunkavalli et al. 2010…]

- Work well on landscapes

Input

Model

Output by Bae et al. [2006]

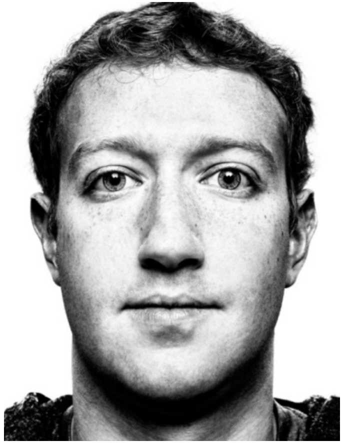
- Do not work as well on portraits

Related work: global transfer

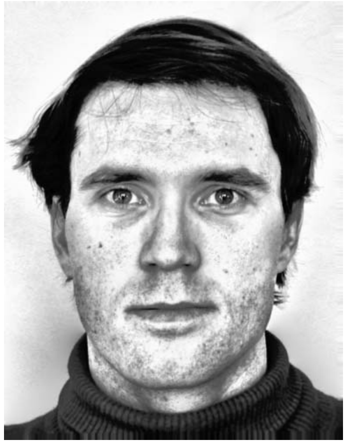
[Bae et al. 2006, Sunkavalli et al. 2010…]

- Work well on landscapes

Input



Model



Output by Bae et al. [2006]

- Do not work as well on portraits

Related work: local style transfer

- Time hallucination [Shih et al. 2013, Laffont et al. 2014]

Input: afternoon

Example images

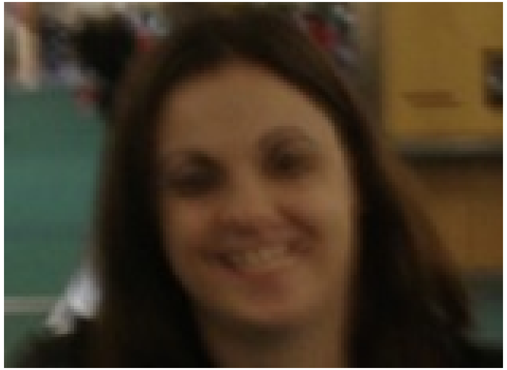
Output: night

- Requires two images: before and after

Related work: face enhancement

[Joshi et al. 2010, Shih et al. 2013 …]

- Image restoration: deblurring, denoising …



Blurred input face

Examples

Output: deblurred face

- We focus on photographic stylization.

Problem statement

- **Input**: a casual frontal portrait and an example

- **Output**:
  - The input portrait rendered in the example style
  - Automatic
  - The style includes texture, tone, and color

Key idea #1: local transfer

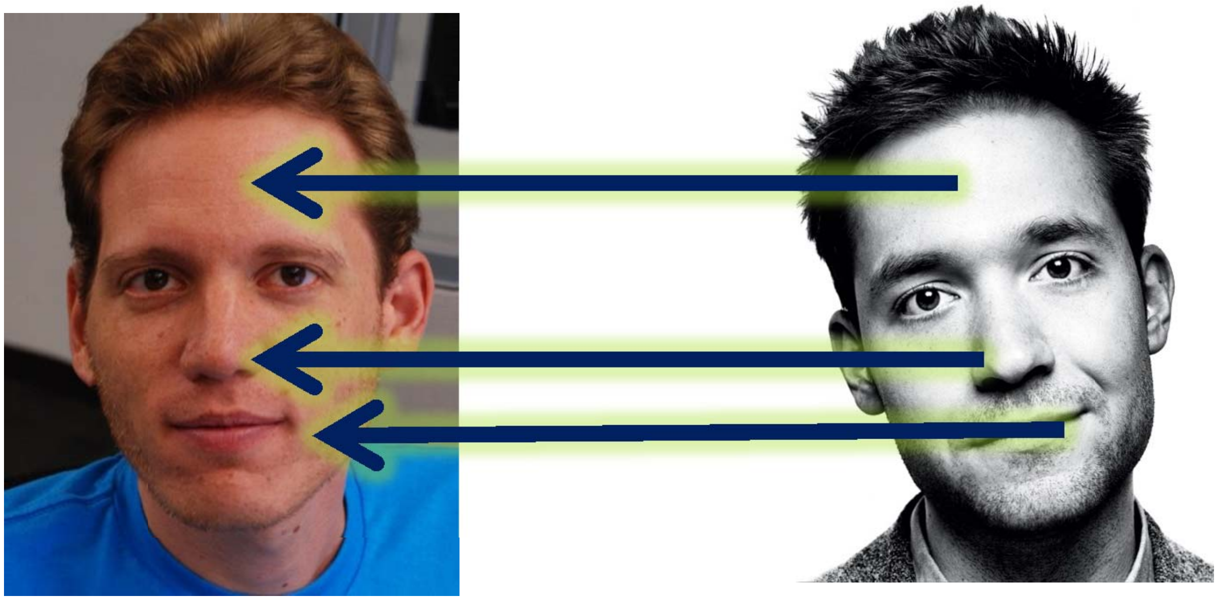
- Local: eyes, nose, skin, etc. are treated differently

Input

Example

Key idea #1: local transfer

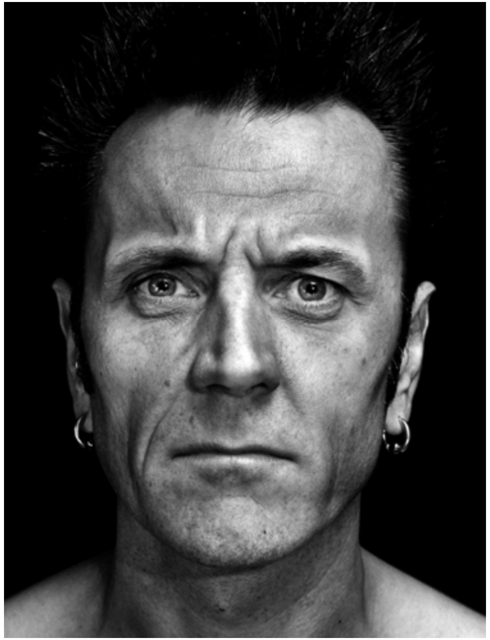
- Local: eyes, nose, skin, etc. are treated differently



Input Example

Key idea #2: multi-scale transfer

- Textures at different scales are treated differently

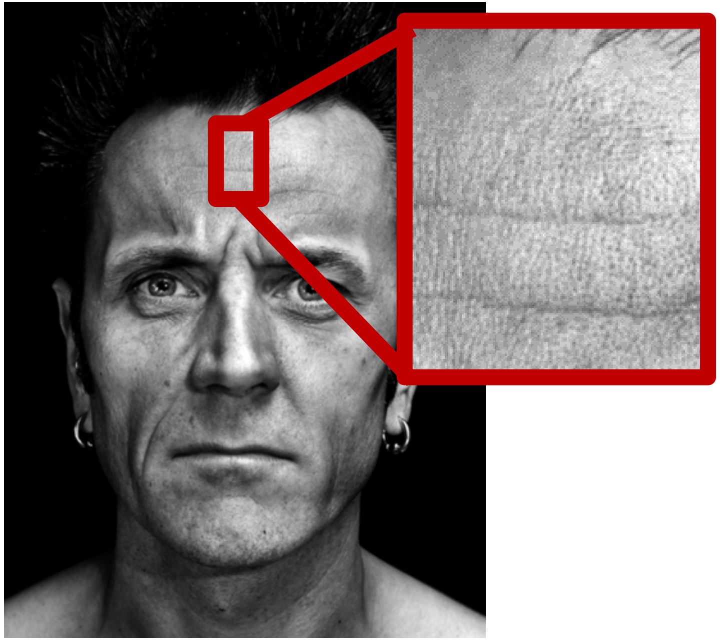


Portrait #1

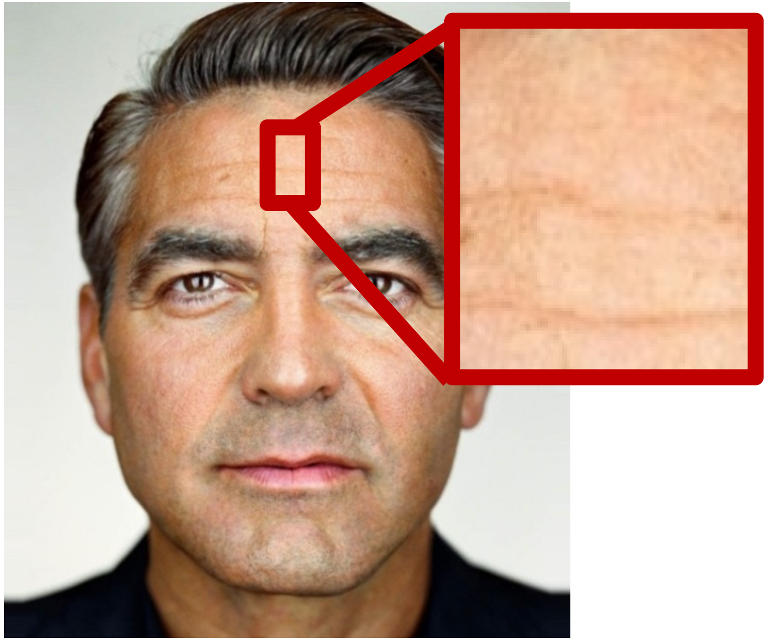
Portrait #2

Key idea #2: multi-scale transfer

- Textures at different scales are treated differently



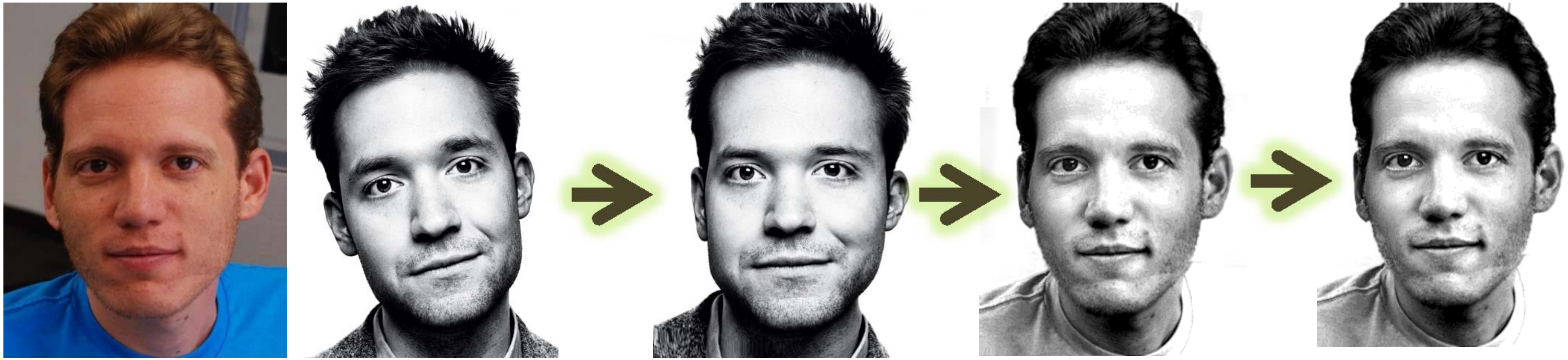
Portrait #1



Portrait #2

Overview of the algorithm

1. Dense matching between the input and example
2. Multiscale transfer of local statistics
3. Post processing on eyes and background



Input Example Step 1: matching Step 2: transfer Step 3: post processing

Step 1: dense matching

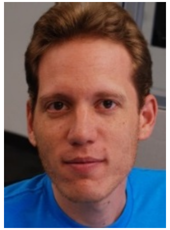
- Rigid warp + SIFT flow to align semantic features [Liu et al. 2008]


Input


Example


Warped example

Step 2: multi-scale local transfer



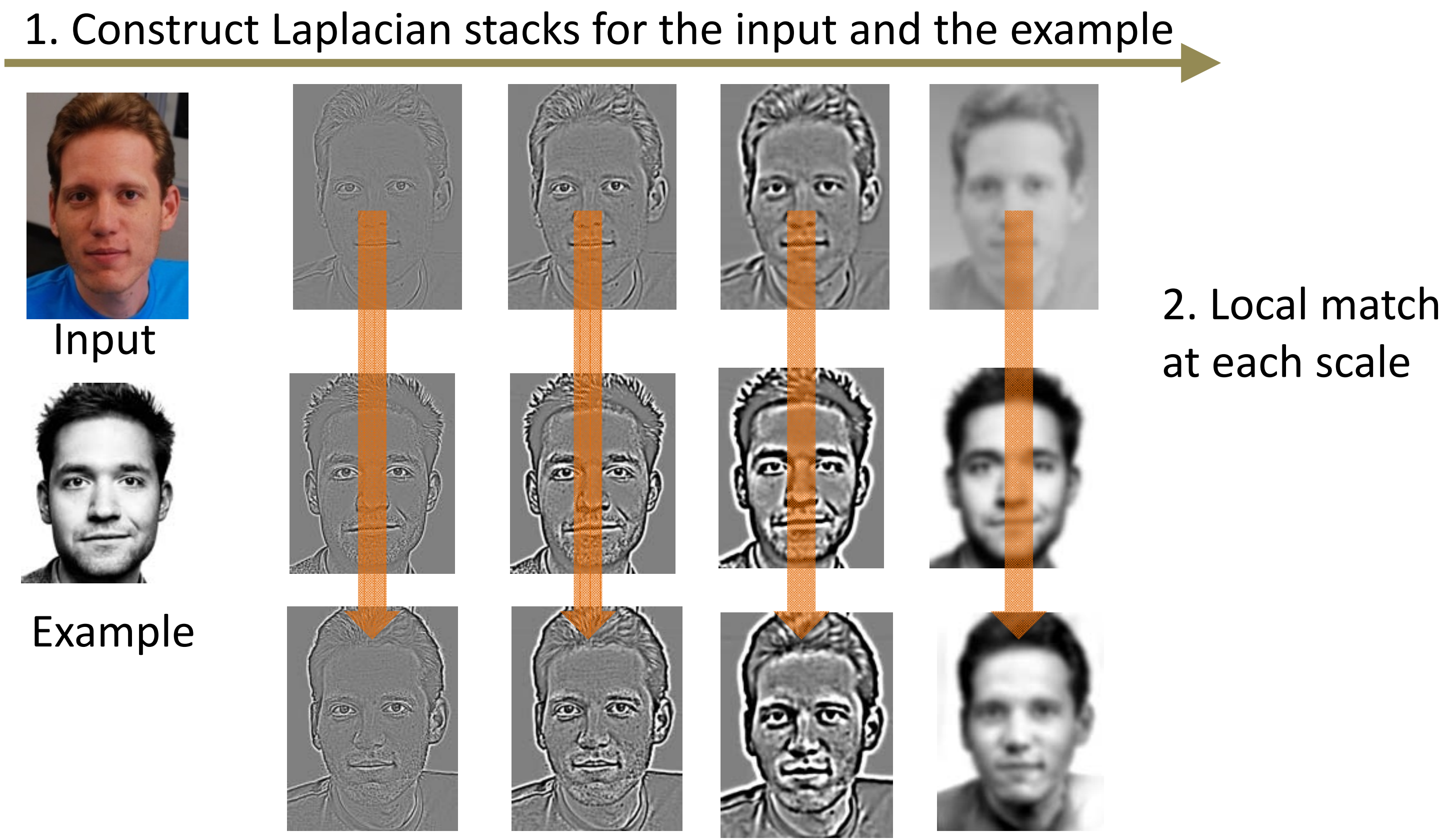
Input

Example

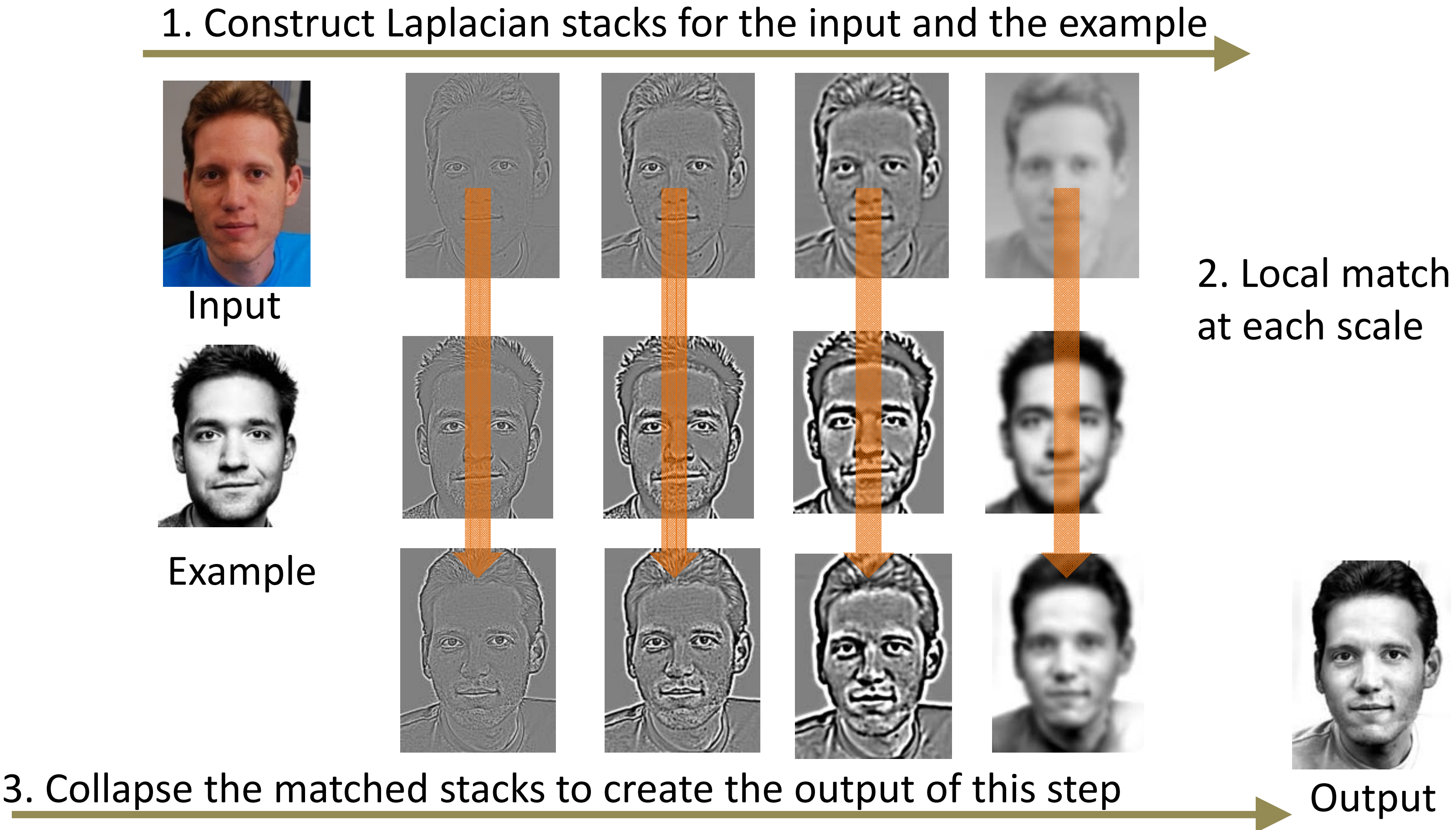
Step 2: multi-scale local transfer

1. Construct Laplacian stacks for the input and the example

Step 2: multi-scale local transfer

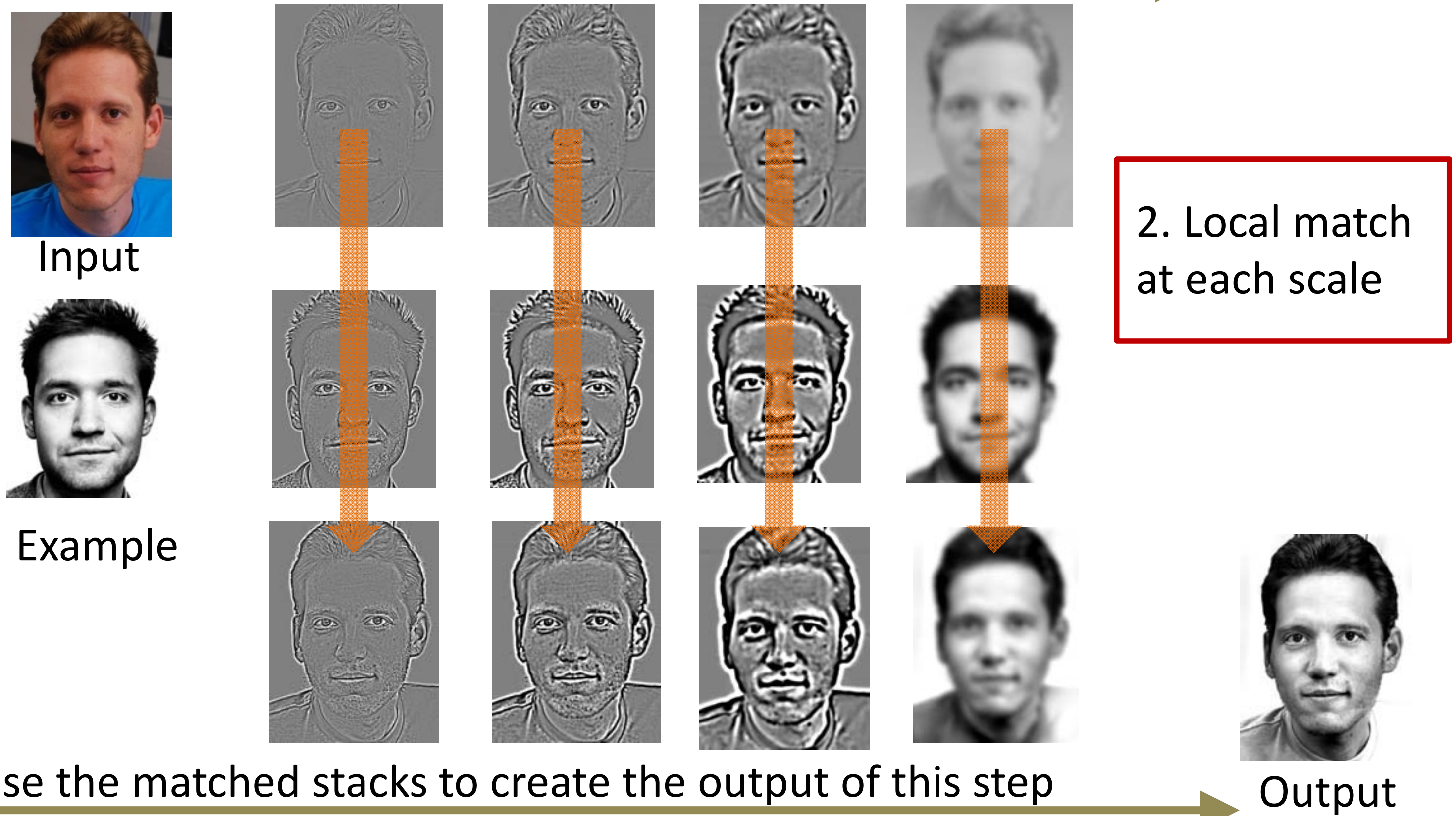


Step 2: multiscale transfer of local statistics



Step 2: multi-scale local transfer

1. Construct Laplacian stacks for the input and the example



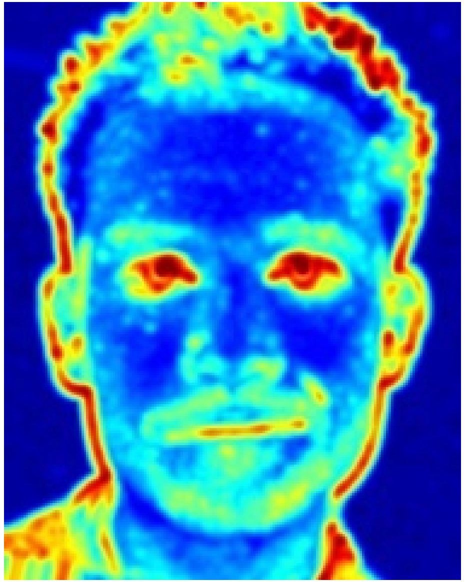
3. Collapse the matched stacks to create the output of this step

Local energy S

$$S = L^2 \otimes G_\ell$$

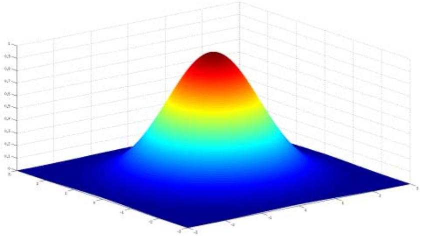
L

Example Laplacian



S

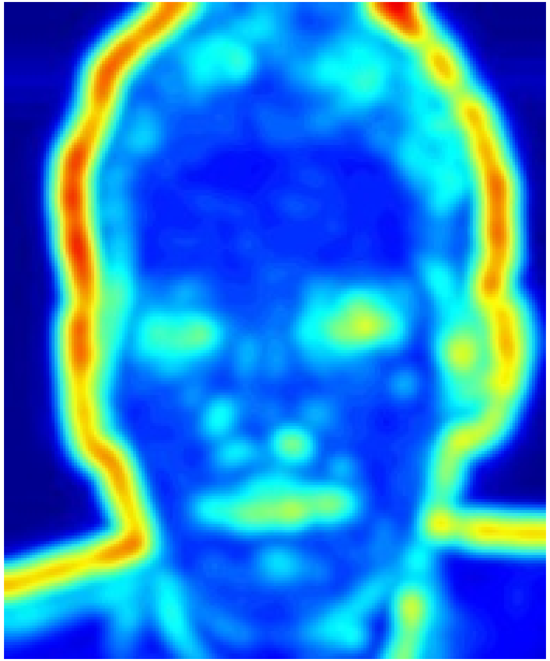
Local energy



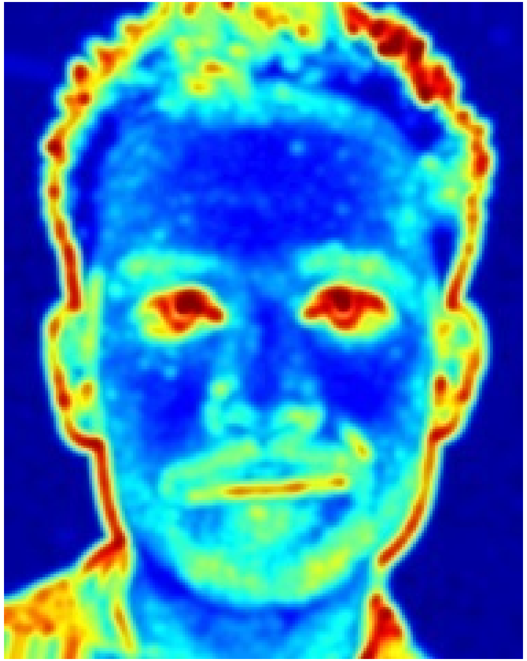
G_ℓ

Gaussian kernel at this scale

At each scale: match local energy



Input energy

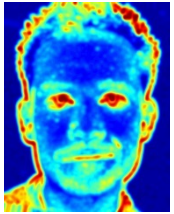


Example energy

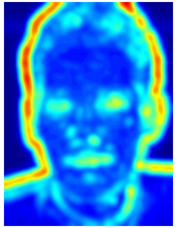
At each scale: match local energy

Compute
the gain map


Example Laplacian


Local energy S[E]


Input Laplacian

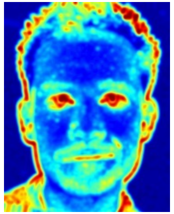

Local energy S[I]

Gain map = $\sqrt{\dfrac{\text{warp}(S[E])}{S[I]}}$

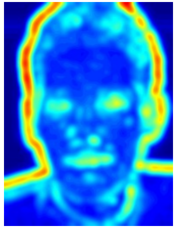
At each scale: match local energy

Compute
the gain map


Example Laplacian


Local energy S[E]


Input Laplacian


Local energy S[I]

Gain map = $\sqrt{\dfrac{\mathrm{warp}(S[E])}{S[I]}}$

Modulate
the input Laplacian


Input Laplacian

×

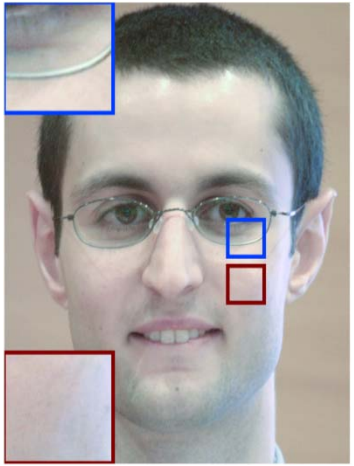

Gain map

=

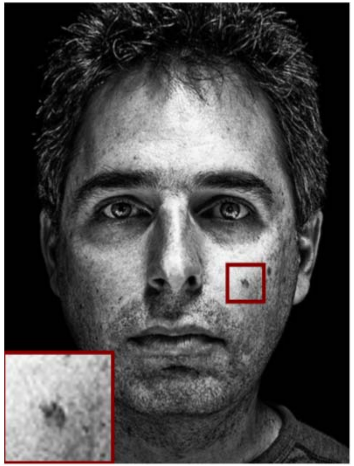

Output Laplacian

Robust transfer

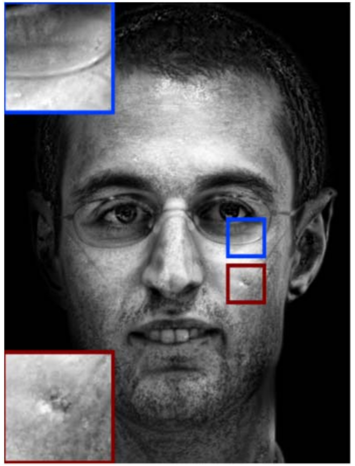
- Clamp the gain map to avoid artifacts caused by moles or glasses on the example



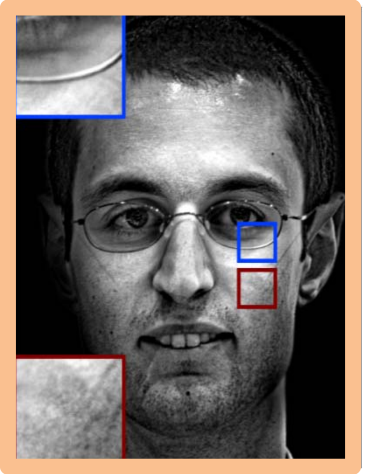
Input



Example



Without robust transfer

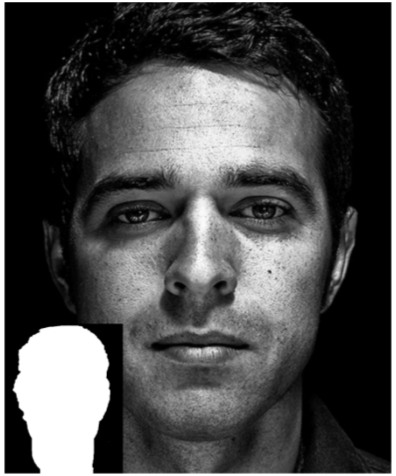


Our robust transfer

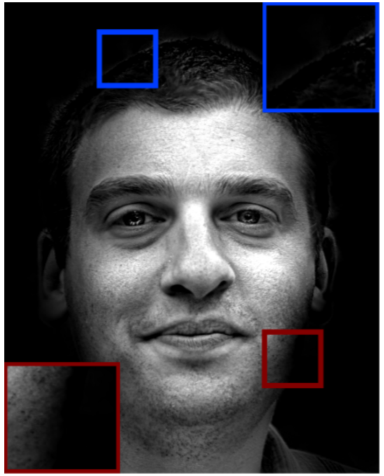
Laplacian using a face mask

- Preserve the hair boundary using normalized convolution and a face mask

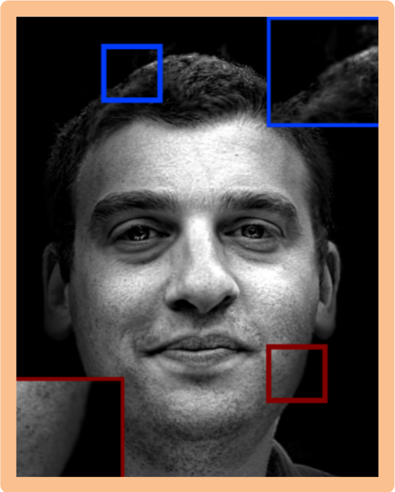
Input



Example



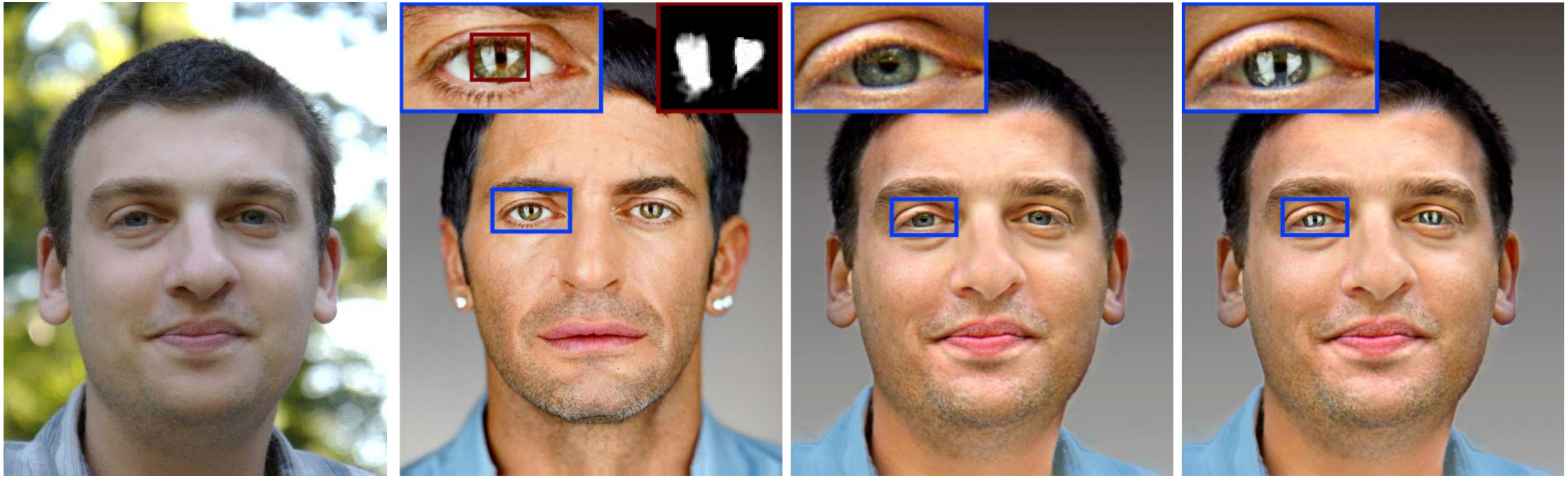
Without using the mask
(the edges disappear)



Our method
(the edges are preserved)

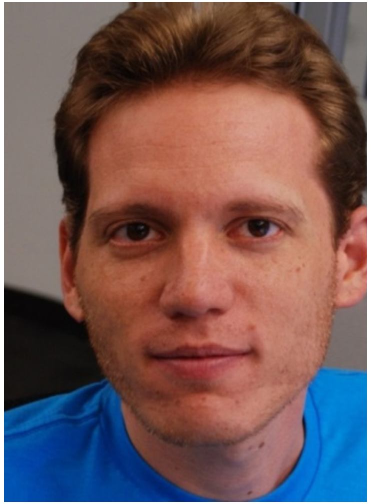
Step 3: post-processing

- Adding eye highlights
- Replacing the background



Input | Example | Without eye highlights | Adding eye highlights (Our final result)

Algorithm recap

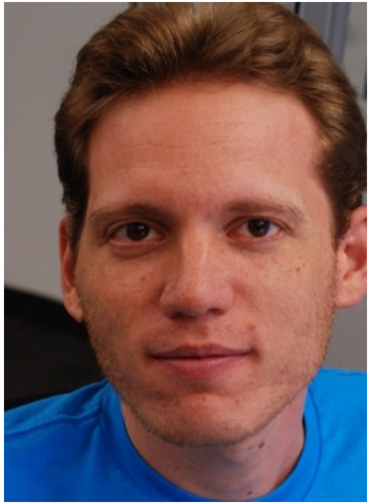


Input

Example

Step 1.
Dense alignment

Algorithm recap



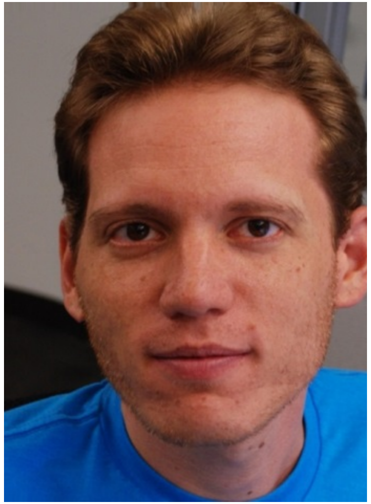
Input

Example

Step 1.
Dense alignment

Step 2.
Local transfer

Algorithm recap



Input

Example

Step 1.
Dense alignment

Step 2.
Local transfer

Step 3.
Eyes and background

Automatic example selection

- Retrieve the best examples based on the face similarity between the input

Input The top three retrieved results

Automatic example selection

- The results are robust to the example choices

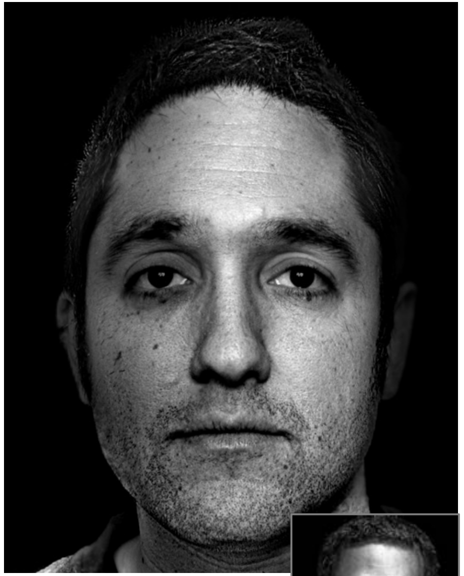
Input

Style transferred results using the top three examples

Results

Examples are shown in the insets

Input



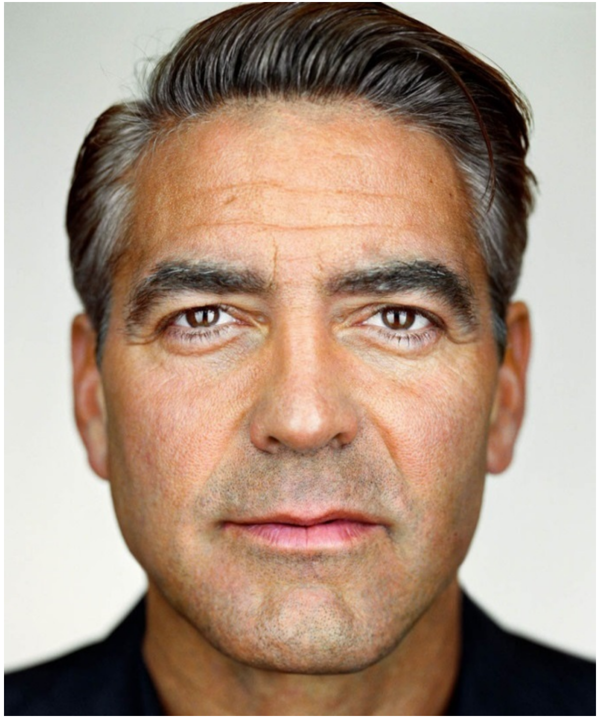
Style 1

Style 2

Style 3

Close-up

Input



Example

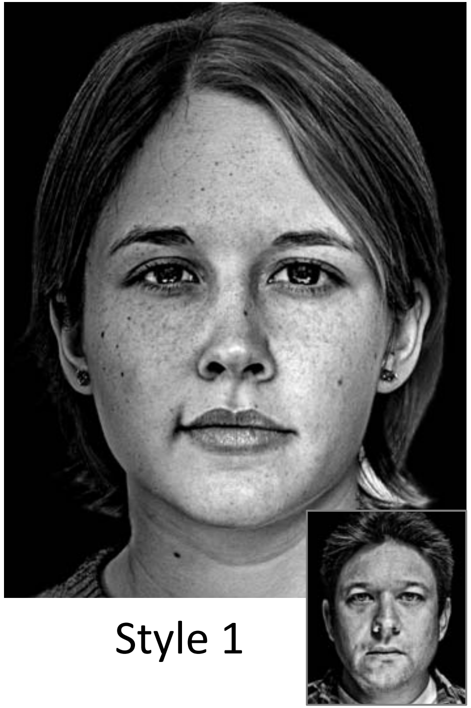
Output


Example

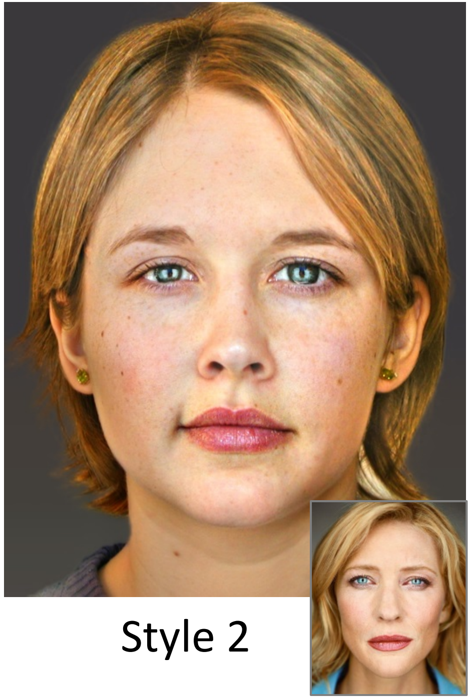
Output

More results

Input



Style 1



Style 2

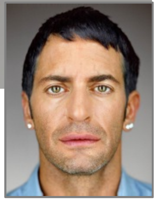
Style 3

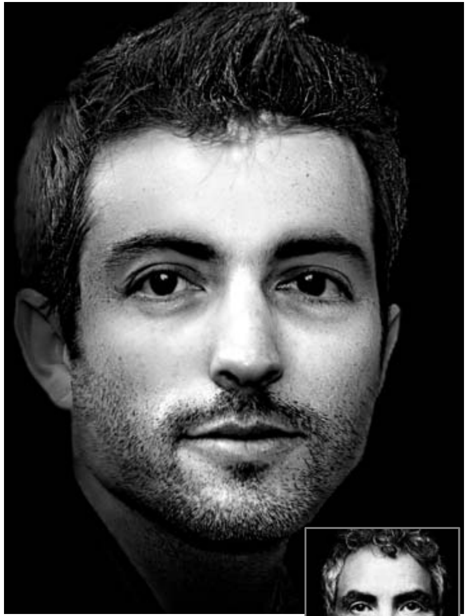
Outdoor input

Input

Style 1

Style 2



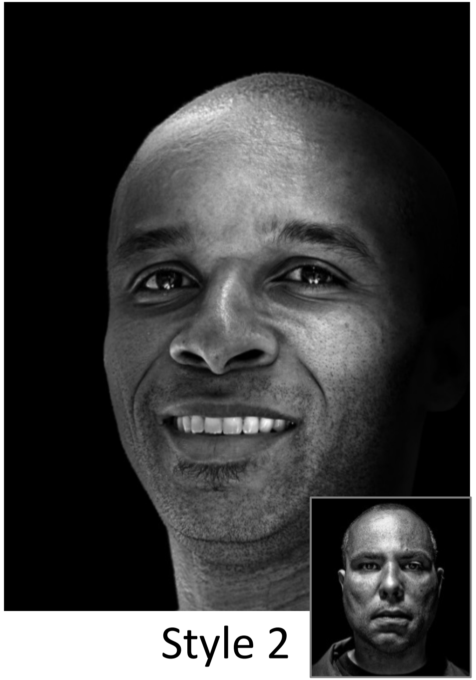


Style 3

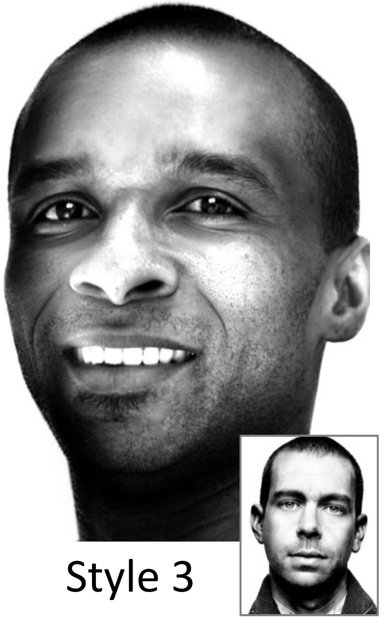
Extra results

Input

Style 1



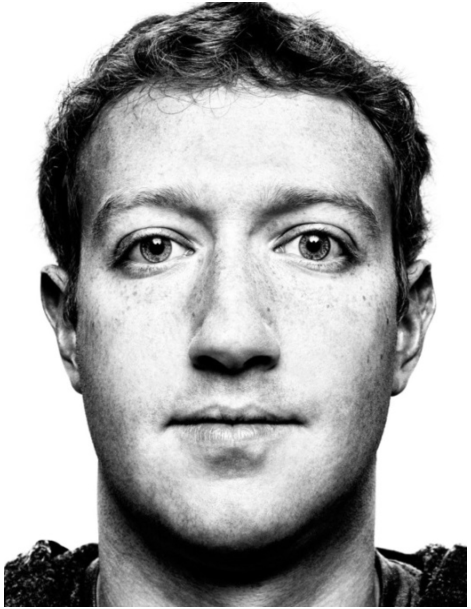
Style 2



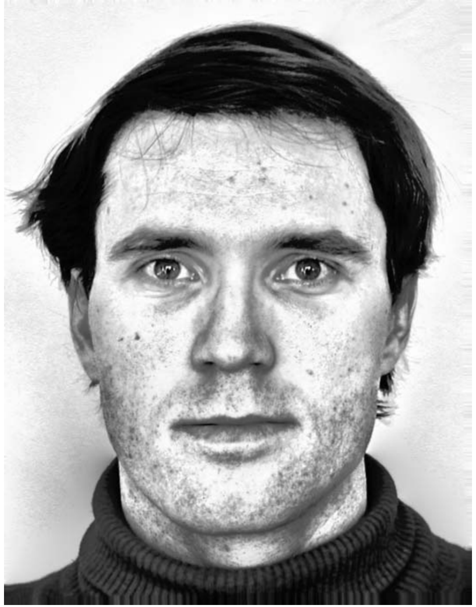
Style 3

Comparisons

Input



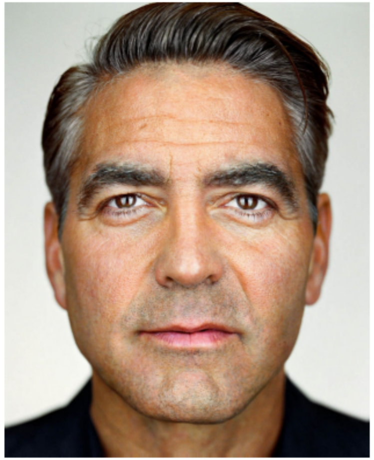
Example



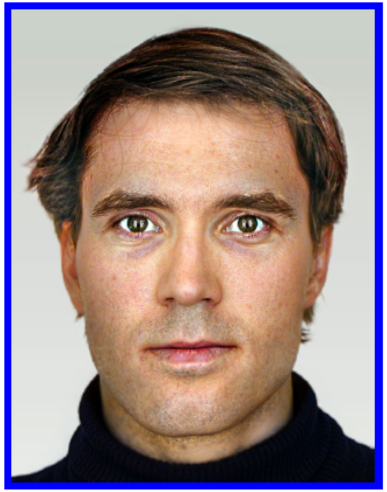
Global transfer
[Bae et al. 2006]

Our result

Input



Example



Our method

[Sunkavalli et al. 2010]

Histogram transfer

[Reinhard et al. 2001]

[Pitié et al. 2007]

Photoshop Match Color

Different success levels: good results

- The inputs are well lit

Input

Output

Hard case

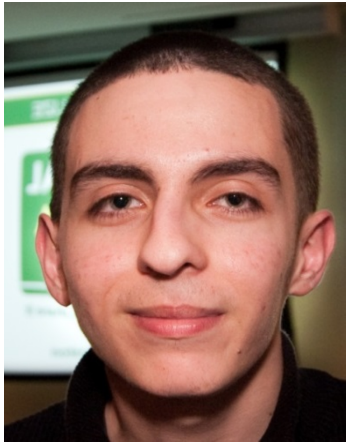
- Matting (face mask) failure

Input

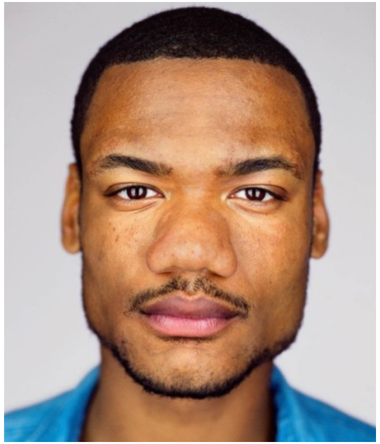
Output

Limitations

- Require the input and the example to have similar facial attributes, e.g., skin color
- Cannot handle hard shadows on the input



Input

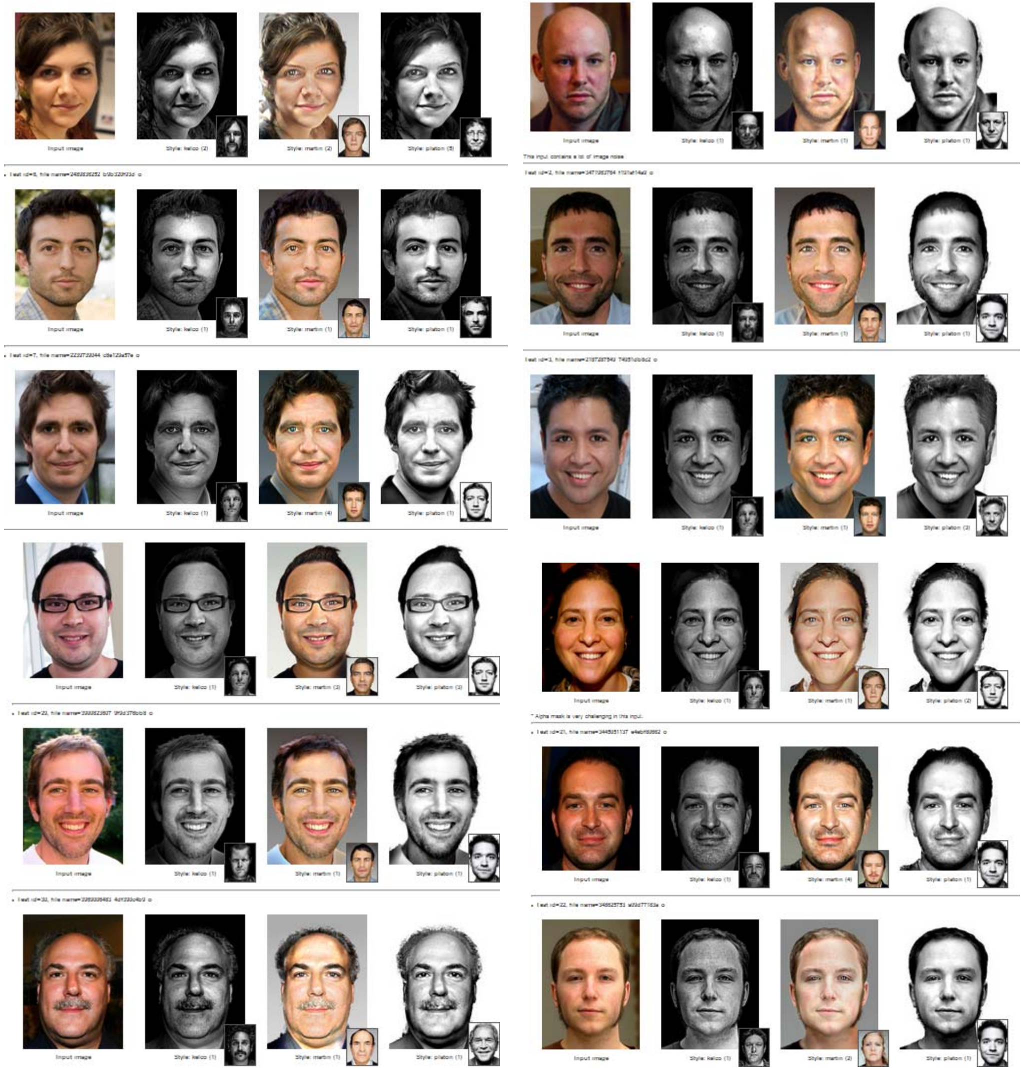


Example

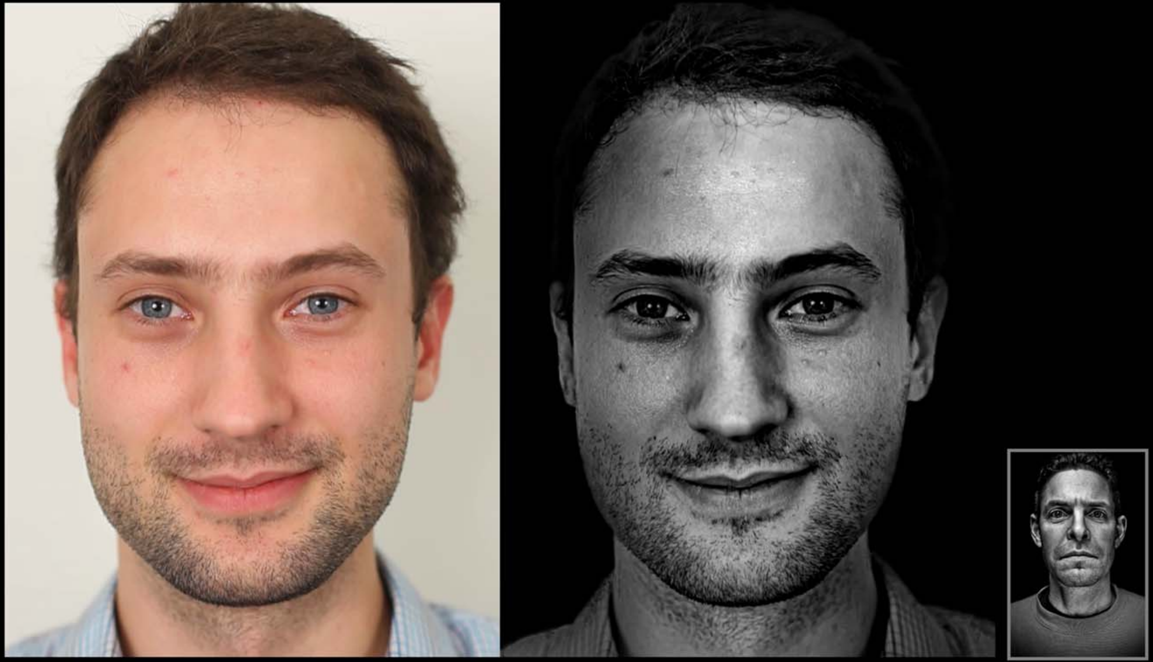
Failure output

Evaluation

- 94 headshot inputs from Flickr
- Available on our website



Extension to videos



Input sequence with extreme facial expressions   Our style transfer result using the example in the gray box

Conclusion

- We introduce a style transfer algorithm tailored for headshot portraits.
- Based on multiscale transfer of local image statistics

Input Example Output

Code and data are available

- Matlab code
- Flickr evaluation dataset

people.csail.mit.edu/yichangshih/portrait_web/

Acknowledgments

- We 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.

Conclusion

- We introduce a style transfer algorithm tailored for headshot portraits.
- Based on multiscale transfer of local image statistics

Input

Example

Output