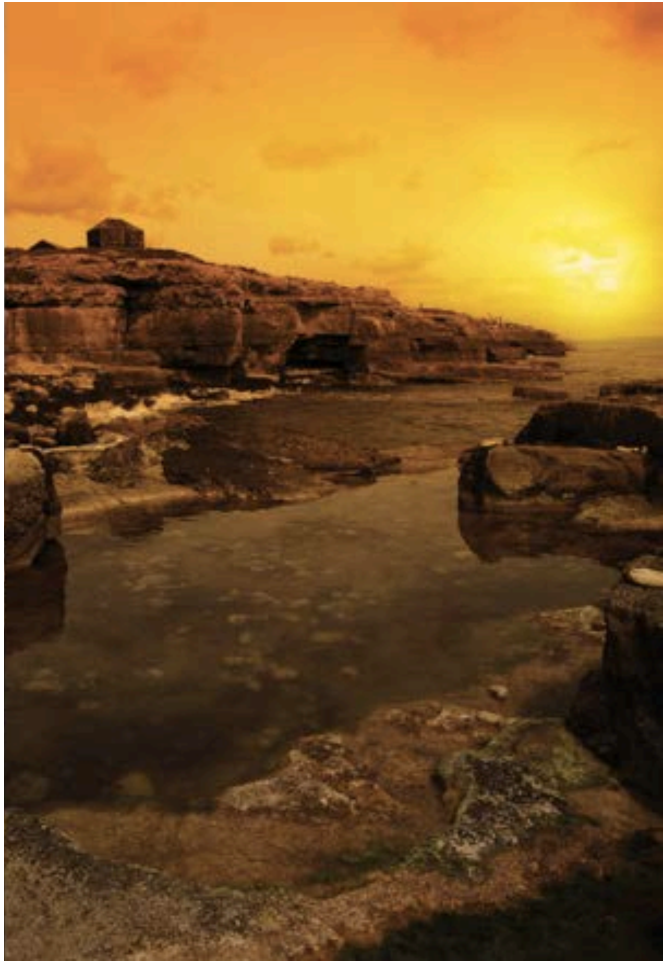
Data-driven Photographic Style using Local Transfer

YiChang Shih
MIT CSAIL, Department of Electrical Engineering and Computer Science
Feb 11, 2015

Image style

Before



After

Correcting exposure

Before

After

HDR tone mapping

Changing a style

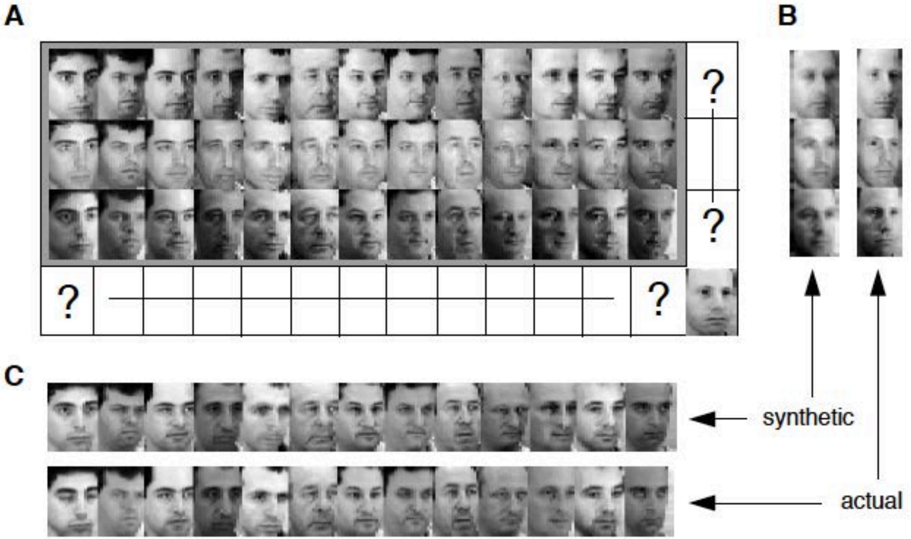
- Same content, different visual appearances

- Convey a unique mood
- Make it memorable
- Impress people

Style vs. content

[Tenenbaum and Freeman 2000]

- Decompose an image into the style and content
- Modify the style, while preserving the content



More examples of changing style

- Contrast adjustment, exposure correction, color restoration …

Bae et al. [2006]

Bychkovsky et al. [2011]

Hacohen et al. [2011]

Kaufman et al. [2012]

Style transfer

[Reinhard et al. 2001, Bae et al. 2006, …]

- Match the colors in the user-supplied example
- Work well on simple scenes

Input

Example

Output [Pitié et al. 2007]

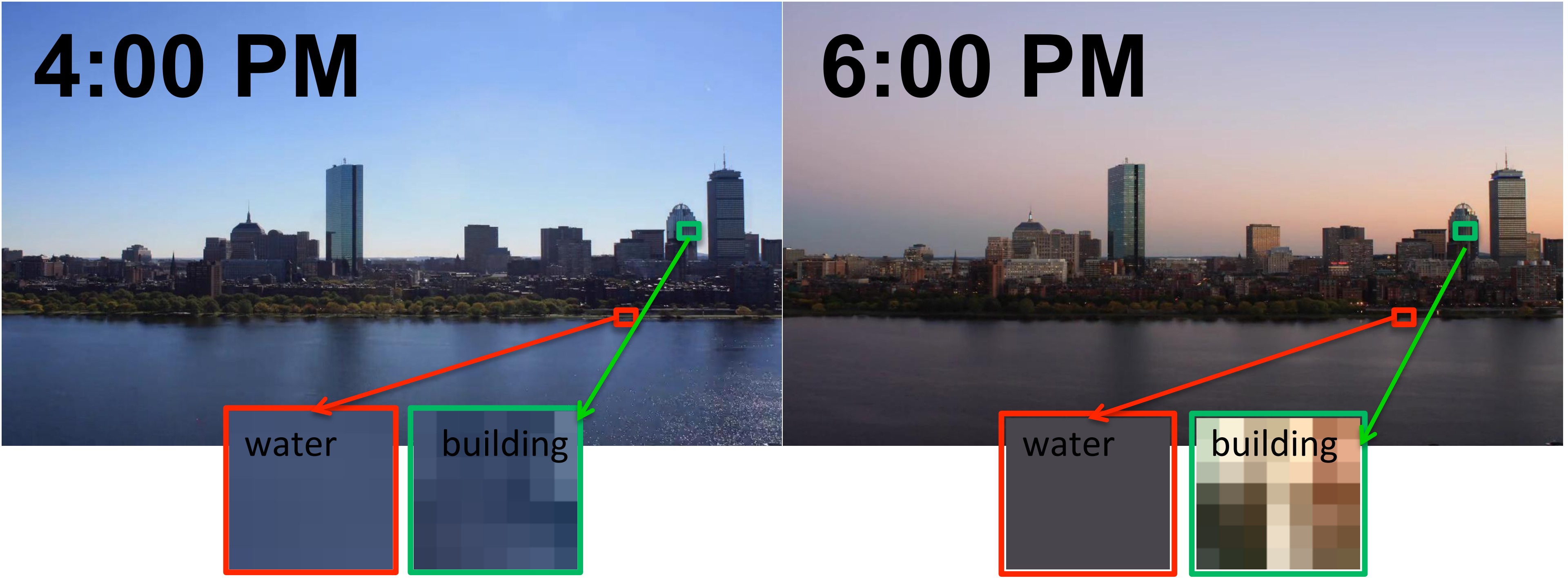
This work: challenging styles

- Changing the time-of-day

Image courtesy of Adrian Dalca

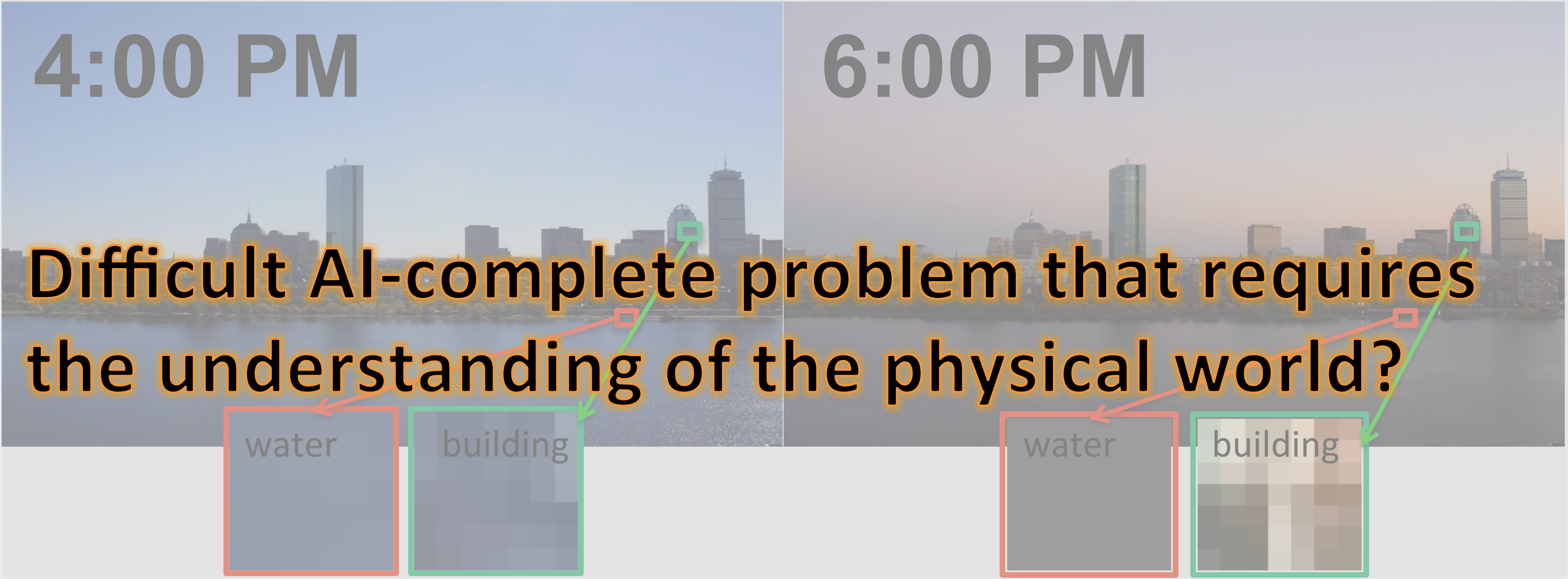
Hard problem

- Water and building become different color
- Depends on material, lighting, physical interaction



Hard problem

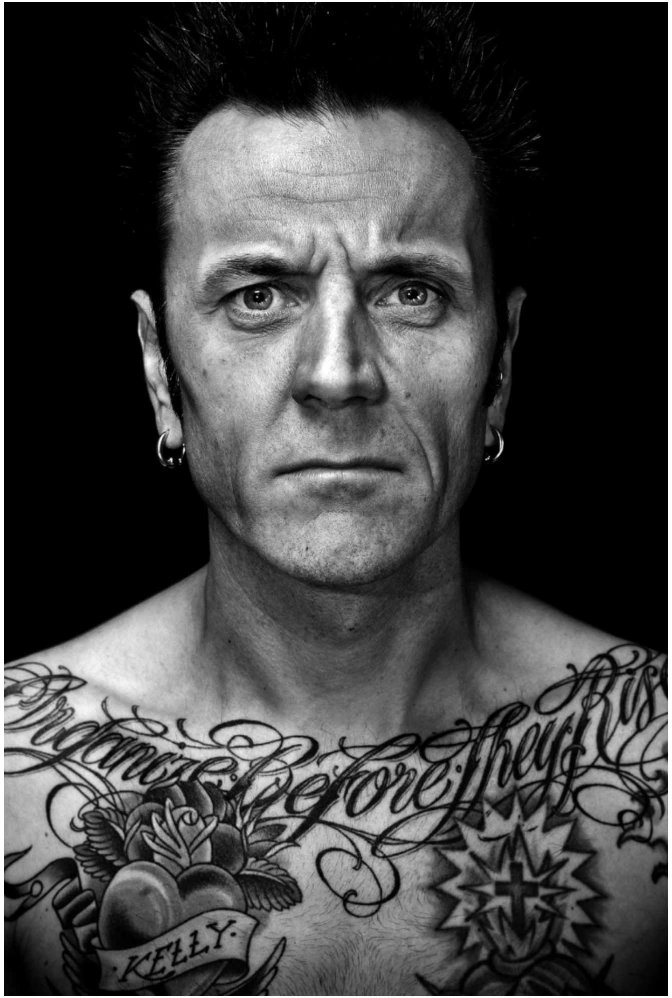
- Water and building become different color
- Depends on material, lighting, physical interaction



Difficult AI-complete problem that requires the understanding of the physical world?

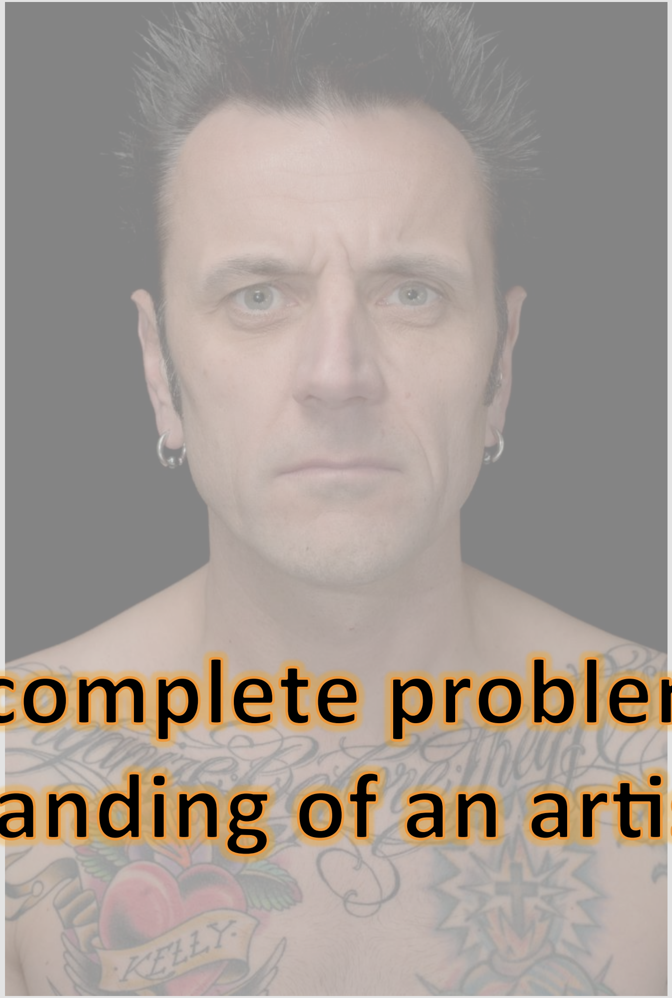
Creative style

Raw image captured by a camera

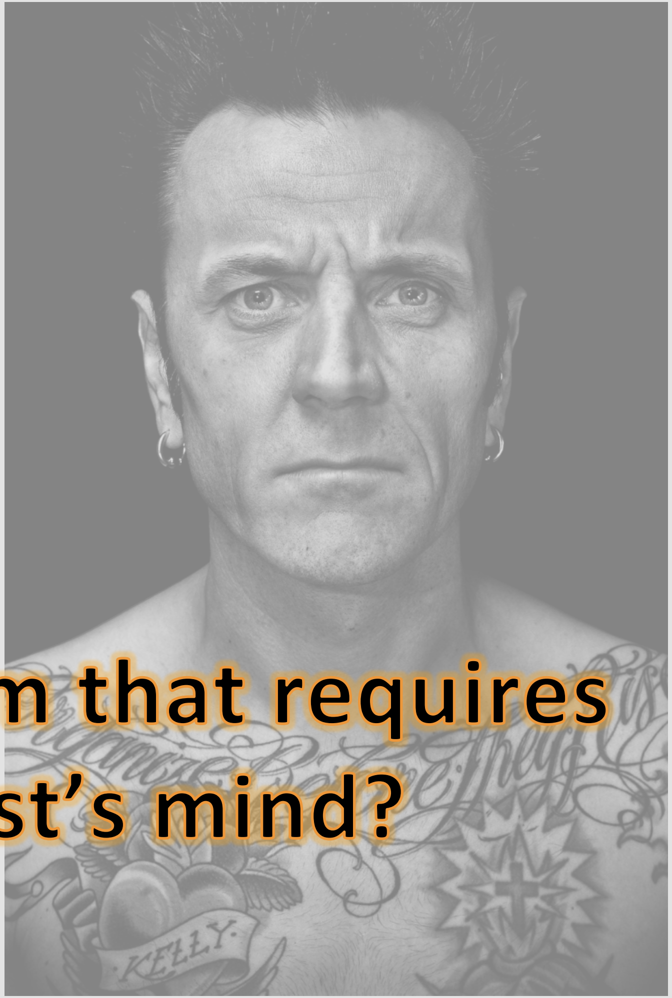


Stylized by photographers

Creative style



Raw image captured by a camera



Stylized by photographers

Difficult AI-complete problem that requires the understanding of an artist's mind?

This work: leverage the power of data

Input Image database Stylized output

Style transfer from a good examples

Input

Example

?

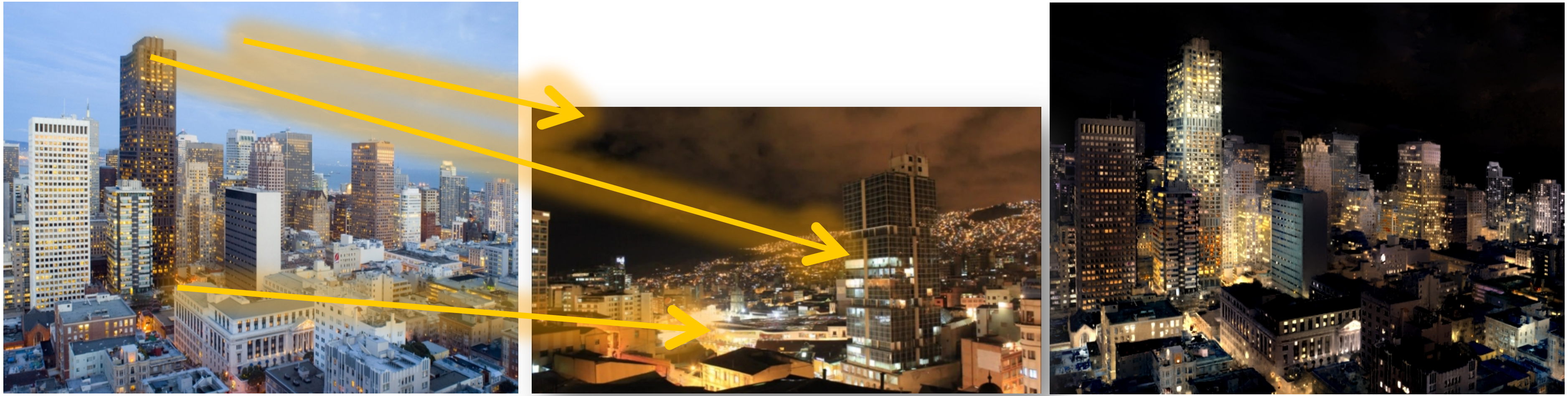
Stylized output

Contribution: local style transfer

- Leverage semantic information by dense correspondences

Input Example Stylized output

Preview



Input Example Stylized output

Close up

Input

Our output at night

Overview of this talk

- Time-lapse hallucination (SIGGRAPH Asia 2013)

Input at afternoon

Output at night

- Portrait style transfer (SIGGRAPH 2014)

Input: ordinary portrait

Output: stylized portrait

Overview of this talk

- Time-lapse hallucination (SIGGRAPH Asia 2013)

Input at afternoon

Output at night

- Portrait style transfer (SIGGRAPH 2014)

Input: ordinary portrait

Output: stylized portrait

Hallucinate scene color variation over time

- Use the photo at time A to predict the photo at time B.

Output: time B **9:00 PM**

?

Hallucinate scene color variation over time

- Use the photo at time A to predict the photo at time B.

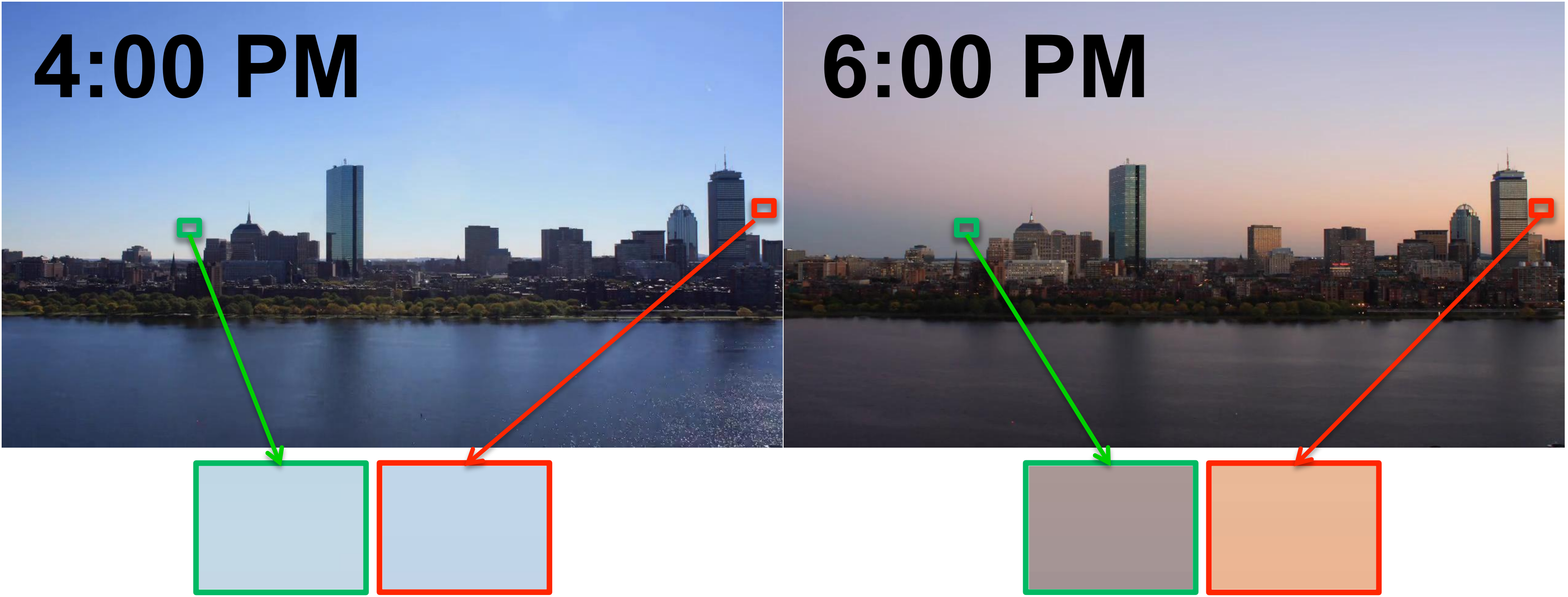
46 minutes too early [kenrockwell.com]

perfect [kenrockwell.com]

39 minutes too late [kenrockwell.com]

Hard problem

- The color change is spatially-variant!

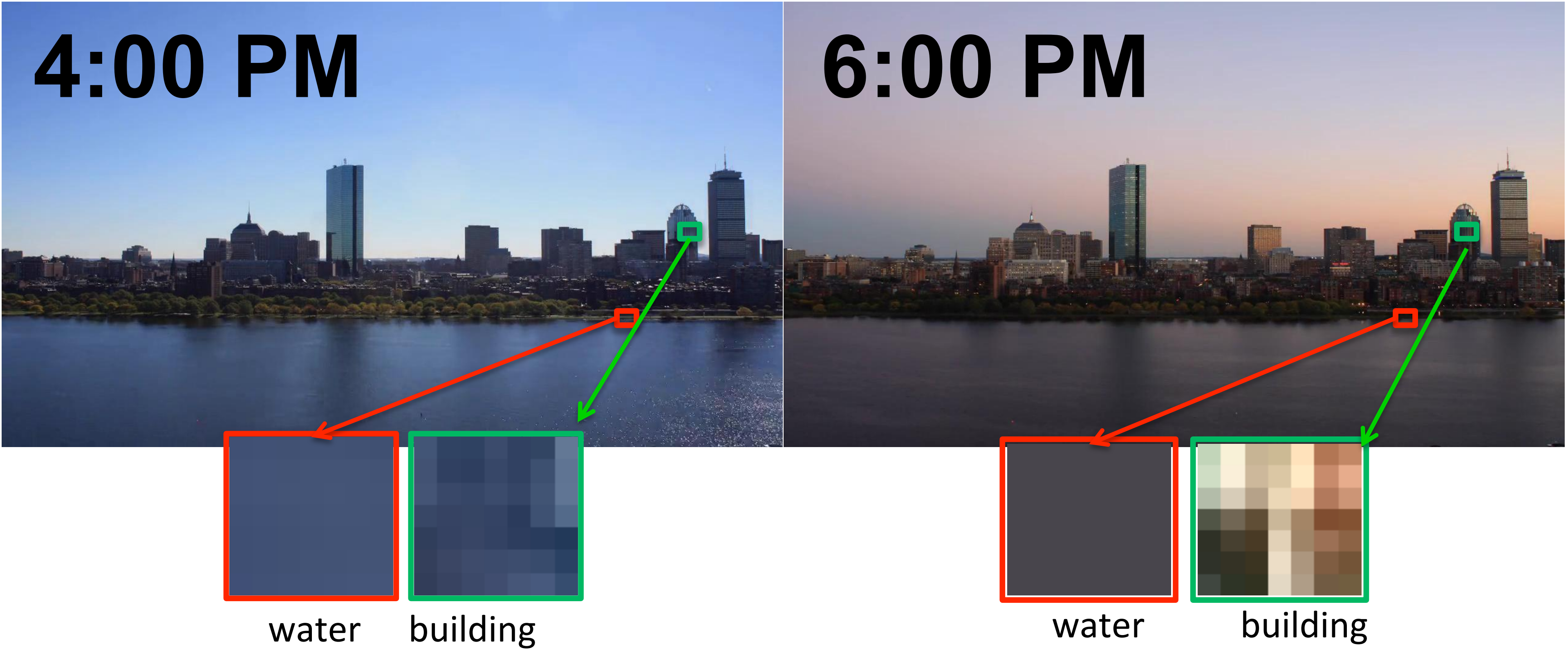


Colors are close at day

Become very different at sunset

Hard problem

- Water and building become different color



Related work: global color transfer

- Match global color statistics
- Works on simple scenes
  [Reinhard et al. 2005, Pouli and Reinhard, 2011, Pitie et al. 2005, …]
- Complex scenes require spatially-variant transfer

Input at daytime

Example at sunset

Output at sunset

Related work: image relighting

- Use image collection of the scene [Laffont et al., 2012]
- Use 3D scene model [Kopf et al., 2009]

Input

Relit result

- We want a general machinery, not rely on data for a specific input image

Related work: analyzing time-lapse video

- Produce good results, but need manually modeling the scene [Lalonde et al, 2009]

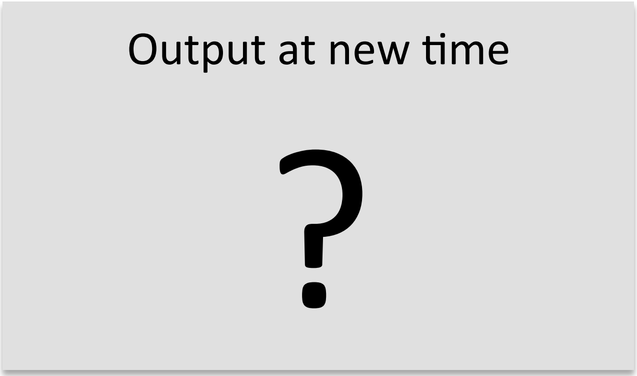
Input at daytime Time A Time B Time C

Problem statement

- Input: a single photo + target time of day
- Output:

  - the same scene as if it was taken at the target time

  - automatic



Our idea: using time-lapse videos

- Cover color changes at different times of a day
- Labeled with time of day

500 videos taken at various outdoor scenes

Overview

1. Match input to video from database

Matched time-lapse video

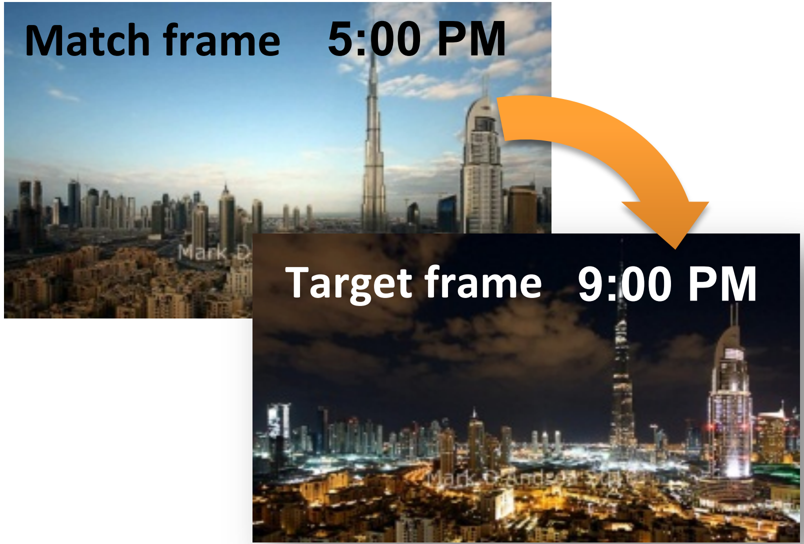
Target time: 9pm

Overview

1. Match input to video from database

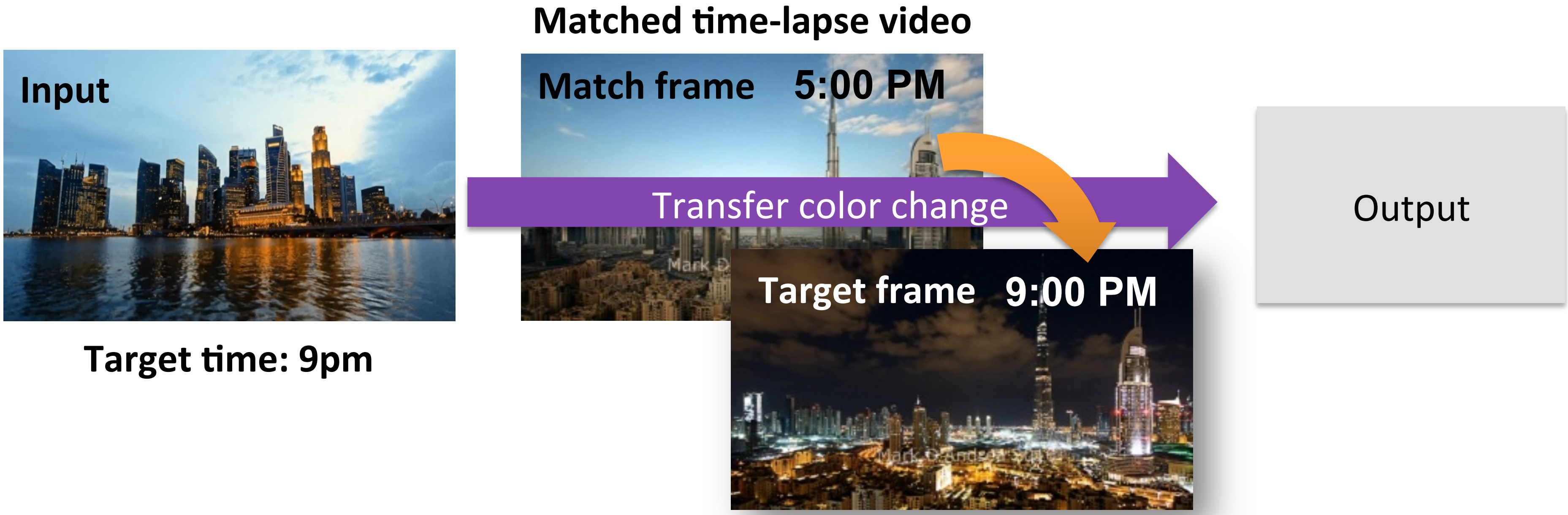
Target time: 9pm

Matched time-lapse video



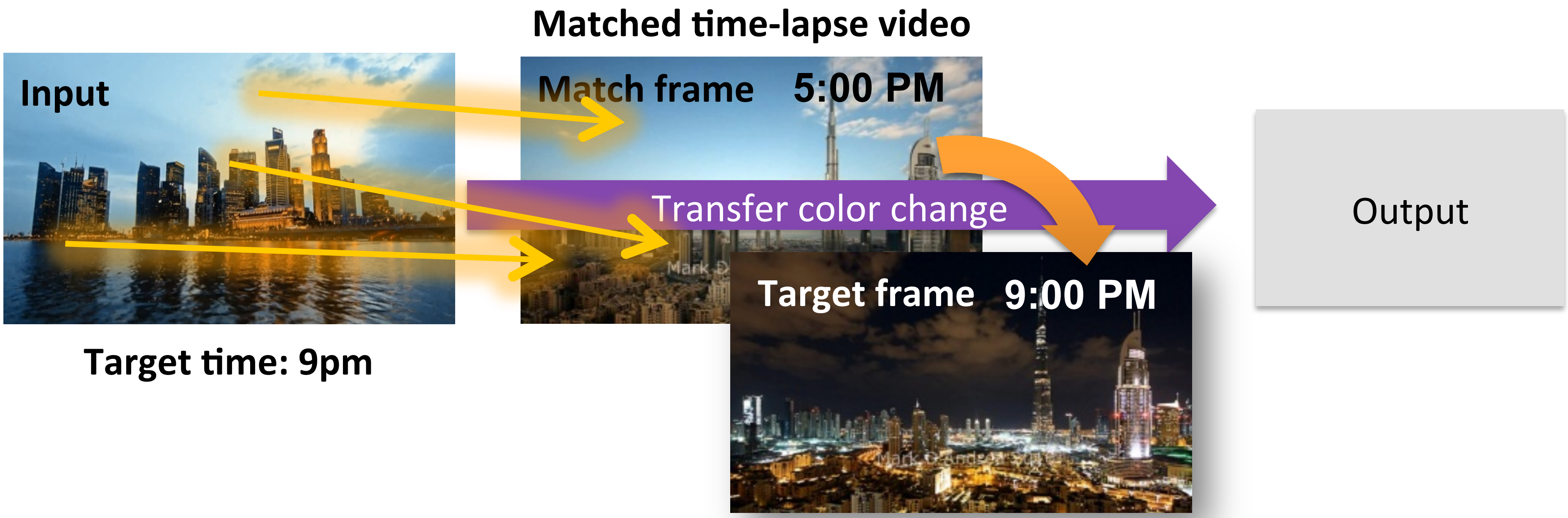
Overview

1. Match input to video from database
2. Transfer color change



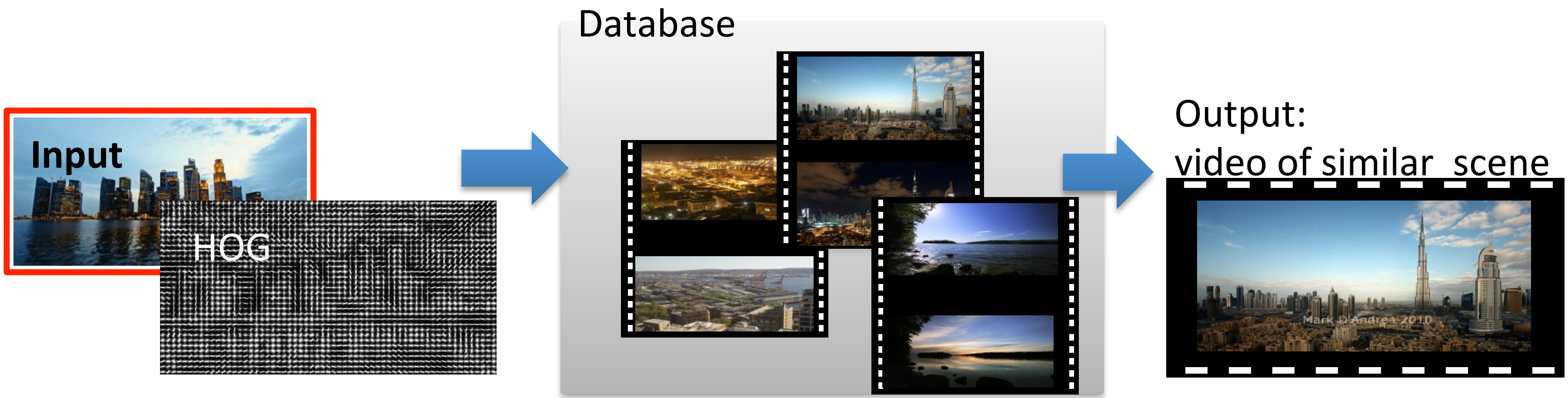
Overview

1. Match input to video from database
2. Transfer color change



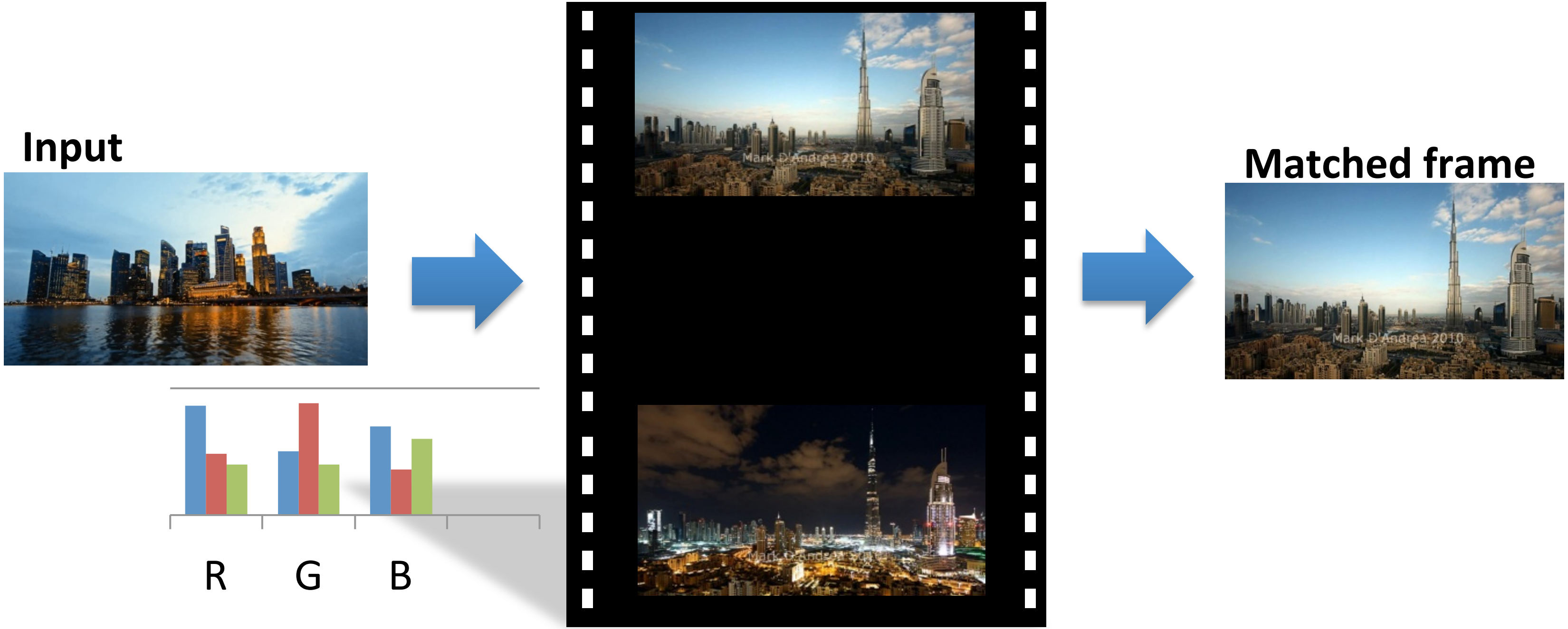
Matching step 1: video level

- Video retrieval with off-the shelf scene matching technique [Xiao et al. 2010]



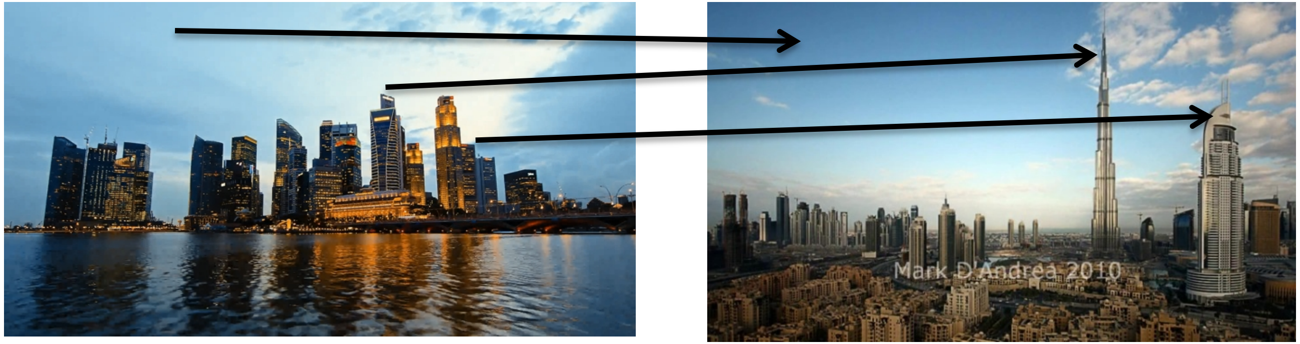
Matching step 2: frame level

- Select the best match frame by color histogram



Matching step 3: pixel level

- Respect scene semantic
  - sky to sky, building to building, etc.



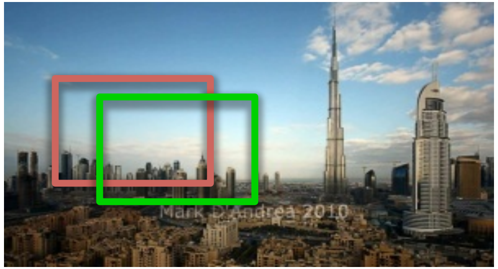
Input   Matched frame

- Dense correspondence using Markov random field

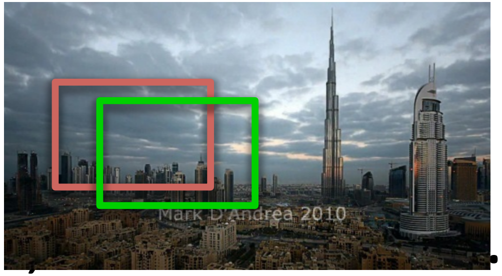
Markov Random Field for dense matching

- Data term: standard L_2 norm
- Regularization term: aggregate over the entire sequence, not just the matched frame
  - Consistency over all time of day

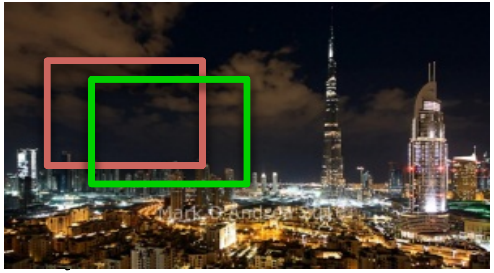
max (



,



, …,



)

Warp the matched frame to the input

- Capture the scene semantics ☺

Input

Matched frame

Warped matched frame

Naïve transfer: warp the target frame

- Using the same correspondence
- The texture in the warped image is wrong

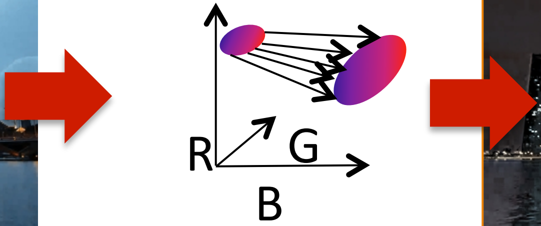
Input

Target frame

Naïve transfer

Our approach: locally affine color transfer

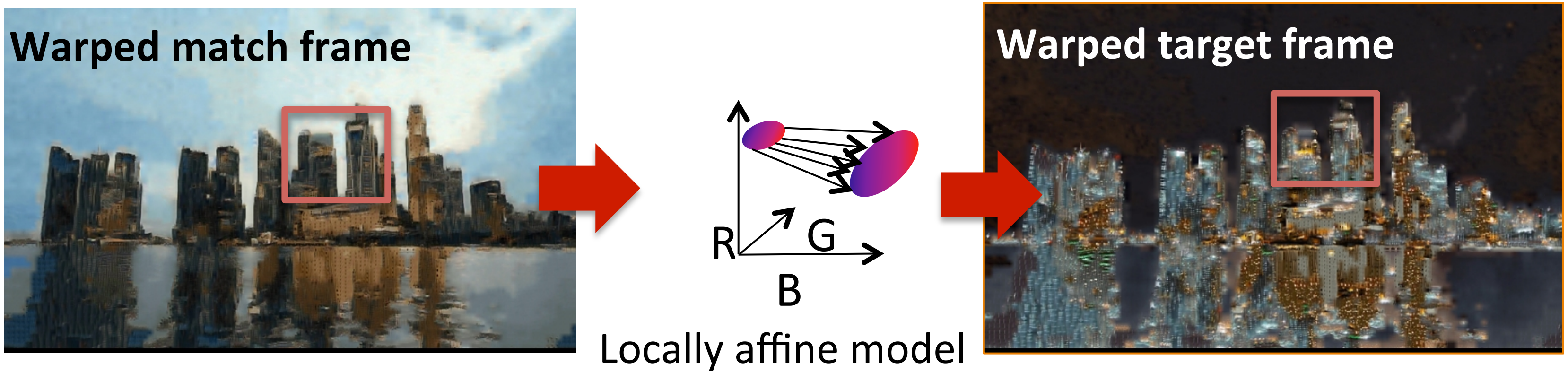
- *Local* to handle complex scenes
- *Affine* color transfer in each patch
  - preserve the structure of the input
  - match ground truth data



Locally affine model

Locally affine model explains the color change of time-lapse data

- In particular, explain the matched and target frame



The transfer needs to be locally affine everywhere

- The patches are overlapping, so we cannot estimate the affine model independently on each patch

Color transfer as an optimization

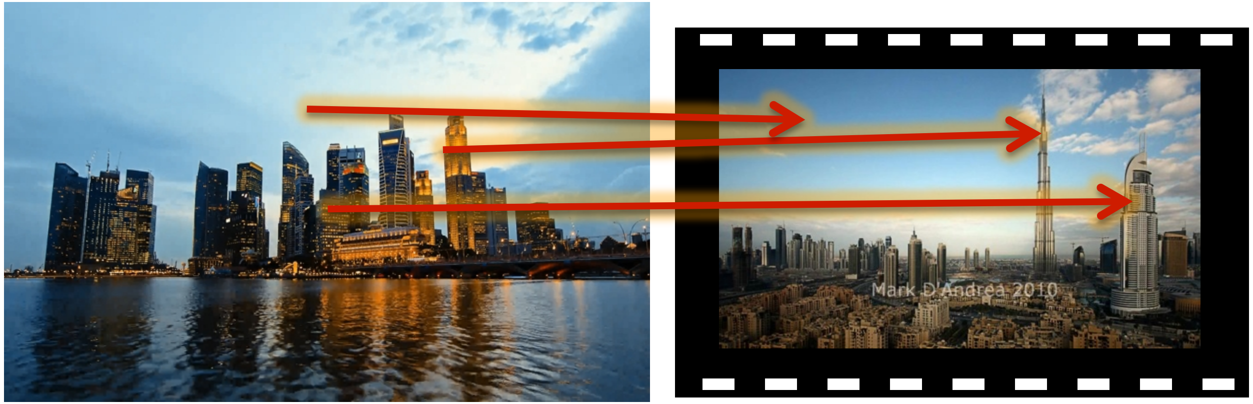
- We are looking for color remapping function
  - Objective #1: explain time-lapse data
  - Objective #2: locally affine everywhere
- We design a least-squares energy
  - Sparse linear system

Recap

Input

1. Matched video

Recap

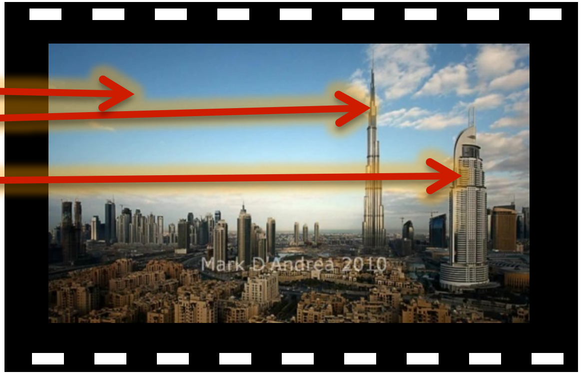


Input

1. Matched video

Recap

Input

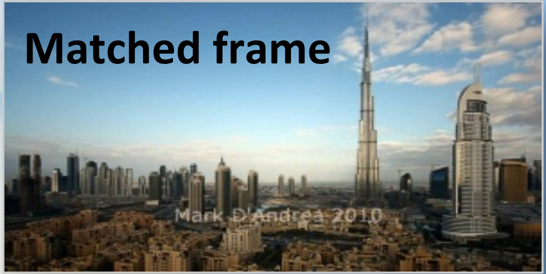


1. Matched video

2. Locally affine transfer

Input at sunset

Input at sunset



Our result at night

Results: four different times of day

Input

Day

Before sunset (golden hour)

After sunset (blue hour)

Night


Day
Before sunset (golden hour)
After sunset (blue hour)
Night
American Legion Post
Weehawken 18 New Jersey
ALL VETERANS WELCOMED
GIANTS

Ground truth validation

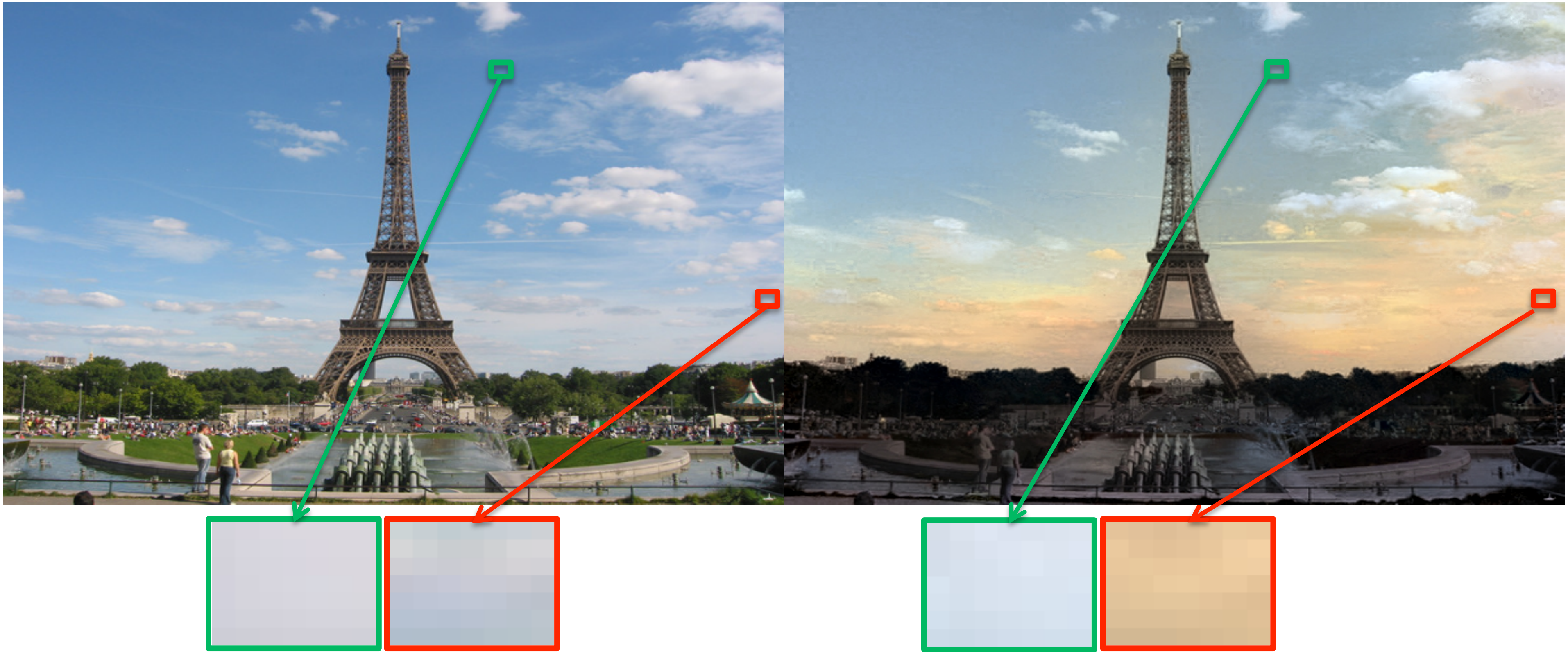

Our result at night


Input at blue hour


Ground truth at night

Our transfer is spatially-variant

Input at day **Output at golden hour**



Our transfer is object-dependent

Input at day **Output at night**

More results: cloudy input

Cloudy input

Output at after sunset

More results

Input at after sunset

Output at night

Mountain view

Input at day

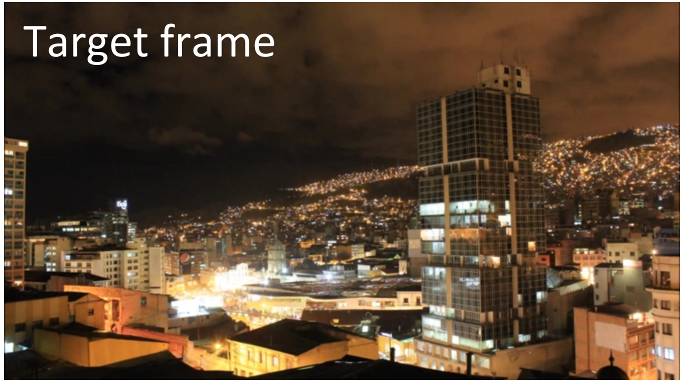
Output at blue hour

Lake scene

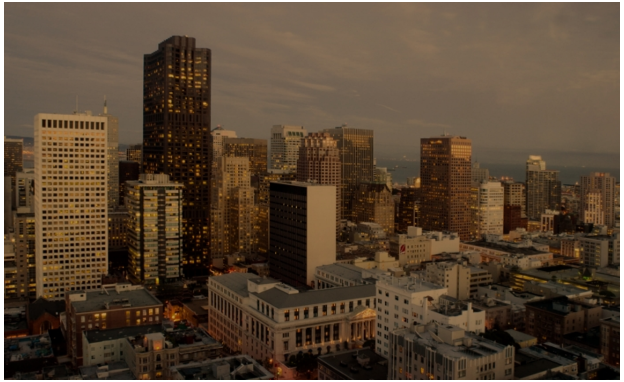
Input at day

Output at night

Comparisons



[Pitié et al. 2005]



[Reinhard et al. 2001]

Our method


Input

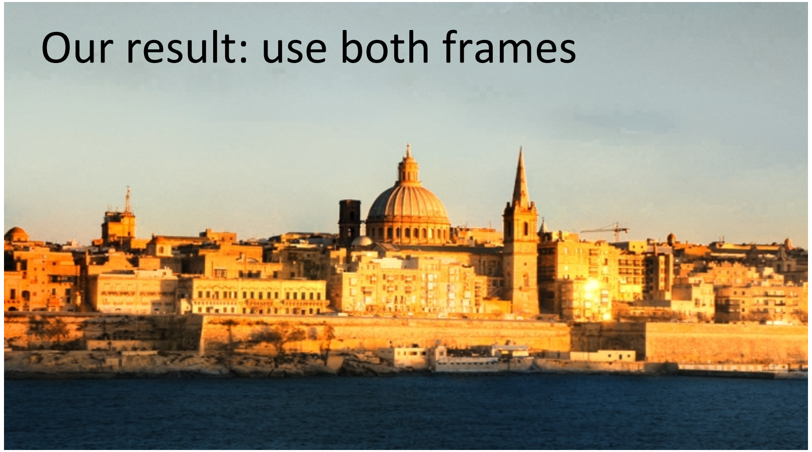

Our method


[Reinhard et al. 2001]


[Pitié et al. 2001]

Color Transform vs Color Distribution

- Our result is more golden



Applications

- Image editing tool

Application: translate the time of day of a painting

Input at day

Output at blue hour

“In the Auvergne”, Jean-Francois Millet

Starry Night at Day?

Limitations

- Dynamic scenes are challenging
- Night-to-day case does not work well

Night to Day

Failed output

Summary on time-lapse hallucination

- Time hallucination: render an image at another time
- Use a time-lapse database + locally affine transfer
- Transfer the color variation

Recently related work : different seasons

- Use time-lapse videos of a year [Laffont et al. 2014]

Input at spring

Winter [Ours, 2013]

[Laffont et al. 2014]

Overview of this talk

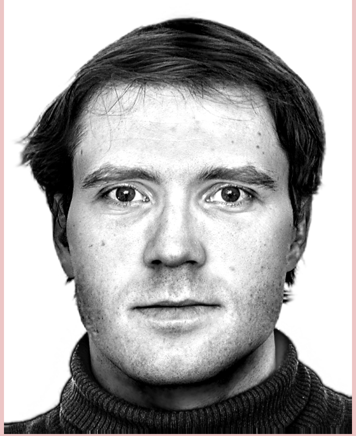
- Time-lapse hallucination (SIGGRAPH Asia 2013)

Input at afternoon

Output at night

- Portrait style transfer (SIGGRAPH 2014)

Input: ordinary portrait



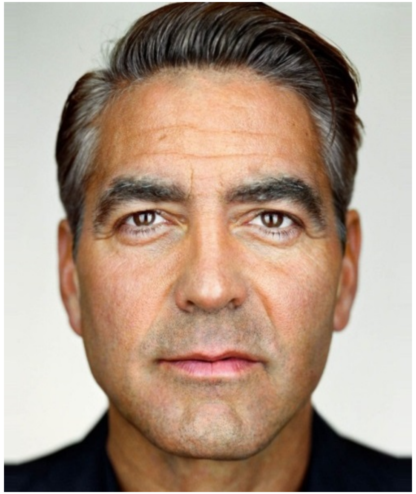
Output: stylized portrait

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

- Global color transfer is not sufficient

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]

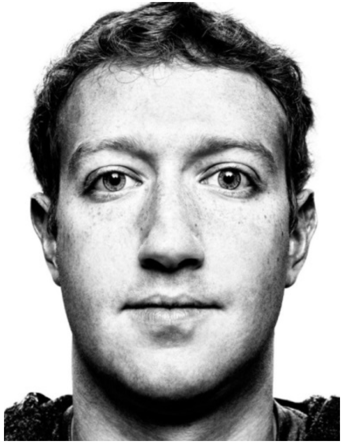
- Do not work as well on portraits

Related work: global transfer

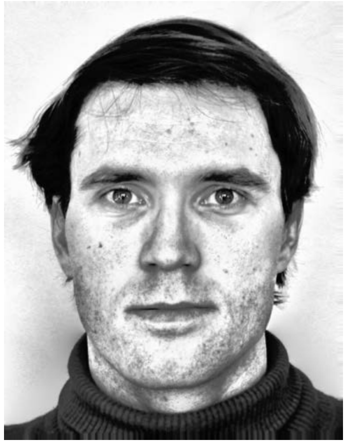
[Bae et al. 2006, Sunkavalli et al. 2010…]

- 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

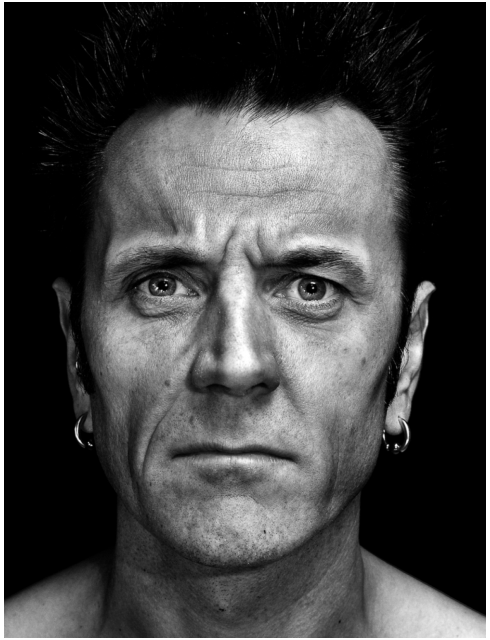
Key idea #1: local transfer

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

Input | Example

Key idea #2: multi-scale transfer

- Textures at different scales are treated differently

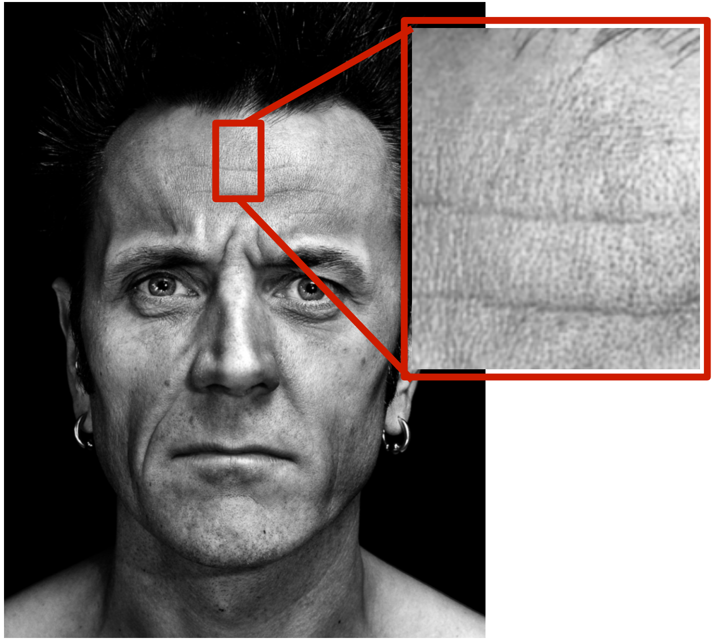


Portrait #1

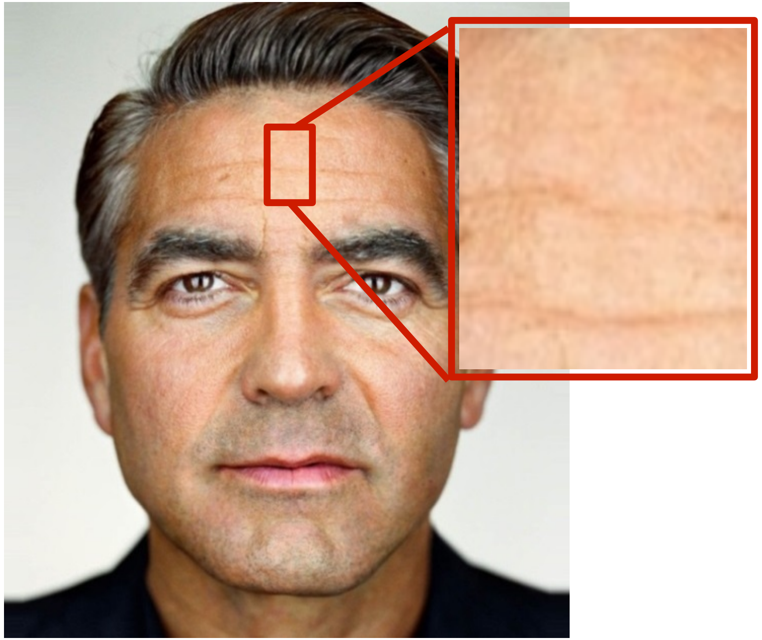
Portrait #2

Key idea #2: multi-scale transfer

- Textures at different scales are treated differently



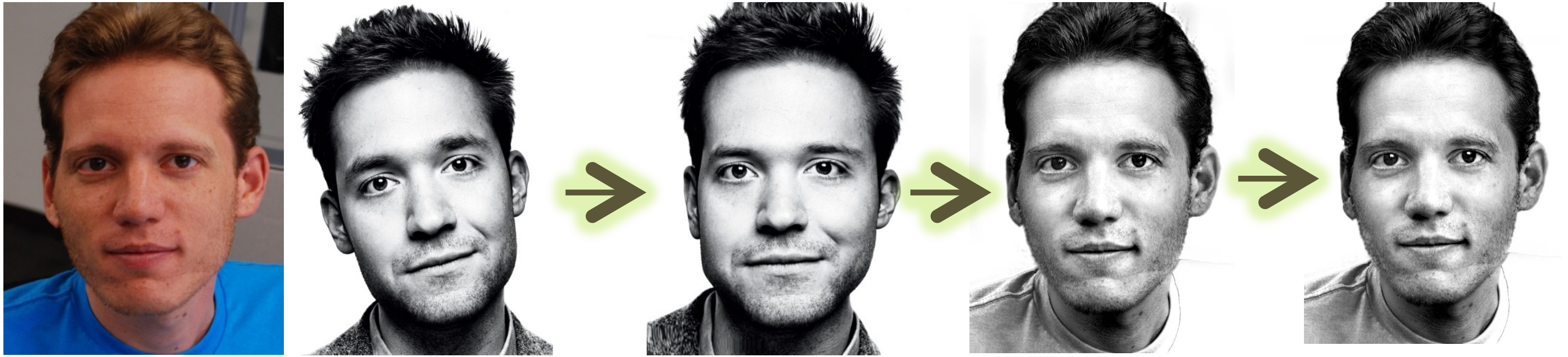
Portrait #1



Portrait #2

Overview of the algorithm

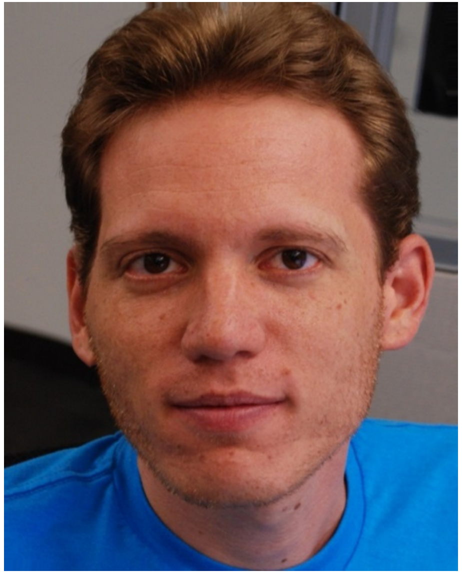
1. Dense matching between the input and example
2. Multiscale transfer of local statistics
3. Post processing on eyes and background



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]

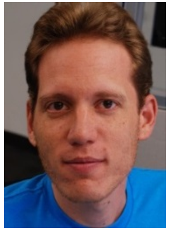


Input

Example

Warped example

Step 2: multi-scale local transfer



Input

Example

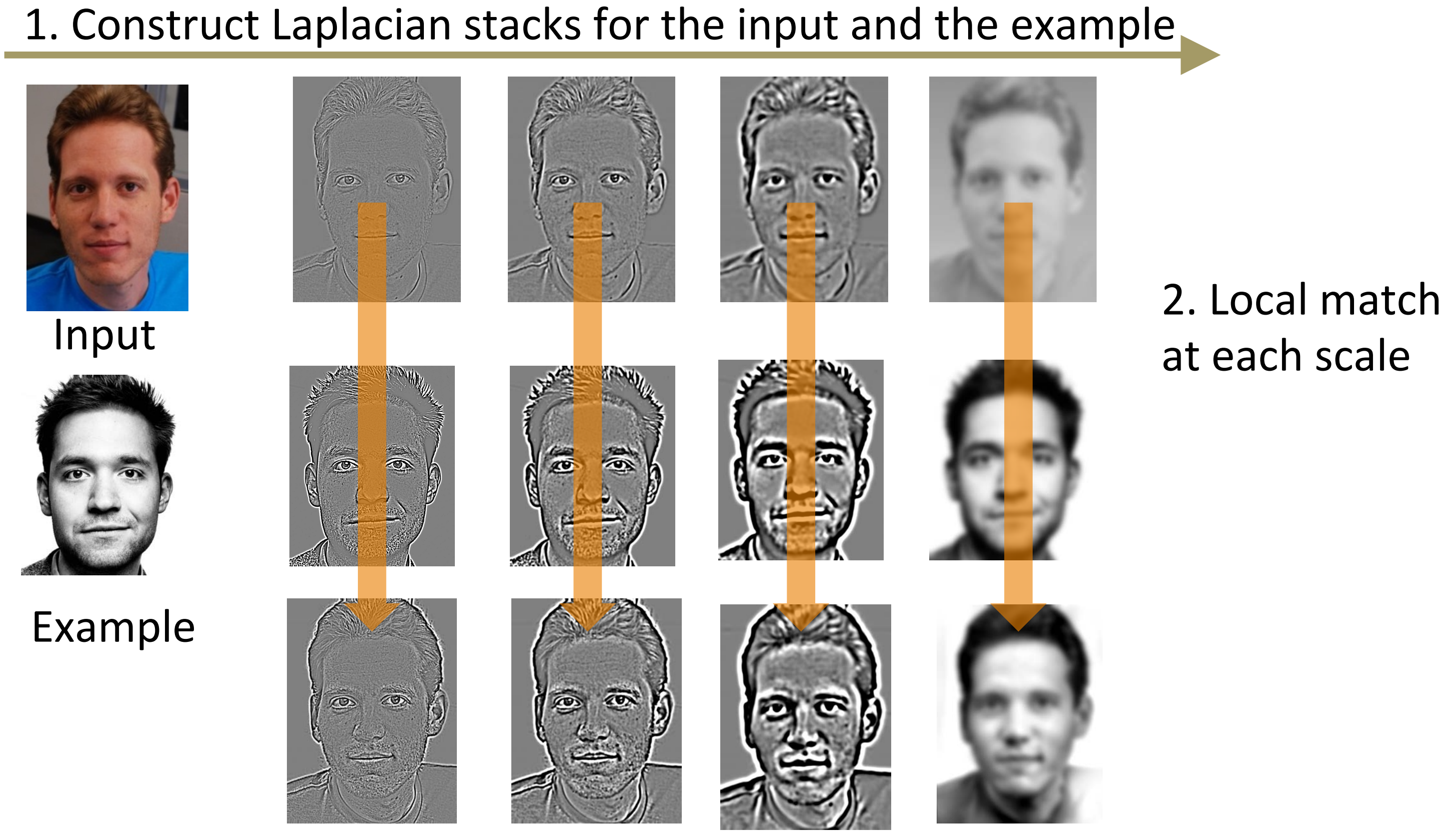
Step 2: multi-scale local transfer

1. Construct Laplacian stacks for the input and the example

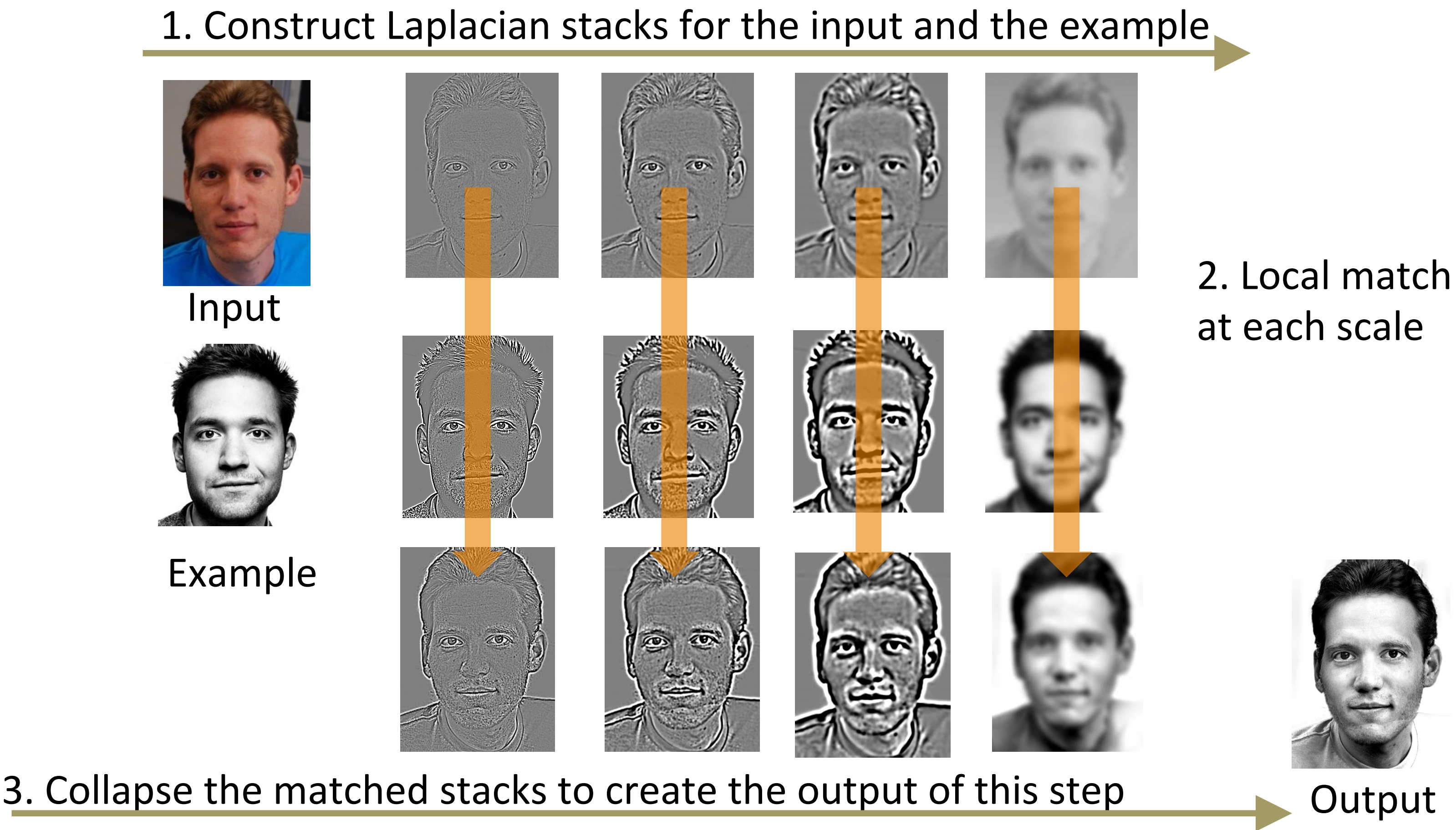
Input

Example

Step 2: multi-scale local transfer

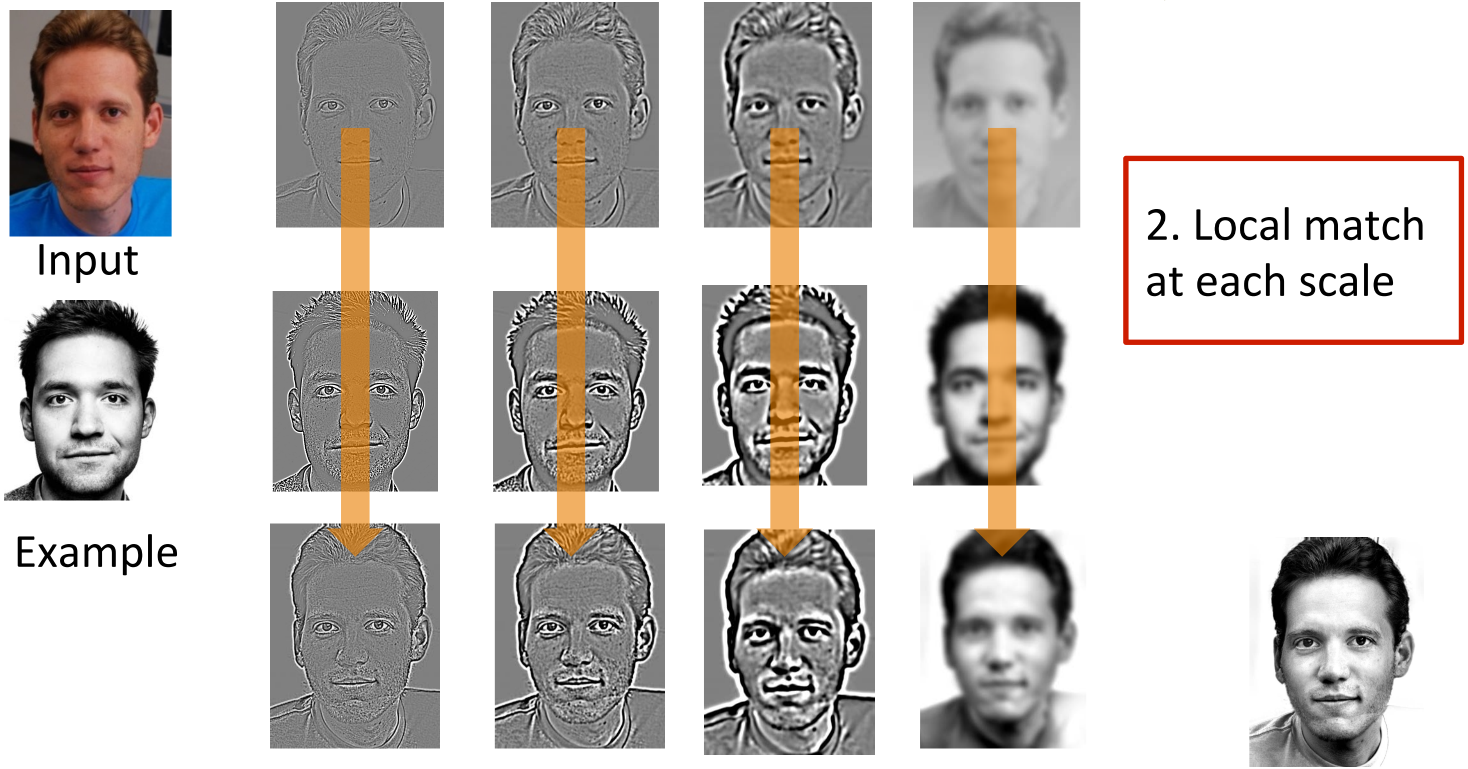


Step 2: multiscale transfer of local statistics



Step 2: multi-scale local transfer

1. Construct Laplacian stacks for the input and the example



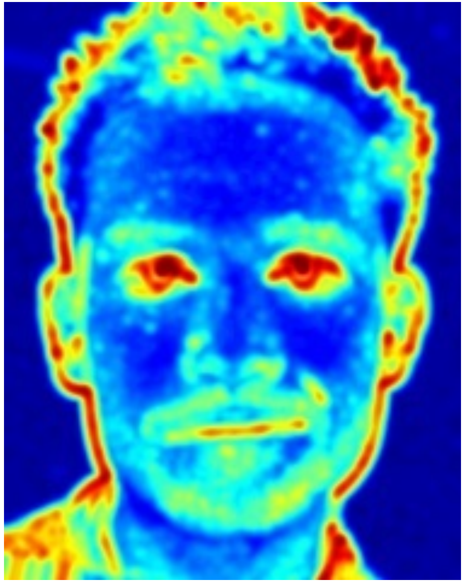
3. Collapse the matched stacks to create the output of this step

Local energy S

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

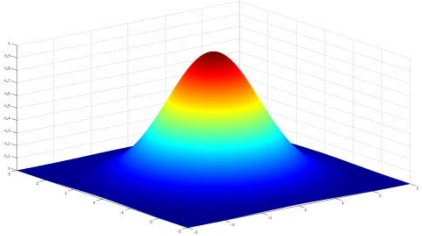
L

Example Laplacian



S

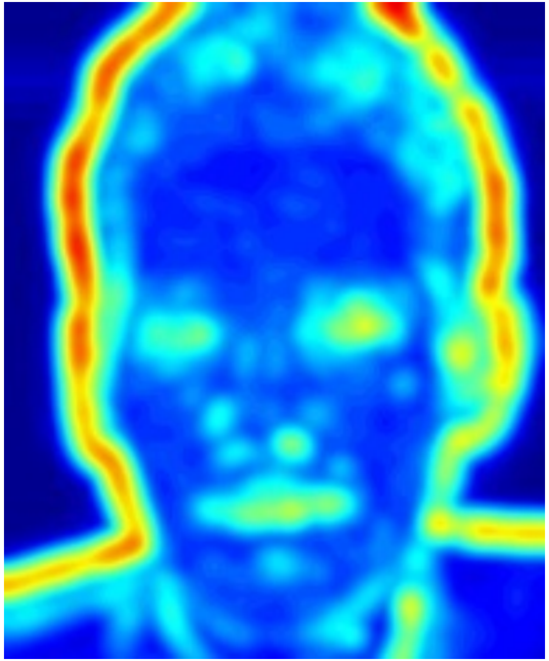
Local energy



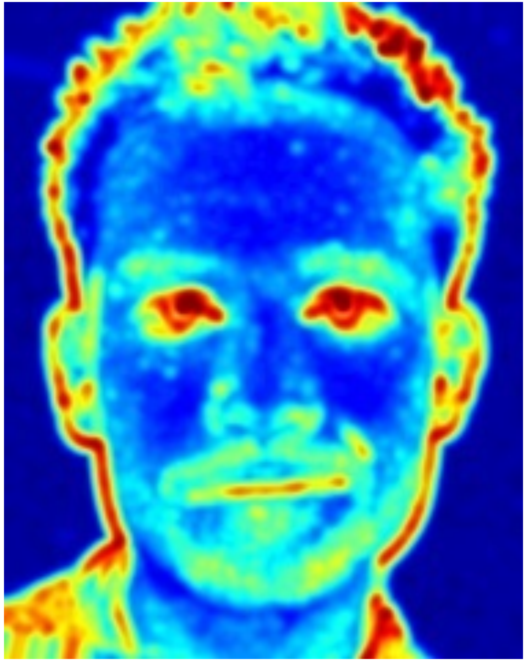
G_ℓ

Gaussian kernel at this scale

At each scale: match local energy



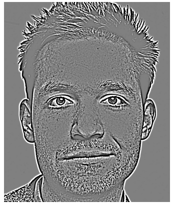
Input energy



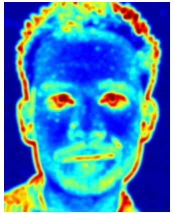
Example energy

At each scale: match local energy

Compute
the gain map

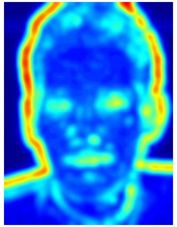


Example Laplacian



Local energy S[E]

Input Laplacian

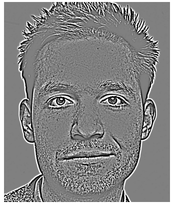


Local energy S[I]

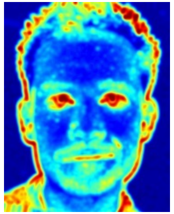
Gain map = $\sqrt{\dfrac{\mathrm{warp}(S[E])}{S[I]}}$

At each scale: match local energy

Compute
the gain map



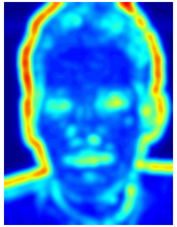
Example Laplacian



Local energy S[E]

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

Input Laplacian



Local energy S[I]

Modulate
the input Laplacian

Input Laplacian

×

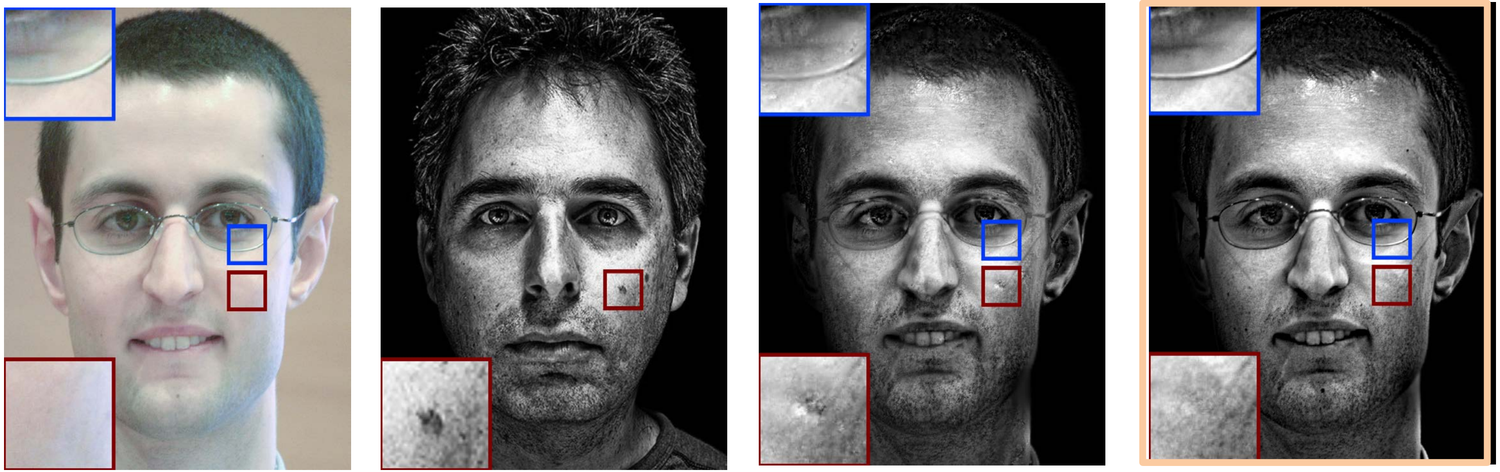
Gain map

=

Output Laplacian

Robust transfer

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

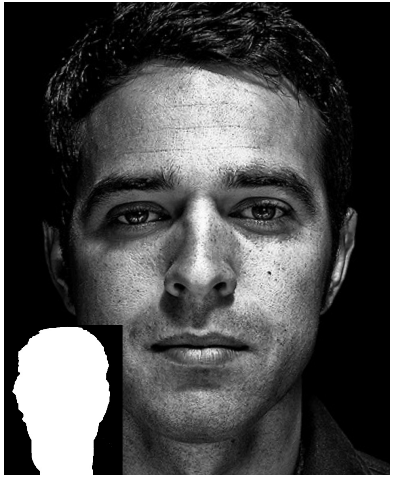


Input Example Without robust transfer Our robust transfer

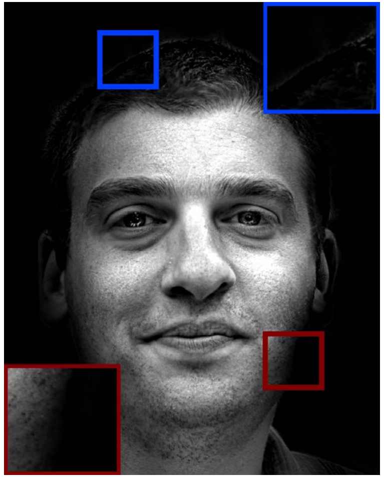
Laplacian using a face mask

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

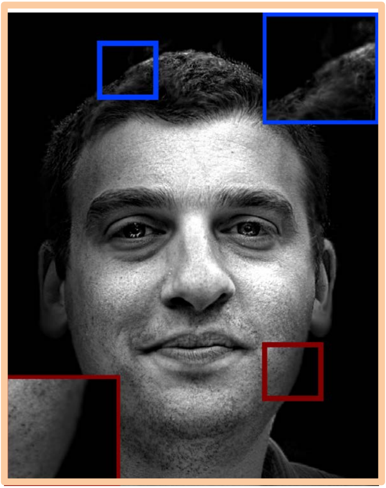
Input



Example



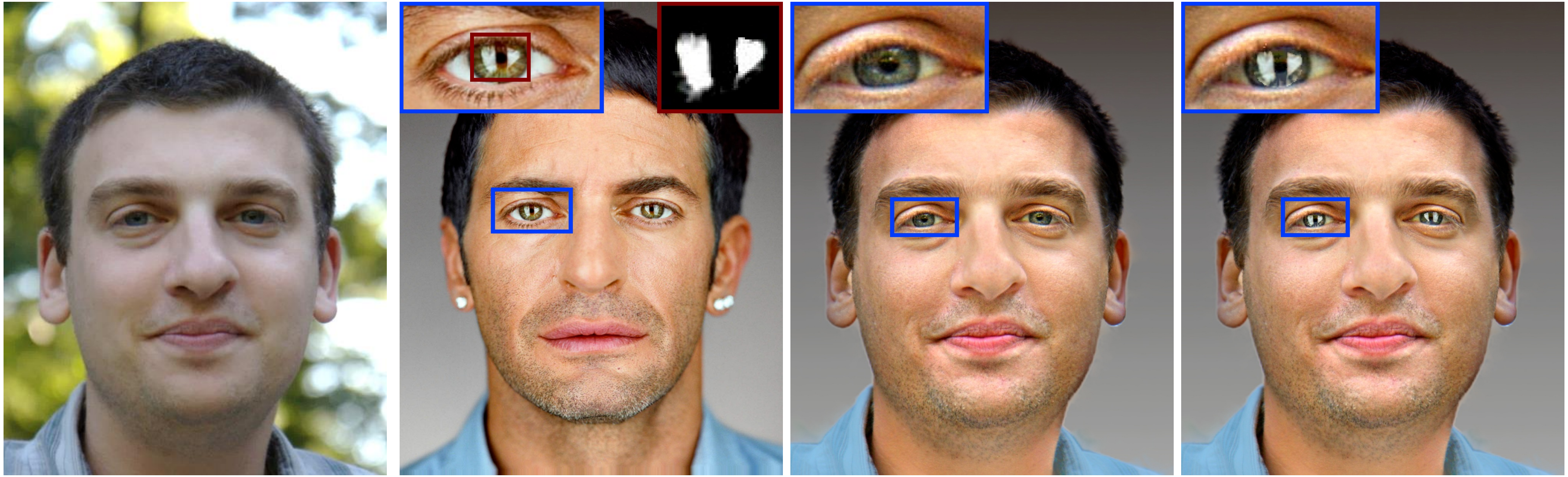
Without using the mask
(the edges disappear)



Our method
(the edges are preserved)

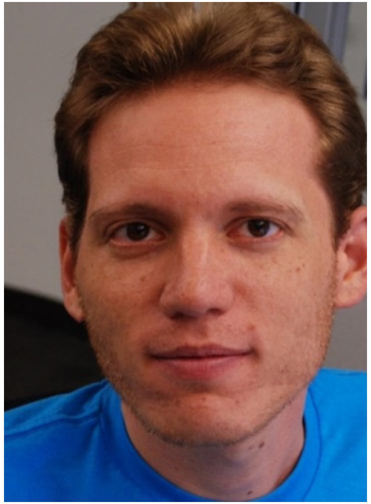
Step 3: post-processing

- Adding eye highlights
- Replacing the background



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

Algorithm recap

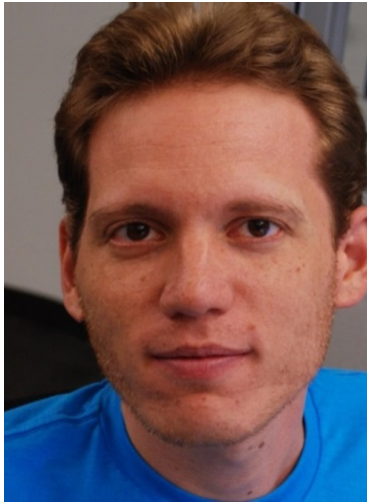


Input

Example

Step 1.
Dense alignment

Algorithm recap



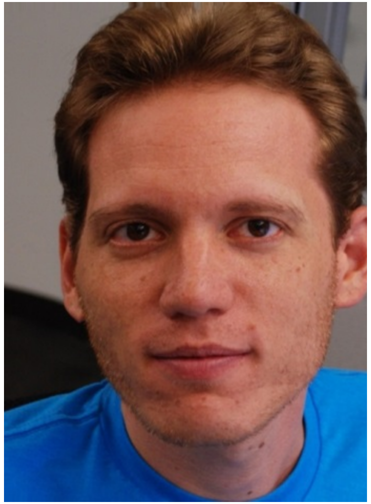
Input

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

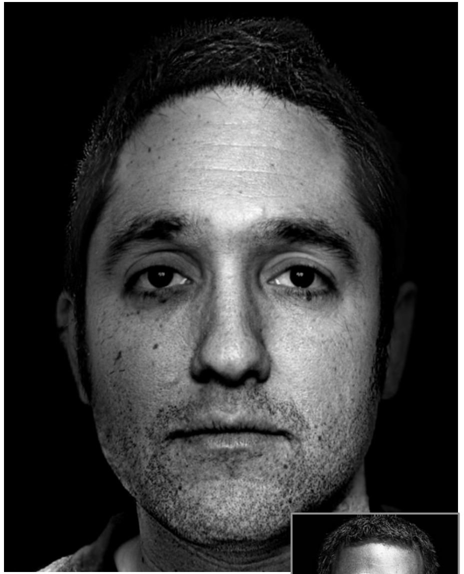
Input

Style transferred results using the top three examples

Results

Examples are shown in the insets

Input



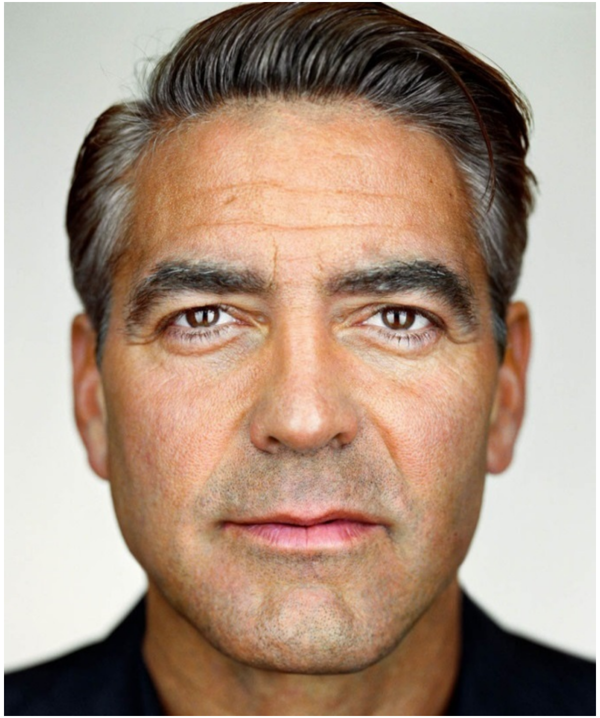
Style 1

Style 2

Style 3

Close-up

Input



Example

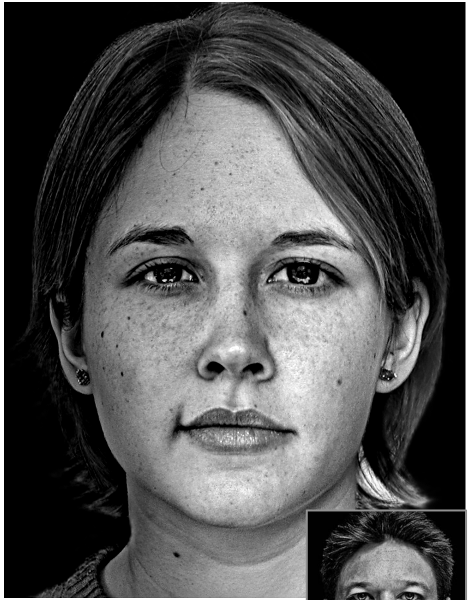
Output

Example

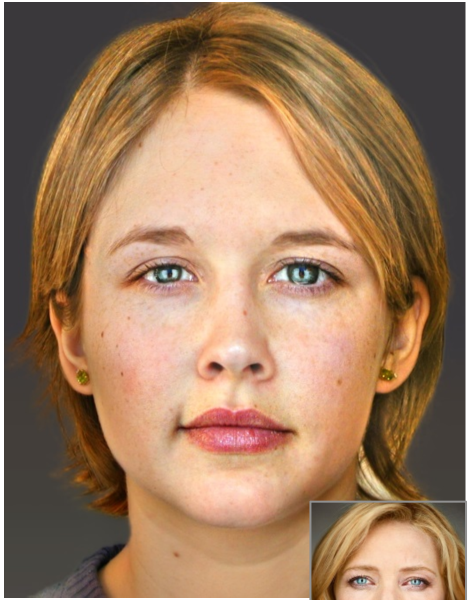
Output

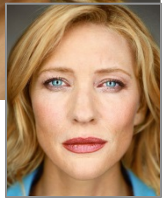
More results

Input



Style 1

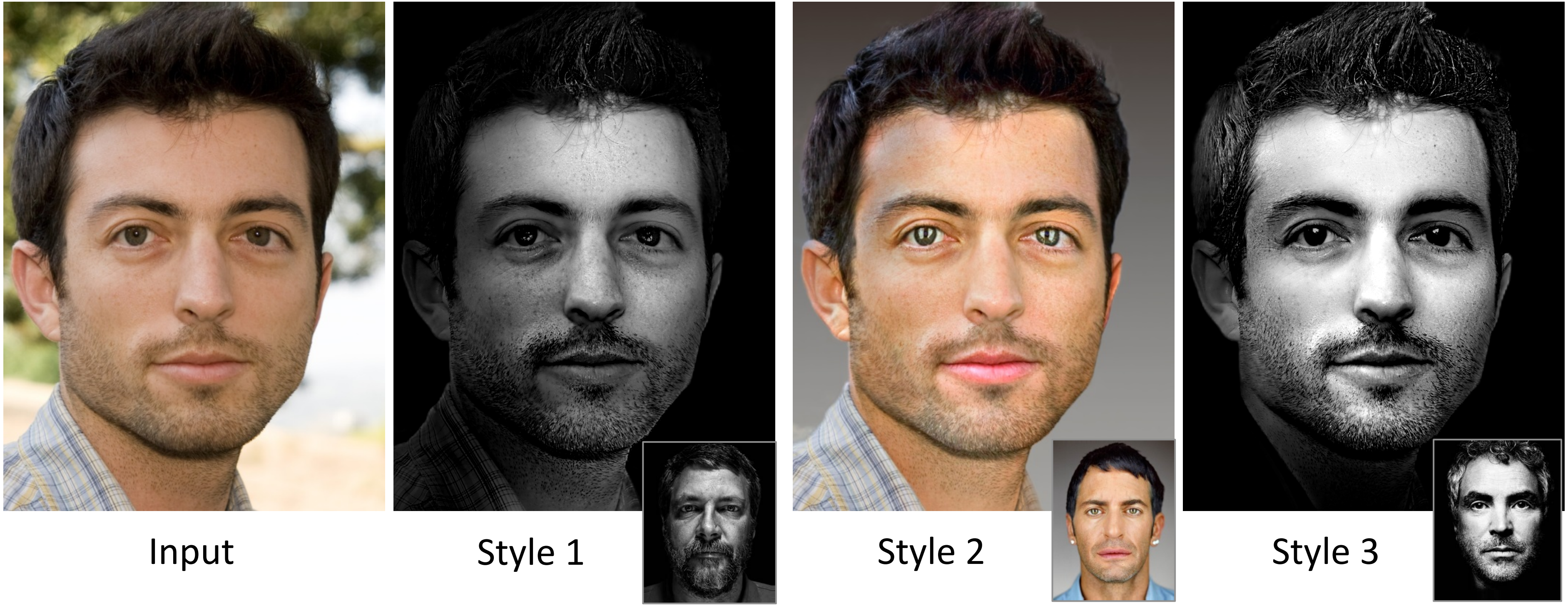




Style 2

Style 3

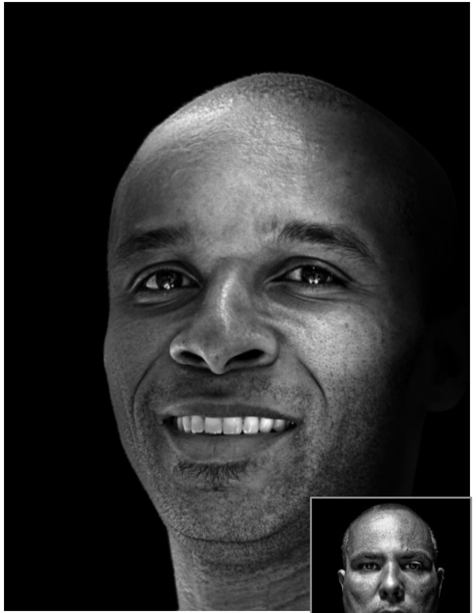
Outdoor input



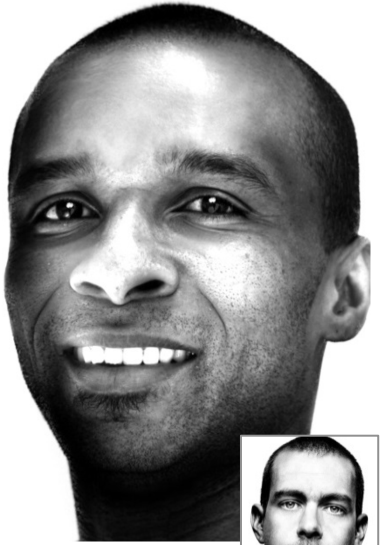
Extra results

Input

Style 1



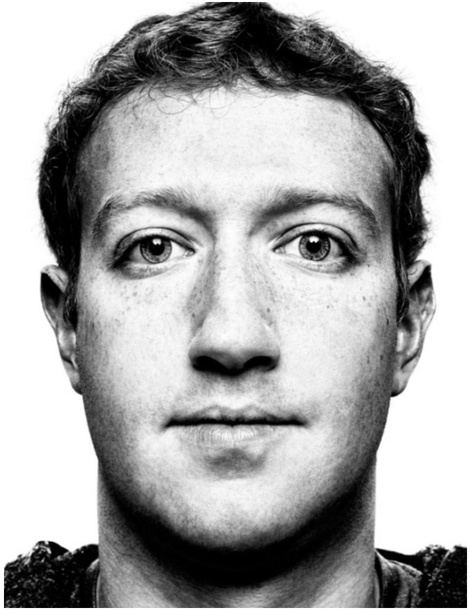
Style 2

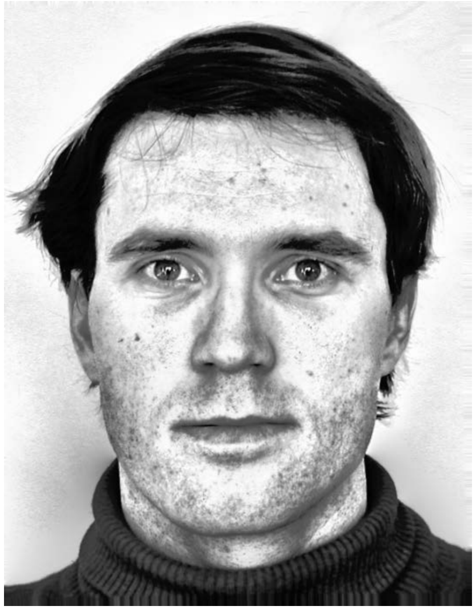


Style 3

Comparisons

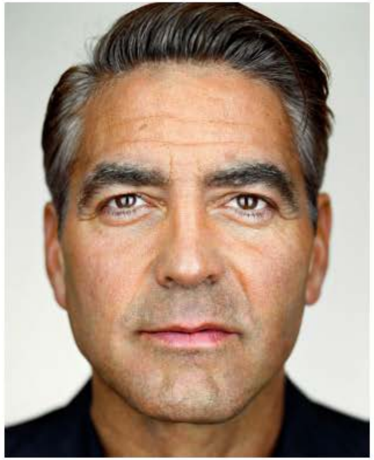

Input

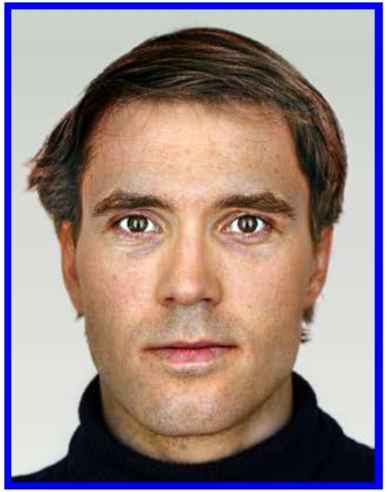

Example


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

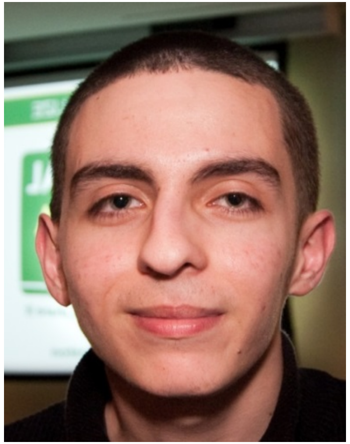
- Matting (face mask) failure

Input

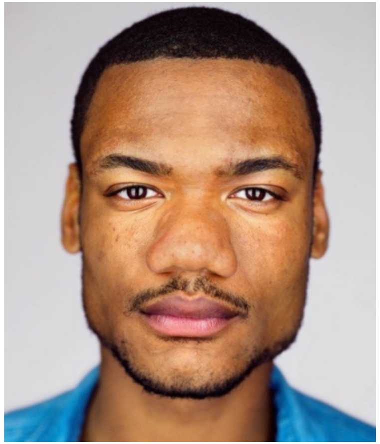
Output

Limitations

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



Input

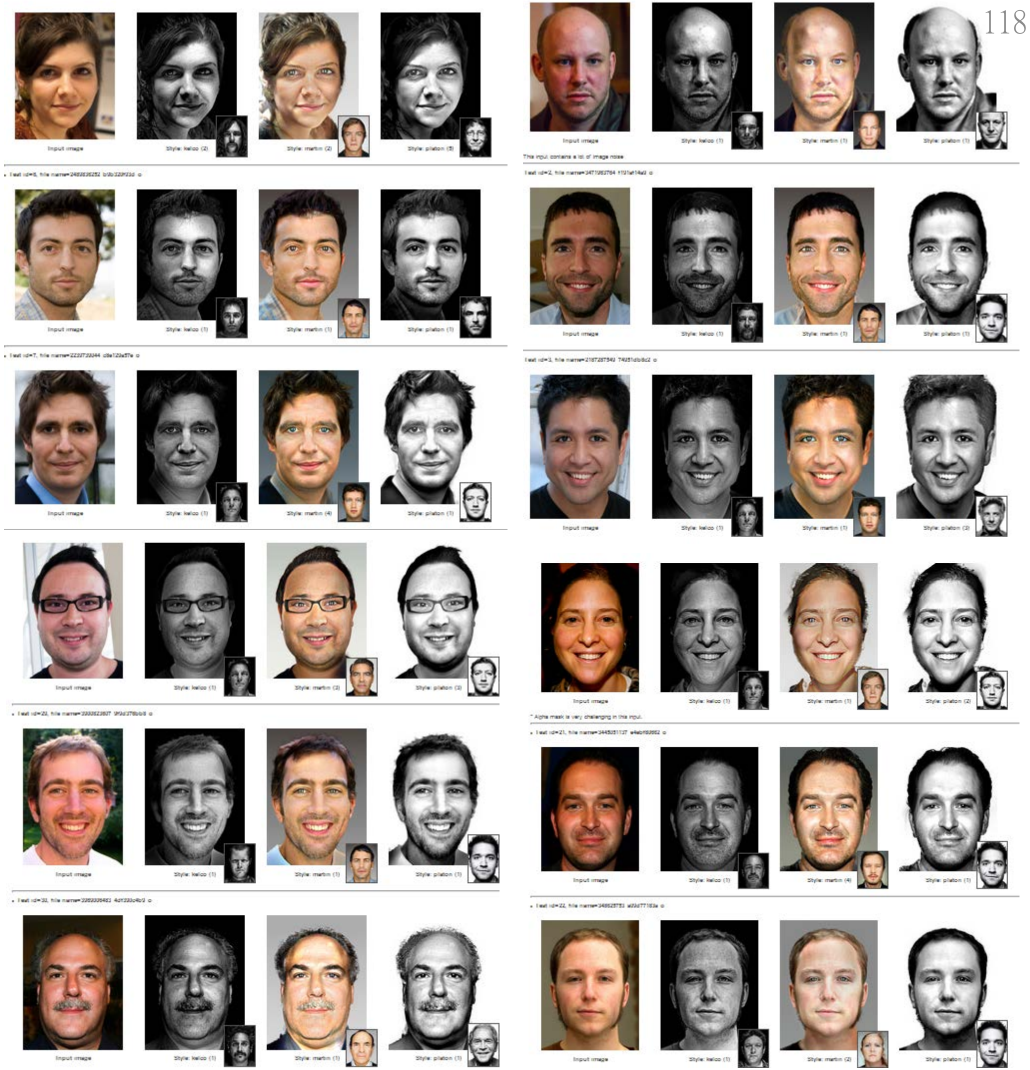


Example

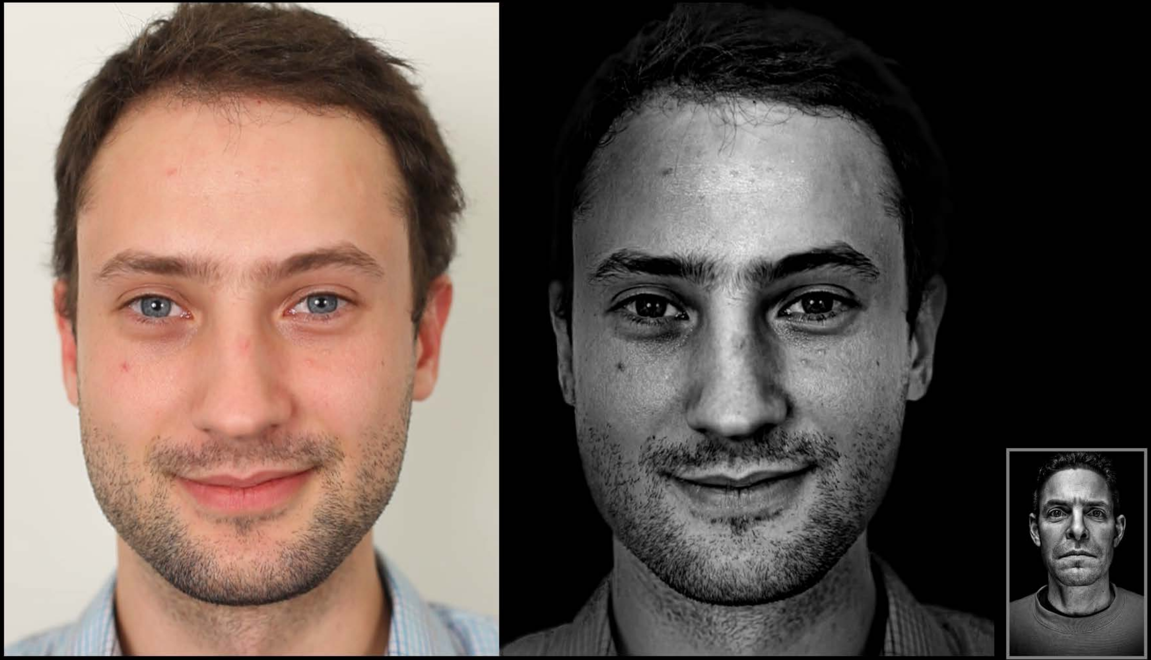
Failure output

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

Summary

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

Input Example Output

Recap of this talk

- Time-lapse hallucination

Input at afternoon

Output at night

- Portrait style transfer

Input: ordinary portrait

Output: stylized portrait

Conclusions

- Dramatic style changes from the example
  - time-of-day, portrait

- Approximate complex physical interactions or creative processes

Key #1: data

- Search for good examples from a database

- "Small data" seems to be already sufficient
  - time-lapse database: 500 videos
  - portrait database: ~50 pics / style

Key#2: local transfer

- A dense correspondence to capture the semantics

- Time-of-day: locally affine transfer
  - exploit color variations

- Portraits: local and multi-scale transfer

Potentials of cloud computing

- Use a shared database

- The transformation is easier than the output image [ongoing work]

  - low-dimensionality of scene appearance variation through different time-of-day

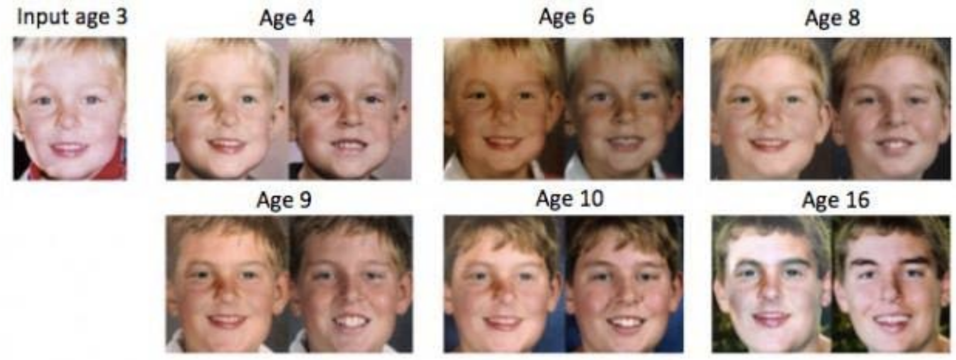
  - low-passed gain maps on portraits

Open questions: beyond graphics?

- We have achieved the visual realism

- Can we extract physical information?

- Portrait: predicting the aging?



[Kemelmacher-Shlizerman et al. 2014]

Acknowledgements

- My advisors and the authors of the two papers in the talk

Fredo Durand

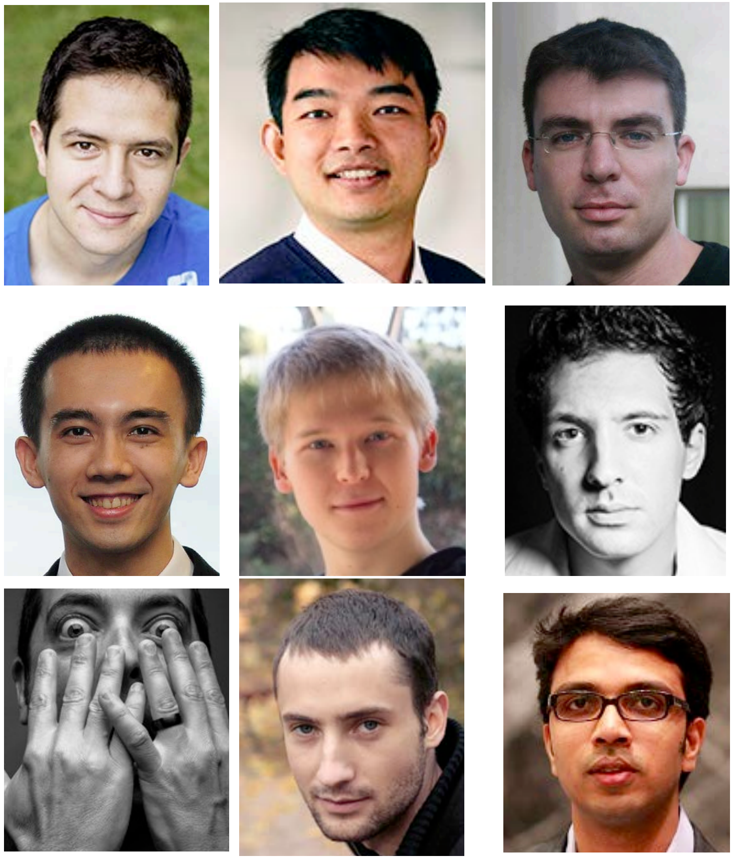
Bill Freeman

Sylvain Paris

Connelly Barnes

Thanks for feedback

Adrian Dalca
Ce Liu
Michael Rubinstein
Pierre-yves Laffont
Jianxiong Xiao
Michael Gharbi
Krzysztof Templin
Kelly Castro
Manohar Srikanth



Acknowledgements

Dilip Krishnan

Sam Hasinoff

Abe Davis

Andrew Adams

Donglai Wei

Neel Joshi

Brian Guenter

Vivek Kwatra

Troy Chinen

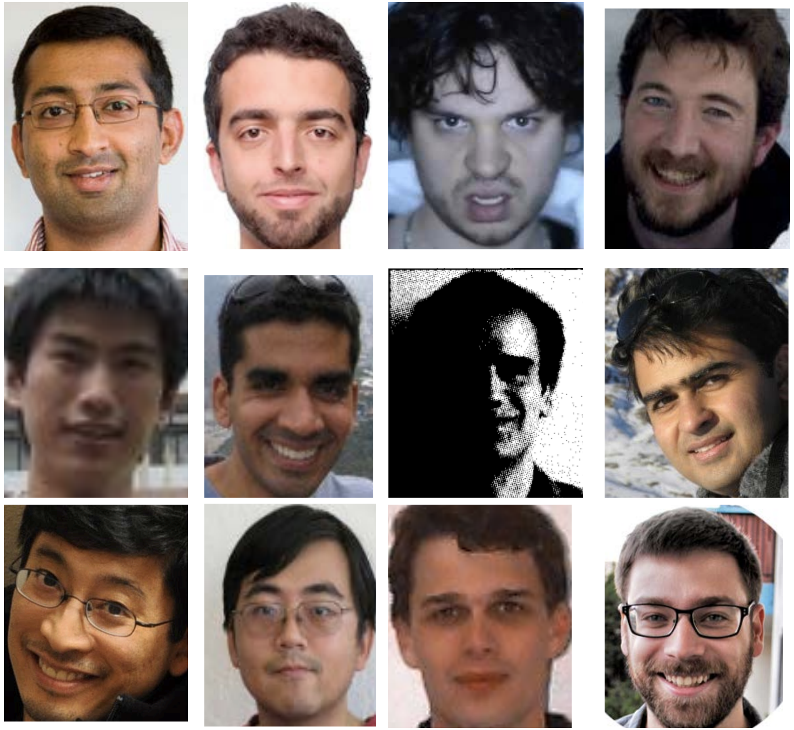
Hui Fang

Sergey Ioffe

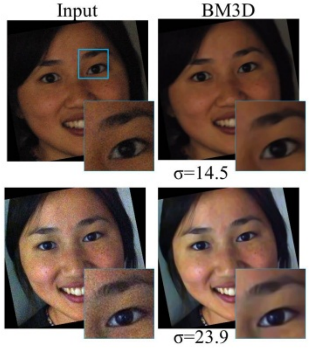
Jon Barron

MIT Graphics Group

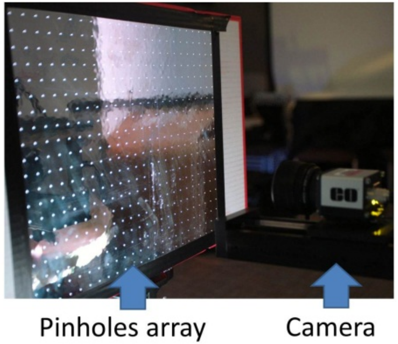
MIT Vision Group



Other projects not in this talk



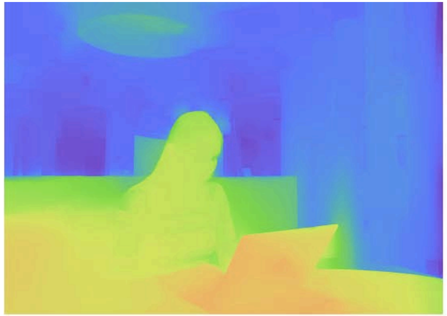
Noise estimation



Lens calibration

Speckle photography

Reflection removal



Efficient stereo

Acknowledgements

Prof. Fredo Durand

Prof. Bill Freeman

Prof. Wojciech Matusik

Dr. Sylvain Paris

Thank you

- Time-lapse hallucination

Input at afternoon

Output at night

- Portrait style transfer

Input: ordinary portrait

Output: stylized portrait

Face noise estimation using personal photos [ICCV2013]

- Use face correspondences to estimate noise levels

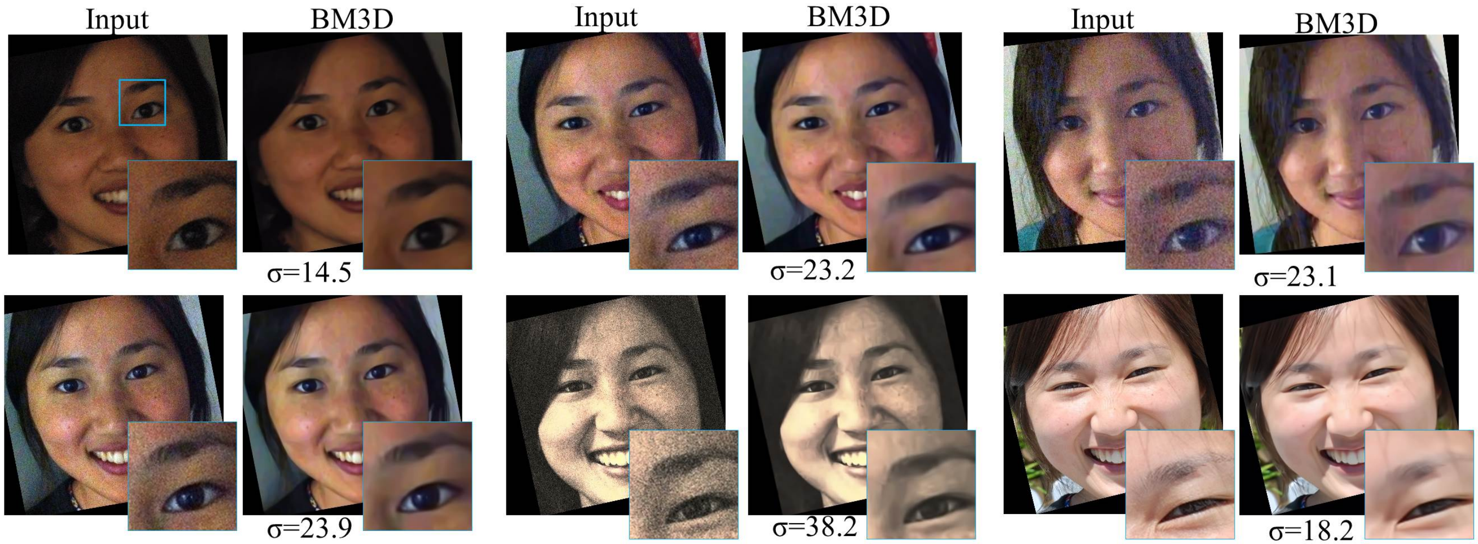


Image enhancement using calibrated lens [ECCV2012]

- Lens has spherical and chromatic artifacts

(b) Lens prescription and simulations

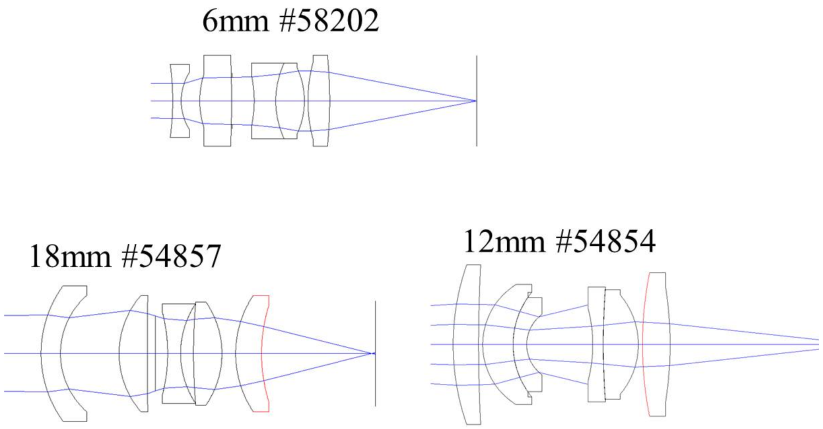
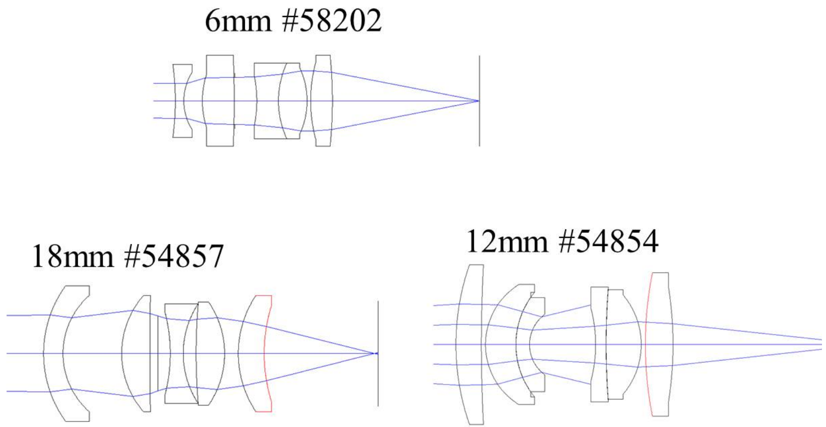
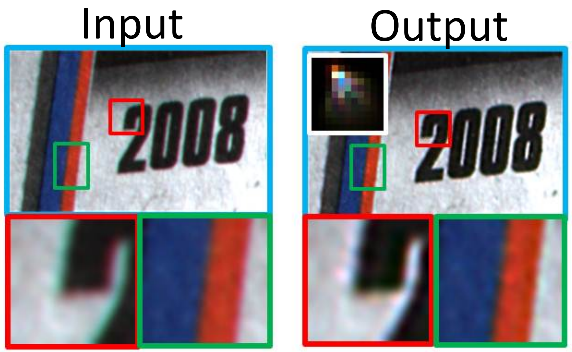


Image enhancement using calibrated lens [ECCV2012]

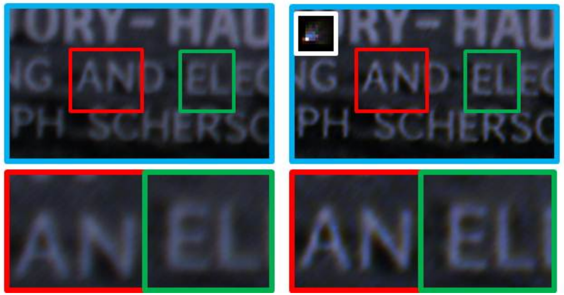
- Lens has spherical and chromatic artifacts

(b) Lens prescription and simulations





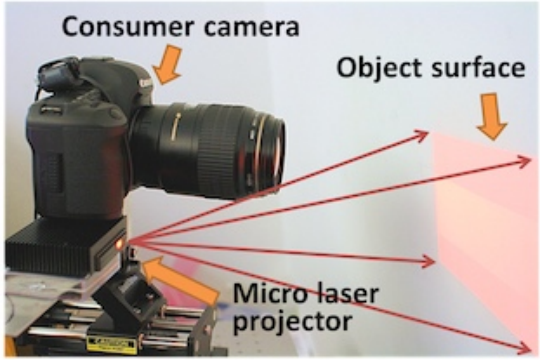
Remove chromatic aberration



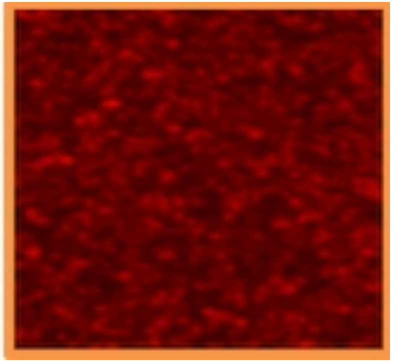
Remove spherical aberration

Laser speckle photography [CVPR 2012]

- Surface tampering detection



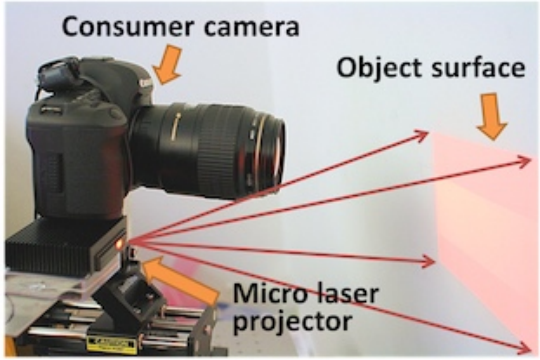
Camera + laser projector



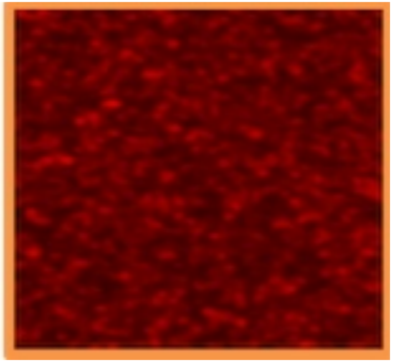
Speckle image

Laser speckle photography [CVPR 2012]

- Surface tampering detection



Camera + laser projector



Speckle image

Invisible tampering detected by us

Reflection Removal [under review for CVPR 2015]

- Key idea: ghosting cues in reflections

Input spoiled by reflections of text

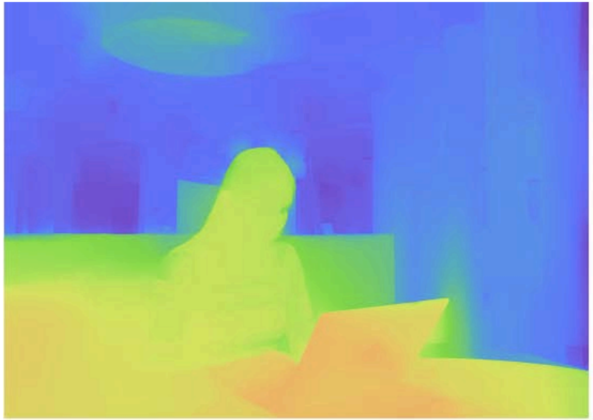
Recovered Transmission

Recovered Reflection

Efficient Stereo for Refocusing [under review for CVPR 2015]

- Key idea: regularize on bilateral grid

Input stereo pair
(left view)



Reconstructed depth map

Shallow depth-of-field effect

Application: lighting transfer

Code and data are available

- Matlab code
- Flickr evaluation dataset

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

Reference

[1] YiChang Shih, Sylvain Paris, Connelly Barnes, William T. Freeman, Fredo Durand, “Style Transfer for Headshot Portraits”, SIGGRAPH 2014

[2] YiChang Shih, Sylvain Paris, Fredo Durand, William T. Freeman, “Data-driven Hallucination of Different Times of Day from a Single Outdoor Photo”, SIGGRAPH Asia2013

[3] YiChang Shih, Vivek Kwatra, Troy Chinen, Hui Fang, Sergey Ioffe, “Joint Noise Level Estimation from Personal Photo Collections”, ICCV 2013

[4] YiChang Shih, Brian Guenter, Neel Joshi, “Image Enhancement using Calibrated Lens Simulations”, ECCV 2012

[5] YiChang Shih, Abe Davis, Samuel W. Hasinoff, Fredo Durand, William T. Freeman, “Laser Speckle Photography for Surface Tampering Detection”, CVPR 2012

[6] YiChang Shih, Dilip Krishnan, Fredo Durand, William T. Freeman, “Reflection Removal using Ghosting Cues”, in submission to CVPR 2015

[7] Jonathan T. Barron, Andrew Adams, YiChang Shih, Carlos Hernandez, “Fast Bilateral-Space Stereo for Synthetic Refocus”, in submission to CVPR 2015

[8] L. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman and B. Schölkopf, W. Freeman, “Seeing the Arrow of Time”, CVPR 2014

Previous work: style transfer

- Make the input look like the provided example

Input

Example

Output by Bae et al. [2006]

Global transfer [Reinhard et al. 2001, Bae et al. 2006, …]

- Work well on simple scenes

Input

Example

Output [Pitié et al. 2007]

Conclusion on local transfer

- Achieve dramatic style changes
- Require examples of similar semantics
- Benefit from a large image database
  - eg., the Internet

Changing time-of-day

- Entail advanced operations
  - brushes, layers, curves…

Original

Retouched

Hard problem

- Physical simulation would need complex modeling

Input image at day time

Ground truth image at night

Hard to retouch

Still non-trivial

Input

Example

Color matching (failed)
[Reinhard et al. 2001]

The semantic information is overlooked!

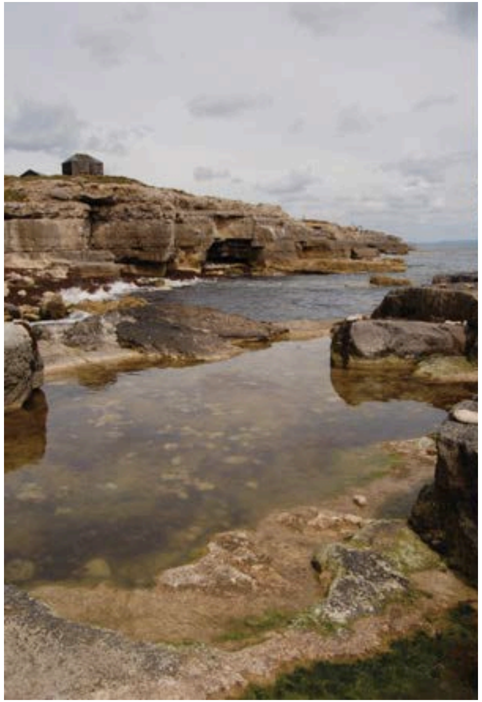
Why care about retouching?

- Conveying a unique mood
- Make a picture more memorable
- Impress people

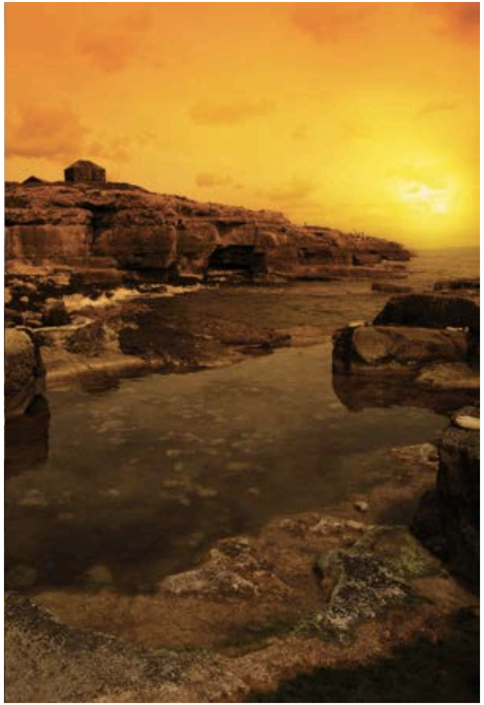
Original

Retouched

Photograph retouching

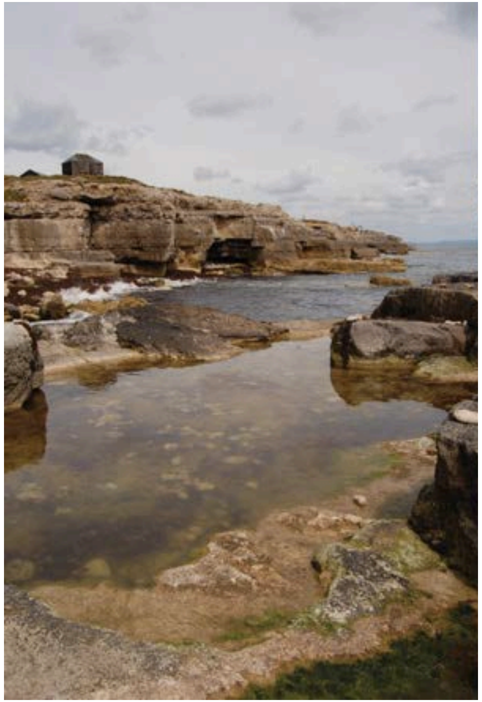


Before

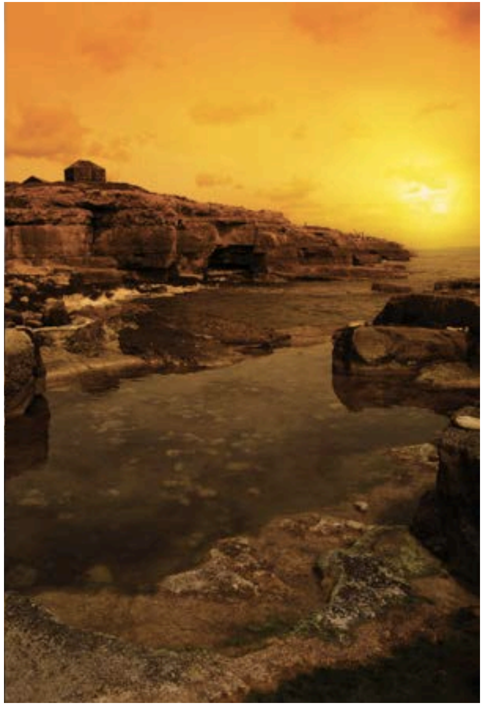


After

Photograph retouching



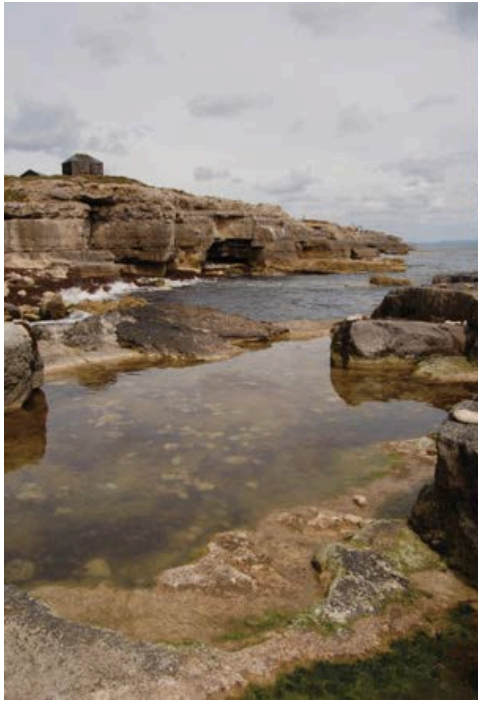
Before



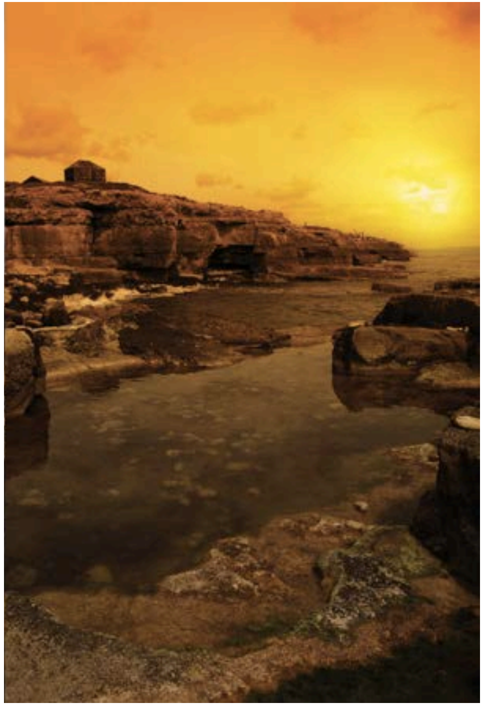
After

- Convey the mood
- Make it memorable
- Impress people

Photograph retouching



Before



After

Light room

Instagram

Photoshop

Auto enhance
(Facebook)

Auto awesome
(Google)

Gimp

Tedious works

- Time consuming
  - 10-20 minutes with tutorials
  [Berthouzoz et al. 2009]

- Let's automate them