Data-driven Photographic Style using Local Transfer

by

YiChang Shih

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

Author . . . .
Department of Electrical Engineering and Computer Science
Feb 27, 2015

Certified by . . . .
William T. Freeman
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by . . . .
Frédo Durand
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses

Data-driven Photographic Style using Local Transfer

by

YiChang Shih

Submitted to the Department of Electrical Engineering and Computer Science
on Feb 27, 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

After taking pictures, photographers often seek to convey their unique moods by altering the style of their photographs, which can involve meticulous contrast management, lighting, dodging, and burning. In this sense, not only are advanced photographers concerned about their pictures' styles; casual photographers who take pictures with cellphone cameras also process their pictures using built-in applications to adjust the image's luminance, coloring, and details. In general, photographers who stylize pictures give them new, different visual appearances, while also preserving the original content. In this context, we investigate problems with novel image stylization, including reproducing the precise time-of-day where the lighting and atmosphere can make a landscape glow, and making a portrait style resemble that created by a renowned photographer. Given an already captured image, however, automatically achieving given styles is challenging. In fact, changing the appearance in a photograph to mimic another time-of-day requires the analysis and modeling of complex 3-D physical light interactions in the scene, while reproducing a portrait photographer's unique style require computers to acquire artistic tastes and a glimpse of the artist's creative process. In this dissertation, we sidestep these AI-complete problems to instead leverage the power of data. We exploit an image database consisting of time-lapse data describing variations in scene appearance during the course of an entire day, and stylish portraits that are already deliberately processed by artists. To leverage these data, we present new algorithms that put input images in dense and local correspondence with examples. In our first method, we change the time-of-day with a single image as the input, which we put in correspondence with a reference time-lapse video. We then extract the local appearance transformations between different frames of the reference, and apply them to the input. In our second method, we transfer the style of a portrait onto a new input by way of local and multi-scale transformations. We demonstrate our methods on public datasets and a large set of photos downloaded from the Internet. We show that we can successfully handle lightings at different times of day and styles by a variety of different artists.

Thesis Supervisor: William T. Freeman
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Frédo Durand
Title: Professor of Electrical Engineering and Computer Science

Thesis Reader: Wojciech Matusik
Title: Professor of Electrical Engineering and Computer Science

Thesis Reader: Sylvain Paris
Title: Doctor, Research Scientist (Adobe Inc.)

Acknowledgments

I would like to express my immense gratitudes to my advisors Prof. Frédo Durand and Prof. Bill Freeman, for their guidances and encouragements over my Ph.D. life. Not only high-quality research, I learned from them how to collaborate with other people, and inspire colleagues novel research ideas. This dissertation would not be possible without their advices. I would also like to thank Dr. Sylvain Paris, a long term collaborator whom I have been working with for all the works in this dissertation. Without his keen eyes, the results presented in this dissertation would not be satisfying. I also want to thank Prof. Connelly Barnes, from whom I work with through the portrait project. I also thank Prof. Wojciech Matusik for his insightful feedback on this dissertation.

During my master study, three summer internships, and final year in MIT, I am lucky to closely work with talented people: Dr. Sam Hasinoff, my colleague Abe Davis, Dr. Neel Joshi, Prof. Brian Guenter, Dr. Dilip Krishnan, Dr. Vivek Kwatra, Dr. Troy Chinen, Dr. Jon Barron, Dr. Andrew Adams, MIT Ph.D. student Donglai Wei, on individual projects separate from this thesis [95, 108, 109, 110]

During my research, Dr. Ce Liu, Dr. Michael Rubinstein, Dr. Pierre-yves Laffont, and Prof. Jianxiong Xiao provide insightful discussions and helps on setting up experiments. Adrian Dalca, my colleague in MIT, generously provides his time-lapse sequence collections that inspire the time hallucination work. Michael Gharbi and Krzysztof Templin kindly agree to be my portrait models, and lead great pictures in this dissertation. I also thank Kelly Castro for his thoughtful feedback from a professional photographer's opinion. I also thank both MIT Computer Graphics and Computer Vision group, who offer me a unique opportunity to learn from both areas.

I acknowledge Irwin Mark Jacobs and Joan Klein Jacobs Presidential Fellowship on my first year, fundings from NSF No.0964004 and NSF CGV-1111415 (Analyzing Images Through Time), and financial supports from Quanta Computer and Adobe System.

Finally, I would like to thank my parents, Tung-Hai and Li-Yng, and my brother Yi-Liang, for their life-long encouragements on any choice I made. They taught me to see the reflection of my mind. I will always be grateful to my dearest family.

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

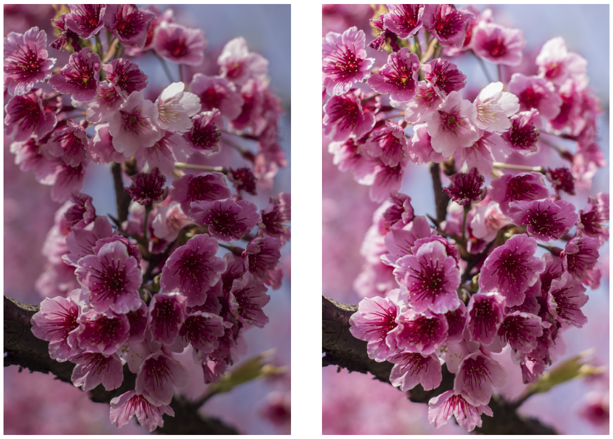
List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction



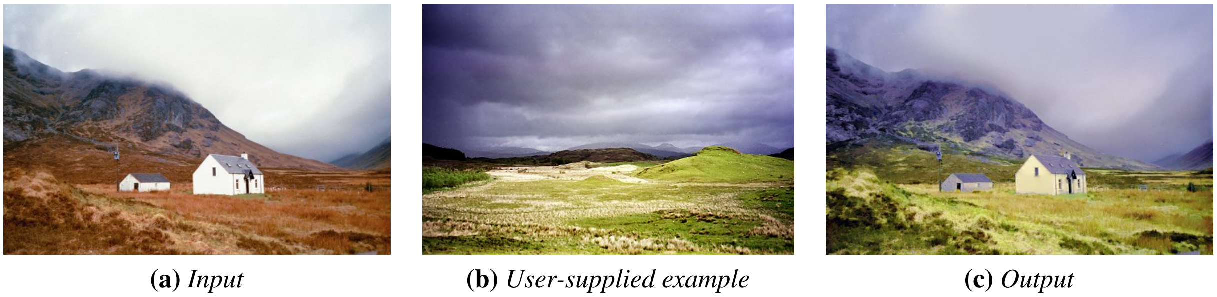
(a) *Input* **(b)** *Stylized output*

Figure 1-1: *A simple example of image stylization by brightening the under-exposed input in (a).*

Enthusiastic photographers transform a scene into an art form by giving styles to their photos. Guided by experienced hands, a creative mind breaks the physical limitations of lens and sensors by carefully managing the picture's contrast, detail, color, and tone until its messages are faithfully conveyed through the pixels. Stylizing a picture can include a wide range of image processing tasks. It can work as simply as modifying the global luminosity of a picture to match the ideal brightness in the photographer's mind. Figure 1-1 restores the picture by brightening the under-exposed input that was erroneously determined by the camera. High-dynamic range (HDR) tone mapping demonstrates a more sophisticated example of image stylization. To truthfully reproduce a scene on a display limited by the low-dynamic range, HDR tone mapping techniques bring out the details from dark regions, while carefully handling bright regions like sky or highlights to avoid over-exposures. For the majority of camera users who take photos with cellphones, creative filters in phone apps like Instagram[1] enable automatic stylization by just a tapping motion.

From a technical point of view, to stylize an input picture, photographers preserve the image content from the input, and produce an output by giving it a new visual appearance. Examples

[1] `http://instagram.com/`



(a) *Input* **(b)** *User-supplied example* **(c)** *Output*

Figure 1-2: *An example of style transfer algorithm by Pitié* et al. *[97]. The algorithm renders an output (c) with the color style from a user-supplied example (b), while preserving the original content in an input (a).*

include photo white balancing, where colors are modified to account for light and human perceptions, and detail enhancement, where image contrasts are restored to compensate for the intrinsic blur common in imperfect lenses, among other things. In these examples, the input and output appear different photographic look while keeping exactly the same content[2]. People stylize pictures for many reasons. Advanced photographers stylize their pictures to convey the unique mood at the moment when shooting the photos. Stylized photos appear to be more memorable. For people in love with social web sites, sharing stylized photographs becomes a modern way to impress friends.

It is nonetheless challenging to stylize a picture. Without a rigorous definition in mathematics, style is merely an elusive concept relying on subjective judgement and difficult for computers to understand. To address the problem, the seminal work by Freeman and Tenenbaum [121] proposes an algorithm to take an input image and separate the style from the image content. Inspired by the fact that human eyes have the great ability to recognize styles from an image, their work employs a bilinear model to mimic the separation process in human brains. Their work demonstrates synthesis of new pictures by modifying their styles, while retaining the original content. The authors apply the synthesis to face relighting, where each relit face corresponds to a new style. Similarly, Bae *et al.* [6] employ a bilateral filter to decompose an input into detail and base layer, and output a different

[2] Modifying input content is beyond this dissertation, e.g., removing unwanted objects or adding accessories in portraits.

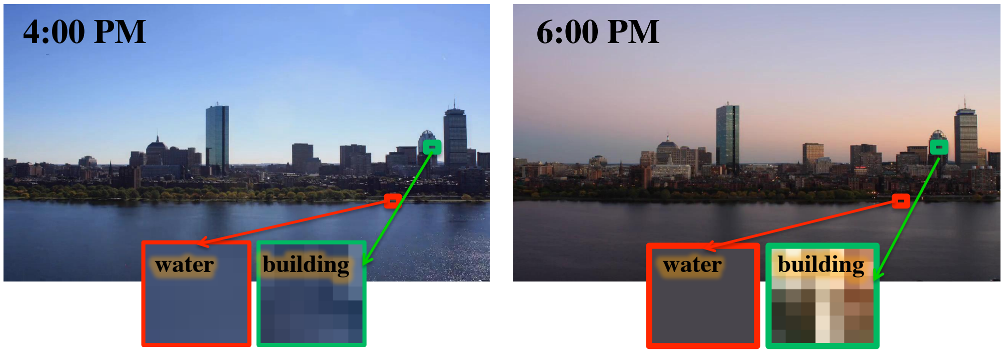
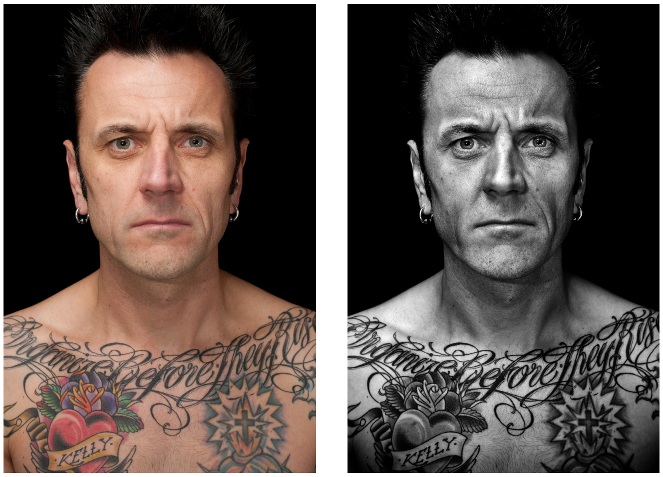


Figure 1-3: *Scene appearance variation through different times of day depends on materials and complicated physical interaction, which makes the time-of-day problem challenging. Image courtesy of Adrian Dalca.*

style by altering the detail layer. The resulting clarity in the output reproduces the style created by Ansel Adams, a famous photographer known for his black-and-white landscape photography. Bychkovsky *et al.* [16] resort to a machine learning technique to learn how photographers stylize their pictures. Based on the content of an input image, their trained classifier predicts the suitable adjustments on luminance and color channels to stylize the image. All these methods retain the content from the input, and give the output a better photographic look. To provide users intuitive controls on image stylization, researchers have proposed the idea of style transfer. In addition to an input picture, a style transfer algorithm takes a user-supplied example image, and outputs a picture with a look similar to that of the example. Figure 1-2 illustrates the spirit of style transfer. Given an input, a style transfer algorithm renders an output with the color style from a user-provided example, while preserving the content from the input image. State-of-the-art style transfer algorithms globally match the example histogram [96, 98], or low-order statistics [100] to produce the output. Previous studies have successfully applied style transfer algorithms to adjusting color, detail, and contrast [6, 45, 97, 100, 118, 120] on simple scenes like landscape photos, where the color distributions of input photos are easy.

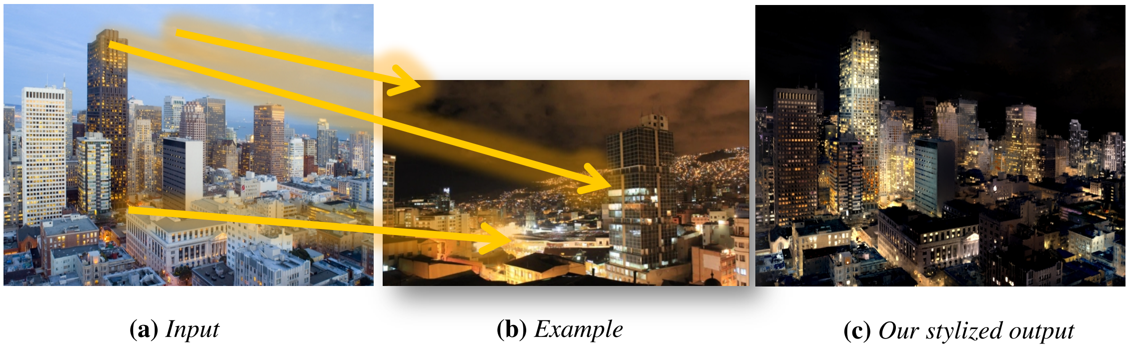
In this dissertation, we pursue the notion of style versus content to address new problems, including rendering a different time-of-day from an input picture. Figure 1-3 shows two pictures



(a) *Input* **(b)** *Manually stylized output*

Figure 1-4: *Photographers convey a unique mood by giving an output a different image style. Image courtesy of Kelly Castro.*

taken at the same scene, from the same camera viewpoints, but at different times of day, one in the afternoon and the other one at sunset. Both images characterize the same image content, showing the same sky, buildings, and river, located at identical positions, but with different color styles. We would like to output the picture at the right in Figure 1-3 by giving the left one a color style plausible at the new time-of-day. Changing time-of-day is a challenging problem, since predicting scene appearance variations would require knowing the materials of every location in a scene. The two patches in the left of Figure 1-3 appear the same color in the afternoon, but then become different colors at sunset because they correspond to different materials. Further, the color changes rely on the geometries, the lighting, and the complicated physical interactions in a scene. Changing time-of-day seems equivalent to challenging AI-complete problems since it would require understanding the complex rules governing the world of physics. In our second problem, we turn the attention to another more difficult problem, portrait stylization. Although Figure 1-4 captures the input portrait with decent lighting, colors, and camera settings, what makes the output stylish is how a photographer processed this picture. The stylization includes deliberate adjustments on details, colors, and tones. Mimicking the style of a photographer would require understanding her artist's mind, which is far more difficult than modeling physics and reminiscent of the challenges in



(a) *Input*

(b) *Example*

(c) *Our stylized output*

Figure 1-5: *Our local transfer algorithm employs a dense correspondence that links the objects in the same categories (yellow arrows) between the input (a) and the example (b), like the sky and the buildings, to create an output (c) that appears similar photographic look as that of the example.*

artificial intelligence.

This dissertation will sidestep AI-complete problems by leveraging the immense power of data. We will present a data-driven approach as a simple way to understand the world. In computer graphics, data-driven approaches have achieved success in restoring pixels, such as hole filling [49], denoising [73], and deconvolution [47]. This dissertation will exploit data for image stylization. Given an input, we will transfer a desired style from an image database to the input. In the database, we will search for an example image that shares similar semantics with the input, like sky and buildings in Figure 1-5. Then we transfer the photographic look from the example to an output by proposing a *local transfer* algorithm, which is the main contribution of this dissertation. Our algorithm uses semantic information between an input and an example, and transfers image style between the same semantic regions, like sky to sky, buildings to buildings, through a dense correspondence field between the two images. The output in Figure 1-5c appears to have colors similar to that in the example, and looks plausible at the time-of-day of the example. By employing a local correspondence, our method achieves a spatially-variant transfer that is more than just a global mapping. Figure 1-5c darkens the sky, while brightening the window lights in the input, resulting in a plausible appearance when changing from day to night. This dissertation investigates two image categories that are popular and challenging: outdoor photos and headshot portraits, separately in Chapter 3 and 4. Here we provide a high-level overview for each work.

Outdoor photos: time hallucination (Chapter 3)

We introduce “time hallucination”: synthesizing a plausible image at a different time-of-day from an input image. This challenging task often requires dramatically altering the color appearance of the picture. In this work, we introduce the first example-based approach to automatically create a plausible-looking photo that appears as though it were taken at a different time-of-day. The time of day is specified by a semantic time label, such as “night." Our approach relies on a database of time-lapse videos of various scenes. These videos provide rich information about the variations in color appearance of a scene throughout the day. Our method transfers the color appearance from example videos with a similar scene as the input photo. We propose a *locally affine model* learned from the example video for the transfer, allowing our model to synthesize new color data while retaining image details. We show that this model can hallucinate a wide range of different times of day. The model generates a large sparse linear system, which can be solved by off-the-shelf solvers. We validate our methods by synthesizing transforming photos of various outdoor scenes to four times of interest: daytime, the golden hour, the blue hour, and nighttime.

Headshot portraits: style transfer (Chapter 4)

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 allows one to easily reproduce the look of renowned artists. At the core of our approach is a new multi-scale 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.

We will demonstrate our local transfer algorithms on automatic photo retouching. While computational photography has made significant progress in capturing light rays of a scene, such as high-dynamic range photography [27], light-field photography [87], tone mapping [31], and even revealing unseen details hidden in an image [108, 131], photo retouching still relies on massive manual work. In portrait editing, photographers have to work with local editing tools such as brushes to create individual layers on facial landmarks. These tasks are unfortunately tedious and time-consuming even using modern editing softwares like Adobe Photoshop[3]. Meanwhile, the quality of manual retouching depends on photographers' own experiences and aesthetics in tone and detail manipulations. Our method provides an alternative to casual users. Given a target style in mind, a user provides a semantic label, such as the name of the photographer who originally created that style. Our method retrieves a few appropriate examples from the photographer's works, and then generates a gallery of result photos by applying our local transfer algorithm to each example. The only remaining task for users is to select good results from the gallery, which is a great deal easier than creating an output on their own. Benefiting from the increasing number of professional

[3] A typical retouching task takes a casual user about ten minutes to follow the corresponding tutorial [10].

pictures and videos available on photo sharing web sites like Flickr[4] and Vimeo[5], our style transfer becomes more attractive than ever before, since people are better at selecting desired styles from a collection than creating them from scratch. This dissertation largely consists of the works presented in ACM Transactions on Graphics (Proceedings SIGGRAPH ASIA 2013 and SIGGRAPH 2014) [111, 112]. All the the accompanying materials referenced here, including software, videos and demos, are available for the research community through the thesis web page:

http://people.csail.mit.edu/yichangshih/PhDThesis/

[4] http://www.flickr.com/

[5] https://vimeo.com/

Chapter 2

A Survey of Style Transfer

Without enough experience in photographic adjustment, a casual user often finds it challenging to edit the visual appearance of a photograph. To assist people in image stylization, previous studies have proposed style transfer algorithms: given an input image, users supply an exemplar image processed in the desired ways by professional photographers, also called a reference image or a model, and style transfer algorithms render an output with the similarly high-quality visual appearance. In this dissertation, we consider photographic features for style transfer, including colors, tones, contrasts and textureness. By virtue of the idea's simplicity, style transfer algorithms have become an active research field over the last few decades. To characterize the related work in style transfer, we propose a novel two-dimensional taxonomy, as shown in Table 2.1. The first axis of the spectrum describes how global and local the transfer is. Given an input and an example, this property often relies on the matching scale between the two images. The second axis describes the relationship between the input and the example. For instance, a local method often requires the example to have a similar scene as the input. We found that the proposed two-dimensional characterization helps to classify most state-of-the-art style transfer algorithms, and reveal the correlation between these two properties. The recent trend in style transfer algorithms is evolving from global methods to local methods that form the foundation of our approach.

2.1 The focus and the scope of this survey

A recent survey of color mapping technologies [36] classifies the existing methods into three categories: geometry-based methods [35, 45, 127], transferring statistical properties, e.g., the seminal work by Reinhard *et al.* [100], and user-assisted solutions such as using manual scribbles [4, 71, 81]. In this survey, we focus on methods working on automatic stylization as our approach, and leave user-assisted methods aside. We found that both geometry-based methods and statistical transferring methods can be better characterized by how global the transfer is. In particular, statistical transferring methods tend to be global, while geometry-based methods often require a notion of local correspondences. Our characterization offers a finer-grained classification based on the amount of local notions used in style transfer algorithms. For example, the methods based on foreground and background matching [25] are more global than those based on semantic region correspondences [120] and super-pixel matching [69]. We are interested in how similar a user-supplied example and an input need to be with each other. From experience, most histogram-based approaches only require the input and the example to have a similar color palette, while dense correspondence methods are more restricted, since they require precisely the same instances across the input and the example. This aspect serves as a constraint on choosing examples when applying a specific style transfer algorithm. Interestingly, we observed correlations between these two dimensions, which implies a trade-off: global methods tend to accept large differences on image content between the input and the example, while local approaches demand similar semantics or instances between the two images. Our method belongs to the latter, which requires the example to share the same object categories as the input, like sky, buildings, and faces.

Starting from colors, we extend the survey to multi-scale and data-driven processing techniques that inspire our approach for depicting visual styles. We include state-of-the-art multi-scale tonal transfer algorithms [6, 90, 112, 118]. We review methods that take both a single example and multiple examples [132]. Some methods leverage the immense power of a large image database such as the Internet pictures [21, 111]. Style transfer on videos requires special treatments for the temporal coherence, which is outside the scope of this survey. We leave aside texture synthesis [53] and non-photorealistic rendering [54], which are often applied in painterly arts but beyond the goal of our approach. Learning-based approaches that require training on a large dataset [16] are outside the scope of this survey. This chapter starts by reviewing the state-of-the-art style transfer methods in the order from global approaches to local approaches (§2.2). These methods are then characterized according to the differences between input content and example content (§2.3).

2.2 Classifications based on matching scales

We survey sixty-one recently published style transfer works related to our approach. Each work is labeled by how global the transfer algorithm is, and the relationship between a required example and input. Table 2.1 groups together the publications with similar features. The papers are annotated with their publication years to show the research trend over time. These works are first reviewed from global methods to local approaches. Style transfer algorithms often employ parametric models or a correspondence field as in our method to transfer image statistics. Given a style transfer algorithm, we determine its scale according to the scale at which the employed parametric models apply, or the scale the correspondence field is computed. For example, the global method by Reinhard *et al.* [100] applies a single affine model on the entire output (§ 2.2.1), while a local method like our time hallucination [111] builds an affine model for each pixel (§ 2.2.5). Global and local methods form the range of our characterization. Between the two opposites, parametric transfers can be established between segmented regions (§ 2.2.2), super-pixels (§ 2.2.3), and sparse feature points (§ 2.2.4).

2.2.1 Global matching

Global style transfer algorithms process an input by employing a spatially-invariant color mapping function, such as an affine transformation or a color lookup table. These algorithms are computationally efficient, and achieve high-quality results on simple scenes like landscape photos.

Statistical matching Reinhard *et al.* [100] stylize an output by matching its mean and standard deviations to that of an example in a decorrelated color space. To circumvent the color palette differences between the input and the example, they propose to manually associate regions of interest on both images, which motivates our algorithm to employ a dense correspondence field. Their statistical matching leads to a simple but effective color mapping function mathematically modeled by a global affine transform. There are some freedoms in designing this affine transform. Pitié and Kokaram *et al.* [96] propose an optimal transformation by minimizing the earth movement distance, and formulate the problem by the classical transportation optimization. Similarly, our style transfer approach is formulated as an optimization problem, resulting in a large sparse linear system. The result quality of statistical matching methods is sensitive to the choice of color spaces. Reinhard and Pouli [101] study style transfer in various color spaces on a range of input ensembles, including indoor, outdoor, and night pictures, and suggest that the covariances of an input image forms an informative predictor of result quality. Abadpour and Kasaei [1] found that no perfectly decorrelated color space exists, and determine a sub-optimal color space by principle component analysis. Xiao and Ma [135] work directly on a correlated color space, and transfer the color correlation matrix from the example to the output. The works by Wang *et al.* [126] and Bonneel *et al.* [11] show the quality of color grading is sensitive to the color space used in the algorithms. Compared to global transfer algorithms, our local approach is more robust to the choice of color spaces, since our locally affine model is sufficient to capture local statistics.

Histogram transfer Histogram transfer algorithms build a non-linear global color mapping function to match the output histogram to that of the example. For gray-scale images, the mapping functions are computed by simply inverting cumulated probability functions. For color images, previous studies carefully deal with the correlations between color channels to transfer the histogram. Inspired by the Radon transform, Pitié *et al.* [97] iteratively perform a series of one-dimensional histogram matching on randomly projected distributions. Histogram transfer methods often leave unwanted color artifacts on the output, since they ignore local spatial relationships in the input. To address the problem, these methods restore the final results using edge information in the input image either by Poisson editing [98], gradient domain image processing [135], or edge-aware filtering [116]. Likewise, to ensure natural looks on final results, our method enforces local consistency on the output, forming a Laplacian regularization. Further, we preserve the fidelity of output details by processing them separately. To give users control of artistic manipulations, Pouli and Reinhard [99] progressively match histograms from low- to high-order image statistics by a coarse-to-fine scheme. Similarly, our local style transfer employs a multi-scale processing to capture contrasts at different scales.

Mood transfer People tend to interpret color styles by subjective feelings, like “calm,” “vivid,” and “elegant,” among other things. From an image database pre-labeled with feeling words, Yang and Peng [138] learn the image statistical models for these words, also called color moods, and ask users to select a target color mood for style transfer. Feng *et al.* [37] divide color styles into a set of pre-defined categories, and assign the best category to the input image by minimizing the earth movement distances between the input and the category. In the same spirit, our data-driven approach saves users from the troubles of looking for examples, and instead transfers target styles according to the semantic labels given by the users.

Multi-scale transfer In the old days of film photography, burning and dodging tools were common for changing the moods of pictures. In digital photography, photographers achieve similar effects by carefully manipulating image contrasts and details. To transfer photographic looks from an example to an input, Bae *et al.* [6] decompose both pictures into details and low-passed bands, and globally match these bands by way of histogram transfer. Sunkavalli *et al.* [118] improve matching quality by employing a multi-scale decomposition constructed by Laplacian stacks. Recently, Paris *et al.* [90] address the halo artifacts in multi-scale processing with a novel local Laplacian filter, which is later sped up to realtime by Aubry *et al.* [5]. At the core of our portrait style transfer is a similar multi-scale processing technique based on Laplacian stacks to capture facial textures at different ranges, from large scales like eyes and noses to minute details like facial pores.

Discussion Although global style transfer algorithms generate high-quality results on simple scenes like landscape photos, unfortunately, when applying the same techniques to complex scenes like cityscapes, which contain more object classes like buildings and streetlights, they work insufficiently. In complex scenes, scene appearance variation is challenging because it depends on local image content. Limited by one-to-one color mappings, global style transfer algorithms cannot perform content-aware color transfer. In the later chapters we will show that our local methods perform better on challenging stylization problems.

2.2.2 Segment-level matching

Global transfer algorithms fail when the color palette of the example consists of a great deal more colors than that of the input, because they are incapable of expressing one-to-many color mappings. Rather than relying on global models, segment-level style transfer algorithms divide the input and the example into a few regions (typically five to ten regions), and transfer the styles between these regions after the correspondences are established.

Color categories Chang *et al.* [18] study color concepts among major languages, and conclude that eleven color categories are common across most cultures. Their method segments the input and the example according to the RGB color distances to the centers of these color categories. Later, the same authors [19] extend the method to color grading. Murray *et al.* [85] divide the input regions by applying a set of pre-defined color statistical models, called color concepts, and transfer color styles between regions of the same color concepts. These methods avoid the artifacts caused by unconstrained color mappings typical in global transfer methods.

Color segmentation Instead of relying on pre-defined models, Shapira *et al.* [106] segment an input and an example using Gaussian mixture models learned from both images, and transfer color styles by matching the modes between the two models. He *et al.* [51] adapt Expectation-Maximization (EM) algorithm to identify main colors in the input and the example, and transfer these main colors through a gradient-preserved optimization scheme. Similarly, Dong *et al.* [30] match dominant colors by minimizing the earth movement distance (EMD), and transfer pixel colors by a set of affine mappings [100] between matched regions. Wang *et al.* [125] apply pre-trained color enhancement models on segmented regions. They demonstrate that a complex transfer over an entire image can be broken into multiple simpler transfers described by linear models on segmented regions. In our stylization problems, however, we have found that color segmentation is insufficient, since scene appearance variations depend on not only colors but also materials and semantics.

Coarse bin matching To make histogram matching robust to the differences between the color palette of input and example, previous studies have divided the histogram into fewer but coarser bins, and transfer pixel colors between these bins through plain models like affine models. Freedman and Kisilev [39] match these coarse bins by an EMD optimization, and apply a novel affine transform that minimizes color distortions. Liu *et al.* [80] extend the coarse matching to multiple examples to leverage the Internet images. Compared to their coarse matching methods, our method uses a dense matching to capture the example styles more accurately.

Spatial segmentation Recent studies in image statistics have shown that natural images appear spatially coherent in color distribution. Tai *et al.* [120] improve image segmentation with a modified Gaussian mixture model (GMM) that leverages the spatial coherence. Their method demonstrates improved color transfer by grouping together pixels of the same object classes, e.g., sky, trees, and water. Xiang *et al.* [132] extend their method to multiple inputs. Yoo *et al.* [140] extract dominant colors from segmented regions for region matching. Wu *et al.* [129] transfer spatial color distributions from the example, by enforcing scene layout of the output during segment matching. Oliveira *et al.* [89] parse the input and a coarsely registered example into several regions and compute a set of transfer functions from the segments.

Semantic segmentation Since image foregrounds show very different statistical properties from backgrounds, Dale *et al.* [25] match the two regions separately to restore images. Instead of image restoration, we work on image stylization. Using automatic scene parsing techniques, Cusano *et al.* [23] annotate each pixel with a semantic label, such as sky, tree and road. Then they use the semantic information to transfer colors between the regions of the same object classes. Wang *et al.* [124] use texture descriptors for image segmentation, and transfer colors between similar objects like grass-to-grass, sky-to-sky. In this dissertation, we found that semantic information is critical to color transfer. We consider spatial and object information by leveraging a locally-consistent dense correspondence field that respects scene semantics.

2.2.3 Super-pixel Matching

Recently, super-pixels [2], an over-segmentation on an input image, have been proposed to offer local matching, which improves classical correspondence problems, such as depth-from-stereo [142].

To capture image semantics, Chia *et al.* [21] apply super-pixels to color transfer. They over-segment an input and an example to super-pixels, and compute a set of per-segment matching based on the local descriptors of the super-pixels. Similarly, Wu *et al.* [130] apply a coarse-to-fine matching from object levels like sky, buildings, and human faces, to the finer levels defined by super-pixels. Using super-pixel level matching, their method achieves content-adaptive color mappings and dramatic appearance changes. In particular, Laffont *et al.* [69] demonstrate appearance transfer to hallucinate different seasons and weather conditions, such as hallucinating a spring picture to a winter look. Our work shares the same idea with these methods in introducing image local semantics for content-aware style transfer. Our method computes the correspondences on image pixels, which offer a fine-grained matching to deal with complex scenes, without worrying about the boundary artifacts typical in super-pixel processing techniques.

2.2.4 Sparse Matching

Pixel-to-pixel matching provides the finest correspondence between an input and an example. A plethora of works in computer vision have been proposing sparse correspondence techniques between image pairs, and demonstrated a variety of photographic applications like image stitching, registration [119], and 3-D-based photo browsing [113]. For color transfer, Irony *et al.* [58] propagate colors from an example to an input through the sparse correspondences between the two images, using an optimization scheme similar to Levin *et al.* [71]. We formulate our style transfer as an L_2-optimization problem, and propagate the colors with a patch-based Laplacian regularizer, which respects local scene structures of the input image.

Intrinsic image relighting Intrinsic image decomposition separates an image into two layers: the lighting-dependent shading layer, and the reflectance layer invariant to the scene lighting conditions. Given image collections taken at different times of a day, Laffont *et al.* [68] estimate sparse correspondences between these different viewpoints to factor out the scene reflectance layer. Similarly, Liu *et al.* [79] solve for the reflectance from an aligned image collection using median filters. To demonstrate lighting transfer, they replace the input shading with a target shading selected from the collections. Our method circumvents the intrinsic image problems by transferring the variations in scene appearance from an example, which is closely related to the lighting layer of a scene [91, 107]. We demonstrate results of similar quality to those methods using intrinsic image decomposition, as shown later. Further, rather than relying on image collections for a specific input, our method uses data that is independent of the input.

Camera calibration In multi-view imaging applications, camera calibration is used to ensure consistent color appearances across different viewpoints. Camera calibration can be seen as color mappings from a raw input to an output. Sparse correspondence is often established as reference points to build the mapping function [35, 48]. Hwang *et al.* [57] compute a global color mapping for camera calibration, but an accurate calibration requires spatially-variant color mappings as in our stylization problem, which is beyond the limitation of global transfer. Recent works have reported that local methods outperform global transfer on public benchmark in camera calibration [136]. For example, Kagarlitsky *et al.* [63] use a k-d tree to divide the input and the example into non-overlapping regions, and compute piece-wise linear mappings between these regions to achieve one-to-many color mappings as in our method.

2.2.5 Dense Matching

Flat regions in images like sky and water are problematic for sparse correspondences, since they lack distinguishable features to be matched. To regularize textureless regions, dense correspondence fields enforce spatial coherence in an input. It is more flexible than sparse matching since it can match image pairs of different instances as long as the pair share the same semantics, like skylines of two distinctly different streets or landscapes of different mountains [78]. Our method is based on a dense correspondence field between the input and the example for content-aware style transfer.

Full matching Given a patch on the input image, Welsh *et al.* [127] search for the most similar patch from the example, and then transfer the chrominance information to colorize the input. To address the ambiguity at textureless regions, the seminal work by Efros and Freeman [32] enforces spatial coherence on the output by penalizing the differences between the overlapping patches in the output. Charpiat *et al.* [20] assign each output pixel a color from the example pixels by maximizing the joint probability of the output colors. In our time hallucination [111], we employ a Markov random field to compute dense correspondences between an input and a reference video. In our portrait transfer [112], we leverage a coarse-to-fine dense matching scheme starting by detecting facial landmarks, rigid warping, and dense alignment. However, it can be impractical to precisely match two different scenes in some cases. For example, a pore on an input portrait could match multiple pores on an example face of a different person, even if the spatially coherent constraint is already imposed. Hence, in our portrait stylization, we deal with the outliers with a robust transfer to achieve high-quality results.

Partial matching In some cases, it is impractical to expect perfect dense correspondences over the entire input. Partial matching circumvents the problem by rejecting the erroneous correspondences, and solely relying on the regions of high confidences. HaCohen *et al.* [45] employ a partial matching for appearance transfer between an input and an example, which was later extended to album editing [46]. Their method adapts the General Patch Match [7] to account for scene appearance variations induced by lenses, lighting, and non-rigid deformation of the input scene. Our method takes a step further to consider different instances from the same category, e.g., faces of different individuals, or skylines of different cities. Using images from the same category, we are able to leverage the rich resources from the Internet images. In our experience, their method is limited to the same instance across the input and the example, and often returns null correspondences when applied to our problems. Further, their works use global mappings, while we propose local color transfer for stylization. Farbman and Lischinski [34] consider the tonal stabilization of videos. Given a video taken by a moving camera, they compute a partial correspondence on the shared objects between two consecutive frames, and apply color remappings to ensure color consistency along the temporal axis. We follow a similar idea to extend our style transfer algorithms from images to videos.

2.2.6 Other appearance transfer techniques

Some style transfer algorithms are difficult to characterize by either global or local approaches. Bychkovsky *et al.* [16] train predictors for image styles from the image database consisting of retouched photographs. To train the predictors, they leverage various image features including histogram bins, adjustment curves, and tones ranging from highlights to shadows. Xue *et al.* [137] apply machine learning techniques to video grading using clips filmed by celebrated directors. Berthouzoz *et al.* [10] take a completely different approach: transferring the Photoshop edit macros to a new input image. Our method requires no editing information but only a pair of before-and-after images, or just an after image as the example. Our method can mimic image editing operations typically performed with digital brushes, like dodging, burning, and local color adjustments. However, introducing new textures that were not present on the input is challenging to us, like adding mustaches on portraits or reflections on windows.

2.3 Constraints on selecting the example

The example supplied in style transfer algorithms are critical to result quality. The guideline for choosing valid examples varies with the style transfer algorithm. For example, the method by

HaCohen *et al.* [45] requires an example that shares the same instance with the input. Conversely, given an example, we want to know what input images the stye transfer algorithm can take. The vertical axis in Table 2.1 describes the valid examples given an input image. We label each algorithm according to the limitations specified by the original paper. If not specified, we assign the label based on the examples demonstrated in the paper. These labels are ranked by the level of freedom in selecting the example. For instance, "unrelated content," which is on the top of Table 2.1, allows users to freely select examples. In contrast, at the bottom, "same instance" requires users to provide examples taken from the camera viewpoint similar to that of the input, such as stereo pairs. By plotting these style transfer algorithms according to their matching scale and constraints on example selection, Table 2.1 reveals the correlation between the two properties: local methods require examples similar to the input image. The observation applies to our work, which requires the input and the example to share the same semantics.

2.3.1 Unrelated contents

Global methods provide users the maximum freedom in selecting examples. They only entail global properties, such as low-order statistics [96, 100], color palette [37, 138], and histograms [6, 96, 99], and discard the detailed content information like scene layouts in the input. Nonetheless, it can lead to weak outputs if applying global transfer to image pairs of entirely different scenes. We transferred the color from night city views to skylines in the afternoon using a global method [100], but the resulting output looks artificial — it only darkens the input without rendering it with night views. Lately, some papers have pointed out that supplying good examples requires aesthetics of users, and can be challenging to casual users [138].

2.3.2 Same semantics

Studies on the ImageNet [28] have shown that visual appearance of an object is highly correlated with its semantics [29]. Semantic information between an input and an example has been recently exploited in style transfer [69, 129], because it improves the result quality by constraining the color transfer between meaningful regions. With the help of content-aware image retrieval techniques, semantic-aware style transfer algorithms could benefit from image databases [28]. To leverage the semantic information, these methods have to compute dense correspondences between the input and the example, which is still a challenging computer vision problem [65, 76, 78]. In our time hallucination, we compute the semantic correspondences between an input and a reference video by leveraging both the spatial and temporal coherence of the scene.

2.3.3 Same instance under non-rigid deformation

Some style transfer algorithms require exactly the same instance across the input and the example [34, 45, 46]. These methods aim for dealing with the lighting and color inconsistency between cameras. Although these methods allow non-rigid transforms between the input and the example scene, the constraint of the same instance is too restrictive and impractical for our problems. In our work, we lift the constraint to allow broader applications and leverage the Internet images.

2.3.4 Different viewpoints

Some style transfer algorithms register the input and the example by homographic transforms, which require the same instance across the two images. These methods are often used to process landmark photos. Given a famous landmark, users search for the images taken at the same location by other photographers, at different times of a day, viewpoints, or scales. These images are then used for style transfer, by leveraging intrinsic image decomposition [68, 79] or the geometry information extracted from the collection [66]. For our approaches, it works best if the input and the example are taken from similar viewpoints, such as frontal portraits for both images, because the correspondences between the two images are easier.

2.3.5 Stereo calibration

The most restricted scenarios require the two images to be taken from very similar viewpoints. This is the scenario for stereo camera calibration, which compensates the color inconsistency across the two camera views. There are only a limited number of photographic applications that satisfy this constraints.

2.3.6 Discussion

We have reviewed recent works in photographic style transfer, including color transfer and multi-scale contrast transfer. We label these algorithms by their global and local properties, and the constraints on example selections. In Table 2.1, the two features are correlated, showing a trade-off between the matching scale and the freedom in selecting examples. Local transfer algorithms often achieve more dramatic appearance change than that by global methods. In contrast to global methods, which are limited to one-to-one color mappings, local methods employ correspondences to compute powerful one-to-many color mappings by merging a set of local remapping functions. Our local methods often output results with good visual realism [102]. In particular, local approaches in our work [111] and the work by Laffont *et al.* [69] alter the input appearances strikingly to synthesize scene variations across distinct weathers and lighting conditions.

	Full image § 2.2.1	Region-based (segmentation) § 2.2.2	Super-pixel § 2.2.3	Sparse correspondence § 2.2.4	Dense correspondence § 2.2.5
(Unrelated contents)					
Entirely different § 2.3.1	**Statistical matching** Reinhard et al. 2001 Bonneel et al. 2013(Video) Kotera 2005 Wang et al. 2006(Video) Abadpour and Kasaei 2007 Pitié and Kokaram 2007 Reinhard and Pouli 2011 Xiao and Ma 2006 **Histogram Transfer** Pouli and Reinhard 2011 Grundland and Dodgson 2005 Pitié et al. 2005 Pitié et al. 2007 Xiao and Ma 2009 Su et al. 2014 Pichon et al. 2003 Morovic and Sun 2003 **Mood transfer** Yang and Peng 2008 Feng et al. 2013 **Multi-scale transfer** Bae et al. 2006 Sunkavalli et al. 2010 Paris et al. 2011 Aubry et al. 2014	**Color Categories** Chang et al. 2006 Chang et al. 2007 Murray et al. 2011 **Unsupervised segmentation** Shapira et al. 2009 He et al. 2014 Dong et al. 2010 Wang et al. 2011 **Coarse bin match** Freedman and Kisilev 2010 Liu et al. 2014		Huang and Chen 2009	**Image analogy** Hertzmann et al. 2001 Efros and Freeman 2001
Same semantics § 2.3.2	Su et al. 2012	**Spatial segmentation** Tai et al. 2005 Xiang et al. 2009 Wu et al. 2011 Yoo et al. 2013 **Semantic segmentation** Dale et al. 2009 Cusano et al. 2012 Wang et al. 2010	**Transient attribute** Laffont et al. 2014 **Content-aware** Wu et al. 2013 Chia et al. 2011	**Colorization** Irony et al. 2005	**Full matching** Charpiat et al. 2008 Welsh et al. 2002 Shih et al. 2013 Shih et al. 2013 Hwang et al. 2012
Same instances, non-rigid transform § 2.3.3			**Feature matching** Oliveira et al. 2011		**Partial matching** Farbman and Lischinski 2011 HaCohen et al. 2011 HaCohen et al. 2013
Same instances, different viewpoints § 2.3.4				**Intrinsic image** Liu et al. 2008 Laffont et al. 2012 **Camera calibration** Faridul et al. 2013	
Same instances, different lightings § 2.3.5				**Stereo calibration** Hasan et al. 2012 Hwang et al. 2014 Kagarlitsky et al. 2009	
(Similar examples)					

(Global matching) → **(Local matching)**

Exceptions: **Learning-based**: Xue et al. 2013, Bychkovsky et al. 2011, **By demonstration**: Berthouzoz et al. 2011.

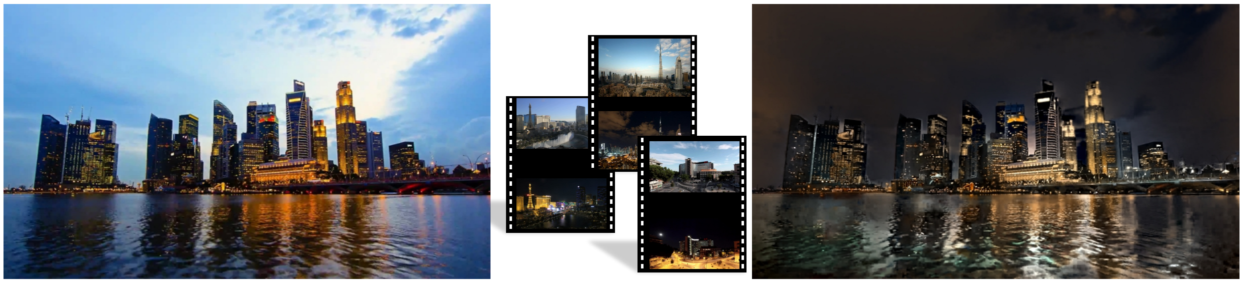
Table 2.1: *Classifications of photographic look transfer techniques.*

Chapter 3

Hallucinating Different Times of a Day

3.1 Introduction

Time of day and lighting conditions are critical for outdoor photography (e.g. [17] chapter "Time of Day"). Photographers spend much effort getting to the right place at the perfect time of day, going as far as dangerously hiking in the dark because they want to reach a summit for sunrise or because they can come back only after sunset. In addition to the famous golden or magical hour corresponding to sunset or sunrise ([104] chapter "The Magical Hour"), the less-known "blue hour" can be even more challenging because it takes place after the sun has set or before it rises ([104] chapter "Between Sunset and Sunrise") and actually only lasts a fraction of an hour when the remaining light scattered by the atmosphere takes a deep blue color and its intensity matches that of artificial lights. Most photographers cannot be at the right place at the perfect time and end up taking photos in the middle of the day when lighting is harsh. A number of heuristics can be used to retouch a photo with photo editing software and make it look like a given time of day, but they can be tedious and usually require manual local touch-up. In this chapter, we introduce an automatic technique that takes a single outdoor photo as input and seeks to hallucinate an image of the same



Input image at blue hour

A database of time-lapse videos

Hallucinate at night

Figure 3-1: *Given a single input image (courtesy of Ken Cheng), our approach hallucinates the same scene at a different time of day, e.g., from blue hour (just after sunset) to night in the above example. Our approach uses a database of time-lapse videos to infer the transformation for hallucinating a new time of day. First, we find a time-lapse video with a scene that resembles the input. Then, we locate a frame at the same time of day as the input and another frame at the desired output time. Finally, we introduce a novel example-based color transfer technique based on local affine transforms. We demonstrate that our method produces a plausible image at a different time of day.*

scene taken at a different time of day.

The modification of a photo suggests the lighting of a different time of day is challenging because of the large variety of appearance changes in outdoor scenes. Different materials and different parts of a scene undergo different color changes as a function of reflectance, nearby geometry, shadows, etc. Previous approaches have leveraged additional physical information such as an external 3D model [66] or reflectance and illumination inferred from a collection of photos of the same scene [68, 70].

In contrast, we want to work from a single input photograph and allow the user to request a different time of day. In order to deal with the large variability of appearance changes, we use two main strategies: we densely match our input image with frames from a time-lapse database, and we introduce an edge-aware locally affine RGB mapping that is driven by the time-lapse data.

First, rather than trying to physically model illumination, we leverage the power of data and use a database of time-lapse videos. Our videos cover a wide range of outdoor scenes so that we can handle many types of input scenes, including cityscape, buildings, and street views. We match the input image globally to time-lapse videos of similar scenes, and find a dense correspondence based on a Markov random field. For these steps, we use state-of-the-art methods in scene matching and dense correspondence, modified to fit our needs. These matches allow us to associate local regions of our input image to similar materials and scenes, and to output a pair of frames corresponding to the estimated time of the input and the desired times of day.

Second, given a densely-aligned pair of time-lapse frames obtained from our first strategy, we still need to address remaining discrepancies with our input, both because the distribution of object colors is never exactly the same and because scene geometry never allows perfect pixel alignment. If we apply traditional analogy methods such as Hertzmann *et al.* [53] and Efros and Freeman [32] designed to achieve a given output texture and simply copy the color from the frame at the desired time of day, the results exhibit severe artifacts. This happens because these methods do not respect the fine geometry and color of the input. Instead, our strategy to address variability is to transfer the *variation of color* rather than the output color itself. Our intuition is simple: if a red building turns dark red over time, transferring this time of day to a blue building should result in a dark blue. We leverage the fact that time lapse videos provide us with registered before-and-after versions of the scene, and we locally fit simple affine mappings from RGB to RGB. Because we use these models locally and because our first step has put our input in dense correspondence with a similar scene, we are able to use a simple parametric model of color change. This can be seen as a form of dimensionality reduction because the RGB-to-RGB *mappings* have less variability than the output RGB *distribution* [42, 117]. In addition, we need to make sure that the affine color changes are coherent spatially and respect strong edges of the image. We thus build on ideas from the matting [72] and intrinsic decomposition fields [12] and derive a Laplacian regularization. We perform the transfer by optimizing an L_2 cost function that simultaneously forces the output to be locally affine to the input, and that this affine model should locally explain the variation between the two frames in the retrieved time lapse. We derive a closed-form solution for the optimization, and show that this yields a sparse linear system. Figure 3-1 previews the result of day-to-night hallucination by our approach.

Contributions Our contributions include the following:

▷ We propose the first time-of-day hallucination method that takes a single image and a time label as input, and outputs a gallery of plausible results.

▷ We introduce an example-based *locally affine model* that transfers the local color appearance variation between two time-lapse frames to a given image.

3.2 Related Work

Image Relighting and Color Transfer In computer graphics, physically rendering a picture at a certain time of day requires meticulous modeling and manual works [60], and becomes impractical to photographic applications. In contrast, current study suggests that a human vision system is far from accurate, and offers a opportunity for image-based rendering. Deep Photo [66] successfully relights an image when the geometric structure of the scene is known. Laffont *et al.* [68] demonstrates that the intrinsic image derived from an image collection of the same scene enables the relighting of an image. In both cases, the key to producing high-quality results is the availability of scene-specific data. While this additional information may be available for famous landmarks, this data does not exist in many cases. Our method targets a more general case that does not need scene-specific data. It only relies on the availability of time-lapse videos of similar-looking scenes.

Approaches for color transfer [97, 99, 100] apply a global color mapping to match color statistics between images. They work well in style transfer, but cannot be applied to time hallucination problem because the problem requires dramatic color appearance change. In comparison, our transfer is local and can distinguish the difference in color change between different image regions in the input even if they have a similar color. Our experiments show that our approach yields better results than global transfer.

Similarly to Lalonde *et al.* [70], we use time-lapse data to study color appearance variation at different times of a day. Lalonde *et al.* 's work creates successful relit images by modeling the scene geometry manually. In contrast to their technique, our method hallucinates images by automatically transferring the color information from a time-lapse.

Example-based Image Colorization Example-based colorization [58] automatically generates scribbles from the example image onto the input gray image, and then propagates colors in a way that is similar to colorization using optimization [71]. In our problem, the scene color appearance is usually different from the input, so the color palette in the time-lapse video is not sufficient. For this, instead of direct copying the color palette from the example, we employ a locally affine model to synthesize the unseen pixels from the time-lapse video.

Image Analogies Our work relates to Image Analogies [32, 53] in the sense that

input : hallucinated image :: matched frame : target frame

where the matched and target frames are from the time-lapse video. However, we cannot simply copy the patches from target frame onto input image, because the texture and color in input are different from time-lapse video. To accommodate the texture differences, we introduce the local affine models to transfer the color appearance from the time-lapse video to the input.

Image Collections Recent research demonstrates convincing graphics application with big data, such as scene completion [49], tone adjustment [16], and super-resolution [41]. Inspired by the

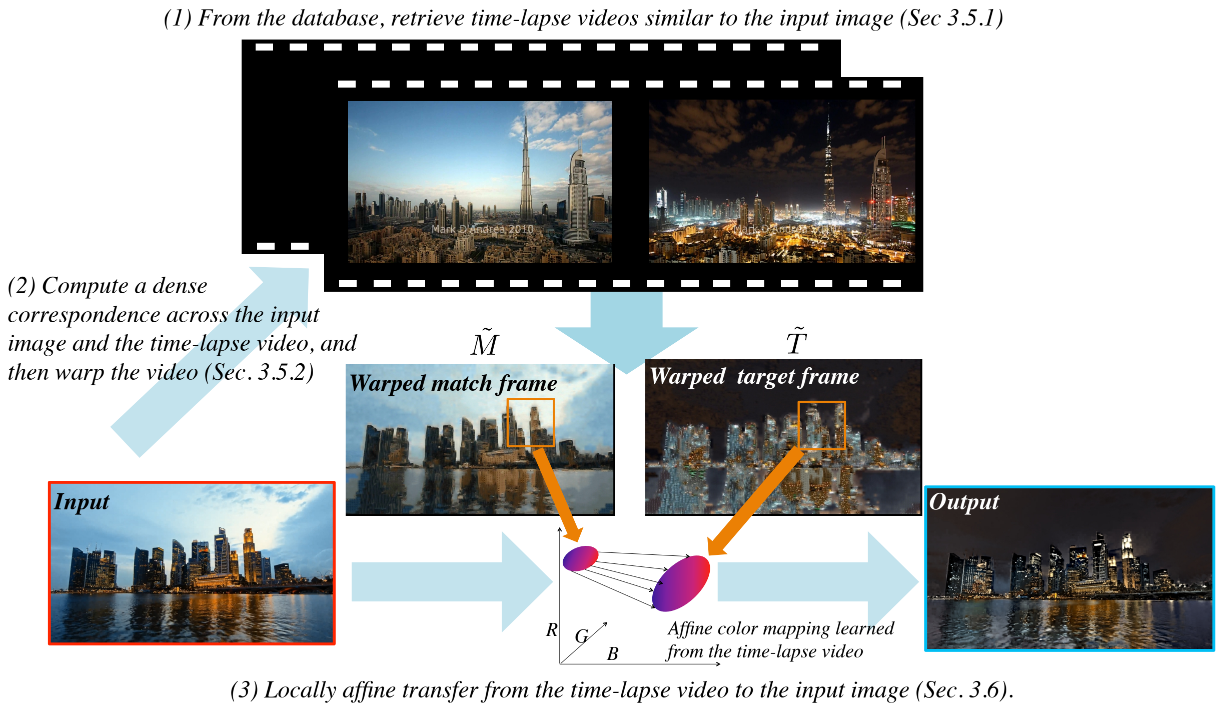

(1) From the database, retrieve time-lapse videos similar to the input image (Sec 3.5.1)
Mark D'Andrea 2010
(2) Compute a dense correspondence across the input image and the time-lapse video, and then warp the video (Sec. 3.5.2)
$\tilde{M}$
$\tilde{T}$
Warped match frame
Warped target frame
Input
Output
R
G
B
Affine color mapping learned from the time-lapse video
(3) Locally affine transfer from the time-lapse video to the input image (Sec. 3.6).

Figure 3-2: *Our approach has three steps. (1) We first retrieve videos of similar scene with the input image (§ 3.5.1), and then (2) find the local correspondence between the input and the time-lapse video (courtesy of Mark D'Andrea) (§ 3.5.2). (c) Finally we transfer the color appearance from the time-lapse video to the input (§ 3.6).*

previous success, our method uses a database of 495 time-lapse videos for time hallucination.

3.3 Overview of our method

The input to our algorithm is a single image of a landscape or a cityscape and a desired time of day. From these, we hallucinate a plausible image of the same scene as viewed at the specified time of day. Our approach exploits a database of time-lapse videos of landscapes and cityscapes seen as time passes (§ 3.4). This database is given a priori and independent of the user input, in particular, it does not need to contain a video of the same location as the input image.

Our method has three main steps (Figure 3-2). First, we search the database for time-lapse videos of scenes that look like the input scene. For each retrieved video, we find a frame that matches the time of day of the input image and another frame at the target time of day (§ 3.5.1). We achieve these two tasks using existing scene and image matching techniques [133].

Next, to locally transfer the appearance from the time-lapse videos, we need to locally match the input and each video. We employ a Markov random field to compute a dense correspondence for each time-lapse video (§ 3.5.2). We then warp the videos to match the input at the pixel level.

Finally, we generate a gallery of hallucinated results, one for each retrieved time-lapse video. To transfer the appearance variations of a time-lapse video onto the input image, we introduce an example-based transfer technique that models the color changes using local affine transforms (§ 3.6). This model learns the mapping between the output and input from the time-lapse video, and preserves the details of the input.

3.4 Database and Annotation

Our database contains 450 time-lapse videos, covering a wide range of landscapes and cityscapes, including city skyline, lake, and mountain view. Figure 3-3 shows a mosaic of all the scenes in the database Unlike most web-cam clips [70] or surveillance camera videos [59], our time-lapse videos are taken with high-end setups, typically a DSLR camera on a sturdy tripod, that are less prone to over-and under-exposure, defocus, and accidental shake. Our database is available at the project website: `people.csail.mit.edu/yichangshih/time_lapse/`

The most interesting lighting for photographers are daytime, golden hour, blue hour (occurs between golden hour and night), and nighttime [17]. For each time-lapse, we label the transition time between the above four different lightings, so that the user can specify the hallucination time by these semantic time labels.

Figure 3-3: *A snapshot of our time-lapse video database.*

3.5 Matching Between the Input Image and Time-lapse Data

The first step of our algorithm is to determine the correspondence between the input image and the time-lapse data. We first find a set of time-lapse videos with a similar scene as the input image, and then compute a dense correspondence between the input image for each matched time-lapse video.

3.5.1 Global Matching

The first step of our algorithm is to identify the videos showing a scene similar to the given input image. We employ a standard scene matching technique in computer vision, adapting the code from Xiao *et al.* [133] to time-lapse data. We sample 5 regularly spaced frames from each video, and then compare the input to all these sampled frames. To assign a score to each time-lapse video, we use the highest similarity score in feature space of its sampled frames. We tried the different descriptors suggested in Xiao *et al.* 's paper, and found that the Histograms of Oriented Gradients (HOG) [24] works well for our data. We show some sample retrieval results in Appendix A.

Now that we have a set of matching videos, for each of them, we seek to retrieve a frame that matches the time of day of the input image. We call this frame the *matched frame*. Since we already selected videos with a similar content as the input image, this is a significantly easier task than the general image matching problem. We use the color histogram and L_2 norm to pick the matched frame. We show sample results in Appendix A. Our approach finding matching videos and frames produced good results for our database but we believe that other options may also work well.

3.5.2 Local Matching

We seek to pair each pixel in the input image I with a pixel in the match frame M. As shown later in Figure 3-11, existing methods such as PatchMatch [7] and SIFT Flow [76] do not produce satisfying results because they are designed to match with a single image and are not designed for videos. We propose a method exploiting the additional information in a time-lapse video by constraining the correspondence field along time. For this, we formulate the problem as a Markov random field (MRF) using a data term and pairwise term.

Similarly to PatchMatch and SIFT Flow, for each patch in I, we seek a patch in M that looks similar to it. This is modeled by the data term of the MRF. We use the L_2 norm over square patches of side length $2r+1$. Formally, for pixels $p \in I$ and the corresponding pixel $q \in M$, our data term is:

$$E_1 = \sum_{i=-r}^{+r} \sum_{j=-r}^{+r} \left\| I(x_p+i, y_p+j) - M(x_q+i, y_q+j) \right\|^2 \tag{3.1}$$

We then leverage the information provided in a time-lapse video. Intuitively, we want the adjacent patches to look similar at any time of the video. This is captured by the pairwise term of the MRF. Formally, we introduce the following notations. For two adjacent pixels p_i and p_j in I, we name Ω the set of the overlapping pixels between the two patches centered at p_i and p_j. For each pixel $o \in \Omega$, we define the offsets $\delta_i = o - p_i$ and $\delta_j = o - p_j$. For the energy we use L_2 norm within each frame t, but L_∞ norm across frames so that the assigned compatibility score corresponds to the worst case over the video V. This gives the pairwise term as:

$$E_2(q_i, q_j) = \max_t \sum_{o \in \Omega} \left\| V_t(q_i + \delta_i) - V_t(q_j + \delta_j) \right\|^2 \tag{3.2}$$

Denoting λ parameter controlling the importance of the compatibility term compared to the data term, N_i the neighboring pixels of i, one could find q by trying to minimize the energy:

$$\sum_{i \in I} E_1(p_i, q_i) \quad + \quad \lambda \sum_{i \in I, j \in N_i} E_2(q_i, q_j) \tag{3.3}$$

by considering all possible pairings between a pixel in I with a pixel in V. However, this would be impractical because of the sheer number of possible assignments. We now explain below how to select a small number of candidate patches so that the optimization of Equation 3.3 becomes tractable.

Candidate Patches A naive way to select a few candidate patches for each location would be to pick the top n patches according to the data term E_1. However, this tends to return patches that are clustered around a small number of locations. This lack of diversity later degrades the transfer. Instead of picking the top candidates, we randomly sample the candidates according to the probability:

$$\frac{1}{Z}\exp\left(-\frac{E_1}{2\sigma^2}\right) \tag{3.4}$$

where Z is a normalization factor and σ controls how diverse the sampled patches are. This strategy yields a candidate set with more variety, which improves the transfer quality. In practice, we sample 30 patches, and use $\lambda = 0.5$ and $\sigma = 20$. We minimize Equation 3.3 using Belief Propagation [139].

Discussion Our sampling strategy is akin to the seminal work proposed by Freeman *et al.* [40], except that we do not explicitly enforce diversity as they do. Testing their approach in our context would be interesting, but since we obtained satisfying results with the approach described above, we leave this to future work.

3.6 Locally Affine Color Transfer

The core of our method is the example-based locally affine color transfer. The transfer starts from the input image I, the warped match frame $\tilde{M}$, the warped target frame $\tilde{T}$, and output the hallucinated image O (See Figure 3-2).

We design the transfer to meet two goals:

- We want it to explain the color variations observed in the time-lapse video. We seek a series of affine models $\{\mathbf{A}_k\}$ that locally describe the color variations between $\tilde{T}$ and $\tilde{M}$.
- We want a result that has the same structure as the input and that exhibits the same color

change as seen in the time-lapse video. We seek an output O that is locally affine to I, and explained by the same affine models $\{\mathbf{A}_k\}$.

A naive solution would be to compute each affine model $\mathbf{A}_k$ as a regression between the k^{th} patch of $\tilde{M}$ and its counterpart in $\tilde{T}$, and then independently apply $\mathbf{A}_k$ to the k^{th} patch of I for each k. However, the boundary between any two patches of O would not be locally affine with respect to I, and would make O have a different structure from I, e.g., allows for spurious discontinuities to appear at patch boundaries. Instead of this naive approach, we formulate this problem as a least-squares optimization that seeks local affinity *everywhere* between O and I. We also specifically account for the possibility of the data of being corrupted by noise and compression artifacts.

3.6.1 L_2-optimal locally affine model

We use a matrix formulation to describe our approach. We use $\mathbf{v}_k(\cdot)$ to denote the k^{th} patch of an image given in argument. For a patch containing N pixels, $\mathbf{v}_k(\cdot)$ is a $3 \times N$ matrix, each column representing the color of a pixel as $(r, g, b)^{\mathsf{T}}$. We use $\bar{\mathbf{v}}_k(\cdot)$ to denote the patch augmented by ones, i.e., $4 \times N$ matrix where each column is $(r, g, b, 1)^{\mathsf{T}}$. The local affine functions are represented by 3×4 matrices, $\mathbf{A}_k$. With this notation, the first term in our energy models the need for the $\mathbf{A}_k$ matrices to transform $\tilde{M}$ into $\tilde{T}$. With a least-squares formulation using the Frobenius norm $\|\cdot\|_{\mathsf{F}}$, i.e., the square root of the sum of the squared coefficients of a matrix, this gives:

$$\sum_k \left\| \mathbf{v}_k(\tilde{T}) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(\tilde{M}) \right\|_{\mathsf{F}}^2 \tag{3.5}$$

We also want the output patches to be well explained by the input patches transformed by the $\mathbf{A}_k$ matrices:

$$\sum_k \left\| \mathbf{v}_k(O) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(I) \right\|_{\mathsf{F}}^2 \tag{3.6}$$

Finally, we add a regularization term on the $\mathbf{A}_k$ matrices for the case when Equation 3.5 is underconstrained e.g., $\mathbf{v}_k(\tilde{M})$ is constant. For this we regularize $\mathbf{A}_k$ using a global affine model $\mathbf{G}$, the regression by the entire picture of $\tilde{M}$ and $\tilde{T}$, with the Frobenius norm. Formally, we solve

$$O = \arg\min_{O,\{\mathbf{A}_k\}} \sum_k \left\|\mathbf{v}_k(O) - \mathbf{A}_k\,\bar{\mathbf{v}}_k(I)\right\|^2 + \epsilon \sum_k \left\|\mathbf{v}_k(\tilde{T}) - \mathbf{A}_k\,\bar{\mathbf{v}}_k(\tilde{M})\right\|^2 + \gamma \sum_k \left\|\mathbf{A}_k - \mathbf{G}\right\|_{\mathsf{F}}^2 \quad (3.7)$$

where ϵ and γ control the relative importance of each term.

Discussion Equation 3.5 alone would correspond to standard local linear regression. With such formulation, overlapping affine transforms would be independent from each other and they could potentially predict widely different values for the same pixel. With Equation 3.6, overlapping transforms are explicitly constrained to produce consistent values, which forces them to produce a result coherent over the whole image.

Closed-form Solution In this section, we derive a closed-form solution for Equation 3.7. We follow a strategy similar to Levin *et al.* [72] and Bousseau *et al.* [12] and remove the $\mathbf{A}_k$ functions from the equations by expressing them as a function of the other variables. That is, assuming that O is known, Equation 3.7 becomes a standard linear least-squares optimization problem with the $\mathbf{A}_k$ matrices as unknowns. Denoting $\mathbf{Id}_n$ an $n \times n$ identity matrix, this leads to:

$$\mathbf{A}_k = \left(\mathbf{v}_k(O)\bar{\mathbf{v}}_k(I)^{\mathsf{T}} + \epsilon\mathbf{v}_k(\tilde{T})\bar{\mathbf{v}}_k(\tilde{M})^{\mathsf{T}} + \gamma\mathbf{G}\right) \left(\bar{\mathbf{v}}_k(I)\bar{\mathbf{v}}_k(I)^{\mathsf{T}} + \epsilon\bar{\mathbf{v}}_k(\tilde{M})\bar{\mathbf{v}}_k(\tilde{M})^{\mathsf{T}} + \gamma\mathbf{Id}_4\right)^{-1} \quad (3.8)$$

Then, defining $\mathbf{B}_k = \left(\bar{\mathbf{v}}_k(I)\bar{\mathbf{v}}_k(I)^\mathsf{T} + \epsilon\bar{\mathbf{v}}_k(\tilde{M})\bar{\mathbf{v}}_k(\tilde{M})^\mathsf{T} + \gamma\mathbf{Id}_4\right)^{-1}$, a minimizer of Equation 3.7 is:

$$\begin{aligned} O &= \mathbf{M}^{-1}\,\mathbf{u} \\ \text{with: } \mathbf{M} &= \textstyle\sum_k \operatorname{lift}_k\left(\mathbf{Id}_N - \bar{\mathbf{v}}_k(I)^\mathsf{T}\mathbf{B}_k\bar{\mathbf{v}}_k(I)\right) \\ \mathbf{u} &= \textstyle\sum_k \operatorname{lift}_k\left(\left(\epsilon\mathbf{v}_k(\tilde{T})\bar{\mathbf{v}}_k(\tilde{M})^\mathsf{T} + \gamma\mathbf{G}\right)\mathbf{B}_k\bar{\mathbf{v}}_k(I)\right) \end{aligned}$$

where $\operatorname{lift}_k(\cdot)$ is an operator that lifts matrices and vectors expressed in the local indexing system of the k^{th} patch into larger matrices and vectors indexed in the global system of the image.

Model Expressivity We demonstrate the expressivity of our model by taking a frame from a time-lapse video as input, and hallucinating to another time using the same time-lapse video. In Figure 3-4 we show this model can express dramatic color appearance, such as day-to-night and night-to-day. We test on various scenes in the Appendix A. For all results in this work, we use $\epsilon = 0.01$, $\gamma = 1$ (pixel value $\in [0, 255]$), $N = 25$ (5×5 patch). We compare the choice of affine model versus linear model in Appendix A. The residuals show locally affine model is better than linear model.

Link with Illumination Transfer If the patches in I and the warped time-lapse are Lambertian, then our method becomes illumination transfer. In this case, the local affine model degenerates to diagonal matrix with the last row equal to zeros. The non-zero components are the quotient of the illuminations between the target and the match frame. For non-Lambertian patches, such as sky and water, our method produces visually pleasing results by using non-diagonal components in the model.

Link with the Matting Laplacian M in Equation 3.9 is similar to the Matting Laplacian [72], except that the local scaling factor $\mathbf{B}_k$ is $\left(\mathbf{v}_k(I)^\mathsf{T}\mathbf{v}_k(I) + \epsilon\mathbf{v}_k(\tilde{M})^\mathsf{T}\mathbf{v}_k(\tilde{M}) + \gamma\mathbf{Id}_k\right)^{-1}$ whereas for the Matting Laplacian, it is $\left(\mathbf{v}_k(I)^\mathsf{T}\mathbf{v}_k(I) + \gamma\mathbf{Id}_k\right)^{-1}$. That is, in addition to the covariance of the input data, our method also accounts for the covariance of the example data.

3.6.2 Dealing with Noisy Input

The affine mapping has a side effect that it may magnify the noise existing at the input image, such as sensor noise or quantization noise. This problem usually appears when the affine model is under-constrained, which may lead into large coefficients in the affine model. We propose a simple yet effective solution to avoid the noise magnification. We first use bilateral filtering to decompose the input image into a detail layer and a base layers, the latter being mostly noise-free. We then apply our locally affine transfer to the base layer instead of the input image. Finally, we obtain the final result by adding the detail layer back to the transferred base layer. Since the base layer is clean, the noise is not magnified. Compared to directly taking the input image, we significantly reduce the noise, as shown in Figure 3-5.

3.7 Results and Comparison

Figure 3-6 illustrates the result of our transferring approach, which transfers the color changes between the target and matched frame to the input. The result produced by our method is more visually pleasing than using only the target frame.

Figure 3-7 shows our method applied to two day-time images. For each of the two images, we hallucinate four times of day: “day”, “golden hour” (i.e., just before sunset), “blue hour” (i.e., just after sunset), and “night”. We use the top two time-lapse videos retrieved in our database, each produces a different plausible hallucination, thereby enabling the exploration of various possible renditions of the desired time of day. These results at four times of day illustrate the ability of our approach to cope with dramatic appearances. We observed that the appearance of city-scape time-lapse usually has larger variability than natural landscape, and so the renditions produced by cityscape input usually have more variations. Figure 3-8 shows the hallucination works from various scenes. Figure 3-9 show that our approach also handles input images taken at different times of day.

Figure 3-10 compares our hallucinated image to an actual photo of the same scene, and shows that, while our result is different, it is nevertheless plausible. In our project website[1], we provide the results of our technique applied to all the landscapes and cityscapes within the first 101 images of the MIT-Adobe fiveK dataset [16].

Figure 3-11 shows that in our context, our MRF-based method to compute the dense correspondence field performs better than PatchMatch [7] and SIFT Flow [76]. This is because we exploit the information across the time-lapse frames, as opposite to only using the target frame. Figure 3-12 demonstrates that our local affine transform model preserves image details better than an edge-aware filter like the Joint Bilateral Filter [33, 92] or the Guided Filter [50].

Performance We measure the average performance using 16 inputs in MIT-Adobe fiveK dataset [16]. We scale all input images to a 700-pixels width. For each input, the matching takes 25 seconds total, split into 23 seconds for local matching and 2 seconds for global matching. For each hallucinated result, the transfer takes 32 seconds. We use conjugate gradient descent in Matlab and incomplete Cholesky decomposition as a preconditioner to solve the linear system.

3.7.1 Comparison to Previous Work

Figure 3-13 compares our approach to techniques based on a global color transfer [97, 100]. While these methods succeed to some degree, their results are not always as accurate as ours. In

[1] `http://people.csail.mit.edu/yichangshih/time_lapse/webpage/`

comparison, our results are cleaner. The local nature of our approach allows it to make better matches, e.g., sky to sky and building to building.

We also tried to compare with the technique of HaCohen *et al.* [45] that first finds dense correspondences and then performs a parametric color transfer. We found their method is not applicable in our case, because our target frame is a different scene from the input image. For all the examples in Results section, their implementation reported that no match was found.

Another thread in recent research that demonstrates successful image illumination transfer uses rich information of the scene, such as Deep Photo, which leverages depth map and texture of the scene [66], or Laffont *et al.* [68], which uses intrinsic image and illumination from a collection of images of the same scene. In Appendix A, we show that our results are on par with these methods even though our approach uses a generic database of time-lapse videos instead of scene-specific data.

Discussion While the methods of Pitié *et al.* [97] and Reinhard *et al.* [100] directly transfer the colors of the target image, our approach transfers the color transformation from the matched frame to the target frame. This may produce less intuitive outputs than a direct color transfer. However, in practice, users do not see the target frame and as a consequence, have no expectation to match its look. And, more importantly, transferring the color transformation allows us to be less sensitive to the image content. For instance, Figure 3-6 shows that a direct color transfer produces a weak golden hour look because it ignores that the input photo has a content that contains warm colors. In comparison, our approach transfers the color transformation and warms up the image a lot more, which corresponds to the change observed in the time-lapse video, and produces a more convincing golden hour rendition.

User Study A successful hallucinated image should look natural to a human observer. Inspired by image inpainting [49], we performed a user study to quantitatively evaluate whether human observers believe our results are real images.

We performed the study with 9 images randomly selected from 9 different time-lapse videos. For each image, we randomly selected 6 or 7 target frames from the top 10 retrieved videos. Then we generated hallucinated images with our approach and Reinhard *et al.* 's method [100]. As baseline, we randomly selected 6 or 7 frames from the input image's time-lapse video. In total, we used 59 results of 9 different scenes for each method. We then mixed the output from our method, Reinhard *et al.* 's technique with real time-lapse frames, and randomized the order. For each image, we ask 5 testers if the image is real or fake.

We performed this task on Amazon Mechanical Turk. 55.2% of our results were classified real. In comparison, the percentage was 66.4% for the real time-lapse frames and 48.8% for Reinhard *et al.* 's method [100]. As expected our approach does not perform as well as actual video frames, but, nonetheless users prefer our method to Reinhard *et al.* 's method.

3.7.2 Applications

In addition to time hallucination, our method can be used for different graphics applications.

Lighting and Weather Transfer In Figure 3-14, the matched and target frames are selected close in time but the target is more sunny. Our algorithm successfully transfers the sunshine to the input image to create a sunny output.

Similarly, we can transfer weather conditions by choosing a target with a different weather from the input. In Figure 3-15, we create a cloudy image from a sunny input by transferring the color properties of a cloudy target image.

Hallucinating Paintings Figure 3-16 shows that our approach also applies to paintings, even though our method is designed for realistic photos.

Synthetic Time-lapse Video By interpolating between the hallucinations at four different times, we generate continuous lighting changes. We show several example videos on our project website[2]. We envision that this could also be used to enable users to choose an arbitrary time of day, e.g., with a slider that selects a frame of the synthetic time-lapse video.

3.8 Discussion and Conclusion

The main novelty of this work is the idea of leveraging time-lapse database for light transfer. Compared to data-driven image completion which leverages millions images [49], it is surprising that with only 450 videos we can achieve convincing results. This is due to our contributions in a example-based locally affine model.

Limitation Our method still has some limitations. If an object is not static or nearly static in the scene, there may be problems finding correspondences. For example, time-lapse videos do not have humans in the scene, so we do not have a proper model for human skin. Moving clouds in the sky can also cause flickering when synthesizing a new time-lapse video with our method using frame-by-frame transfer. Picking a few keyframes and interpolating between them would perform better as shown in the videos on the project website, but the motion of the clouds would still not be captured.

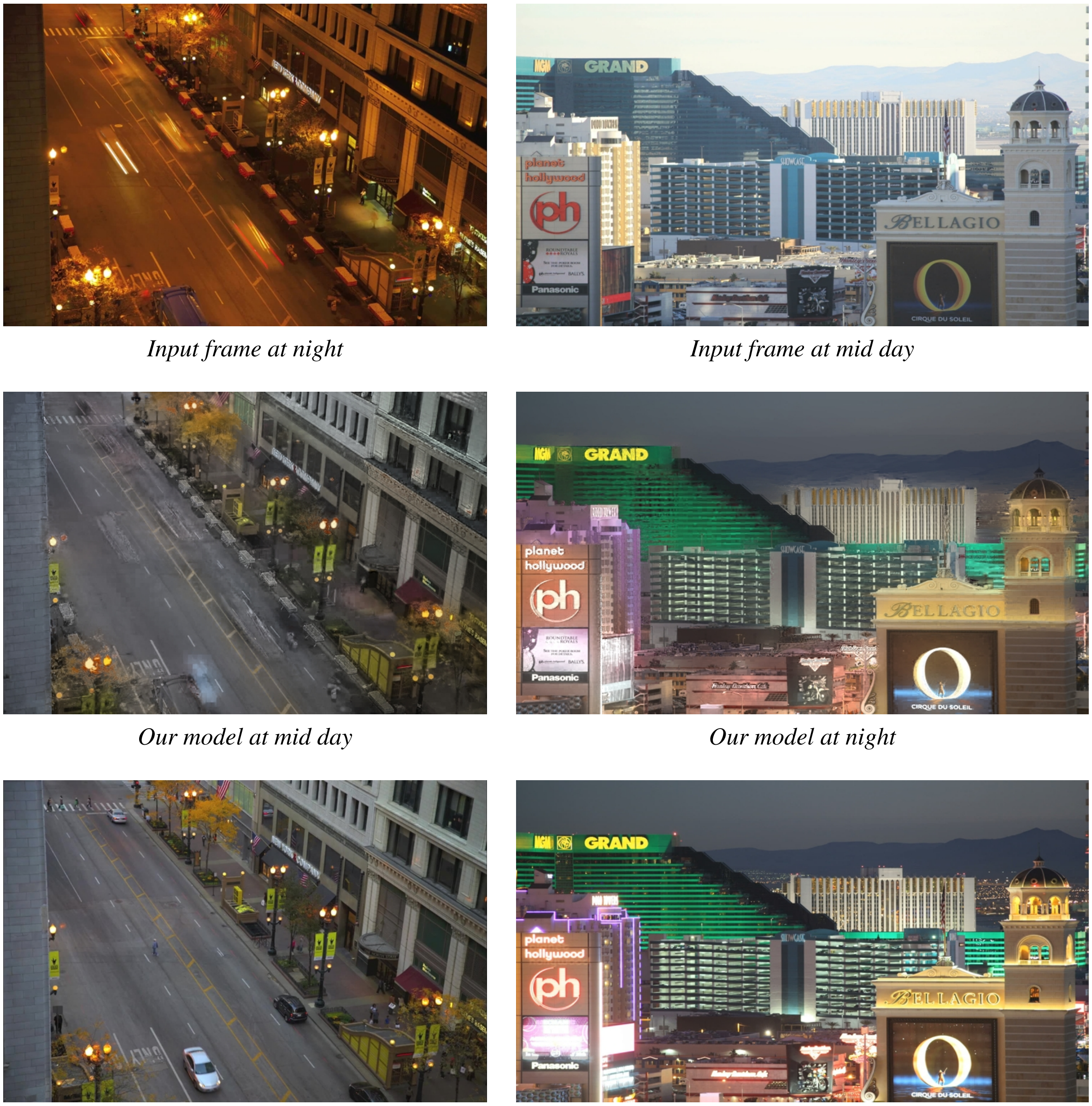
Our method can hallucinate results that, while visually plausible, may not be physically accurate, for example, shadows and highlights that are not consistent. Even if an hallucination is

[2] `http://people.csail.mit.edu/yichangshih/time_lapse/`

technically successful, the result may not always be visually pleasing. For instance, landscapes at night may be overly dark due to the lack of lights. The ability to choose among several results rendered from different time-lapse videos helps mitigate these issues.

Our method can be applied to many graphic applications. For example, in scene completion and image-based rendering, our approach could hallucinate images from different times of a day into a similar time as a pre-processing step.

Beyond the graphics application, perhaps a deeper question is this: *can we learn the image feature evolution along time by observing enough time-lapse data?* We are excited at more research using time-lapse data.



Input frame at night

Input frame at mid day

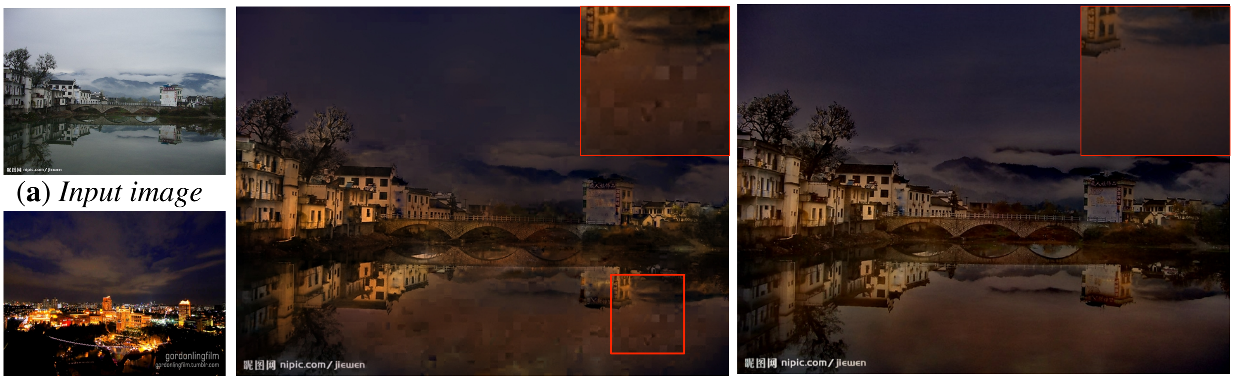
Our model at mid day

Our model at night

The ground truth

The ground truth

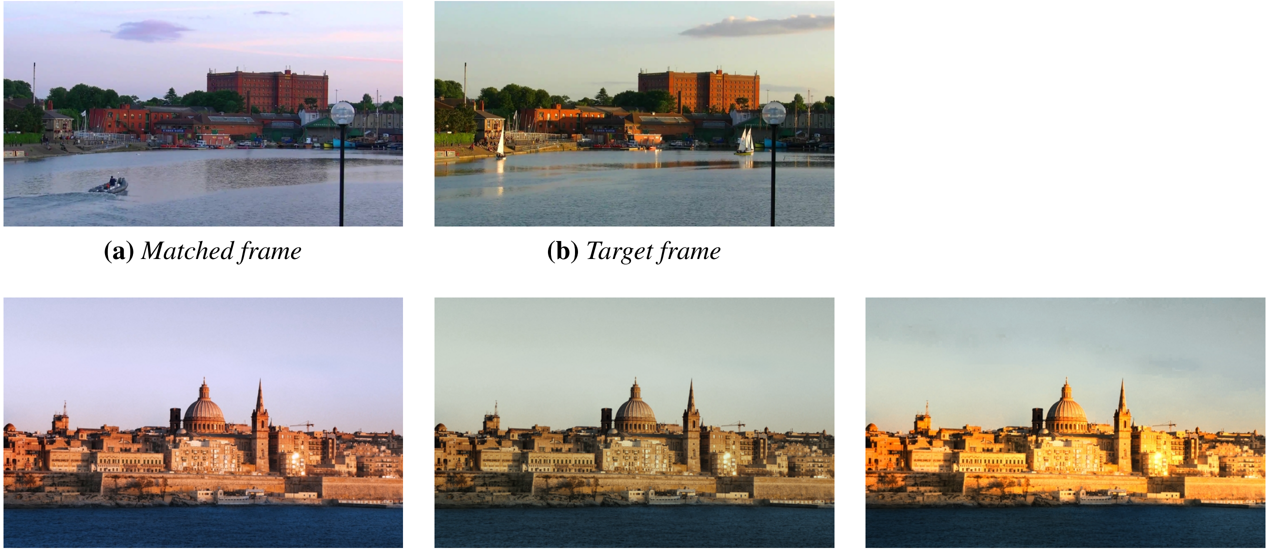
Figure 3-4: *Our locally affine model is expressive enough to approximate dramatic color change, such as night-to-day and day-to-night (left and right column). As a sanity check, we pick two frames from the same video at different times as input and target. We also use the input as matched frame and apply our model. In this situation, if local affine transforms correctly model the color changes, the output should closely match the target, which the ground truth in this scenario. Our result shows that this is the case.*



(**a**) *Input image*

(**b**) *Target frame* (**c**) *Locally affine model* (**d**) *Our noise reduction transfer*

Figure 3-5: *The noise in JPEG input (a) results in artifact at the output of locally affine model (c). Our noise-robust affine model significantly reduces the noise (d). Image courtesy of Jie Wen (a) and Reanimated Studio* `https://vimeo.com/34362088` *(b).*



(**a**) *Matched frame* (**b**) *Target frame*

(**c**) *Input* (**d**) *Photoshop Match Color* (**e**) *Our result*

Figure 3-6: *Producing a golden hour rendition of a scene that contains warm colors (c) using a direct color transfer from image (b) generates a weak effect (d). We created this result with the Photoshop Match Color function. In comparison, our approach transfers the color transformation between the matched frame (a) and the target frame (b) and captures the strong color change characteristic of the golden hour (e).*

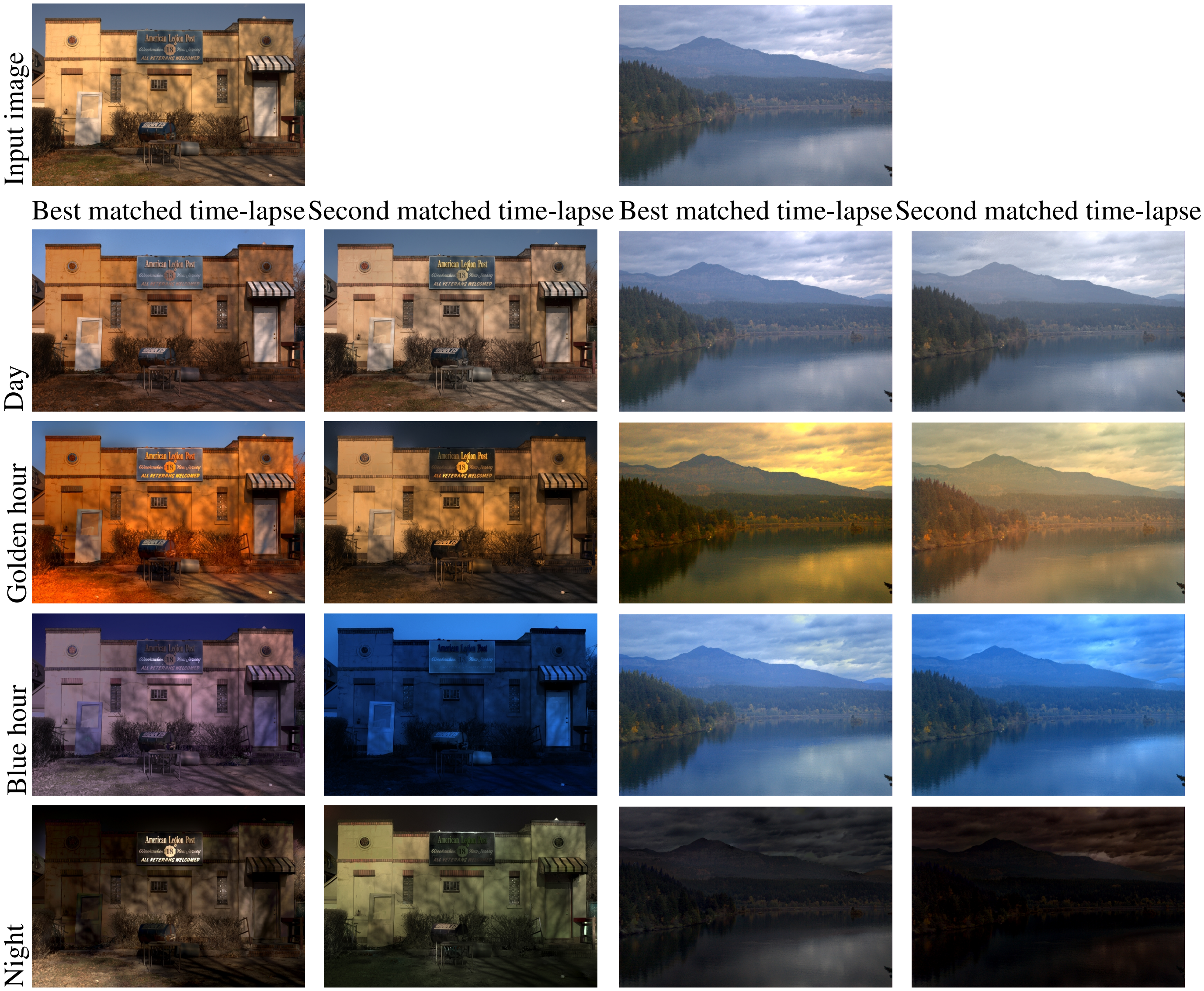
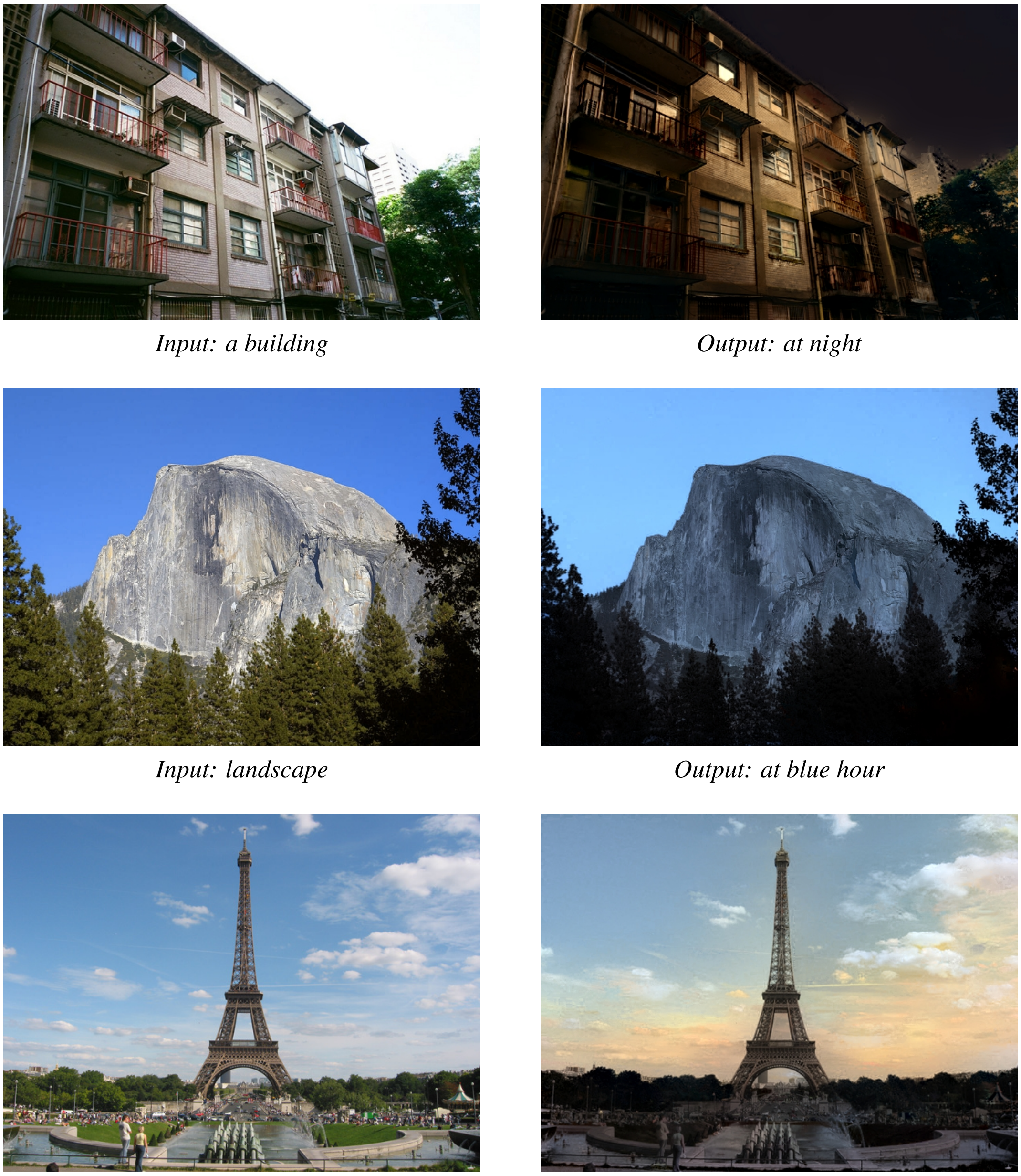


Figure 3-7: *We hallucinate the house and lake at four different times of day. Each time, we show the results for two retrieved videos.*



Input: a building

Output: at night

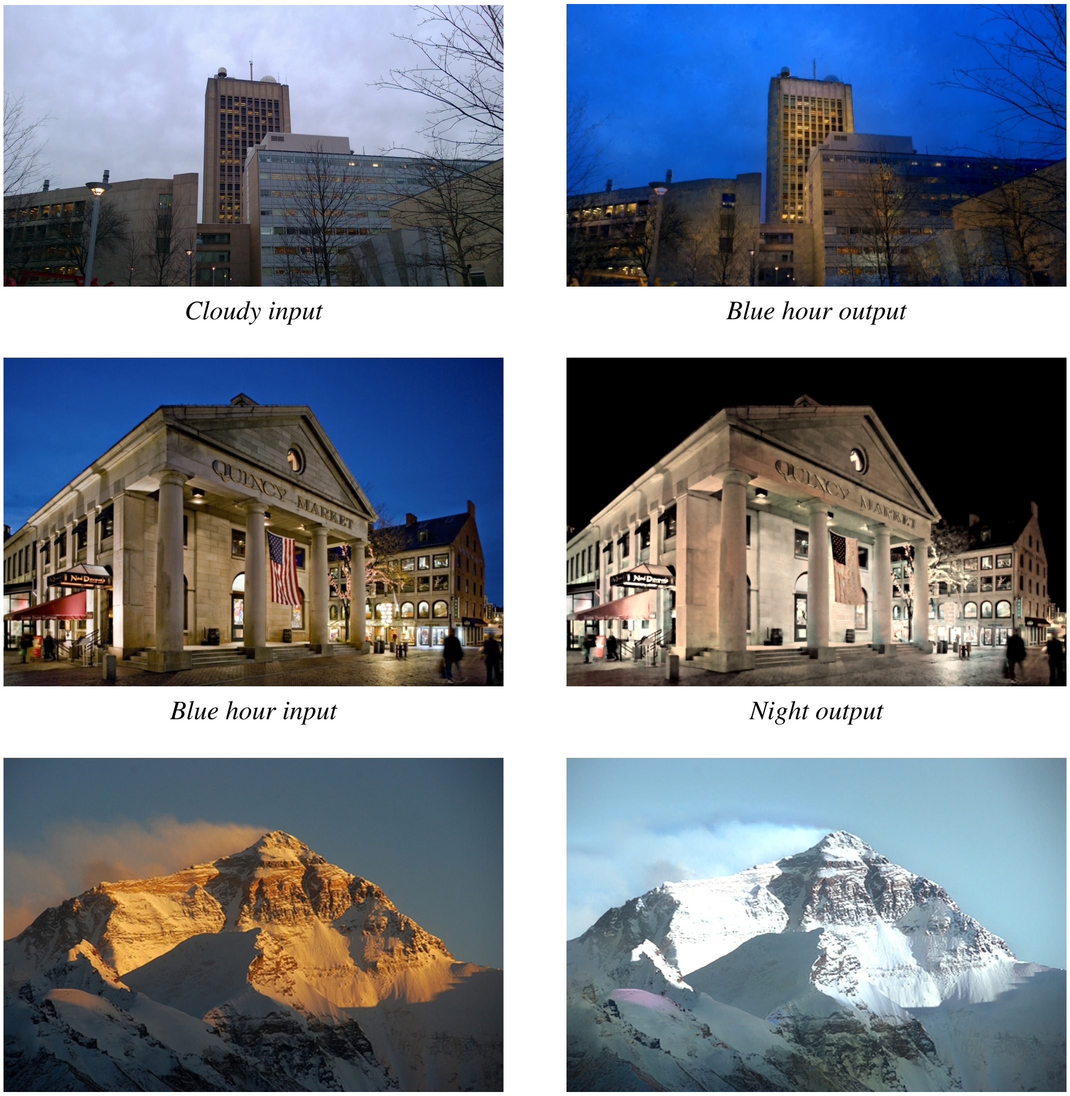
Input: landscape

Output: at blue hour

Input: a landmark

Output: at golden hour

Figure 3-8: *Our approach works for various scenes, including a building, a mountain, and a famous landmark. The dramatic changes for different times of day are visually plausible.*



Cloudy input

Blue hour output

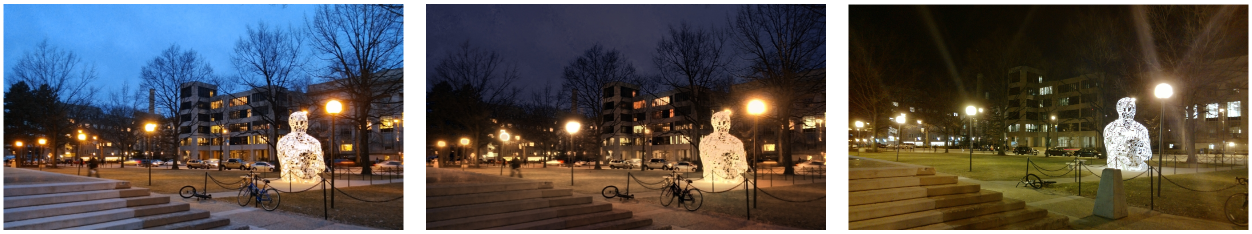
Blue hour input

Night output

Golden hour input

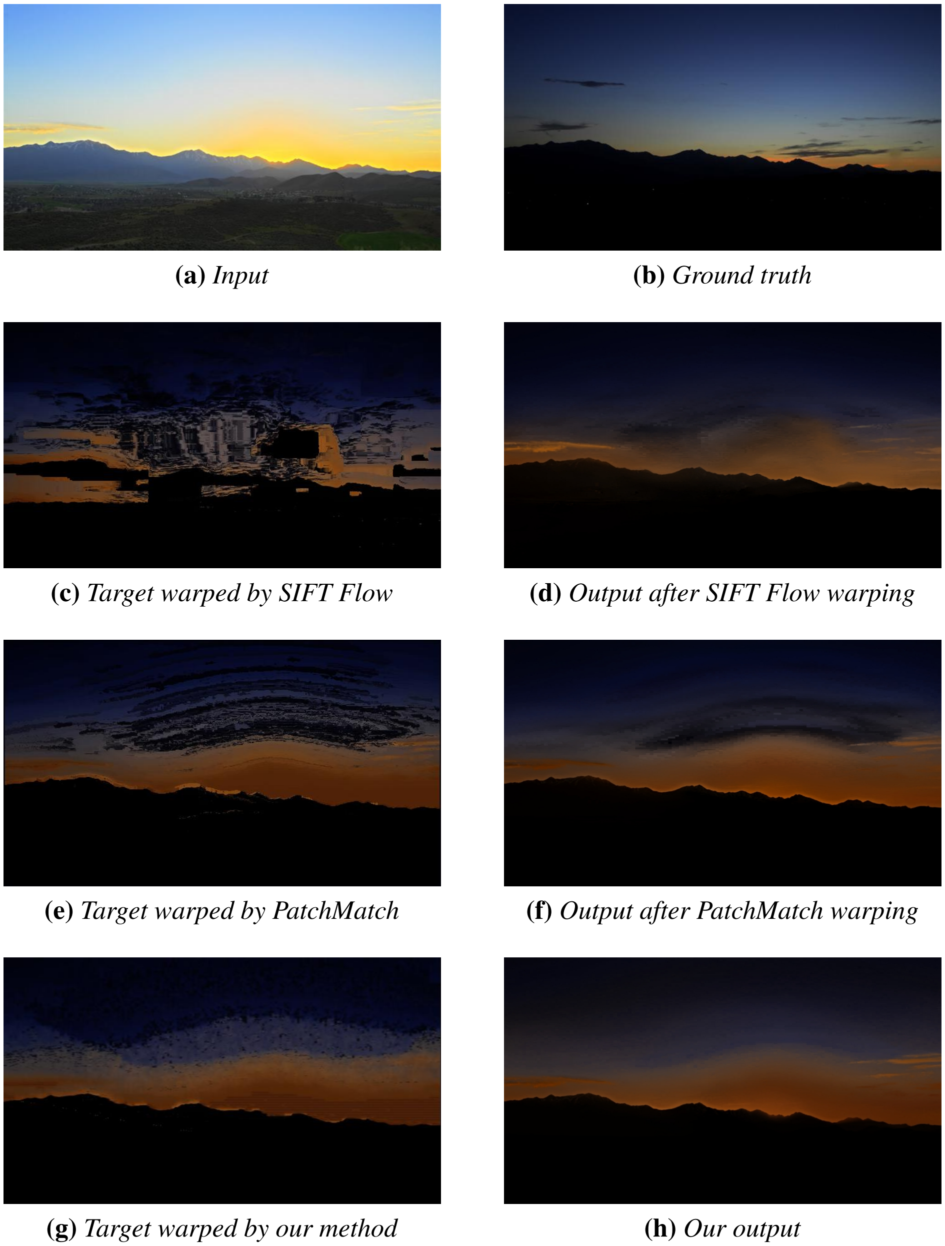
Day output

Figure 3-9: *Our method can take input at various times of day: cloudy, blue hour, and golden hour.*



(a) *Input* **(b)** *Our hallucinated result* **(c)** *Actual night photo of the same scene*

Figure 3-10: *We hallucinate a photo at night, and compare to a reference photo at the same location at night. Our result (b) is different from the actual photo (c) but nonetheless looks plausible.*



(a) *Input*

(b) *Ground truth*

(c) *Target warped by SIFT Flow*

(d) *Output after SIFT Flow warping*

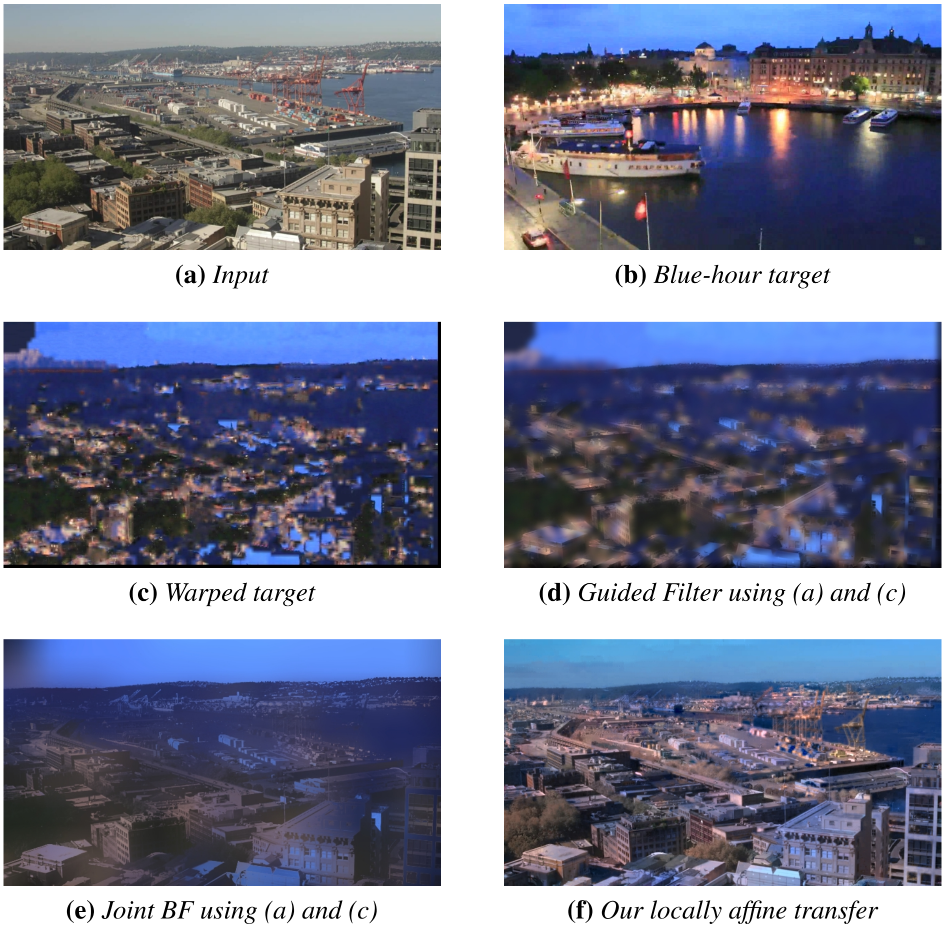
(e) *Target warped by PatchMatch*

(f) *Output after PatchMatch warping*

(g) *Target warped by our method*

(h) *Our output*

Figure 3-11: *We picked a frame in a time-lapse video (a) and hallucinate it at night. We compare the warped target frame and the final outputs by PatchMatch [7], SIFT Flow [76], and our approach. Since the input comes from a time-lapse video, we compare the outputs to the ground truth (b). Warping the target frame using PatchMatch or SIFT Flow produces unsightly discontinuities (c,e) that are still visible in the final outputs (d,f). In comparison, our algorithm does not introduce strong discontinuities in the warped frame (g) and produces a better result (h). While none of the outputs (d,f,h) is similar to the ground truth (b), ours is more plausible and visually more pleasing.*



(a) *Input*

(b) *Blue-hour target*

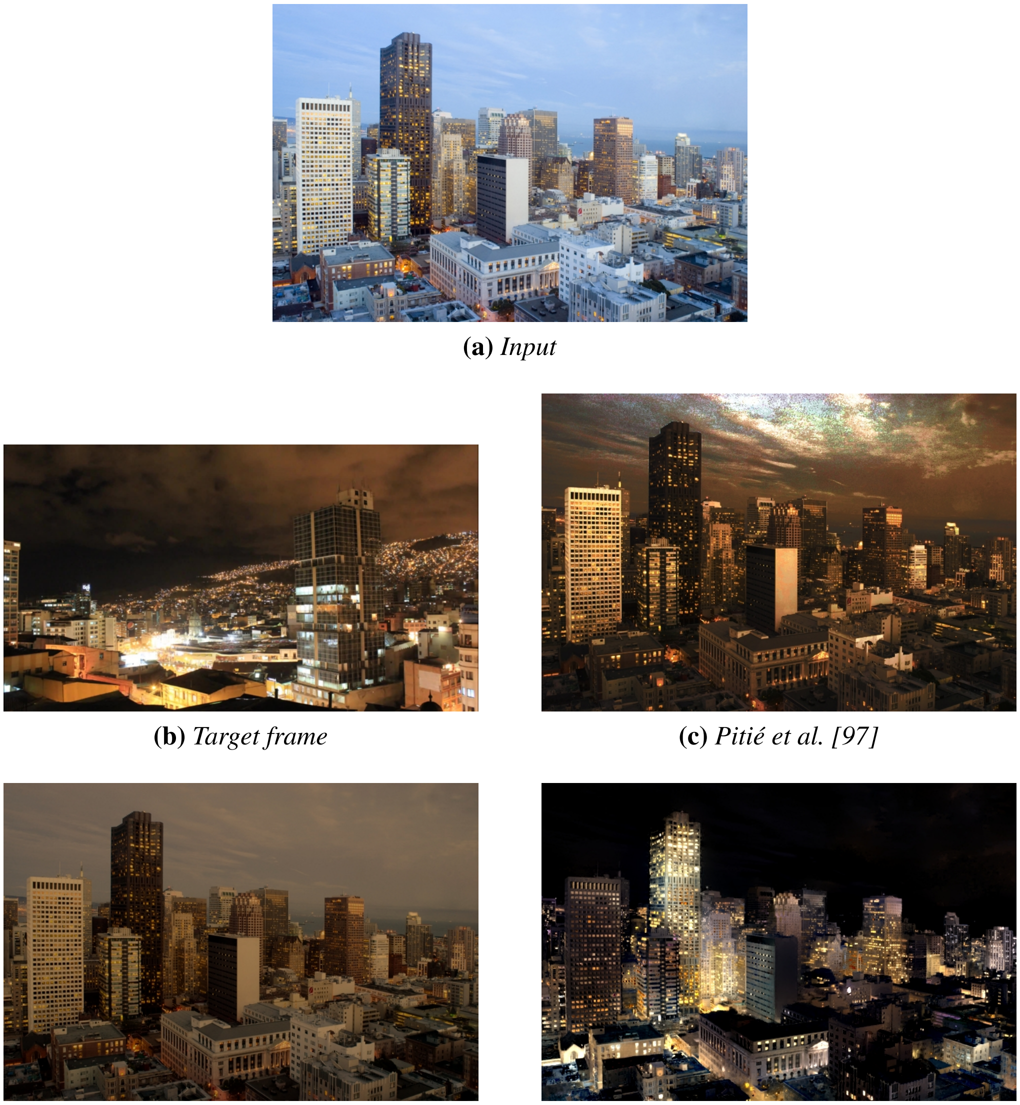
(c) *Warped target*

(d) *Guided Filter using (a) and (c)*

(e) *Joint BF using (a) and (c)*

(f) *Our locally affine transfer*

Figure 3-12: *We compare our model to the Joint Bilateral Filter [33, 92] and the Guided Filter [50]. For these filters, we use the warped target as the input, and the original input as guidance. In both cases, the results exhibit significant loss of details. In comparison, our approach produces sharp outputs.*



(a) *Input*

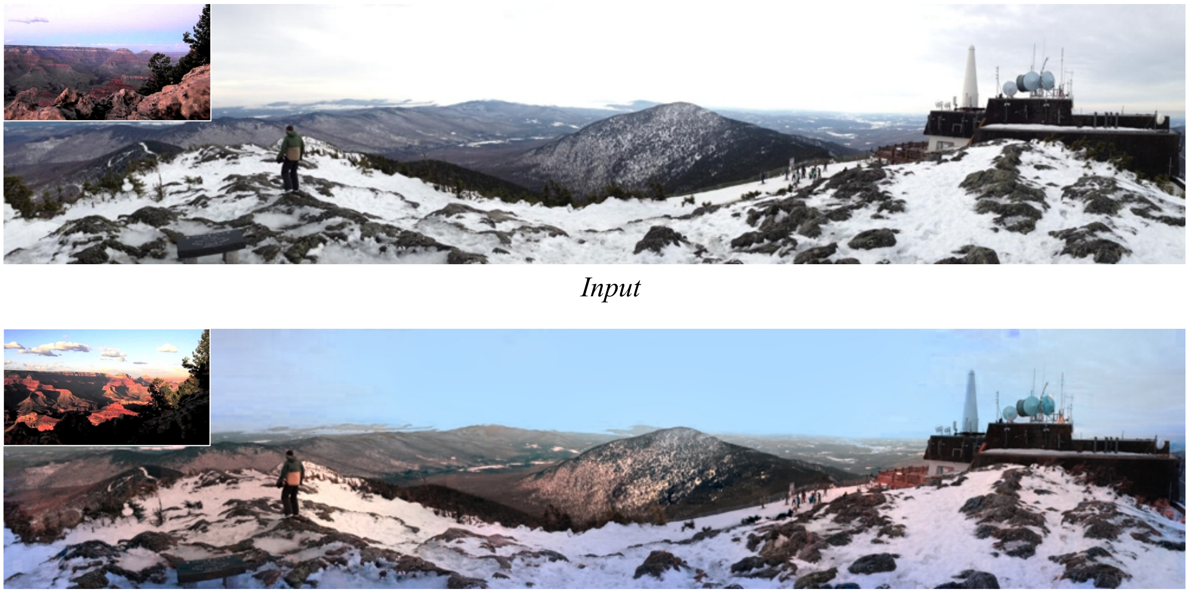
(b) *Target frame*

(c) *Pitié et al. [97]*

(d) *Reinhard et al. [100]*

(e) *Our result*

Figure 3-13: *Global methods generate only moderately convincing results (c,d). In comparison, our local affine transforms provide more flexibility in modeling spatially varying color changes, which produces a better result (e).*



Input

Light transfer

Figure 3-14: *The two target frames shown in the insets are taken at close times but under different lighting conditions. Our method increase the vibrancy by transferring the lighting to an ordinary photo.*

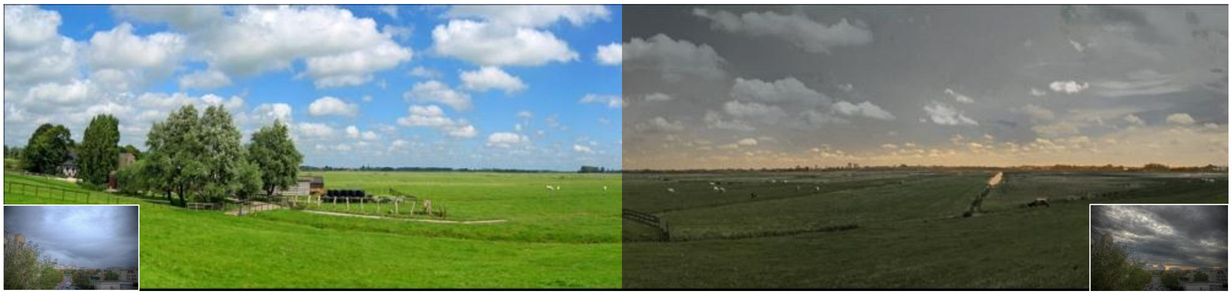


Figure 3-15: *We hallucinate the weather for the right half of this panorama. We transfer the difference between two kinds of weather in the time-lapse to a photo.*

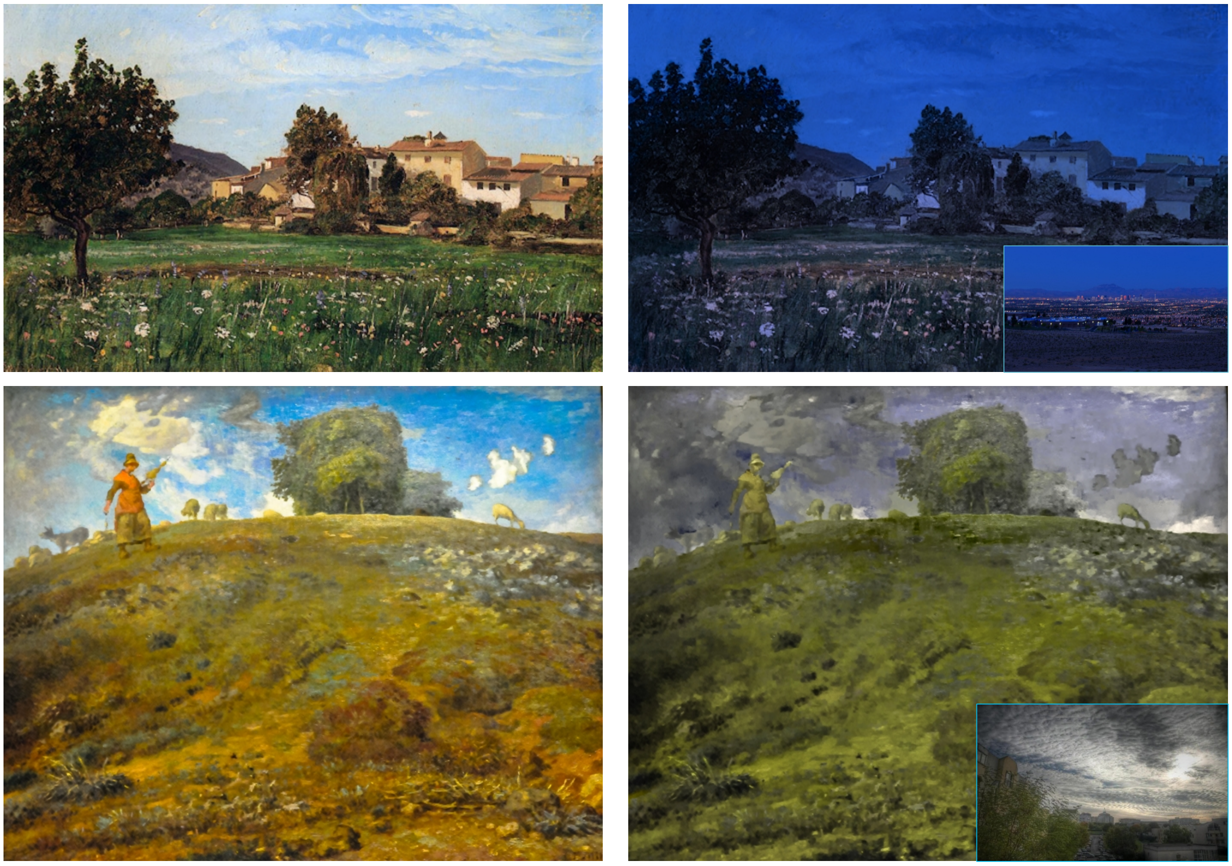


Figure 3-16: *Paintings in realism. From top to bottom:- "In the Auvergne", Jean-Francois Millet. "Lourmarin", Paul-Camille Guigou. We hallucinate the top one to blue hour, and handpick a cloudy frame for the bottom one.*

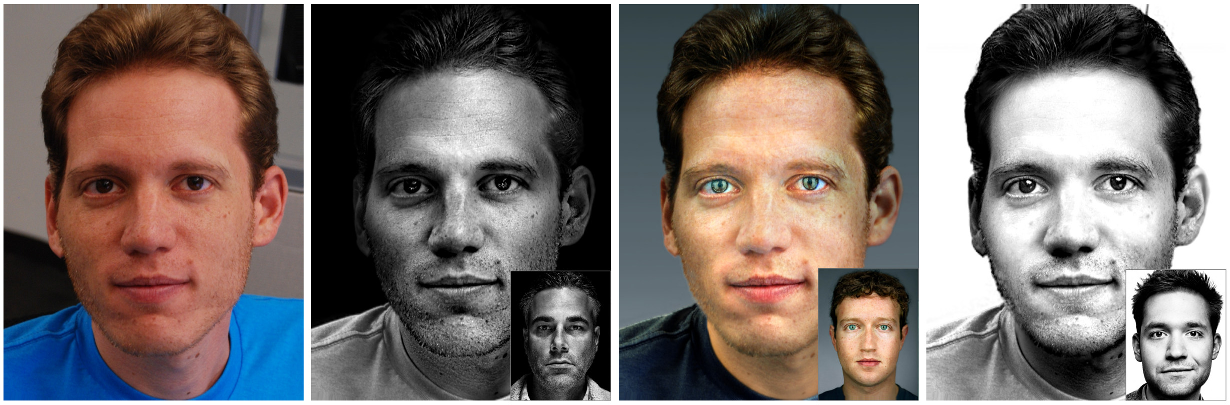
THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Portrait Style Transfer

4.1 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,



(a) ***Input***: *a face photo* **(b)** ***Outputs***: *new headshots with the styles transferred from the examples in insets*

Figure 4-1: *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.*

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.* [6] 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. Figure 4-1 previews some results by our algorithm. 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 work 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.

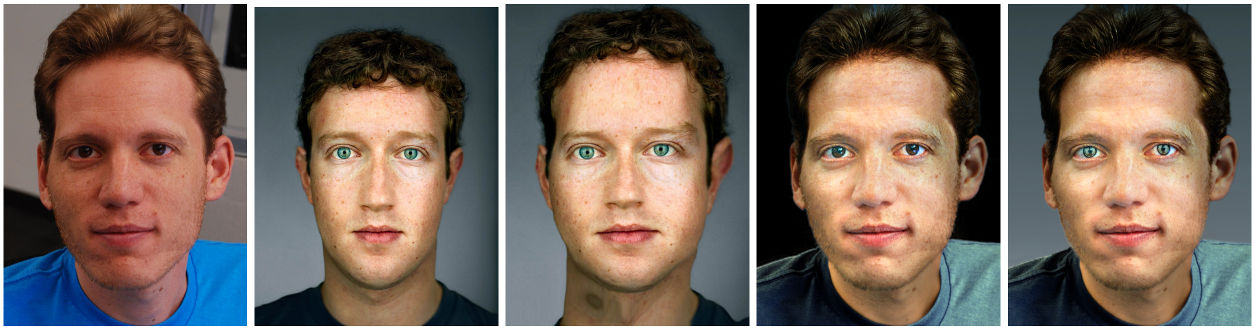
▷ We describe an automatic algorithm to select a suitable example among a collection of consistently stylized headshots.

4.2 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., [6, 45, 97, 100, 118, 120]. 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.* [100], Pitié *et al.* [97], and Tai *et al.* [120], we transfer the color palette. Like Sunkavalli *et al.* [118], we use a multiscale image decomposition [15, 88]. We rely on a dense correspondence between the input and example akin to HaCohen *et al.* [45] and, like Bae *et al.* [6], 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.* [22] locally change image colors to produce images with a more harmonious color palette, Wang *et al.* locally apply color schemes [124] and transfer the look of specific cameras [125], and Shih *et al.* [111] 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 Face-specific applications has been gaining interests in computer graphics [38, 62]. Joshi *et al.* [61] and An and Pellacini [4] successfully transfer color balance and overall exposure. Tong *et al.* [122] and Guo *et al.* [44] transfer make-up. Brand and Pletscher [13] and Leyvand *et al.* [74] improve face appearance. In comparison, we focus specifically on photographic style transfer, including aspects such as skin texture, local contrast, and light properties.



(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 4-2: *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).*

.

Face Synthesis Our work is related to face synthesis [77, 83] in that we generate a portrait that can differ dramatically from the input. However, unlike Mohammed *et al.* [83], we seek to retain the identity of the person shown in the input photo. Liu *et al.* [77] 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., [91, 128, 141]. 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.

4.3 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 4-2 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 (§ 4.3.1). Then, we transfer the local statistics of model onto the input (§ 4.3.2) — this is the core of our approach. Finally, we transfer the eye highlights and the background (§ 4.3.3).

4.3.1 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 [105]. 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.* [61]. Then, we morph the example to the input using the segments on the face template [9]. This initial estimation often successfully aligns the eyes and mouth, but misses important edges such as the face contour and mouth. The final step is to refine the correspondence using SIFT Flow [78]. Figure 4-2 (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.

4.3.2 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 [52]. Our technique builds upon the notion of power maps [6, 75, 82, 114] to estimate the local energy in each image frequency subband. Similarly to Li *et al.* [75], 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 4-3, 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:

$$L_\ell[I] = \begin{cases} I - I \otimes G(2) & \text{if } \ell = 0 \\ I \otimes G(2^\ell) - I \otimes G(2^{\ell+1}) & \text{if } \ell > 0 \end{cases} \tag{4.1}$$

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

$$R[I] = I \otimes G(2^n) \tag{4.2}$$

Local Energy Inspired by power maps [6, 75, 82, 114], 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:

$$S_\ell[I] = L_\ell^2[I] \otimes G(2^{\ell+1}) \tag{4.3}$$

For the example image E, we account for the correspondence field that we have computed previously (§ 4.3.1). Using $W(\cdot)$ for the warping operator defined by this field, we compute:

$$\tilde{S}_\ell[E] = W(S_\ell[E]) \tag{4.4}$$

where we compute $S_\ell[E]$ with Equation 4.3. Estimating the energy before warping the data avoids potential perturbations due to distortion and resampling.

Robust Transfer Using these two estimates (Equations 4.3 and 4.4), 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:

$$L_\ell[O] = L_\ell[I] \times \text{Gain} \tag{4.5a}$$

$$\text{with Gain} = \sqrt{\frac{\tilde{S}_\ell[E]}{S_\ell[I] + \epsilon}} \tag{4.5b}$$

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 4.3. The gain maps (Equation 4.5b) in Figure 4-3 show how they vary over space to capture local contrast. Intuitively, gain values below 1.0 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 mismatch. For instance, if the example has a mole and not the input, the gain map (Equation 4.5b) 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 (Equation 4.5b) has low values along the glasses, which produces unsightly phantom glasses in the output. Figure 4-4 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:

$$\text{RobustGain} = \max(\min(\text{Gain}, \theta_h), \theta_l) \otimes G(\beta\, 2^\ell) \tag{4.6}$$

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.* [128]

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 4.3 strikes a balance and transfers the example style while preserving the input identity. Figure 4-5 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 (Equations 4.1, 4.2, 4.3, and 4.6) as follows:

$$\text{Image} \otimes G \rightsquigarrow \frac{(\text{Image} \times \text{Mask}) \otimes G}{\text{Mask} \otimes G} \tag{4.7}$$

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

In practice, we run GrabCut [103] initialized with a face detection result to find a binary mask that we refine using the Matting Laplacian [72]. As shown in Figure 4-6, 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.

4.3.3 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 [26]. 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 [72] (Figure 4-7). 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., [8], 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 radius.

Background The background also contributes to the mood of a portrait [93, § 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 4-2e illustrates this process.

4.3.4 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 [3, 123], we use the local energy S in Equation 4.3 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.

For our experiments, we use a portrait collections database by three 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.

4.4 Results

Figure 4-8 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 4-9 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 on our project website[1]. To collect the dataset, we searched for photos with the keywords “headshot" and “portrait," then automatically 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 4-10 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 4-11 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.

Comparison to the reference image Figure 4-12 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 Appendix B, we provide the result of using the reference in Figure 4-12 as the example. This is to test the ideal scenario that the database is sufficiently large so that we can find an example

[1] `http://people.csail.mit.edu/yichangshih/portrait_web/#results`

almost identical to the input.

Automatic example selection Figure 4-13 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 Appendix B, 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 4-14. 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 4-15 shows examples of manual corrections applied to correct failures of the automatic method. These are the only results in this work 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.

4.4.1 Comparison with Related Work

Figure 4-16 compares our method with related work in multiscale transfer and color transfer. For Sunkavalli *et al.* [118], 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 [97, 100]. Our result captures details more faithfully because our method is local.

Figure 4-17 shows comparisons to Bae *et al.* [6] 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 Appendix B, we provide black-and-white comparisons to other related methods.

We also attempted to compare with HaCohen *et al.* [45], 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 4-8. In Appendix B, 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 Appendix B, we provide comparisons using the adapted methods, as well as the comparisons on all three styles used in Figure 4-8.

4.4.2 Extensions

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 [14] within the video. First, we choose the exemplar that best matches the first video frame, using the automatic selection described in § 4.3.4. 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 4-18 shows that we successfully transfer the style to the input video, even with the frames of very different facial expressions. Our video result at our project website[2] shows good temporal coherence in the presence of extreme facial expressions.

Facial makeup transfer Figure 4-20 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 Appendix B, we show a comparison with the state-of-the-art works [44, 122].

[2] `http://people.csail.mit.edu/yichangshih/portrait_web/#video`

4.5 Discussion and Conclusion

The main novelty of our work 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 4-19 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.

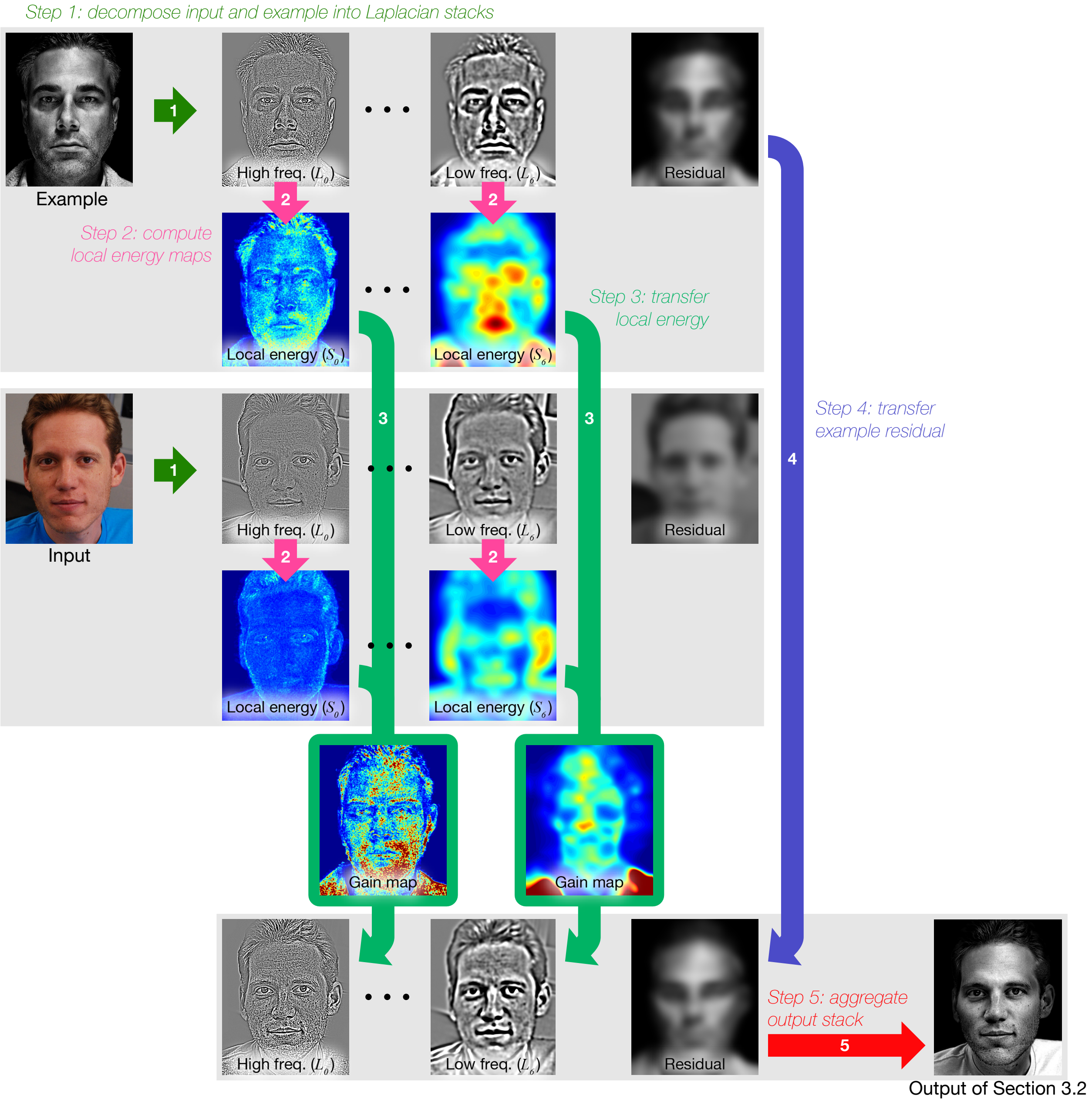
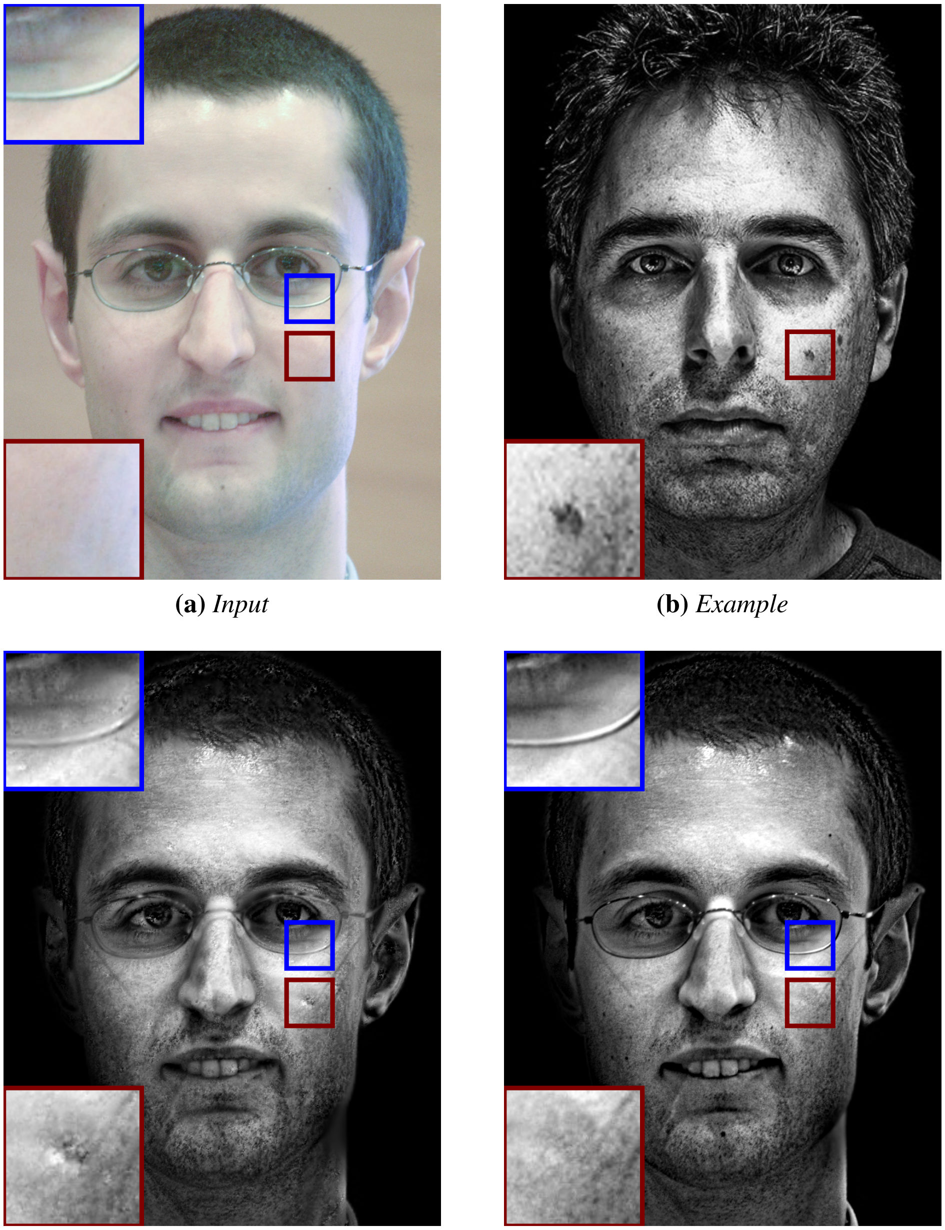

Step 1: decompose input and example into Laplacian stacks
Example
1
High freq. (L_0)
. . .
Low freq. (L_6)
Residual
2
Step 2: compute local energy maps
Local energy (S_0)
Local energy (S_6)
Step 3: transfer local energy
3
4
Step 4: transfer example residual
Input
Gain map
Step 5: aggregate output stack
5
Output of Section 3.2

Figure 4-3: *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.*



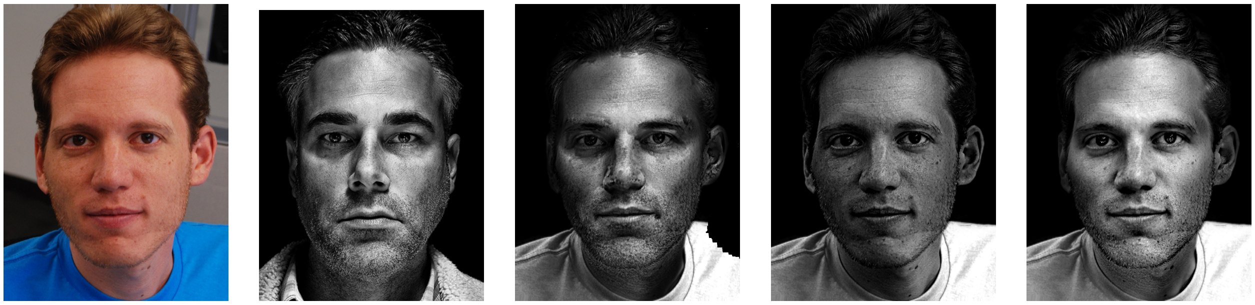
(a) *Input*

(b) *Example*

(c) *Without robust transfer*

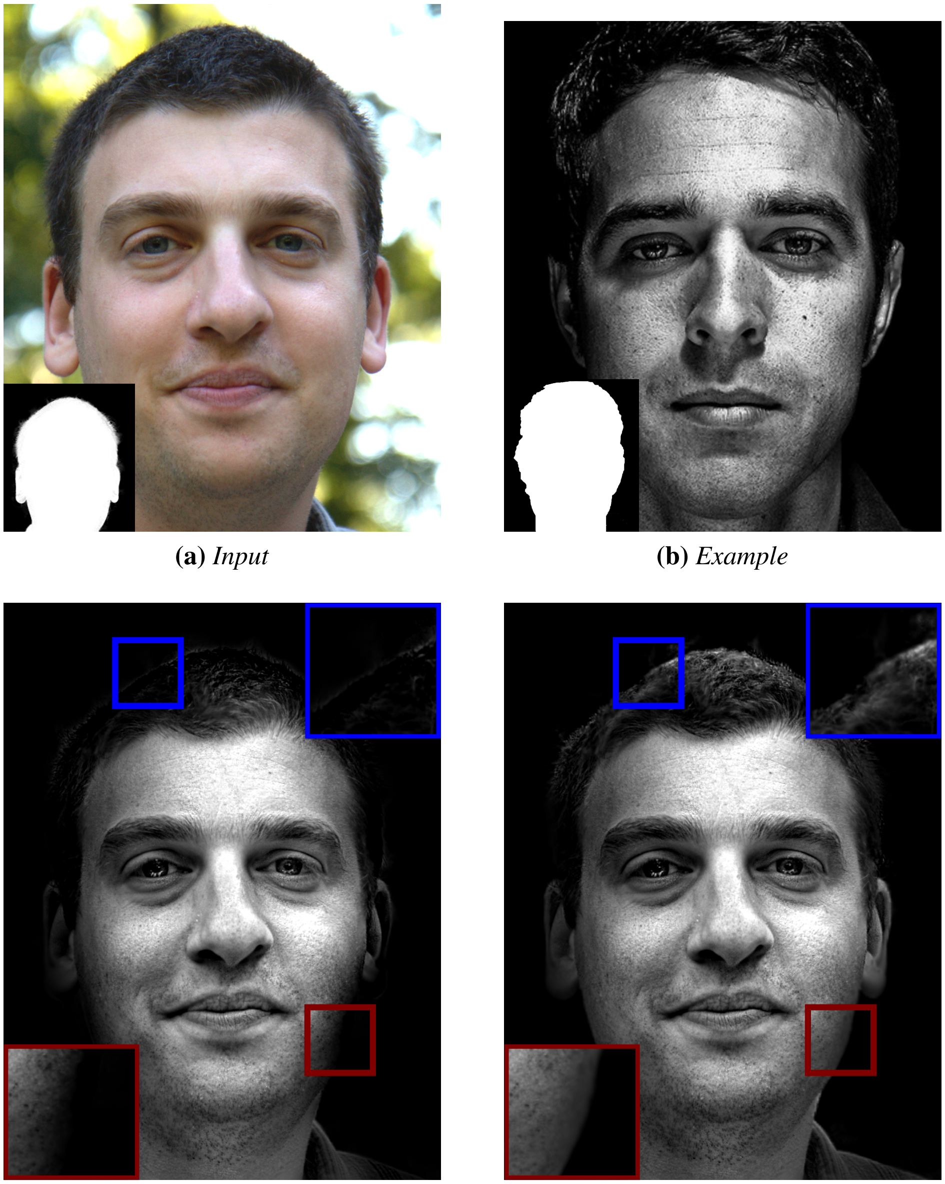
(d) *Our robust transfer*

Figure 4-4: *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.*



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

Figure 4-5: *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 Equation 4.3 produces a successful transfer.*



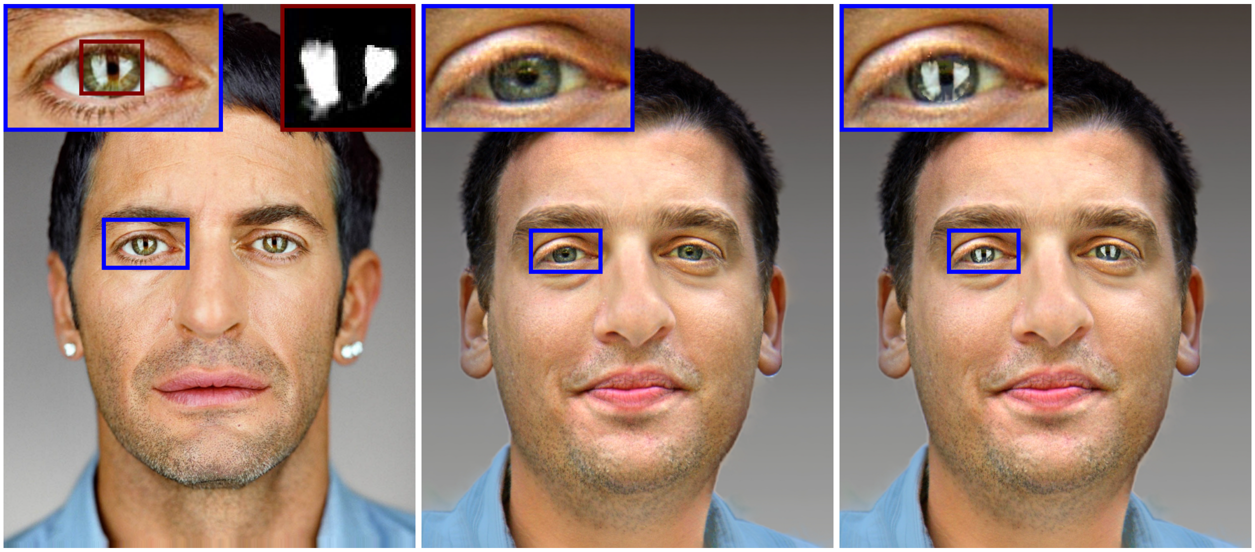
(a) *Input*

(b) *Example*

(c) *Without adapting to face mask, the face contour disappeared*

(d) *Our Laplacian that is adapted to face mask*

Figure 4-6: *(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 4-7: *Taking the input in Figure 4-6, 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.*

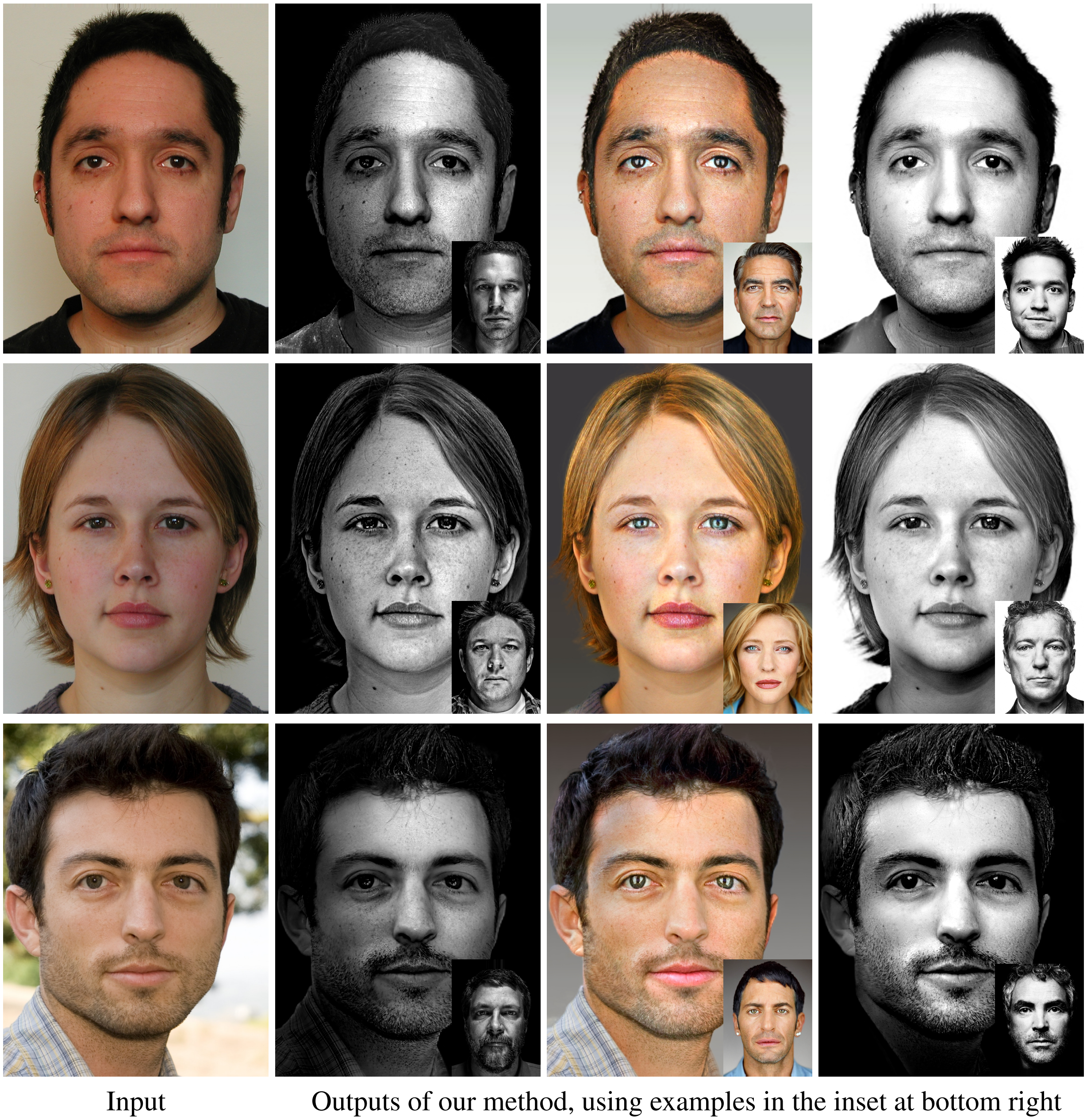
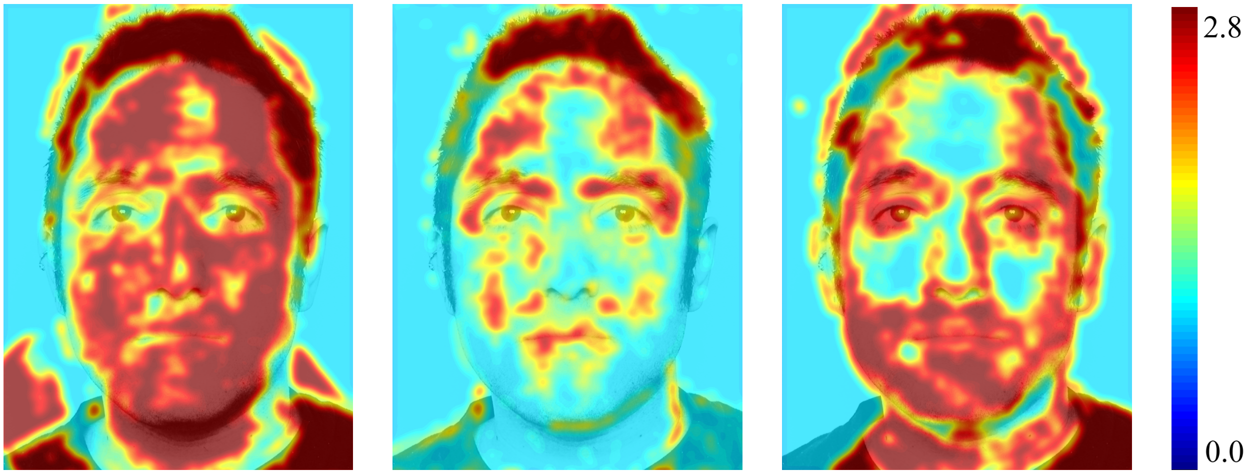

Input
Outputs of our method, using examples in the inset at bottom right

Figure 4-8: *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.*

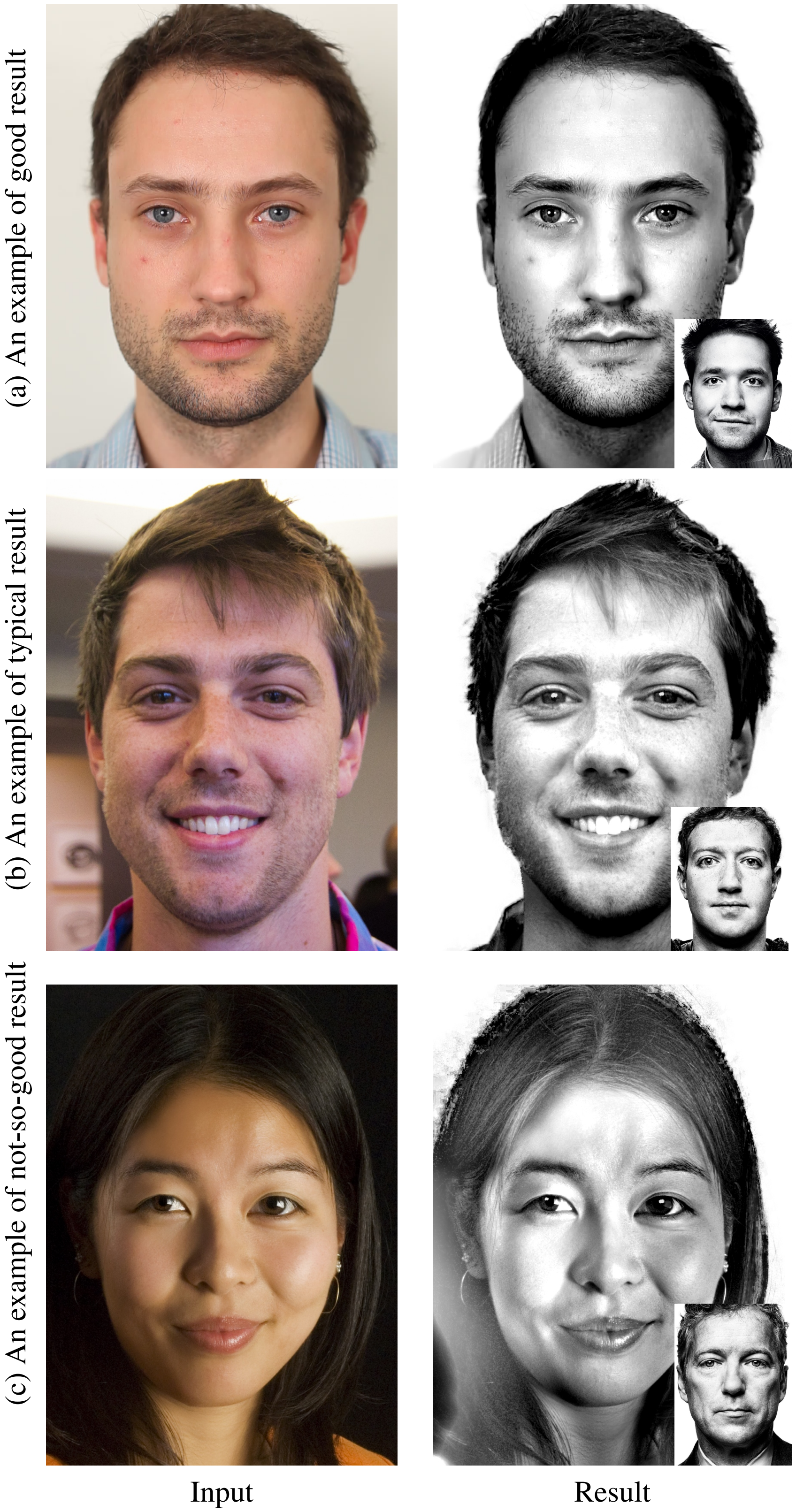


(a) *Gain map at $\ell = 2$ for the low-key style, (row 1, col 2) in Figure 4-8*

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

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

Figure 4-9: *We overlay the gain maps of the first row in Figure 4-8 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.*



(a) An example of good result

(b) An example of typical result

(c) An example of not-so-good result

Figure 4-10: *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. (a) ©YiChang Shih*

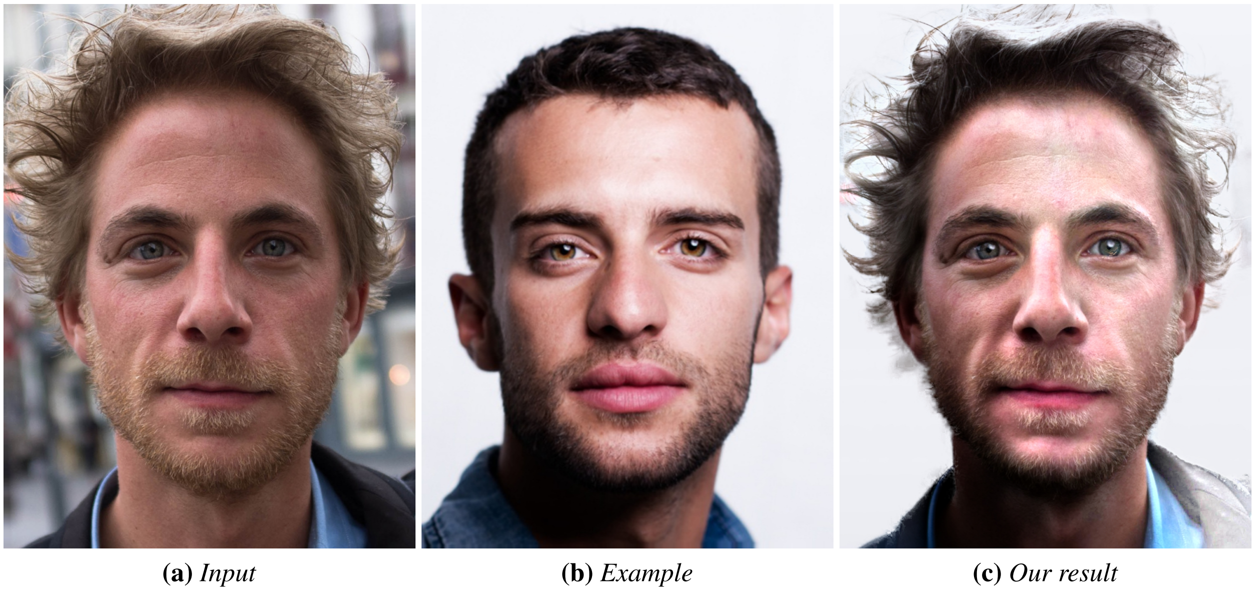


Figure 4-11: *Our method captures some amount of lighting change in the case that the lighting in the example (b) is different from the input (a).*

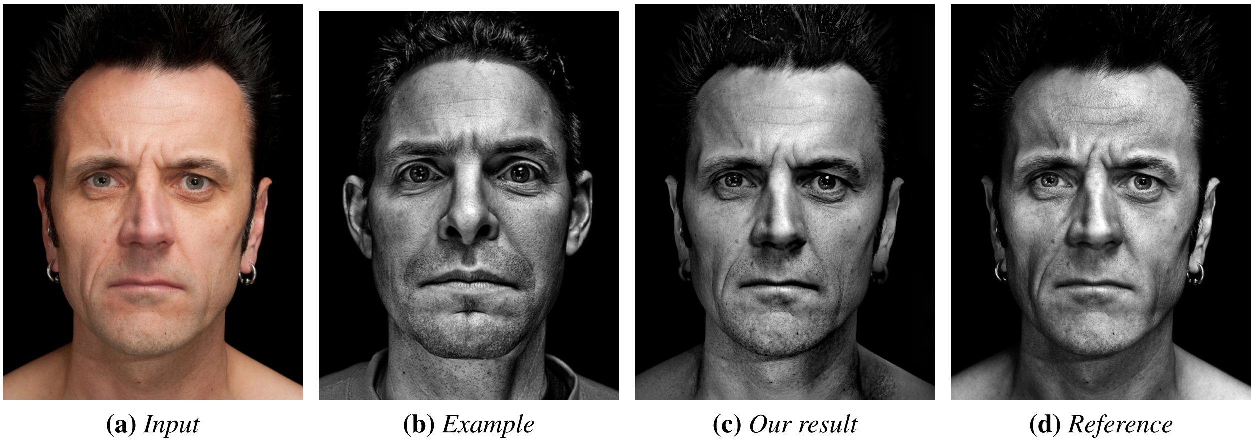
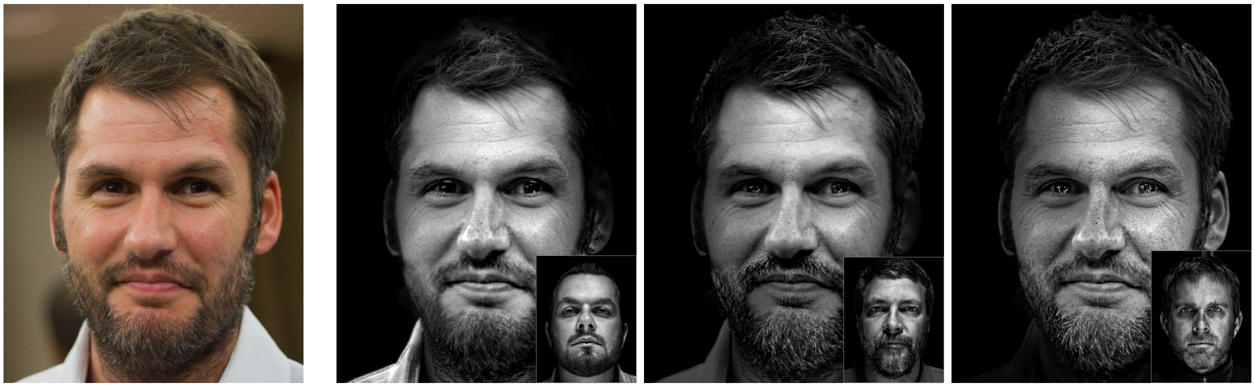


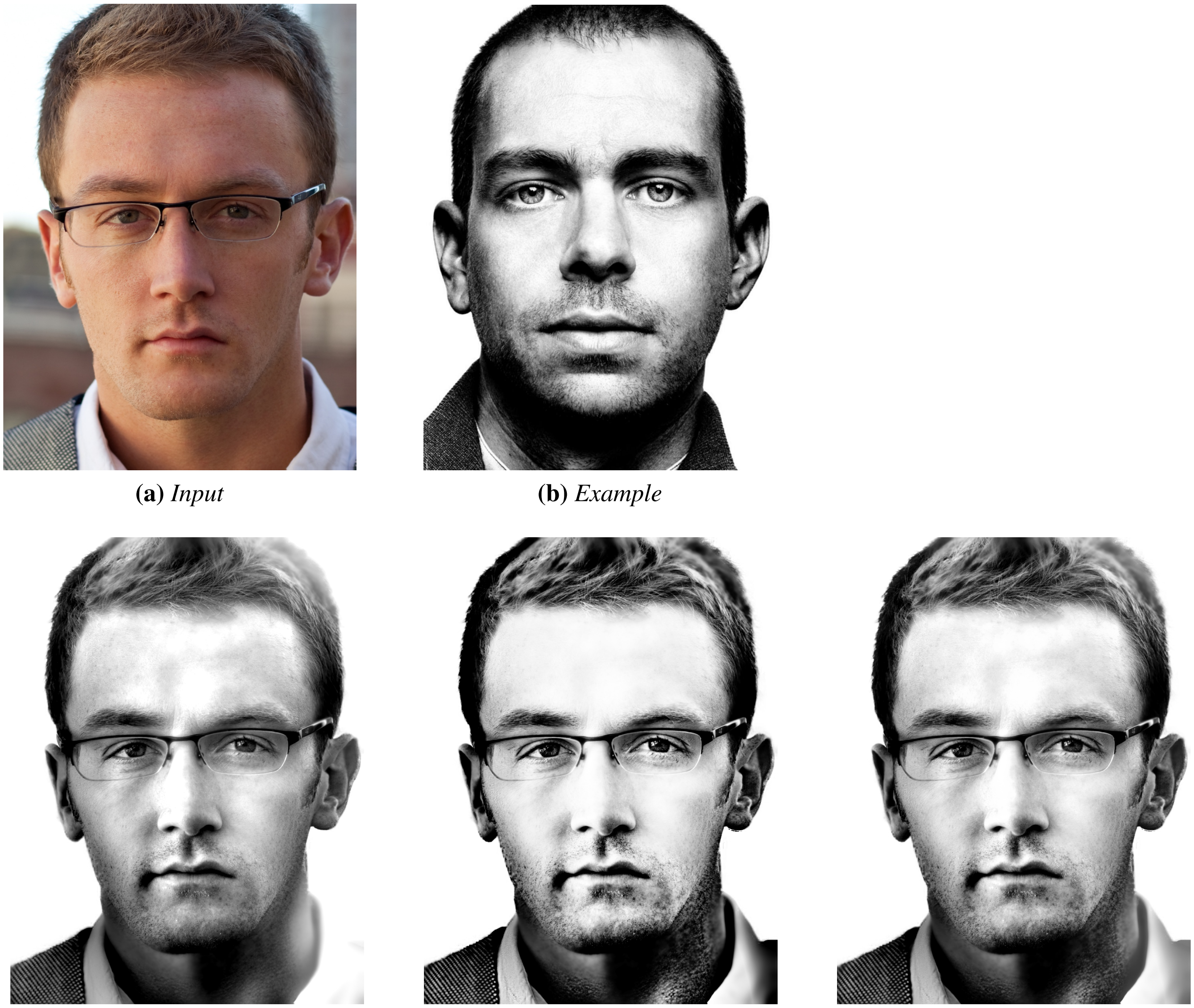
Figure 4-12: *We compare the transfer result in (c) to a reference portrait (d) made by the same photographer who created the example in (b). While our transfer is not exactly identical, it looks visually similar.*



(a) *Input*

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

Figure 4-13: *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.*



(a) *Input*

(b) *Example*

(c) *Only local matching*

(d) *Transferring histogram from (b) to (c)*

(e) *Averaging (c) and (d)*

Figure 4-14: *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).*

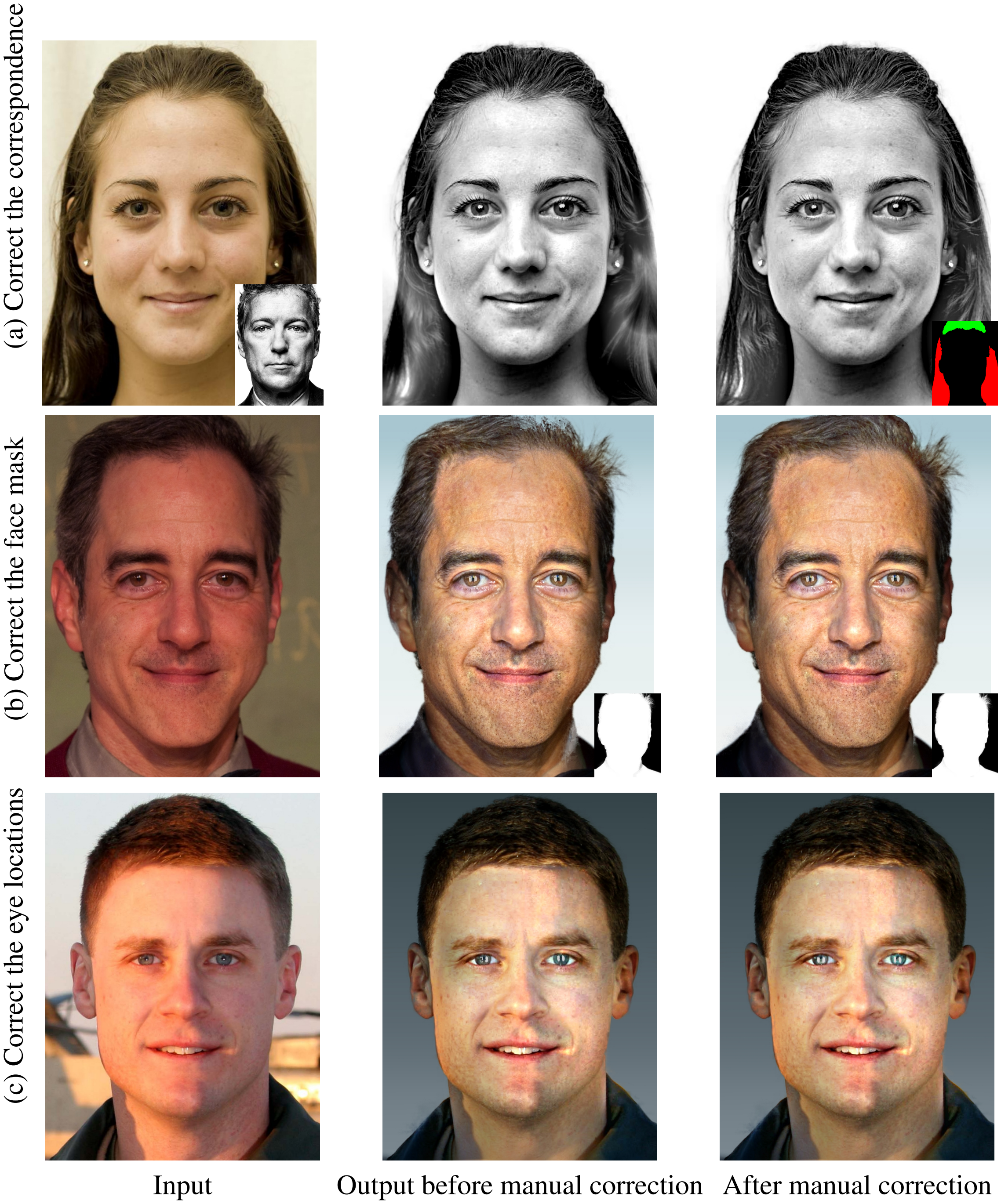
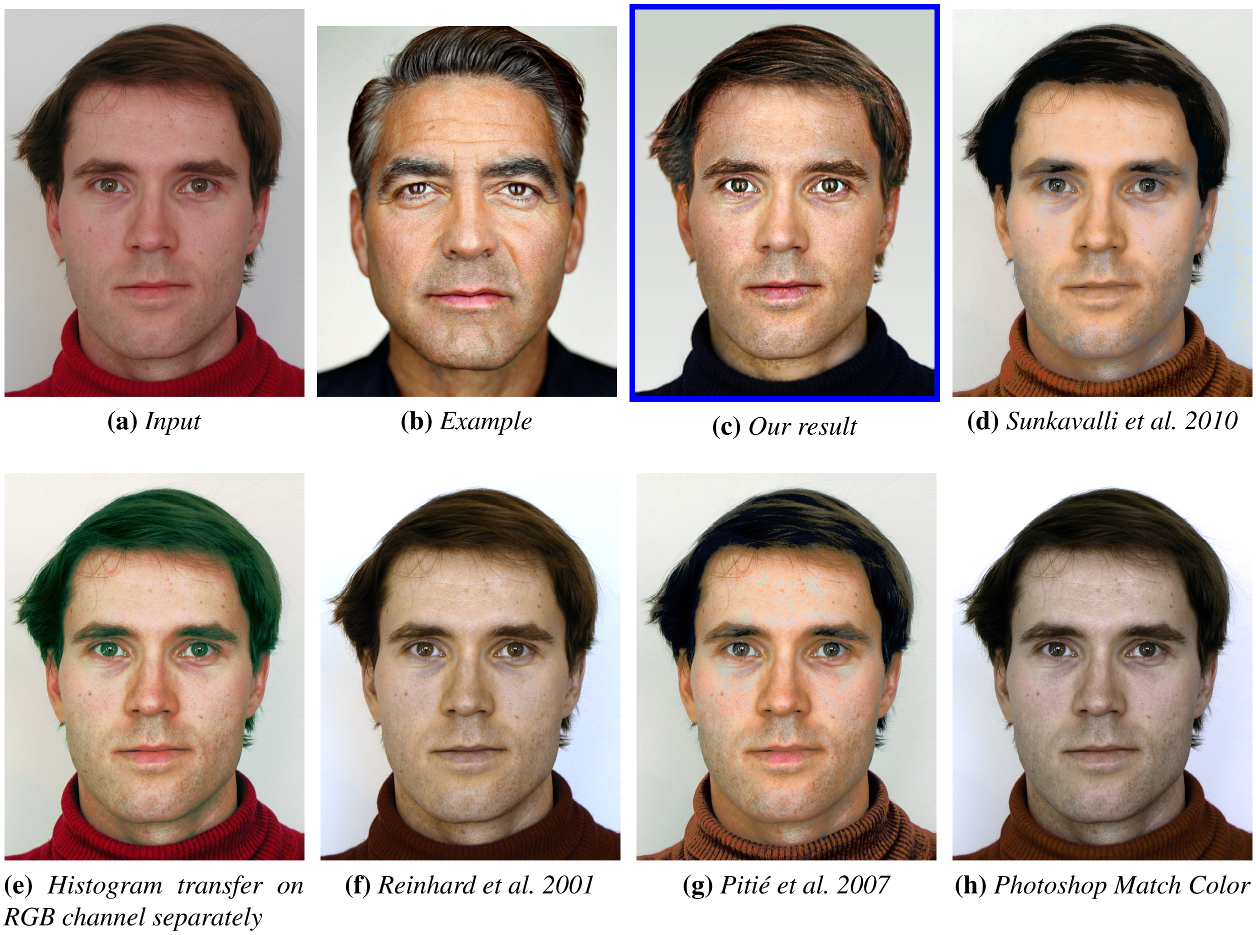

(a) Correct the correspondence
(b) Correct the face mask
(c) Correct the eye locations
Input
Output before manual correction
After manual correction

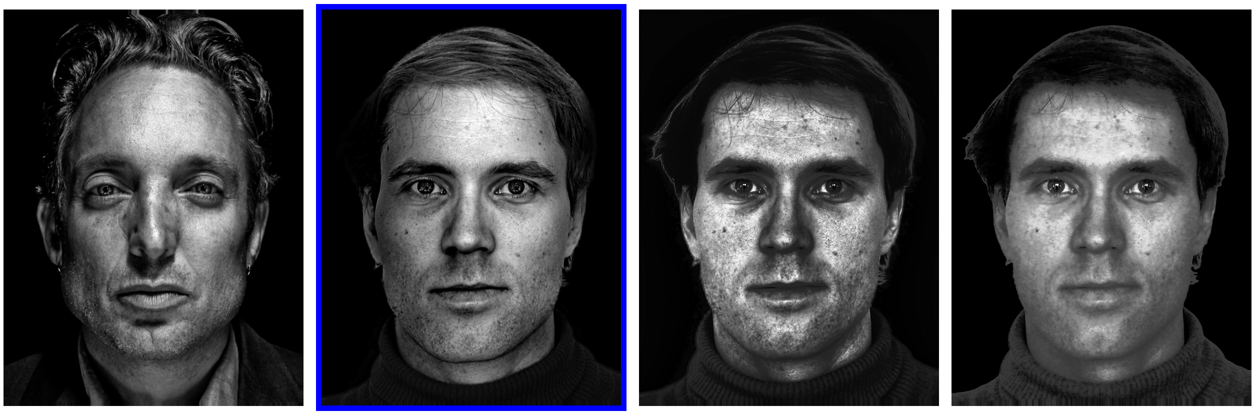
Figure 4-15: *We propose manual corrections to fix the rare failure cases of the automatic method. (a) The mismatching between the input hair and the example (in the bottom right inset) results in artifacts on the output. We correct the correspondence through a user-provided map shown in the inset in 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*

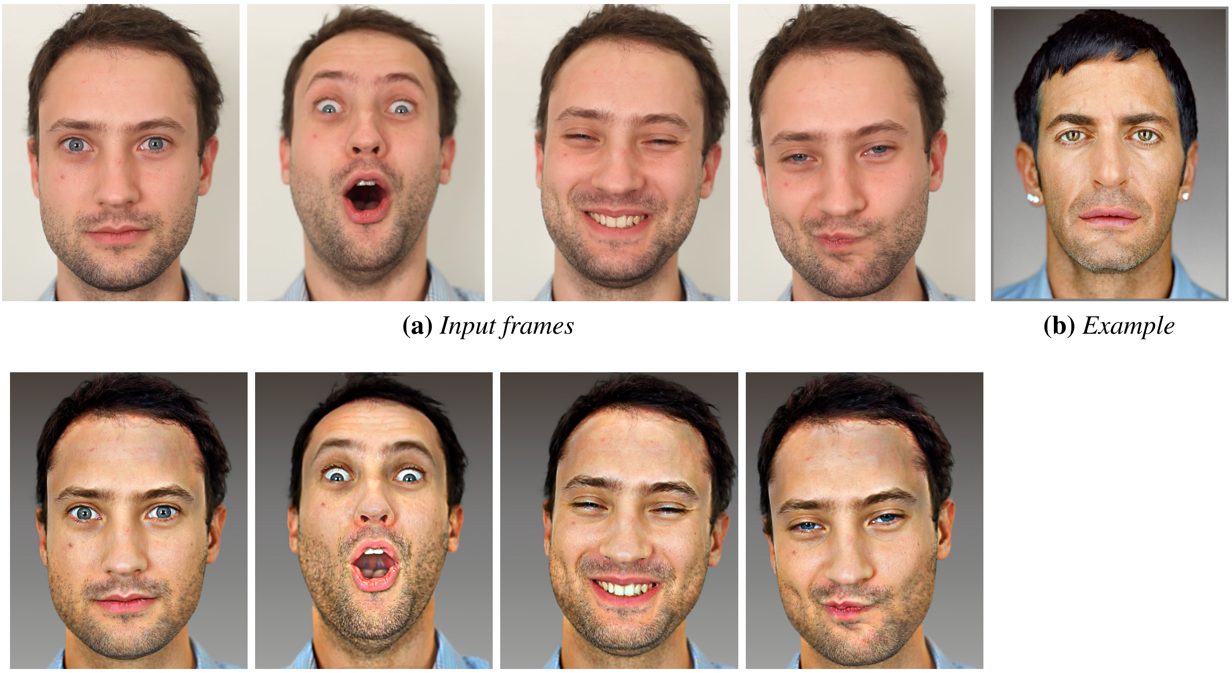
(e) *Histogram transfer on RGB channel separately* **(f)** *Reinhard et al. 2001* **(g)** *Pitié et al. 2007* **(h)** *Photoshop Match Color*

Figure 4-16: *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*

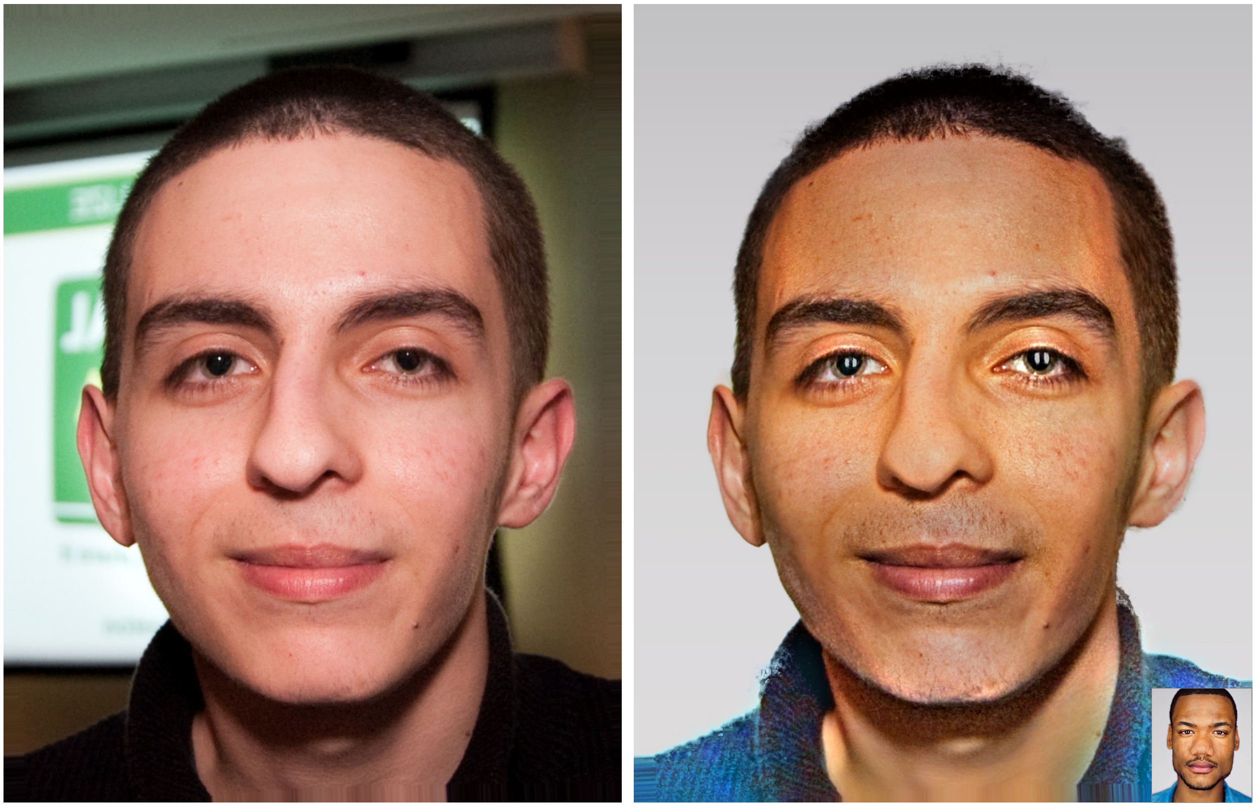
Figure 4-17: *We compare with Bae et al. that works on tonal (black-and-white) transfer, as well as the multi-scale transfer of Sunkavalli et al. 2011]. These methods have been adapted to use the face mask because the input and example have different background colors.*



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

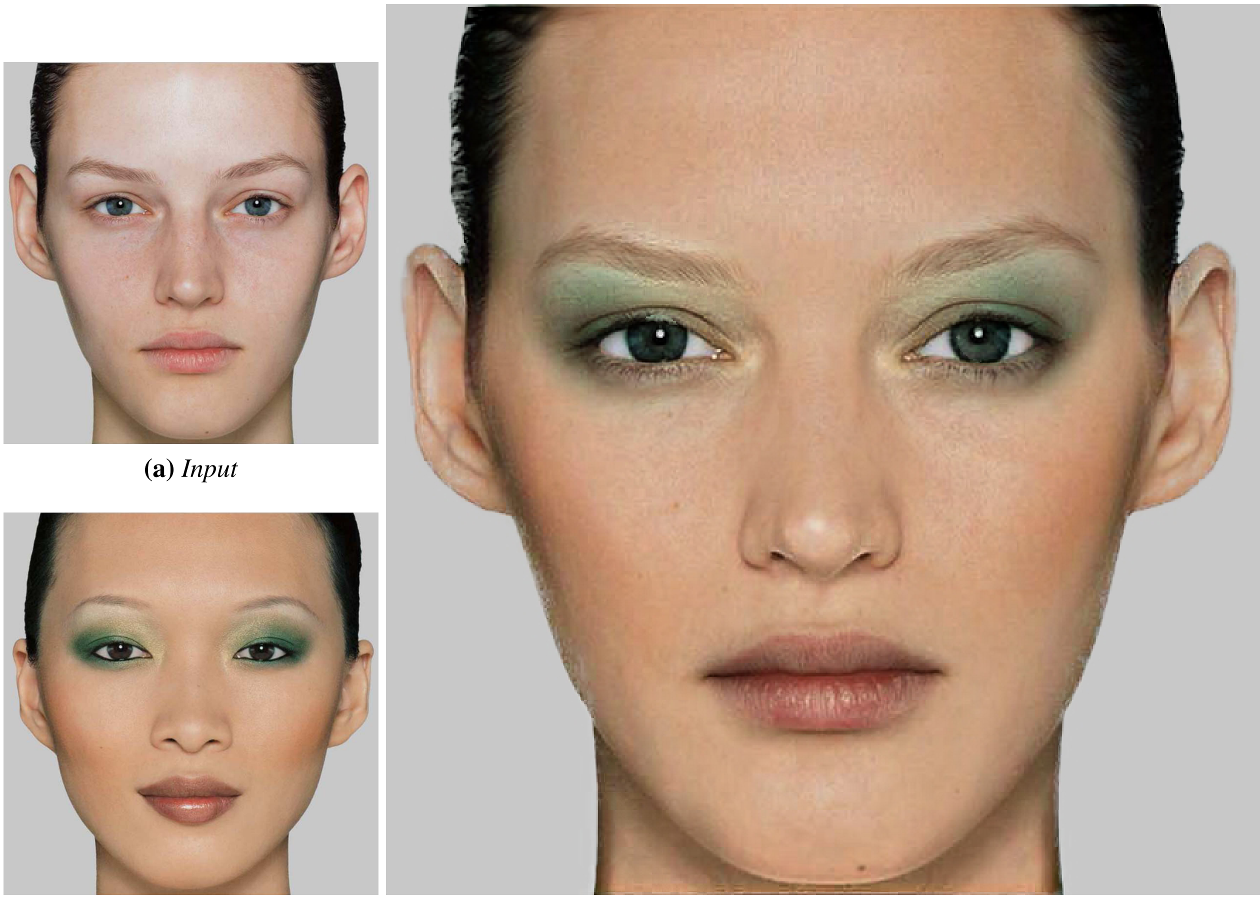
(c) *Output frames*

Figure 4-18: *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 videos on our project website for more results. (a)(c) ©YiChang Shih.*



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

Figure 4-19: *A failure case: matching a white male to an African male in the inset creates an unrealistic result.*



(a) *Input*

(b) *Example*

(c) *Our makeup transfer result*

Figure 4-20: *We extend our method to makeup transfer with minor modification. We transfer the example makeup in (b), taken from a professional makeup book [86]. The result in (c) shows that skin foundation, lip color and eye shadow are successfully transferred to the input (a).*

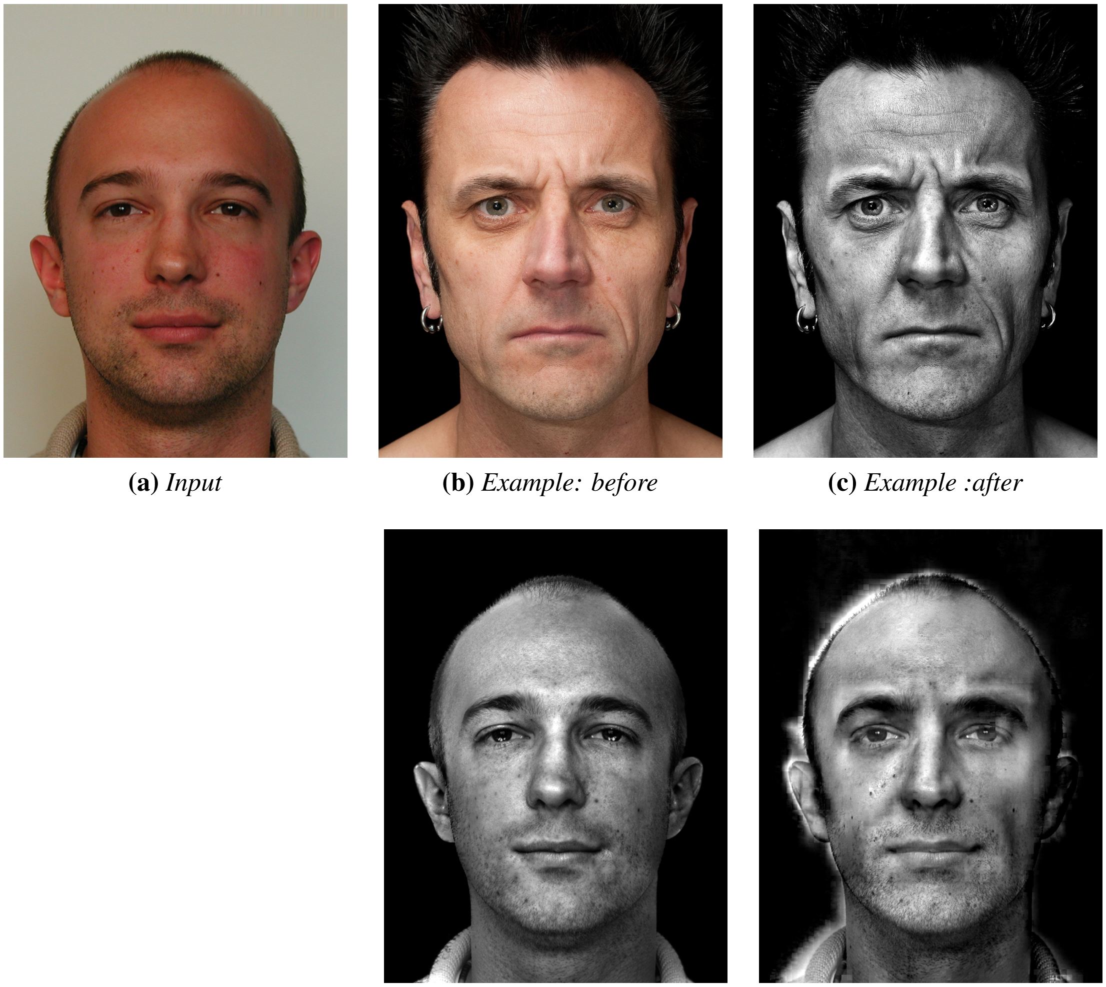
THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Conclusion

In this dissertation, we have addressed two challenging problems of image stylization that require dramatic alterations of image appearances. In the first problem, we help users to render their pictures at their desired times of day, such as converting daytime pictures to the golden hour. In our second problem, we allow people to enjoy stylized portraits of themselves by rendering an input portrait with styles created by renowned photographers, such as the bright and warm color styles of Martin Schoeller. These two popular photo categories, outdoor photographs and portraits, cover most topics in photo retouching [10][1]. In essence, the two works share quite a bit of similarity. Our first key idea is to leverage the immense power of image data. In time hallucination, we leverage a database of time-lapse videos taken at more than five hundred different locations, including landscapes and city skylines. For portrait stylization, we employ image collections retouched by celebrated photographers. To leverage the data, our second idea proposes a local style transfer algorithm. We use a dense correspondence field that respects the scene semantics in an input and an example, and then locally transfers the image style through the correspondences. In both works, our local approaches achieve spatially-variant and one-to-many color transfer.

[1] They randomly sampled 106 tutorials in photo retouching, and found out that 40 of them are about portraits, and 27 of them are about landscape photographs.



(a) *Input* **(b)** *Example: before* **(c)** *Example :after*

(d) *Multiscale transfer* **(e)** *Locally affine*

Figure 5-1: *Given an input portrait (a) and a pair of exemplar portraits before and after retouching (b-c), we compare our multi-scale technique in portrait stylization (§ 4) to the locally affine transfer intended for time hallucination (§ 3). The multi-scale method in (d) successfully reproduces the style in the example (c), while preserving the identity of the input. Since the transfer in (e) is limited to a single scale, it fails to capture textures of multiple scales on human faces, resulting in unnatural looks on facial landmarks like the eyebrows.*

5.1 Comparisons between the two works

Even though at the root level both methods are data-driven, they show many differences from a technical viewpoint. In time hallucination, the dense correspondence is more challenging than that in portrait stylization. To align two faces, the spatial continuity on the correspondences provides strong cues for the task, allowing our method to achieve good results with off-the-shelf techniques like SIFT flow [78]. In contrast, aligning city skylines may require preserving spatial discontinuity of the correspondences. To capture long-range correspondences, we need to sample matchings in global scope, which largely increases the computation complexity and makes the alignment intractable. We address the challenge with an efficient sampling strategy along with novel regularization terms on a Markov random field.

In time hallucination, we employ a locally affine transfer, since the appearance variation of outdoor scenes is low-dimensional [42]. For portraits, we use a multi-scale transfer to capture facial contrasts at different scales, ranging from large textures likes facial landmarks to minute details like pores. The locally affine transfer takes input as a pair of before-and-after images, while our multi-scale transfer only requires a single example image. Figure 5-1 compares the locally affine transfer intended for time hallucination against the multi-scale transfer on an input portrait. In this task, the multi-scale transfer performs better since it processes textures separately for different scales. In contrast, the locally affine transfer uses a single affine model for different scales, and results in artifacts on the output.

The locally affine transfer is more robust to the choice of color space, since affine mappings model any linear color transformations of the standard RGB space. In portrait stylization, we consider human perceptions by using a Lab color space, which is more intricate and requires non-linear mappings from the RGB space. Both works extend the style transfer algorithms to videos. While the resulting portrait videos can handle facial expressions in inputs, the synthetic time-lapse videos, however, are limited to static objects, because object motions and occlusions in input videos are still challenging to us.

5.2 Future work

Our portrait transfer work has attained some public media coverage[2][3][4][5], as has our time hallucination work[6][7]. Laffont *et al.* [69] have applied our work to hallucinate a photograph taken in springtime to winter renditions, using time-lapse data over the course of a year. From where this dissertation ends, there are a few possible starts. Our method can benefit from improved techniques on semantic correspondences, which is still a fundamental problem in computer vision. Our method is robust to correspondences between inputs and examples, but excessive errors on the correspondences sometimes result in halo artifacts. The correspondence technique in our time hallucination work could be applied to semantic segmentation on time-lapse videos.

We could make our databases more compact. In the time-lapse database, we could extract parametric models to describe scene appearance variations for different object categories, like trees, sky, roads, and buildings. For portraits, we could pre-process example image collections by warping their energy maps to a standard face template, and perform the style transfer from the template without the database.

Our work has potential in cloud processing, since users can share the databases at remote servers. Further, in our works, the transformation representations – the gain maps in portraits or the affine models in time-of-day work – are easier than the output images, since they are low-passed or low-dimensional. Cloud processing could save bandwidth by transferring these simpler representations instead of outputs at full-resolution.

Our results have achieved good "visual realism," which means they look plausible although not guaranteed to be physically correct. With image databases, it is an open question whether we

[2] Engadget: http://www.engadget.com/2014/05/31/mit-selfie-portrait-project/
[3] PetaPixel: http://petapixel.com/2014/06/01/researchers-turn-average-smartphone-portraits-stylized-pieces-art/
[4] MIT News: http://newsoffice.mit.edu/2014/spruce-your-selfie
[5] TechCrunch: http://techcrunch.com/2014/05/30/mit-researchers-create-an-app-that-turns-selfies-into-works-of-art/
[6] Adobe System: http://blogs.adobe.com/conversations/2015/01/light-and-magic.html
[7] PetaPixel: http://petapixel.com/2014/10/10/adobe-shows-features-changing-time-day-lighting-removing-fog/

can extract physical information for other applications, such as accurately predicting the aging processes of a person by a headshot database of herself or other people. Some recent works have started to study on this exciting direction [64].

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

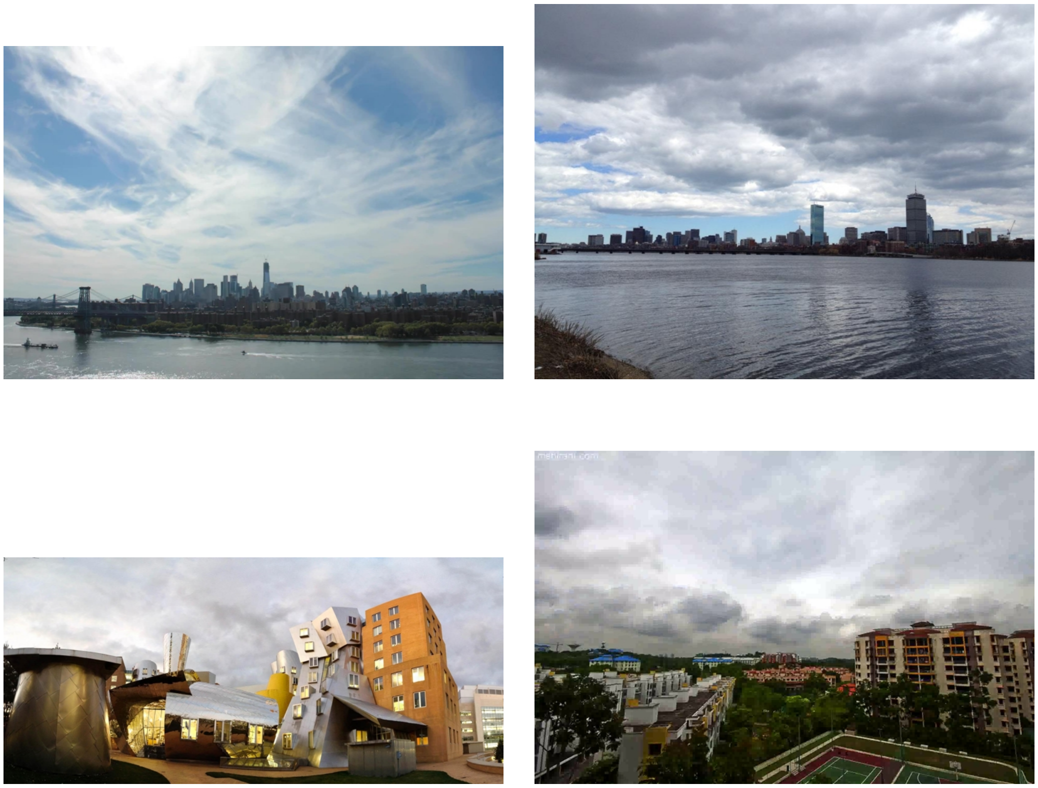
Additional Results for Time Hallucination

We describe additional results for time hallucination work in Chapter 3. The corresponding sections are labeled in the following texts.

Time-lapse video retrieval and the match frame Figure A-1 illustrates the time-lapse video retrieval results and the match frames. The retrieval is based on a standard scene matching technique [133] (Section 3.5.1) and color statistics (Section 3.5.1).

Locally linear vs affine Figure A-2 compares the choices of locally affine model and linear model. Similar to expressivity test, we hallucinate from one input frame to another ground truth frame in a single time-lapse video. We perform the transfer with locally linear and affine model. The difference between the output and the ground truth shows that affine model yields better results.

Expressivity of locally affine transfer Figure A-3 illustrates the expressivity of locally affine model under various scenes (Section 3.6), including harbor, lake, skyline, river side. As described in Section 3.6, we take a frame from a time-lapse video as the input, and another frame as the ground

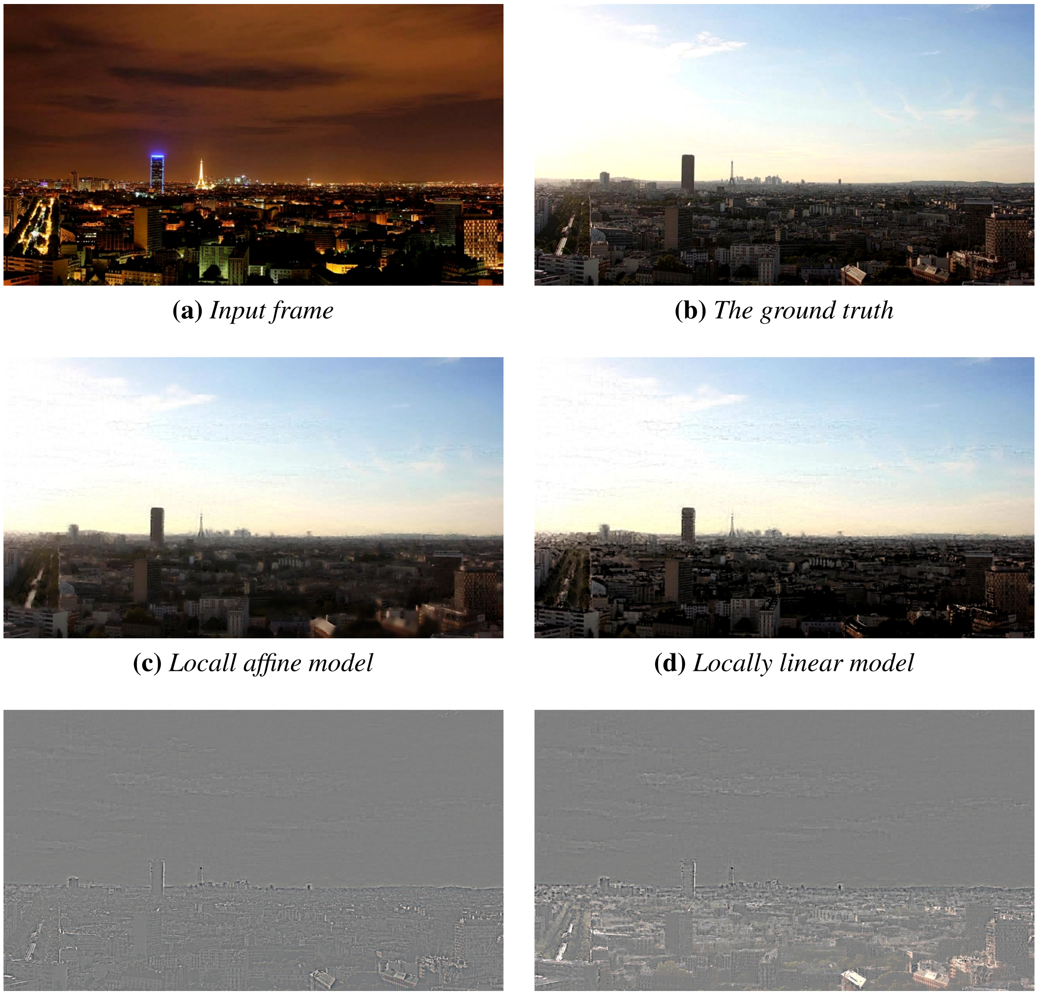


Input image *Retrieved video*

Figure A-1: *Video retrieval and match frame results under two different input scenes.*

truth. We hallucinate the input to the ground truth frame using the same time-lapse video. The output is visually close to the ground truth, even the lighting between the ground truth and the input frame is very different.

Compare to Deep photo In Figure A-4, we compare our results to Deep Photo [66], which uses scene 3D information to relight the image (Section 3.7.1). We use the input and the result relit at dusk on their web-site. For comparison, we hallucinate the input to "golden hour". Both results are plausible, but we do not rely on scene-specific data.


(a) *Input frame*

(b) *The ground truth*

(c) *Locall affine model*

(d) *Locally linear model*

(e) *Difference map between (c) and (b)*

(f) *Difference map between (f) and (b)*

Figure A-2: *We show locally affine model is a better choice than linear model. We hallucinate the input to another frame (ground truth) in the same time-lapse video with two different models. The affine model is closer to the ground truth.*

Compare to Laffont *et al.* In Figure A-5, we compare our results to Laffont *et al.* [68], which uses a collection of photos under the same scene for illumination transfer (Section 3.7.1). They decompose the image into an intrinsic image and an illumination layer, and then transfer the illumination from one image to another image. In this experiment, they used 17 images for decomposition, and transfer the illumination from a photo under faint light. For comparison, we

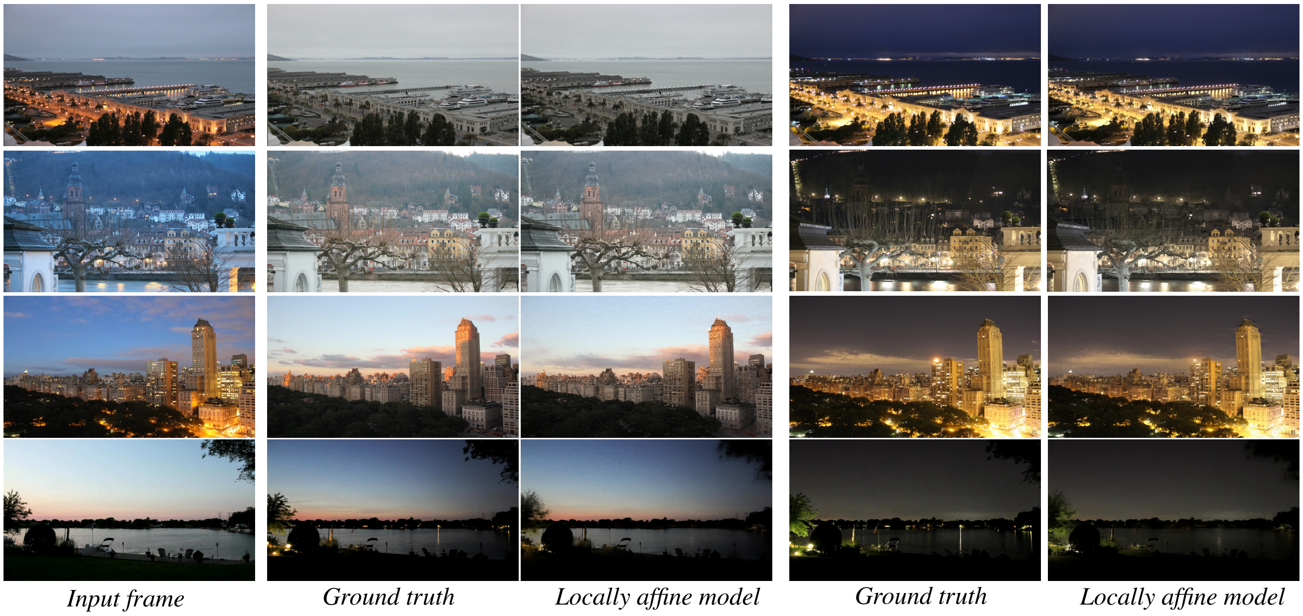


Figure A-3: *Locally affine model is expressive enough to model different times of a day. For each row, we pick up a frame from a time-lapse video as input. We choose another ground truth frame from the same time-lapse video as input, and produce the result using our model. Our result is very close to the ground truth and shows our model is expressive for time hallucinations even lighting between input and ground truth is very different.*

hallucinate the input to “blue hour”. Again, both are plausible, but we only require a single input photo.

A.1 Accompanying video

We show synthetic time-lapse videos (Section 3.7.2) on our project website:

`http://people.csail.mit.edu/yichangshih/time_lapse/`

We generate results at different times from a single input, and then linearly interpolate these results to simulate a time-lapse video.

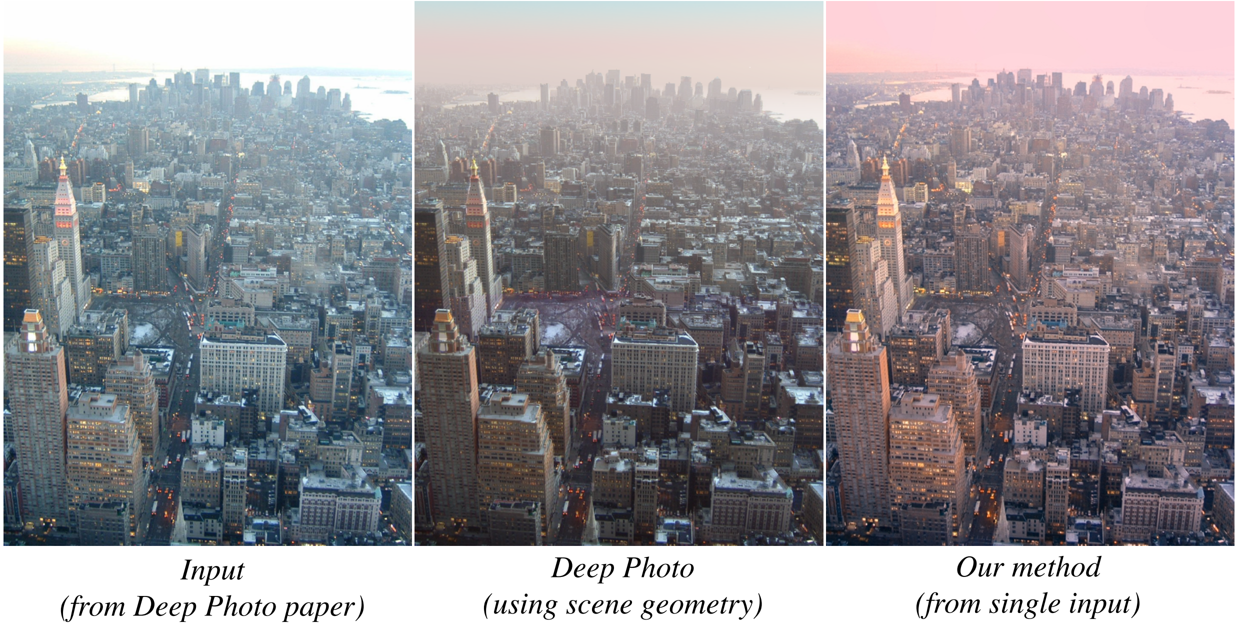
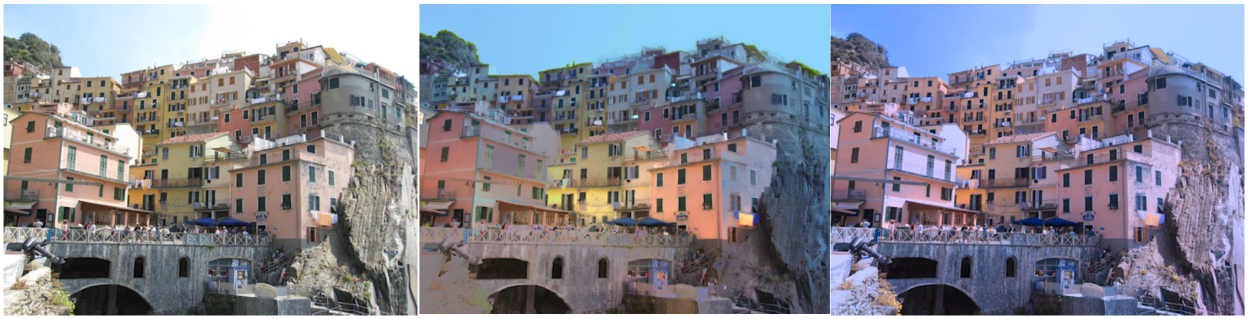


Figure A-4: *Deep photo leverages depth map and texture of the scene to relight an image. Our method uses less information and produces plausible looking results. We hallucinate the input to "golden hour" to match their result. We use results directly from Deep Photo project website.*

A.2 Accompanying web page

We show our evaluation on MIT-Adobe fiveK dataset [16]. (Section 3.7) at the following webpage:

`http://people.csail.mit.edu/yichangshih/time_lapse/webpage/`



Input image
(from Laffont et al.'s paper)

Laffont et al.
(using 17 images)

Our method
(from a single input image)

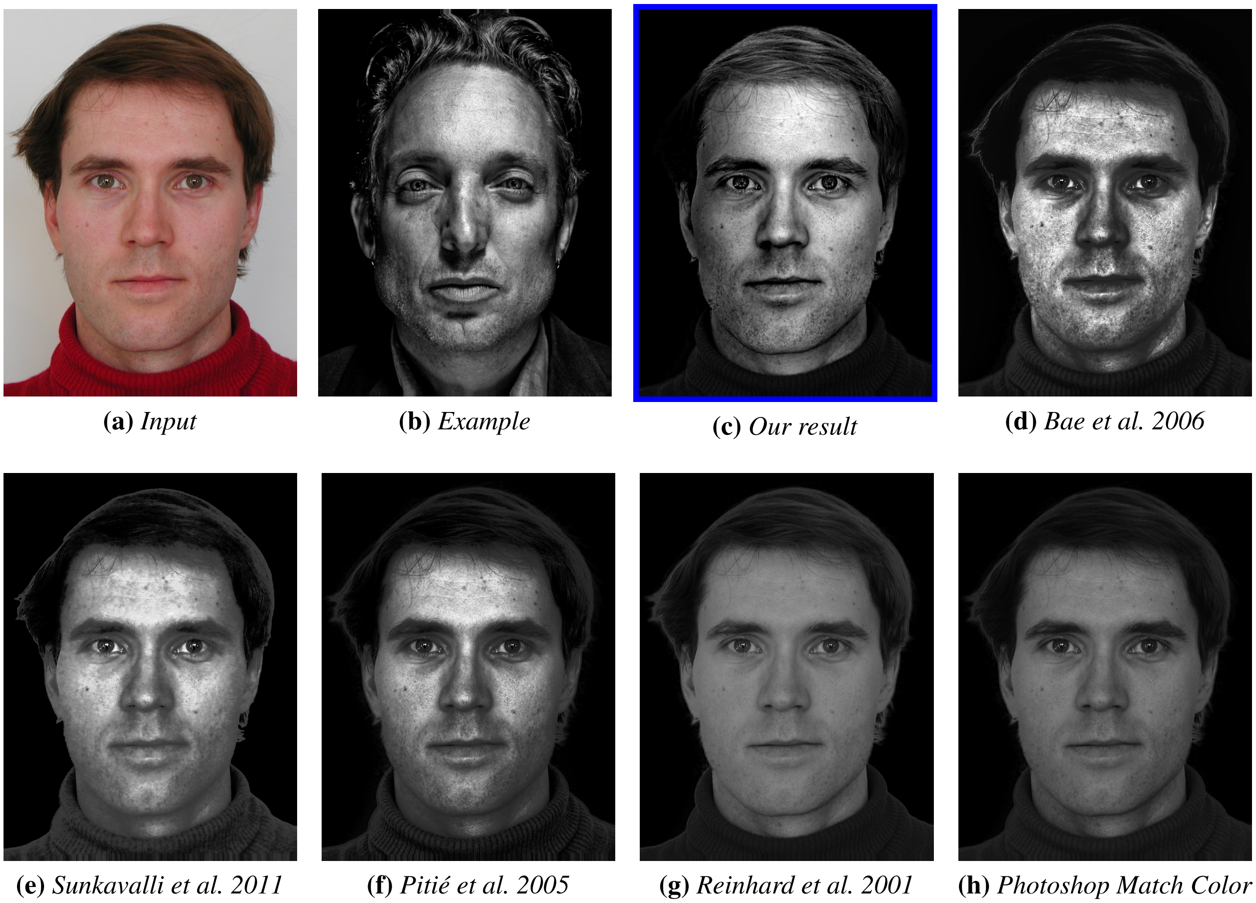
Figure A-5: *Laffont et al. use multiple images at the same scene for intrinsic image decomposition, and then relight the image by transferring illumination to the intrinsic image. We use different data for relighting. We hallucinate the input to "blue hour" to match their result. Laffont's result is directly from their website.*

Appendix B

Additional Results for Portrait Style Transfer

Here we describe additional results for our work on portrait style transfer. In the title of each paragraph, we put the section number referenced in the original chapter (Chapter 4)

Additional comparisons to related work (Section 4.4.1) Figure B-1 shows the comparisons on an extreme and low-key style. Without adaptation to the face mask, all the global methods fail in this case, since the background in the input is brighter than the foreground, but vice versa in the example. For fair comparison, we adapted the related methods to face mask. We replaced the input with the example background for Bae *et al.* [6] and PhotoShop MatchColor. We limit the transfer in the face region defined by the mask for Sunkavalli *et al.* [118], Pitié *et al.* [97], and Reinhard *et al.* [100]. For Sunkavalli *et al.* , we started by their setup demonstrated on face portraits, and tested a few options. We found that disabling noise matching produces the best result. For Pitíe *et al.* [97], we ran 30 iterations. We also tried HaCohen *et al.* [45], but their implementation reports that no matching is found.



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

(e) *Sunkavalli et al. 2011* **(f)** *Pitié et al. 2005* **(g)** *Reinhard et al. 2001* **(h)** *Photoshop Match Color*

Figure B-1: *We show comparisons on an extreme style, using related methods adapted to face mask. We replaced the input with the example background for Bae et al. and PhotoShop MatchColor. We limit the transfer in the face region defined by the mask for Sunkavalli et al. Pitié et al. , and Reinhard et al.. Our method captures the smoothly fall-off lighting on the forehead and details on the face.*

Figure B-2 shows the comparison on a nearly all-black-and-white style. Our method transfers the right amount of details and brightness without being over-exposed or under-exposed. We used the same adaptation for the related methods. We also tried HaCohen *et al.* [45], but their implementation again reports that no matching is found in this case.

Figure B-3 shows the comparison on a color style to two methods adapted to face mask, as described above. The comparisons with the unadapted methods are described in Chapter 4.

Figure B-4 shows a close-up comparison between our result and the example. Our result matches well on lighting, color, facial details in all scales.

Figure B-5 compares to HaCohen *et al.* [45] on an example that their method finds non-empty matchings. The inset in Fig. B-5(d) shows the matching area. Among all the examples used in our project, this example has the largest matching region.

Comparison to reference image (Section 4.4) Figure B-6 shows comparison to a reference image. We use the reference as example. This is to show the ideal situation that the database is sufficiently large such that we can find an example almost identical to the input.

Comparison on makeup transfer (Section 4.4.2) Figure B-7 shows the comparison on our extension to makeup transfer. Fig. B-7(c) shows our original method before modification. The green eye shadow is bled to the sclera (the white of the eye). Our adapted method automatically transfers the sclera from the input to fix the problem, as shown in Fig. B-7(d). The rest of Fig. B-7 compares our result with two state-of-the-art methods designed for makeup transfer [44, 122]. All three methods achieve plausible results. Tong *et al.* require the before image of the example makeup, which is not shown here. Their results are directly taken from Guo and Sim's work.

Additional results on automatic selection algorithm (Section 4.4) Figure B-8 shows the style transfer results using the top four examples selected by our automatic algorithm. We show three styles.

User correction (Section 4.4) Our dense matching using computer vision techniques often produces satisfactory results. However, there are cases where matching is challenging, such as matching long hair to short hair in Figure B-9. In this case, we provide users a manual correction work flow by using an user-created constraint map. Then our algorithm re-run the transfer, but this time we assign the energy gains of each pixel in the red region by the average of the gains in the green region. This process can be repeated as needed for additional corrections. To avoid discontinuities, we filter the gain map with a small Gaussian kernel after applying the constraint map. Figure B-9d shows the successful result after user correction. In our results on Flickr data set, 5 out of 94 are corrected in such a way. All the results in Chapter 4 are generated automatically; we did not correct them.

The artifacts due to transferring the example identity Figure B-10 shows a failure case that the identity of the example is transferred to the input. This may occur when the example identity has different genders or very different skin colors.

Massive results using Flickr data set (Section 4.4) We use inputs downloaded from an online web site, Flickr, on three different styles, and show the results at our project web page in the following:

`http://people.csail.mit.edu/yichangshih/portrait_web/#results`

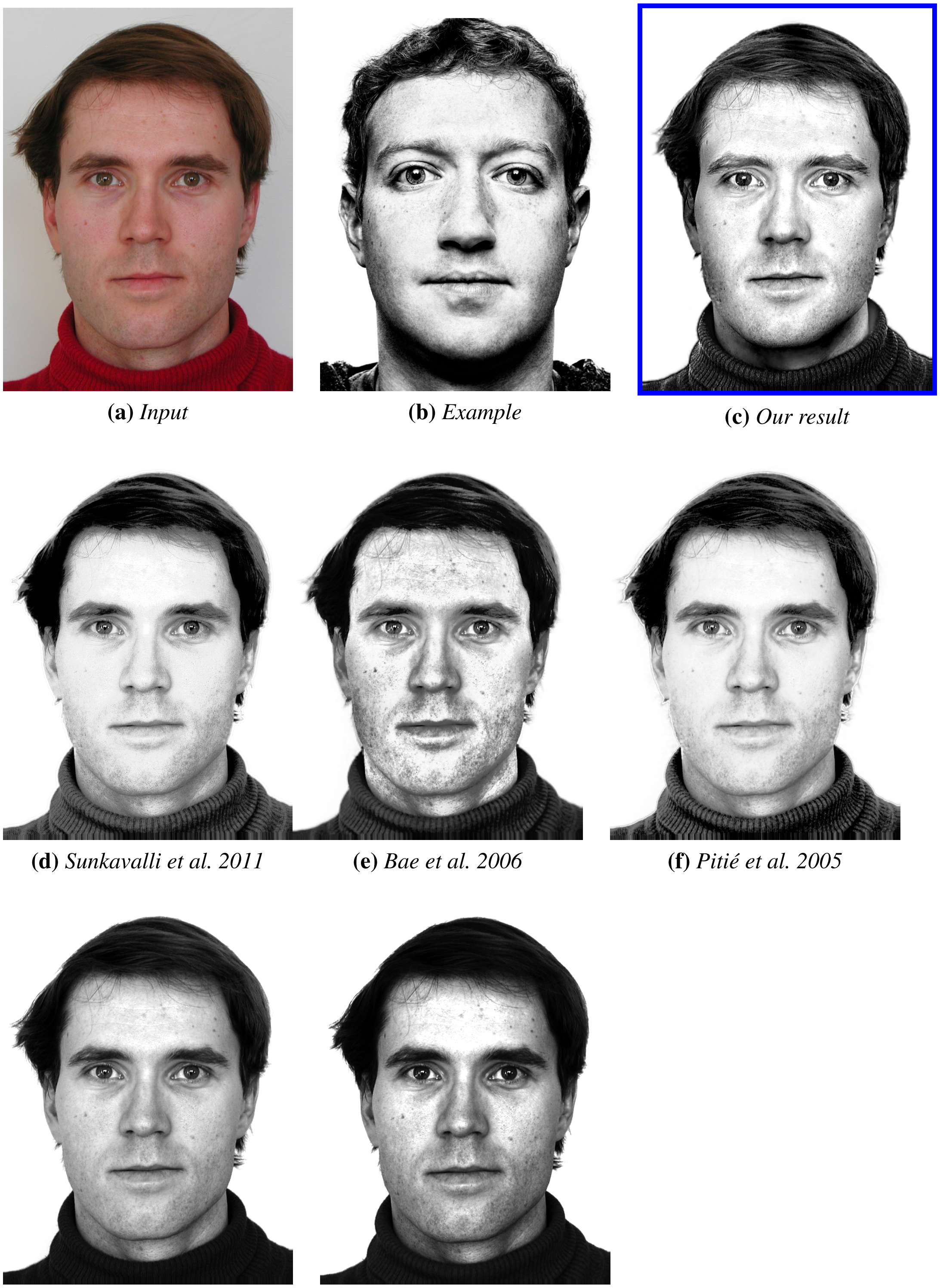
The data collection workflow is described in Chapter 4. The data set contains 94 images with various facial contents, expressions, under arbitrary lighting conditions. All inputs are under creative commons license.

B.1 Accompanying Video (Section 4.4.2)

We show our video style transfer extension at our project webpage:

`http://people.csail.mit.edu/yichangshih/portrait_web/#video`

We test two different inputs with moderate motion and extreme facial expressions, using three different styles. No audio in the video.



(a) *Input*

(b) *Example*

(c) *Our result*

(d) *Sunkavalli et al. 2011*

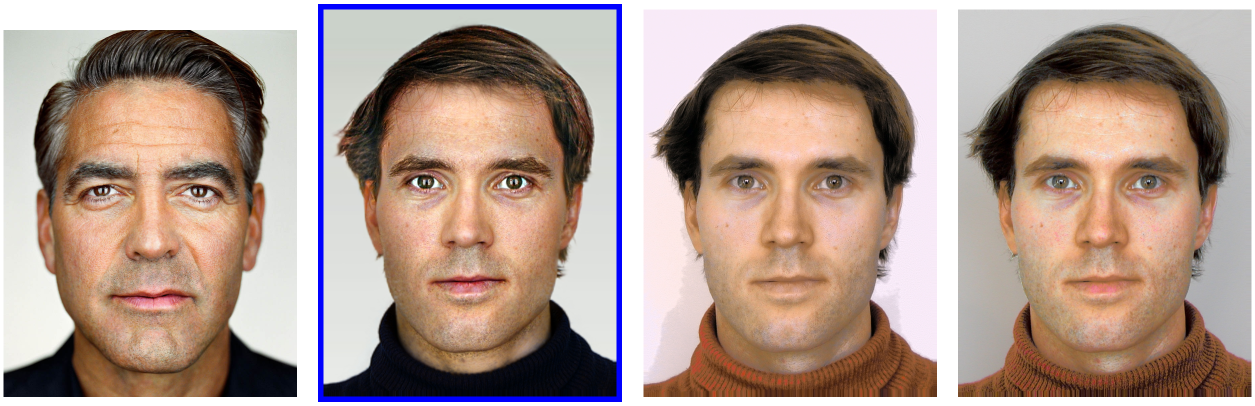
(e) *Bae et al. 2006*

(f) *Pitié et al. 2005*

(g) *Reinhard et al. 2001*

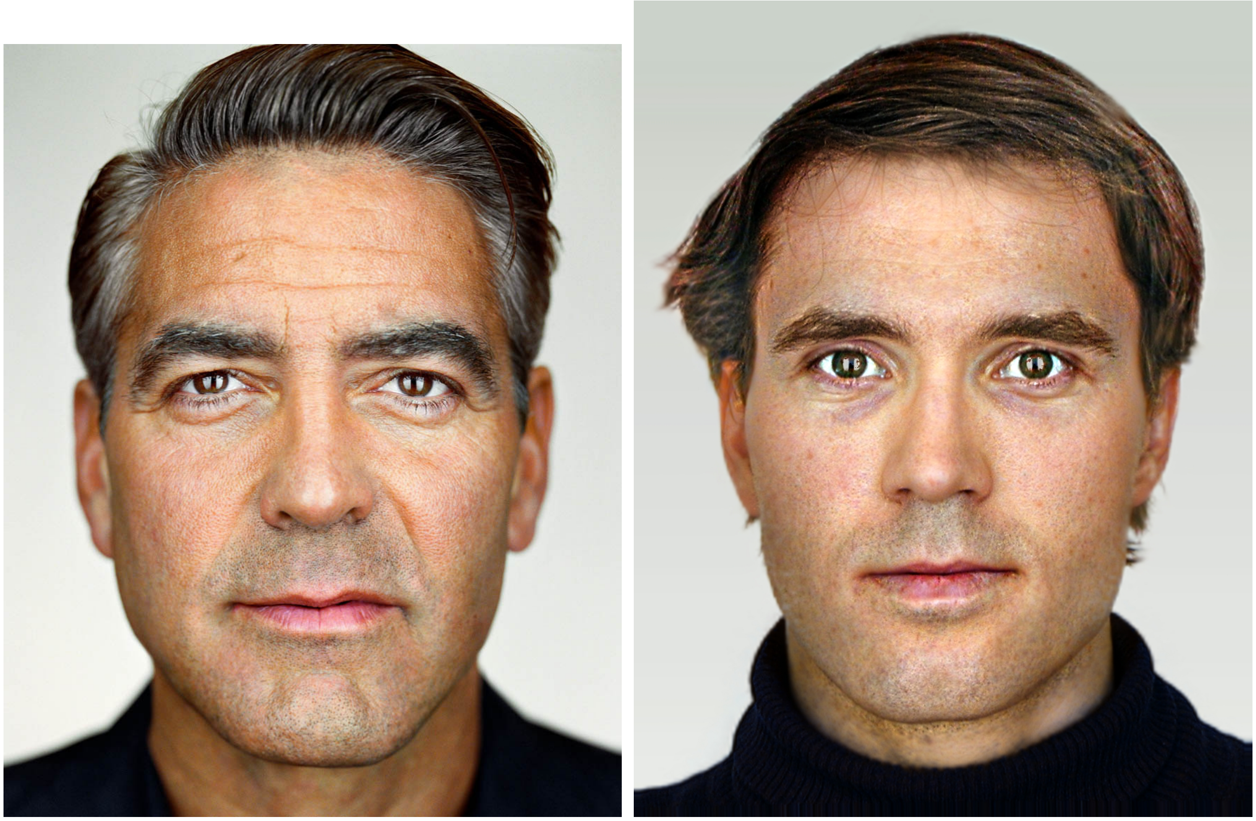
(h) *PhotoShop Match Color*

Figure B-2: *We show a comparison on a nearly all-black-and-white style. Our method captures the right amount of exposure and details on the face and hair.*



(a) *Example* **(b)** *Our result* **(c)** *Sunkavalli et al. 2011* **(d)** *Pitíe et al. 2005*

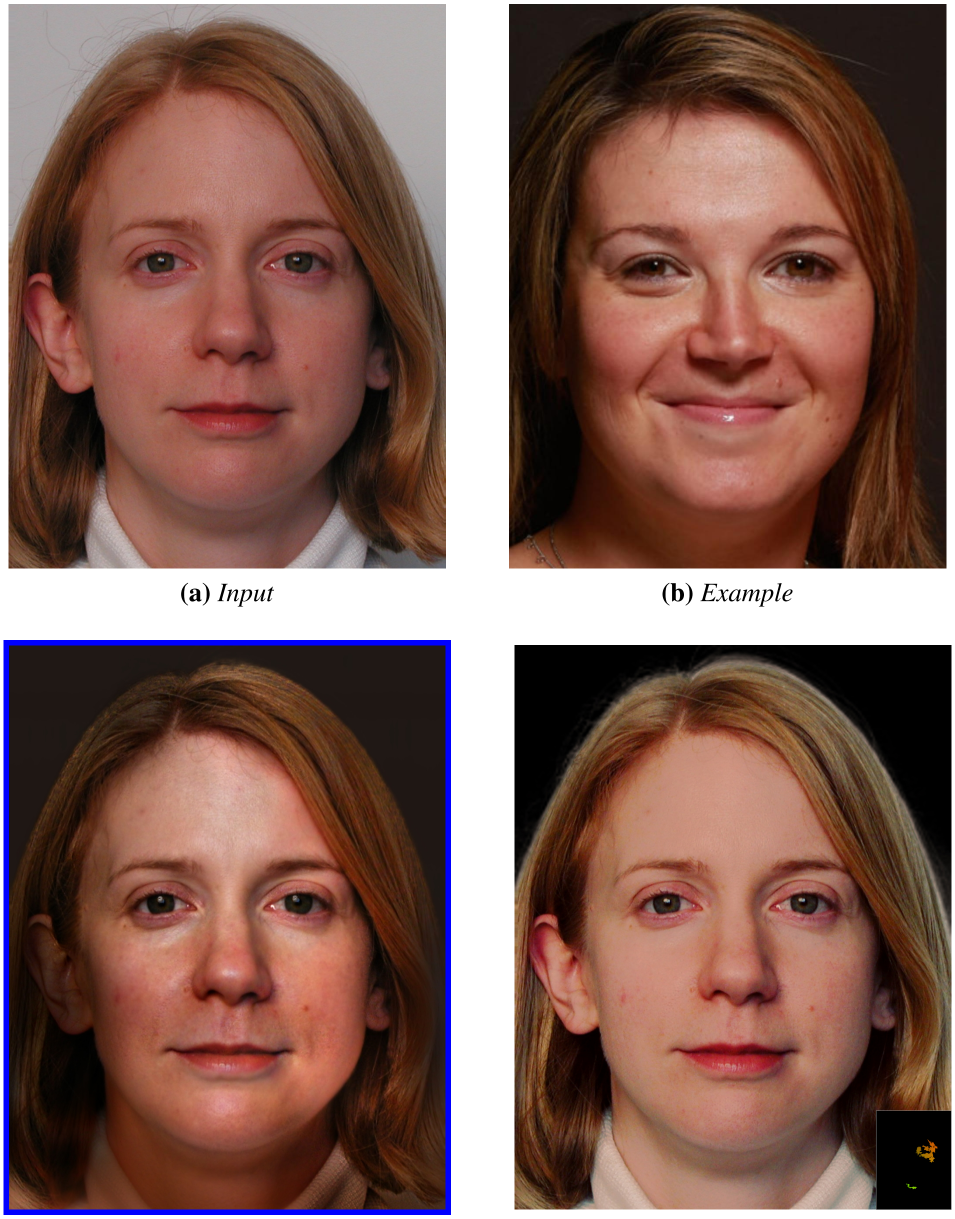
Figure B-3: *Using the input in Fig. B-2, we compare two methods adapted to face mask on a color style. The comparison to the unadapted methods are described in Chapter 4 .*



(a) *Example*

(b) *Our result*

Figure B-4: *Close-up comparison to the example, using the input in Fig. B-2. Our result matches well on lighting, color, facial details in all scales.*



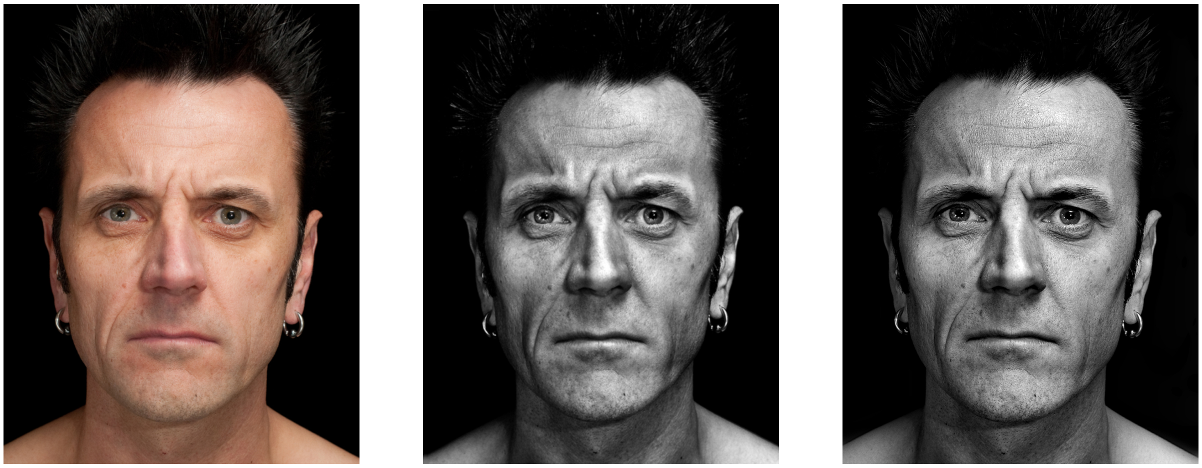
(a) *Input*

(b) *Example*

(c) *Our result*

(d) *HaCohen et al. 2011*

Figure B-5: *We compare to HaCohen et al. on a case that their method finds matching region, shown in the inset in (d). This example is has the largest matching region among all examples used in this work.*

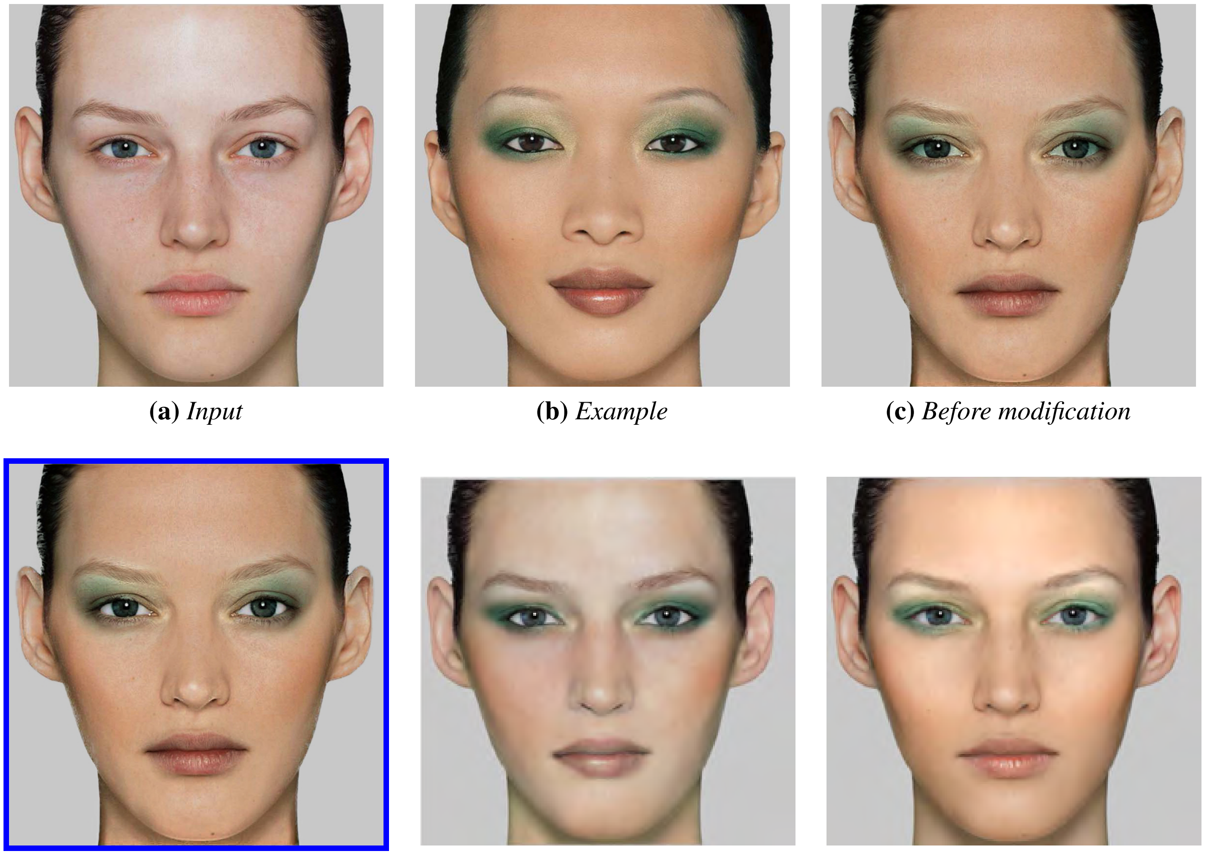


(a) *Input (before editing)*

(b) *Example (after editing), reference image.*

(c) *Our result*

Figure B-6: *We test the "upper bound" of our method by using a pair of before/after editing images in (a) and (b) as input and example. Our result (c) is visually close to (b). This is to simulate the ideal situation that we can find an example subject whose look is very close to the input.*



(a) *Input*

(b) *Example*

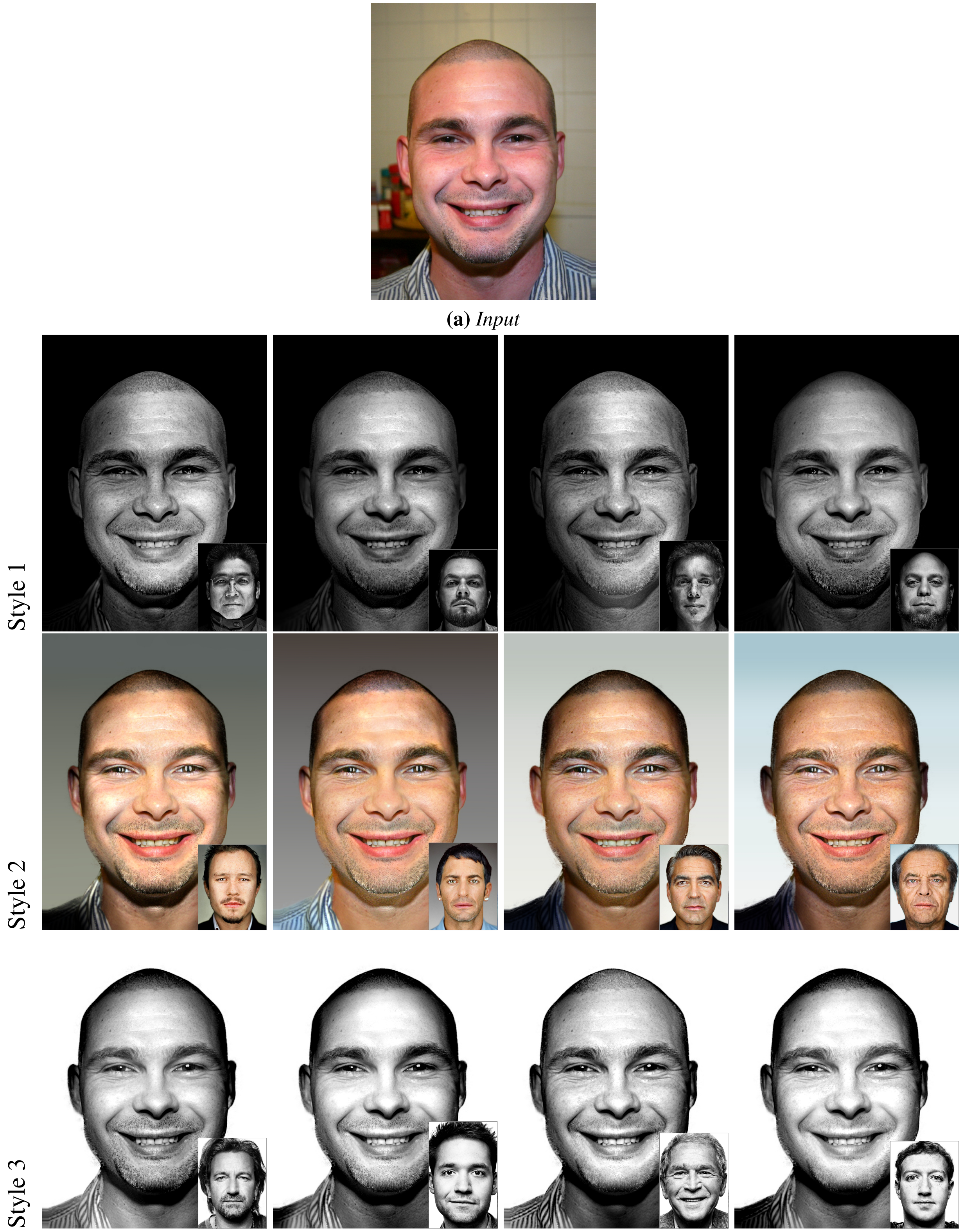
(c) *Before modification*

(d) *Our final result*

(e) *Tong et al. 2007, require before image of (b)*

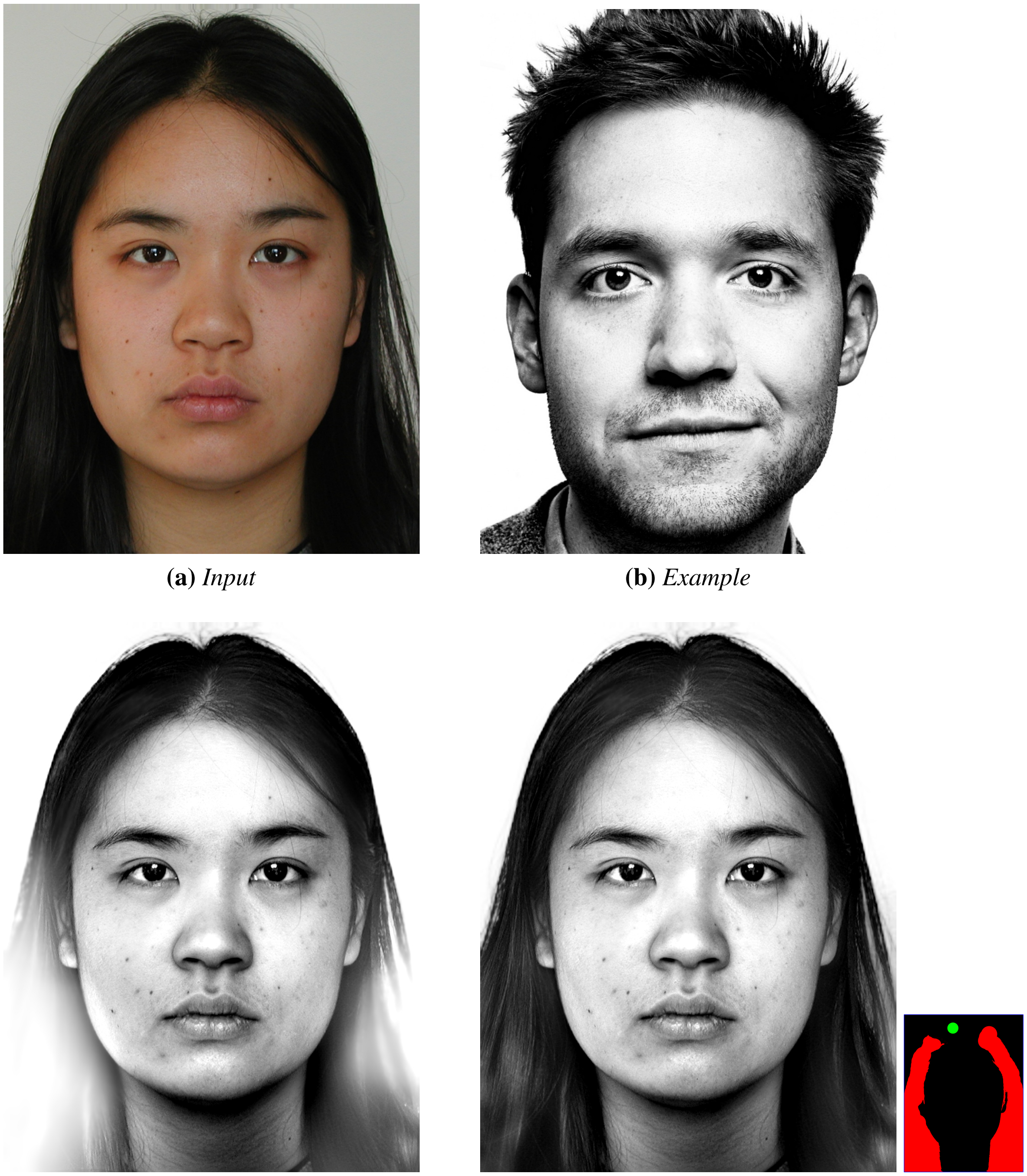
(f) *Guo et al. 2009*

Figure B-7: *We extend our method to makeup transfer. Directly using our algorithm results color bleeding on eyes (c). With minor modification that handles eye sclera (eye white), we can achieve better result (d). We show comparison with two state-of-art methods designed for makeup transfer. (e) requires before image of (b), which is not shown here. (f) explicitly models foundation, eye shadow and lip color. All results achieve plausible makeup transfer. (e) and (f) are directly taken from their papers, respectively.*



(a) *Input*

Figure B-8: *We show style transfer results on the input in (a), using different styles in the three rows. We use the top four examples selected by our automatic selection algorithm, shown in the insets.*



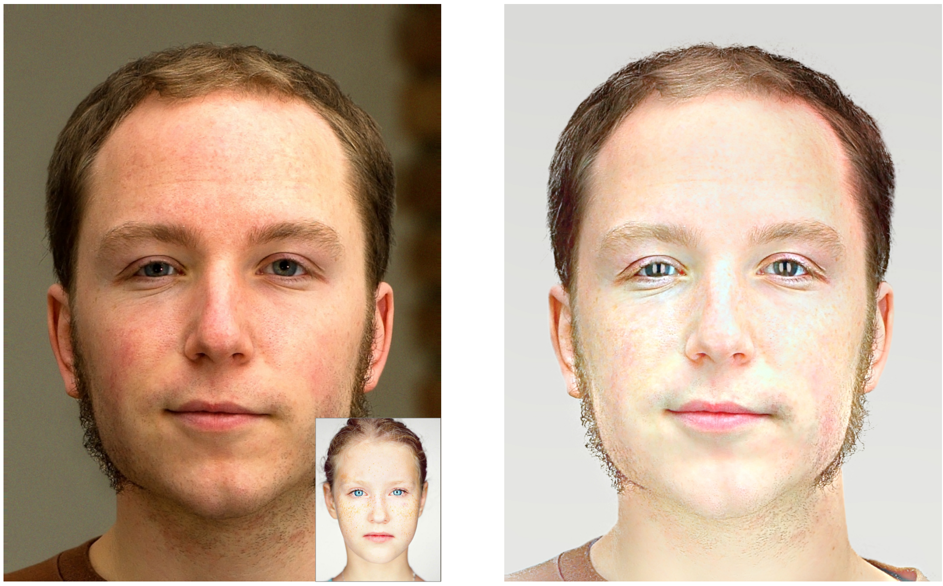
(a) *Input*

(b) *Example*

(c) *Result failed on hair*

(d) *Corrected result using a user-provided constraint mask in the blue box*

Figure B-9: *Our transfer can fail if the input (a) and example (b) have very different hair styles, and cause artifacts on the hair in (c). We demonstrate that the user can fix this in (d) by providing a constraint map in the blue box. This map constrains that the gains of the red region to be the same as those of the green region*



(a) *Input*

(b) *Failed output*

Figure B-10: *A failure case that the identity of the example (inset in (a)) is transferred to the output.*

Bibliography

[1] Arash Abadpour and Shohreh Kasaei. An efficient pca-based color transfer method. *Journal of Visual Communication and Image Representation*, 18(1):15–34, 2007.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2274–2282, 2012.

[3] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. 2006.

[4] Xiaobo An and Fabio Pellacini. User-controllable color transfer. In *Computer Graphics Forum*, volume 29, pages 263–271, 2010.

[5] Mathieu Aubry, Sylvain Paris, Samuel W Hasinoff, Jan Kautz, and Frédo Durand. Fast local laplacian filters: Theory and applications. *ACM Transactions on Graphics (TOG)*, 33(5):167, 2014.

[6] Soonmin Bae, Sylvain Paris, and Frédo Durand. Two-scale tone management for photographic look. 2006.

[7] C. Barnes, E. Shechtman, D. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. *European Conference on Computer Vision*, pages 29–43, 2010.

[8] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. 2009.

[9] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. volume 26, 1992.

[10] Floraine Berthouzoz, Wilmot Li, Mira Dontcheva, and Maneesh Agrawala. A framework for content-adaptive photo manipulation macros: Application to face, landscape, and global manipulations. *ACM Transactions on Graphics (TOG)*, 30(5):120, 2011.

[11] Nicolas Bonneel, Kalyan Sunkavalli, Sylvain Paris, and Hanspeter Pfister. Example-based video color grading. 2013.

[12] A. Bousseau, S. Paris, and F. Durand. User-assisted intrinsic images. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 28, page 130, 2009.

[13] M. Brand and P. Pletscher. A conditional random field for automatic photo editing. 2008.

[14] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. pages 25–36. 2004.

[15] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[16] V Bychkovsky, S Paris, E Chan, and F Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 97–104, 2011.

[17] Robert Caputo. In *Potography field guide*, pages 104–115. National Geographics, 2005.

[18] Youngha Chang, Suguru Saito, Keiji Uchikawa, and Masayuki Nakajima. Example-based color stylization of images. *ACM Transactions on Applied Perception*, 2(3):322–345, 2006.

[19] Youngha Chang, Suguru Saito, and Masayuki Nakajima. Example-based color transformation of image and video using basic color categories. *IEEE Transactions on Image Processing*, 16(2):329–336, 2007.

[20] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. In *European Conference on Computer Vision*, pages 126–139. Springer, 2008.

[21] Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin. Semantic colorization with internet images. *ACM Transactions on Graphics (TOG)*, 30(6):156, 2011.

[22] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. *ACM Transactions on Graphics (TOG)*, 25(3):624–630, 2006.

[23] Claudio Cusano, Francesca Gasparini, and Raimondo Schettini. Color transfer using semantic image annotation. In *IS&T/SPIE Electronic Imaging*, pages 82990U–82990U. International Society for Optics and Photonics, 2012.

[24] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[25] Kevin Dale, Micah K Johnson, Kalyan Sunkavalli, Wojciech Matusik, and Hanspeter Pfister. Image restoration using online photo collections. In *IEEE International Conference on Computer Vision*, pages 2217–2224, 2009.

[26] John G Daugman. High confidence visual recognition of persons by a test of statistical independence. 15(11):1148–1161, 1993.

[27] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, page 31. ACM, 2008.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[29] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imagenet. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1777–1784, 2011.

[30] Weiming Dong, Guanbo Bao, Xiaopeng Zhang, and Jean-Claude Paul. Fast local color transfer via dominant colors mapping. In *ACM SIGGRAPH ASIA 2010 Sketches*, page 46. ACM, 2010.

[31] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 257–266. ACM, 2002.

[32] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Annual Conference on Computer Graphics and Interactive Techniques*, pages 341–346, 2001.

[33] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 23, pages 673–678, 2004.

[34] Zeev Farbman and Dani Lischinski. Tonal stabilization of video. In *ACM Transactions on Graphics (TOG)*, volume 30, page 89. ACM, 2011.

[35] Hasan Sheikh Faridul, Jurgen Stauder, Jonathan Kervec, and Alain Trémeau. Approximate cross channel color mapping from sparse color correspondences. In *IEEE International Conference on Computer Vision Workshops (ICCVW),*, pages 860–867, 2013.

[36] Hasan Sheikh Faridul, Tania Pouli, Christel Chamaret, Jürgen Stauder, Alain Trémeau, Erik Reinhard, et al. A survey of color mapping and its applications. In *Eurographics 2014-State of the Art Reports*, pages 43–67, 2014.

[37] Wenya Feng, Yilin Guo, Okhee Kim, Yonggan Hou, Long Liu, and Huiping Sun. Color transfer based on earth mover's distance and color categorization. In *Computer Analysis of Images and Patterns*, pages 394–401. Springer, 2013.

[38] Juliet Fiss, Aseem Agarwala, and Brian Curless. Candid portrait selection from video. In *ACM Transactions on Graphics (TOG)*, volume 30, page 128. ACM, 2011.

[39] Daniel Freedman and Pavel Kisilev. Object-to-object color transfer: optimal flows and smsp transformations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 287–294, 2010.

[40] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.

[41] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[42] Rahul Garg, Hao Du, Steven M Seitz, and Noah Snavely. The dimensionality of scene appearance. In *IEEE International Conference on Computer Vision*, pages 1917–1924, 2009.

[43] Mark Grundland and Neil A Dodgson. Color histogram specification by histogram warping. In *Electronic Imaging 2005*, pages 610–621. International Society for Optics and Photonics, 2005.

[44] Dong Guo and Terence Sim. Digital face makeup by example. 2009.

[45] Y. HaCohen, E. Shechtman, D.B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (SIGGRAPH)*, 30(4):70, 2011.

[46] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Optimizing color consistency in photo collections. *ACM Transactions on Graphics (TOG)*, 32(4):38, 2013.

[47] Yoav Hacohen, Eli Shechtman, and Dani Lischinski. Deblurring by example using dense correspondence. In *IEEE International Conference on Computer Vision*, pages 2384–2391, 2013.

[48] Sheikh Faridul Hasan, Jurgen Stauder, and Alain Tremeau. Optimization of sparse color correspondences for color mapping. In *Color and Imaging Conference*, volume 2012, pages 128–134. Society for Imaging Science and Technology, 2012.

[49] J. Hays and A.A. Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 26, page 4, 2007.

[50] K. He, J. Sun, and X. Tang. Guided image filtering. *European Conference on Computer Vision*, pages 1–14, 2010.

[51] Li He, Hairong Qi, and Russell Zaretzki. Image color transfer to evoke different emotions based on color combinations. *Signal, Image and Video Processing*, pages 1–9, 2014.

[52] David J Heeger and James R Bergen. Pyramid-based texture analysis/synthesis. pages 229–238. ACM, 1995.

[53] A. Hertzmann, C.E. Jacobs, N. Oliver, B. Curless, and D.H. Salesin. Image analogies. In *Conference on Computer Graphics and Interactive Techniques*, pages 327–340, 2001.

[54] Aaron Hertzmann. Non-photorealistic rendering and the science of art. In *International Symposium on Non-Photorealistic Animation and Rendering*, pages 147–157. ACM, 2010.

[55] Tzu-Wei Huang and Hwann-Tzong Chen. Landmark-based sparse color representations for color transfer. In *IEEE International Conference on Computer Vision*, pages 199–204, 2009.

[56] Sung Ju Hwang, Ashish Kapoor, and Sing Bing Kang. Context-based automatic local image enhancement. In *European Conference on Computer Vision*, pages 569–582. Springer, 2012.

[57] Youngbae Hwang, Joon-Young Lee, In So Kweon, and Seon Joo Kim. Color transfer using probabilistic moving least squares. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3342–3349, 2014.

[58] Revital Irony, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. In *Eurographics Conference on Rendering Techniques*, pages 201–210, 2005.

[59] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[60] H.W. Jensen, S. Premoze, P. Shirley, W.B. Thompson, J.A. Ferwerda, and M.M. Stark. Night rendering. *IN University of Utah Technical Report (UUCS-00-016)*, 2000.

[61] Neel Joshi, Wojciech Matusik, Edward H Adelson, and David J Kriegman. Personal photo enhancement using example images. *ACM Transaction on Graphics (TOG)*, 29(2):1–15, 2010.

[62] Alexei A. Efros Eli Shechtman Jue Wang Jun-Yan Zhu, Aseem Agarwala. Mirror mirror: Crowdsourcing better portraits. *ACM Transactions on Graphics (TOG)*, 2014.

[63] Sefy Kagarlitsky, Yael Moses, and Yacov Hel-Or. Piecewise-consistent color mappings of images acquired under various conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2311–2318, 2009.

[64] Ira Kemelmacher-Shlizerman, Supasorn Suwajanakorn, and Steven M Seitz. Illumination-aware age progression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3334–3341, 2014.

[65] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2307–2314, 2013.

[66] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski. Deep photo: Model-based photograph enhancement and viewing. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 27, page 116, 2008.

[67] Hiroaki Kotera. A scene-referred color transfer for pleasant imaging on display. In *IEEE International Conference on Image Processing*, volume 2, pages II–5, 2005.

[68] Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frédo Durand, George Drettakis, et al. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics*, 31(6), 2012.

[69] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG)*, 33(4):149, 2014.

[70] J.F. Lalonde, A.A. Efros, and S.G. Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 28, page 131, 2009.

[71] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 23, pages 689–694, 2004.

[72] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 61–68, 2006.

[73] Anat Levin and Boaz Nadler. Natural image denoising: Optimality and inherent bounds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2833–2840, 2011.

[74] Tommer Leyvand, Daniel Cohen-Or, Gideon Dror, and Dani Lischinski. Data-driven enhancement of facial attractiveness. In *ACM Transactions on Graphics (TOG)*, volume 27, page 38. ACM, 2008.

[75] Yuanzhen Li, Lavanya Sharan, and Edward H Adelson. Compressing and companding high dynamic range images with subband architectures. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 836–844, 2005.

[76] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman. Sift flow: Dense correspondence across different scenes. *European Conference on Computer Vision*, pages 28–42, 2008.

[77] Ce Liu, Heung-Yeung Shum, and William T Freeman. Face hallucination: Theory and practice. *International Journal of Computer Vision*, 75(1):115–134, 2007.

[78] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5), 2011.

[79] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. Intrinsic colorization. In *ACM Transactions on Graphics*, volume 27, page 152, 2008.

[80] Yiming Liu, Michael Cohen, Matt Uyttendaele, and Szymon Rusinkiewicz. Autostyle: Automatic style transfer from image collections to users' images. In *Computer Graphics Forum*, volume 33, pages 21–31. Wiley Online Library, 2014.

[81] Qing Luan, Fang Wen, and Ying-Qing Xu. Color transfer brush. In *Pacific Conference on Computer Graphics and Applications*, pages 465–468, 2007.

[82] Jitendra Malik and Pietro Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7, 1990.

[83] Umar Mohammed, Simon JD Prince, and Jan Kautz. Visio-lization: generating novel facial images. In *ACM Transactions on Graphics (TOG)*, volume 28, page 57, 2009.

[84] Ján Morovic and Pei-Li Sun. Accurate 3d image colour histogram transformation. *Pattern Recognition Letters*, 24(11):1725–1735, 2003.

[85] Naila Murray, Sandra Skaff, Luca Marchesotti, and Florent Perronnin. Towards automatic concept transfer. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, pages 167–176. ACM, 2011.

[86] Francois Nars. Makeup your mind. *PowerHouse Books*, 2004.

[87] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005.

[88] Aude Oliva, Antonio Torralba, and Philippe G Schyns. Hybrid images. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 527–532, 2006.

[89] Miguel Oliveira, Angel Domingo Sappa, and Vitor Santos. Unsupervised local color correction for coarsely registered images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 201–208. IEEE, 2011.

[90] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. Local laplacian filters: edge-aware image processing with a laplacian pyramid. *ACM Transactions on Graphics (TOG)*, 30(4):68, 2011.

[91] Pieter Peers, Naoki Tamura, Wojciech Matusik, and Paul Debevec. Post-production facial performance relighting using reflectance transfer. *ACM Transactions on Graphics (TOG)*, 26 (3):52, 2007.

[92] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 23, pages 664–672, 2004.

[93] Norman Phillips. Lighting techniques for low key portrait photography. *Amherst Media*, pages 12–16, 2004.

[94] Eric Pichon, Marc Niethammer, and Guillermo Sapiro. Color histogram equalization through mesh deformation. In *IEEE Conference on Image Processing*, volume 2, pages II–117, 2003.

[95] Lyndsey C Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Schölkopf, and William T Freeman. Seeing the arrow of time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[96] F Pitié and A Kokaram. The linear monge-kantorovitch linear colour mapping for example-based colour transfer. 2007.

[97] F. Pitié, A.C. Kokaram, and R. Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *IEEE International Conference on Computer Vision*, volume 2, pages 1434–1439, 2005.

[98] François Pitié, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1):123–137, 2007.

[99] T. Pouli and E. Reinhard. Progressive color transfer for images of arbitrary dynamic range. *Computers and Graphics*, 35(1):67–80, 2011.

[100] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

[101] Erik Reinhard and Tania Pouli. Colour spaces for colour transfer. In *Computational Color Imaging*, pages 1–15. Springer, 2011.

[102] Erik Reinhard, Alexei A Efros, Jan Kautz, and H-P Seidel. On visual realism of synthesized imagery.

[103] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314, 2004.

[104] Galen Rowell. In *Mountain Light*. Sierra Club Books, 2012.

[105] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Face alignment through subspace constrained mean-shifts. In *IEEE Conference on Computer Vision*, pages 1034–1041, 2009.

[106] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Image appearance exploration by model-based navigation. In *Computer Graphics Forum*, volume 28, pages 629–638. Wiley Online Library, 2009.

[107] Amnon Shashua and Tammy Riklin-Raviv. The quotient image: Class-based re-rendering and recognition with varying illuminations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):129–139, 2001.

[108] Yi Chang Shih, Abe Davis, Samuel W Hasinoff, Frédo Durand, and William T Freeman. Laser speckle photography for surface tampering detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 33–40, 2012.

[109] YiChang Shih, Brian Guenter, and Neel Joshi. Image enhancement using calibrated lens simulations. In *European Conference on Computer Vision (ECCV)*, pages 42–56. Springer, 2012.

[110] YiChang Shih, Vivek Kwatra, Troy Chinen, Hui Fang, and Sergey Ioffe. Joint noise level estimation from personal photo collections. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2896–2903, 2013.

[111] YiChang Shih, Sylvain Paris, Frédo Durand, and William T. Freeman. Data-driven hallucination for different times of day from a single outdoor photo. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 2013.

[112] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG)*, 33(4):148, 2014.

[113] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM transactions on graphics (TOG)*, 25(3):835–846, 2006.

[114] Sara Su, FrÈdo Durand, and Maneesh Agrawala. De-emphasis of distracting image regions using texture power maps. In *Proc. of ICCV Workshop on Texture Analysis and Synthesis*, 2005.

[115] Zhuo Su, Daiguo Deng, Xue Yang, and Xiaonan Luo. Color transfer based on multiscale gradient-aware decomposition and color distribution mapping. In *ACM International Conference on Multimedia*, pages 753–756, 2012.

[116] Zhuo Su, Kun Zeng, Li Liu, Bo Li, and Xiaonan Luo. Corruptive artifacts suppression for example-based color transfer. *IEEE Transactions on Multimedia*, 16(4):988–999, 2014.

[117] Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Factored time-lapse video. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 26, page 101. ACM, 2007.

[118] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. 29(4):125, 2010.

[119] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[120] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, 2005.

[121] Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000.

[122] Wai-Shun Tong, Chi-Keung Tang, Michael S Brown, and Ying-Qing Xu. Example-based cosmetic transfer. In *IEEE Pacific Graphics*, 2007.

[123] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *European Conference on Computer Vision*, pages 589–600. 2006.

[124] B. Wang, Y. Yu, T.T. Wong, C. Chen, and Y.Q. Xu. Data-driven image color theme enhancement. In *ACM Transactions on Graphics*, volume 29, page 146, 2010.

[125] Baoyuan Wang, Yizhou Yu, and Ying-Qing Xu. Example-based image color and tone style enhancement. *ACM Transactions on Graphics*, 30(4), 2011.

[126] Chung-Ming Wang, Yao-Hsien Huang, and Ming-Long Huang. An effective algorithm for image sequence color transfer. *Mathematical and Computer Modelling*, 44(7):608–627, 2006.

[127] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 277–280. ACM, 2002.

[128] Zhen Wen, Zicheng Liu, and Thomas S Huang. Face relighting with radiance environment maps. volume 2, pages II–158, 2003.

[129] Fuzhang Wu, Weiming Dong, Xing Mei, Xiaopeng Zhang, Xiaohong Jia, and Jean-Claude Paul. Distribution-aware image color transfer. In *SIGGRAPH Asia 2011 Sketches*, page 8. ACM, 2011.

[130] Fuzhang Wu, Weiming Dong, Yan Kong, Xing Mei, Jean-Claude Paul, and Xiaopeng Zhang. Content-based colour transfer. In *Computer Graphics Forum*, volume 32, pages 190–203. Wiley Online Library, 2013.

[131] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John V Guttag, Frédo Durand, and William T Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph.*, 31(4):65, 2012.

[132] Yao Xiang, Beiji Zou, and Hong Li. Selective color transfer with multi-source images. *Pattern Recognition Letters*, 30(7):682–689, 2009.

[133] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[134] Xuezhong Xiao and Lizhuang Ma. Color transfer in correlated color space. In *international Conference on Virtual Reality Continuum and its Applications*, pages 305–309. ACM, 2006.

[135] Xuezhong Xiao and Lizhuang Ma. Gradient-preserving color transfer. In *Computer Graphics Forum*, volume 28, pages 1879–1886. Wiley Online Library, 2009.

[136] Wei Xu and Jane Mulligan. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 263–270, 2010.

[137] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. Learning and applying color styles from feature films. In *Computer Graphics Forum*, volume 32, pages 255–264. Wiley Online Library, 2013.

[138] Chuan-Kai Yang and Li-Kai Peng. Automatic mood-transferring between color images. *IEEE Computer Graphics and Applications*, 28(2):52–61, 2008.

[139] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Generalized belief propagation. In *Advances in Neural Information Processing Systems*, volume 13, pages 689–695, 2000.

[140] Jae-Doug Yoo, Min-Ki Park, Ji-Ho Cho, and Kwan H Lee. Local color transfer between images using dominant colors. *Journal of Electronic Imaging*, 22(3):033003–033003, 2013.

[141] Lei Zhang, Sen Wang, and Dimitris Samaras. Face synthesis and recognition from a single image under arbitrary unknown lighting using a spherical harmonic basis morphable model. 2005.

[142] C Lawrence Zitnick and Sing Bing Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1):49–65, 2007.