Enhancement and stylization of photographs

by

Vladimir Leonid Bychkovsky

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 2013

Author ....
Department of
Electrical Engineering and Computer Science
May 15, 2013

Certified by ....
Frédo Durand
Professor
Thesis Supervisor

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

Enhancement and stylization of photographs

by

Vladimir Leonid Bychkovsky

Submitted to the Department of
Electrical Engineering and Computer Science
on May 15, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

A photograph captured by a digital camera may be the final product for many casual photographers. However, for professional photographers, this photograph is only the beginning: experts often spend hours on enhancing and stylizing their photographs. These enhancements range from basic exposure and contrast adjustments to dramatic alterations. It is these enhancements – along with composition and timing – that distinguish the work of professionals and casual photographers.

The goal of this thesis is to narrow the gap between casual and professional photographers. We aim to empower casual users with methods for making their photographs look better. Professional photographers could also benefit from our findings: our enhancement methods produce a better starting point for professional processing.

We propose and evaluate three different methods for image enhancement and stylization. First method is based on photographic intuition and is fully automatic. The second method relies on expert's input for training; after the training this method can be used to automatically predict expert adjustments for previously unseen photographs. The third method uses a grammar-based representation to sample the space of image filter and relies on user input to select novel and interesting filters.

Thesis Supervisor: Frédo Durand
Title: Professor

Acknowledgments

I am immensely grateful to all teachers, colleagues, friends, and family whose continuous support and encouragement I have enjoyed over the years. I would like to express my gratitude to:

- To my advisor, Frédo Durand – Thank you for taking a risk in taking me on as a student. Your patience is matched only by the depth of your insights and your ability to guide and educate your students.

- To Sylvain Paris – Thank you for your direct, honest, and constructive feedback, and, most importantly, thank you for making time even in the busiest of times.

- To Antonio Torralba – Thank you for your contagious enthusiasm and for teaching me new ways to think about computer vision.

- To Bill Freeman – Thank you for making complex concepts simple and thank you for your sage advice about ”how to be a good graduate student”. Even though I have never been a good graduate student, your advice has always been a lighthouse in the stormy ocean of my graduate career.

- To Deborah Estrin – Thank you for developing my taste for research and for encouraging me to go to MIT.

- To Hari Balakrishnan – Thank you for showing me the importance of writing well: your writing inspired me to learn more and to try harder.

- To Tilke Judd – Thank you for welcoming me to the to the Graphics Group, for being an excellent project partner, and for being a friend when it mattered.

- To Connelly Barnes, Eric Chan, and Jeff Chen – Thank you for being great collaborators!

- To David Levin, Shinjiro Sueda, Piotr Dydik – Thank you for participating in my endless user interface trials, for your invaluable feedback, and for your constant encouragement and support.

- To Krzysztof Templin – Thank you for your feedback on this manuscript, it is now a lot more readable.

- To Bryt Bradley – Thank you for keeping things running!

- To all members of the MIT Graphics Group – Thank you for being a great group!

- To Shanna Minior – Thank you for encouraging me to try again.

- To Lewis Girod – Thank you for being there when it mattered the most.

- To my brother, Alex – Thank you for being an awesome roommate, for challenging me, and for helping me make hard decision.

- To my parents, Tatyana and Leonid – Thank you for your unconditional love and eternal support.

- To my fiancée, Brittany Lee – Thank you for believing in me when I did not.

Contents

List of Figures

List of Tables

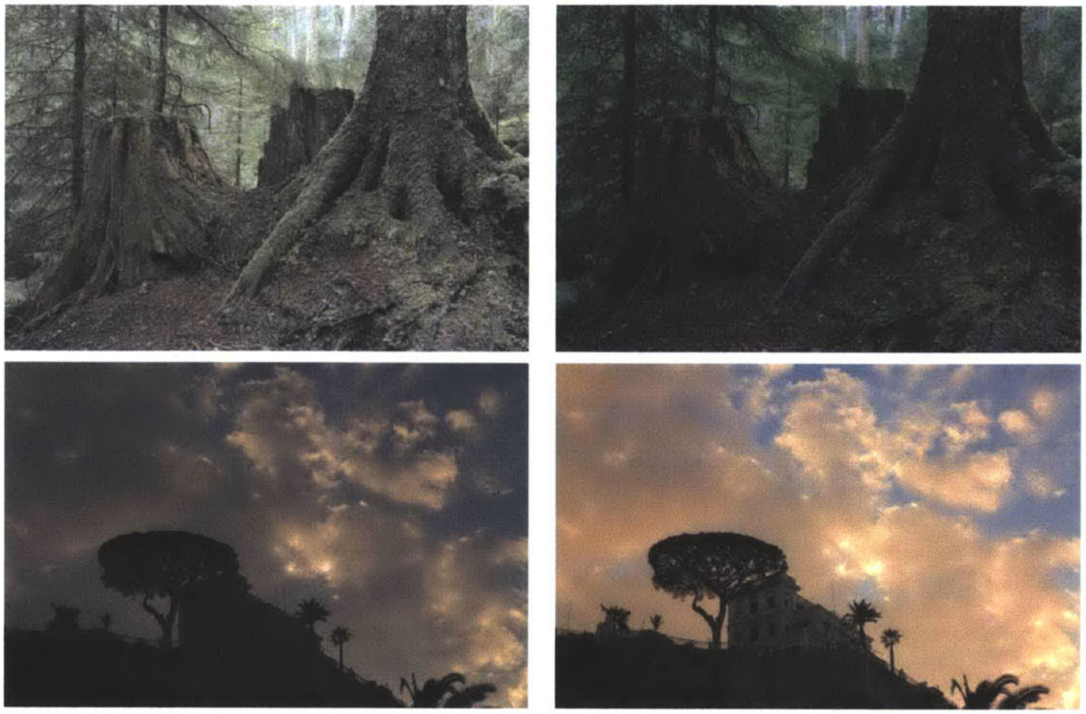
Chapter 1

Introduction

Correct post-processing is crucial for photography: poor exposure, incorrect contrast, and unsightly color shifts can render any photograph unappealing. Professional photographers always understood the importance of post-processing. In the days of film, they spent hours in the darkroom with toxic chemicals processing [3] and printing [4] their photographs before sharing them with the public. Today, film and chemicals have been replaced with digital sensors and sophisticated software, but professionals still spend hours retouching [19, 46] photographs before sharing them.

The post-processing enhancements and stylizations performed by professionals range from subtle corrections for exposure and contrast to marked changes in color rendering to dramatic alterations of the content. Exposure correction aims to fix errors made by the camera exposure system. Contrast adjustment allows to make a better use of the available contrast range and to emphasize the important content in the photograph. White balance correction accounts for the color of light in the photographed scene and allows to subtly alter the mood. Local alternations can bring out important parts of the photograph, for example, eyes and face in a portrait. Some dramatic alterations may result in making the photograph look more like a painting or a sketch. Professional photographers enjoy the full variety of these options when enhancing their photographs. As the result, it is not surprising that professional photographs look better than those produced by casual users.

As imaging technology advanced, most of the photographic capture became auto-



(a) Adobe Lightroom 3 Auto-Tone (b) Human expert adjustment

Figure 1-1: Even the best of the automatic adjustment heuristics (a) pale in comparison with a human expert (b). Note that automatic adjustment methods fail in different ways: the results could be either too bright or too dark.

mated. Most digital cameras today feature automatic exposure metering systems that analyze the scene being photographed. Some cameras even detect faces [75] and smiles to aid capture: these cameras insure that faces are well exposed and that photograph is taken only when smiles are detected. These advances in automatic exposure have freed photographers from doing manual metering thus making photography easier and more enjoyable.

Unlike the automatic exposure, automatic post-processing leaves a lot to be desired. Photographs produced by consumer cameras often have incorrect contrast or exhibit color shifts due to incorrectly set white balance. This difference is especially striking when photographs produced by the automatic methods are compared with a photograph retouched by an expert. Even though little is known about internals of commercial cameras and software, most of the processing seems to be based on simple rules and heuristics. These methods often do improve photographs, however, they also often fail: Figure 1-1 shows examples of such failures. The major cause of failures for rule-based methods is overfitting. The rules are usually designed and hand-tuned using a small set of examples.

In the research literature, photographic post-processing has received only limited treatment. There has been some work on photograph restoration [11], limited attempts at personalization of adjustments [30, 32], a lot of work on style transfer [5,29,52,59], and image abstraction [14,22] and stylization [31,78]. Photo restoration work was focused on recovering photographs that are severely damaged. Adjustment personalization work attempted to learn from a large dataset to avoid overfitting. However, this work relied on a dataset produced by the rule-based method thus limiting the quality of the final result. Style transfer techniques produce very high quality results but only when appropriate examples images are provided. Abstraction and painterly rendering research discovered a number of image filters capable of producing very interesting renditions when filter parameters are set correctly.

In this thesis we propose a data-driven approach to photographic post-processing. We collect and analyze datasets of photographic adjustments and stylizations. Based on this analysis we create fully automatic methods for photograph enhancement and stylization. Using large datasets for learning allows us to avoid overfitting. In addition to addressing the problem of photographic post-processing we propose crowdsourcing-based method for designing novel image filters.

From the practical point of view, the goal of this work is to narrow the gap between casual and professional photographers. We aim to empower casual users by providing them with a number of photographic post-processing methods. These methods vary in the degree and type of user involvement as well as the type of enhancements they perform. Professional photographers could also benefit from our findings: our methods produce a better starting point for their later enhancements.

In the first part, this thesis focuses on one of the most fundamental enhancement operations: global tonal adjustment. Correctly adjusting photographic tones can dramatically improve the appearance of a photograph by correcting exposure and contrast. The global nature of this adjustment guarantees freedom from local contrast reversal artifacts – these artifacts, also know as halos, are common in local enhancement methods. In this thesis, we present and evaluate three methods for adjusting global tonal properties of photographs. These methods differ in the amount of control they provide to the user.

In the later part, this thesis focuses on photograph stylization. First we present a new dataset of stylizations performed manually by professional photographers. Then we propose a method that allows creative professionals to discover novel image filters without having to understand the complexities of image processing. This approach is based on a grammar-based representation of image filters. Such representation enables direct sampling of the space of image filters and frees users from having to tune (and understand) the large number of parameters involved. The results of the sampling are presented to the users for aesthetic evaluation. We propose three different methods for discovering novel filters and evaluate them.

Chapter 2

Background and related work

Photographic post-processing has enjoyed a lot of attention in the past. Many books have been written about post-processing of film and digital photographs. As photography became more popular, cameras and film manufactures created cameras and films that produced distinct looks (or stylizations) of photographs, however photographers still had to spend hours experimenting in the darkroom to obtain the desired look. Digital photography revolutionized post-processing by enabling fast and easy experimentation and very precise creative control over final result. This freedom resulted in a number of new methods for editing, enhancing, and stylizing photographs that form the foundation for our work.

2.1 Rule-based and parametric enhancements

Rules in photographic literature There is a large number of books about photographic enhancements and stylizations [3, 4, 19, 46]. One of the earliest and most influential works is the series of books by Ansel Adams documenting the processes of photographic capture [2], exposure and processing [3], and printing [4] for black-and-white photographs. Adams was first to formalize and document the Zone System [3] for repeatable results in photographic capture and printing. The Zone System introduced photographers to the notion of *dynamic range*. Dynamic range is the number of shades of gray, or *tones*, that can be represented in a given medium, such as photographic film, or paper. Adams pointed out that (1) many natural scenes have more tones than can be captured by film and (2) only some of these tones can be printed. However, Adams also discovered that it is possible to circumvent these limitations by remapping the tones through chemical processing. Understanding the limitations of film and paper and remapping possibilities lead Adams to the notion of *pre-visualization.* The process of pre-visualization can be translated into a set of rules for metering the scene brightness, for setting exposure, and for printing. Armed with these rules, photographer could think through the whole photographic processes and set capture parameters according to the desired (pre-visualized) final result.

Digital photography made it much easier to control photographic tones, colors [46], and to composite photographs [19]. However, even digital technology has its limitations: The photographer has to decide what to give up and what to keep, what to hide and what to emphasize. Making this trade-off correctly requires understanding of the technical limitations and good taste.

Cameras and commercial software Most digital cameras today are equipped with an automatic exposure metering system. There is little documentation available about these proprietary systems; at least some of the cameras seem to rely on very simple criteria such distribution of the scene brightness [1, 37]. A lot of cameras also offer in-camera enhancements options. The associated algorithms are part of the camera's proprietary firmware, which makes them difficult to evaluate. A lot of these algorithms seem to focus on increasing color saturation and doing some form of tone-mapping to bring out shadow details.

Comprehensive photo editing software, such as Adobe Photoshop, enables arbitrary pixel modifications with a plethora of tools. However, Photoshop and similar tools require a considerable expertise to use. There is also a number of specialized software tools available for photographers. For example, Adobe Lightroom and Apple Aperture provide photographers with essential controls for adjusting individual photographs and photographic collections. Even though these tools are easier to use than full-fledged image editing software, they still require considerable expertise.

A lot of commercial photo editing tools offer automatic adjustment heuristics, however, their results leave a lot to be desired (see Figure 1-1). These heuristics are often based on simple rules that apply only to a small number of photographs. These heuristics are often developed by hand-tuning parameters and visual examination of results on a small set of photos. This manual tuning and small dataset example photographs often leads to overfitting, which is, at least in part, responsible for poor performance on difficult images.

Global methods Histogram equalization [62] is a simple technique that is commonly used to automatically enhance photographs. By remapping intensity values such that the image has a uniform histogram, this method increases global and local contrasts and reveals the image details. Histogram equalization amounts to computing the intensity histogram of the image and its cumulative distribution function (CDF). Remapping each intensity value using the CDF produces an image with a uniform histogram. This approach is simple, computationally inexpensive, and works well on data such as x-ray images. Furthermore, since the same non-decreasing remapping curve is applied to every pixel, it guarantees that no gradient reversal appears, which is often the cause of halos. Unfortunately, histogram equalization is often unsuccessful on photographs, because equalizing the histogram may reduce contrast in important parts of the scene. Moreover, it tends to exaggerate the contrast in uniform areas, producing unpleasant gradients and banding.

Mantiuk et al. [43] describe a technique to compute a global remapping curve for tone mapping high-dynamic range photographs by accounting for the display capabilities and the viewing environment. This technique focuses on a faithful reproduction of high-dynamic range photographs, whereas our methods aim at enhancing the visual appeal of regular photographs.

Cohen-Or et al. [10] have developed a method for color harmonization. Their method does not alter the tones, but shifts color in a photograph towards more harmonious hues, thus making the photograph more attractive.

Local methods Local tone-mapping methods[1] convert a high dynamic range (HDR) photograph into a photograph that can be displayed on a regular (i.e. low dynamic range) display. The goal of these methods is to produce attractive and realistic photographs. Durand et al. [16] note that HDR images contain large regions of very different brightness (such as sky and ground) and that the absolute brightness of these regions does not affect the perceived quality of the image. Armed with this observation, they propose a tone-mapping method that separates the image into a base and detail layers, compresses the base layer, and recomposes the layers.

Fattal et al. [21] propose a gradient-based method for creating an aesthetically pleasing and naturally looking rendition of a high dynamic range image. This method relies on the observation that humans are not sensitive to the absolute value of large gradients. This method compresses image gradients that are above a certain threshold while leaving smaller gradients untouched.

Farbman et al. [20] propose a tone-management method that builds on two-layer approach of Durand et al. [16] and decompose a photo into multiple layers. Such decomposition allows to manipulate image features at various scales providing photographers with additional creative control.

Local image edits such as blemish removal [8], other face-specific edits [7, 26], and search-and-replace edits for image collections [27] rely on detecting local feature detection and on image blending [55] and other local techniques transfer edits between images. These techniques work very well for specific local edits as long as sufficient local features are available.

Parametric enhancement and stylizations A lot of research in the past has focused on creating specific parametric filters that either enhanced or stylized images. Specific examples include tone-mapping for HDR images [38, 58], detail enhancement, [20,49,56], abstraction, [14,22], and painterly rendering, [31,78]. Each of these techniques is carefully designed to produce one look or a range of similar looks based on parameters specified by the user.

[1] See [60] for a complete survey.

2.2 Data-driven photograph enhancements

Restoration of photographs Dale et al. [11] restore damaged photos using a corpus of images downloaded from Internet. This method is designed to recover grossly incorrect white balance and exposure. This approach relies on image segmentation and on learning nominal appearance of different parts of the scene. An Internet-scale database of photos is searched for similar image patches and color. After the match is found, tone properties are transferred from these patches to the damaged photograph.

Example-based photo enhancement Kang et al. [32] personalize the output of an automatic adjustment method by using a small but carefully chosen set of examples from their collection. Given a new image, their approach copies the adjustment of nearest user-retouched example. To determine the similarity metric between photos, Kang et al. use metric learning and sensor placement [36]. However, metric learning requires a large training set to be effective. On that issue, Kang et al. note that it is infeasible for any user to find these parameters manually because no large collection of photos including untouched input and retouched versions is available, which motivates their use of synthetic training data.

White balance correction Gehler et al. [23] have shown that supervised learning can be a successful approach to inferring the color of the light that illuminates a scene. In photographic terms, this refers to setting neutral white balance, which is most often the starting point for further enhancements.

Example-based style transfer Efros and Freeman [18] propose fast variant of non-parametric texture synthesis [17] and adapt it to texture transfer, which is a form of image abstraction. Texture transfer is performed using a correspondence map. This map insures that generated texture matches the source image in blurred intensity or some other property. This technique produces very impressive image stylizations when appropriate texture and matching metric is chosen for transfer. Hertzmann et al. [29] draw inspiration from non-parametric texture synthesis methods and develop a method of non-parametric style transfer that is a capable of simulating different styles of painting.

Bae et al. [5] observe that tonal adjustment styles of different photographers can often approximated by global and local histogram statistics. Authors use this insight to decompose a photograph into base and detail layers using an edge-aware filter [50] and to compute histograms for each layer. Then they use histogram transfer at two scales to transfer the style. Since independent histogram matching at different scales can result in local gradient reversals, a Poisson correction is applied as the last step.

Pitie et al. [52] note that the style of a color photograph depends on its color distribution. They apply this insight to style transfer by transferring color distributions of the example and target images. Such transfer alone can often increase color noise. Authors use an additional smoothing stage to reduce noise introduced by color distribution matching.

Pouli et al. [53] propose a style transfer method based on the observation that colors distributions need to match only approximately. Armed with this intuition authors match major modes of histograms independently but at multiple scales.

Inspired by the variety of the color transfer work, Reinhard at el. [59] use a data-driven approach to find the best color space for color transfer. After performing color transfer in a number of different color spaces author conclude that CIE Lab is the best color space to use. This is the colorspace used for all methods in this thesis.

All of the above style transfer methods have been shown to perform well when given an example photo that semantically matches the target. The challenge, however, lies in finding an appropriate example photo for a given target, as the content of the example photo affects the style transfer quality. For example the high contrast style taken from a mountainous landscape may make a close up of a baby unappealing. In this thesis, we focus on methods that work equally well for various types of content.

Content vs. style in photographic adjustments The photograph adjustment process can be viewed as two separate steps: *repair* and *stylization*. For example, given an underexposed photograph, the repair step would be to increase exposure of the image and maximize its overall contrast; the stylization step would amount to correcting contrast, brightness, and color to attain a certain look (e.g. high-key, low-key, vintage, etc.). From this point of view, this problem is related to style vs. content separation work by Tenenbaum and Freeman [73].

Portrait beautification Leyvand et al. [39] used a data-driven approach to model facial attractiveness. Facial attractiveness metrics are derived from a large collection of labeled portraits. These metrics are later used to evaluate and reshape new portraits to alter their attractiveness.

2.3 Image filters, evolution, and crowd-souring

Sims [66] generated abstract images with genetic algorithms. Sitthi-amorn et al. [67] used genetic algorithms to simplify rendering shaders. Tan et al. [70] generated swimming creatures. Reynolds [61] evolved camouflaging patterns using genetic algorithms. Our work differs from the above into two major ways: (1) we apply genetic algorithms to image filters, and (2) we rely on crowd-sourcing to provide a fitness function for genetic algorithms.

Procedural modeling based on a grammar is well adapted to create structured objects such as trees [51, 54]. Unlike these techniques, our work does not use a grammar to define the final visual results. We use a grammar to represent image filters that are later applied to photographs.

The development of platforms such as Amazon Mechanical Turk makes it easy to work with a large number of users. Common tasks related to computer graphics are the analysis and annotation of images to obtain information that would otherwise challenging to obtain with an algorithm, e.g. depth discovery by Gingold et al. [24]. Our work is different in that it focuses on a creative and open-ended task. MacCallum et al. [41] use genetic algorithms and crowd-sourcing to evolve music. Xu et al. [79] use user-directed genetic programming to generate 3D models. This research is very similar to ours in spirit, but we focus on evolving image filters.

Marks et al. [47] introduced the idea of design galleries as a general approach of parameter tuning to the computer graphics community. This approach was later been applied to various problems such as BRDF modeling [48] and image adjustment exploration [63], and generating 3D models [69]. Our approach is similar as we also hide the model parameters from the user. However, instead of having a fixed number of parameters (and, thus a fixed number of dimensions), we use a grammar-based representation that is significantly more complex (as the structure of the filer and the number of parameters vary between filters).

2.4 Visual similarity and quality

Similarity metrics Data-driven approaches to image enhancement rely on an image similarity metric to compare generated images to ground truth. Visual difference predictor [44, 45] is designed to measure differences between images as they would be perceived by a human observer. It is specifically designed to simulate human visual system when comparing two images.

The structural similarity index (SSIM) [76] is a method that was developed as a more perceptually accurate alternative to the mean sum of squared differences method. SSIM is based on the observation that human observers are more sensitive to structures changes in the image (i.e. changes in edges) than to low-frequency changes. Given two images, SSIM produces a scalar in the range of $[-1; 1]$ to signify how similar these images to each other.

Visual quality Photo enhancement problem can be viewed as a problem of increasing the quality of input photo through various adjustments. Datta et al. [13] and Dhar et al. [15] study image quality, evoked emotions [12], and low-level image features. The authors use image ranking data from a popular photo-sharing websites to analyze and predict image quality. This method relies on various classification and regression techniques to predict the quality and rank of a previously unseen images based on low-level features such as image luminance and color histograms, and compressibility (i.e. image smoothness).

Luo et al. [40] improve on results of Datta et al [13] by carefully designing set of image features that corresponds to photographic intuition. The proposed features aim to take into account the scene composition and pay special attention to the elements of the photograph that are in focus.

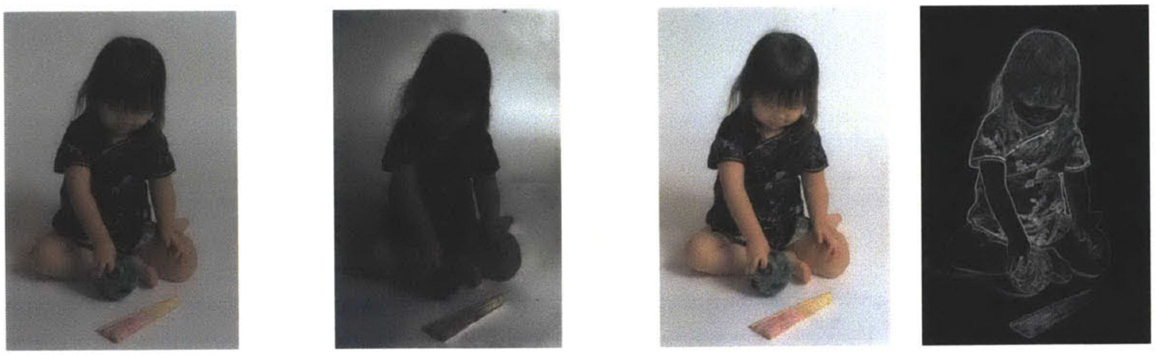
Yoshida et al. [81] study the effect of tone-mapping parameters for high dynamic range display on reported visual quality. As the result of this study a new tone-mapping operator with an intuitive set of controls is created. These controls allow the viewer to easily improve their viewing experience.

Chapter 3

Detail equalization: Automatic tonal enhancement

3.1 Abstract

Tonal enhancement is a major step in editing a photograph. Even though histogram equalization can successfully enhance some images, it often flattens contrast in important regions of photographs. We propose a new automatic photograph enhancement method that addresses this problem while maintaining the simplicity and speed of histogram equalization. We extend histogram equalization by associating weights with every pixel according to the local amount of detail. Using these weights, we build a weighted histogram and derive a tone remapping curve that increases contrast more in textured areas and less in flat areas, thereby revealing details. We show that our approach does not exhibit the unpleasing tone compression artifacts generated by histogram equalization, and that it naturally avoids halos that can appear with local techniques. We demonstrate our automatic tone enhancement method on photographs and HDR images.



(a) Input

(b) Naive histogram equalization

(c) Our result

(d) Weight: black=0, white=1

Figure 3-1: Untouched photographs (a) often lack contrast. Naive histogram equalization (b) increases the overall contrast but often causes unsightly artifacts. For instance, in this picture (b), the contrast of the background is overly increased and the one of the girl overly reduced. In comparison, our approach (c) uses a weight map (d) to allocate a larger portion of the tonal range to the detailed regions, the girl in this picture, thereby producing a pleasing image.

3.2 Introduction

Tonal adjustment is an important step when editing a photograph. Histogram equalization is a simple and popular technique to remap intensity values so that the output histogram is uniform. It has the advantages of ensuring a few common rules of thumb, such as "the brightest point should be white" and "the darkest point should be black". On most images, it also increases contrast, making details more visible. However, histogram equalized photographs are not always visually pleasing. Contrast increase is controlled by the number of pixels sharing a given intensity value, regardless of what is represented by these pixels. For instance, in Figure 3-1(b), the white background is emphasized because it is larger than the girl. However, the background is actually of little interest and the produced image is not satisfying. In this chapter, we introduce detail equalization, an extension of histogram equalization that accounts for the local amount of detail in the image to define the remapping curve. Instead of distributing the intensity values uniformly, our approach seeks to distribute details equally across the intensity range. That is, we aim for an image with as much detail in the bright areas as in the dark areas and mid-tones. As an example, since the girl in Figure 3-

1(c) is more detailed than the background, our approach emphasizes her more, which yields a better image.

Our approach is based on the notion of a weighted histogram. We associate a weight to each pixel and each histogram bin sums the weights corresponding to a given intensity value. Then we use the standard equalization technique to this weighted histogram to obtain a remapping curve. We derive weights such that they represent the local amount of detail in the neighborhood of each pixel. To estimate this quantity, we propose a simple, computationally inexpensive scheme based on image gradients. We demonstrate that our approach achieves satisfying results for automatic photo adjustment and tone mapping of high dynamic range images.

Photographers have developed a wealth of techniques to ensure that tones and details are accurately captured and preserved until the image is displayed or printed [3, 4]. All photo editing software provides tools, such as the curve tool and the unsharp mask, to remap image tones and to adjust details. Whereas these tools are user-driven, in this chapter we focus on automatic adjustment.

Histogram equalization is a simple technique that is commonly used to automatically enhance images. By remapping intensity values such that the image has a uniform histogram, this method increases global and local contrasts and reveals the image details. Histogram equalization amounts to computing the intensity histogram of the image and its cumulative distribution function (CDF). Remapping each intensity value using the CDF produces an image with a uniform histogram. This approach is simple, computationally inexpensive, and works well on data such as x-ray images. Furthermore, since the same non-decreasing remapping curve is applied to every pixel, it guarantees that no gradient reversal appears, which is often the cause of halos. However, histogram equalization is not always successful on photographs, because equalizing the histogram can in some cases reduce the contrast in small but important regions. Moreover, it tends to exaggerate the contrast in uniform areas, producing visually unpleasing gradients (Figure 3-1(b)).

In the context of tone mapping of high-dynamic range (HDR) scenes [60], Ward et al. [38] prevent exaggerated contrast due to histogram equalization by imposing an upper bound onto the slope of the remapping curve. However, pixel count is still the underlying criterion to determine regions to be emphasized, which can lead to ignoring small important areas. Nonetheless, we use the same approach to prevent contrast stretching, but, instead of pixel count, we use the local amount of detail to determine the regions to be emphasized.

Contributions In this chapter, we introduce *detail equalization*, a method that globally remaps pixel intensities to automatically enhance photographs. We explain how to extend histogram equalization to uniformly distribute details instead of intensities. We show that this approach better reveals image details without introducing artifacts. We apply detail equalization to automatic adjustment of photographs and automatic tone mapping of HDR images. Our results are visually pleasing and our algorithm is computationally lightweight.

3.3 Detail Equalization

We first describe our input data. Then we introduce *weighted histograms* and use them to define tone remapping curves. Then, we expose how to compute the weights. Finally, we explain how we modify the remapping curve to prevent contrast stretching.

Input We use RGB images as input. We compute a luminance channel $Y = 0.265R + 0.670G + 0.065B$ [77]. We apply our method to the logarithmic luminance which approximates human perception of contrast [5, 38]. That is, we work on a gray-scale image I defined as $I = \log Y$. Once we have processed I, we reconstruct a color image using $C_{\text{out}} = C_{\text{in}} Y_{\text{out}} / Y_{\text{in}}$ where C represents any of the RGB channels and the subscripts indicate whether we refer to the input or output data. This model performed well in our experiments. Nonetheless, our approach can also be used with other models such as the one of Mantiuk et al. [42].

Weighted Histogram In this paragraph, we assume that we are given a weight function w that assigns a positive value to each image pixel $\mathbf{p}$. We will later define

w so that it quantifies the local amount of detail around each pixel. We compute the normalized weighted histogram H_w as follows:

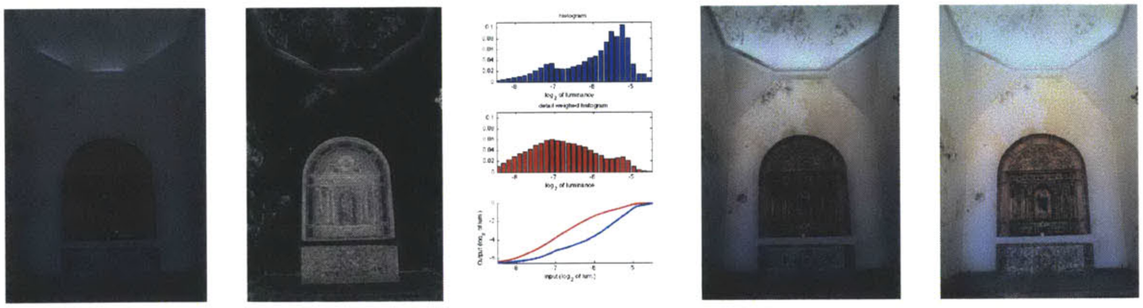
$$H_w(i) = \frac{\sum_{\mathbf{p}} w(\mathbf{p})\, \delta(I(\mathbf{p}) - i)}{\sum_{\mathbf{p}} w(\mathbf{p})} \tag{3.1}$$

where i is an intensity value and $\delta(x)$ is the Kronecker symbol equal to 1 if $x = 0$ and to 0 otherwise. If w is constant and equal to 1, H_w is the standard histogram of I that counts how many pixels have an intensity i. In general, $H_w(i)$ sums the weights of the pixels with an intensity i.

Remapping Curve We use the CDF of H_w to define a remapping function R_w:

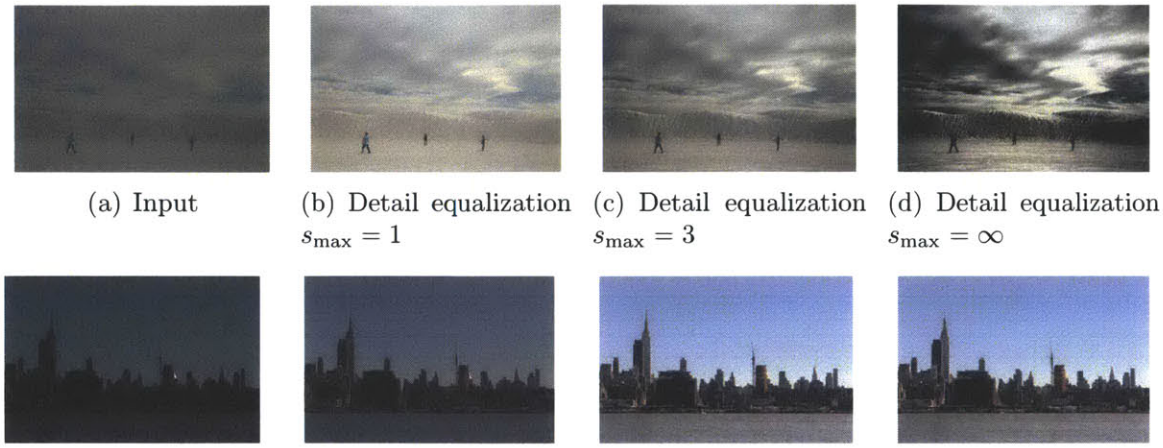
$$R_w(i) = i_{\min} + (i_{\max} - i_{\min}) \sum_{j \leq i} H_w(j) \tag{3.2}$$

where $i_{\max}$ and $i_{\min}$ are the maximum and minimum intensities of the tonal range. This is exactly histogram equalization [25] except that we use the weighted histogram H_w instead of the standard intensity histogram. When applied to I, this remapping curve equalizes H_w, that is, it redistributes the w values uniformly over the tonal range. Figure 3-2 illustrates the difference between intensity histogram equalization



(a) Input (b) Weights (c) Histograms and curves (d) Constrained equalization (e) Our result

Figure 3-2: Our weights (b) significantly alter the histogram (c) by giving more influence to the detailed regions. The remapping curves (c) computed from these histograms are also significantly different: standard histogram equalization increases mostly the contrast of the wall (d) because it has a high pixel count whereas our approach emphasizes more the bas-relief (e) because it is more detailed (b).



(a) Input

(b) Detail equalization $s_{\max} = 1$

(c) Detail equalization $s_{\max} = 3$

(d) Detail equalization $s_{\max} = \infty$

(e) Input

(f) Detail equalization $s_{\max} = 1$

(g) Detail equalization $s_{\max} = 3$

(h) Detail equalization $s_{\max} = \infty$

Figure 3-3: In rare cases, detail equalization can overly stretch the image contrast (a,d). To prevent this artifact, we enforce an upper bound $s_{\max}$ on the slope of the remapping curve. Setting $s_{\max} = 3$ successfully removes the artifacts (c). We also tested stricter bounds such as $s_{\max} = 1$ but such a low value also alters artifact-free results and often produces duller outputs (f). In comparison, $s_{\max} = 3$ only removes extreme stretching (c) and keep unchanged visually pleasing images (g,h).

and our approach.

Weights We seek to define weights w that represent how much detail lies in the neighborhood of each pixel. We estimate the local variations of I with its gradients ∇I. In practice, we use the amplitudes of the image gradients as weights, that is, $w(\mathbf{p}) = \|\nabla I(\mathbf{p})\|$, which is simple and has produced satisfying results in our experiments. We compared this scheme to other approaches such as the *power maps* of Su et al. [68] and the *textureness* of Bae et al. [5]. We obtained similar outputs and decided to work with image gradients since they require less computation. Nonetheless, if desired, it is straightforward to apply another technique to estimate the local amount of texture.

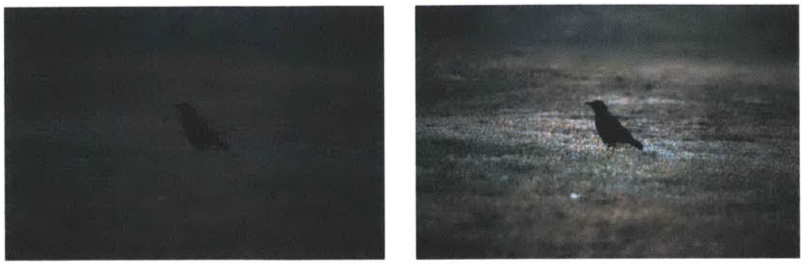
Figure 3-1(d) shows an example of our weights. In this case, the girl received most of the high weights while the background is assigned almost no weight. This explains why our method assigns most of the tonal range to the girl, thereby rendering a satisfying image (Figure 3-1(c)). In comparison, histogram equalization emphasizes the background because of its high pixel count and reduces the contrast on the girl, which produces a visually unpleasing image (Figure 3-1(b)).

Preventing Contrast Stretching As pointed out by Ward et al. [38], image contrast can be overly stretched if the slope of the remapping curve is too steep. To prevent this from happening, we apply the *histogram adjustment* procedure suggested by Ward et al. This procedure takes as parameter the upper limit $s_{\max}$ that is imposed on the slope of the remapping curve. Ward et al. show that this imposes the same limit on the contrast increase. A difference between the approach of Ward et al. and ours is that we aim for visual enhancement, which often requires some amount of contrast increase, whereas Ward et al. target visual faithfulness and strictly enforce no contrast increase with $s_{\max} = 1$. In our experiment, we found that $s_{\max} = 3$ successfully prevents artifacts while still allowing detail enhancement (Figure 3-3). All the results in this chapter are computed with this value unless otherwise specified.

3.4 Results

Figure 3-7 shows a series of results obtained from regular photographs. We compare our approach to histogram equalization. Both methods use histogram adjustment with $s_{\max} = 3$ to prevent contrast stretching. Our method consistently produces visually more pleasing images. Figure 3-5 shows results from HDR images. We compare our method with histogram equalization. For the latter, we use $s_{\max} = 3$ as our method and $s_{\max} = 1$ as suggested by Ward et al. for tone mapping. As expected, on these HDR images, our approach produces more detailed outputs whereas histogram equalization results are more neutral. While the final choice may be a subjective matter, our approach offers a useful alternative.

Figure 3-2 illustrates the effect of the weights that we introduce in our approach. While the original image has a histogram that is dominated by the bright tones because of the white background, our weights produce a histogram with strong mid tones because of the detailed bas-relief. These different histograms translate into different curves which, in turn, render different images. Histogram equalization



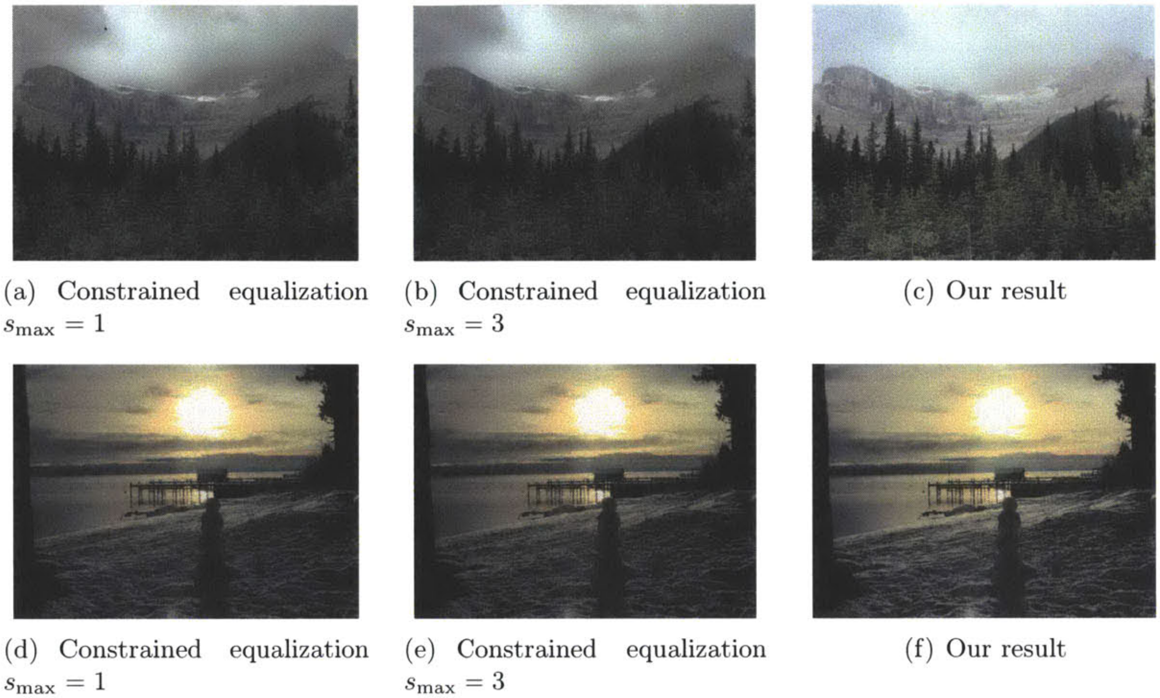
(a) Input (b) Our result

Figure 3-4: In this example, our method overly emphasizes the grass in the background because it is as detailed as the bird. However, these cases are rare and our method performs well in general (Figure 3-7).

phasizes the cracks in the walls by increasing the contrast in bright regions. Our method reveals the fine details of the bas-relief by increasing the contrast in areas with medium brightness.

Running Time Our Matlab prototype takes about 1.9 second to process a 1 megapixel image on an Intel Core 2 Duo 3.06GHz with 6MB of cache.

Discussion The underlying assumption of our method is that regions of interest are detailed. While our experiments show that this is often the case, some images do not have this property and our method produces less satisfying results on them (Figure 3-4). During our tests, we have also remarked that histogram equalization and our method behave consistently differently on cloudy scenes. The former strongly increases the contrast of the clouds because they cover half of the image, which renders compelling dramatic skies but at the expense of compressing the rest of the scene. In comparison, our method flattens the clouds because they are smooth and reveals the details in the rest of the scene. Figure 3-6 illustrates this difference. While we believe that our approach is a safer choice because it does not compress the scene content, an interesting future work would be to combine both techniques using a cloud detector [71].



(a) Constrained equalization $s_{\max} = 1$

(b) Constrained equalization $s_{\max} = 3$

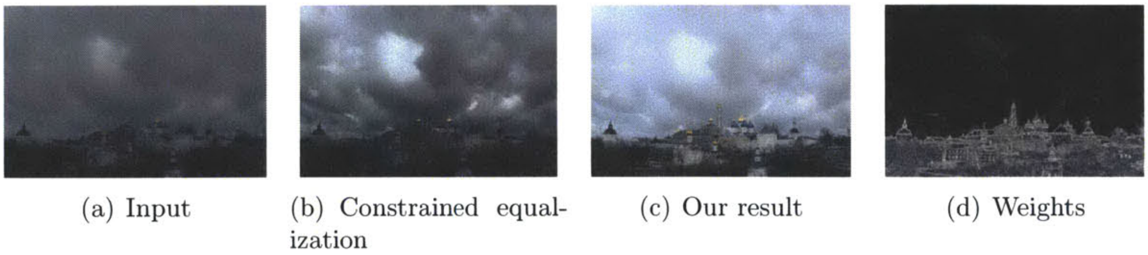
(c) Our result

(d) Constrained equalization $s_{\max} = 1$

(e) Constrained equalization $s_{\max} = 3$

(f) Our result

Figure 3-5: When applied to HDR images, both histogram equalization and our method render satisfying tone-mapped results. The former produces more neutral renditions (a,b,d,e) and the latter more detailed ones (c,f). These differences are mostly visible on the trees (a,b,c) and on the snow (d,e,f). These results may be better seen on a screen.



(a) Input

(b) Constrained equalization

(c) Our result

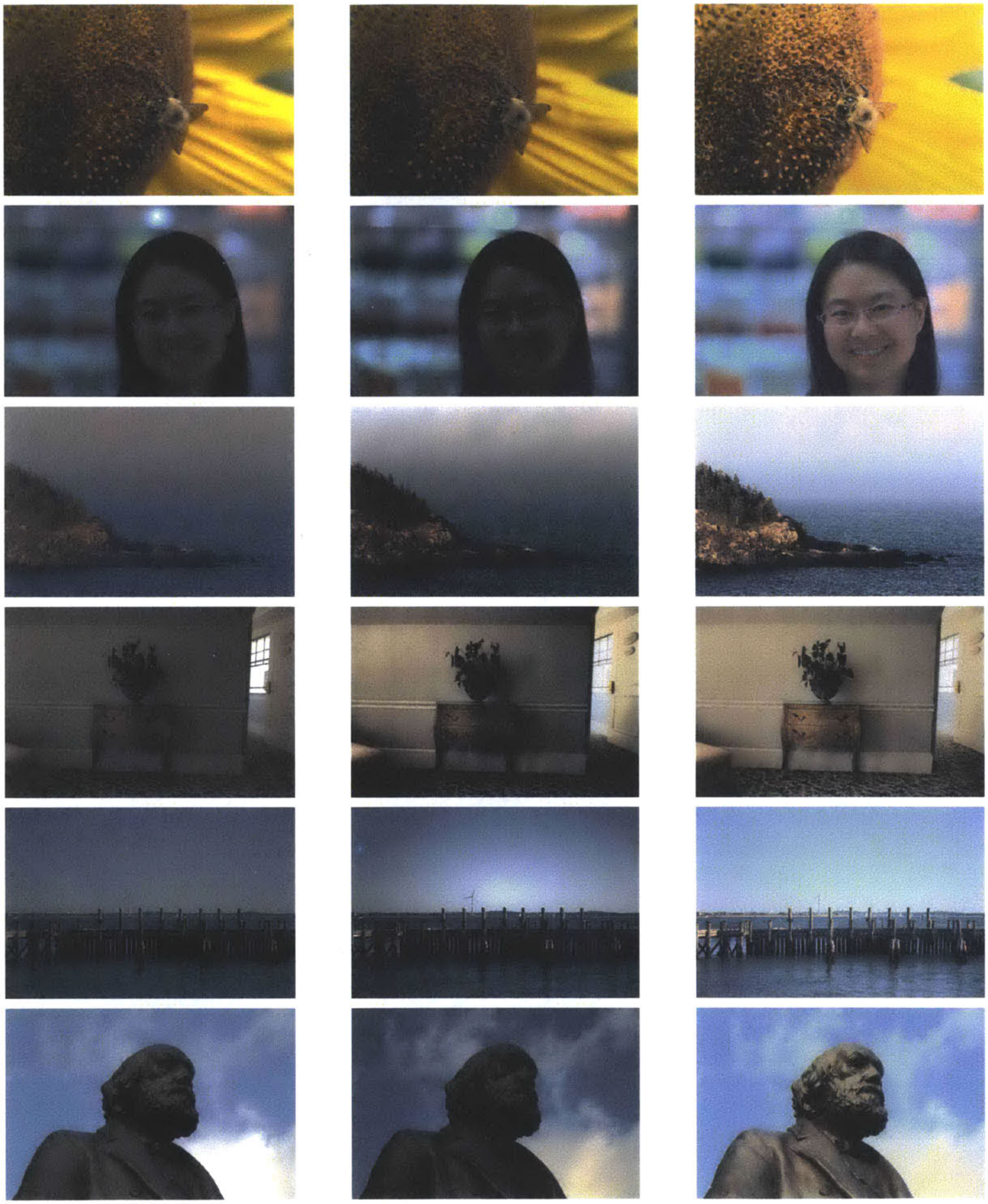
(d) Weights

Figure 3-6: For cloudy scenes (a), constrained histogram equalization (b) and our method (c) consistently generate different renderings. The former produces dramatic skies while the later emphasizes the rest of the scenes where most of the details are (d). Our approach is a safer choice because histogram equalization sometimes overly compresses the scene. Nevertheless, combining both methods is an interesting avenue for future work.

3.5 Conclusions and lessons learned

We presented detail equalization, a method to compute remapping curves to enhance photographs. It relies on a weight map computed using image gradients. Our method is simple and our tests on a variety of scenes show that it reliably produces visually pleasing images.

Even though this method produces pleasing results for large number of photographs it fails for some photographs. Unfortunately, this is the faith of most heuristic-based method, as they are usually designed using a small set of examples. A more structured approach to this problem is to collect a large dataset of input photographs and desired enhancements. Such dataset would allow to use statistical methods and machine learning to train an adjustment prediction algorithm. This is the approach we take in the next chapter.



(a) Input photograph

(b) Constrained equalization

(c) Our result

Figure 3-7: Input photographs (a) often lack contrast. Histogram equalization (b) increases contrast by uniformly redistributing intensities across the available range. Such redistribution often expand the contrast of large uniform regions and compresses small but important details in the images, which yields unsatisfying outputs. Our approach (c) does the opposite: it flattens uniform areas to preserve important detail. Depending on the printer, these results may be better seen in the electronic version. More example photographs are provided in supplemental material.

Chapter 4

Learning and predicting global tonal adjustments

4.1 Abstract

Adjusting photographs to obtain compelling renditions requires skill and time. Even contrast and brightness adjustments are challenging because they require taking into account the image content. Photographers are also known for having different retouching preferences. As the result of this complexity, rule-based, one-size-fits-all automatic techniques often fail. This problem can greatly benefit from supervised machine learning but the lack of training data has impeded work in this area. Our first contribution is the creation of a high-quality reference dataset. We collected 5,000 photos, manually annotated them, and hired 5 trained photographers to retouch each picture. The result is a collection of 5 sets of 5,000 example input-output pairs that enable supervised learning. We first use this dataset to predict a user's adjustment from a large training set. We then show that our dataset and features enable the accurate adjustment personalization using a carefully chosen set of training photos. Finally, we introduce difference learning: this method models and predicts difference between users. It frees the user from using predetermined photos for training. We show that difference learning enables accurate prediction using only a handful of examples.

4.2 Introduction

Adjusting tonal attributes of photographs is a critical aspect of photography. Professional retouchers can turn a flat-looking photograph into a postcard by careful manipulation of tones. This is, however, a tedious process that requires skill to balance between multiple objectives: contrast in one part of the photograph may be traded off for better contrast in another. The craft of photo retouching is elusive and, while a plethora of books describe issues and processes, the decision factors are usually subjective and cannot be directly embedded into algorithmic procedures. Casual users would greatly benefit from automatic adjustment tools that can acquire individual retouching preferences. Even professional photographers often wish they could rely more on automatic adjustment when dealing with large collections in a limited amount of time (*e.g.* a wedding photoshoot). Photo editing packages offer automatic adjustment such as image histogram stretching and equalization. Unfortunately, such simple heuristics do not distinguish between low- and high-key scenes or scenes with back-lighting and other difficult lighting situations.

We propose to address the problem of automatic global adjustment using supervised machine learning. As with any learning approach, the quality of the training data is critical. No such data are currently available and previous work has resorted to rule-based, computer-generated training examples [32]. Another alternative is to use on-line photo collections such as Flickr, e.g. [11]. However, since only the adjusted versions are available, these methods require unsupervised learning. This is a hard problem and requires huge training sets, up to a million and more. Furthermore, it is unclear how to relate the adjusted output images to the unedited input [11]. This makes it impossible to train such methods for one's style, as a user would have to manually adjust thousands of images. To address these shortcomings and enable high-quality supervised learning, we have assembled a dataset of 5,000 photographs, with both the original RAW images straight from the camera and adjusted versions by 5 trained photographers (see Figure 4-1 for an example).

The availability of both the input and output image in our collection allows us to use supervised learning to learn global tonal adjustments. That is, we learn image transformations that can be modeled with a single luminance remapping curve applied independently to each pixel. We hypothesize that such adjustments depend on both low level features, such as histograms, and high-level features such as presence of faces. We propose a number of features and apply a regression techniques such as linear least squares, LASSO, and Gaussian Process Regression (GPR). We show a good agreement between our predicted adjustment and ground truth.

While a brute-force supervised learning approach is convenient for learning a single "neutral" rendition corresponding to one of the photographers hired to retouch our dataset, it necessitates a large investment in retouching thousands of photographs. In order to accommodate a greater variety of styles without requiring thousands of examples for each style, we build on Kang et al. [32]: We seek to select a small number of photographs so that adjustments on new photos can be best predicted from this reduced training set. A user then only needs to retouch this small set of training photographs to personalize future adjustments. We show that our dataset together with our new features provide significant performance improvement over previous work.

The above-mentioned approach still requires users to retouch a predefined set of images that come from the database, as opposed to their own photos. We want to alleviate this and learn the adjustments of a user directly from arbitrary photographs. We hypothesize that there is a correlation between users. We use a two-step approach, where the prediction from our neutral style trained on thousands of images is combined with a method that learns on-the-fly the difference between neutral and the new style of adjustment. The learning is further helped by the use of a covariance matrix learned on the large database. We show that this can enable good predictions using only a handful of user-provided adjustments.

4.2.1 Contributions

A reference dataset We have collected 5,000 photos in RAW format and hired 5 trained photographers to retouched each of them by hand. We tagged the

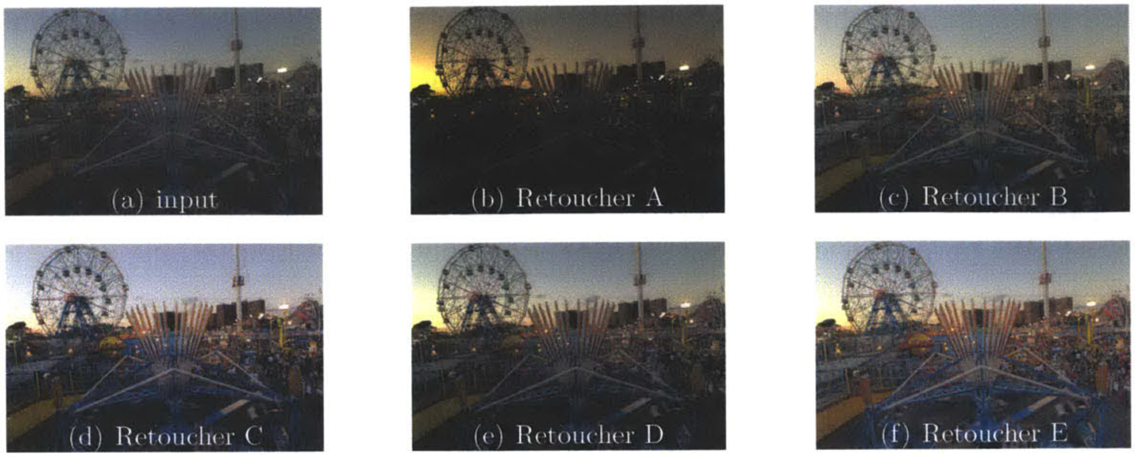


Figure 4-1: On this photo, the retouchers have produced diverse of outputs, from a sunset mood (b) to a day light look (f). There is no single good answer and the retoucher's interpretation plays a significant role in the final result. We argue that supervised machine learning is well suited to deal with the difficult task of automatic photo adjustment, and we provide a dataset of reference images that enables this approach. This figure may be better viewed in the electronic version.

photos according their content and ran user study to rank the photographers according to viewers' preference.

Global learning We use this dataset for supervised learning. We describe a set of features and labels that enable the prediction of a user's adjustment.

Sensor placement Our dataset enables sensor placement to select a small set of representative photos. Using adjustments made to these photos by new users we accurately learn preferences of new users.

Difference learning We show that predicting the difference between two photographers can generate better results than predicting the absolute adjustment directly, and that it can be used for learning users' preferences on-the-fly.

4.3 A Dataset of Input-Output Photographs

We have collected 5,000 photographs taken with SLR cameras by a set of different photographers. They are all in RAW format, i.e., all the information recorded by the camera sensor is available. We have made sure that the photographs cover a broad diversity of scenes, subjects, and lighting conditions. We then hired five photography students in an art school to adjust the tone of the photos. Each of them retouched all the 5,000 photos using a software dedicated to photo adjustment (Adobe Lightroom) on which they were extensively trained. We asked the retouchers to achieve visually pleasing renditions, akin to a postcard. The retouchers were compensated for their work. A visual inspection reveals that the retouchers made large modifications to the input images. Moreover, their adjustments are nontrivial and often differ significantly among the retouchers. Figure 4-1 shows an example of this diversity. We numerically evaluate these points with statistics computed in the CIE-Lab color space. The difference between the input photo and the retouched versions is 5.5 on average and can be as much as 23.7. And the average difference between the retouched version is 3.3 and the maximum is 23.5. For reference, the difference between white and black in CIE-Lab is 100. We also augmented the dataset with tags collected with Amazon Mechanical Turk to annotate the content of the photos. We also ran a user study in a controlled setting to rank photographers according to users' preference on a subset of our dataset.

We studied the dimensionality of the tone remapping curves that transform the input image luminance into the adjusted one. We found that the first three principal components explain 99% of the variance of the dataset and that the first component alone is responsible for 90% of it. This is why we focus our learning on this component.

4.4 Learning problem setup

4.4.1 Labels

We express adjustments as a remapping curve from input luminance into output luminance, using the CIE-Lab color space because it is reasonably perceptually uniform. The curve is represented by a spline with 51 uniformly sampled control points. We fit the spline to the pairs of input-output luminance values in a least-squares sense.

We want to avoid bias due to the type of camera used for a photo and the skill of the particular photographer. In particular, different camera metering systems or a user's manual settings might result in different exposures for a given scene. This is why we normalize the exposure to the same baseline by linearly remapping the luminance values of each image so that the minimum is 0 and the maximum 100.

We focus on learning the first PCA coefficient of the remapping curves, which is a good approximation to the full curve (§ 4.3). At run time, we predict the new adjustment by reconstructing the full curves and interpolating linearly between samples.

4.4.2 Features

The features that we use for learning are motivated by photographic practice and range from low level descriptions of luminance distribution to high-level aspects such as face detection. Before computing features, we resize the images so that their long edge is 500 pixels.

▷ *Intensity distributions:* Photographers commonly rely on the distribution of intensities as depicted by a log-scale histogram to adjust the tonal balance. We consider the distribution of the log-intensity $\log(R+G+B)$ and compute its mean and its percentiles sampled every 2%. We also evaluate the same percentiles on two Gaussian-convolved versions of the photo ($\sigma = 10$ and $\sigma = 30$) to account for the tonal distributions at larger scales.

▷ *Scene brightness:* We hypothesize that scenes that are dark vs. bright in the real world might be adjusted differently. We evaluate the scene brightness as: $(\hat{Y} \times N^2)/(\Delta t \times ISO)$ where $\hat{Y}$ is the median intensity, N is the lens aperture number that is inversely proportional to the aperture radius, Δt is the exposure duration, and ISO is the sensor gain. This quantity is proportional to the light reaching the camera sensor and assumes that there is no filter attached.

▷ *Equalization curves:* Photographers tend to use the entire available intensity range. Histogram equalization is a coarse approximation of this strategy. We compute the corresponding curve, i.e., the cumulative distribution function (CDF) of the image intensities, and project it on the first 5 PCA components of the curve

▷ *Detail-weighted equalization curves:* Detailed regions often receive more attention. We represent this by weighting each pixel by the gradient magnitude, and then project the weighted CDF onto the first 5 PCA components of the curve. We estimate the gradients with Gaussian derivatives for $\sigma = 1$, $\sigma = 100$, and $\sigma = 200$ to account for details at different scales.

▷ *Highlight clipping:* Managing the amount of highlight that gets "clipped" is a key aspect of photo retouching. We compute the label values that clip the following fraction of the image: 1%, 2%, 3%, 5%, 10%, and 15%.

▷ *Spatial distributions:* The fraction of highlights, mid-tones, and shadows are key aspects discussed in the photography literature. However, their percentage alone does not tell the whole story, and it is important to also consider how a given tone range is spatially distributed. We split the intensity range in 10 intervals. For each of them, we fit a 2D spatial Gaussian to the corresponding pixels. The feature value is the area of the fitted Gaussian divided by the number of pixels in the given tone range. We also use the xy coordinates of the center of the Gaussian as a feature representing the coarse spatial distribution of tones.

▷ *Faces:* People are often the main subject of a photo and their adjustment has priority. We detect faces and compute the following features: intensity percentiles within facial regions (if none, we use the percentiles of the whole image), total area, mean xy location, and number of faces.

We also experimented with other features such as local histograms, color distributions, and scene descriptors but they did not improve the results in our experiments.

4.4.3 Error Metric

We use the L_2 metric in the CIE-Lab color space to evaluate the learning results because this space is perceptually uniform. The difference between white and black is 100, and distance of 2.3 corresponds to a just-noticeable-difference (JND) [64]. Since we focus on tonal balance, we measure the difference in luminance between the predicted output and the user-adjusted reference. We evaluate our learning methods by splitting our dataset into training on 80% dataset and testing on the remaining

20%.

4.5 Learning Automatic Adjustment

We consider two practical cases. First, we aim for reproducing the adjustment of a single photographer given a large collection of examples. In the second case, we seek to learn adjustments from a specific user from a small set of examples, assuming that we have access to a large collection of examples by another photographer. To validate our approach, we compare it to the recent method of Kang et al. [32] because it tackles similar issues and requires only minor changes to work on our dataset.

4.5.1 Predicting a User's Adjustment

In this scenario, we have a large dataset of examples from a single user and we learn to adjust images similarly to this photographer. This is useful for a camera or software company to train an automatic adjustment tool. We tested several regression algorithms: linear regression as a simple baseline, LASSO as a simple and still efficient technique [28], and Gaussian Processes Regression (GPR) as a powerful but computationally more expensive method [57]. LASSO performs a linear regression on a sparse subset of the input dimensions. We trained it using 5-fold cross-validation on the training set. GPR has been shown to have great abilities to learn complex relationships but is also significantly more expensive in terms of computation. To keep the running time reasonable, we trained it only on 2,500 randomly selected examples.

Comparison to Metric Learning For comparison, we implemented a variant of the method by Kang et al. [32] so that it uses our dataset and handles a single user. We used the user's adjustments instead of computer-generated data for learning the metric. We kept sensor placement unchanged, i.e., we select the images that maximize the mutual information with the user's adjustments. The nearest-neighbor step is also unaltered except that we transfer the tonal curve extracted from our data instead of Kang's parametric curve and white balance.

Results We selected Retoucher C for our evaluation because the high ranking in our user study. Using labels from Retoucher C we compared several options: the mean curve of the training set; the metric learning method using 25 sensors as recommended by the authors of [32]; least-squares regression (LSR); LASSO set to keep about 50 features; and GPR. The prediction accuracy is reported in Table 4.1. Regression techniques perform significantly better than other approaches. We also computed the leave-one-out performance of metric-learning method: 9.8, which means that it is limited independently of the number of sensors that we select. This is further confirmed in the next section.

mean	metric learning	LSR	LASSO	GPR
13.2	11.5	5.2	4.9	4.7

Table 4.1: LAB error of several methods (lower is better) when predicting a user's adjustment. For reference, not adjusting the photos at all produces an error of 16.3.

Figure 4-2 shows error CDFs for automatic image adjustment methods. In this figure, all methods were evaluated on the same test set of 2,500 photos. Our method was trained on 2,500 examples. The metric computation (a variant of [32]) used all 5,000 examples, but the nearest neighbor prediction was done only using 2,500 training examples. Commercial methods were not trained on our dataset and are shown for reference only.

Data versus Covariance GPR proceeds in two steps. During training, it optimizes the hyper-parameters of a covariance function so that it best explains the training set. At run time, it uses this covariance function to drive the combination of some of the training curves. To find out whether the performance of GPR comes from the covariance or from the training data, we did the following comparison. First, we trained the GPR covariance on the whole training set of 2,500 photos but used only small number n of example curves at run-time for prediction. We also trained the covariance with only n images and used the same n images for prediction, practically reducing the size of the training set. In the two tests, the run time data are the same, but in the first case, the covariance function comes from a rich training set while in the second case, it comes from a small set. Figure 4-4 shows that using the well-

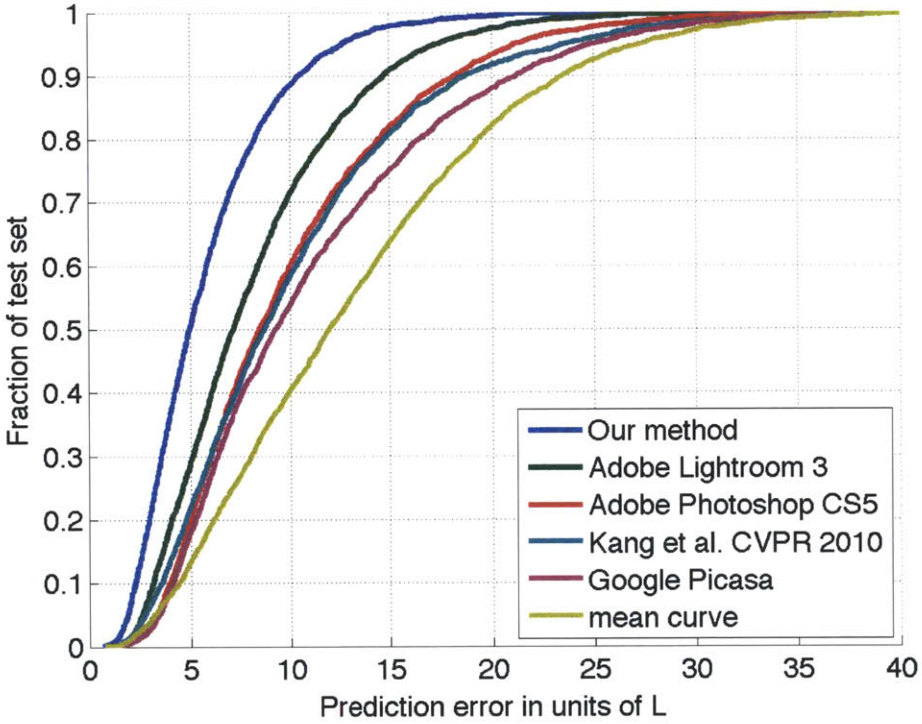
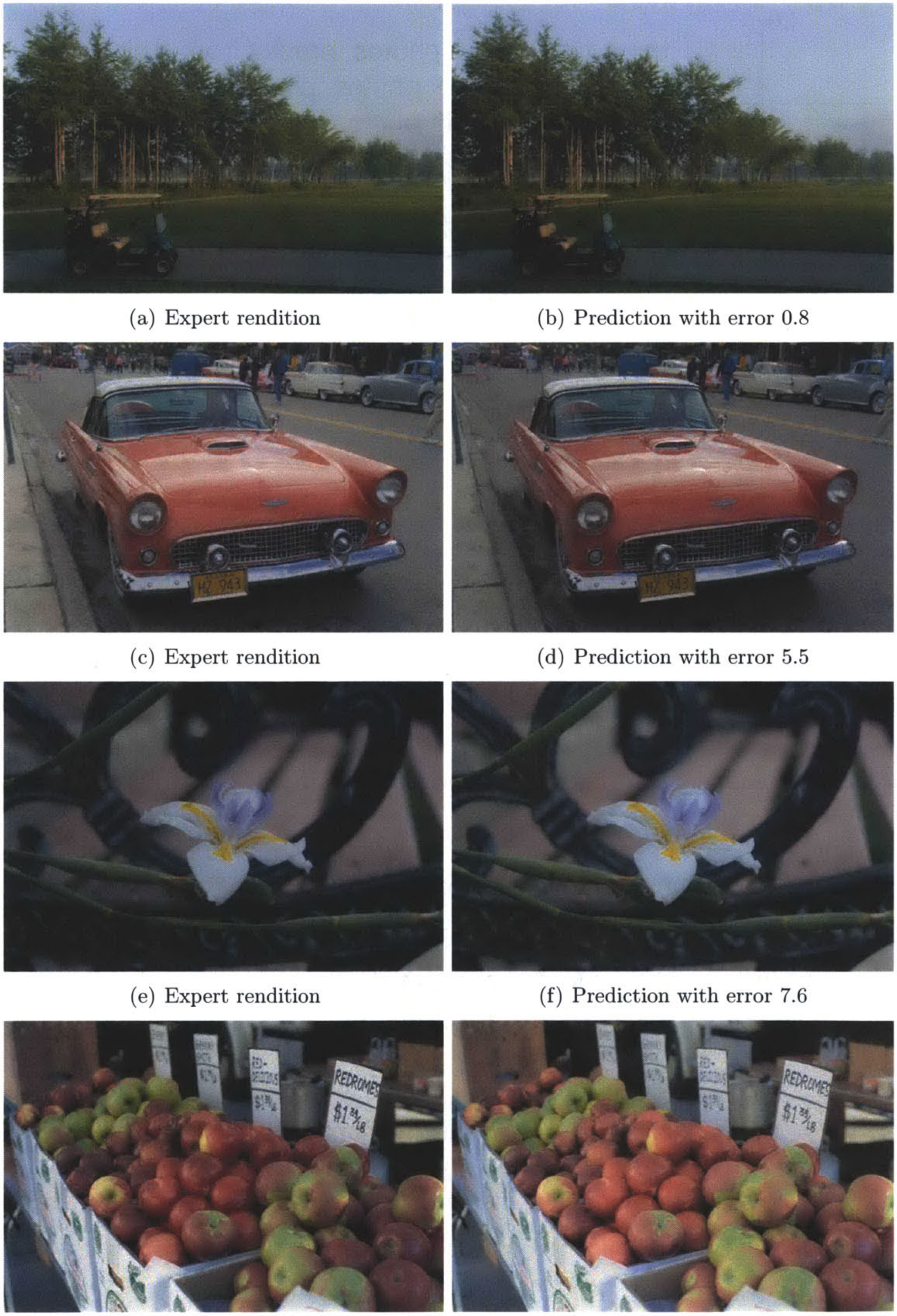


Figure 4-2: Error CDFs of automatic photo adjustment methods (higher is better). An error of 2.3 L units corresponds to 1 JND (just noticeable difference). For visual calibration see Figure 4-3. Lightroom, Photoshop, and Picasa were not trained on our dataset and are shown for reference only.



(a) Expert rendition

(b) Prediction with error 0.8

(c) Expert rendition

(d) Prediction with error 5.5

(e) Expert rendition

(f) Prediction with error 7.6

(g) Expert rendition

(h) Prediction with error 10.6

Figure 4-3: Sample prediction results for our method provided for visual calibration of error values in Figure 4-2.

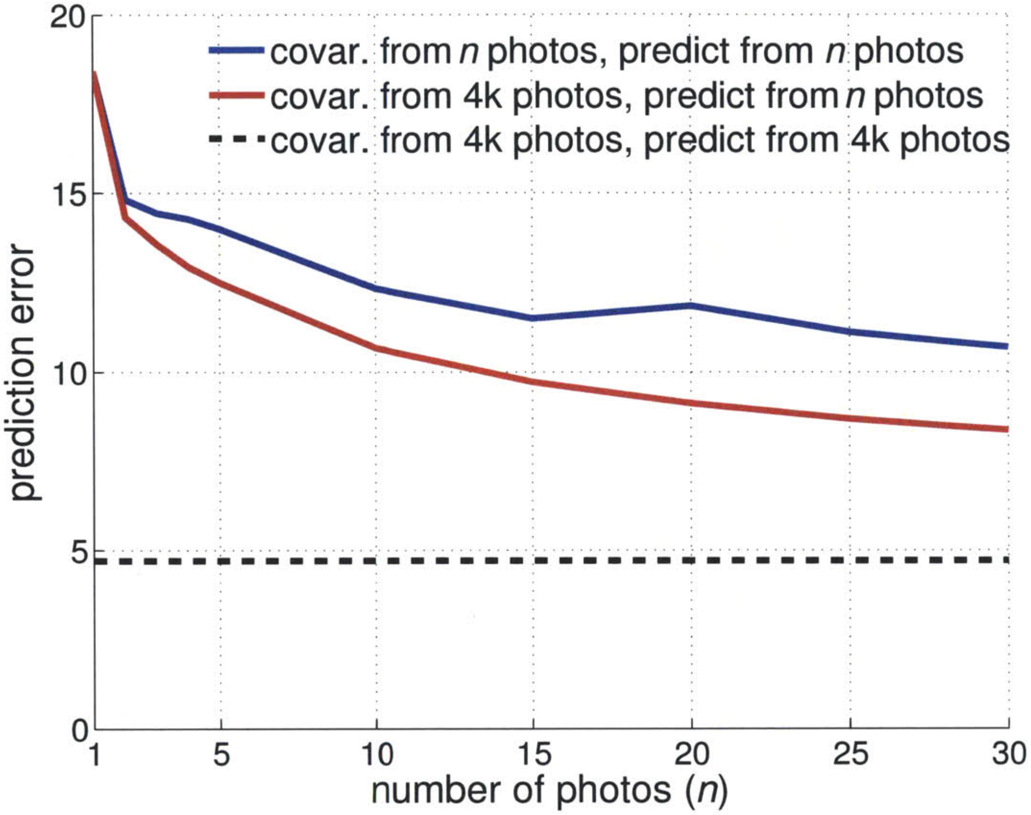
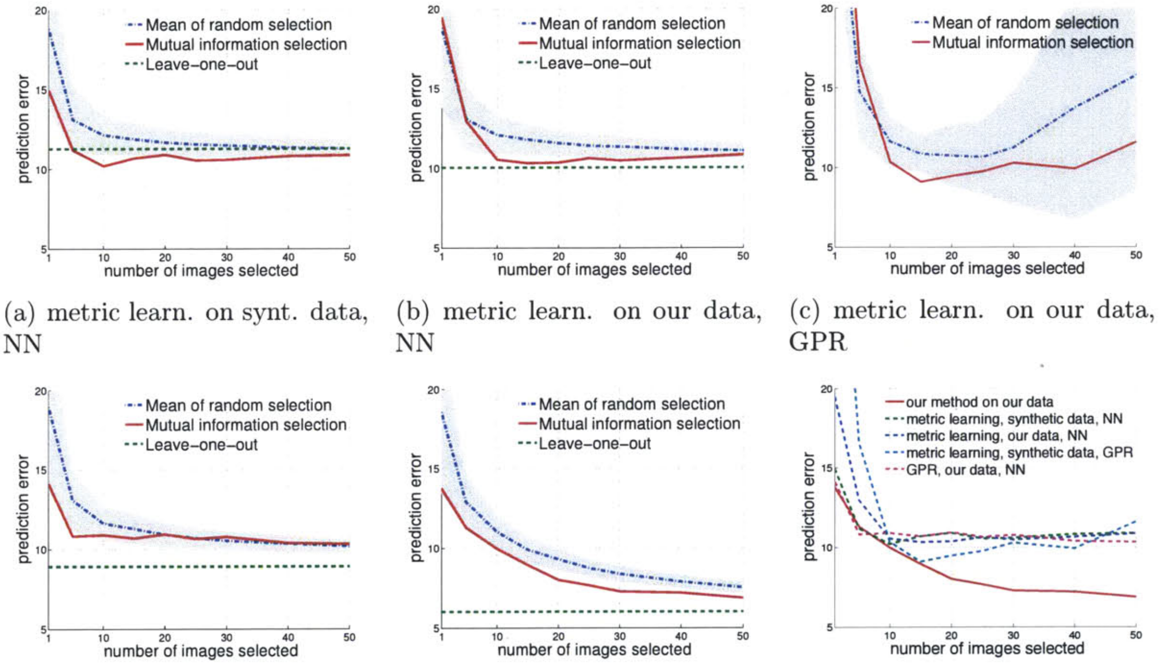


Figure 4-4: Using pre-trained covariance function improves the accuracy of prediction when only a few examples are available. In the above example directly learning from 30 images results in the same error as learning from 10 images and using pre-trained covariance.

trained covariance function yields significantly better prediction given the same small number of run-time data. This highlights the importance of the covariance function in the prediction process since it models the structure of the photograph space. We build upon this insight in the following sections.

4.5.2 Transferring a User's Adjustments

The technique described in the previous section is suitable for off-line training. However, adjusting 5,000 images requires several weeks of work, normal users cannot reasonably train this algorithm for their own style. In this section, we leverage the fact that we already have a large dataset $\mathcal{L}$ of 5000 images adjusted by the *reference* retoucher to enable learning from only a small set of examples $\mathcal{S}$ by a new



(a) metric learn. on synt. data, NN

(b) metric learn. on our data, NN

(c) metric learn. on our data, GPR

(d) GPR metric on our data, NN

(e) GPR metric (our data), GPR

(f) summary (lower is better)

our method

Figure 4-5: Performance of various options to predict Retoucher C's adjustments using a small set $\mathcal{S}$ of his examples and a large set $\mathcal{L}$ of examples, either synthetic or from Retoucher D. We report the prediction error in CIE-Lab units as function of the size of $\mathcal{S}$. We plot the accuracy the sensor-placement selection (in red), of a random selection (average in blue, up to one standard deviation in gray), and of a leave-one-out bound (in green). See text for detail.

photographer.

Experimental Setup

To evaluate our approach, we implemented the following algorithm. We run GPR on the large set $\mathcal{L}$ to compute a covariance function. Akin to Kang et al. [32], we use sensor placement [36] to select a small set $\mathcal{S}$ of images to be adjusted by the new photographer. For the covariance matrix needed to compute the mutual information, we use $\Sigma_{\mathcal{L}}$ (see [36] for detail). To predict the new photographer's adjustment on an unseen photo, we use the covariance function trained on the large set $\mathcal{L}$ to run GPR interpolation on the labels of the small set $\mathcal{S}$.

For comparison, we also implemented the method of Kang et al. We reproduced the automatic adjustment procedure that generates 4D vectors for each image of $\mathcal{L}$. We implemented the photo similarity functions that are proposed, and linearly combined them to approximate in a least-squares sense the L_2 distance on the 4D coefficient vectors. We ran sensor selection [36] to select $\mathcal{S}$ using the described covariance matrix. Given an unseen image, we search its nearest neighbor in $\mathcal{S}$ according to the learned metric, and apply its tone curve onto the new image. We also implemented variants to evaluate specific aspects. We trained the metric of Kang et al. on a photographer's curve instead of the original synthetic data. We also replaced nearest-neighbor search by GPR based on the covariance matrix used for sensor placement.

Results

Figure 4-5 reports the results for several options. For each scenario, we plot the accuracy as a function of the size of $\mathcal{S}$. In this figure, we compare a random selection with sensor placement selection; we also indicate the leave-one-out bound for reference. For all options but ours, the accuracy quickly reaches 10 and then plateaus (a, b, and d) or degrades (c). Using our dataset instead of synthetic data improves the leave-one-out performance (a vs. b). A similar improvement happens when using the metric learned with GPR instead of metric learning using a least-squares fit [32] (a vs. d). Using GPR with a covariance matrix optimized with metric learning yields poor results when $\mathcal{S}$ grows (c). In all cases, sensor placement using mutual information produces better results on average than a random selection. Although comparisons with Picasa have a limited scope because Picasa is not trained on any data, our results are consistent with the findings of Kang et al.: the difference between Picasa and their method using 25 images is below 1 (11.4 vs 10.6), which is marginal. In comparison, our approach performs significantly better than Picasa with an improvement of almost 4 (11.4 vs 7.6). Most importantly, our tests show that using GPR with our dataset yields results equivalent to other options up to 10 images, and performs significantly better than them for larger sizes of $\mathcal{S}$, producing an accuracy of about 7.6 with 25 images, instead of about 10.6 for the other techniques, i.e. an improvement on the order of 30% (f).

4.5.3 Difference Learning

The method described in the previous section reduces the number of training examples to a few tens. However, new users may prefer to train the system using their own photos instead adjusting a predefined set of example to train the system. In this section, we explore the scenario where the preferences are learned on-the-fly using adjustments on random pictures for training. Instead of learning the adjustment of the new photographer directly, we propose to learn the difference between the reference photographer and the new adjustment. For a new photo, we first predict the reference adjustment and then predict its difference with the new photographer's version. Our experiments described in the results section show the benefits of difference learning.

Our Approach We first trained GPR on the large training set $\mathcal{L}$. Then, we predict the reference curves for each photo of the small training set $\mathcal{S}$ and compute their difference with the curves of the new photographers. This gives a series of *adjustment offsets*. Given a new photo, we first predict the reference adjustment $\mathbf{r}$ using the covariance trained on $\mathcal{L}$ and the adjustments in $\mathcal{L}$. We also predict an adjustment offset $\mathbf{o}$ using the $\mathcal{L}$ covariance and the offsets computed on $\mathcal{S}$, and add it to the reference adjustment $\mathbf{r}$. Finally, we apply this combined adjustment $\mathbf{r} + \mathbf{o}$ to the photo.

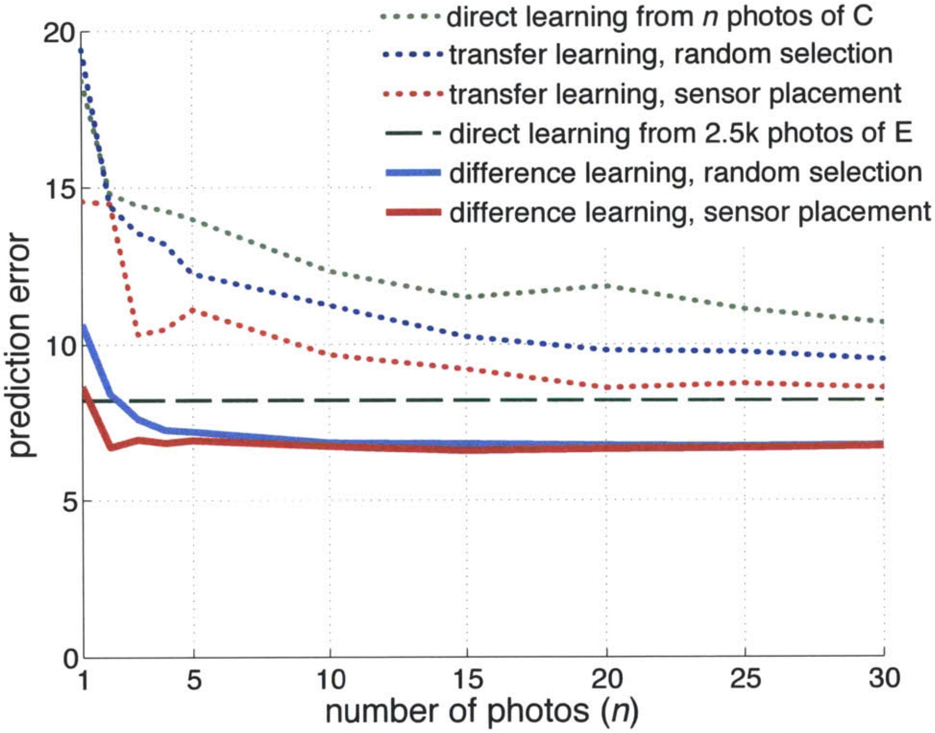


Figure 4-6: Several strategies to predict Retoucher C's adjustments from only n of his or her photos. We can directly train GPR on these examples only but the predictions are poor (first plot from the top). To improve the results, we can use transfer learning and precompute the GPR covariance function using a large dataset by Retoucher E (§ 4.5.2). This significantly improves the result (second plot) and if the we can select which photos of Retoucher C are available, sensor placement further improves the result (third plot). However, in this case, C and E produce adjustments similar enough so that applying GPR directly on E's photos without using any data from C better predicts C's adjustment than the previously mentioned options (fourth plot). This means that if our system was trained off-line with E's photos, the previous options would not allow C to get predictions closer to his or her preferences. In comparison, learning differences between C and E (§ 4.5.3) yields better results. If the photos of C are random, the improvement starts when 3 or more of C's photos are available (fifth plot). If we can select the photos with sensor placement, two example photos are sufficient to see an improvement (bottom plot).

Results Figure 4-6 shows that our approach using Retoucher E for $\mathcal{L}$ and Retoucher C for $\mathcal{S}$. In this case, Retoucher E and Retoucher C adjust photos similarly and transfer learning as described in the previous section (§ 4.5.2) does not predict C's adjustments better than using GPR directly (§ 4.5.1) using only E's photos. That is, the curves predicted using of E's data only are already a good approximation of C's adjustments, and transfer learning is unable to improve over this baseline. In comparison, our difference learning approach yields better predictions than this baseline, even if the available photos of C are randomly selected. On average, as few as 3 examples photos are enough to produce better results. Although the crossing point may depend on the considered photographers, we believe that the ability of difference learning to learn preferences from only a few examples and its accuracy make it highly practical.

4.6 Conclusions and lessons learned

We have built a high-quality reference dataset for automatic photo adjustment, which addresses a major need and will enable new research on the learning of photographic adjustment. In particular, we include data from five different users to enable not only training but also comprehensive validation. We have demonstrated that our photo collection is a powerful tool to learn photo adjustment and study various aspects of it. We have shown that with high-quality data, supervised learning can perform better than existing techniques based on simple rules or synthetic training sets. We have also found that regression with our new set of image features outperforms previous methods. We have performed transfer learning and shown that our dataset enables better selection through sensor placement. We have also shown that difference learning enables preference learning in a on-the-fly context where the training photos are not predetermined. In addition to enabling these applications, our dataset proves invaluable for validation.

Even though our dataset of image adjustments is unprecedented in size and quality, it has a number of limitations that could be addressed by new dataset and methods.

First major limitation is the relative uniformity of styles within the dataset. Specifically, all photographers were instructed to produce postcard-like image renditions. In addition to that photographers were limited to using only a subset of adjustment tools. This resulted in a dataset cleaner, more consistent dataset for a single style. However, in practice, photographers use a diverse set of image processing tools to produce a large variety of styles. Having experts adjust thousands of photographs is not only expensive but also error prone. Many photographers may have unique and interesting styles, but not all of them are capable of consistently retouching thousands of photographs. Some of these problems are addressed by the method developed in the following chapter by decoupling artistic and technical expertise in the context of image stylization.

Chapter 5

Genetic filters: decoupling taste and image processing expertise

5.1 Abstract

Transformations applied to photographs range from basic exposure and white balance corrections to aged film look to extreme stylizations such as painterly renditions. Constructing such photographic looks and renditions requires a careful combination of low-level elements such as tone curves on individual color channels and band-pass filters as well as a great deal of domain-specific know-how and experience. While successful techniques have been designed this way, this is a tedious and time-consuming process that requires both technical and artistic expertise. In this work, we propose a stochastic approach to image filter discovery that decouples the technical and artistic aspects of image filter design. We introduce a functional grammar that is able to describe a wide variety of options, including typical photographic operations and standard NPR filters such as oriented smoothing and soft thresholding. To explore the vast space of effects offered by our grammar, we propose an approach that uses crowd-sourcing and genetic algorithms. We rely on Amazon Mechanical Turk to rate the outputs produced by several filters and use the result as the fitness function driving a genetic evolution. By combining the most successful filters and introducing random variations, the genetic algorithm evolves new filters of increasing quality. We

demonstrate that after a few generations, this process yields a great variety of effects and unveils numerous complex filters that would have been challenging to discover otherwise. We also build an interface for interactive filter discovery and conduct a user-study to evaluate its effectiveness.

5.2 Introduction

Image filtering applications (such as Instagram) are very popular among consumers. These applications usually offer users a set of pre-made image filters. The standard approach to designing these effect filters involves an expert assembling a pipeline out of low-level operations such as applying tone curves, convolutions, color transforms, and others. Obtaining a desired look requires advanced skills and much care to build and tune. Since few experts are able to do so and a given filter is only valid for a specific look or a small set of effects, the design process is long, painstaking, and repetitive. Designing a filter for a given effect requires image processing expertise and time, but creating a variety of interesting filters also requires imagination.

In this chapter, we introduce a new approach to designing image filters that allows even non-experts to create a variety of novel image filters. We use a genetic algorithm to let users explore and search the space of possible image filters. We define the space of image filters with a grammar made of low-level image operations and rules for combining them. In order to get a good coverage, our grammar contains a large number of basic blocks used in the literature, including tone curves, color transforms, image decomposition and re-composition, and line integrals. This ensures that classical filters such as white balance, exposure/contrast corrections, sharpening/blurring, bilateral filtering, and painterly abstraction can be expressed with our grammar.

In the past, research has focused on designing and tuning specific filters such as painterly renditions and sketches. In contrast, we seek to explore a much larger space of filters broadly and systematically. For this, we leverage a large group of human subjects via Amazon Mechanical Turk. We generate filters with our grammar, apply these filters to images, and have users score these images. We aggregate scores from multiple users and drive a genetic algorithm with the resulting score. This algorithm generates new filters by mutating and combining (parts of) filters popular among users. We demonstrate that this approach generates a set of nontrivial and novel filters that would have been difficult to discover by other means. Moreover, by design, these filters are most interesting from the users' standpoint. Finally, we allow users to have greater artistic control over results by a user interface for interactive discovery of filters.

5.2.1 Contributions

In this chapter, we introduce the following contributions:

- *Grammar for image filters.* We describe a grammar-based representation for image filters. This grammar consists of common image processing operations. We rely on data types and high-order functions to make this grammar compact, modular, and expressive. We demonstrate that a number of interesting filters can be generated by randomly sampling this grammar.

- *Crowd-sourced filter discovery.* We develop a system for fully automatic filter discovery. This system is based on a combination of genetic programming and crowd-sourcing. We show that this approach allows to discover novel image filters. We also evaluate the effectiveness of this approach. We attempt to address the question of filter and content interdependence.

- *Interactive filter discovery.* In addition to the fully automated filter discovery, we develop an interface that enables individual users to explore the space of images filters in a directed manner. We run an informal user study using this interface and share its results.

The common goal of the above contributions is to make novel filter design easier and to decouple the creative selection process from the low-level image filter design. Each of the above contributions is discussed in detail in the following sections of the chapter.

5.3 A grammar of image filters

We have two requirements for the image filter representation: expressiveness and ease of generating visually interesting filters. Expressiveness is important because the space of possible image filters is vast and our representation should span it. Ability to easily generate visually interesting filters is crucial because we would like to avoid wasting resources on evaluating uninteresting filters. These two requirements are at odds with each other: the first one argues for a low-level, Turing-complete image processing language, while the second argues for using a library of high-level primitives. In this chapter we resolve this conflict by choosing an intermediate-level representation based on a typed grammar of image processing operations. The presence of low-level and high-level operations and rules for their composition enables high expressivity and ensures that a significant portion of generated filters are visually interesting.

5.3.1 Operations and data types

Filters in our system are represented as recursive typed symbolic expressions. Effectively, a filter is a tree of expressions, where each expression is an operation that takes zero or more arguments. Each of these arguments may itself be an expression. The topmost expression, the root of the tree, produces the final result. To support parameters in our image operations we introduce types into our grammar. For example, Gaussian blur operator takes the size of the blur as a scalar parameter. In addition to ubiquitous image and scalar types, we introduce the following types: channel, color matrix, image tensor. All of these types are produced and consumed by different image operations. The grammar production rules map the output type to operations that generate a result of this type. These operations, in turn, require zero or more arguments of specified types. Expressions generated using these rules are guaranteed to be valid and executable image filters. A simple grammar may consist of two operations such as image addition and blur:

```
image Blend(image A, image B)
image GaussianBlur(image I, float radius)
```

5.3.2 Operation templates

Many processing operations apply to different data types. For example, addition (or blending) can be performed on images, channels, or even matrices. To avoid managing separate copies of operations for each type, we added a simple templating layer that generates grammar rules automatically for operations that support multiple data types.

5.3.3 High-order operations

We used the approach described above in an early prototype of our system, but we noticed that randomly generated filters never included effects that required matching decomposition and re-composition. For example, sharpening effect can be thought of as a blending of the input image and amplified high-frequencies of the same input image. Even though blending and extraction of amplified high frequencies (blurring, subtraction, and multiplication) were all parts of the grammar the probability of generating an expression that performed sharpening was very low. This is due to the fact that inputs to the image addition operation have to match: one must be an image and the other must be high frequencies extracted from the same image.

To address this problem we introduced high-order operations. High-order operations take other functions of images as inputs or produce them as outputs. For example, a high-order blur operation takes a scalar radius parameter and returns a function that blurs the image by a specified amount:

```
image↦image HOGaussianBlur(float radius)
```

Many imaging operations, including sharpening and soft-focus, can be described as decomposition of the input into two parts, applying different operations to each part, and recomposing these parts together. Using high-order operations this pattern can be implemented as a decomposition-recomposition operation that takes three functions: a decomposition filter, and two filters for processing the two parts of decomposition. In the case of sharpening, the decomposition function is a small radius blur, one of the transformation functions is amplification and the other one is an identity transformation. The following is an implementation of sharpening using high-order operations:

```
(HODecompRecomp (HOGaussianBlur 11.5) (HOIdentity) (HOMultiply 2.0))
```

Such a small expression is much more likely to be generated than the expression based only on the grammar of low-order operations.

Another benefit of using high-order operations is that function composition – think filter composition – is naturally expressed in our grammar:

```
image↦image HOFuncComp (image↦image A, image↦image B).
```

Function composition of two high-order functions $A(x)$ and $B(x)$ returns a function $F(x) = B(A(x))$

5.3.4 Summary of operations

Our objective is to construct a grammar of imaging operations that is both expressive and is likely to generate interesting renditions when sampled. One way to achieve the latter objective would have been to include well-known effect filters as atomic operations for the grammar. However, since we are less interested in reproducing well-known filters and more interested in producing novel filters, we opt for a different approach. Specifically, we found it useful to decompose well-known effects such as cross-processing, sharpening, soft-focus, tonal range compression, painterly rendering, sketching, into atomic operations. This approach not only ensures that it is possible to reproduce each of these effects using our grammar, but also enables novel combinations of filters to be created. We found that when combined with filter composition and basic image blending operations this approach yields a large variety of novel filters. Below is a list of operations grouped by semantics.

Arithmetic operations These operations include negation, addition, multiplication, and exponentiation. When applied to pairs of images these operators correspond to different blending modes.

Convolutions These operations include the Gaussian blur, the bilateral filter, and convolution with a specified 3x3 kernel. When applied to images these operations produce various types of blurring.

Image adjustments Tone-curve and white-balance correction.

Image decomposition/re-composition Operations in this group decompose images (or image tensors) into components, allow each component to be filtered independently, and recompose the result. These operations include decomposition into channels, and into luminance and chrominance.

Filter composition and effect modulation Filter composition combines the effect of two filters by applying these filters sequentially. The effect modulation operation controls the strength of a given filter by linearly blending the input image with the output of the filter. The blending weight in the effect modulator varies between -1 and 1 which allows application of the negative version of the effect.

High-level operations These operations include flow-bilateral filter [31] and gradient/tangent line integrals [78], each of which integrate along a specified tensor field, image gradient, and image tensor computation, soft-thresholding, and a noise generating function. These operations are often used to generate NPR effects such as painterly rendering and sketches.

Space change operations These operations enable filtering images in a different space. For example, one of these operations undoes gamma correction, applies a provided filter, and reapplies gamma correction, thus enabling filtering in linear space. Other operations in this group allow taking a logarithm, exponentiating the image, or applying an invertible color transform.

Most of the above operations can be applied not only to pairs of images, but also to image channels and matrices. Using the templating mechanism described in section 5.3.2 we automatically generate the corresponding grammar rules.

5.3.5 Sampling the grammar

The grammar defined in the previous section covers a large set of possible filters. We can directly sample the space of image filters by generating valid expressions using our grammar. Figure 5-1 shows results of such random sampling. It is very encouraging to see a number of novel and promising filters generated. This leads us to believe that our grammar is sufficiently flexible and expressive.

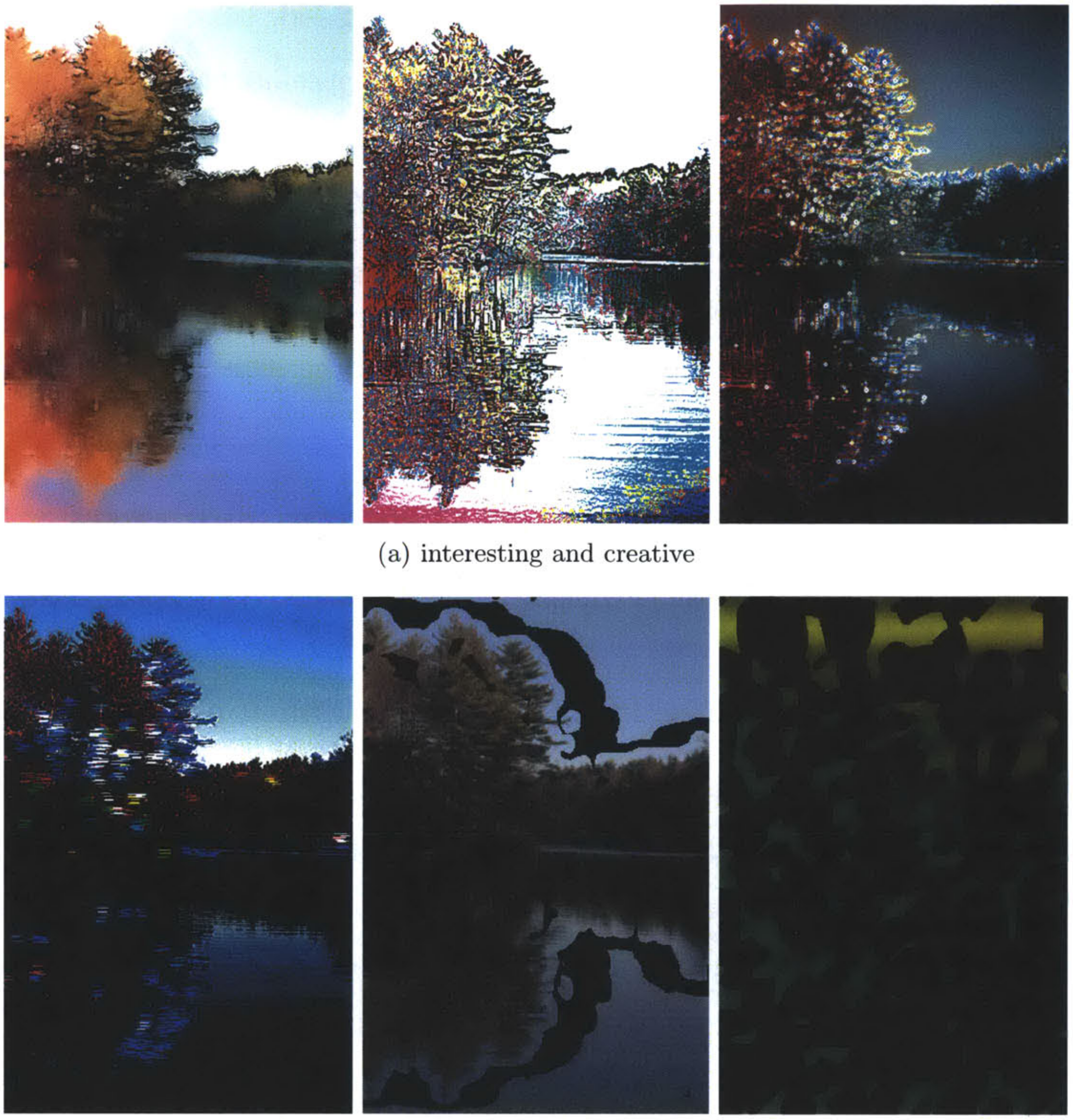
5.4 Crowd-sourcing filter discovery

Random sampling of the grammar tends to produce a large number of filters that result in unattractive or, even, unrecognizable images. In order to discover novel and interesting image filters we rely on genetic programming and human input.

5.4.1 Genetic programming for image filters

Genetic programming is an optimization algorithm inspired by biological evolution (see e.g. [34]). Possible solutions are referred to as *individuals.* Each individual is characterized by a compact code that can be thought of as their DNA. Individuals are evaluated using a *fitness function* that decides if a given individual dies or if it gets to *reproduce* and to pass on its DNA to the next *generation.* Genetic algorithm repeats this selection and reproduction process until an individual with a desired level of fitness is produced. This individual represents the solution to the optimization problem.

Reproduction in classical genetic programming is usually accomplished either by *mutation* or *cross-over* operations. Mutation alters an individual's DNA in hopes of improving the fitness. In the context of image filters, mutation may change the



(a) interesting and creative

(b) unattractive and unrecognizable

Figure 5-1: Grammar-based representation of image filters enables random sampling. Our proposed grammar is expressive and flexible enough to produce many novel and promising filters (a) by randomly sampling expression in the grammar. However, a large portion of randomly generated filters are uninteresting or, even, unrecognizable (b). We use genetic algorithms and crowd-sourcing to further evolve and refine image filters.

structure of the filter or alter parameter values for some operation. Cross-over combines portions of DNA from two fit individuals to produce an individual with potentially even higher fitness. Effectively, cross-over copies portions of the image filtering pipeline from one successful filter and pastes it into filter. We augment the set of standard genetic programming reproduction operations with *composition*. Intuitively, combining successful filters may yield an even more appealing filter.

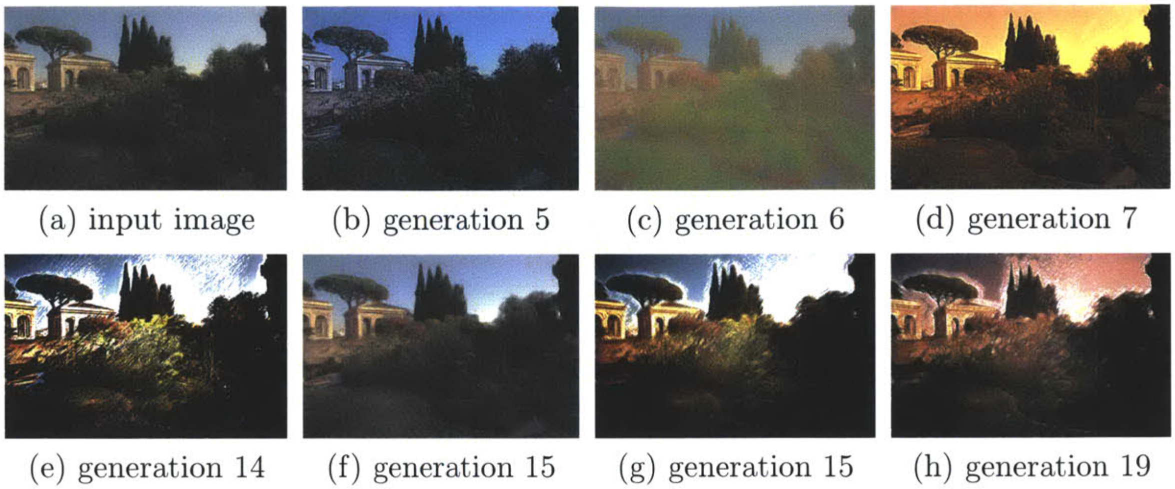
One of the unique characteristics of genetic programming is its ability to solve problems with a complicated structure. For example, in the case of analog circuit design, genetic programming was able to successfully discover the topology and the component parameters for a given specification [35]. Image filter design problem is akin to analog circuit design in that we aim to find the sequence of operations as well as parameters for these operations that produce an attractive filter. The main difference between image filter design and circuit design is the fitness function. While a circuit can be checked against a specification automatically, the visual appeal of a filter requires human aesthetic judgment.

5.4.2 Crowd-sourcing filter quality evaluation

Our grammar-based image filter representation (Section 5.3) and genetic programming (Section 5.4.1) provide an approach for image filter discovery and refinement. This approach relies on having a fitness function to measure quality of individual image filters. We rely on human judgment to evaluate the quality of image filters.

As noted in many other studies relying on Amazon Mechanical Turk [6,24,41,79], as many as 30% of workers on Mechanical Turk try to game the system by providing random inputs. Our experience confirms this observation. Traditionally, workers not paying attention to the task are detected by means of control questions. We found that the number of workers trying to game the system is high even among workers with high reputation score. We believe that this occurs because many requesters do no bother rejecting incorrect answers thus rendering reputation system unreliable.

We use image filters that produce completely unrecognizable output (flat black or gray image) as control questions. In our instructions for workers we provide example images (much like those in Figure 5-1) and a scoring scale from 6 (interesting/creative) to 0 (unrecognizable). We only accept assignments where workers give control images very low scores (per our instructions). This means that workers who are not paying attention to the instructions do not receive payment. This policy resulted in reduction


(a) input image (b) generation 5 (c) generation 6 (d) generation 7

(e) generation 14 (f) generation 15 (g) generation 15 (h) generation 19

Figure 5-2: Automatic novel filter discovery using Amazon Mechanical Turk. This figure shows the input image (a) and image filters that score highly from different generations of evolution. In just a few generations, the system discovers a number of interesting filters such as a color change and sharpen filter (b), an NPR filter (c), and a color change and add diagonal lines filter (d). With more generations, it explores other depiction effects such as swirls (f) and continues to refine a high-contrast/high saturation NPR effect (e,g,h).

of grossly incorrect answers from 30% to 2%. In other words, workers attempting to cheat stopped working on our tasks.

After filtering out grossly incorrect answers, we average scores from multiple users. These averaged scores serve as fitness values for different filters. Once scores for all images are collected, our system creates the next generation of filters using reproduction operations described in section 5.4.1 and posts a new task to Amazon Mechanical Turk. Since we do not have an *a priori* fitness criteria we repeat this process for a few generations.

5.4.3 Evolving image filters

We evolve 50 different filters per generation. The first generation contains filters randomly sampled from the grammar. Each generation of filters is scored by Mechanical Turk workers with 4 scores per image. The average score is used as the fitness value of the individual filter in reproduction process.

The results of this automatic image filter evolution are shown in Figure 5-2. After

generation	0	5	15	20
median score	3.0	3.5	3.3	3.58
75th percentile	4.0	4.2	4.0	4.3

Table 5.1: Evolution of scores. Even though image filters become more interesting and more diverse with evolution, absolute fitness scores do not seem to increase. We believe this is because quality judgments represented by these scores are relative rather than absolute.

generation	0	20
mean score	3.55	4.06
standard deviation	0.13	0.14
mean rank (lower is better)	12.6	6.40

Table 5.2: Results of comparing top filters from early and late generations. The large difference in mean scores and ranks supports the hypothesis that people inadvertently make relative judgments when evaluating image filters in our system.

only a few generations this system discovers a number of interesting filters such as sharpen and color change effects (b, d) and painterly effects (c, f). The system also continuously evolves and refines a high-contrast NPR effect that is very popular among the workers.

Even though the images in the later generations are qualitatively more interesting and diverse than in the earlier generations, average scores (shown in Table 5.1) do not reflect this trend. We believe this is because the quality judgments with in each generation are relative (and this do not represent absolute quality).

To verify this hypothesis, we selected top 10 filters from the generation 0 and 20 and ran a separate experiment. In this experiment each image received 12 votes. The results are shown in Table 5.2. The mean score improved with between the first and last generations in this comparison. The difference in average ranks between the two groups is even more apparent. These results support the that hypothesis quality judgments made by people in our experiments are relative rather than absolute.

The relative nature of human judgments does not hinder the operation of the genetic programming, but it does have implications for the interpretation of results. Specifically, absolute scores should not be compared across generations. The individual with the highest absolute score across generation may not be the most interesting

filter group	portrait	landscape
mean score	3.43	3.88
standard deviation	0.24	0.15
mean rank (lower is better)	8.43	4.57

Table 5.3: Scoring separately trained filters on a photograph of a landscape. Filters trained on landscape photos perform better on landscape photos.

filter group	portrait	landscape
mean score	3.44	3.53
standard deviation	0.17	0.19
mean rank (lower is better)	6.44	6.56

Table 5.4: Scoring separately trained filters on a portrait. There is no significant difference in scores between the filter groups when applied to a portrait image.

or novel in the absolute sense.

5.4.4 Filter and content interdependence

The semantic content of a photograph often impacts how the photograph is retouched. In order to determine whether this assertion applies to image filters, we compare filters evolved on different images. Specifically, we are interested in learning if filters that are evolved on a landscape do as well when applied to portraits and visa versa. After evolving portrait and landscape filters independently we select the top 7 filters from each dataset. We ran two comparison experiments: (1) a landscape is adjusted using each of the 14 filters, (2) a portrait is adjusted using each of the 14 filters.

The results of these experiments are presented in Table 5.3 and Table 5.4. According to these results, filters evolved (or trained) on a landscape image tend to perform better on other landscape images than filters trained on a portrait images. One could expect images trained on a portrait would perform better on portraits, but our results in Table 5.4 do not support this hypothesis. Visual inspection of results suggests a possible explanation. The landscape-trained filters tend to be more aggressive, while portrait-trained filters tend to be more conservative. When users see the effects of these filters on a portrait, portrait-trained filters look relatively uninteresting.

5.5 Interactive filter discovery

We also designed a user interface for image filter discovery. The crowd-sourced and interactive interfaces have complementary advantages: crowd-sourcing allows for continuous or mostly automated creation of filters, while the interactive user interface allows for more creative and artistic control.

In this interface, we allow a user to interactively evolve an image filter based on her personal preferences. In the spirit of design galleries [47], we present a number of variations of the current image filter (Figure 5-3). The user can select a variation which becomes the newly selected image filter. Repeatedly selecting the most interesting variation allows user to evolve a novel image filter.

The interactive interface works by genetically sampling filters in the same manner as before, but the parent generation consists of only the single image selected by the user. New individuals are generated by sampling mutated and random filters until the resulting image is no more than a threshold average distance from the currently selected filter in L*a*b* space (in practice we use a threshold of 0.5).

We ran an informal study where we asked users to experiment with our system, explore variations on their own images or ones we provided, and exercise their creativity. Example results are presented in Figure 5-4. Typical evolution runs had between 4 and 20 generations and took just a few minutes of user time.

The feedback from the users indicated that exploration if fun and addictive, and that the interface was intuitive, and, in some cases, that the system "knew" what they were looking for. Users also gave suggestions for improvement, including that it would be nice to have more variety of filters at the beginning, more fine-tuned subtle changes at the end, and some users wanted fewer choices (we presented 20 variations of each image). We conclude that this interface offers an intuitive method for casual users to discover new image filters according to their own taste.

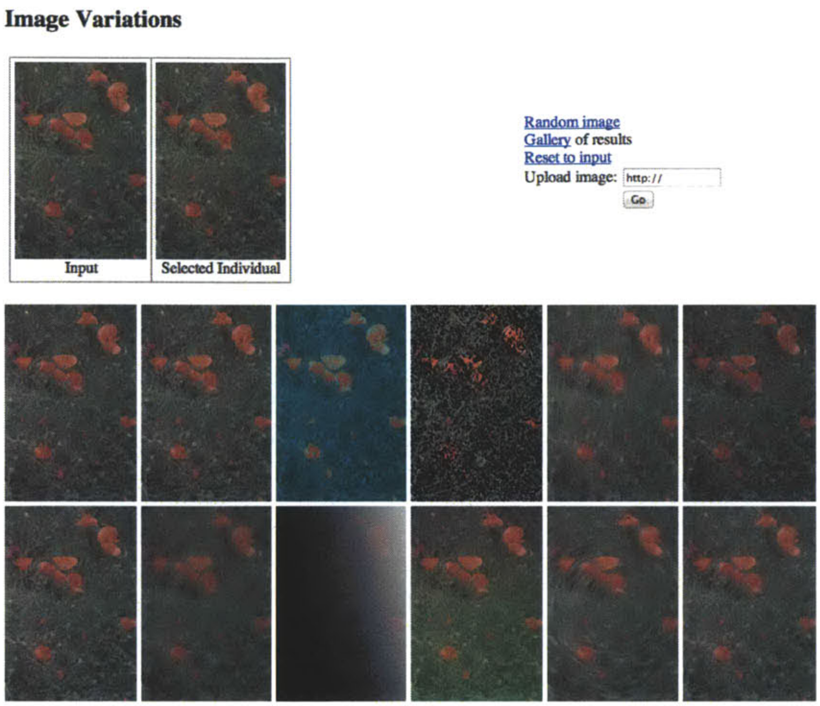


Figure 5-3: Directed evolution interface. Above is the input image and currently selected filter. Below, the user is presented with variations on the current filter. Selecting one filter causes it to become the genetic parent for the next generation of filters. By repeatedly selecting interesting images the user can evolve interesting filters interactively.

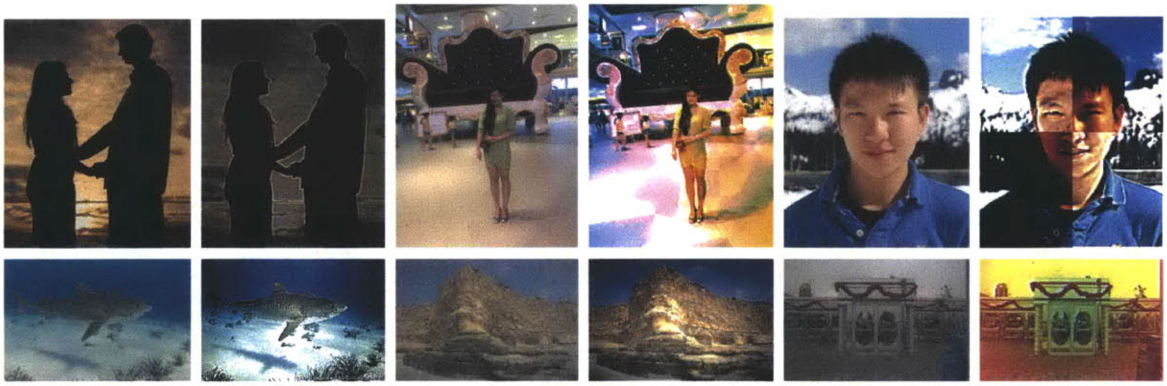


Figure 5-4: Results of directed user exploration. Top row (from left): users produced a filter that outlines a couple in a sunset image, a festive filter that adds colorful regions to a whimsical picture, and a filter that divides a portrait into four quadrants and adds contrast to each. Bottom row (from left): an NPR filter that amplifies local contrast and edges, a contrast and vignetting filter, and a "flower power" filter.

5.6 Off-line filter discovery interface

The use of grammar-based representation for filters enables fast and easy sampling the space of image transformations. However, small changes in the filter tree structure can result in large changes in the final output. The converse is also often true, because some sub-branches may have little or no effect (e.g.their are multiplied by small constant). These effects make it difficult to compare filters without applying them to images first.

Pre-generating a large number of filters and applying them to images allows for direct comparison of filters in the image space. This presents an opportunity for directed filter discovery. Instead of searching for filters by mutating the structure of the filter tree it is now possible to navigate in the image space directly in a more continuous manner.

Even if all filters are rendered into images off-line finding interesting filters in this collection of images is challenging. To address this challenge we designed an interface for navigating such image collections and conducted a user study that evaluated its effectiveness.

5.6.1 Creative search and exploration task

Before we consider the interface design, we need to consider how we are going to evaluate it. Recall that the main motivation automatic filter generation is to enable users with acute sense of taste but no image processing expertise to discover interesting filters. In other words, users of our system may have only have a vague idea for what constitutes an attractive filter. Our interface should help users define and refine their vision.

It is temping to view this task as a search task: a user could simply be asked to find a specific image in the collection. However, this view leaves no room for creativity and interpretation. Since we are interested in enabling the discovery of *novel* filters, users cannot know exactly what the final image would look like. We address this problem by asking users to search for a filter applied to a different image. This analogy-based method forces the user to extract and compare abstract properties of a given filter. We believe that trying to match these abstract properties better simulates having a vague idea for a filter in mind than directly searching for a specific image.

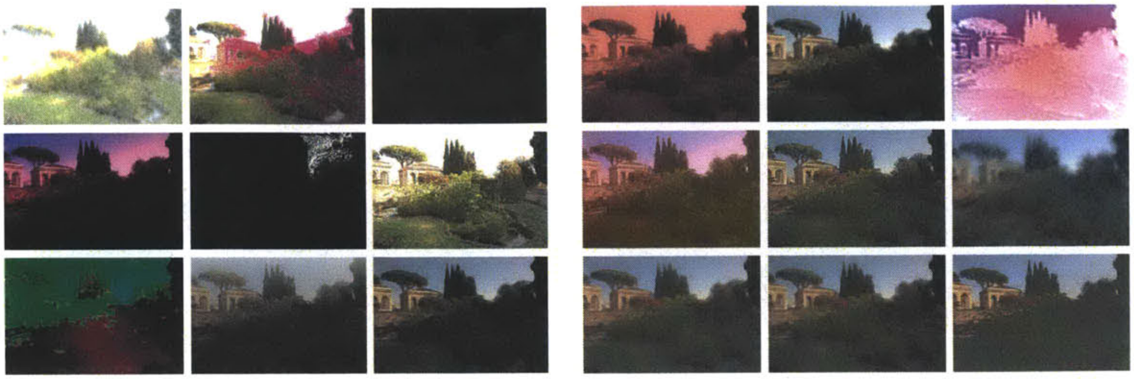
5.6.2 Features and interface design

Since the number of variations in the structure and parameters of image filters is very large, we take the design galleries [47] approach. The key insight of this approach is to hide the complexity of the underlying representation and show a set of examples instead. The challenge, however, is in choosing a set of images at every step such that users can converge on filter with desired properties quickly. This requires having a perceptually accurate image similarity metric.

Image similarity metrics have been an active area of research: Visual difference predictor [44,45] and SSIM [76] methods are examples of commonly used image similarity metrics. Unfortunately, classical image similarity metrics are expensive to compute. Computing them in real-time for thousands of images is not feasible, while pre-computing requires $O(N^2)$ time and space, which is impractical for large N in our case.

Instead of using expensive pairwise image metrics our system relies on compact image features. The choice of these features is informed by our experiments with real users. We noticed that users look for images by independently matching simple properties such as contrast, white balance, and more complex ones such as image "smoothness". For example, once a user found a filter with appropriate "smoothness", the user attempts to find a smooth filter that also has the right color. This user behavior prompted us to select similar images using three different features: luminance, gradients, and tiny images [74].

Luminance Luminance feature is computed as percentiles of the L channel. These answer the question "How bright is N-th brightest pixel in the image?" We found that percentiles robustly capture the notion of global image brightness and contrast.



(a) Random selection

(b) Feature-guided selection

Figure 5-5: Sample view of images interfaces. Random selection (a) provides a lot of diversity, but only a few of the shown are potentially interesting. Feature-guided selection (b) shows less variety, but most variation are reasonable.

Gradients To estimate perceived image roughness we use a histogram of gradient magnitudes of L channel. These, again, are represented as percentiles.

Tiny images We have attempted to capture the notion of color similarity between images using color histograms, but this resulted counter-intuitive matches. We found that when users are looking for images with similar colors they assume that colors stay in the same place in the image. Consequently, we switched away from using 2D color histograms and started using 5 × 5 color images instead.

The process of exploration is modeled after the notion of gradient descent. The filter chosen by the user in the previous round is used as the starting point (i.e. best-so-far filter) for local exploration. The system finds a few of nearest neighbors according to each feature *independently* and presents them to the user along with the best-so-far filter. In our experiments we found that presenting the best-so-far filter is crucial: all of the new alternatives may easily be less interesting than the previous best. This happens because there are many more dimensions of variation of images filters than the number of alternatives we present to the user at any step (currently 9 images total).

statement	relative preference	95% conf. int.
it was easy to find the target filter	3.4	1.1
there were a lot other interesting filters	1.0	1.5
interface was enjoyable and engaging	1.7	1.5

Table 5.5: Relative preference for feature-guided interface. Relative preference is computed as a difference of user ratings for the two interfaces on a 9-point Likert scale. Users consistently preferred image-guided interface to the random one. Note that random interface usually shows a good variety of unrelated images, so user preference for the second question is naturally weaker. Overall, users enjoyed using the guided interface more.

5.6.3 User-study results

We conducted a user-study to evaluate the effectiveness of the proposed user interface. Users were asked to select a filter they liked and then perform the search task (based on a different image) using two different versions of the search interface. Study participants were asked to rate the quality of their experience with each interface. All ratings were performed on a 9-point Likert scale. One interface presented random images from the collection. The other interface used the feature-guided strategy described above to select images. All participants used both interfaces, but the order of interfaces was randomized. A sample view of the two interfaces used in the study is shown in Figure 5-5.

Users were asked to rate the ease of finding images, to rate the quality of other images they saw during the search, and to rate the quality of their experience for each interface. The relative preference for our proposed interface is summarized in Table 5.5. The relative ratings are computed as a difference of ratings given to the two interfaces. Users felt that having feature-based guidance made the search task easier. Note that users saw a lot of other interesting images while using both interfaces. This can be explained by the high variance of the random interface: it presents much more variety than the guided one. However, despite the increased variety, users consistently reported having a more enjoyable and engaging experience with the feature-guided interface.

The participants in the study were also asked to rate the relative degree of control

statement about feature-guided UI	Likert-9 score	95% conf. int.
I felt more in control	7.3	0.9
I felt more creative	6.9	0.7

Table 5.6: Average user agreement with a given statement on a 9-point Likert scale (0=Strongly disagree, 9=Strongly agree). Users felt more in control and more creative when using the feature-guided interface.

that interfaces provided. The results are summarized in Table 5.6. Overall users felt that they were more in control with a feature-guided UI. However, a few users were not able to find a satisfactory filter in limited time of the experiment. Consequently, these users reported having only slightly more control with the feature-guided UI. Overall users reported feeling more creative when using the feature-guided UI.

5.7 Conclusions and lessons learned

In this chapter we proposed an approach that enables users to create and tune image filters without requiring any image processing expertise. At the core of our approach is a typed grammar of image operators. This powerful and expressive grammar allows to create novel image filters by generating expressions. In order to further improve the quality of generated filters we rely on genetic programming which adapts for filter generation. We have demonstrated that using our grammar and genetic programming we can completely automate image filter design using Amazon Mechanical Turk. We have also demonstrated that our approach is practical for interactive image filter discovery by users with varying levels of image processing expertise.

One important lesson learned in conducting experiments for this chapter is that Amazon Mechanical Turk may not be the best place to look for creative users. Workers on Amazon Mechanical Turk are trying to, first and foremost, earn money by completing tasks. These tasks are usually very clearly defined and have only one right answer. There is also an assumed penalty for a wrong answer. This setup is unsuitable for creative tasks. Any task that requires a creative aesthetic judgment involves risk. This may be the reason why workers shied away from (novel) unusual looks and gravitated towards (boring) filters that produced only minor alterations. We believe that more controlled user studies with users who are genuinely interested in the outcome are more appropriate for this context.

Chapter 6

Conclusions and future work

6.1 Summary

In this thesis we have explored and evaluated a number of photo enhancement and stylization methods ranging from simple automatic techniques to data-driven learning and stochastic approaches. We have shown that carefully designed heuristics (Chapter 3) can reliably produce pleasing renditions. We have also shown that creative users with little or no image processing expertise can discover novel and interesting photo stylizations (Chapter 5).

We have collected and analyzed a unique dataset of 5,000 raw photographs along with professional adjustments (Chapter 4). Using this dataset, we found that despite of a large number of controls used by professional photographers the space of tonal adjustments is low-dimensional. We have also demonstrated that it is possible to learn and accurately predict photographic tonal adjustments. Since its publication, our dataset has been used in a number of research projects [30, 33, 65, 72, 80, 82]. On the commercial front, Adobe Systems included our method for automatic tone correction in their flagship product, Adobe Photoshop.

6.2 Future work

Even though this thesis represents years of work on photograph enhancement, it only scratches the surface of the problem. There is a number of possible future directions that start where this thesis ends.

6.2.1 Content-aware local adjustments

In this thesis we focused on learning and predicting on global adjustments. However, even in 1950s Ansel Adams performed local "dodging and burning" [4] to enhance his photographs: such local enhancements can have a dramatic effect on the appearance of photos. Related work in this direction includes work by Berthouzos et al. [7] and Hwang et al. [30]. However, the former targets only certain portrait specific edits, while the latter uses a dataset with global adjustments for inferring local edits. A more general approach to this problem would be to collect a dataset of images with non-destructive local edits (or before/after pairs) and to learn from it.

6.2.2 Data-driven exploration of photographic style

Even though our database (Chapter 4) contains adjustments from 5 different photographers, all of these photographers were asked to produce neutral, postcard-like renditions. In practice, photographers employ a variety of styles. Analyzing the space of these styles represents potentially fruitful direction for research. Related work in this direction includes Caicedo et al. [9]. Unfortunately, instead of collecting data from experts, this research relied on Amazon Mechanical Turk users with unknown photographic skills and likely uncalibrated monitors to collect image stylization data.

Inspired by the success of our dataset, we have collected a dataset of 60 photographic styles: 20 photographers adjusted the same set of 50 raw photographs in 3 different ways. We make this dataset available along with this thesis to enable future research in this area. Having a large number of styles applied to the same content presents an opportunity to revisit the *style-vs-content* problem studied by Tenenbaum and Freeman [73]. If successful, such decomposition opens the door to a number of exciting applications. One such application would be helping novice users by projecting their manual image adjustments to the nearest point on the manifold of "good" styles. If the space of styles if low-dimensional, one could set the style of photo collection by simply manipulating a few style sliders.

Bibliography

[1] Andrew Adams, Eino-Ville Talvala, Sung Hee Park, David E. Jacobs, Boris Ajdin, Natasha Gelfand, Jennifer Dolson, Daniel Vaquero, Jongmin Baek, Marius Tico, Hendrik P. A. Lensch, Wojciech Matusik, Kari Pulli, Mark Horowitz, and Marc Levoy. The frankencamera: an experimental platform for computational photography. *ACM Trans. Graph.*, 29(4):29:1–29:12, July 2010.

[2] Ansel Adams. *Camera and Lens: The Creative Approach.* Morgan and Lester, 1948.

[3] Ansel Adams. *The Negative: Exposure and Development Basic Photo 2.* Morgan and Lester, 1948.

[4] Ansel Adams. *The Print: Contact Printing and Enlarging.* Morgan and Lester, 1950.

[5] Soonmin Bae, Sylvain Paris, and Frédo Durand. Two-scale tone management for photographic look. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 637–645, New York, NY, USA, 2006. ACM.

[6] Michael S. Bernstein, Joel Brandt, Robert C. Miller, and David R. Karger. Crowds in two seconds: enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 33–42, New York, NY, USA, 2011. ACM.

[7] 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 Trans. Graph.*, 30(5):120:1–120:14, October 2011.

[8] M. Brand and P. Pletscher. A conditional random field for automatic photo editing. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –7, june 2008.

[9] Juan C. Caicedo, Ashish Kapoor, and Sing Bing Kang. Collaborative personalization of image enhancement. In *CVPR*, pages 249–256. IEEE, 2011.

[10] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. *ACM Trans. Graph.*, 25(3):624–630, 2006.

[11] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister. Image restoration using online photo collections. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[12] R. Datta, Jia Li, and J.Z. Wang. Algorithmic inferencing of aesthetics and emotion in natural images: An exposition. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 105–108, 2008.

[13] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Studying aesthetics in photographic images using a computational approach. In *Proceedings of the European Conference on Computer Vision*, 2006.

[14] Doug DeCarlo and Anthony Santella. Stylization and abstraction of photographs. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 769–776, New York, NY, USA, 2002. ACM.

[15] S. Dhar, V. Ordonez, and T. L. Berg. High level describable attributes for predicting aesthetics and interestingness. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1657–1664, Washington, DC, USA, 2011. IEEE Computer Society.

[16] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 257–266, New York, NY, USA, 2002. ACM.

[17] Alexei Efros and Thomas Leung. Texture synthesis by non-parametric sampling. In *In International Conference on Computer Vision*, pages 1033–1038, 1999.

[18] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001.

[19] Katrin Eisman and Wayne Palmer. *Adobe Photoshop Restoration & Retouching.* New Riders Press, 2006. ISBN: 0321316274.

[20] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics*, 27(3), 2008. Proceedings of the ACM SIGGRAPH conference.

[21] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. *ACM Trans. on Graphics*, 21(3), 2002. Proc. of ACM SIGGRAPH conf.

[22] Eduardo S. L. Gastal and Manuel M. Oliveira. Domain transform for edge-aware image and video processing. *ACM TOG*, 30(4):69:1–69:12, 2011. Proceedings of SIGGRAPH 2011.

[23] Peter Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *Proceedings of the conference on Computer Vision and Pattern Recognition*, 2008.

[24] Yotam Gingold, Ariel Shamir, and Daniel Cohen-Or. Micro perceptual human computation. *ACM Transactions on Graphics (TOG)*, 31(5):119:1–119:12, August 2012.

[25] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA., 2001.

[26] Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. Generating photo manipulation tutorials by demonstration. *ACM Trans. Graph.*, 28(3):66:1–66:9, July 2009.

[27] S.W. Hasinoff, M. Jozwiak, F. Durand, and W.T. Freeman. Search-and-replace editing for personal photo collections. In *Computational Photography (ICCP), 2010 IEEE International Conference on*, pages 1 –8, march 2010.

[28] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, 2009. ISBN: 0387848576.

[29] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, New York, NY, USA, 2001. ACM.

[30] Sung Ju Hwang, Ashish Kapoor, and Sing Bing Kang. Context-based automatic local image enhancement. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV (1)*, volume 7572 of *Lecture Notes in Computer Science*, pages 569–582. Springer, 2012.

[31] Henry Kang, Seungyong Lee, and Charles K. Chui. Flow-based image abstraction. *IEEE Transactions on Visualization and Computer Graphics*, pages 62–76, 2009.

[32] Sing Bing Kang, Ashish Kapoor, and Dani Lischinski. Personalization of image enhancement. In *Proceedings of the conference on Conference on Computer Vision and Pattern Recognition*, 2010.

[33] Liad Kaufman, Dani Lischinski, and Michael Werman. Content-aware automatic photo enhancement. *Comp. Graph. Forum*, 31(8):2528–2540, December 2012.

[34] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[35] John R. Koza, Forrest H Bennett Iii, David Andre, Martin A, and Frank Dunlap. Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Transactions on Evolutionary Computation*, 1:109–128, 1997.

[36] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[37] T. Kuno, H. Sugiura, and N. Matoba. A New Automatic Exposure System for Digital Still Cameras. *IEEE Transactions on Consumer Electronics*, 44(1):192–199, 1998.

[38] Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, 1997.

[39] Tommer Leyvand, Daniel Cohen-Or, Gideon Dror, and Dani Lischinski. Data-driven enhancement of facial attractiveness. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–9, New York, NY, USA, 2008. ACM.

[40] Yiwen Luo and Xiaoou Tang. Photo and video quality evaluation: Focusing on the subject. In *Proceedings of the European Conference on Computer Vision*, 2008.

[41] Robert M. MacCallum, Matthias Mauch, Austin Burt, and Armand M. Leroi. Evolution of music by public choice. *Proceedings of the National Academy of Sciences*, 109(30):12081–12086, July 2012.

[42] Radoslaw Mantiuk, Rafal Mantiuk, Anna Tomaszewska, and Wolfgang Heindrich. Color correction for tone mapping. *Eurographics (Computer Graphics Forum)*, 28(2), 2009.

[43] RafałMantiuk, Scott Daly, and Louis Kerofsky. Display adaptive tone mapping. *ACM Trans. Graph.*, 27(3):1–10, 2008.

[44] Rafał Mantiuk, Scott Daly, Karol Myszkowski, and Hans-Peter Seidel. Predicting visible differences in high dynamic range images - model and its calibration. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Scott J. Daly, editors, *Human Vision and Electronic Imaging X, IS&T/SPIE's 17th Annual Symposium on Electronic Imaging (2005)*, volume 5666, pages 204–214, 2005.

[45] Rafał Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. Visible difference predicator for high dynamic range images. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 2763–2769, 2004.

[46] Dan Margulis. *Photoshop LAB Color: The Canyon Conundrum and Other Adventures in the Most Powerful Colorspace.* Peachpit Press, 2005. ISBN: 0321356780.

[47] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 389–400, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[48] Addy Ngan, Frédo Durand, and Wojciech Matusik. Image-driven navigation of analytical brdf models. In *Eurographics Symposium on Rendering*, pages 399–407. Eurographics Association, 2006.

[49] Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz. Local laplacian filters: edge-aware image processing with a laplacian pyramid. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 68:1–68:12, New York, NY, USA, 2011. ACM.

[50] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frdo Durand. A gentle introduction to bilateral filtering and its applications, 2007. In *Course at the ACM SIGGRAPH conference.*

[51] Sören Pirk, Ondrej Stava, Julian Kratt, Michel Abdul Massih Said, Boris Neubert, Radom´ir Měch, Bedrich Benes, and Oliver Deussen. Plastic trees: interactive self-adapting botanical tree models. *ACM Trans. Graph.*, 31(4):50:1–50:10, July 2012.

[52] François Pitié, Anil C. Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Comput. Vis. Image Underst.*, 107(1-2):123–137, July 2007.

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

[54] Przemyslaw Prusinkiewicz and James Hanan. *Lindenmayer Systems, Fractals, and Plants.* Springer-Verlag, 1989.

[55] Patrick Prez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3), July 2003. Proceedings of the ACM SIGGRAPH conference.

[56] Maneesh Agrawala Raanan Fattal and Szymon Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):to appear, 2007.

[57] Carl Edward Rasmussen and Chris Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[58] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting.* Morgan Kaufman, 2010.

[59] Erik Reinhard and Tania Pouli. Colour spaces for colour transfer. In *Proceedings of the Third international conference on Computational color imaging*, CCIW'11, pages 1–15, Berlin, Heidelberg, 2011. Springer-Verlag.

[60] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging.* Morgan Kaufmann Publishers, 2005.

[61] Craig Reynolds. Interactive Evolution of Camouflage. *Artificial Life*, 17(2):123–136, March 2011.

[62] John C. Russ. *The Image Processing Handbook.* CRC, 2002.

[63] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Image appearance exploration by model-based navigation. *Comput. Graph. Forum*, 28(2):629–638, 2009.

[64] Gaurav Sharma. *Digital Color Imaging Handbook.* CRC Press, Inc., Boca Raton, FL, USA, 2002.

[65] Hyunjung Shim and Seungkyu Lee. Technical section: Automatic color realism enhancement for computer generated images. *Comput. Graph.*, 36(8):966–973, December 2012.

[66] Karl Sims. Artificial evolution for computer graphics. *Computer Graphics*, pages 319–328, 1991.

[67] Pitchaya Sitthi-amorn, Jason Lawrence, Lei Yang, Pedro V. Sander, Diego Nehab, and Jiahe Xi. Automated reprojection-based pixel shader optimization. In *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, pages 127:1–127:11, New York, NY, USA, 2008. ACM.

[68] Sara L. Su, Frédo Durand, and Maneesh Agrawala. De-emphasis of distracting image regions using texture power maps. In *Texture 2005: Proceedings of the 4th IEEE International Workshop on Texture Analysis and Synthesis in conjunction with ICCV'05*, pages 119–124, October 2005.

[69] Jerry O. Talton, Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. Exploratory modeling with collaborative design spaces. In *ACM SIGGRAPH Asia 2009 papers*, SIGGRAPH Asia '09, pages 167:1–167:10, New York, NY, USA, 2009. ACM.

[70] Jie Tan, Yuting Gu, Greg Turk, and C. Karen Liu. Articulated swimming creatures. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 58:1–58:12. ACM, 2011.

[71] Litian Tao, Lu Yuan, and Jian Sun. Skyfinder: attribute-based sky image search. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–5, New York, NY, USA, 2009. ACM.

[72] Michael W. Tao and Aravind Krishnaswamy. Fast adaptive edge-aware mask generation. In *Proceedings of Graphics Interface 2012*, GI '12, pages 77–83, Toronto, Ont., Canada, Canada, 2012. Canadian Information Processing Society.

[73] Joshua B. Tenenbaum and William T. Freeman. Separating style and content. In *Proceedings of the conference on Advances in Neural Information Processing Systems*, 1997.

[74] Antonio Torralba, Robert Fergus, and William T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1958–1970, 2008.

[75] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511 - I–518 vol.1, 2001.

[76] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004.

[77] Greg Ward. Radiance file format.

[78] Holger Winnemöller, Jan Eric Kyprianidis, and Sven Olsen. XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics*, March 2012.

[79] Kai Xu, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Fit and diverse: Set evolution for inspiring 3d shape galleries. *ACM Transactions on Graphics, (Proc. of SIGGRAPH 2012)*, 31(4):to appear, 2012.

[80] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. Understanding and improving the realism of image composites. *ACM Trans. Graph.*, 31(4):84:1–84:10, July 2012.

[81] Akiko Yoshida, Rafa Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. Analysis of reproducing real-world appearance on displays of varying dynamic range. In Eduard Gröller and Lászlö Szirmay-Kalos, editors, *The European Association for Computer Graphics 27th Annual Conference: EUROGRAPHICS 2006*, volume 25(3) of *Computer Graphics Forum*, pages 415–426, Vienna, Austria, September 2006. Eurographics, Blackwell.

[82] Lu Yuan and Jian Sun. Automatic exposure correction of consumer photographs. In *Proceedings of the 12th European conference on Computer Vision - Volume Part IV*, ECCV'12, pages 771–785, Berlin, Heidelberg, 2012. Springer-Verlag.