Perceptually Inspired Image Estimation and Enhancement

by

Yuanzhen Li

M.S., Pattern Recognition and Intelligent Systems (2003)
Chinese Academy of Sciences

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author ....
Department of Brain and Cognitive Sciences
February 20, 2009

Certified by ....
Edward H. Adelson
Professor of Vision Sciences
Thesis Supervisor

Accepted by ....
Earl Miller
Professor of Neuroscience
Chairman, Department Committee on Graduate Theses

Perceptually Inspired Image Estimation and Enhancement

by

Yuanzhen Li

Submitted to the Department of Brain and Cognitive Sciences
on February 20, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In this thesis, we present three image estimation and enhancement algorithms inspired by human vision.

In the first part of the thesis, we propose an algorithm for mapping one image to another based on the statistics of a training set. Many vision problems can be cast as image mapping problems, such as, estimating reflectance from luminance, estimating shape from shading, separating signal and noise, etc. Such problems are typically under-constrained, and yet humans are remarkably good at solving them. Classic computational theories about the ability of the human visual system to solve such under-constrained problems attribute this feat to the use of some intuitive regularities of the world, e.g., surfaces tend to be piecewise constant. In recent years, there has been considerable interest in deriving more sophisticated statistical constraints from natural images, but because of the high-dimensional nature of images, representing and utilizing the learned models remains a challenge. Our techniques produce models that are very easy to store and to query. We show these techniques to be effective for a number of applications: removing noise from images, estimating a sharp image from a blurry one, decomposing an image into reflectance and illumination, and interpreting lightness illusions.

In the second part of the thesis, we present an algorithm for compressing the dynamic range of an image while retaining important visual detail. The human visual system confronts a serious challenge with dynamic range, in that the physical world has an extremely high dynamic range, while neurons have low dynamic ranges. The human visual system performs dynamic range compression by applying automatic gain control, in both the retina and the visual cortex. Taking inspiration from that, we designed techniques that involve multi-scale subband transforms and smooth gain control on subband coefficients, and resemble the contrast gain control mechanism in the visual cortex. We show our techniques to be successful in producing dynamic-range-compressed images without compromising the visibility of detail or introducing artifacts. We also show that the techniques can be adapted for the related problem of "companding", in which a high dynamic range image is converted to a low dynamic range image and saved using fewer bits, and later expanded back to high dynamic range with minimal loss of visual quality.

In the third part of the thesis, we propose a technique that enables a user to easily localize image and video editing by drawing a small number of rough scribbles. Image segmentation, usually treated as an unsupervised clustering problem, is extremely difficult to solve. With a minimal degree of user supervision, however, we are able to generate selection masks with good quality. Our technique learns a classifier using the user-scribbled pixels as training examples, and uses the classifier to classify the rest of the pixels into distinct classes. It then uses the classification results as per-pixel data terms, combines them with a smoothness term that respects color discontinuities, and generates better results than state-of-art algorithms for interactive segmentation.

Thesis Supervisor: Edward H. Adelson
Title: Professor of Vision Sciences

Acknowledgments

In an endeavor that has spanned five and a half years, it is not possible to adequately thank all of those involved. The following is only a small token of the gratitude I feel towards those who have been my colleagues, guides, and supporters in this undertaking.

I would like to thank my advisor, Ted Adelson, for his wisdom, guidance, and support. I have learned a tremendous amount from Ted, about a great many things: images, how the brain works, statistics, signal processing, how to give clear presentations, making molds, casting objects, etc. But most importantly, I have learned how not to jump to conclusions but to always try to think a problem through. Ted's endless knowledge and personal charm have also made this journey very enjoyable.

I thank members of my thesis committee - Frédo Durand, Josh Tenenbaum, and Jay Thornton, for their encouragement and invaluable feedback on my thesis.

I am fortunate to have had the opportunity to work with and learn from many thoughtful people, including, Ruth Rosenholtz, Ramesh Raskar, Aseem Agarwala, Steve Lin, Harry Shum, Sing Bing Kang, Hanqing Lu, Lavanya Sharan, and Amit Agrawal. I have also benefited greatly from past and current members of the Persci Lab, including, Marshall Tappen, Roland Fleming, Ce Liu, Kimo Johnson, Nadja Schinkel-Bielefeld, Nat Twarog, Lisa Nakano, Alvin Raj, Josh Mcdermott, Robert Rauschenberger, and John Canfield.

Last, but certainly not least, I would like to thank my family, for their love and support. It really would not have been possible without them. To them I dedicate this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, we present three image estimation and enhancement algorithms inspired by human vision.

In the first part of the thesis, we propose an algorithm for mapping one image to another based on the statistics of a training set. Many vision problems can be cast as image mapping problems, such as, estimating reflectance from luminance, estimating shape from shading, separating signal and noise, etc. Such problems are typically under-constrained, and yet humans are remarkably good at solving them. Classic computational theories about the ability of the human visual system to solve such under-constrained problems attribute this feat to the use of some intuitive regularities of the world, e.g., surfaces tend to be piecewise constant. In recent years, there has been considerable interest in deriving more sophisticated statistical constraints from natural images, but because of the high-dimensional nature of images, representing and utilizing the learned models remains a challenge. Our techniques produce models that are very easy to store and to query. We show these techniques to be effective for a number of applications: removing noise from images, estimating a sharp image from a blurry one, decomposing an image into reflectance and illumination, and interpreting lightness illusions.

In the second part of the thesis, we present an algorithm for compressing the dynamic range of an image while retaining important visual detail. The human visual system confronts a serious challenge with dynamic range, in that the physical world has an extremely high dynamic range, while neurons have low dynamic ranges. The human visual system performs dynamic range compression by applying automatic gain control, in both the retina and the visual cortex. Taking inspiration from that, we designed techniques that involve multi-scale subband transforms and smooth gain control on subband coefficients, and resemble the contrast gain control mechanism in the visual cortex. We show our techniques to be successful in producing dynamic-range-compressed images without compromising the visibility of details or introducing artifacts. We also show that the techniques can be adapted for the related problem of "companding", in which a high dynamic range image is converted to a low dynamic range image and saved using fewer bits, and later expanded back to high dynamic range with minimal loss of visual quality.

In the third part of the thesis, we propose a technique that enables a user to easily localize image and video editing by drawing a small number of rough scribbles. Image segmentation, usually treated as an unsupervised clustering problem, is extremely difficult to solve. With a minimal degree of user supervision, however, we are able to generate selection masks with good quality. Our technique learns a classifier utilizing the user-scribbled pixels as training examples, and then uses the classifier to classify the rest of the pixels into distinct classes. It then uses the classification results as per-pixel data terms, combines them with a smoothness term that respects color discontinuities, and generates better results than state-of-art algorithms for interactive segmentation.

1.1 Image estimation using local and global statistics

The human visual system is able to extract remarkably reliable information about world properties from a highly variable and complex visual environment. A surface illuminated by sunlight and the same surface in cloudy light reflect drastically different amounts of light. The image of a teapot, a mug, or a shampoo bottle, can change dramatically when the lighting condition or the viewing angle changes even by a very small amount. Yet, humans are able to "see through" the variability and complexity, and reach very good estimates about properties of the world, such as surface reflectance and shape. In order to understand how the human visual system achieves this, it is important to ask questions on the "computational level" [67, 66], such as, what is the problem that a visual system has to solve? What are the constraints and the nature of the problem, given the natural environment in which the visual system operates? As argued eloquently by J. J. Gibson [33], one cannot understand the visual system without understanding the world in which it operates.

In 1971, Land and McCann [52] asked the question of how humans can reliably estimate surface reflectance under varying lighting conditions, and sought to explain this ability in terms of some intuitive statistics of the world. They noted that reflectance tends to be piecewise constant, whereas shading tends to vary smoothly across space. As seen in images, sharp edges tend to be caused by reflectance changes, and smooth gradients by shading changes. Land and McCann set up psychophysical experiments with uniform-reflectance paint patches lit by slowly varying illumination, and confirmed that sharp edges play a crucial role in people's estimation of surface reflectance. They also considered how the human visual system might solve the problem algorithmically. Their algorithm, named "Retinex", involves first taking the derivatives of the log of an image, then thresholding the derivatives in order to differentiate those caused by reflectance changes from those caused by illumination, and finally integrating to reconstruct reflectance. An image is transformed into a different domain, i.e., the derivatives domain, where reflectance tends to be either large or zero, and becomes statistically differentiable from illumination.

A very similar image model was developed for a seemingly very different problem: removing noise from an image. Coring techniques, sometimes also referred to as wavelet shrinkage [11, 20], make use of the fact that when a noisy image is convolved with linear filters and transformed into the subband domain, the signal component tends to be either large or zero, and becomes statistically differentiable from the noise component. When the filtered values are thresholded and then used to reconstruct an image, the noise is reduced.

The image models used by retinex and wavelet coring have their limitations. Coring can result in overly-flattened regions and ringing along edges. Retinex works well for situations that satisfy the assumption of piecewise constant reflectance and smoothly varying illumination, e.g., the "Mondrian"-like images originally used by Land and McCann. But the real world is far more complex. Illumination could contain sharp edges, for example when there is an abrupt change in surface normal, a small light source giving sharp shadows, or in any number of other situations. Reflectance could lead to derivatives of small or medium amplitude. Another limitation of retinex and simple coring is that they involve examining and thresholding one single filtered value at a time. If we imagine neurons performing either task, each neuron would have a very small receptive field.

In Chapter 2, we propose a set of techniques to obtain better image models by learning from training sets containing natural images. We use example image pairs as training data, and learn the mapping between the input and the output. The same formulation has been used by others, most notably by Freeman et al. [29] and Hertzman et al. [41]. Their methods, however, produce models that are expensive to store or to query, and give results that are less than satisfying to the human eye. Our approach, which we will describe in Chapter 2, leads to significantly more compact representations of the learned models, significantly more efficient queries of the learned models, and much improved results.

As with Retinex and Coring, our approach makes the mapping problem more tractable by convolving the pixel values with linear filters and transforming the image into a different domain. But unlike Retinex or traditional Coring, which both use one single filtered value as the input variable, we use multiple filtered values, and also nonlinear functions of the filtered values, as input variables. Using a neuron analogy, an imaginary neuron making a decision about the horizontal derivative of reflectance at a particular location would be allowed to use information not only from the neuron responding to the input horizontal derivative at that location, but also from neurons responding to the input image derivatives at neighboring locations, in both horizontal and vertical orientations.

Besides local constraints, we also impose global constraints expressed as subband histograms. Subband histograms capture the textural quality of an image [39], and can provide important statistical constraints for images that are perceptually alike. By making the estimated image have similar subband histograms to those observed from a particular class of images, we make it a more probable instance of that class. One concern, however, is that the subband marginals of different images of the same class, while having similarities (*e.g.*, all being kurtotic), may still be different. In Chapter 2, we show how to account for variations among images, and how to estimate the target histograms from a training set.

1.2 Dynamic range compression and companding with multiscale architectures

The human visual system confronts a serious challenge in handling the dynamic range seen in everyday life. The neurons in the visual cortex have low dynamic ranges, and their responses are noisy, therefore it is important to keep them within an optimal operating range whenever possible. The physical world, however, has an extremely high dynamic range. A field of view containing dark shadows, sunlight, or specular reflections is likely to have a dynamic range of 10^6 or higher. When you sit in a dimly lit room on the beach and look at a sunny scene outside, your visual system is dealing with this challenge. When you watch a beautiful sunset, or when you glance at the lit lamp by your bedside in the night, your visual system is dealing with this challenge.

The human visual system handles the dynamic range challenge effortlessly, perceiving the bright, the dark, and the detail in between. In contrast, displaying images on computer screens with limited dynamic ranges presents the challenge of compressing the dynamic range of an image without compromising the visibility of detail. A straightforward method is to take the log of pixel luminance, which crudely models the adaptation of retinal photoreceptors adapting to the ambient level. The dynamic range is compressed, but the contrast of detail is reduced in high-luminance regions, and the image can look washed out. Stockham [91] proposed to estimate the illumination at each pixel and subsequently divide it out, based on the assumption that illumination can vary greatly from region to region and thus cause dynamic range problems. In Stockham's method, local illumination is estimated as the geometric mean over a local patch. This method is equivalent to subtracting a blurred version of the image in the log luminance domain. Unfortunately, the technique introduces artifacts known as "banding" or "halos" when there is an abrupt change of luminance, i.e., at large step edges.

It is important to have techniques that can effectively compress the dynamic range of an image without introducing artifacts, and at the same time maintain or enhance the visibility of detail. In recent years, high dynamic range (HDR) image data is increasingly available from such sources such as digital photography, computer graphics, and medical imaging

[106, 16, 70, 65]. Although new HDR display systems are being developed [86], the dominant display technologies, such as printed paper, CRTs, and LCDs, have limited dynamic ranges. Various techniques have been developed for compressing the dynamic range of the a high dynamic range image so it can be shown effectively on a low dynamic range display [99, 106, 25, 53, 72, 18, 21, 24, 82]. Among them, the multi-scale techniques [47, 74, 100] have aspects that are designed to capture properties of the human visual system, but they produce the same type of halo artifacts, although in much reduced forms, as those seen with Stockham's approach. In fact, the graphics community has come to believe a multi-scale technique for dynamic range compression will almost always produce halos, and has therefore focused on other approaches.

We believe a carefully designed multi-scale range compression algorithm can overcome such artifacts and will have great utility. The human visual system has a multi-scale solution to the dynamic range problem it faces. Automatic gain control occurs in the first stages of the visual cortex, area V1, where it is known as "contrast gain control" or "contrast normalization" [38, 40, 81, 96]. Responses to moderately low contrasts are boosted, while responses to high contrasts are reduced. This makes good use of the information capacity of the neurons by keeping the responses away from the ceiling and floor. Area V1 has neurons tuned to different orientations and spatial frequencies, and can be thought of as doing a multi-scale subband decomposition of images using filters similar to wavelets. The gain of a given neuron is controlled by the activity level of many neurons in its immediate neighborhood. Additionally, the gain is not just controlled by neurons tuned for the same orientation and spatial scale; rather, the gain signal involves a pooling of multiple orientations and scales.

In Chapter 3, we present a multi-scale algorithm that produces dynamic-range-compressed and visually pleasing images, free of halo artifacts. We decompose an image into subbands, and perform smooth gain control on the subband coefficients, in a style resembling the gain control mechanism of the visual cortex. This smooth gain control proves to be crucial in eliminating the halo artifacts. It keeps gains at neighboring locations and scales well matched, and therefore reduces nonlinear distortions at important image features such as strong edges. Another important factor that leads to minimal haloes is the use of symmetric analysis-synthesis filters, which "filter away" the cross-frequency aliasing that comes with the nonlinear operations needed for compression of the dynamic range.

1.3 Localized image and video editing

Local manipulation of color and tone is one of the most common operations in the digital imaging workflow. For example, to improve a photograph or video sequence an artist might want to increase the saturation of grass regions, make the sky bluer, and brighten the people. Traditionally, localized image editing is performed by carefully isolating the desired regions using various selection tools (e.g., Photoshop Magic Wand). This process can be extremely time-consuming, especially for video, given its many frames, and factors like motion and changing perspective. In recent years, sophisticated matting techniques for the purpose of compositing have been developed [103, 104, 14, 55]. For this purpose, precise modeling of the fractional contributions of each layer at each pixel is required, and the user is normally asked to specify a trimap that labels the border between the foreground and the background. This process is more burdensome than necessary if the user simply wishes to locally adjust color and tone. One property of human visual perception is that we are much more sensitive to local luminance ratios at image edges than we are to slow changes in color and brightness. As a result, localized edits that do not substantially alter the strength or location of luminance edges tend to look natural, regardless of whether the edits are controlled by accurate mattes.

Recent algorithms referred to as edge aware interpolation [56, 62, 13, 107] offer the user a different approach to localized manipulation. Instead of carefully isolating regions or specifying trimaps, a user simply draws rough scribbles on the image, and attaches adjustment parameters to each scribble. These adjustments parameters are then interpolated over the rest of the image or video in a fashion that respects image edges. This is achieved by having a cost function that penalizes differences in interpolated values among neighboring pixels, but with the strength of penalty weakened by image edges or significant differences in colors of the neighboring pixels.

While edge aware interpolation promises to be a powerful technique for localized image and video manipulation, there are a number of problems that currently limit its success in this context. It doesn't handle textured objects well because image edges caused by texture slow the propagation of scribble influence. It also doesn't handle fragmented appearances well unless the user painstakingly scribbles each fragment. Moreover, the influence maps tend to be too soft, due to the quadratic cost functions used in such systems, and color manipulations using such influence maps tend to have significant color spilling. Finally, manipulating video proves to be difficult with edge aware interpolation.

In Chapter 4, we significantly improve the performance of edge aware interpolation for local image and video adjustment by taking advantage of an additional cue that is not utilized in existing edge aware interpolation systems. Typically, the regions that a user wishes to adjust differently are not only separated by image edges, but also tend to have distinct distributions of color and texture. We attempt to classify pixels into regions by learning a discriminative classifier, i.e., Adaboost [32, 31], and then combine this per-pixel data term with the spatial regularization provided by the original smoothness term of edge aware interpolation systems. When combining the data and the smoothness terms, we propose a novel approach to the relative weighting of each in a fashion that takes into account the accuracy of the classifier on the training data. These contributions allow our system, using just a few user-drawn scribbles, to achieve substantially better results than previous work.

With our interactive system prototype, a user draws scribbles in different colors, indicating the individual classes of content that she wishes to manipulate differently. Our system takes the scribbled pixels as training examples, and builds an Adaboost classifier that discriminates between the classes. The Adaboost training consists of sequentially finding a number of weak classifiers, each focusing on the training examples mis-classified by previous weak classifiers. The final classifier classifies a pixel by a weighted sum of the beliefs of each weak classifier. The class label of each pixel is later used as the data term. Note that there is uncertainty associated with the class labels. The more the weak classifiers agree with each other, the more confident we are as to which class the pixel belongs to. Moreover, the smaller the mis-classification rate of the final classifier, the more confident we are as to how differentiable the classes are. We combine these two measures of confidence, and weight the per-pixel data term accordingly, so that the smoothness term takes over when the classification confidence is low. The smoothness term imposes the constraint that neighboring pixels with similar colors should have similar class labels, and thus respects color discontinuities. Our algorithm produces results that compare favorably against a number of state-of-art techniques for selection or interactive segmentation [7, 58, 104, 62].

Chapter 2

Image estimation using local and global statistics

We propose a set of techniques for mapping one image to another based on the statistics of a training set. We apply these techniques to the following problems: removing noise from an image, estimating a sharp image from a blurry one, decomposing an image into reflectance and illumination, interpreting lightness illusions, and generating line drawings from photographs. Example image pairs are used as training data, and the statistical mappings between the image pairs are learned. The learning is challenging because images are very high dimensional. The techniques we describe in this chapter produce compact representations of the learned models; these representations can be efficiently queried and easily stored. The techniques can be easily adapted to address a wide range of vision and image processing problems that take the form of image to image mapping.

2.1 Introduction

Many tasks in image processing and machine vision take the form of image to image mapping. Given an image, X, the problem is to estimate another image Y, which is in register with the first. Classical image processing problems such as denoising and superresolution

[1] Part of this chapter (2.1-2.5.2) has appeared as: [59] Yuanzhen Li, Edward H. Adelson. Image Mapping Using Local and Global Statistics. *Human Vision and Electronic Imaging XIII, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6806*, 2008.

are examples of image mapping. Within machine vision, estimating intrinsic images [8] such as albedo and illumination, or depth, or optic flow would be other examples. Stylistic image mapping, under the names "texture transfer" [22] and "image analogies" [41], forms another category.

There has been much interest in image mapping methods based on machine learning. Given a set of training pairs, (X_1, Y_1); (X_2, Y_2); $\ldots$, one learns the statistical relationships between X and Y, so that a new image X_{new} can be mapped to its corresponding pair Y_{new}. Markov properties are generally assumed, so that the problem can be approached by modeling local dependencies. But even when we look at relatively small local patches, e.g., 5×5 ones, there are still 25 dimensions, and most of the 25-dimensional space will be empty of observations. For this reason, it is popular to use non-parametric representations of the conditional density, and to use nearest neighbor techniques to estimate the values of the target image [29, 41]. The density is not explicitly represented; instead a large number of examples are remembered and later queried. In high dimensional spaces, it is necessary to store a large number of examples, and the queries become quite slow. In addition, artifacts are introduced by the difficulties in enforcing coherence between adjoining patches.

For this reason, we have reconsidered the problem of directly representing the conditional density by learning parametric or semi-parametric models. To make the problem tractable, we work in the subband (e.g., wavelet) domain, so that we can take advantage of the kurtotic distributions found with natural images. In most cases, we represent a single number (the mean) rather than the full density, so that our problem reduces to function approximation. Our thinking is illustrated in Fig.2-1, where we begin by considering nearest neighbor methods.

Suppose that we wish to represent a 2D function and are given example values at the points shown in Fig.2-1(a). The points are densest near zero (here represented by a Gaussian falloff, which is much less kurtotic than image subbands). We can simply store all the examples, and later do a nearest neighbor search. In effect, we place a Voronoi neighborhood around every observation. When we have many examples, it becomes prohibitive to store and to query this information. We can reduce it to a smaller set of neighborhoods that are deemed representative. In vector quantization (VQ), shown in Fig.2-1(b), nearby

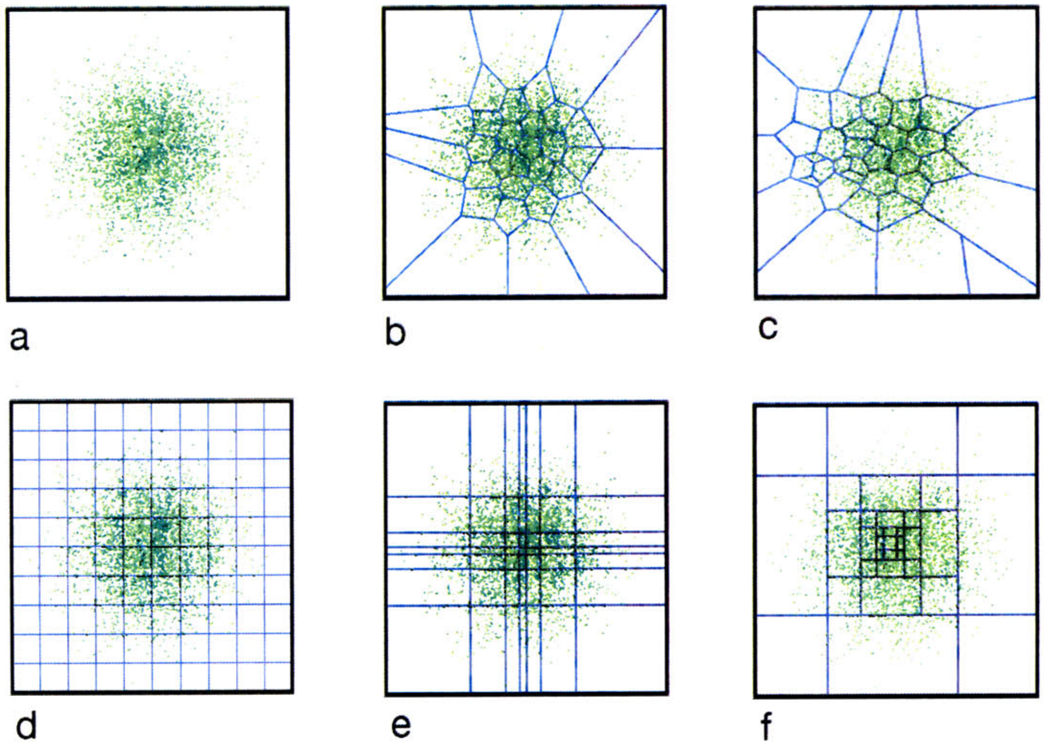


Figure 2-1: Illustration of different space partitioning techniques. (a) Data cloud. (b) Vector Quantization. (c) Mixture of Experts. (d) Cartesian quantization. (e) Separable binning representative of marginal densities. (f) Nested binning.

points are merged to form larger neighborhoods containing a certain number of examples. Popat and Picard [76] utilize VQ. Rather than basing the neighborhoods on example density, Tappen et al. [94] greedily select a subset that are determined to be most informative. Fig2-1(c) is suggestive of the sort of neighborhoods they might find. Next, consider a set of more direct representations. Fig.2-1(d) shows a simple case of Cartesian quantization. The space is divided into rectangular bins and the function is represented parametrically within each bin. Instead of storing examples, one represents the function explicitly, and there is no need for nearest neighbor computations at query time. There is one big disadvantage: representing a high dimensional function requires a great many bins. The number of bins required grows exponentially with dimensionality. We can improve things by moving the bin boundaries around to reflect the marginal densities, as shown in Fig.2-1(e). However, this separable method still leaves a great many bins.

We have devised a better approach, which we call Nested Binning Regression, shown in Fig2-1(f). The separable bins in (e) which are representative of the marginal densities get merged at certain places, leading to partitions better matched to image statistics, and still extremely easy to query. We will give more details in section 2.3. Within each bin, the function is approximated parametrically, with the parameters fit by regression within the bin. We achieve state of the art performance with compact models and very fast lookup. Our method is faster than k nearest neighbor methods by factors of 1000 or higher, even when compared to fast methods like k-d trees [46, 36, 43, 71].

In this chapter, we demonstrate the performance of Nested Binning Regression on the following applications: removing noise from an image, estimating a sharp image from a blurry one, decomposing an image into reflectance and illumination, interpreting lightness illusions, and generating line drawings from photographs.

For denoising and super-resolution, we also impose global constraints expressed as subband histograms, on top of the local constraints learned through Nested Binning Regression. The reason we add global constraints is that the outputs of Nested Binning Regression, although matching or exceeding the performances of state-of-art algorithms, tend to look too soft. We argue that perceptual "sharpness" can be thought of as a textural quality, and modeled via subband marginals, in the style of Heeger and Bergen [39].

2.2 Algorithm overview

Let $\{(X_i,Y_i)\}_{i=1}^{N}$ be a set of N training pairs. Each pair contains input image X_i and output image Y_i. We decompose the images into subbands, and for each subband coefficient of Y_i compute a feature vector from X_i and/or its subbands. We train regressions to predict Y_i's subband coefficients given X_i's feature vectors. Given a new input image, feature vectors are computed, and the above learned regressions are used to predict the subband coefficients of the output image. Subband coefficients constrain local neighborhoods of an image, and we refer to this stage as the learning of local constraints. Details of the regression algorithm will be given in section 2.3.

The output image reconstructed from the above estimated subband coefficients tends to look too soft. We impose textural statistics such as subband histograms, to correct the "look". Subband histograms constrain the whole image, and we refer to this stage as the learning of global constraints. Details will be given in section 2.4.

2.2.1 Markov random fields

The learning and utilization of local constraints, can be understood as learning and inference on a Markov random field, with the assumption that the probability of a local neighborhood can be modeled as a product of probabilities of individual subband coefficients, and that each coefficient is Gaussian when conditioned on the input. In probability language, we wish to model the conditional probability of Y given X, and when given a new input image X_{new}, the output can be estimated as the image that maximizes the probability of Y given that X is equal to X_{new}, *i.e.*, $\hat{Y}_{new} = \arg\max p(Y|X = X_{new})$.

Both X and Y are very high-dimensional. We assume that Y obeys the Markov property when conditioned on X: $p(y_k|X, \mathbf{y}_w, w \neq k) = p(y_k|X, \mathbf{y}_w, w \in N^y_{(k)})$, where y_k is a pixel value at location k, and $N^y_{(k)}$ is a neighborhood of k in Y but not containing k. By the Hammersley-Clifford theorem of random fields, the joint distribution of $(y_1, \dots, y_n)$ given X has the form

$$p(Y|X) \propto \exp\left(-\sum_k M_k(\mathbf{y}_{(k)}|X)\right), \tag{2.1}$$

where $\mathbf{y}_{(k)}$ is a clique, and $M_k(\mathbf{y}_{(k)}|X)$ is the potential function of that clique, given the observed image X.

We model the probability distribution $p(\mathbf{y}_{(k)}|X)$, which is related to $M_k(\mathbf{y}_{(k)}|X)$ by an exponential, by a product of T one-dimensional distributions:

$$p(\mathbf{y}_{(k)}|X) = \prod_{t=1}^{T} p(B_t^T \mathbf{y}_{(k)}|X), \tag{2.2}$$

where $\{B_t\}_{t=1}^T$ are a set of subband basis vectors. Here the cliques are defined on $m \times m$ neighborhoods where all the m^2 pixels are connected.

The next question is to model each of the individual conditionals, $p(B_t^T \mathbf{y}_{(k)}|X)$. We simplify $B_t^T \mathbf{y}_{(k)}$, the subband coefficient at location k in subband t, by $b_{t,k}$. The simplest choice for $p(b_{t,k}|X)$ is a conditional Gaussian:

$$b_{t,k}|X \sim N(\hat{b}_{t,k}, \sigma_{t,k}^2) \tag{2.3}$$

$\hat{b}_{t,k}$ depends on X:

$$\hat{b}_{t,k} = E\left[b_{t,k}|X\right] = \hat{b}_{t,k}(X), \tag{2.4}$$

and can be estimated through regression. $\sigma_{t,k}$ can also be estimated during regression, but for simplicity we make it be a constant.

Once $(\hat{b}_{t,k}, \sigma_{t,k})$ are in place, the clique potentials can be written as:

$$M_k(\mathbf{y}_{(k)}|X) = \sum_t \frac{1}{2\sigma_{t,k}^2}\left(B_t^T \mathbf{y}_{(k)} - \hat{b}_{t,k}(X)\right)^2. \tag{2.5}$$

The log probability of the random field is (offset by a constant):

$$\begin{aligned}\log\left(P(Y|X)\right) &= -\sum_k M_k(\mathbf{y}_{(k)}|X) \\ &= -\sum_k \sum_t \frac{1}{2\sigma_{t,k}^2}\left(B_t^T \mathbf{y}_{(k)} - \hat{b}_{t,k}(X)\right)^2,\end{aligned} \tag{2.6}$$

which we wish to maximize by estimating an image $\hat{Y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$. When $\sigma_{t,k}$ is a constant, and $\{B_t\}_{t=1}^T$ is a tight frame [15], $\hat{Y}$ can be reconstructed from $\hat{b}_{t,k}$ by applying the self-inverting subband transform to the estimated coefficients $\hat{b}_{t,k}$.

2.2.2 Subband histograms

The log probability in (2.6) is quadratic, and can easily be maximized. By constraining overlapping neighborhoods with subband coefficient estimates, we do not need to explicitly enforce spatial coherence, as Freeman *et al.* and Hertzmann *et al.* do. But we do have a problem of blurring, which is related to the Gaussian conditional model we use. Blurring is reflected in the subband marginals, in that they tend to be much less kurtotic than those of a sharp image. The global, marginal histograms of subband coefficients can provide important statistical constraints for images that are perceptually alike. Heeger and Bergen [39] demonstrated that by matching the subband histograms of a Gaussian noise image to those of a stochastic texture, new examples of the texture can be synthesized with matching appearance.

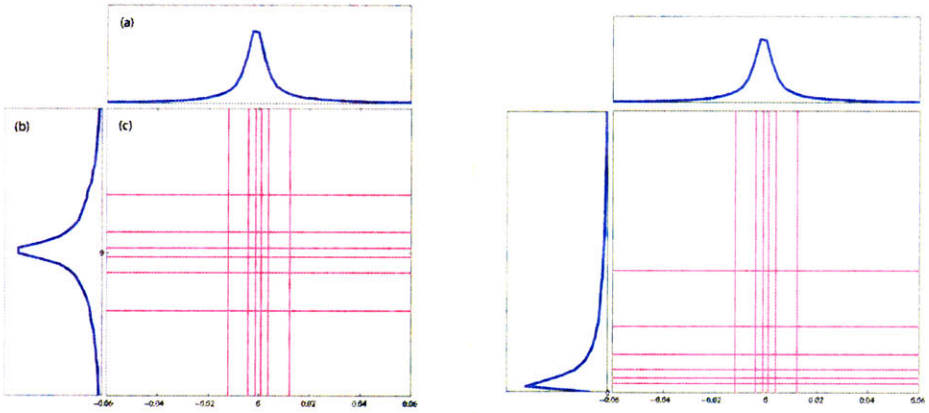
We use subband histograms as constraints to impose globally on the output image, Y. By making Y have similar subband histograms to those observed from a particular class of images, we make it a more probable instance of that class. One concern, however, is that the subband marginals of different images of the same class, while having similarities (*e.g.*, all being kurtotic), may still be different. In section 2.4, we show how to account for variations among images, and how to utilize the observed image X to retrieve good estimates of the target histograms.

2.3 Learning local constraints

For each subband of Y, we train a regressor to predict its coefficients using features extracted from local neighborhoods of the input image X. In general, three types of features are considered: linear subband coefficients, nonlinear local energy values, and pixel values. Local energy is computed by first taking the absolute values of the subband coefficients and then blurring them across space or orientations.

We can use powerful nonparametric methods to perform the regression. A problem is that they tend to be very costly when the training set is large and the dimensionality is high. For the applications considered in this paper, the size of the training set can be quite large, because each subband coefficient (together with its feature vector) constitutes an example. For example, if we have 20 500 by 500 images for training, and use spatially oversampled subband decompositions, then for each subband there will be 5 million training examples. When the training set is so large, even with a moderate number of dimensions, the computation involved in nonparametric methods can be very expensive.

We notice that with the subband coefficient based features we are considering, images demonstrate interesting regularities in the feature space. Based on the such regularities we propose a partitioning procedure to divide the space into easily retrievable bins. In each bin we perform parametric regression, leading to a compact representation of the learned model.



(a) (b)

Figure 2-2: Marginal equiquantization using two input features. (a) Input features are two subband coefficients at adjacent scales. (b) One feature (horizontal axis) is subband coefficient, and the other (vertical axis) is local energy.

2.3.1 Space partitioning

A conceptually easy way to obtain an easily retrievable partition, is to divide the space into hyper-rectangular bins. Marginal equiquantization [69, 42], used in the estimation of information theoretic measures, is a method to acquire separable hyper-rectangular bins. Each feature dimension is examined separately, and the partition is defined by marginal bins which are not equidistant but equiprobable intervals, meaning each marginal bin encapsulates approximately the same number of observed data examples. The partition is separable, so is extremely easy to obtain (via order statistics) and to retrieve. Fig.2-2 shows two such partitions on two dimensions, with different feature pairs. In (a), the features are subband coefficients at the same location and orientation, but at adjacent scales. In (b), one feature is subband coefficient, and the other is local energy computed by rectifying and blurring the coefficients. All the features here have kurtotic distributions, as shown by the marginal histograms on the top and the left of the partitions, and the data are highly clustered around zero. The bins are small near zero, and big away from zero.

If the features were statistically independent of each other, the joint bins obtained through marginal equiquantization would also be statistically equivalent. But the fea-

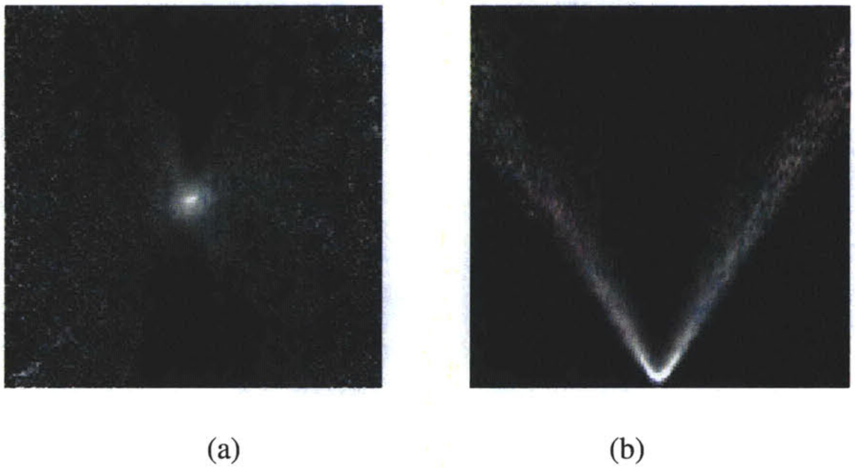


Figure 2-3: Conditional histograms of: (a) subband coefficient, conditioned on coefficients at the parent scale; (b) local energy, conditioned on subband coefficients. Dark means low density, and bright means high density.

tures are usually not independent of each other. Subband coefficients at adjacent locations, scales, or orientations, tend to be independent when amplitudes are small, but correlated when amplitudes are large [87], as illustrated by a conditional histogram in Fig.2-3(a). Coefficient amplitudes and local energy values, also tend to be correlated, as shown by a conditional histogram in Fig.2-3(b).

We find such pattern to be quite consistent when more features, such as coefficients and energy values at neighboring locations and orientations, are added. The pattern is that the data tend to be highly clustered around zero, and get predictably sparse away from zero, except when amplitudes of features show correlations. The joint bins acquired via marginal equiquantization, therefore, tend to be empty when some feature amplitude is large but not correlated with the amplitudes of others. We use a "nesting" strategy to merge the bins that tend to be sparsely populated. The nesting is performed in a hierarchical fashion, as illustrated in Fig.2-4. The boundaries of the nested bins are still parallel to the feature axes, therefore the retrieval is efficient. Nesting slows down the rate of total bin number growing with dimensionality. After nesting, most bins will be filled with enough data examples, *i.e.*, no fewer than the number of unknown parameters, to perform regression. Some bins will still be empty, and for them we progressively include points from neighboring bins until we can reliably estimate the regression parameters.

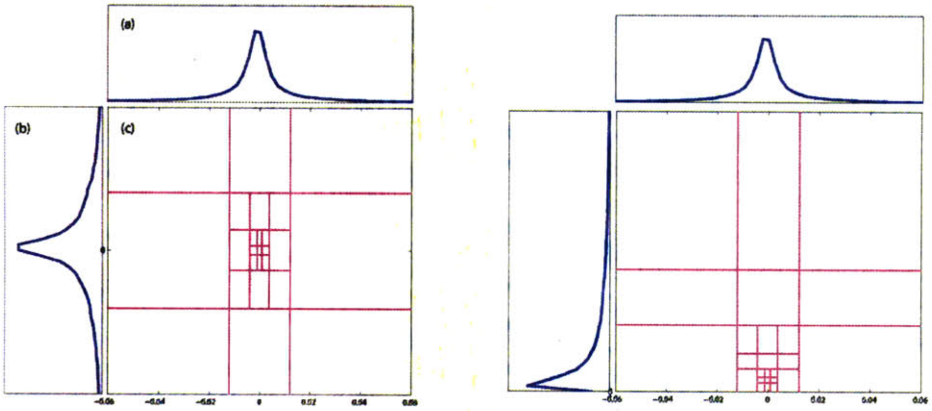


Figure 2-4: Nested binning using the same feature pairs as those used in Fig.2-2 and Fig.2-3. The bins in Fig.2-2 are merged at places where one feature but not the other has high amplitude.

2.3.2 Piecewise linear regression

After the feature space is partitioned into bins, parametric regression is performed within each bin. If we choose the simplest parametric model, *i.e.*, linear model with Gaussian noise, the regression parameters can be estimated using the standard least squares method.

When there are many training images, it is often desirable to start learning without having to wait for all data to be observed and stored. We make the regression process online, using recursive least squares [92]. The space partitioning requires marginal order statistics (percentiles), which can be estimated from partial training data.

Compared to nonparametric methods, our method leads to a much more compact model, which can be efficiently queried. For each subband, only the bin boundaries and a small number of parameters per bin need to be stored. We compared the running time of our method empirically to those of two implementations of kd-tree k nearest neighbors [71, 43]. When the parameters are tuned so that all implementations produce matching accuracy, our method is $>$1000 times faster.

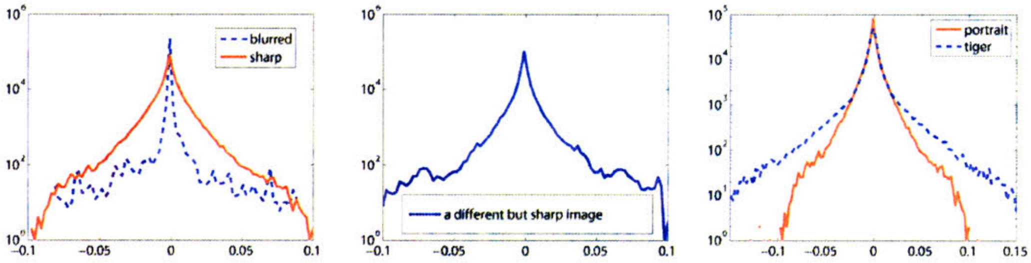
2.4 Learning global constraints

In the stage of learning local constraints, we typically learn a conditional mean of each subband coefficient, given a feature vector representing the surrounding patch. If the conditional probability of the coefficient is truly Gaussian, as has been assumed, then the conditional mean is also the maximum probability solution. But in some cases, the conditional cannot be well approximated with a Gaussian, and by doing regression we end up blurring the estimated signal. This phenomenon can be observed in the application of superresolution. A superresolution result using the regression estimates is shown in Fig.2-8(e), which looks blurred.

We think that "blur" is a textural quality, and can be captured by textural statistics, for example subband histograms. Fig.2-5(a) shows a subband histogram of an image, against that of a blurred version of the image. The two histograms are vastly different. A different sharp image, on the other hand, has subband histogram much more alike, as shown in (b). We propose to enforce "sharpness" as global textural constraints, matching an image's subband histograms to ones that are observed from the training images.

Another way to think about this, is that the "local" result that maximizes the log probability in (2.6), has been found in the entire continuous space of $\Re^n$. Images are special signals, and are commonly hypothesized to lie within a subspace of $\Re^n$ instead. One possibility to take this into account is to find the subspace, and search within it for an image that maximizes (2.6). But it is difficult to do so. Instead we employ a two-step strategy, first finding a solution in $\Re^n$ that maximizes (2.6), and then projecting it to the closest point that satisfies the target subband marginal constraints. This point is found using Heeger-Bergen style histogram matching.

One thing to keep in mind, is that the subband marginals of natural images, while all being kurtotic, may still vary a lot from image to image. Fig.2-5(c) shows two histograms of a portrait image and a tiger image, respectively. The overall "texture" of the two images is different, and the histograms are different. When we have a varied set of training images, we want to pick one or a few that are texturally similar to the target result. The problem is, with a blurred image whose real high frequencies are unknown, how do we



(a) (b) (c)

Figure 2-5: Subband histograms, on a log scale, (a) of a sharp image (solid), and of its blurred version (dash); (b) of a different but sharp image; (c) of a portrait image (solid), and a tiger image (dash).

pick such images? Note that the low frequency subbands of a blurred image are basically undegraded, and their marginals can be used as global textural features indexing into the training set. More specifically, the histograms of low frequency subbands of each training image are compared with the low frequency subband histograms of the blurred image, and the k nearest neighbors are found. The histograms of the higher frequency subbands of the k nearest neighbors are then averaged to give the target histograms. Distances between subband marginals are calculated using K-L divergence.

2.5 Applications and experiments

2.5.1 Image denoising

To test our system on denoising, we added Gaussian white noise. We generated image training pairs by adding synthetic noise to known clean images. The mapping from "noisy" to "clean" was learned, and then applied to new noisy images.

First we will present denoising results using only the local constraints, *i.e.*, the regression estimates of subband coefficients. Training such a regression model, can be understood as learning a multi-dimensional "coring" function from actual image data. In the classical coring technique [20, 88], a subband coefficient is subjected to a 1D nonlinear function which suppresses small amplitude values and preserves high amplitude values.

For the multidimensional case we use multiple features as input and learn a multidimensional function as the output. We experimented with various feature vectors, and settled on a 6D vector with the following features: the observed subband value; the observed value of its parent (same location, same orientation, but lower frequency); the observed value of its grandparent; and the local energies corresponding to these three subbands. Local energy is calculated within a subband by taking the absolute value and convolving with a blur kernel that is about the same size as the subband kernel. All operations were performed on an oversampled pyramid. For the results reported here, 9 bins per dimension were used for marginal equiquantization, which after nesting gave 2913 bins in total. The lowpass was kept unchanged.

We tested the algorithm using two different subband representations, and two different training sets. The first subband representation was a variant of the steerable pyramid [89], using 8 orientations and 4 scales as in [78]. The second subband representation is an oversampled QMF pyramid [3]. The first training set contains 18 images (5.7 million pixels altogether) which all have people in them. The second training set contains 40 images (6.2 million pixels) from the Berkeley segmentation data set [68]. They are the same 40 used by Roth and Black for training image priors.

The combination of subband representation and training set gives four different settings. Performances under the four settings are compared in Table 2.5.1, in terms of Peak Signal-to-Noise Ratios (PSNRs), on standard test images for denoising. When the four results are averaged, the average tends to have higher PSNR than the individual results, as shown in column 7. We vary the noise levels, and list more results in Table 2.2. In general, the performance of our algorithm matches those of the state-of-art denoising algorithms, such as Portilla *et al.*[78], Roth and Black [83], etc.

Upon the local results we then impose global statistics, *i.e.*, subband histograms. For global statistics we use the steerable subband representation with 3 scales and 4 orientations, and the first training set. Fig.2-6 shows the local and global results on Lena, and Fig.2-7 shows a blow-up, compared against Portilla et al., and Roth and Black. Note the fine details we are able to recover. One interesting thing to note, is that although we feel our global result looks the most pleasing, it has the lowest PSNR among the four. The reason is

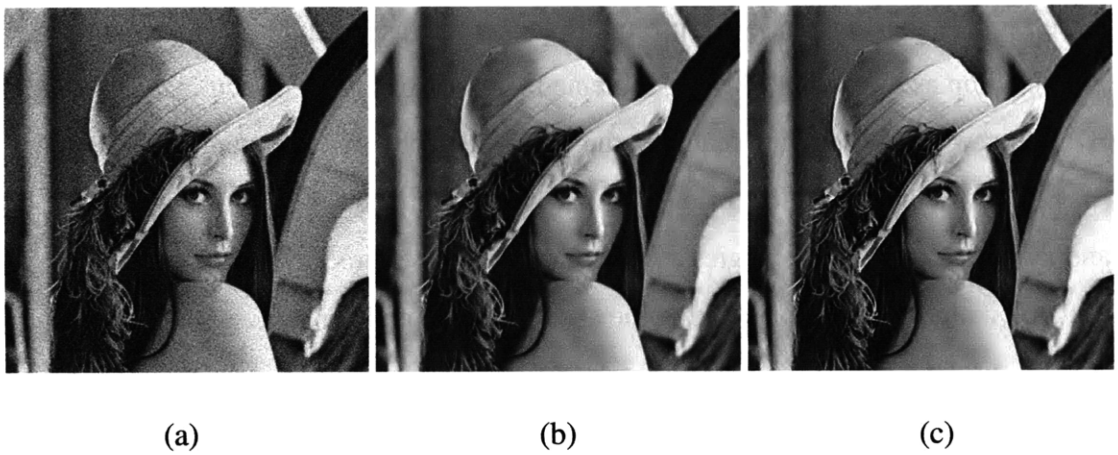


Figure 2-6: Denoising results. (a) Noisy image, with Gaussian additive noise of standard deviation 20. Noisy PSNR is 22.11dB. (b) Denoised image, using local constraints only. PSNR = 32.47dB. (c) Denoised image, using both local and global constraints. It looks pleasantly sharper, but the PSNR is 31.13dB, lower than (b).

that, by imposing global statistics, we are forcing the overall "texture" (such as the appearance of sharp edges), to resemble that we have seen in other clean images, thus pushing some individual pixel values away from the minimum square error (MSE) estimates. PSNR is a square error based criterion, and does not capture all aspects of distortions relevant to perceptual quality. Here we argue that textural similarity is often a valid additional criterion for perceptual similarity. In order to test this argument, we conducted psychophysical experiments where human subjects were asked to judge the perceptual qualities of results produced by our algorithm and the BLS-GSM algorithm by Portilla *et al.*. Rather than using a rating scale, which can be hard to interpret, we asked our subjects to match the subjective quality of images from one technique with those of the other technique at a different noise level. In this way, we could determine that one technique tolerated a certain amount more noise than the other. We express the advantage in terms of dB of the additive noise signal.

We ran 6 subjects, used 7 images and 3 levels of noise (σ=15,25,50). The results depended on the individual subjects and images, but when averaged our method outperformed BLS-GSM. When averaged across subjects, the advantage ranged from 0 dB to 6.16 dB, with mean at 3.15 dB.

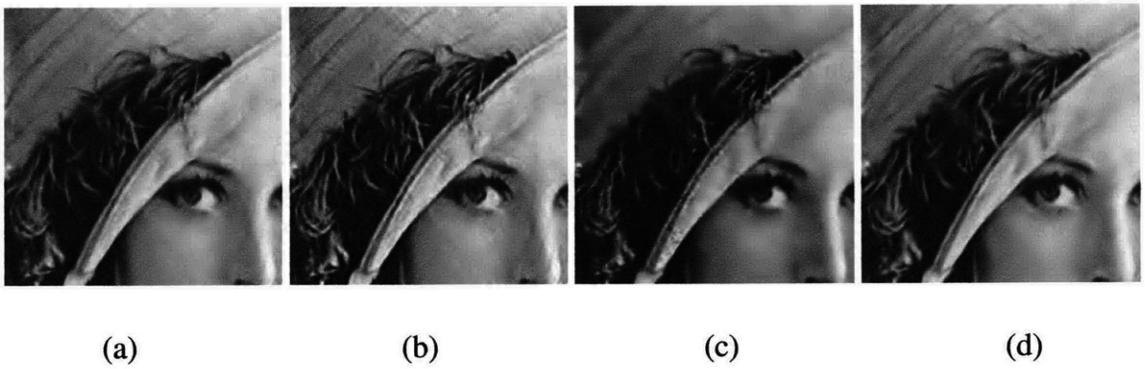


Figure 2-7: Denoising results. (a) Ours, using local constraints only. (b) Ours, using local and global constraints. (c) Roth and Black [83]. (d) Portilla *et al.* [78]. Note the fine eyelash and feather boa details recovered by our method in (b) but not by others.

2.5.2 Super-resolution

For super-resolution, the task is to estimate a sharp looking high- resolution image given a low-resolution image as input. The low res input was generated by downsampling the image by a factor of c, and then upsampling it by a factor of 1/c, both via bicubic interpolation, leading to a blurry image. We used high and low resolution pairs for training. Our super-resolution algorithm is almost exactly the same as that for denoising, except that the training pairs are different.

In Fig.2-8 (f) we show our super-resolution result on an image downsampled by a factor of 4 in each direction, and compare it to competing techniques, including Freeman *et al.* [30], Hertzmann *et al.* [41], and Genuine Fractals, which is a leading commercial product. Our result with both local and global constraints looks sharp and is free of disturbing artifacts. The training set here was the set of 18 people images. Steerable subband representation with 3 scales and 4 orientations were used. In the local constraints learning stage we used the same set of 6 features as those used for denoising. The same subband representation was used for global constraints estimation and imposition.

For super-resolution, to the best of our knowledge, there has not been a satisfying measure quantifying the quality of results. Here the goal is to hallucinate missing high frequencies, especially those belonging to sharp edges. If the hallucinated edge position is slightly wrong, the square error will be large, although the image may look good. Therefore PSNR

	St-TS1	Qmf-TS1	St-TS2	Qmf-TS2	mean of PSNRs	σ of PSNRs	PSNR of mean result
Lena	31.47	31.44	31.09	31.12	31.28	0.205	31.52
Barbara	25.66	25.96	27.62	27.08	26.58	0.926	26.80
Boats	29.05	29.17	29.15	29.15	29.13	0.056	29.32
House	31.08	31.05	30.56	30.75	30.86	0.246	31.14
Peppers	27.94	28.71	28.43	28.89	28.49	0.416	28.80

Table 2.1: Denoising performance as PSNRs in dB, with Gaussian noise $\sigma = 25$ (20.17dB), using two different subband decompositions and two different training sets. The columns stand for: St-TS1 (steerable subbands, training set 1); Qmf-TS1 (QMF subbands, training set 1); St-TS2 (steerable, training set 2); Qmf-TS2 (QMF, training set 2); mean of PSNRs (average of the PSNRs in the first four columns); σ of PSNRs (standard deviation of the PSNRs in the first four columns); PSNR of mean result (PSNR of the average result using all the four settings).

	Lena		Barbara		Boats		House		Peppers	
$\sigma_n/PSNR_n$	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
15/24.61	33.45	33.65	30.02	30.40	31.33	31.52	33.07	33.32	31.20	31.51
20/22.11	32.25	32.47	28.00	28.30	30.16	30.33	31.88	32.13	29.76	30.05
25/20.17	31.28	31.52	26.58	26.80	29.13	29.32	30.86	31.14	28.49	28.80
50/14.15	28.23	28.47	23.69	23.80	26.17	26.33	27.55	27.87	24.83	25.06

Table 2.2: Denoising performance as PSNR, in dB, on a few standard test images, with varying levels of noise. Two PSNRs are reported for each image, the first one (C1) being the average of four result PSNRs (St-TS1,Qmf-TS1,St-TS2,Qmf-TS2, as in Table 1), the second one (C2) being the PSNR of the average of the four results.

is not a good way of measuring subjective quality. Again we argue that textural similarity, reflected by the differences in global subband statistics, can be a valid criterion. We conducted psychophysics experiments to compare out method with the others. In this case, the image from one method at a given magnification was matched in quality to another technique at another magnification. Our method performed very well in this comparison. The best competitor was Genuine Fractals. Averaged across subjects and images, we are able to tolerate between 1.16 and 1.52 more magnification than Genuine Fractals, with 1.30 as the average.

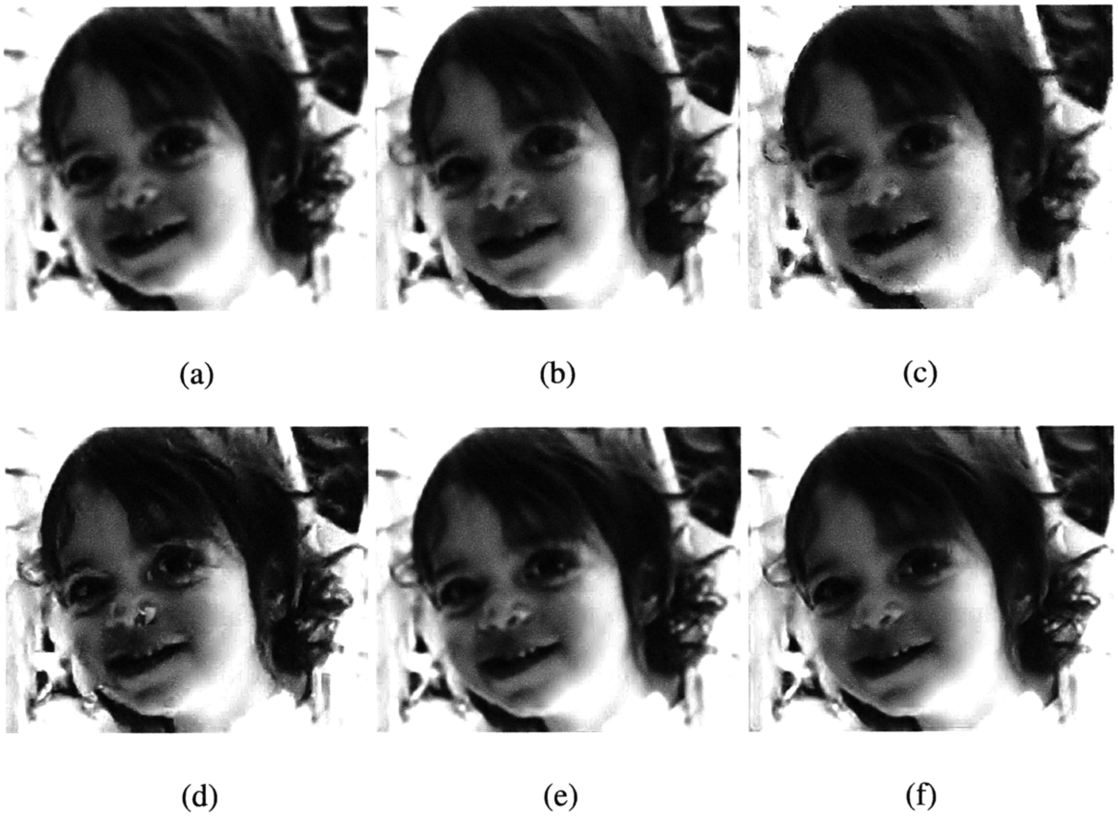


Figure 2-8: Super-resolution results. 311x258 image is downsampled by 4 in each direction, and then super resolved. The downsampling is done by bicubic interpolation. (a) Bicubic upsampling. (b) Commercial software Genuine Fractals v4.1. (c) Hertzmann *et al.*, image analogies [41]. (d) Freeman *et al.*, example-based [30]. (e) Our result, without global constraints. (f) Our result, with global constraints.

2.5.3 Intrinsic image decomposition

We apply our image mapping algorithm to another application, *i.e.*, intrinsic image decomposition. "Intrinsic images" is a term introduced by Barrow and Tenenbaum [8]. It refers to a mid-level image representation that decomposes an image into its "intrinsic" components, each of which is represented as a separate image. The intrinsic images can be combined through point-wise operations to produce the original image. An intrinsic image can be an image for illumination at every point, an image for surface reflectance, or an image for surface shape. We are interested in a particular type of intrinsic image decomposition, i.e., shading and albedo decomposition. The shading component depends on both the illumination and the surface shape at each point, and includes highlight and shadows. The albedo component describes the fraction of light reflected from the surface at each point. The two component images, when multiplied pixel by pixel, produce the original image. Separating the two components is an ill-posed problem, because one needs to get two unknown variables out of one single measurement at each point. The human visual system is however remarkably good at discounting shading and correctly estimating surface reflectance, achieving lightness constancy [4].

In [52], Land and McCann proposed the Retinex model, in which they postulated that lightness constancy is possible because of certain constraints of the world, *i.e.*, albedo tends to be piecewise constant while shading tends to vary smoothly across space. Reflected in images, big derivatives tend to be caused by reflectance changes, and small derivatives by shading changes. The Retinex algorithm for separating shading and albedo, goes as follows. First, take the log of an image to turn the multiplication into addition. Then, take the derivatives of the image, and threshold the derivatives in order to differentiate derivatives caused by albedo changes and those caused by illumination. Finally, integrate the derivatives considered to correspond to albedo changes to reconstruct the albedo image. Land and McCann used a "Mondrian world", where the albedo image consists of a collage of uniform-reflectance patches and the shading image consists of slowly varying illumination.

The Retinex algorithm is effective for this situation, but the real world is far more complex than the Mondrian world. The shading image could contain abrupt edges, for example, when there are concentrated highlights or sharp-edged shadows. Tappen *et al.* [95, 94] brought single-image shading/albedo decomposition to real-world images that are significantly more complex. In [95], they train a classifier that classifies each image derivative as being caused by reflectance change or shading change. They use Generalized Belief Propagation to propagate labels from pixels where the classification has high certainty to pixels where the classifier has low certainty. In [94], Tappen *et al.* formulate the problem as a non-linear regression problem, and solve it via Mixture of Experts.

Our method has the same flavor as [94], in that we aim to estimate a number of linear constraints on pixel values, in this case the Haar coefficients, through regression, and then reconstruct the result image from the estimated coefficients, in the least-square sense. The differences between our method and [94] lie in the choice of input features and the regression method. Tappen *et al.* use Laplacian pyramid coefficients as input features, while we use the Haar coefficients and energy values (Haar coefficients rectified and blurred) as input features. We choose the Haar filters because unlike the Laplacians, which are center-surround, the Haar filters are directional, and may reveal more information, such as differences between edges and junctions. Such differences offer important cues about the shading and reflectance at a point [4]. We also use a different regression method. Tappen *et al.* use a Mixture of Experts estimator, which contains a number of "prototype patches" added sequentially with the goal of minimizing a cost function on the training data. We use the Nested Binning Regression algorithm described in Section 2.3. We use the Haar coefficients and local energy values (on two successive scales, without subsampling) as features, and the constraints to estimate are one-level Haar coefficients of the reflectance image.

There is a very important concept from Tappen *et al.*'s work [94] that we borrow, that is, "weights to constraints". Each estimated coefficient constitutes a linear constraint on the pixel values of the reflectance image, and there is uncertainty associated with each such constraint. A junction can lead to a shading/albedo interpretation with far higher confidence than an edge [4, 90]. Incorporating confidence measures can help propagate high-confidence interpretations to places with low-confidence interpretations. Tappen *et al.* [94] estimate the weights through learning Gaussian Conditional Random Fields. For us, such uncertainty measures come readily out of Nested Binning Regression.

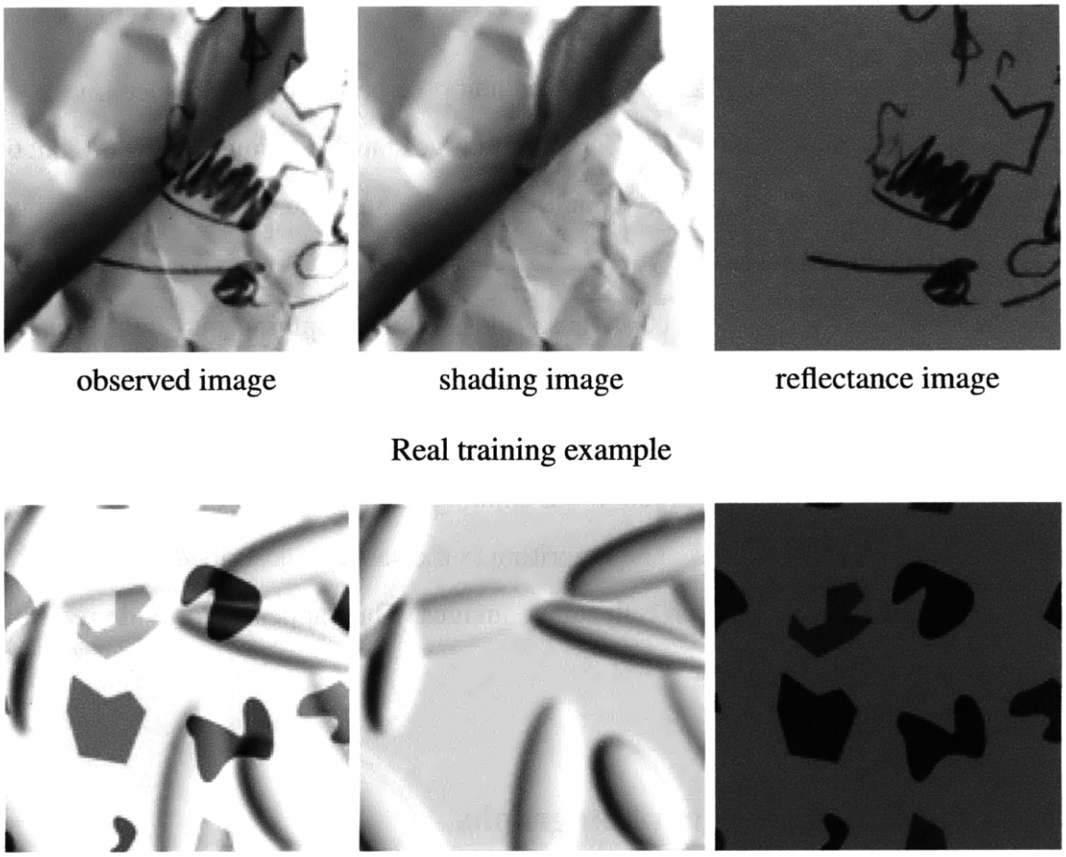
In order to estimate uncertainty, we calculate the mean squared error of regression in each bin from all the training examples falling within, which then serves as the uncertainty measure for every test example falling within that bin. Putting it into the Markov Random Field framework described in Section 2.2.1, this mean squared error of regression is the $\sigma_{t,k}^2$ in Equations 2.3, 2.5, and 2.6. More accurately, $\sigma_{t,k}$ should be written as $\hat{\sigma}_{t,k}(X)$ because it depends on the input image X and is estimated during regression. After getting $\hat{\sigma_{t,k}}$ and $\hat{b}_{t,k}(X)$ for each pixel (t,k), the log probability in Equation 2.6 can easily be maximized through a pseudo-inverse operation. We use the set of training data provided by Marshall Tappen (*http : //www.cs.ucf.edu/ mtappen/shading_data.zip*), which includes synthetic data of shaded ellipses and reflectance patterns, and real data of ground-truth shading and albedo decompositions obtained by taking the Red and the Green channels of crumpled white paper with green marker scribbles. The Red channel contains both shading and albedo, while the Green channel only contains shading. We show one example of a synthetic training pair, and another of a real training pair, in Figure 2-9.

Results of shading and reflectance decomposition using our algorithm are compared with results of Tappen *et al* [94] in Figure 2-10. The albedo image recovered using our method, shown in Fig-2-10(d), seems to have less shading residue. The sum of squared error on our estimated shading image is 1.69×10^6, smaller than those of Tappen *et al.*'s, which are 8.4×10^6 for Fig-2-10(b) and 2.4×10^7 for Fig-2-10(c), respectively.

To address the problem of albedo residue in our estimated shading image, we add a post-processing step that penalizes the accidental coincidence of shading and albedo derivatives. This is related to the "generic view assumption" [9, 28]: the perfect alignment of an albedo edge and a shading edge would require special lighting and viewing conditions and thus is not very likely. The accidentalness penalty is implemented as follows: if the magnitude of the estimated Haar coefficient for the albedo component is over 75 percent at a point, increase it to 100 percent. We show the effect of this added penalty in Figure 2-11. In the checker-shadow illusion created by Edward Adelson, checks A and B are of the same gray value but appear to be very different in lightness to a human observer. One explanation for this illusion is that a human observer can discount the lighting and estimate the albedo of check A to be lower than that of check B. The albedo images estimated using our method, shown in the right column of Fig-2-11, reflect this. We show more results in Fig-2-12.

2.5.4 Lightness illusions

Intrinsic images have been proposed for understanding lightness illusions [5, 1, 4]. Lightness illusions are cases where lightness (perceived reflectance) is different though luminance is identical. Checker-shadow (Fig 2-11) is an example of lightness illusion: check A is perceived to be much darker than check B, though A and B are of equal luminance. Many other remarkable illusions are demonstrated and analyzed in [4]. Some illusions with seemingly minor modifications become much stronger or weaker, as seen with variants of the Koffka rings in Fig 2-13, and variants of the snake illusion in Fig 2-14 (reproduced



observed image shading image reflectance image

Real training example

observed image shading image reflectance image

Synthetic training example

Figure 2-9: Examples of training data for intrinsic image decomposition, provided by Marshall Tappen.

from [4]).

Using intrinsic image decomposition, the illusions can be interpreted as the result of the human visual system discounting illumination and estimating surface reflectance. Adelson [4] argued that the lightness problem can be understood as one of statistical estimation: one constructs an optimal mapping between luminance and reflectance, given prior knowledge about distributions of reflectance and shading. Different kinds of image features, such as edges and various types of junctions, are believed to be associated with different distributions of reflectance/shading decomposition. For example, edges are ambiguous, while sign preserving X-junctions are less ambiguous as they are almost always associated with transparency. Information also needs to be spatially propagated from low-ambiguity locations to high-ambiguity locations.

Our learning algorithm explicitly learns the mapping between a luminance image and a reflectance image from a training set. Can the intuitive statistics about edges and junctions in [4] be obtained from training data? Can the differing strengths of modified illusions be predicted by the learned statistics?

We try to address some aspects of these questions by applying our algorithm to the interpretation of the modified Koffka rings and snake illusions. In order to incorporate transparency which is important for lightness perception, transparency patterns, with an example shown in Fig 2-15, are added to our existing intrinsic image decomposition training set described in section 2.5.3. The algorithm is the same as described in section 2.5.3. We show in Fig 2-16 the estimated reflectance images. They appear to provide reasonable predictions of illusion strengths.

2.5.5 Line drawings from photographs

Generating a line drawing from a photograph is another problem that takes the form of mapping one image to another. In order to learn the mapping from a photograph to a line drawing, a set of image pairs, each containing a photo and a line drawing in register with the photo, are needed as training data. We obtain such data from [12]. A set of photo and line drawing pairs used as training data are shown in Figure 2-17. Four frontal face photographs, also taken from [12], are used as testing data (Figure 2-18). The subband decomposition used for this application is an oversampled steerable pyramid with 4 orientations and 4 spatial scales.

The mapping from photo to line drawing, learned from all training pairs, is applied to the training photos themselves, as well as the testing photos, giving line drawing estimates shown in Figure 2-19. The estimated line drawings are going in the right direction. We can utilize bigger input image patches without increasing dimensionality, by performing "cascaded learning", i.e., taking the line drawing estimates for the training photos (first three rows in Figure 2-19) as input and the original target line drawings (Figure 2-17) as output, and doing another iteration of image mapping. This process can be repeated multiple times, as illustrated in Figure 2-20. The multiple layers of mappings obtained through such a process of cascaded learning can be applied to the testing photos, giving line drawing results shown in Figure 2-21.

2.6 Discussion

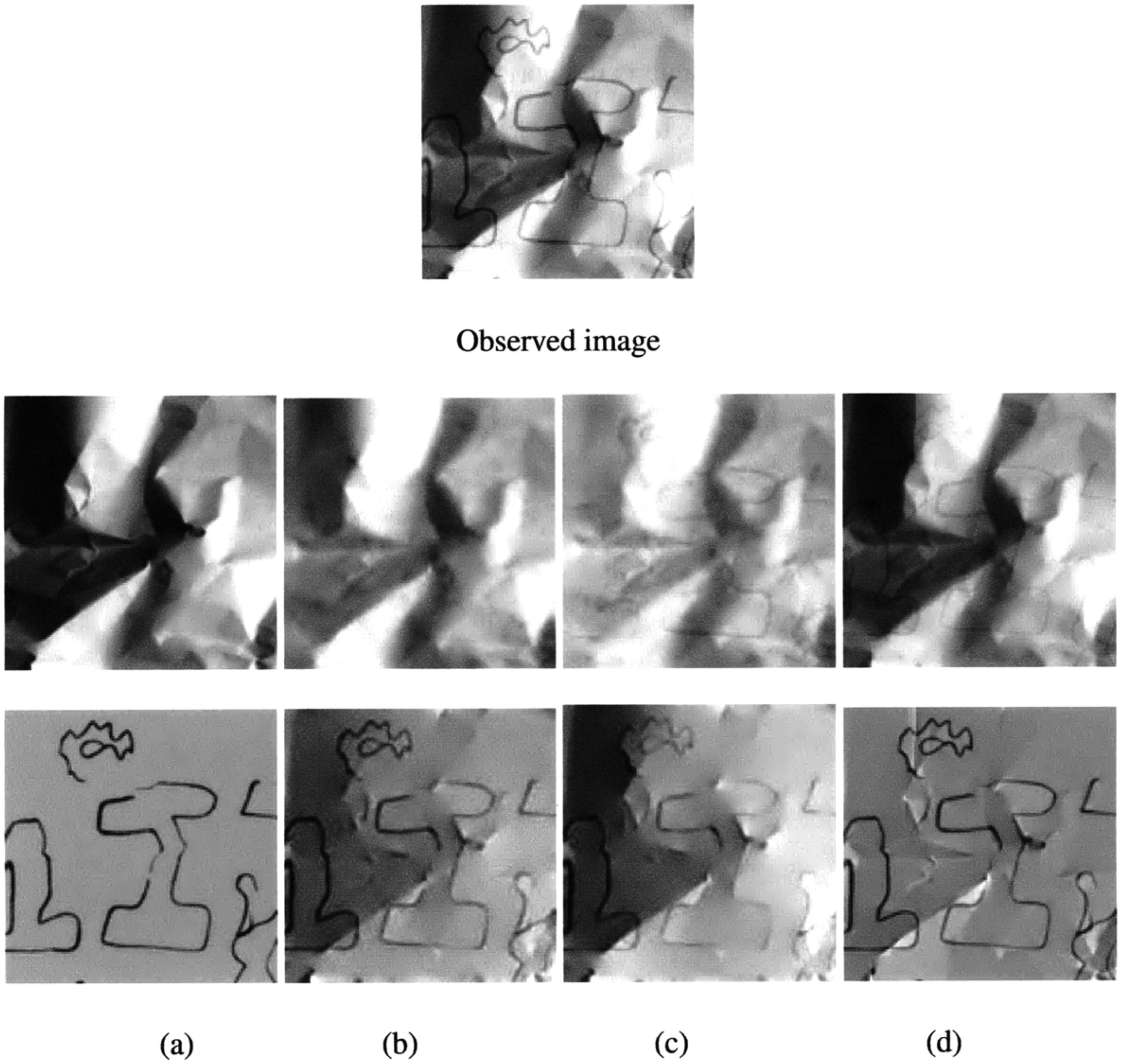
It is interesting to ask what it is that has been learned by Nested Binning Regression, and how it relates to the earlier, more intuitive models for denoising and intrinsic image decomposition.

For denoising, Nested Binning Regression can be thought of as learning a multi-dimensional "coring" function from data. Coring techniques [88, 19, 20] make use of the statistical constraint that natural images tend to consist of piecewise constant segments interspersed with edges. Large subband coefficients are more likely to be caused by the image signal, while small subband coefficients are more likely to be due to noise. Therefore, in order to reduce noise, large coefficients are kept unchanged while small coefficients are reduced in magnitude. Such a coring function is shown in Figure 2-22-(a). We compare it to the function learned using Nested Binning Regression, shown in Figure 2-22-(b). Instead of a thin curve, Fig-2-22-(b) is fat and cloudy. This is because it is a multi-dimensional function projected onto 1D. In order to illustrate the effect of additional features, we show a plot of multiple curves corresponding to different local energy values in Figure 2-23. The multiple curves indicate, for two input subband coefficients of the same magnitude, the one associated with a lower local energy is more likely to be due to noise, and should be multiplied with a smaller factor to give an estimate of the clean coefficient. The intuition is that coefficients caused by the image signal tends to be structured while those caused by noise tends to be more isolated.

Likewise, for shading and albedo decomposition, Nested Binning Regression can be thought of as learning a multi-dimensional Retinex estimator from data. The classic Retinex algorithm makes use of the constraint that the albedo image tends to consist of piecewise constant segments interspersed with with edges, whereas the shading image tends to consist of gradual changes. Therefore, in order to remove shading from albedo, derivatives with large magnitudes are kept unchanged while those with small magnitudes are put to zero. A Retinex estimator is shown in Figure 2-24-(a). The estimator obtained through Nested Regression Binning is shown in Figure 2-24-(b). It is a fat curve for the same reason as mentioned in the last paragraph, *i.e.*, it is a multi-dimensional function projected onto 1D. The x-axis is the horizontal derivative of the input image (with albedo and shading together), and the y-axis is the horizontal derivative of the output image (the albedo image estimate). To illustrate the effect of additional features, we show a plot of multiple curves corresponding to different local energy values in Figure 2-25, and another plot of multiple curves corresponding to different values of the derivative in the orthogonal orientation, in Figure 2-26.

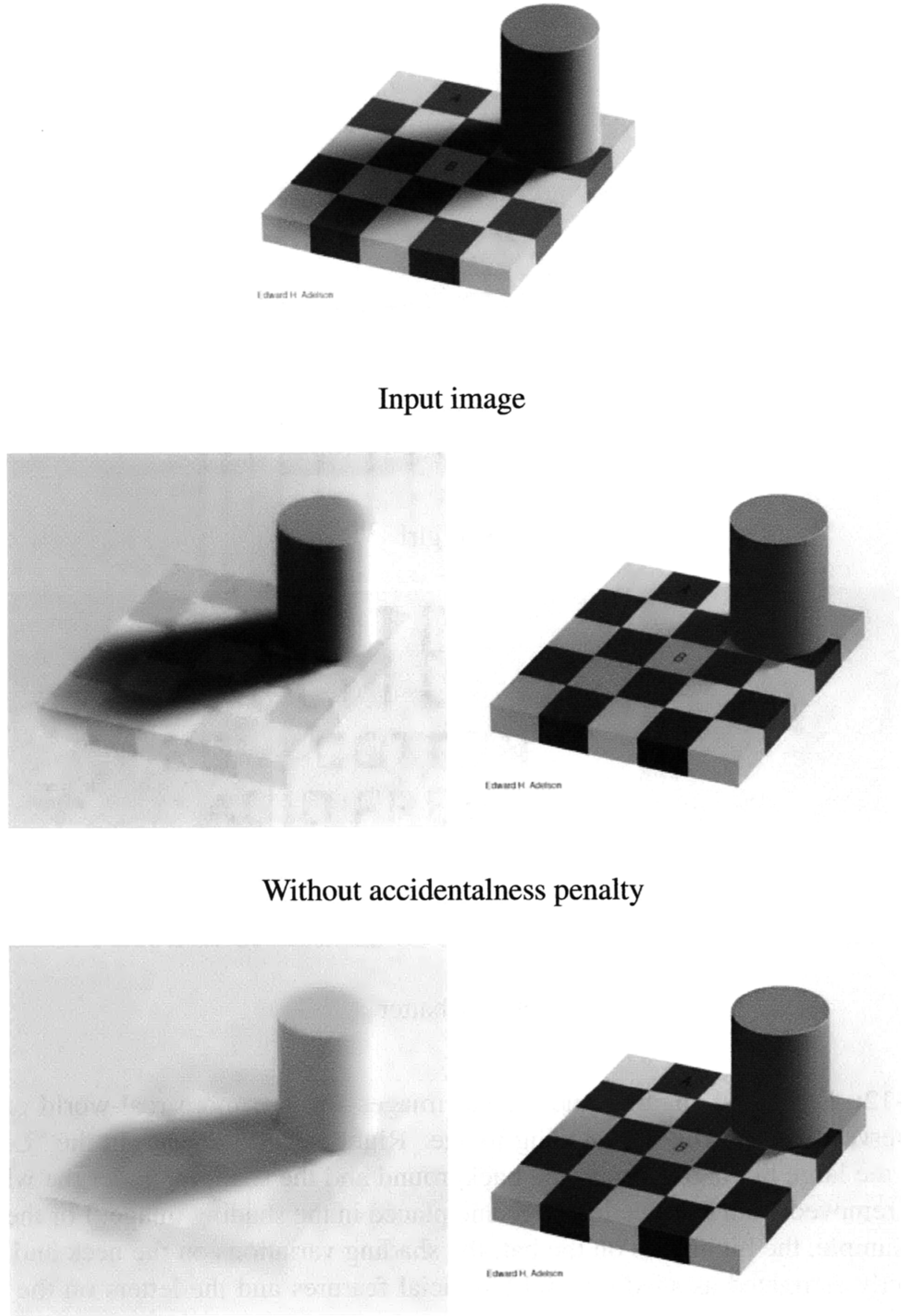
By now the reader may have noticed a lot of similarity between the denoising curves and the shading/albedo decomposition curves. In both Figure 2-22-(a) and 2-24-(a), input values with large magnitudes are kept unchanged and the ones with small magnitudes are pushed closer to the x-axis. Also, Figure 2-22-(b) and 2-24-(b) look very similar. The image models for these two seemingly very different problems are essentially the same, when viewed as one-dimensional mappings of filtered values. But interestingly, when we incorporate additional input features, there shows a big difference. The multiple curves in Fig 2-22 and those in Fig 2-24, both using local energy as the second input feature, are in reverse orders with the increase of local energy. With the increase of local energy, the denoising curves become steeper, whereas the albedo curves become flatter (closer to the x-axis), most evidently in the medium-magnitude portion of the graph. If an image derivative of medium magnitude is associated with a low local energy value, it is deemed more likely to be due to albedo than one associated with a high local energy value. This pattern at first glance may seem counter intuitive. One possible explanation is that shading-related changes in an image tend to have spatial structure, for example, with the cast shadows in the checker-shadow illusion shown in Fig 2-11, and the dark creases on crumpled paper shown in Fig 2-10. Albedo-related changes, on the other hand, can be isolated. If an image derivative has medium magnitude and is surrounded by zero derivatives and thus small local energy, it is more likely to be due to albedo than to shading. Of course, highlights could be small and isolated. But small and concentrated highlights are not included in any of our training data, and therefore are not captured by the learned model.

Figure 2-26 shows the effect of a derivative in the perpendicular orientation instead of local energy as the second input feature. The first input feature is the horizontal derivative of the observed image, and the second is the magnitude of the vertical derivative of the observed image. The output value is the horizontal derivative of the estimated albedo image. The blue curve corresponds to vertical derivatives of magnitude 0.00, cyan 12.75, and red 38.25. When the vertical derivative increases in magnitude, the curve becomes flatter. It indicates, when there is a horizontal edge (corresponding to large vertical gradients), whether caused by albedo or shading, more often than not there is no albedo change along the horizontal direction.



Observed image

Figure 2-10: The shading and albedo images generated from the observed image. (a) Ground truth shading image (above) and albedo image (below). (b) Mixture of Experts (Tappen *et al.* '06), trained using MSE (mean square error) criterion. (c) Mixture of Experts (Tappen *et al.* '06), trained using robust error criterion. (d) Ours, Nested Binning Regression. Our estimate of the albedo image appears to have less shading residue (notice the shadow near the top left) than (b) and (c).

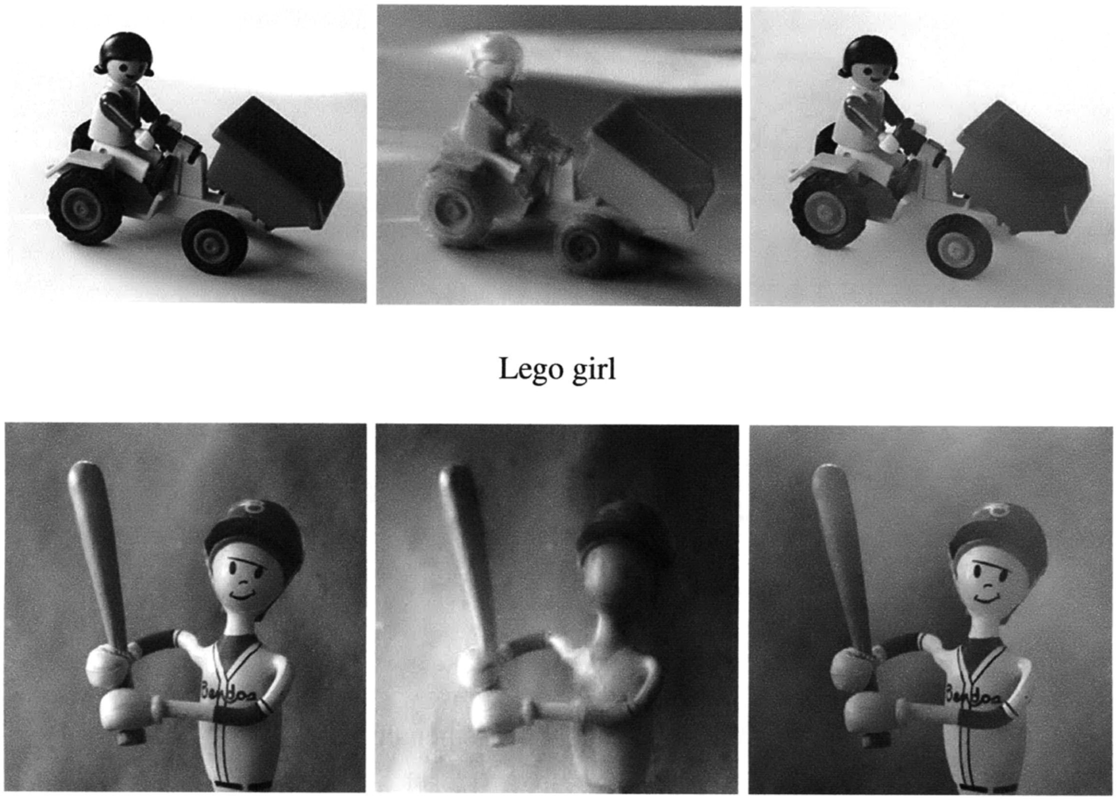


Input image

Without accidentalness penalty

With accidentalness penalty

Figure 2-11: Shading and albedo images estimated for the checker-shadow illusion image (by Edward Adelson), without and with accidentalness penalty. The shading image is much cleaner with the penalty.



Lego girl

Bendy batter

Figure 2-12: Estimated shading and albedo images for two more real-world examples. Left: observed image. Middle: shading image. Right: albedo image. In the "Lego girl" example, the large highlight area in the background and the shadows under the wheels are correctly removed from the albedo image and placed in the shading image. For the "Bendy batter" example, the highlights on the bat, the shading variations on the neck and the shirt are correctly estimated as shading, and the facial features and the letters on the shirt are correctly estimated as albedo.

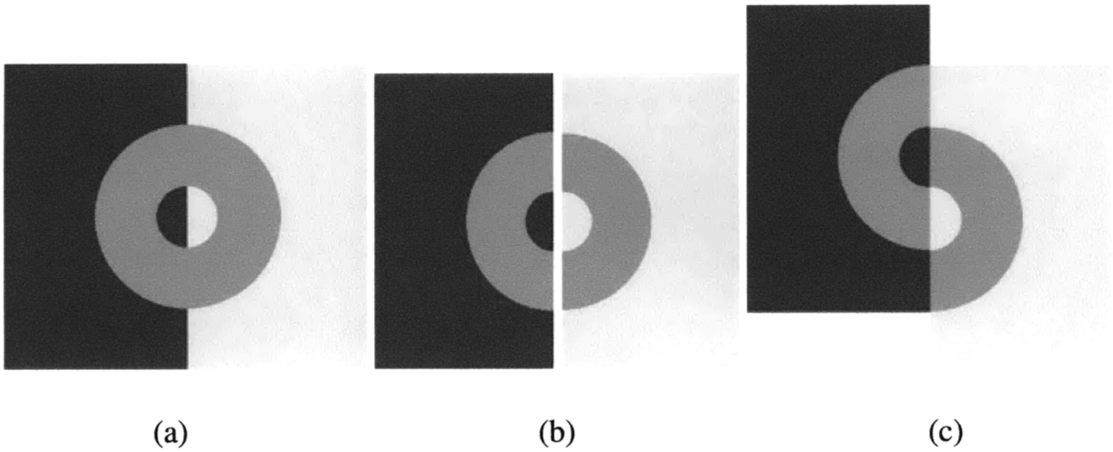


Figure 2-13: Variants of the Koffka rings. The two half rings have identical luminance, but appear to be different in (b) and (c). The illusion is stronger in (c) than in (b). Reproduced from Adelson [4].

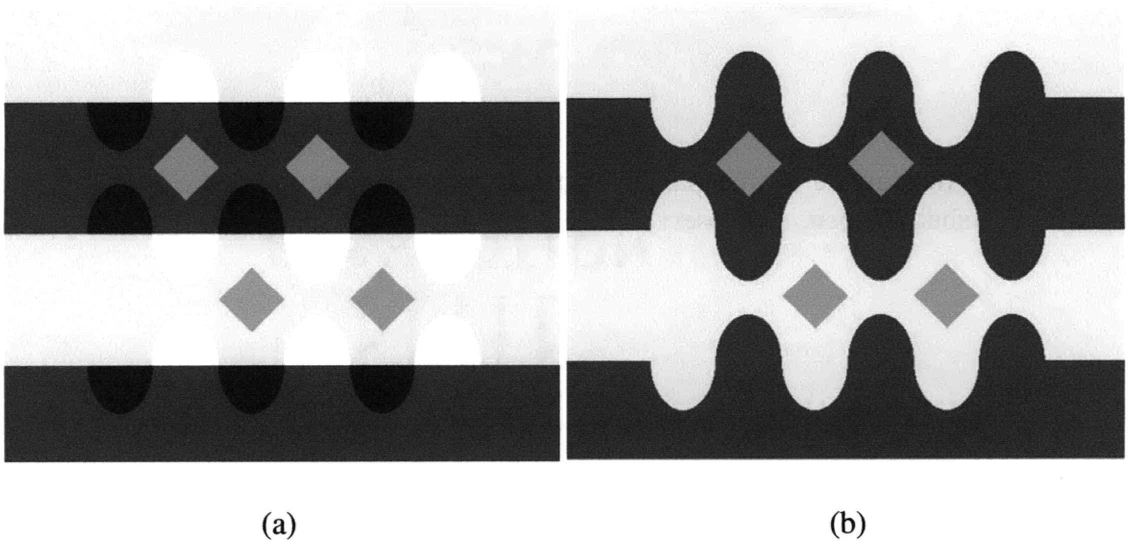


Figure 2-14: The "snake" and the "anti-snake" illusions. All diamonds have identical luminance, but appear to be very different in (a). In (b), the illusion is much weaker. Reproduced from Adelson [4].

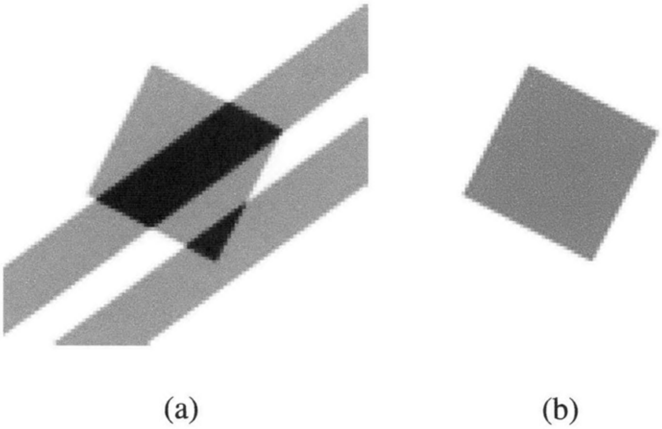


Figure 2-15: An example of transparency patterns, used as extra training examples. Provided by Marshall Tappen. (a) Observed image. (b) Reflectance image.

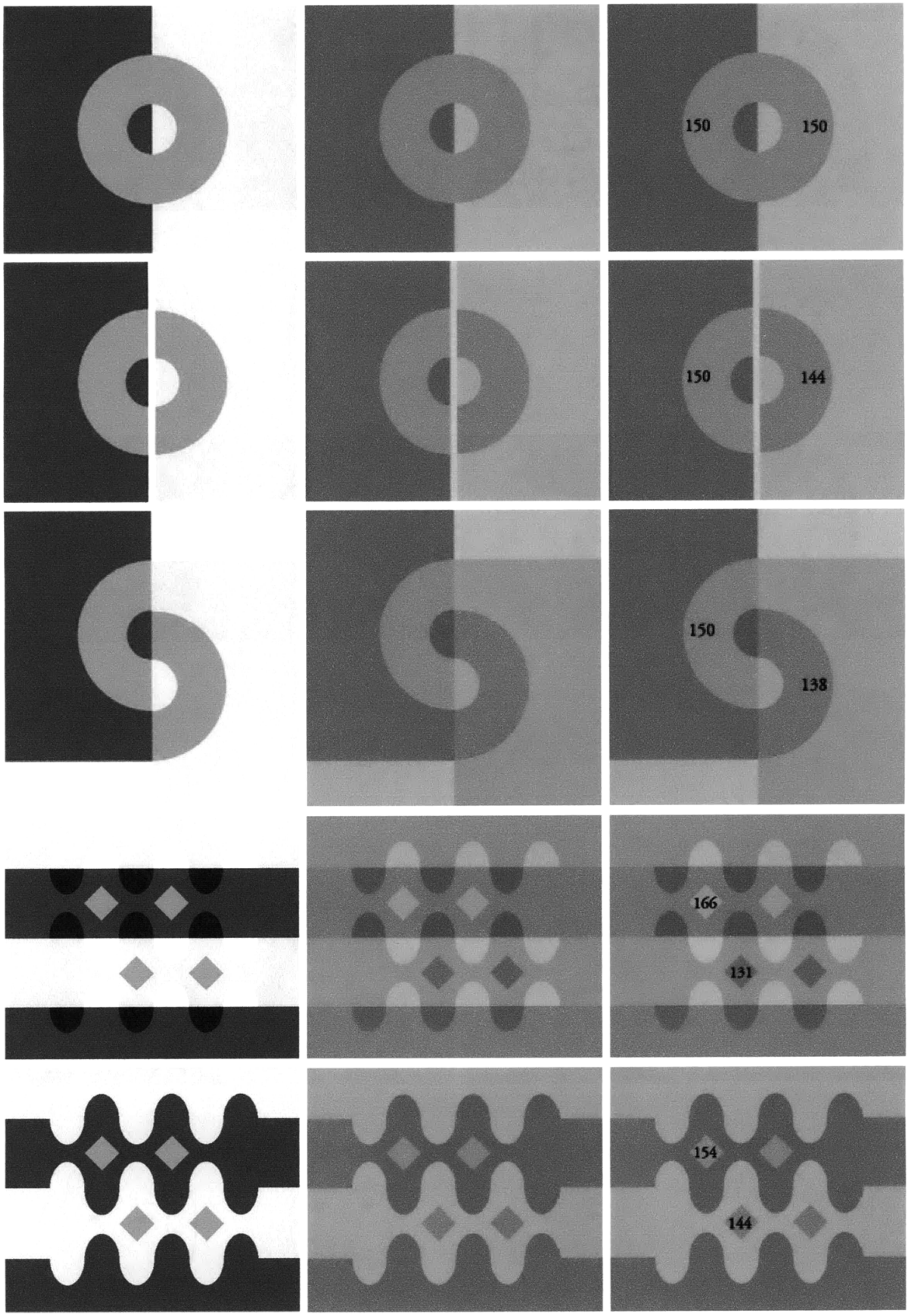


Figure 2-16: Results for interpreting lightness illusions as reflectance estimation. Left column: observed images. Middle column: estimated reflectance images, using our learning algorithm. Right column: the same images as shown in the middle column, but with estimated reflectance values marked on the rings and the diamonds. This demonstrates reasonable predictions of illusion strength by the algorithm.

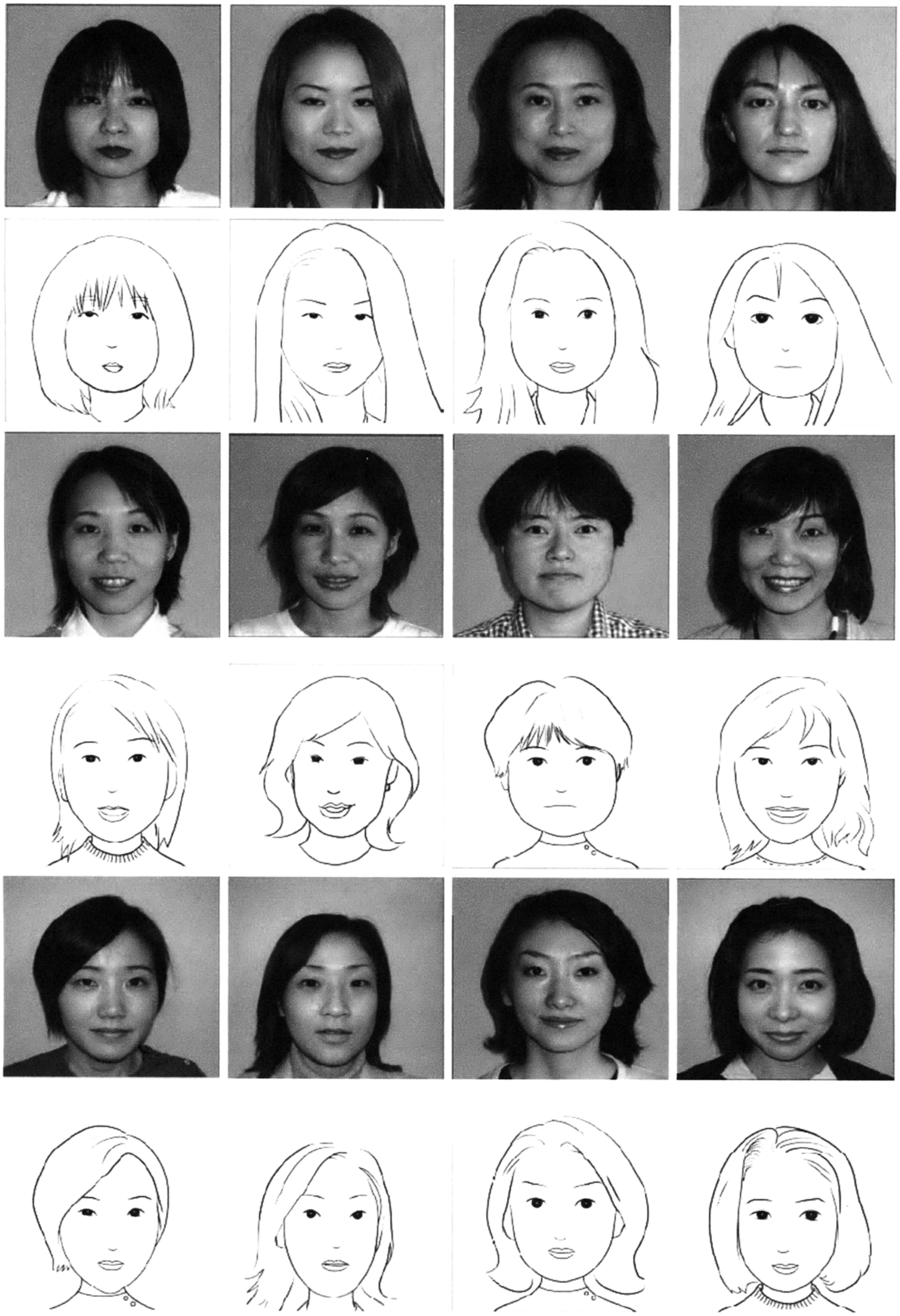


Figure 2-17: Training data for generating line drawings from photographs. Data source: [12].

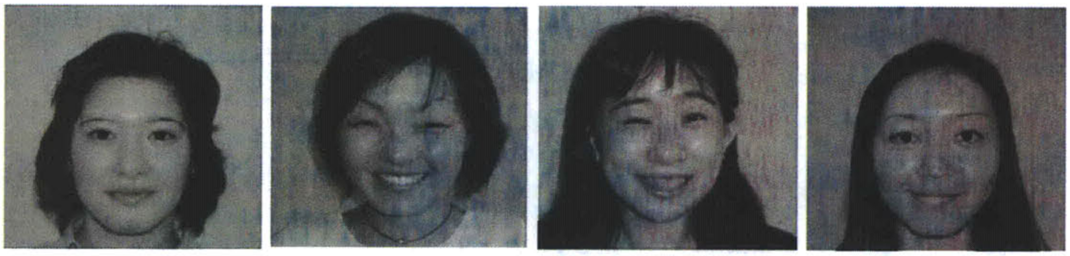
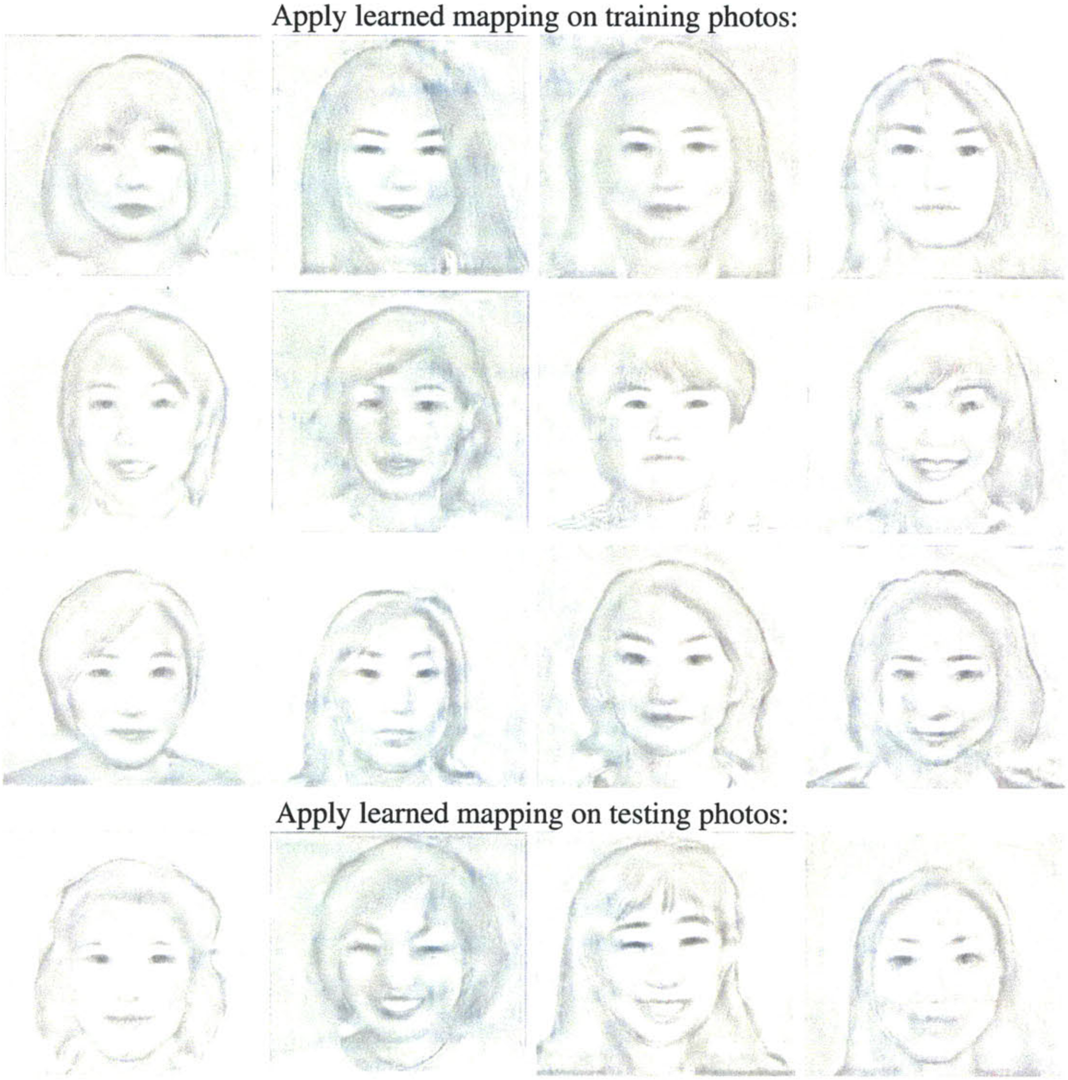


Figure 2-18: Test data for generating line drawings from photographs. Data source: [12].



Apply learned mapping on training photos:

Apply learned mapping on testing photos:

Figure 2-19: Results of mapping photo to line drawing (one-pass), through Nested Binning Regression.

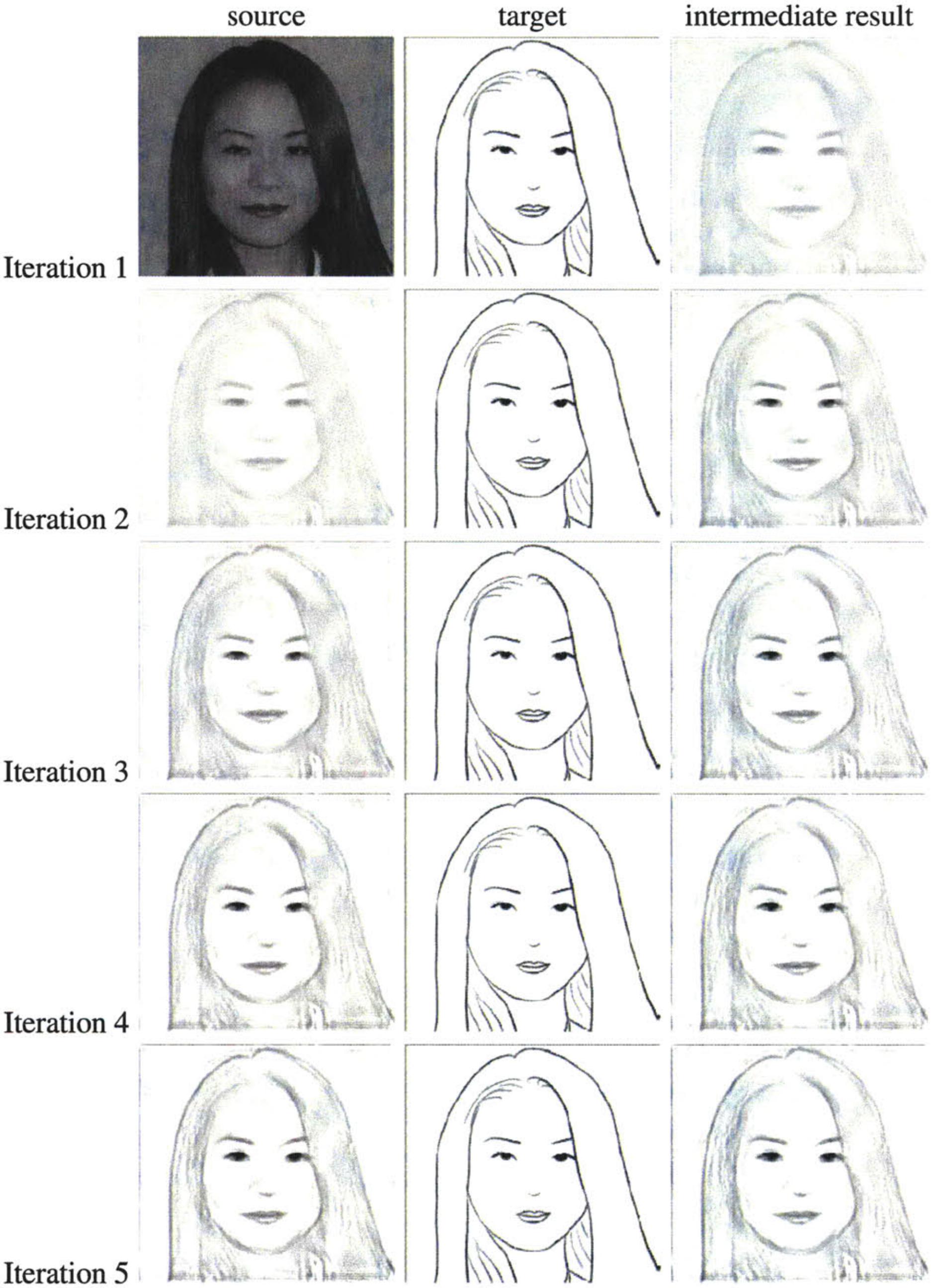


Figure 2-20: Cascaded learning. In every iteration, the mapping from "source" to "target" is learned using all training pairs, and then applied to "source", producing "intermediate result". The "intermediate result" is then used as "source" for the next iteration. For all iterations, "target" is always the line drawing originally provided in the training data shown in Figure 2-17.

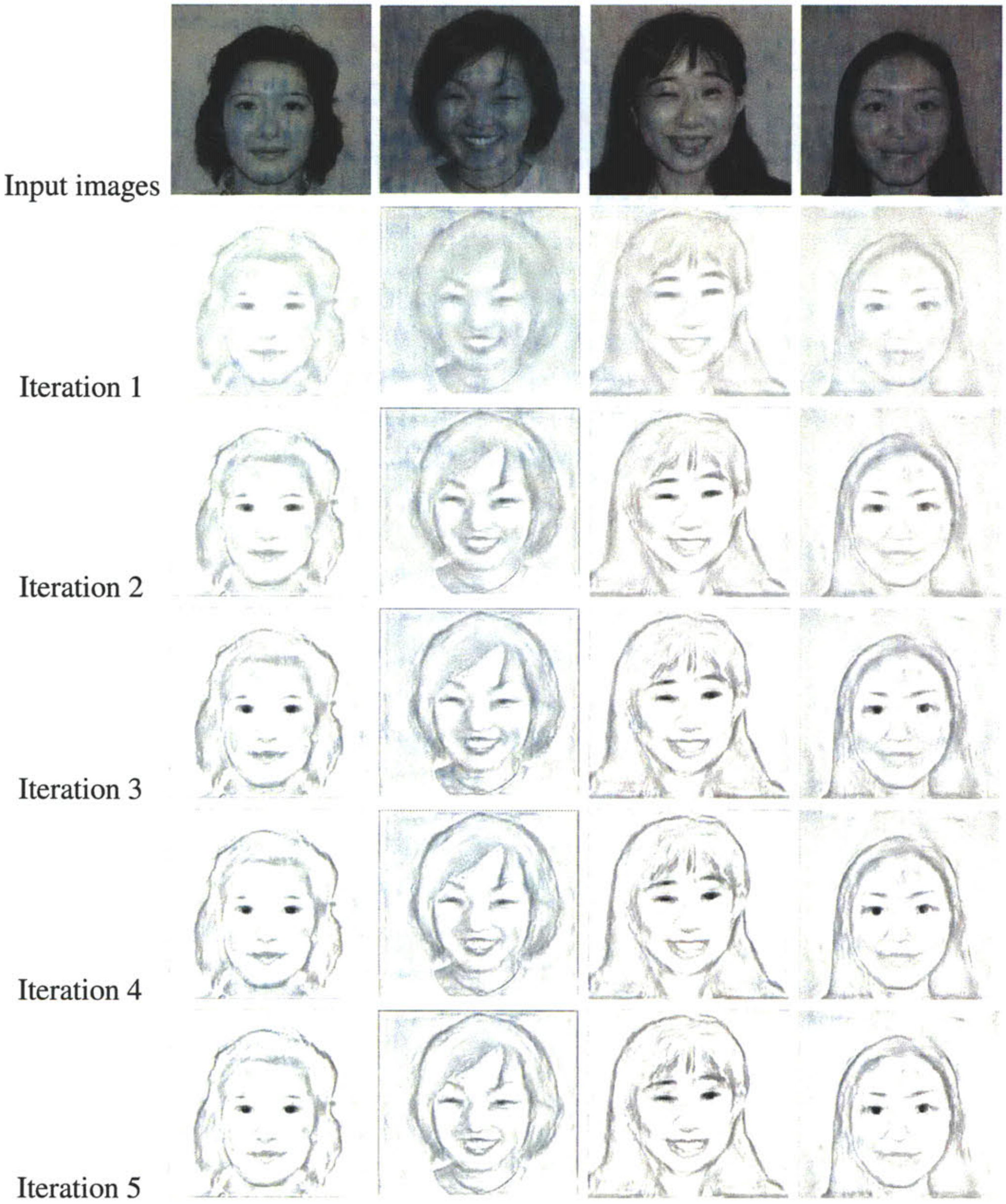
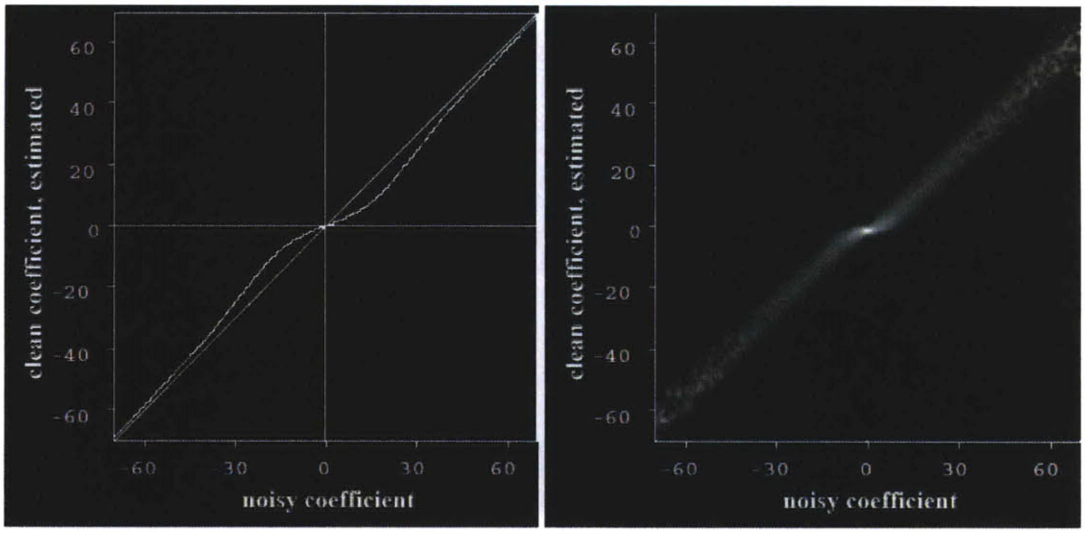


Figure 2-21: Results of mapping photo to line drawing (cascaded), through Nested Binning Regression.



(a) (b)

Figure 2-22: Coring vs. learned model for denoising. (a) Bayesian coring estimator (Simoncelli and Adelson [88]. (b) The estimator learned using our Nested Binning Regression algorithm. In (b), the varying brightness signifies the conditional probability density, and brighter means higher. (b) is not a thin curve because it's a multi-dimensional function projected onto 1D.

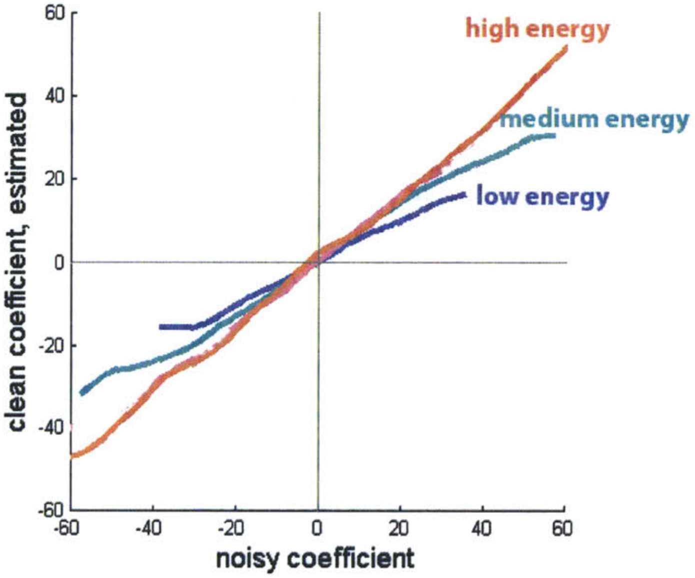
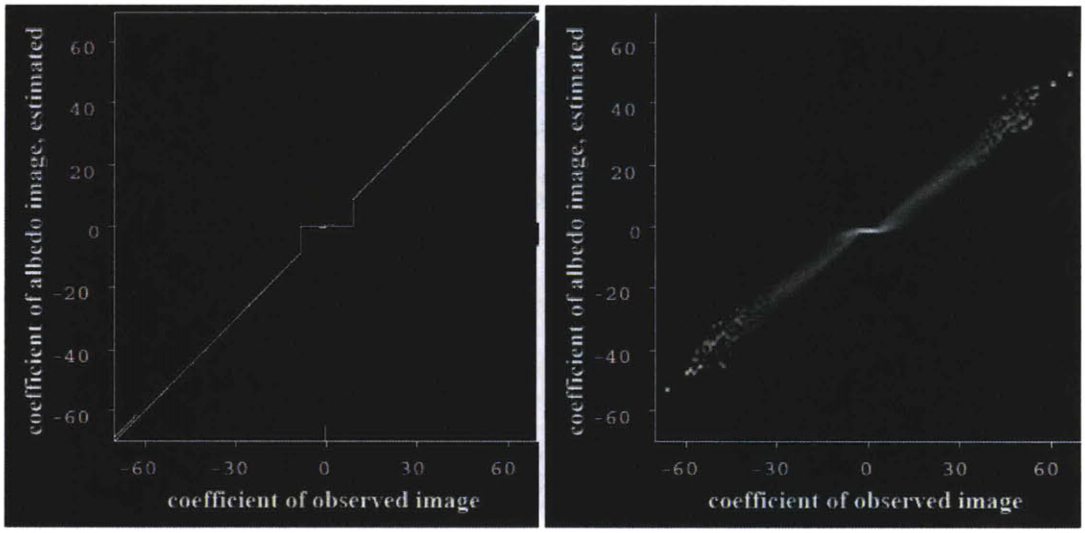


Figure 2-23: Learned mappings from noisy coefficient to clean coefficient, corresponding to high, medium, and low local energy values, respectively. The intuition is, for two input subband coefficients of the same magnitude, the one associated with a lower local energy value is more likely to be due to noise, and should be multiplied with a smaller factor to give the estimated clean coefficient.



(a)

(b)

Figure 2-24: Retinex vs. learned model for estimating albedo image from observed image. (a) Classic Retinex estimator, where image derivatives are thresholded to remove shading. (b) The estimator learned using our Nested Binning Regression algorithm. In (b), the varying brightness signifies the conditional probability density, and brighter means higher. (b) is not a thin curve because it's a multi-dimensional function projected onto 1D.

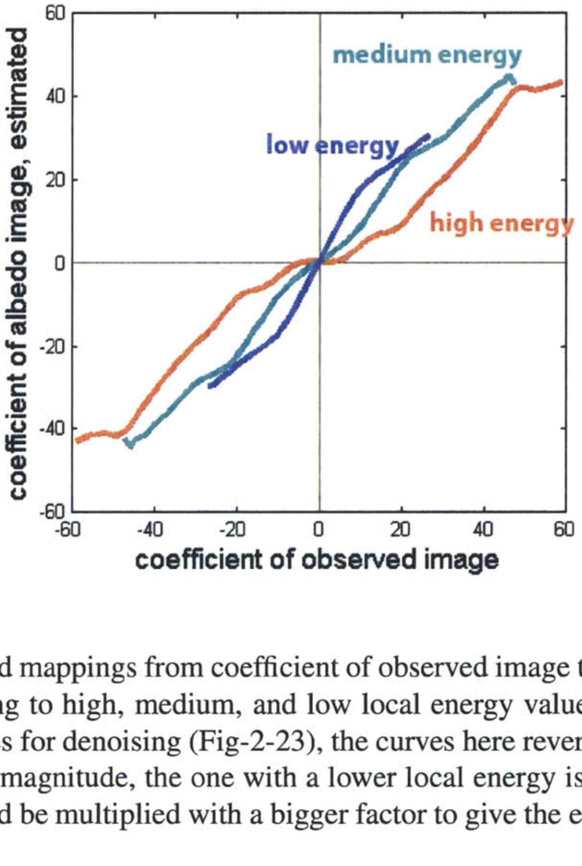


Figure 2-25: Learned mappings from coefficient of observed image to coefficient of albedo image, corresponding to high, medium, and low local energy values, respectively. Compared with the curves for denoising (Fig-2-23), the curves here reverse: for two input coefficients of the same magnitude, the one with a lower local energy is more likely to be due to albedo, and should be multiplied with a bigger factor to give the estimated albedo image coefficient.

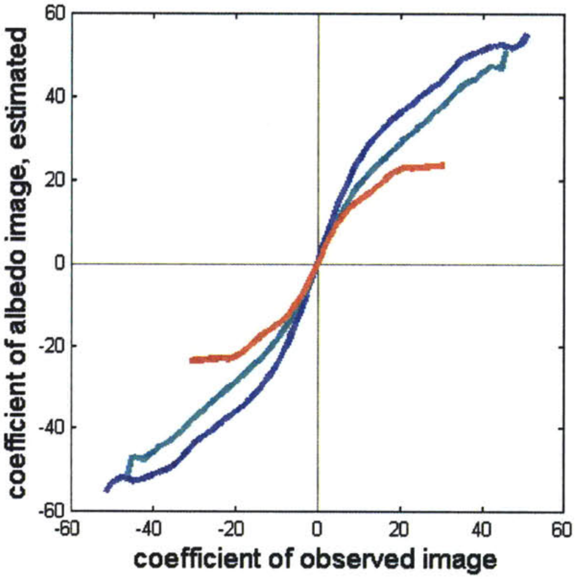


Figure 2-26: Another illustration of the multi-dimensional model learned by Nested Binning Regression, for shading/albedo decomposition. The first input feature is the horizontal derivative of the observed image, and the second is the magnitude of the vertical derivative of the observed image. The output value is the horizontal derivative of the albedo image. The blue curve corresponds to vertical derivatives of magnitude 0.00, cyan 12.75, and red 38.25. When the vertical derivative increases in magnitude, the curve becomes flatter. It indicates, when there is a horizontal edge (corresponding to large vertical gradients), whether caused by albedo or shading, more often than not there is no albedo change along the horizontal direction.

Chapter 3

Dynamic range compression and companding with multiscale architectures

High dynamic range (HDR) imaging is an area of increasing importance, but most display devices still have limited dynamic range (LDR). Various techniques have been proposed for compressing the dynamic range while retaining important visual information. Multiscale image processing techniques, which are widely used for many image processing tasks, have a reputation of causing halo artifacts when used for range compression. However, we demonstrate that they can work when properly implemented. We use a symmetrical analysis-synthesis filter bank, and apply local gain control to the subbands. We also show that the technique can be adapted for the related problem of "companding", in which an HDR image is converted to an LDR image, and later expanded back to high dynamic range.

[2] Part of this chapter (3.1-3.6) has appeared as: [61] Yuanzhen Li, Lavanya Sharan, Edward H. Adelson. Compressing and Companding High Dynamic Range Images with Subband Architectures. *ACM Transactions on Graphics (TOG), 24(3), Proceedings of SIGGRAPH* 2005.

3.1 Introduction

In recent years there has been an explosion of interest in high dynamic range (HDR) imagery. HDR image data is increasingly available from such sources such as digital photography, computer graphics, and medical imaging [106, 16, 70, 65]. Although new HDR display systems are being developed [86], the dominant display technologies, such as printed paper, CRTs, and LCDs, have limited dynamic ranges. Therefore various techniques have been developed for compressing the dynamic range of the signal so the information can be displayed effectively. Ideally, these techniques will be easy to implement, and will work automatically, with minimal human intervention. They should also avoid introducing unpleasant artifacts.

It would also be desirable to retrieve an HDR image from an LDR image with minimal degradation. In accord with audio terminology, we refer to the compression/expansion process as "companding". We will describe a technique that can, for example, turn a 12 bit/channel image into an 8 bit/channel TIFF, and later convert it back to a good approximation of the original 12-bit image. Since a great deal of hardware and software is designed around 8 bit imagery, this could have many uses. It is possible to do further data compression with JPEG, and still retrieve a 12 bit image with only modest degradations.

3.2 Previous work

The recent literature on HDR range compression has been extensively reviewed by others [98, 18, 17] and we refer the reader to these sources. The most straightforward techniques, sometimes called "global" tone-mapping methods, use compressive point nonlinearities. The image, $I(x,y)$, is simply mapped to a modified image, $I'(x,y) = f(I(x,y))$, where f is a compressive function such as a power function, or a function that is adapted to the image histogram [99, 106, 25, 53]. The dynamic range is reduced, but the contrast of details is compromised and the images can look washed out. To compress the range while maintaining or enhancing the visibility of details, it is necessary to use more complex techniques.An early technique was described by Stockham [91], who observed that the image $L(x,y)$ is a product of two images: an illumination image $I(x,y)$, and a reflectance image, $R(x,y)$. The illumination can vary greatly from region to region, which causes the dynamic range problems. Stockham estimated the local illumination as a geometric mean over a patch, and divided it out. This is equivalent to subtracting a blurred version of the image in the log luminance domain. The method unfortunately introduces artifacts known as "banding" or "halos" when there is an abrupt change of luminance, i.e., at large step edges. The size of the halo depends on the size of the blur. Multiscale techniques [47, 74, 100], including some designed to capture properties of the human visual system, have reduced the visibility of the halos but have not removed them, and the computer graphics community has therefore explored other approaches. One popular approach is to estimate the illumination level, and a corresponding gain map, with an edge-preserving blur. The notion is that the gain map should have sharp edges at the same points that the original image does, thereby preventing halos [72, 18]. Durand and Dorsey [21] achieved particularly good results by computing a gain map with the bilateral filter described by Tomasi and Manduchi [97]. They also developed methods for fast computation. An alternate approach is to work in the gradient domain, as is done in Retinex algorithms [52]. Fattal et al [24] computed a gain map for the gradient of the image, reducing large gradients relative to small ones, and then solved Poisson's equation to retrieve an image with compressed range. Solving Poisson's equation after manipulating the gradient field can be problematic, but Fattal et al developed approximations that gave visually satisfying results with reasonable computation times.

Although multiscale representations have lost favor in the computer graphics community, there is some patent literature that suggests their utility. Labaere and Vuylsteke [51] adapted Mallat and Zhong's wavelet method [64], which represents signal in terms of positions of and magnitudes of maxima of the outputs of edge-sensitive filters. By reducing the size of the high magnitude edges, the dynamic range can be controlled. Lee [54] described a method that combines multiscale processing with traditional tone mapping. First, an image is run through a point non-linearity to reduce its dynamic range. The resulting image suffers from the usual reduced visibility of edges and other details. Lee then computes a subband decomposition of the original image, and adds portions of the subbands back to the the tone-mapped image in order to augment the visibility of detail at various scales. Gain maps are used to control the amount of augmentation from the subbands. Vuylsteke and Schoeters [101] describe the use of several subband decompositions, including Laplacian pyramids, wavelets, and Gabor transforms, along with sigmoidal nonlinearities to limit the amplitude of the subband signals. This approach is effective, but can introduce distortions including haloes. We have explored a set of methods with a similar structure, in an effort to achieve good range compression with minimal artifacts.

3.3 Subbands and nonlinear distortion

There are many ways of building subband systems for decomposing and reconstructing images. Each has its advantages and disadvantages. Here we discuss how this choice interacts with the problem of dynamic range compression.

For simplicity, we start by considering continuous signals. A simple multiscale decomposition is shown in Figure 3-1(a). A signal, $s(x)$, is split into a set of bandpass signals, $b_1(x), b_2(x), \ldots$ with filters $f_1, f_2, \ldots$ chosen so that the original signal can be reconstructed by directly summing these bandpass signals:

$$s(x) = \sum_n b_n(x)$$

A nonlinearity, labelled “NL”, can be imposed on the bandpass signals before summation.

Suppose that the filters consist of difference-of-Gaussians, with scales increasing by factors of two. Figure 3-2(a) shows a step edge, along with four subbands (Figure 3-2(b)) when decomposed using this filter bank. The full set of subband signals can be summed to retrieve the original input signal.

To limit the amplitude of strong edges, we can limit the amplitudes of the strong subband responses to these signals. If a particular subband signal is $b(x)$, then a soft limit can be imposed with a sigmoid, e.g., $b'(x) = \frac{b(x)}{(b(x)+\varepsilon)^{2/3}}$ (ε is a constant, if equal to 0 then $b'(x)$ is the cube root of $b(x)$). Figure 3-2(d) shows a picture of the nonlinearity, and Figure 3-2(f) shows the result of imposing it on one of the subbands. The peaks are flattened, and the low values are expanded. This prevents $b'(x)$ from being too large, but it also leads to a distortion in its shape. When the subbands are summed, they produce a distorted signal.

To get better results we need to reduce the distortion of the subband signals. There are various ways to do this, either by modifying the way that signal strength is controlled (gain control), or by modifying the filter bank architecture. We will discuss both.

3.3.1 Smooth gain control

It is useful to think of the sigmoid as controlling the gain at each location. The gain is low for high values and high for low values. In the case considered above, the effective gain, $G_1(x)$ is shown in Figure 3-2(e). It dips twice, at the two extrema of the signal. The compressed subband signal, can be expressed as $b'(x) = b(x)G_1(x)$. The rapid variation of $G_1(x)$ is the cause of the distortion of the compressed signal $b'(x)$.

To prevent the rapid variation in gain, we can simply compute a new gain signal (gain map) and force it to be smooth. If the gain varies more slowly than the subband signal itself, then there will be reduced distortion. In Figure 3-2(h), we have constructed a smooth gain signal, $G_2(x)$, by taking the absolute value of the subband signal and blurring it. The compressed subband signal $b''(x)$ is shown in Figure 3-2(i). It is almost the same shape as $b(x)$, but attenuated in amplitude.

The use of smooth gain maps leads to a major reduction in artifacts, and is one of the most important improvements one can make in a subband scheme. The details of computing gain maps for range compression are discussed in section 3.3.3 and 3.3.4.

The implementation of the subband decomposition is also important, as will be discussed in section 3.3.2.

3.3.2 Analysis-synthesis filter banks

The filter bank above is conceptually simple, but in many applications a different architecture is preferred. Figure 3-1(b) shows an analysis-synthesis filter bank, in which one set of filters, $f_1, f_2, \ldots$, called the analysis filter bank, is used to split the signal $s(x)$ into bands $b_1(x), b_2(x), \ldots$ and then another set of filters, $g_1, g_2, \ldots$, called the synthesis filter bank, is applied to those band signals $b_1(x), b_2(x), \ldots$ to produce signals $c_1(x), c_2(x), \ldots$. These post-filtered band signals $c_1(x), c_2(x), \ldots$ are summed to reconstruct the original signal $s(x)$. It is common for the filter bank to be constructed symmetrically, so that the synthesis filters are essentially the same as the analysis filters. Nonlinear distortions generally produce frequencies outside the original subband, and these will tend to be removed by the corresponding synthesis filter. The signal is forced into its proper frequency band before summation, which reduces distortion.

Analysis-synthesis filter banks are often implemented with hierarchical subsampling, leading to a pyramid. Wavelets and quadrature mirror filters (QMFs) are often used this way, in which case they yield orthogonal transforms. This is most easily explained by starting in 1-D and using the Haar wavelet pair, which consists of a lowpass filter $f_0 = [1,1]$ and a highpass filter $f_1 = [-1,1]$. In Figure 3-1(c), an input signal $s(x)$ is split into a low band and a high band by convolution with f_0 and f_1. The filter outputs are subsampled by a factor of two, meaning that every other sample is dropped. If the input has N samples, each subband will have $N/2$ samples (sometimes called subband coefficients). The subbands are now upsampled by a factor of two by inserting a zero between each sample. Each of these zero-padded subband signals is convolved with a second filter, which is $g_0 = [1,1]$ for the low band and $g_1 = [1,-1]$ for the high band. These signals are summed, and the original is reconstructed exactly.

If the same bandsplitting and subsampling procedure is applied to the lowpass signal, as shown in Figure 3-1(d), and the process is iterated, we have a Haar pyramid. The number of samples falls by 1/2 at each stage. The effective spatial scale of the corresponding highpass filter doubles, and the effective peak spatial frequency halves.

In 2-D, the process can be applied separably in the x and y directions. This leads to three highpass filters and one lowpass filter at each stage, with a subsampling by a factor of 2 in each dimension.

The subsampled pyramids are highly efficient in terms of computation and representation, because the number of samples falls by half in each dimension at each level. The subsampling can lead to problems with aliasing. In the absence of nonlinearities, the aliasing from one subband cancels that from the others, by construction. However, if nonlinearities are imposed, the aliasing cancellation no longer holds. Since range compression inherently involves nonlinearities, this is a concern.

A straightforward solution is to avoid the subsampling altogether. The doubling of spatial scale is achieved by spreading the filter taps and padding with zeros, so that $f_1 = [1,-1]$ becomes $[1,0,-1]$ and then $[1,0,0,0,-1]$ on succeeding stages. f_0 is padded in the same way, and by combining f_0 and f_1 separably in the x and y directions we get four 2D zero padded filters (hi_x, hi_y, hi_{xy}, lo in Figure 3-1(e)). The synthesis filters are basically the same, also combining f_0 and f_1 separably, except that f_1 is temporally reversed. This means that the transform is highly overcomplete, but the math still works out so that the output is a replica of the input, if no operations are performed on the subband signals. This oversampling technique is commonly used in denoising.

The Haar filters that we have used in the above discussion are not very frequency selective, and so don't cleanly separate the information in the subbands. Vuylsteke and Schoeters [101] specifically eschew the Haar filters due to their poor bandpass characteristics. However, they are the easiest filters to explain and to implement. We find that they can produce surprisingly good results when coupled with the appropriate modifications.

Since step edges are such important stimuli, one might assume that the best filters would be those that are specifically responsive to edges, i.e., odd-symmetric filters such as first derivatives. Retinex and other gradient domain methods have this attractive property, and both the Lee [54] and the Labaere and Vuylsteke [51] patents advocate the use of the Mallat and Zhong wavelets, which are discrete derivatives on the analysis side and more extended edge operators on the synthesis side. However, we have found that even-symmetric filters such as Adelson et al's 9-tap QMFs [2] performs very well on this task, often giving more pleasing results than the Haars. Note that these QMFs have much better frequency tuning than the Haars.

It is interesting at this point to compare the Haar bandsplitting approach to the gradient domain approach used by Fattal et al [24], in the simple case of 1-D signals. In both cases the signal is convolved with the filter $f_1 = [-1,1]$, which is a discrete derivative operator and emphasizes the high frequencies. In the case of the one stage Haar, there is a second filter path containing the low frequencies passed by the filter $f_0 = [1,1]$. Reconstruction (the inverse transform) involves convolutions and summation using matching filters. By contrast, in the gradient (derivative) domain, although the gradients are modified in a multi-scale fashion, there is no second signal "containing" the low frequencies. All the information (except DC) is carried in the highpass signal, and the inversion process implicitly involves amplification of the low frequencies.

The Laplacian pyramid is another example of a subsampled system with analysis and synthesis filters. Note, however, that it is not symmetrical. The analysis filters are bandpass, and the synthesis filters are lowpass. Thus the synthesis filters can remove high frequency artifacts introduced by nonlinear processing, but not low frequency artifacts. It is possible to use the Laplacian pyramid architecture without subsampling, which reduces aliasing effects, though the asymmetry remains. When nonlinearities introduce distortions that show up in low frequencies, the synthesis filters cannot remove them.

In summary, there are many ways to build subband systems, and they will deal with nonlinear distortions differently. It is generally better to oversample, in order to avoid the introduction of aliasing artifacts. It is generally better to use an analysis-synthesis filter bank, with the nonlinear operations sandwiched in the middle. A symmetrical analysis-synthesis filter bank, in which the synthesis filters are tuned to the same frequency band as the analysis filters, will be especially effective in controlling the nonlinear distortions.

3.3.3 Automatic gain control

As noted above, it is advantageous to use a smooth gain map to control the strength of the subband signals. For ideas on creating this map, it is interesting to consider the use of gain control in the human visual system.

The human visual system confronts a serious challenge with dynamic range in everyday life. The neurons in visual cortex have a low dynamic range, and they are noisy, so it is important to keep them within an optimal operating range whenever possible. The first type of automatic gain control happens at the retina, where the photoreceptors rapidly adapt to the ambient light level. For our purposes this process can be crudely modeled as taking the log of the input intensity. Another type of gain control occurs in the first stages of visual cortex, area V1, where it is known as "contrast gain control" or "contrast normalization" [38]. Responses to moderately low contrasts are boosted, while responses to high contrasts are reduced. This makes good use of the information capacity of the neurons by keeping the responses away from the ceiling and floor. Area V1 has neurons tuned to different orientations and spatial frequencies, and can be thought of as doing a local subband decomposition using filters similar to wavelets. The gain of a given neuron is controlled by the activity level of many neurons in its immediate neighborhood. Additionally, the gain is not just controlled by neurons tuned for the same orientation and spatial scale; rather, the gain signal involves a pooling of multiple orientations and scales.

The gain control varies from point to point depending on the activity, so we can think of it as forming a gain map in register with the subband image. This is analogous to Fattal et al's gain map applied to the gradient image.

In building gain maps for range compression, we first construct an activity map from local filter responses. Since the responses can be positive or negative, we take the absolute value. We then pool over a neighborhood with a simple blur. The activity map is then converted to a gain map, which has lower gain in regions of high activity.

Here is a more detailed description of the construction of a gain map. In a standard separable n-level subband pyramid there are $3n+1$ subband images, and they are denoted as $B_i(x,y)$ $(i = 1,\ldots,3n+1)$, where $B_{3n+1}(x,y)$ is the lowpass residue. We rectify each subband image B_i by taking the absolute value, and then blur it with a Gaussian kernel to get an activity map:

$$A_i(x,y) = g(\sigma) * |B_i(x,y)| \tag{3.1}$$

The size of the Gaussian kernel is proportional to the subband's scale. If the kernel used for the subbands at the finest scale has variance σ_1, then the kernel for the subbands at the next coarser level will be twice as big.

The nonlinear function $f()$ used to derive a gain map from an activity map, should be monotonic decreasing, turning the gain down where the activity is high and up where the activity is low. There are various choices as of the particular form of $f()$. One of them gives gamma-like mapping:

$$G_i(x,y) = f\{A_i(x,y)\} = \left(\frac{A_i(x,y)+\varepsilon}{\delta}\right)^{(\gamma-1)} \tag{3.2}$$

where γ is a compressive factor between 0 and 1, ε is a noise level related parameter which prevents the noise from being blown up, and also prevents singularities in the gain map, considering the power $(\gamma-1)$ is below zero. δ can be understood as a gain control stability level: the gain is turned up for places where activities are below δ and turned down for places where activities are above δ, in either case bringing the activities closer to δ.

Since we are modifying each subband separately, it is possible that gains at different scales will be mismatched at important features, leading to distortions of these features. Therefore we need a method that keeps the gains matched. Similar to the method proposed by Fattal et al. [24], we set the parameter δ (both in Eq.(3.2)) and in Eq.(3.3)) according to the activity statistics (with M and N being the width and height of the subband image):

$$\delta_i = \alpha_i \frac{\sum\limits_{(x,y)} (A_i(x,y))}{M \times N} \tag{3.3}$$

where α_i is a constant related to spatial frequency. We have it linearly range from 0.1 at the lowest frequency to 1.0 at the highest frequency. In natural images, the subband activity measures are highly correlated at different scales, and the separate gain maps with σ set this way, tend to line up. Other parameters like γ and ε in Eq.(3.2), and γ in Eq.(3.3)) are set to be the same for all the subbands.

After the gain maps are computed they are used to modify the subbands:

$$B_i'(x,y) = G_i(x,y) \times B_i(x,y) \tag{3.4}$$

The modified subbands are then convolved with the synthesis filters and summed to reconstruct the range compression result.

3.3.4 Aggregated gain map

To some extent, the matching of local subband gains depends on accidents of image statistics: it is usually the case that high activity in one band is spatially correlated with high activity in adjacent bands. To avoid depending on this assumption, we can create a single gain map that will be used to modify all the subbands. This is straightforward to apply, since all of the subbands are represented at full resolution. To compute the gain map, we first compute an aggregated activity map by pooling activity maps over scales and orientations:

$$A_{ag}(x,y) = \sum_{i=1,...,3n+1} A_i(x,y) \tag{3.5}$$

A single gain map can then be derived from this aggregated activity map $G_{ag}(x,y) = f(A_{ag}(x,y))$, where $f()$ is of the same form as in Eq.(3.2) or (3.3)). σ is set to one tenth the average of A_{ag}.

This gain map is then used to modify all the subbands, and a scale-related constant m_i is used to control to what extent different frequencies are modified:

$$B'_i(x,y) = m_i G_{ag}(x,y) \times B_i(x,y) \tag{3.6}$$

Such a gain map G_{ag} with a Haar pyramid is shown in Figure 3-3(b), along with the corresponding aggregated activity map A_{ag} shown in Figure 3-3(a), from which G_{ag} is derived. Figure 3-3(c) shows the gray-scale range compression result after G_{ag} is applied to the subbands. Figure 3-3(d,e,f) show G_{ag}, A_{ag}, and the range compression result using QMFs. As A_{ag} is pooled from all frequencies, G_{ag} has energy in all frequencies. At first it may seem strange to modify the low frequency subbands with a gain map that contains a lot of high frequency detail, or vice versa, but due to the symmetric analysis-synthesis subband architecture, modified subbands are post-filtered by the synthesis filter bank, and therefore all modifications are confined within the subbands themselves.

3.4 Experimental results on range compression

Handling color and clipping. For color images we first convert RGB to the HSV space. Then we perform range compression on the V (value) channel, keep the hue (H) and the saturation (S) unchanged, and then convert it back to RGB to get the result. Sometimes the range compressed images look over-saturated, in which cases they can be desaturated, by reducing the saturation (S) by a factor of r_s ($S' = S/r_s$) before converting back to RGB. r_s can be set between 1.0 and 2.0.

As a final step the extreme percentiles of the intensities are clipped, and values in between are linearly scaled, so as to eliminate the sparse regions on the ends of the final histogram, and to maximize the use of the display range. This can cause some minor clipping in the very brightest and the very darkest pixels, but in practice does not cause visible problems.

Experimental Results. Figure 3.4 shows the effects of smooth gain control and different subband architectures on the "igloo" picture. We get Figure 3.4(a) using Laplacian pyramid and a point-wise sigmoid on the coefficients, Figure 3.4(b) using oversampled Haars and a point-wise sigmoid, Figure 3.4(c) using Laplacian pyramid and smooth gain control, Figure 3.4(d) using oversampled Haars, where each subband is modified by its own gain map (section 3.3.3), Figure 3.4(e) using oversampled Haars, where all the subbands are modified by one single gain map computed from an aggregated activity map (section 3.3.4). Note the halo artifacts around the pole, in (a) and (b). The worst haloes are seen with the Laplacian pyramid and a sigmoid (Figure 3.4(a)), however, the Laplacian pyramid performs fairly well when smooth gain control is used (Figure 3.4(c)). Pattanaik et al. [74] also used Laplacian pyramids with gain control, but got halo artifacts. The difference between their method and the one giving Figure 3.4(c) lies in how the gains are computed. Pattanaik et al. [74] controls the bandpass gains using the lowpass signals, whereas for Figure 3.4(c) the gain of each bandpass signal is controlled by a rectified and blurred version of the bandpass signal itself. We also compare these results with that published by Fattal et al [24], shown in (f). The colors of (f) are adjusted so that they match those of (a-e).

Shown in Figure 3.5 are range compression results on the memorial HDR image. For

Figure 3.5(a) each subband is modified by its own gain map (section 3.3.3), while for Figure 3.5(b) all the subbands are modified by the aggregated gain map (section 3.3.4). The one using a single gain map achieves a cleaner look. We compare our results with the ones published by Durand and Dorsey [21] (Figure 3.5(c)), and by Fattal et. al. [24] (Figure 3.5(d)). All of the methods give visually pleasing results, and are successful in making detail visible in both the bright and dark regions. There are some differences between the results, including overall difference in color and sharpness, but these should not be over-interpreted since they may change depending on the details of the implementation.

More results with a single gain map are shown in Figure 3.6. For all the results shown here, gamma nonlinearity (Eq.(3.2)) is used, and γ is set to 0.6. m_i in Eq.(3.6) is set to 1.0 for the three subbands at the finest scale, 0.8 for the three subbands at the second finest scale, and 0.6 for all the others including the lowpass.

3.5 Companding of HDR images

Given that we can compress the range of an HDR image into an LDR image, it is interesting to ask whether the process can be inverted. Suppose, for instance, that we have squeezed a 12-bit image into an 8-bit image. Can we retrieve a good 12-bit image? Clearly we can't do it perfectly, but perhaps we can get a good approximation. We will refer to this process as "HDR image companding". This problem appears to have received little attention.

There are various ways of representing 12 bit images, including various lossless and lossy standards. There are also some hybrid techniques that combine an 8-bit format like JPEG with auxiliary information (a second image) to increase the dynamic range [105]. However, the question we ask is this: Can we retrieve a high quality 12 bit image from an 8 bit image without sending another image in a side channel? And further, can we do this so that the 8 bit image is one that we would want to view directly on an 8 bit display?

The default method for converting a 12 bit image to 8 bits is simply to divide by 16 and quantize the 4096 levels to 256 levels. To retrieve a 12 bit image, the 256 levels are stretched back to the original 4096. It is better to do this with non-linear quantization, in which the original linear intensity values are compressed with, for example, a log or a power function, followed by quantization. The 12 bit image is retrieved by applying the inverse function. This method will lead to visible quantization steps in the 12 bit image, since there are only 8 bits worth of intensity levels.

Suppose, however, that we convert the 12 bit image to an 8 bit image through subband range compression, and then invert the process to retrieve a 12 bit image. The compression process amplifies low amplitudes and high frequencies, and the expansion process reduces them (relative to the other components). Since quantization artifacts tend to be dominated by low amplitudes and high frequencies, this means that the artifacts will have less visibility in the expanded image than they would with ordinary quantization. One application would be in driving HDR displays. Most software applications today only handle 8 bit images, and most video cards can only put out 8 bit images. It would be very useful if our laptop could output an 8 bit image and have it magically converted into a clean12 bit image by a specialized display. Of course, we cannot hope to make this conversion without any loss of information, but we can distort our image space so that the accessible set of images more closely matches the ones that we wish to display.

Another application is HDR image storage and transmission. After we turn a 12 bit image into an 8 bit one, the image can be stored in a standard lossless 8 bit format, or can be further compressed with a lossy format such as JPEG. The JPEG will not have the same quality as the original raw 12 bit image, but it will require much less storage space and will be in a standard format. A digital camera that stores HDR JPEGs rather than standard JEPGs will give its user much more flexibility when manipulating the captured image data.

Suppose we ran our range compression algorithm and generated an 8 bit image. If we knew the gain map that was used for each subband, the inversion process would be simple. Unfortunately, we don't know the gain maps, since they were not stored; all we have is the range compressed image itself. We can estimate the gain maps from this image, but these estimates will be imperfect so we will not get the original image back.

To solve this dilemma it is useful to begin at the end. Let us establish a standard method for doing range expansion; i.e., given an 8 bit image, we have an algorithm for expanding it to a 12-bit image. This can be thought of as a decoding process. Our problem now is to create an "encoded" image that will yield the desired image when it is decoded. We do not have a method for finding this image directly, but we can search for it using an iterative technique. In the next section, we describe our range-expansion method, and then an iterative range-compression method that can be coupled with it.

3.5.1 Range expansion

The range expansion follows almost exactly the same scheme as the range compression does, except that instead of multiplying the subband coefficients with their gains we divide them by their gains. The gain maps are computed in the same way as described in section 3.3. An LDR image I_l is first decomposed into subbands $B_{l,i}$, which are then rectified and blurred to give the activity maps. Gain maps $G_{l,i}$ are then computed from the activity maps using Eq.(3.2), and they are used to modify the subbands:

$$B'_{l,i}(x,y) = \frac{B_{l,i}(x,y)}{G_{l,i}(x,y)} \tag{3.7}$$

A range expanded image I_e is reconstructed from the modified subbands.

Next, given this range expansion method, we want to find an LDR image I_l that, when expanded using the above method, well approximates a target HDR image I_h. A first thought would be to get I_l directly by compressing the range of I_h, using subband decomposition and automatic gain control as described in section 3. Gain maps $G_{h,i}$ are computed from the subbands $B_{h,i}$ of I_h, and are multiplied with the subbands: $B'_{h,i}(x,y) = G_{h,i}(x,y) \times B_{h,i}(x,y)$. If the transforms are orthogonal, and somehow magically $G_{h,i}(x,y)$ is equal to $G_{l,i}(x,y)$, then by doing the expansion in Eq.(3.7) we can get I_e equal to I_h. This will not occur because $G_{h,i}$ and $G_{l,i}$ cannot be the same, since one is estimated from the subbands of I_l and the other from the subbands of I_h. But these will be close, as the subbands of I_l and those of I_h are highly correlated, which makes $G_{l,i}$ and $G_{h,i}$ highly correlated. We can look at how much I_e and I_h differ, and add a signal E_l to I_l in order to reduce the error between I_e and I_h. We do this iteratively until we find a satisfactory result.

3.5.2 Error feedback search

The search procedure is illustrated in Figure 3-7. We start the search by computing the initial estimate as the range-compressed version of the original image. This initial estimate is then quantized and passed through the RE (range expansion) box. We feed the reconstruction error back into the loop and improve our estimate. We compute the difference between the expanded image and the original image, run this error image through RC (range compression), and add this compressed error back to the previous quantized estimate. The resulting image is then quantized to get the updated estimate. This process is repeated. In our experience we reach satisfactory results after 8-12 runs. The choice of parameters (γ, and α's) (Eq. (3.2)) in our experience doesn't affect the process much, but RE and RC will have to use the same set of parameters, which means the parameters should be sent as header information with the LDR image.

We find the 8 bit image iteratively, but the procedure for expanding it to 12 bits is a one-shot multiscale procedure.

Note that the RC and RE boxes in the above iterations don't include taking the log of the image intensities. For high dynamic range images the companding is assumed to be applied in the log domain, i.e., the original image has gone through a log transformation before going into the loop.

3.5.3 Experimental results on companding

For companding color images we first convert RGB to the HSV space. The value (V) is then run through the companding loop and a compressed V is obtained when the iterations stop. This compressed V is combined with the original hue (H) and the original saturation divided by a factor of r_s ($r_s = 1.8$ for all the results shown in the paper), and converted back to RGB to get the compressed color image. This is the same as what we did for color HDR image compression. Similarly when we're going to expand a compressed color image up to 12 bits, the one-step range expansion is done on its V channel. The saturation is multiplied by the same r_s, the hue is kept the same, and they are combined with the expanded V to get the HDR color image back.

Since it is impossible to display a true HDR image in this paper, we will demonstrate an example in which the "HDR image" is 8 bits, and the "LDR image" is 3 bits. That is, we will compress an 8 bit image to a 3 bit image - dropping its bit depth by 5 bits – and then expand it back to 8 bits.

Figure 3-8 (a) shows an ordinary picture of a baby at 8 bits (256 levels). The dynamic range of the displayed image is appropriate for an 8 bit image. Figure 3-8 (b) shows the same image after it was scaled down to a smaller range and linearly quantized to 3 bits (8 levels). This image is shown with lower contrast and brightness, to suggest a low dynamic range device. (Since the image has 5 fewer bits, we might in principle show it at 1/32 the dynamic range of the original image, but here we show it at about 1/3.) Figure 3-8 (c) shows the same 3-bit image with the brightness scaled up to fill the full range of the display. The quantization artifacts are quite visible as contouring. It is possible to improve this result using non-linear quantization, but only slightly.

Figure 3-8 (d) shows an image that has been compressed and quantized to 3 bits. Figure 3-8 (e) shows this image as it would appear on our hypothetical LDR display. Figure 3-8 (f) shows the same image after expansion using our subband technique. This picture appears nearly identical to the original picture and it has no visible contouring artifacts.

This companding scheme provides us with an image that can be displayed directly on a low dynamic range device, or can be displayed after range expansion on a high dynamic range device. Figure 3-8 (g-i) shows the baby image in color, at 8 bits per color channel (i.e. a normal RGB image). Figure 3-8 (h) shows the image having been compressed to a 3-bit/channel image. Figure 3-8 (i) shows the 8 bit image that is reconstructed by the expansion technique. The expanded picture is not identical to the original, but the errors are almost invisible.

Turning now to the more pertinent problem of coding an HDR image consider the two examples in Figure 3-9. The 8 bit range compressed versions of the HDR lamp is shown in Figure 3-9 (a). Figure 3-9 (b)-(d) are a few "slices" of the reconstructed HDR images simulating increasing exposures. Figure 3-9 (e) shows a closeup of part of a monochrome intensity slice of the original HDR lamp image. Figure 3-9(f) shows the reconstruction of this slice achieved by expanding our 8 bit compressed image. It replicates the visual impression of the original. Figure 3-9(g) shows the result of compressing and expanding with 8 bits in the log intensity domain. This image shows visible contouring due to quantization. In our experience the PSNR (Peak Signal to Noise Ratio) on a typical image (measured in the log intensity domain) is 60-75 dB. From the standpoint of squared error, the proposed companding method doesn't perform as quite well as ordinary LUT companding, but it is much better visually. The artifacts do not take the form of visible contours; instead, they are small errors in local contrast within subbands, and these are not visually disturbing. Even when there is a visible difference between the original and the companded image, it is difficult to guess which is which.

A final question is whether we can get the best of both worlds, and full backward compatibility. Is the 8 bit image that is best for expansion to 12 bits also the image that looks best when displayed directly on a standard LDR display? We cannot guarantee it is, due to the emphasis the high frequencies. But in our experience the images look good visually.

3.5.4 Combining JPEG with companding

It would be useful to take one more step, and encode the 8 bit image with JPEG. JPEG compression is lossy and introduces its own artifacts. The question is how bad these artifacts will become after the expansion step. We find it is possible to get good results if the JPEG encoding is done correctly. Not surprisingly, it is necessary to code the JPEG at a fairly high bit rate, such as 1.5 to 4 bits per pixel. This still represents a substantial savings: When a 12 bit/channel image is converted to a 4 bit/pixel JPEG, the compression is from 36 bits to 4 bits, for a factor of 9.

The most troublesome artifacts, for our technique, arise when the chrominance channels (Cr, Cb) are subsampled, as is done in most off-the-shelf JPEG encoders. We used the IJG (Independent JPEG Group [44]) encoder with chrominance subsampling turned off. Figure 3-10 shows results at a bit rate of 1.7 bpp and 4.0 bpp.

3.6 Discussion

There are a number of techniques for compressing high dynamic range images in such a way that they are viewable on ordinary displays. Multiscale techniques sometimes have the reputation of being difficult to use without introducing halo artifacts. However, the implementation we describe here, based on analysis-synthesis subband architectures and smooth gain control, gives good range compression without disturbing halos. We describe some simple implementations of subband range compression, and show that the results are competitive with the leading techniques such as Durand and Dorsey [21], Fattal et al. [24], and Reinhard et. al [82].

We have not attempted to write optimized code, and cannot compare our speed with the other techniques. However, the filtering operations involved are simple to compute, and there is no need to use large or complex filters. In the future, it is likely that hardware wavelet processing will be common in image processing systems, and it will be straightforward to utilize this hardware for range compression.

This compression scheme can be inverted, so that a low dynamic range image, e.g., an 8 bit image, can be expanded into a high dynamic range image, e.g., a 12 bit image. Given an original 12 bit image, we can compute an 8 bit image that offers a good visual rendition of the HDR image, and which can be expanded to approximate the original 12 bit image with minimal degradation. This could be useful, for example, when using a standard video card to drive both LDR and HDR displays. The ability to represent 12 bit images in 8 bit file formats is also an advantage for backward compatibility in various systems, and when combined with JPEG compression can lead to further savings in storage.

3.7 Appendix

3.7.1 More result comparisons

In this section we show one more set of comparisons between our range compression algorithm and three state-of-art algorithms: Durand and Dorsey [21], Fattal *et al.* [24], and Reinhard *et al.* [82]. We constructed a high dynamic range image of a doll scene (Figure

3-11) from pictures taken at different exposure times. It is a challenging example, with the light level of the highlight on the reflective sphere exceeding 10^9 times the light level of the shadow near the little bear between the doll's lap. The high dynamic range can be appreciated in the three individual original images in Figure 3-11, where each image only captures a very limited range of light levels and leaves large parts of the scene entirely under- or over- saturated. Interestingly, when one looks at the actual scene, the gain control mechanism in our visual system is so effective that one does not notice the high dynamic range - our eyes are not blinded by the highlights, and the bear in the shadow is clearly visible. We sent the high dynamic range image to Fredo Durand, Erik Reinhard, and Ranaan Fattal, respectively, and asked them to run their algorithms on the same example. We show their results along with ours in Figure 3-12.

All the four algorithms effectively compress the dynamic range of the scene, with the bright and the dark regions displayed and important detail preserved. Our result (Figure 3-12-a) seems to be the most visually pleasing. We also seem to retain the most amount of visual detail in highlight and shadow regions, as shown by blowup comparisons in Figure 3-13. We would like to point out, though, that the appearance of a range compressed image can be affected by factors other than the main range compression algorithm, such as color, clipping of highlights and shadows in the end, *etc.*

3.7.2 Companding vs. dithering

The appearance of the high-frequency, noise-like patterns in the 3-bit companded images (Figure 3-8-d,h) may remind the reader of a halftoned or dithered image. There are actually interesting analogies to be made between companding and dithering, though there are also important differences between them.

Companding and dithering share the task of presenting higher-bit images on lower-bit medium, but unlike dithering, companding also needs to deal with a significantly compressed dynamic range. For dithering and halftoning applications, e.g. printing, the bit depth on paper is limited because the greys or color levels of ink are limited, but in terms of contrast, *i.e.*, the ratio between the brightest and the darkest, a similar level can often be achieved on paper as on computer screen. A dithered or halftoned image should look good on a display that has fewer bits but a similar dynamic range, whereas a range-compressed image should look good on a display that has a much lower dynamic range. Dithering is often performed with the explicit goal of making the low frequency component of the dithered image as close as possible to the original image. Companding, on the other hand, is done with the goal of preserving as much high frequency detail as possible of the original image. It is therefore difficult to compare companding and dithering directly. But it is interesting to compare how much information of the original image can be recovered from the quantized versions when you do companding vs. dithering.

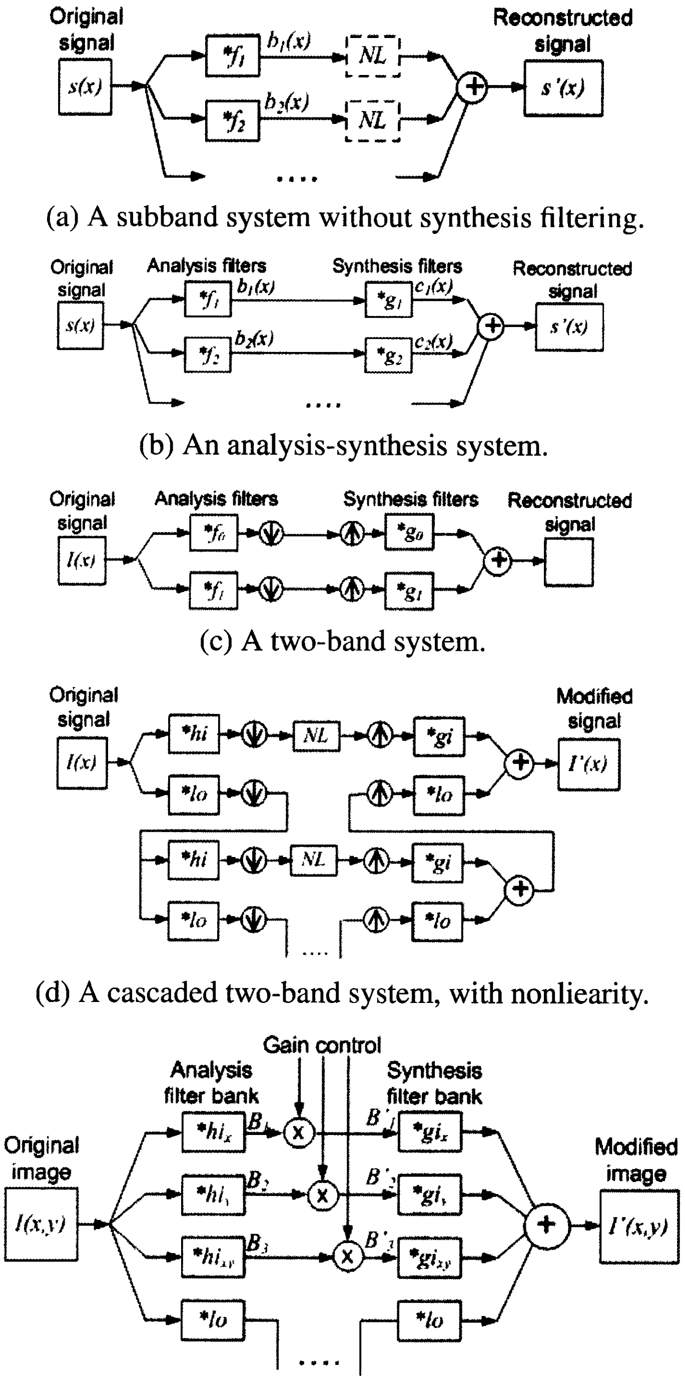
Companding and dithering both involve quantization. Simple gray level quantization often results in banding artifacts, as shown in Figure 3-9-(g), which consist of contours in smooth regions. A smooth region is originally devoid of energy in the middle and high frequencies, but the quantization contours introduce such energy, constituting distortions. At the same time, there is no other energy to mask the visibility of the distortions, which makes the artifacts jarring to the eye.

A number of dithering systems, such as Floyd-Steinberg dithering [27] and Jarvis-Judice-Ninke dithering [45], utilize an "error diffusion" process that "diffuses" the quantization error from each pixel to pixels to the right and below in a local neighborhood. As a result, the low frequency component of the dithered image closely resembles that of the original image. Because the visual system takes local averages, the dithered image looks similar to the original when viewed from a distance. The error diffusion process can be modeled as a sum of colored noise and the original image convolved with a linear filter [50, 49]. Blue noise and green noise, both with minimal energy in the low frequencies but high energy in high or middle frequencies, are often used for dithering, giving rise to a noisy look when the image is viewed from a close distance.

Our companding technique, on the other hand, first performs a nonlinear pre-emphasis of the high-frequency, low-amplitude components during the range compression stage, and then does a de-emphasis of these components in the range expansion stage. This may be counterintuitive because of the terms "compression" and "expansion". The high amplitude coefficients are actually first compressed and then expanded; however, the low amplitude coefficients are first expanded and then compressed, at least relatively, when compared to the high amplitudes. In a smooth region like the baby's face in Figure 3-8, the "compression" stage boosts the low amplitude coefficients, giving rise to the noisy look. This is analogous to dithering where noise is added. But unlike dithering, where the noise is extra, companding boosts existing middle and high frequency components that are low in amplitude. The "expansion" stage tunes down the previously boosted gains in the middle and high frequency subbands, and so the originally smooth regions become smooth again in the dynamic-range-expanded image. This stage is analogous to inverse dithering, which attempts to recover a higher-bit image from the lower-bit, dithered image.

Inverse dithering/halftoning algorithms need to reduce the added noise, and at the same time preserve image features such as edges. Low-pass filtering seems to be a natural solution for removing noise, but simple low-pass filtering results in undesirable blurring of image edges and texture. For this reason, more sophisticated techniques have been developed. Kite *et al.* [48] make use of anisotropic diffusion [75] to smooth inside smooth regions but not across edges. Neelamani *et al.* [73] base their algorithm on the formulation in [49] of error diffusion as a linear filter convolving with the original image plus noise, and performs a deconvolution with the linear filter (known by the type of error diffusion) followed by wavelet domain shrinkage.

We compare our companding with dithering + inverse-dithering techniques in Figure 3-14,3-16, 3-15, and 3-17. We present four combinations of dithering and inverse-dithering techniques: Floyd-Steinberg dithering & Neelamani *et al.* inverse dithering, Floyd-Steinberg dithering & Kite *et al.* inverse dithering, Jarvis-Judice-Nike dithering & Neelamani *et al.* inverse dithering, and Jarvis-Judice-Nike dithering & Kite *et al.* inverse dithering. Our companding achieves the highest PSNR (38.00 vs. 30.43 and below for dither/inverse-dither, for the baby example), and does the best in terms of preserving high frequency detail in the original image, as shown by the blowups in Figure 3-15 and 3-17.



(a) A subband system without synthesis filtering.

(b) An analysis-synthesis system.

(c) A two-band system.

(d) A cascaded two-band system, with nonliearity.

(e) Our architecture: symmetric, non-subsampled system, with gain control.

Figure 3-1: Subband Architectures.

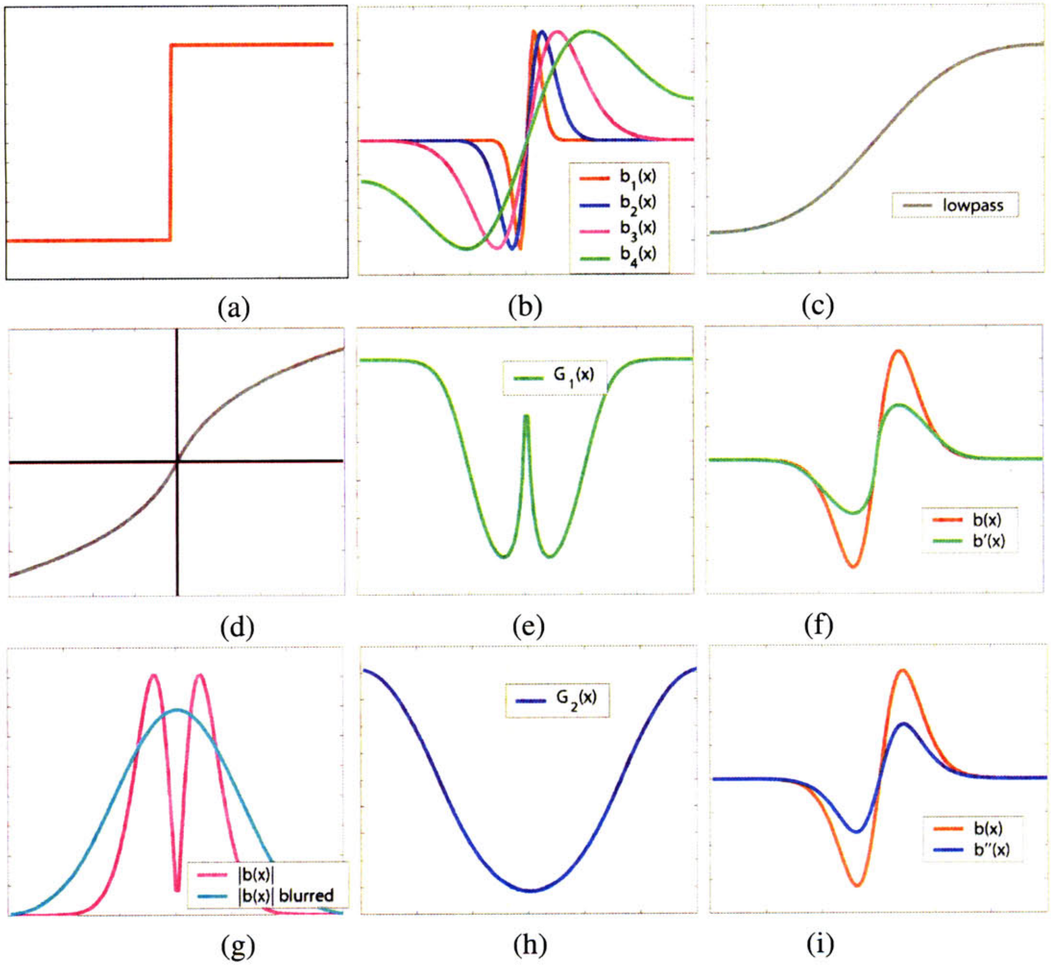
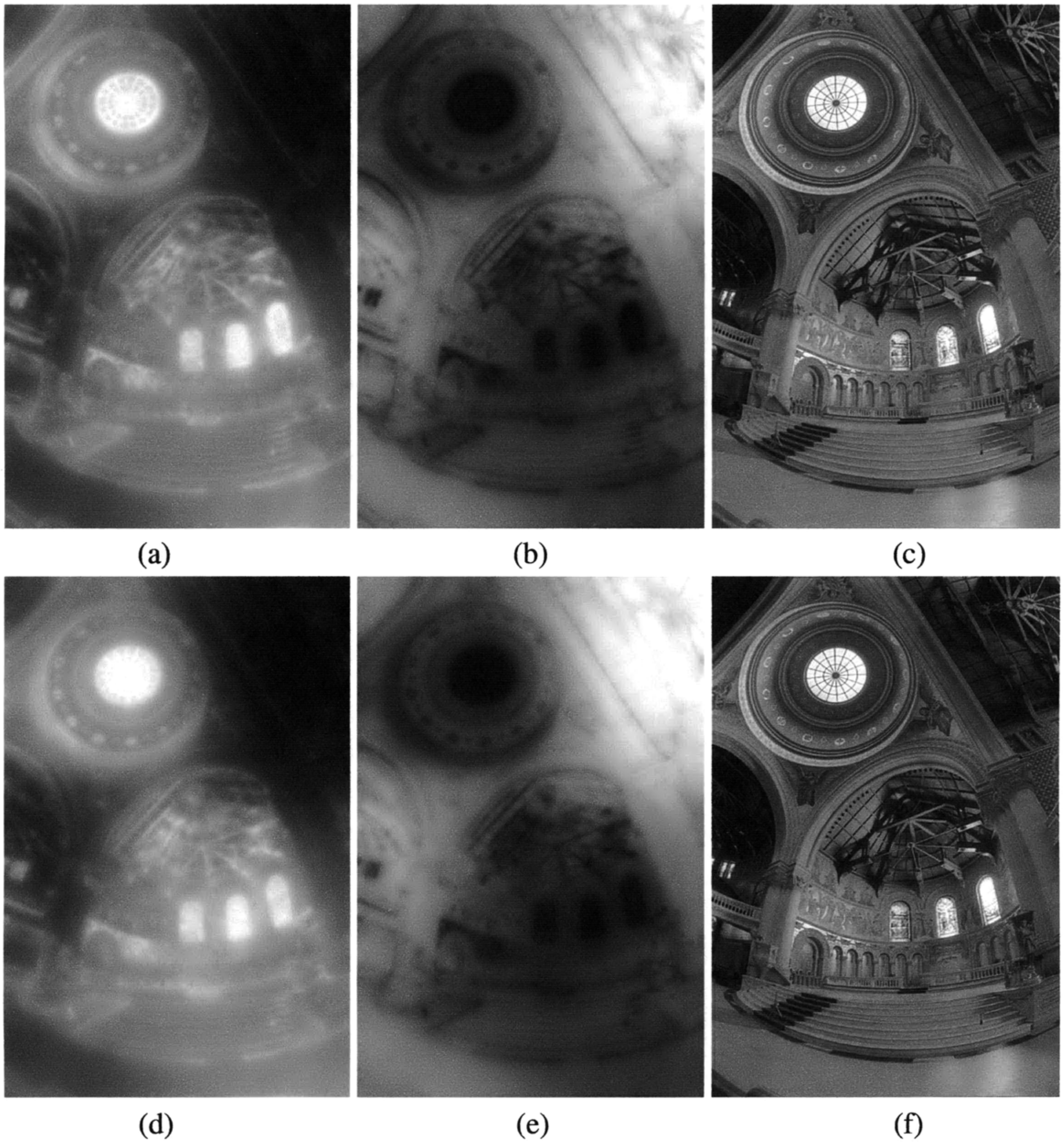


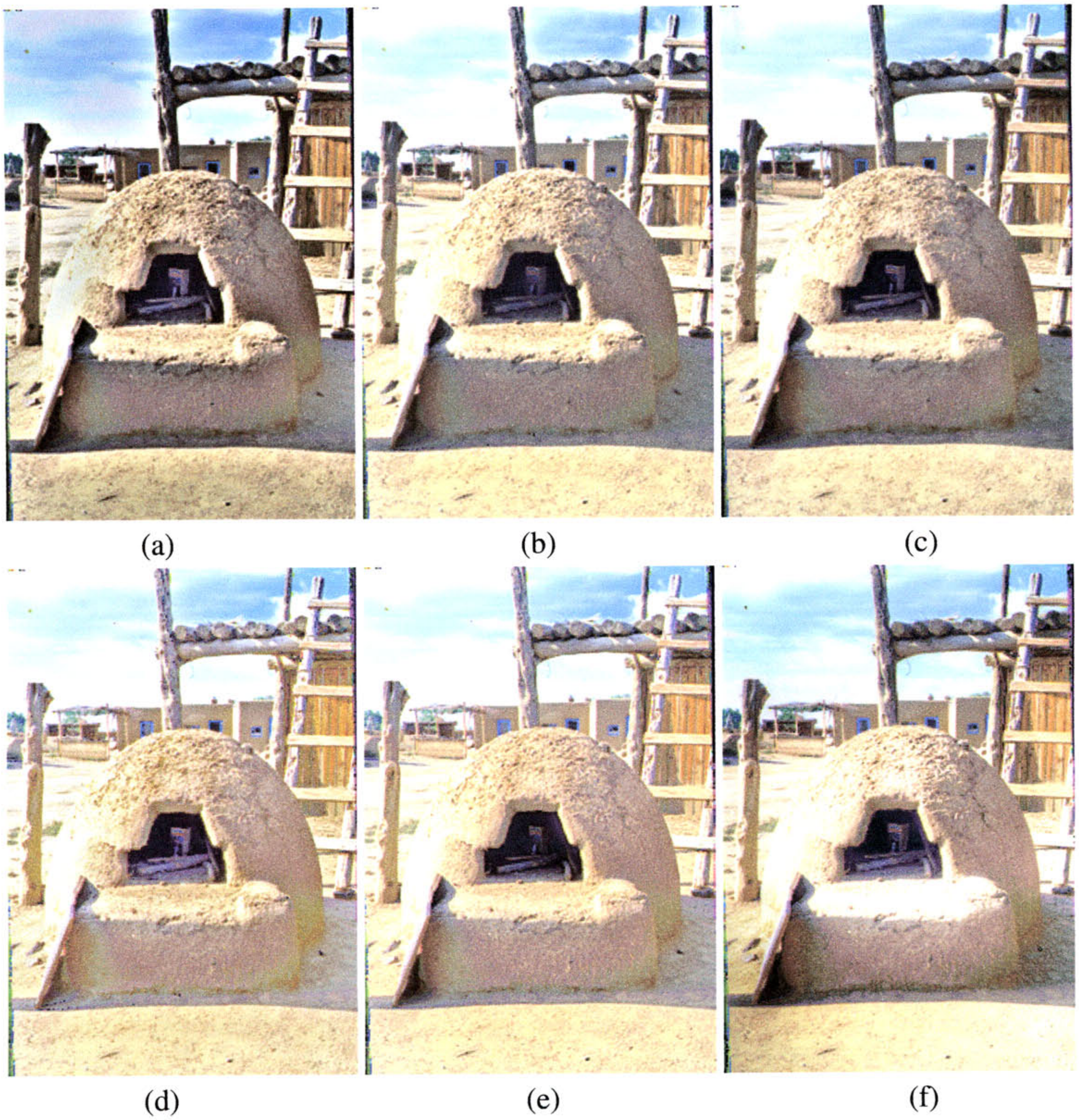
Figure 3-2: Subbands and nonlinear distotions. (a) A step edge $s(x)$. (b) Subbands of $s(x)$. (c) Lowpass residue of $s(x)$. (d) A sigmoid. (e) Effective gain $G_1(x)$ of the sigmoid. (f) Subband $b(x)$ modified by $G_1(x)$. Note the shape distortions. (g) Rectified and blurred subband to derive a smooth gain control signal $G_2(x)$. (h) $G_2(x)$. (i) Subband modified by $G_2(x)$. Distortions are reduced.



(a) (b) (c)

(d) (e) (f)

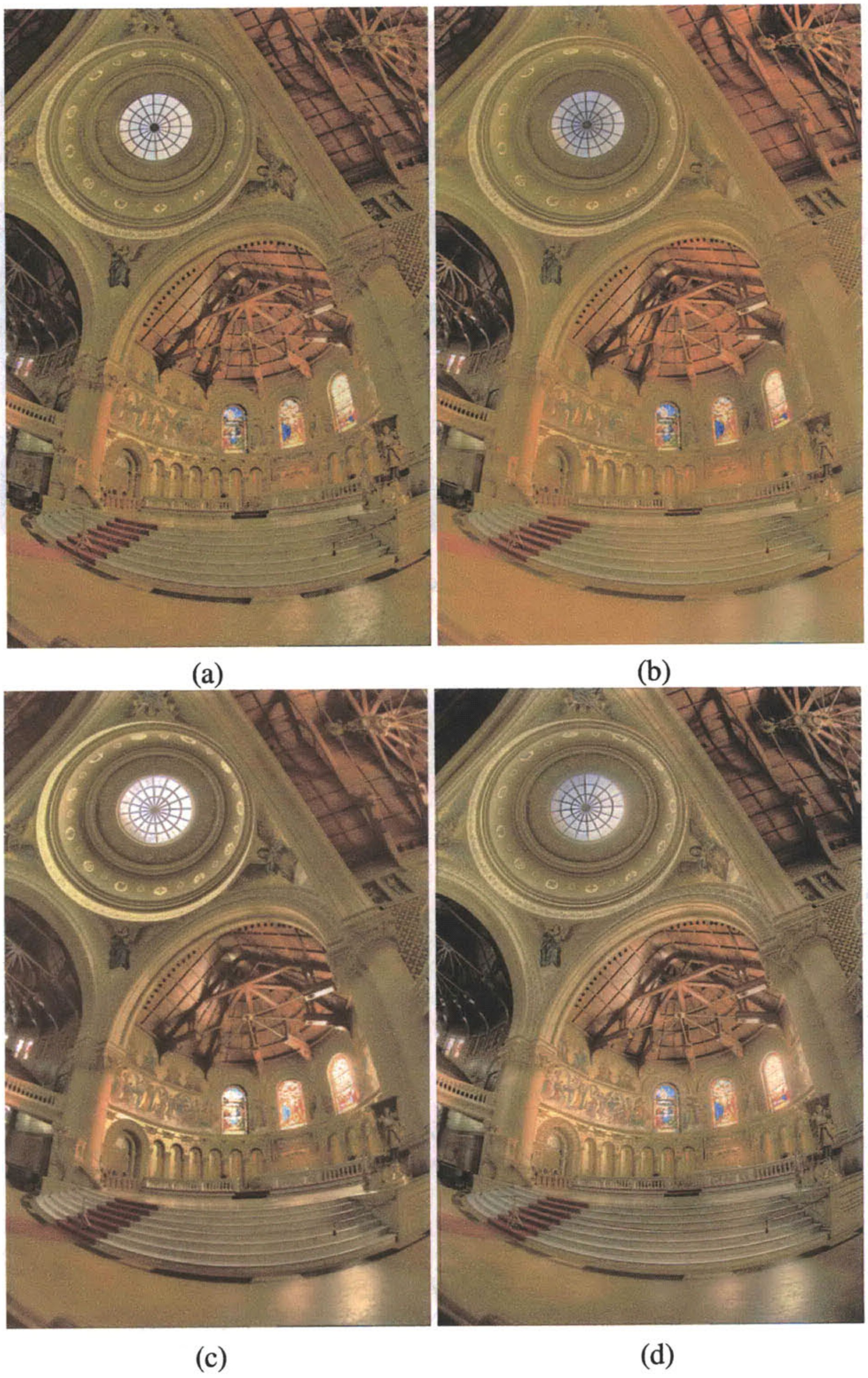
Figure 3-3: Activity and gain maps, with Haars (a-c) and QMFs (d-f), respectively. (a,d) A single activity map A_{ag} pooled from all orientations and scales. (b/e) A gain map G_{ag} computed from (a,d). (c,f) The resulting range-compressed monochrome image.



(a) (b) (c)

(d) (e) (f)

Figure 3-4: Igloo. (a)Laplacian pyramid with sigmoid. (b) Oversampled Haars with sigmoid. (c)Laplacian pyramid with smooth gain control. (d) Oversampled Haars with multiple gain maps. (e) Oversampled Haars with an aggregated gain map. (f) Result by Fattal et al [2002] (color is modified).



(a) (b) (c) (d)

Figure 3-5: Memorial Church. (a) Our result using multiple gain maps; (b) Our result using one aggregated gain map. (c) Result by Durand and Dorsey [2002]. (d) Result by Fattal et al [2002].

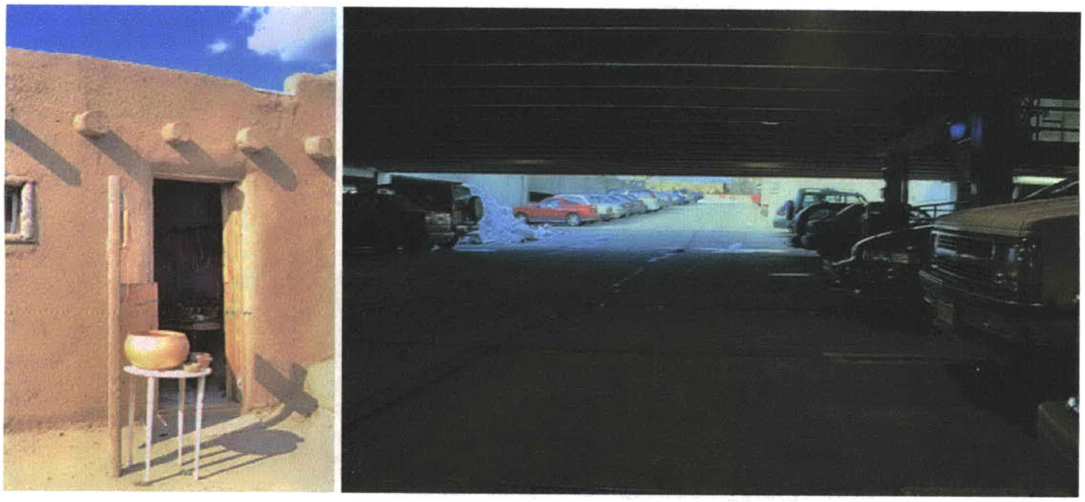


Figure 3-6: More range compression results.

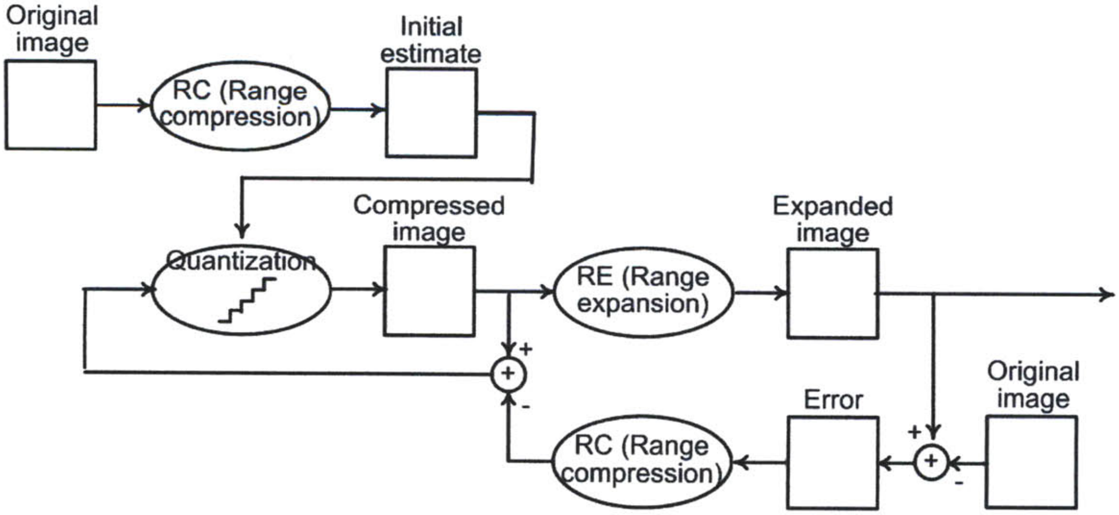


Figure 3-7: The companding flowchart.

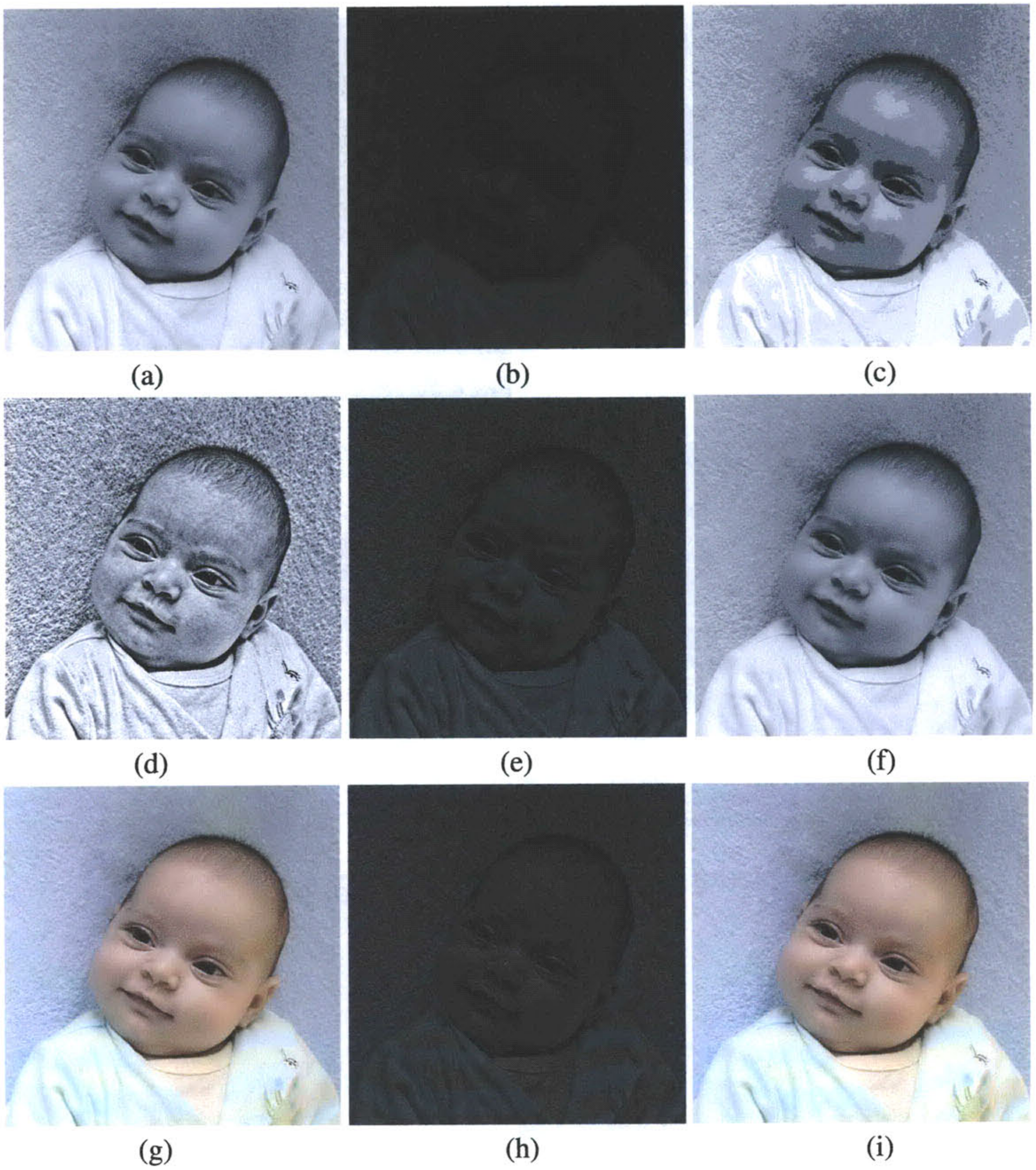
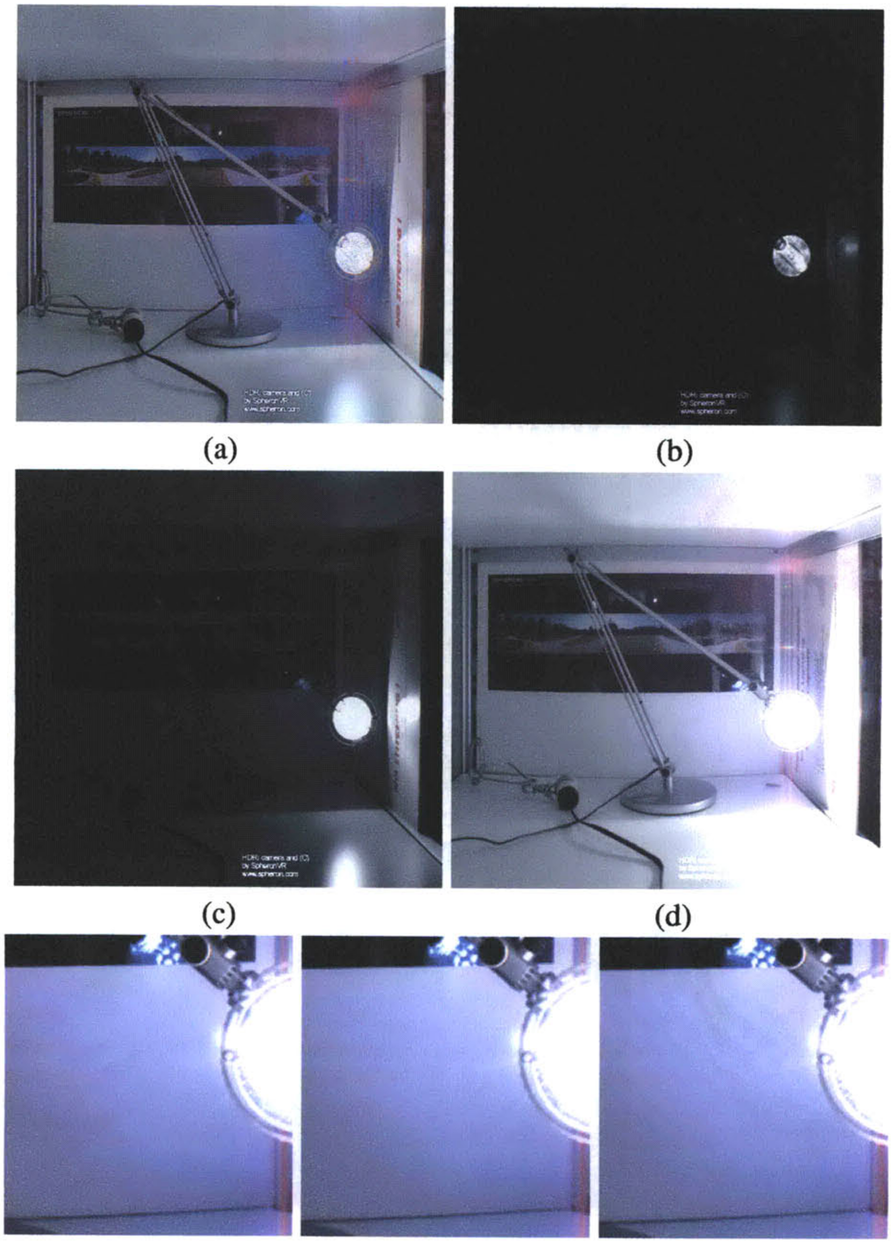


Figure 3-8: Baby companding. (a-f) in monochrome: (a) original. (b) low contrast, quantized to 3 bits. (c) 3 bit image scaled up to fill range. (d) compressed image at 3 bits. (e) compressed image at low contrast. (f) 8 bit image reconstructed from 3 bit image using the expansion technique. (g-i) in color: (g) original, 8 bits/channel. (h) compressed, 3 bits/channel, at 1/4 contrast. (i) expanded from (h).



(a) (b)

(c) (d)

(e) (f) (g)

Figure 3-9: Lamp companding. (a) the range compressed image, 8 bits/channel. (b)-(d) three exposure slices of the reconstructed HDR image from (a). (e) close-up of the original. (f) close-up of our reconstructed HDR image. (g) close-up of the image reconstructed with log quantization.

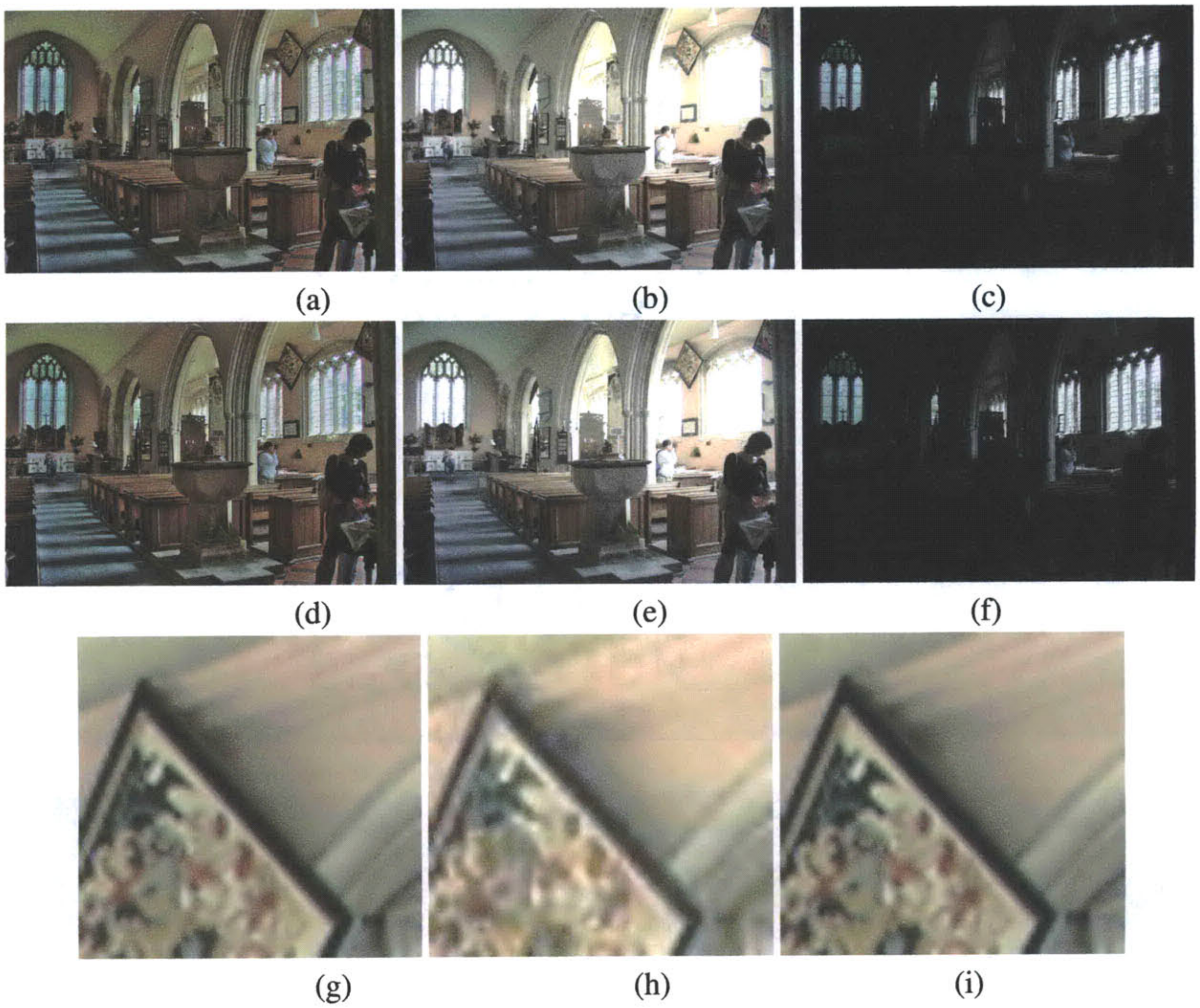


Figure 3-10: Dyrham Church companding with JPEGs. (a,d) The range compressed images, saved as (a) 4.0 bpp JPEG and (d) 1.7 bpp JPEG, respectively. (b,c) Two exposure slices of the HDR image reconstructed from the 4.0 bpp JPEG in (a). (e,f) Two exposure slices of the HDR image reconstructed from the 1.7 bpp JPEG in (e). (g) Close-up of (b). (h) Close-up of (e). (i) Close-up of the original.

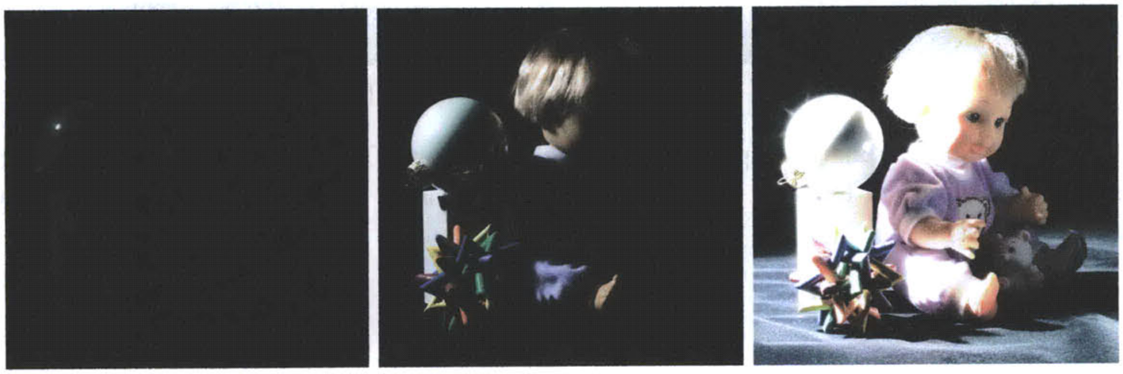
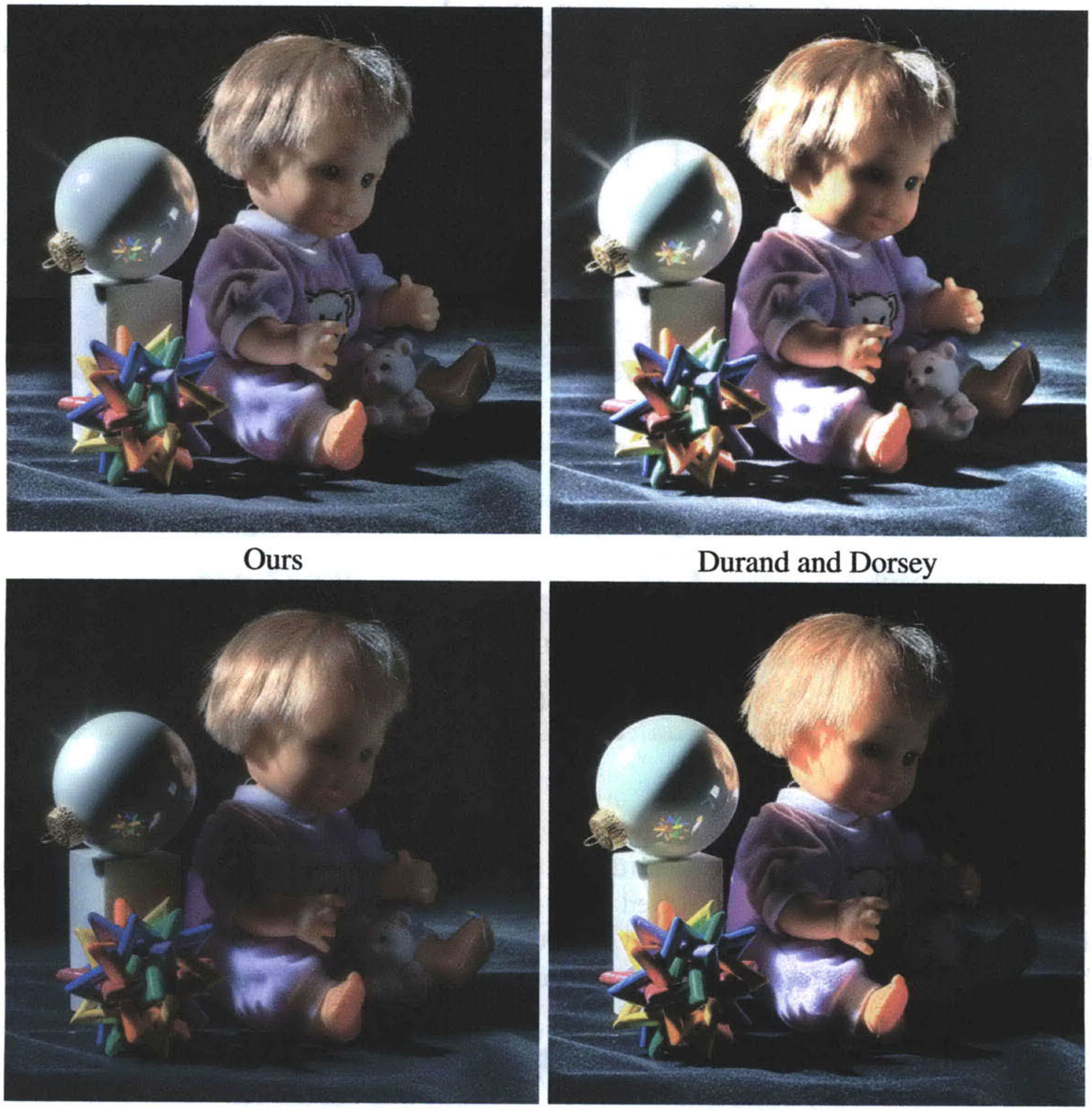


Figure 3-11: Doll - original images taken with different exposure times.



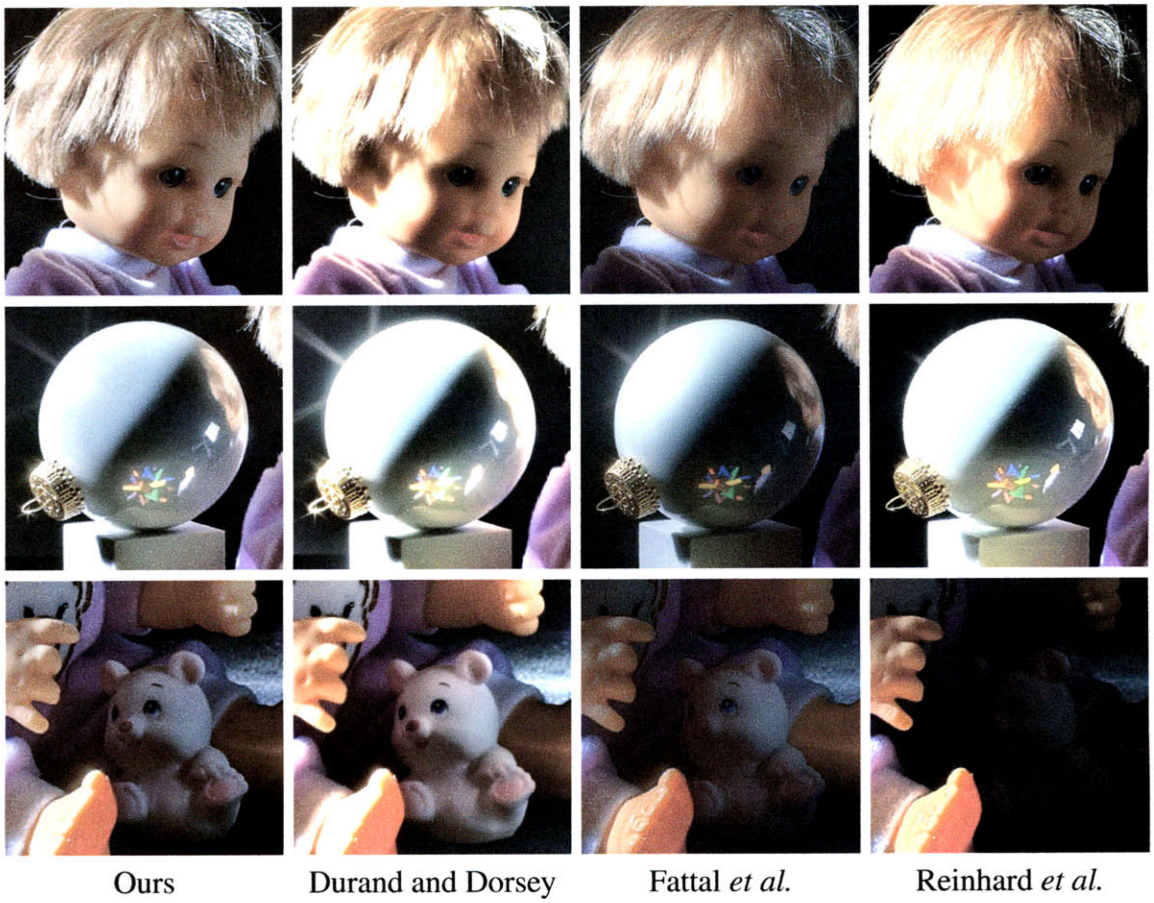
Ours

Durand and Dorsey

Fattal *et al.*

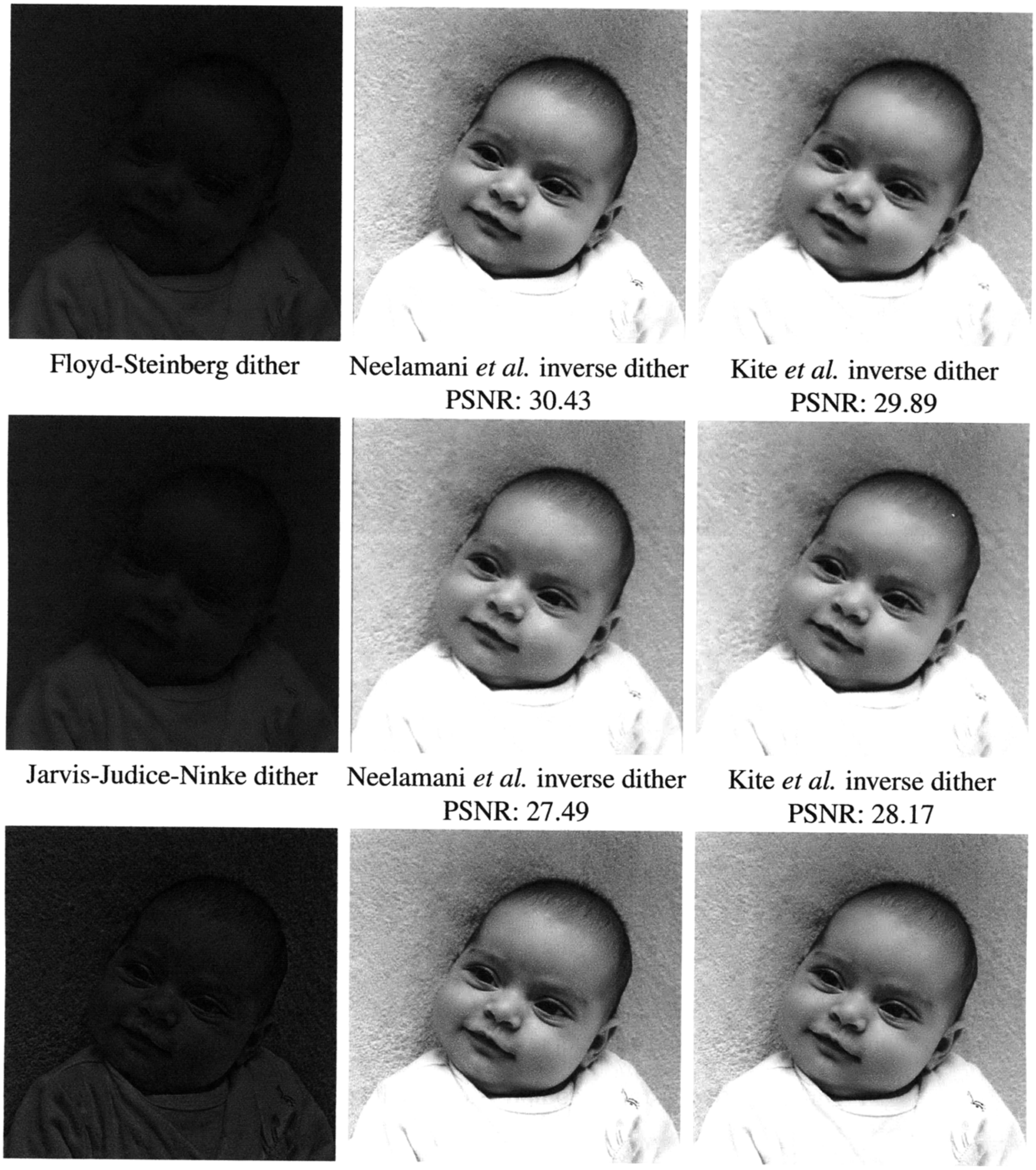
Reinhard *et al.*

Figure 3-12: Doll - range compression result comparisons. Top-left, our result. Top-right, result by Durand and Dorsey. Bottom-left, result by Fattal *et al.* Bottom-right, result by Reinhard *et al.*



Ours Durand and Dorsey Fattal *et al.* Reinhard *et al.*

Figure 3-13: Doll - range compression result comparison blowups.



Floyd-Steinberg dither

Neelamani *et al.* inverse dither
PSNR: 30.43

Kite *et al.* inverse dither
PSNR: 29.89

Jarvis-Judice-Ninke dither

Neelamani *et al.* inverse dither
PSNR: 27.49

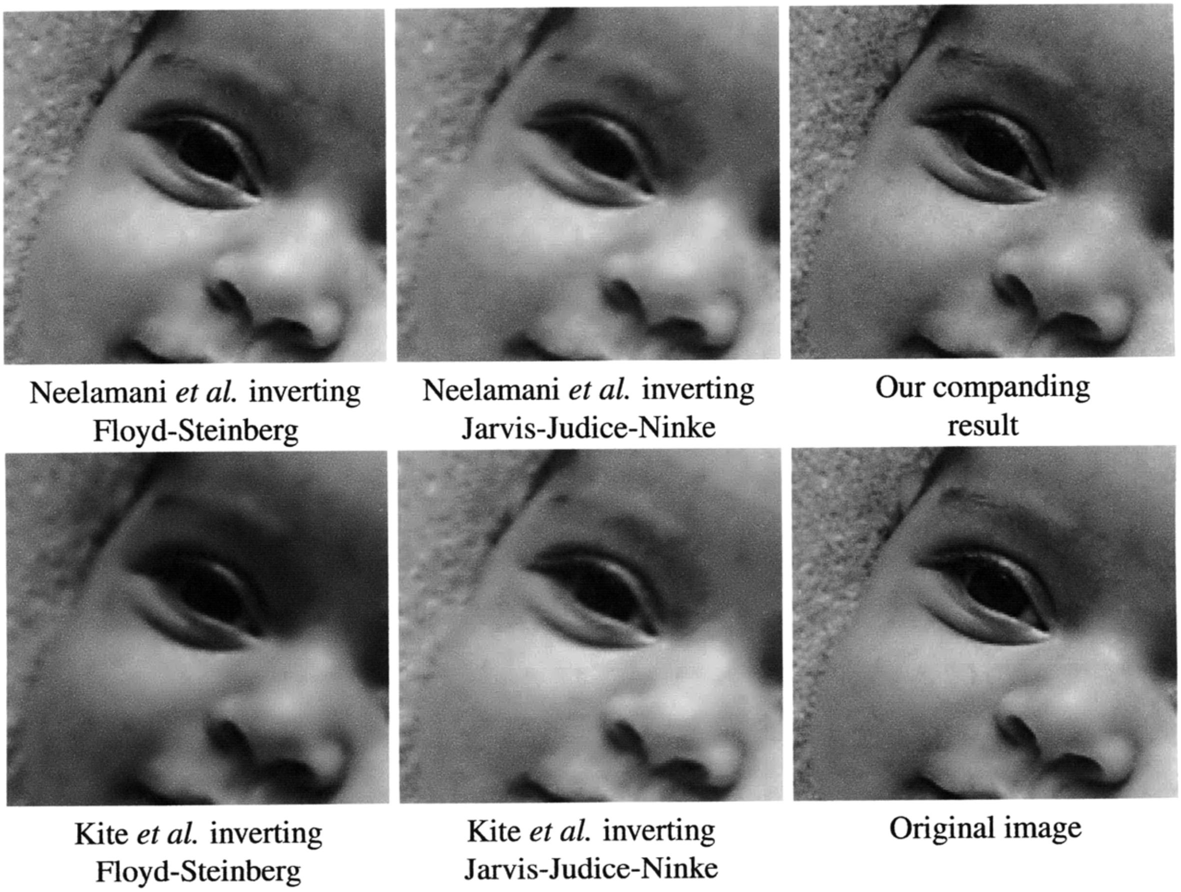
Kite *et al.* inverse dither
PSNR: 28.17

Range compression, ours

Range expansion, ours
PSNR: 38.00

Original image

Figure 3-14: Comparison between companding and dithering+inverse-dithering, on "baby" example. Shown in the left column are 3-bit images obtained through dithering or range compression plus quantization. The 3-bit images are shown at 1/3 contrast, to simulate a low dynamic range display.



Neelamani *et al.* inverting Floyd-Steinberg

Neelamani *et al.* inverting Jarvis-Judice-Ninke

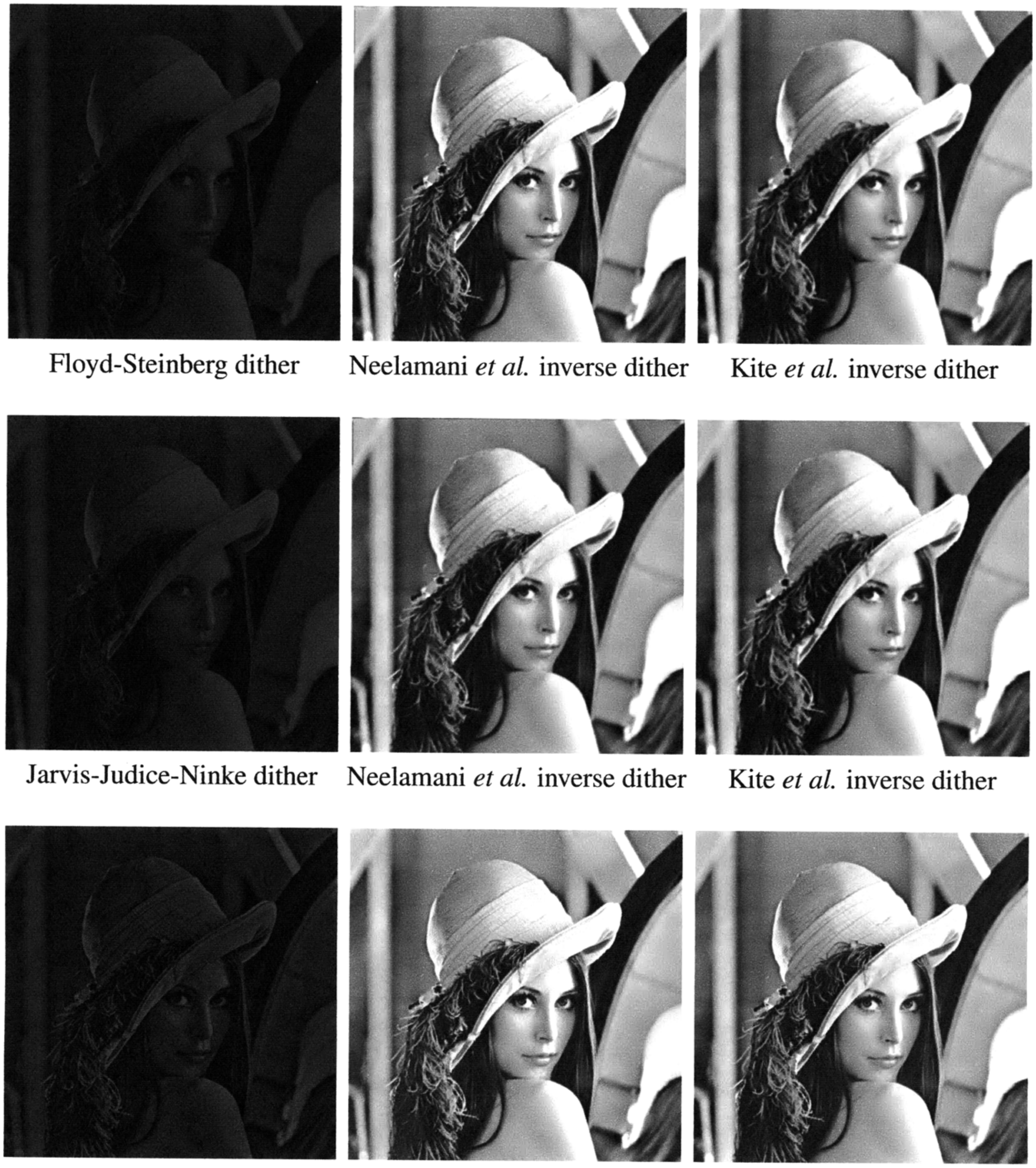
Our companding result

Kite *et al.* inverting Floyd-Steinberg

Kite *et al.* inverting Jarvis-Judice-Ninke

Original image

Figure 3-15: Blowup for comparison in Figure 3-14. Our companding does the best in terms of preserving high frequency detail in the original image (note the eyelashes and the glint in the eye).

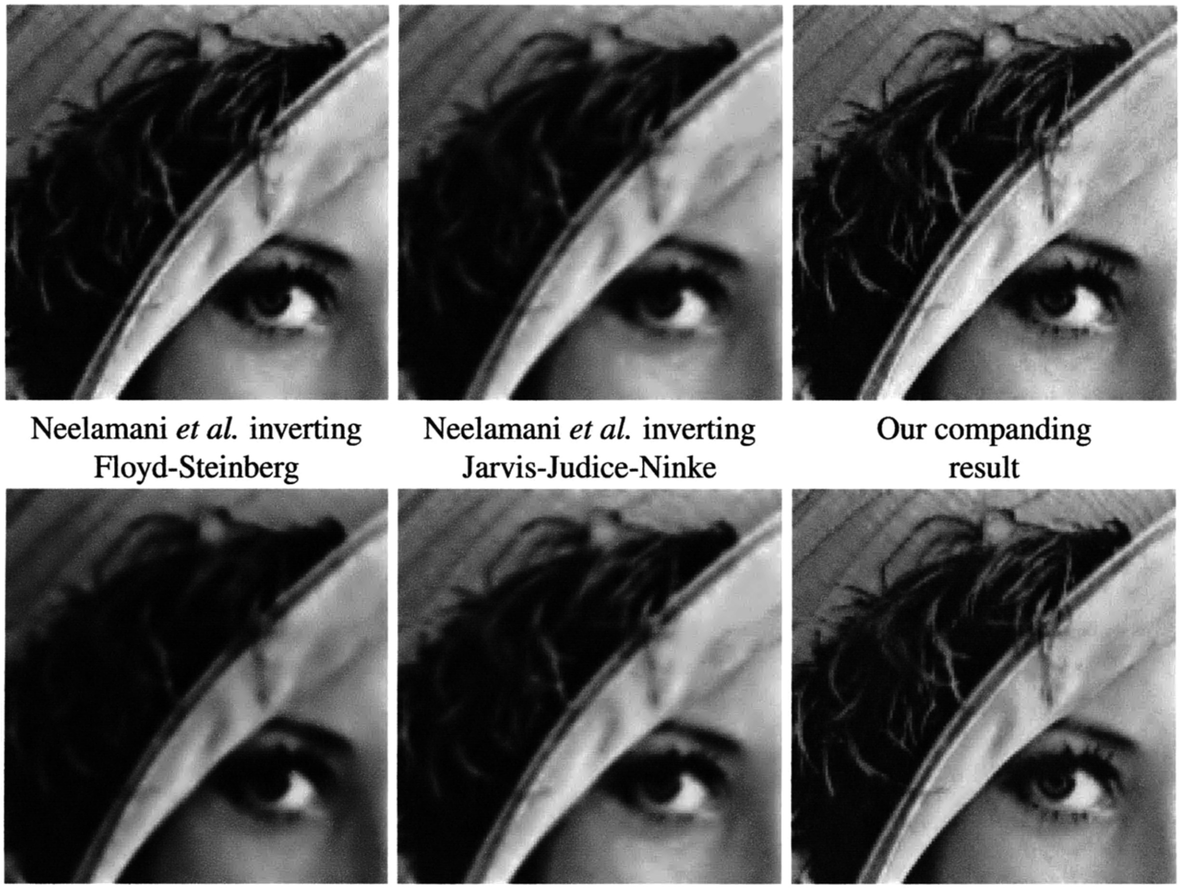


Floyd-Steinberg dither   Neelamani *et al.* inverse dither   Kite *et al.* inverse dither

Jarvis-Judice-Ninke dither   Neelamani *et al.* inverse dither   Kite *et al.* inverse dither

Range compression, ours   Range expansion, ours   Original image

Figure 3-16: Comparison between companding and dithering+inverse-dithering, on "lena" example. Shown in the left column are 3-bit images obtained through dithering or range compression plus quantization. The 3-bit images are shown at 1/3 contrast, to simulate a low dynamic range display.



Neelamani *et al.* inverting Floyd-Steinberg

Neelamani *et al.* inverting Jarvis-Judice-Ninke

Our companding result

Kite *et al.* inverting Floyd-Steinberg

Kite *et al.* inverting Jarvis-Judice-Ninke

Original image

Figure 3-17: Blowup for comparison in Figure 3-16. Our companding does the best in terms of preserving high frequency detail in the original image (note the eyelashes and the hat boa).

Chapter 4

Scribbling for localized image and video editing

One of the most common tasks in image and video editing is the local adjustment of various properties (e.g., saturation or brightness) of regions within an image or video. Edge-aware interpolation of user-drawn scribbles offers a less effort-intensive approach to this problem than traditional region selection and matting. However, the technique suffers a number of limitations, such as reduced performance in the presence of texture contrast, and the inability to handle fragmented appearances. We significantly improve the performance of edge-aware interpolation for this problem by adding a boosting-based classification step that learns to discriminate between the appearance of scribbled pixels. We show that this novel data term in combination with an existing edge-aware optimization technique achieves substantially better results for the local image and video adjustment problem than edge-aware interpolation techniques without classification, or related methods such as matting techniques or graph cut segmentation.

[3] Part of this chapter (4.1-??) has appeared as: [60] Yuanzhen Li, Edward H. Adelson, Aseem Agarwala. ScribbleBoost: Adding Classification to Edge-Aware Interpolation of Local Image and Video Adjustments. *Graphics Forum, 27(4), Proceedings of Eurographics Symposium on Rendering* 2008.

4.1 Introduction

Local manipulation of color and tone is one of the most common operations in the digital imaging workflow. For example, to improve a photograph or video sequence an artist may increase the saturation of grass regions, make the sky bluer, and brighten the people. Traditionally, localized image editing is performed by carefully isolating the desired regions using selection tools to create mattes. While effective, this approach can be much more time-consuming than is necessary for color and tone adjustments, especially for video. Matting techniques are primarily designed for the challenge of cutting an object from one image and pasting it into another, in which case it is important to solve the matting equations and recover foreground colors de-contaminated of the background. In contrast, in the case of color and tonal adjustment everything is performed in place, within the original image. Thus, local edits can be interpolated directly and more easily without the need to solve the matting equations.

Recent experiments in *edge-aware interpolation (EAI)* [56, 62, 107, 13] take this approach and offer the user a different interface to localized manipulation that does not require any explicit selection or masking from the user. Instead, a user simply draws rough scribbles on the image (e.g., one on the grass, one one the sky, and one on the people), and attaches adjustment parameters to each scribble. These adjustments parameters are then interpolated to the rest of the image or video in a fashion that respects image edges, i.e., the interpolation is smooth where the image is smooth. While EAI promises to be a powerful technique for localized image and video manipulation, there are a number of problems that currently limit its success in this context. At a high-level, EAI works by propagating the influence of each scribble along paths of pixels of similar luminance; image edges slow this propagation. One problem with this approach is that texture edges within an object also slow propagation. Texture edges may not be a problem if they are weak relative to object boundary edges, but this is often not the case. Another problem is the manipulation of fragmented appearances (such as blue sky peeking through the leaves of a tree, or a multitude of flowers) since the influence of scribbles will be stopped by the edges in-between; the user must therefore scribble each fragment. Finally, manipulating video is a challenge for EAI, since the time-axis tends to be much more aliased than the spatial axes, leading to strong temporal edges that slow propagation. Estimating video motion can sometimes address this limitation, but optical flow algorithms tend to be brittle and computationally-intensive.

In this paper, we significantly improve the performance of EAI for local image and video adjustment by taking advantage of an additional cue that is overlooked in existing EAI systems. Typically, the regions that a user wishes to adjust differently are not only separated by image edges, but they also *appear* different; that is, they have different distributions of color and texture. For this reason, many selection tools in commercial software (such as "select color range" in Adobe Photoshop) operate in color space, independent of a pixel's coordinates. Advanced users can often create a set of selections and rules in color space alone that accurately differentiate desired and un-desired regions [23]. This option can be faster to specify than a spatial selection, and perform better in the presence of fragmented appearances and video motion. In this paper, we attempt to *learn* a good color space selection by training a discriminative classifier (gentleboost [32]) to differentiate between the appearance of the pixels within different scribbles, and combine this per-pixel data term with the spatial regularization provided by the original smoothness term of EAI systems. Thus, in our interactive system, which we call ScribbleBoost, a scribble indicates that pixels *similar in appearance* to the scribbled pixels should be adjusted similarly, rather than only a continuous region containing the original scribble.

Of course, the combination of a per-pixel data term and a neighboring-pixel smoothness term is commonplace in algorithms for image segmentation [84, 34] and matting [104]. In that light, our main contribution is to extend traditional edge-aware interpolation with a novel, discriminatively-learned and weighted data term that uses a boosting-based classifier. A key feature of our data term is a weighting scheme that considers the accuracy of the classifier over its continuous output range. As a result, the weighted data term creates "crisper" transitions between regions when the classifier is confident, while the smoothness term takes over when classification is more ambiguous. Our data term also significantly improves performance in the presence of texture edges and fragmented appearances. As we show with an extensive comparison to previous work, our approach yields substantially better results with just a few user-drawn scribbles.

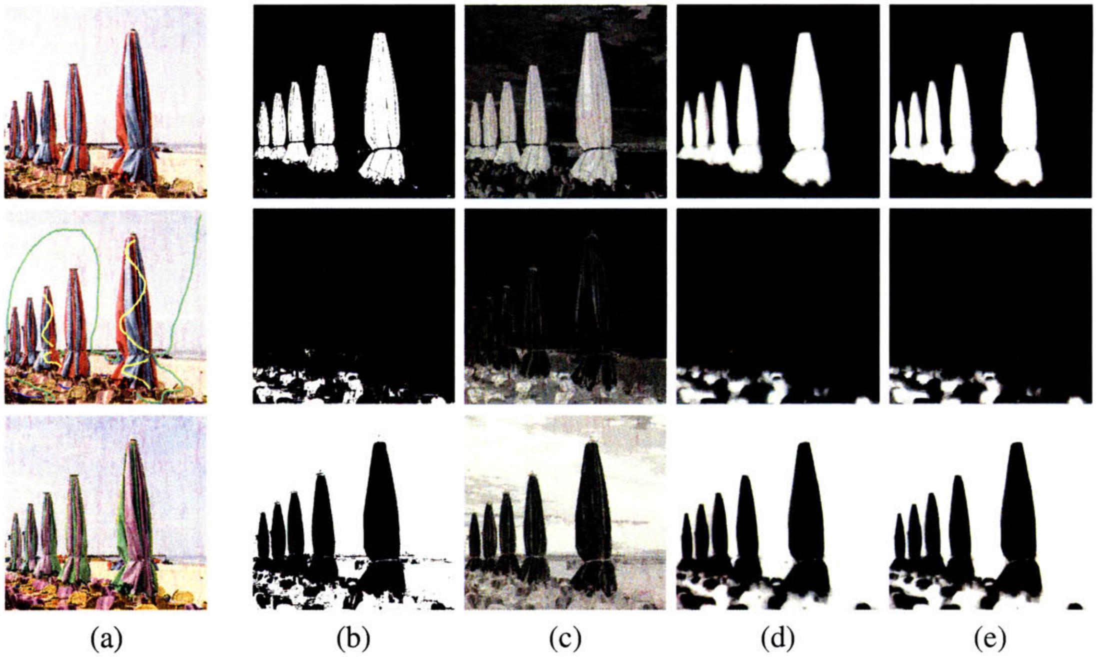


Figure 4-1: One example of localized editing using our system. Column **(a)** shows the inputs and output of our system: the input images, the user-drawn scribbles to separate the image into three classes (umbrellas, chair upholstery, and everything else), and a manipulated result that changes the hue of the umbrellas and chairs differently and saturates the rest. The other columns show intermediate outputs with one row per class. **(b)** The binary output of the three classifiers, and **(c)** the continuous output of each classifier with zero mapped to gray. **(d)** The output of edge-aware interpolation of the scribble constraints and classifier outputs, and **(e)** the final blending weights after post-processing.

4.2 Related work

Edge-aware interpolation was first introduced by Levin et al.[56] for the purpose of colorizing a grayscale image from a set of user-defined scribbles. They demonstrated that a colorized image appears natural if the color parameters specified at scribbled pixels are interpolated in a fashion that respects luminance edges. Colorization from user-drawn scribbles continues to be an active topic of research for both natural images [107, 63] and hand-drawn illustrations [80]; one significant difference from our problem is that these algorithms are not designed to take advantage of color input. Grayscale pixels are much harder to discriminate between than color pixels, and thus require the use of texture features in a neighborhood around each pixel [80, 63]. In our experience (Section 4.4.2), color at a single pixel discriminates more reliably than texture in a pixel's neighborhood.

Edge-aware interpolation was first generalized beyond colorization by Lischinski et al. [62] for the purpose of interactive tone mapping. From the perspective of a user, our system is very similar to theirs; the primary difference for the user is that, in our system, adjustments can propagate not only to pixels that are spatially close, but also to pixels that are close in appearance. Their system also included a brush that allowed the user to select and scribble any pixel similar to the color or luminance of a specified pixel (similar to "select color range" in Photoshop). This brush is, in a sense, a simple appearance-based data term that can sometimes handle fragmented appearances. However, the appearance of many objects is not confined to a narrow enough color range for this approach to be effective; in our supplemental materials, we show that such a brush is not effective for any of our examples.

Edge-aware interpolation of color and tone parameters can be seen as scattered data interpolation; given a set of constraints specified at scribbles, interpolate those parameters to the entire image or video. For image manipulation the best results are achieved if the interpolation respects image edges. To that end, a number of EAI techniques have been developed, including smooth interpolation across a bilateral grid [13], edge-weighted geodesics [107], and linear least squares optimization [56, 62]; our system utilizes the latter since the framework naturally accepts our novel data term. The bilateral grid approach is qualitatively different than the others, since strong edges do not necessarily stop propagation; we show better results on an example from their paper. The edge-weighted geodesics and least squares approaches both suffer in the presence of texture contrast and fragmented appearances. The colorization system of Luan et al. [63] also addresses these same concerns. However, since they assume grayscale input, they first create a color labeling by executing a hard graph-cut segmentation of the image based on texture segmentation cues; as a result, they are not able to achieve the long-range, soft transitions that we believe are necessary for smoothly interpolating color and tone adjustments.

Several matting and segmentation systems create masks from user-drawn scribbles [58, 35, 57, 104, 7], and these masks can certainly be used for color and tone manipulation. However, it is not clear how to blend more than two adjustment parameter constraints using mattes. Also, even for only two constraints, we find that our approach can better interpolate adjustment parameters with less user-effort than both matting and segmentation techniques, as we show in Section 4.7 with several comparisons. Algorithmically, the least squares problem solved in our system is similar to those used in both binary segmentation [35] and matting [57, 104]. However, the smoothness term used by EAI systems is typically simpler and more efficient to compute than those used in matting algorithms, which involve a larger neighborhood that better captures the precise mix of foreground and background at each pixel. Also, if a data term is used for matting, it usually involves a foreground and background model local to each pixel [104], rather than our global data term. Finally, the combination of data and smoothness terms that we describe could be solved using graph cuts, which are used by several interactive segmentation systems [84, 58]. However, the long-range, soft transitions created by edge-aware interpolation are better suited to our problem than the discrete result of graph cuts, and our results compare favorably (Section 4.7).

Finally, both Protiere and Sapiro [79] and Wang [102] have explored the use of texture cues and automatic feature selection in the context of interactive matting and segmentation. Also, a boosting classifier was used for binary image segmentation by Avidan [6], though their focus was the incorporation of spatial priors into the Adaboost algorithm.

4.3 System overview

Our approach to local color and tone manipulation is implemented as a simple interactive prototype that allows the user to draw scribbles indicating the different classes of content that the user wishes to manipulate differently, as shown in Figure 4-1(a). In this example, yellow scribbles are drawn to indicate the umbrellas, blue scribbles indicate the chair upholstery, and green scribbles indicate everything else. The user chooses to adjust the hue of the umbrellas, adjust the hue of the chairs by a different amount, and increase the saturation of everything else (the edits might be more extreme than typical, but help to demonstrate the system). The result is shown at the bottom of Figure 4-1(a).

Our algorithm could have interpolated these hue and saturation parameters directly; instead, after the user clicks a button our system computes the blending weight masks shown in the right-most column (Figure 4-1e). As shown by Lischinski et al. [62], computing a set of per-pixel blending weights that linearly blend adjustments made to the different scribble classes is equivalent to directly interpolating the adjustment parameters themselves. So, our system computes these blending weights, which sum to one (pure white) at each pixel, and loads them as layer masks into Adobe Photoshop so that the user can adjust the different layers in real-time. (Ideally, these blending weights would never be exposed to the user, and scribbles and adjustments could be performed in a single interface.) If there are only two scribble classes, blending weight compositing is identical to alpha compositing (this equivalence is also true for the blending weights of Lischinski et al. ; for more than two scribble classes, the compositing equations are different (they require "add" rather than "over" compositing [77]).

Our approach to calculating per-pixel blending weights consists of three simple steps (the intermediate results of each step are shown in Figure 4-1).

1.**Per-pixel classification.** In the first step, our system builds a boosting-based classifier to discriminate between the appearance of the different classes. In this example, the classifier attempts to learn whether a pixel more resembles the appearance of the umbrellas, the chairs, or everything else, given the training data of the scribbled pixels. The result of the classifier is a per-pixel, per-class scalar that is positive if the classifier believes the pixel belongs to the class, and negative if not (Figure 4-1(b,c)); the magnitude of the scalar represents the confidence of the classification.

2.**Edge-aware interpolation.** The second step computes an initial set of blending weights by performing edge-aware interpolation of both the scribbles and the per-pixel classification (Figure 4-1d). The interpolation is performed as a least-squares minimization of the sum of a per-pixel data term and a smoothness term per pair of neighboring pixels. Scribbled pixels are used as hard constraints, and the data term is weighted by the confidence of the classifier.

3.**Post-processing.** The third step improves the above-calculated weights in two ways. First, our system enforces a simple constraint; fractional weight values should only exist in a transition from a region of pixels of one class to a region of pixels in another class. Second, as in previous work [57, 13], we apply a sigmoid to the weight values to bias the

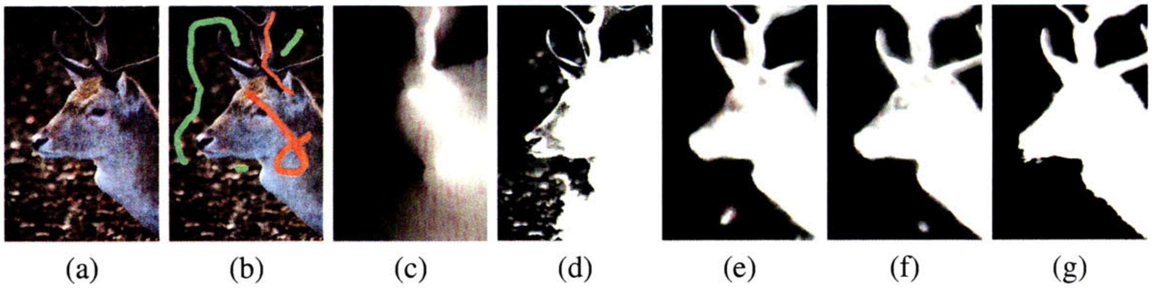


Figure 4-2: (a) A challenging image with strong texture contrast and similar color distributions between foreground and background. (b) Image with scribbles, results of (c) Lischinski *et al.* [62], (d) RobustMatting [104], (e) using GMMs in the data term, (f) our result using RGB only, and (g) Our result using additional classification features as described in Section 4.4.2.

weights towards one or zero. An example of the final blending weights can be seen in Figure 4-1e.

4.4 Per-pixel classification

A user-drawn scribble in our system not only signifies a region that should be affected by the scribble, but also an *appearance*; regions of similar appearance to the scribbled region should also be affected. This appearance prior benefits our approach in two ways. For one, the user does not need to scribble every disconnected region of a fragmented class. For example, in Figure 4-1 all the chair covers are selected even though only a few are scribbled. The second benefit of the appearance prior, as we show in Section 4.7 with comparisons to results generated without it, is that it causes our masks to be much crisper than those generated solely through spatial interpolation. When a pixel is caught between the influence of two different scribbles, the classifier can use its appearance to disambiguate its class membership, whereas spatial interpolation alone might resort to an overly soft transition.

To accomplish this appearance selection, our system supplements edge-aware interpolation with a classifier that learns how to discriminate between the appearances of the different classes. We use the gentleboost classifier [32], which is a member of the larger family of classifiers based on boosting [85, 37]. (We expect most boosting variants would

Example	GMM loss (%)	Gentleboost loss (%)
Deer (Figure 4-2)	8.14	7.70
Deer, extra features	*	1.63
Chocolates (Figure 4-3)	0.11	0.00
Birds (Figure 4-4)	4.70	3.23
Buddha (supp.)	1.80	0.56
Girl (Figure 4-5)	3.55	2.64

Table 4.1: A comparison of the classification loss on the training data as a sum of the percentage of positive pixels and the percentage of negative pixels misclassified by the GMM-based classifier and gentleboost. The classifiers use RGB values as features, except for the second row, where additional features were also used.

perform similarly; we choose gentleboost because it is simple and efficient). Boosting operates on the principle that a good classifier can be built as the weighted combination of many simple classifiers, each of which might perform just better than chance on the training data. One advantage of boosting-based classifiers is that they are discriminative rather than generative. That is, the classifier does not attempt to build a model that would generate the observed examples, but instead simply seeks to separate the data. Matting and interactive segmentation systems more commonly use the generative Gaussian Mixture Model (GMM) [14, 84, 104] to describe appearance; when color distributions are not well approximated by a small number of Gaussians, our classifier performs better. In Section 4.7 we compare our results to ones generated by replacing gentleboost with GMMs; we also compare against the results of state-of-the-art matting and interactive segmentation algorithms. In Table 4.1 we compare the classification losses of a GMM-based classifier (with five Gaussians) and gentleboost.

If there are more than two scribble classes, we are faced with a multi-class classification problem [37]. We therefore train one classifier per class in a one-versus-all framework. That is, we form the training data for the i'th class by simply aggregating the N scribbled pixels and setting label $z_j = +1$ if the j'th pixel belongs to the class, and $z_j = -1$ if not. Gentleboost then creates an ensemble classifier H_i for the i'th class as a sum of many simple weak classifiers. That is, it fits an additive model

$$H_i(v) = \sum_r h_r(v)$$

where v is the feature vector for the pixel being classified, $h_r(v)$ is a weak classifier, and r indexes over the weak classifiers. In our case, each weak classifier is modeled as a simple decision boundary in feature space. Such a decision boundary is often called a Perceptron [37], and is represented by a hyperplane θ, where $\theta_r \cdot v$ splits the feature space; if the result is positive, the weak classifier believes that v belongs to the class, and vice-versa. Each training example v_j is associated with a weight w_j and label z_j. We fit each hyperplane as perpendicular to the axis of maximum separability of the weighted training data, which is computed using weighted Fisher's Linear Discriminant (FLD) [26]. The offset of the hyperplane along this axis is computed to minimize the weighted classification loss by a simple 1D search.

4.4.1 Gentleboost

Like most boosting algorithms, gentleboost (which we describe for completeness) adjusts the weights of the training data as each weak classifier is added to the ensemble so that new weak classifiers focus on the training data that is misclassified by the current combination of simple classifiers. The weak classifiers themselves also have weights that are proportional to their performance on the training data. The algorithm begins by first initializing the training data weights $w_j = 1$ and then normalizing so that the weights of the positive examples sum to 0.5, and the weights of the negative examples sum to 0.5. Let $\delta(\cdot)$ be the indicator function that is 1 if its argument is true, and 0 otherwise. Then, for each $r = 1, 2, \ldots, M$, where M is the number of weak classifiers,

1. Fit hyperplane θ_r to weighted training data using FLD.

2. Fit weak classifier

$$h_r(v_j) = a_r \delta(\theta_r \cdot v_j > 0) + b_r \delta(\theta_r \cdot v_j \leq 0)$$

by calculating weak classifier weights a_r, b_r as

$$a_r = \frac{\sum_j w_j z_j \delta(\theta_r \cdot v_j > 0)}{\sum_j w_j \delta(\theta_r \cdot v_j > 0)} \quad b_r = \frac{\sum_j w_j z_j \delta(\theta_r \cdot v_j \leq 0)}{\sum_j w_j \delta(\theta_r \cdot v_j \leq 0)}$$

3. Update weights by $w_j = w_j e^{-z_j h_r(v_j)}$, and then re-normalize.

The final classifier $H_i(v)$ classifies a pixel by a weighted sum of the beliefs of its weak classifiers; the more the weak classifiers agree with each other, the larger the magnitude of $H_i(v)$, and the higher the confidence of the classifier in its belief. We use 100 weak classifiers. Fewer classifiers yields a less continuous confidence measure, while more requires additional computation time; we found 100 to be a good compromise. To communicate this information to the next stage of our algorithm, we evaluate each classifier on each image pixel; that is, we compute each $m_{i,p} = H_i(v_p)$, where $m_{i,p}$ is the output of the i'th classifier on pixel p. The magnitude of $m_{i,p}$ can be considered the confidence of the i'th combined classifier for pixel p.

4.4.2 Features

All the results other than Figure 4-2(g) in this paper were generated simply using the RGB color as the feature vector v at each pixel. However, one of the benefits of boosting is feature selection, i.e., it can choose the best-performing features to train the next weak classifier $h_r(v)$ given the current weighting. We therefore experimented with using a wider set of features to measure appearance and texture at a pixel, including alternative color models such as LAB and HSV, texture features such as local derivatives and Laplacians, and even the spatial coordinates of the pixel. Figure 4-2 shows an example where these extra features did indeed help (see Table 4.1 for a numerical comparison). In this example, the color distributions of foreground and background are heavily overlapping, but the shallow depth of field allows the Laplacian to be highly discriminative. Overall, though, we found that extra features hurt as often as they helped, since the extra dimensionality allowed a greater possibility of over-fitting, and texture features often fail near object boundaries.

4.5 Edge-aware interpolation

The previous step of our approach calculates a measure of the belief that each pixel belongs to each stroke class, expressed as $m_{i,p}$. In this step, our system calculates per-class, per-pixel blending weights by performing spatial regularization, so that neighboring pixels of

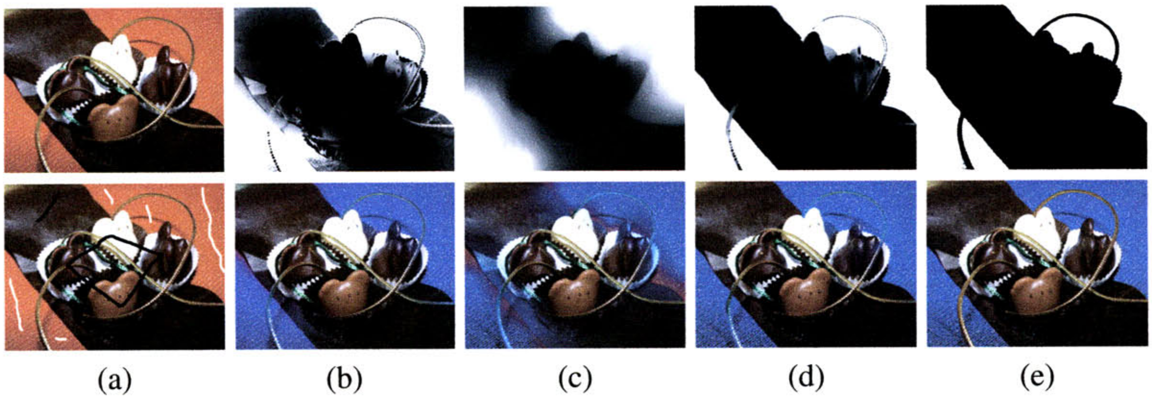


Figure 4-3: (a) An example from the bilateral grid paper [13] (top) and a similar set of scribbles separating the image into two classes (bottom). The mask and local image adjustment result of (b) bilateral grid, (c) Lischinski *et al.*, (d) RobustMatting, and (e) ScribbleBoost.

similar appearance are manipulated similarly.

Though the output of this step is a set of blending weights, the equations are easier to understand if we first present them as directly interpolating an adjustment parameter. That is, we assume that the user has already defined the desired values of some adjustment parameter (e.g., saturation or brightness) for each stroke class; we represent this scalar value as c_i for the i'th class. We then compute the value of this adjustment parameter f_p for each pixel p. To do so, we compute f_p that minimizes the sum of a per-pixel data term and smoothness term per pair of neighboring pixels,

$$\sum_{p \notin \Omega} D_p + \lambda \sum_{p,q} S_{p,q} \tag{4.1}$$

subject to the constraint that $f_p = c_i$ for all pixels $p \in \Omega_i$, where Ω is the set of stroked pixels, D_p is the data term on pixel p, $S_{p,q}$ is the smoothness term on neighboring pixels p and q, and λ weights the smoothness term relative to the data term (we use $\lambda = 1$ in all examples). Our formulation is similar to that of Lischinski *et al.* [62], except for the data term. Our smoothness term is nearly identical, i.e.,

$$S_{p,q} = \frac{(f_p - f_q)^2}{\nabla I_{p,q} + \varepsilon}$$

where $\nabla I_{p,q}$ is the magnitude of the color gradient between pixels p and q, and $\varepsilon = .001$

prevents division by zero. The smoothness term encourages neighboring pixels to have similar values of parameter f, but the strength of the term is weakened across image edges.

The data term, which is novel to our formulation, is designed to encourage the value f_p to be c_i if we believe pixel p belongs to the i'th class.

$$D_p = \sum_i w_{i,p}(f_p - c_i)^2$$

The most interesting aspect of this data term is the design of the weight $w_{i,p}$, which depends on the classifier output $m_{i,p}$ computed in Section 4.4.1. Obviously, if $m_{i,p}$ is less than or equal to zero, the classifier does not believe pixel p belongs to the i'th class, so $w_{i,p}$ should be zero. Otherwise, the weight should be proportional to the confidence of the classification. This confidence can be measured in two ways. The first is simply the absolute value of $m_{i,p}$, which measures the confidence of the i'th classifier specifically for pixel p. However, there is an additional and valuable cue for measuring confidence: the *overall* accuracy of the i'th classifier on its training data. In cases where the color distributions of the different classes are well-separated, the classifier may achieve no or almost no loss, in which case the data term weight should be higher. Otherwise, the color distributions may overlap significantly, and thus the classifier may perform poorly and offer almost no discriminative insight – in this case, the weight should be close to zero, and the overall optimization should revert to the original formulation of Lischinski et al. [62] that does not use classification.

One simple measure of overall accuracy is the classifier loss; however, this measure does not express how the classifier's performance varies over the range of classifier outputs. That is, the classifier might be quite inaccurate for low values of $m_{i,p}$, but very accurate for higher values. So, we instead ask a simple question: above what value of m_i does the i'th classifier perform perfectly on the training data? That is, what is the maximum value of $m_{i,p}$ for the negative training examples? For classifier outputs above this threshold (which we call m_i^*), we can be more confident of the classifier. For outputs below this threshold, we know that that classifier sometimes misclassifies, so confidence should be very low. We thus define a weighting function that decreases very rapidly below the threshold m_i^*, and increases less rapidly above it (as overly-strong weights can render the linear system that computes the minimum ill-conditioned).

$$w_{i,p} = \begin{cases} 0 & m_{i,p} \leq 0 \\ \left(\frac{m_{i,p}}{m_i^*}\right)^4 & 0 < m_{i,p} \leq m_i^* \\ \left(\frac{m_{i,p}}{m_i^*}\right)^2 & m_i^* < m_{i,p} \end{cases}$$

We add one additional caveat to the computation of m_i^*. If the classifier performs very well, m_i^* may be zero or even negative. Even positive values of m_i^* that are very small can be problematic, as the data term becomes too strong and the resultant masks almost binary. We thus do not allow m_i^* to be any smaller than $\frac{1}{10}$ of the overall range of positive classifier outputs.

The result of this weighting scheme is that the blending weights are softer in areas where the classifier has low confidence, and vice-versa. This effect can be seen by comparing Figures 4-2(f) and (g); the latter uses a better-performing classifier than the former, and so its transitions are much crisper. In effect, our scheme can minimize the negative effects of uncertainty by resorting to soft transitions that do not introduce new edges that attract the eye.

The minimization problem in equation (4.1) is quadratic, and its global minimum can be found by computing a linear system $Af = b$ with respect to the per-pixel adjustment parameter f. How can we, instead, compute a set of blending weights so that the linear system does not need to be re-solved each time the parameters are changed? We again take inspiration from the approach of Lischinski et al. , and separate the linear system into a set of per-class linear systems. To do so, we assume the adjustment parameters are linear. Then, with a variable substitution $f_i' = \frac{1}{c_i} f_i$ and $b_i' = \frac{1}{c_i} b_i$, the linear system $Af = b$ can be expressed as $\sum_i A_i \sum_i f_i' = \sum_i b_i'$. We can compute each f_i' as $(\sum_i A_i) f_i' = b_i'$ and the final parameter vector f can be expressed as $f = \sum_i c_i f_i'$. We can therefore use each f_i' as a blending weight mask, and simply use "add" compositing to compute a final image. Note that this approach treats adjustment parameters as linear even though certain adjustments, such as hue, are not; none the less, treating these parameters as linear typically generates

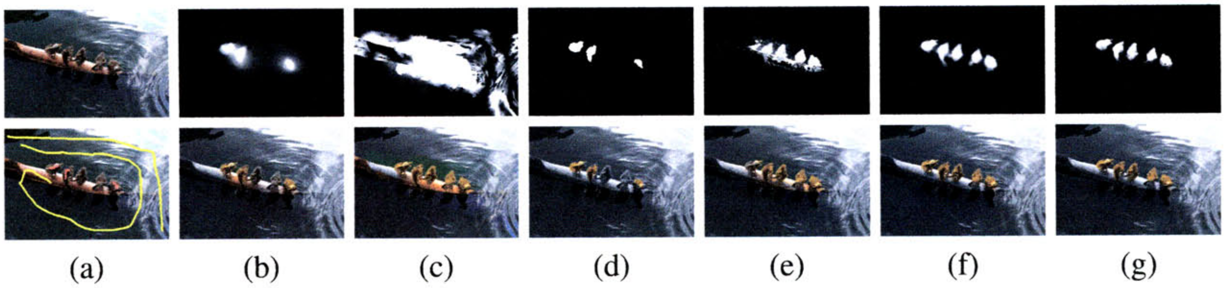


Figure 4-4: (a) An example image (top) and a set of scribbles isolating the birds from the background (bottom). The mask and local image adjustment result of (b) Lischinski *et al.*, (c) Robust-Matting, (d) Lazy Snapping, (e) Bai and Sapiro, (f) our technique using GMMs in the data term, and (g) ScribbleBoost.

results that match our mental model of what we would expect to see.

There are a number of approaches to efficiently solving large, sparse linear systems of this form, including multigrid algorithms on the GPU [10]. For ease of implementation we use locally-adapted hierarchical basis preconditioning [93]. Notice that each linear system (one per f_i') can be computed in parallel.

4.6 Post-processing

In the third and final step of blending weights calculation, the masks are improved in two ways. The first step is motivated by one of the artifacts that can be seen in the results of the previous step in Figure 4-1(d); there are occasional patches of soft, fractional values that are disconnected from any fully opaque pixels that definitely belong to the class represented by the mask. For example, in the second row of Figure 4-1(d) there are soft patches of pixels well above the covered chairs that this class represents. We make the observation that fractional values should only exist at the transition from one class to another,[1] and modify the masks to enforce this constraint. First, we assume that blending weights more than 95% opaque are definitely in the corresponding class, weights less than 5% opaque are definitely not in the class, and weights values in-between are transitional. Then, we compute a flood fill from in-class pixels to transitional pixels to identify those transitional pixels that are, in fact, connected. Any transitional pixel that are not connected to in-class

[1] This observation is not always true; counter-examples include partially transparent regions such as smoke, or structures thinner than one pixel for their entire extent. We ignore these cases.

pixels are set to zero. This operation is performed for each class, and then the weights are re-normalized to sum to unity. As can be see in Figure 4-1(e), this operation removes these errant regions.

The final post-processing step (which is also performed in other EAI systems [57, 13]) simply biases the masks slightly towards zero and one; we scale the weights from the center of their range by a factor of 1.1, and clamp and re-normalize so that the weights sum to unity. The output is the set of final blending weights.

4.7 Results

In Figures 4-2-4-7 we show a number of results created using our system as well as comparisons to results created using previous work with the same set of scribbles (we recommend zooming in on the image in the electronic version of this paper to better see the differences). In most examples we use only two scribble classes so that comparisons can be made to the output of matting and segmentation algorithms. In these cases we show blending weights from our technique and that of Lischinski *et al.*, as well as mattes from matting and segmentation algorithms. Comparing blending weights and mattes directly can be misleading, as mattes are computed to model the matting equations and produce precise foreground colors de-contaminated from the background, while blending weights are designed for in-place editing. If our blending weights were used for compositing onto novel backgrounds, the result would likely not be successful. However, these masks can be useful to bring attention to problematic areas in the final edited results, which were created in Adobe Photoshop by adjusting hue, saturation, contrast, and/or brightness of the differently masked layers. We often chose more drastic edits than might be typical since they better reveal the differences in the outputs of different systems. Finally, we show examples of multi-class edits for which matting algorithms cannot directly be compared in Figure 4-6, and a video result in Figure 4-7. Finally, several additional results and comparisons are shown in the supplemental materials.

Our comparison results of Lischinski *et al.* [62] were created using our system with the classification-based data term disabled; without this term, our systems are largely identical.

In fairness, it should be noted that we are applying their method to a different problem than the one they were trying to solve; very soft masks that work well for HDR tone mapping might not work for color adjustments. For our application, we can see that their technique does not handle fragmented appearances where each fragment is not scribbled (e.g., several of the birds in Figure 4-4), and suffers in the presence of texture edges (e.g., the textured dress in Figure 4-5). An extreme example of a fragmented appearance can be seen in the lilypads in Figure 4-6; stroking each lilypad would be very time-consuming. The matting results of RobustMatting [104] and Bai and Sapiro [7] were created using the authors' systems (we manually drew similar strokes in their interfaces). Matting algorithms are challenged by the rather sparse set of scribbles used in these examples. Figure 4-3 shows a comparison using an image and result from the bilateral grid paper [13]. Their result show significantly more color spilling than ours, which benefits from the rather easy separability of the colors in the separate classes of this example (Table 4.1). In Figures 4-4 and 4-5 we compare against the results of a publicly available implementation[2] of Lazy Snapping [58], which uses graph cuts to create binary masks.

We also show comparisons to results created using a GMM classifier instead of gentleboost. These results are still quite good, in part due to the other components of our technique which remain the same, such as edge-aware interpolation, the weighting scheme described in Section 4.5, and the post-processing stage. The classification loss comparison between GMMs and boosting in Table 4.1 varies from little difference up to a factor of 3.2 (gentleboost always performs better). In our experience, boosting also exhibits better accuracy in classification *confidence*; the effect of this difference can be seen in the generally softer masks from GMMs. Differences can also be seen in the editing results, most notably near the edges of the dress in Figure 4-5.

Most of our examples show blending weight masks that resemble alpha mattes and crisply separate different objects; however, users do not always apply different adjustments strictly to different objects. In Figure 4-8 we show an example from the paper of Lischinski *et al.* that has different scribbles on the same object (a tablecloth) to interpolate a depth-of-field effect. This example requires a longer-range, smooth transition, which our system

[2] `http://www.cs.cmu.edu/~mohitg/segmentation.htm`

can still produce.

Finally, our technique works well for video sequences, and we show an example in Figure 4-7 where scribbles are drawn on only 1 out of 123 frames (several more examples are shown in the accompanying video; for each, scribbles were drawn on just one frame). The spatial regularization and post-processing steps are performed independently for each frame. While we have not noticed any temporal coherence artifacts, more challenging video sequences might benefit from adding temporal smoothness terms to our EAI formulation. Our reliance on per-pixel classification benefits our video results, whereas pure EAI systems must depend solely on propagating information across time.

Failure cases. Our system does not always yield the desired result. One source of failure is when the color distributions of the layers that the user wishes to separate are very similar; an example can be seen in Figure 4-2. In this case, extra features can help. Also, we assume that the user wishes to manipulate pixels of similar appearance in the same fashion, which isn't always true. For example, if the user wished to edit only one umbrella in Figure 4-1, our system would hinder the user more than help. Perhaps the ideal system would involve two types of strokes; ours, and the scribbles of traditional EAI which only indicate a region and not an appearance.

Performance. Our system involves substantial computation. The bottlenecks, in decreasing order, are the solution of the sparse linear systems, the evaluation of the classifier (which involves 100 weak classifiers) on each pixel, and the training of the classifiers. However, our algorithms can easily benefit from recent GPU and multi-core processing models. For example, Szeliski [93] points out that his solver easily maps to the GPU (our implementation is software-only), and the classification of each pixel can be performed in parallel. Each classifier can also be trained in parallel. While we have not experimented with GPU execution, we did achieve some parallelization with just a few lines of OpenMP (www.openmp.org) code. As a result, the one megapixel, three scribble-class example in Figure 4-1 took about 10 seconds on a multi-core machine to compute all weights. The 0.7 megapixel, two scribble-class example in Figure 4-5 took only 3 seconds. These execution times lead us to believe that a GPU-based implementation could respond in real-time to a new scribble at a preview resolution.

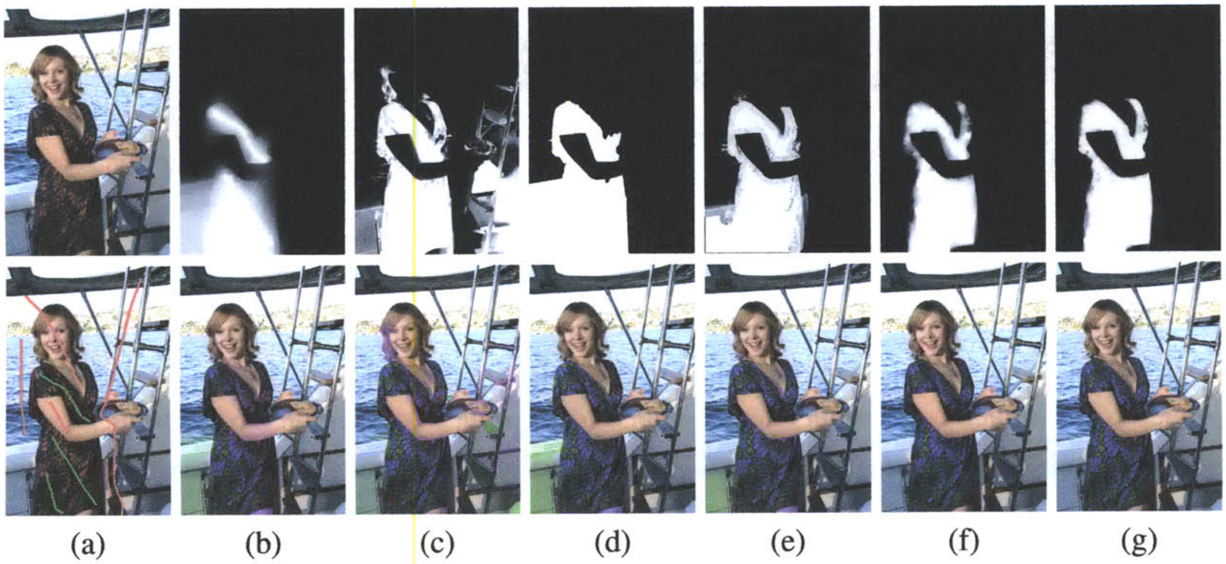


Figure 4-5: (a) Example image (top) and scribbles isolating the dress (bottom). The mask and local image adjustment result of (b) Lischinski et al., (c) RobustMatting, (d) Lazy Snapping, (e) Bai and Sapiro, (f) our technique using GMMs in the data term, and (g) ScribbleBoost.

Sensitivity to user scribbles. Our comparisons show that our technique achieves significantly better performance than previous work. However, did we simply choose scribbles that favor our technique? To address this question, we performed an experiment comparing the robustness of various methods to a variety of scribble styles. We asked five users to draw scribbles that separate a target object from the rest of the image; the resultant scribbles varied widely in terms of positioning and density, as shown in the supplemental materials. In spite of this variation, our method consistently performs better than the compared techniques.

4.8 Conclusion

Local color and tone manipulation is a very frequent task for image and video editors, and we believe that our technique has the potential to significantly reduce their burden. There are many ways that our approach could be further improved. One direction is improvements to our classifier, which is currently very simple. Classification is an actively researched topic and recent advances could be applied to our problem. For example, our classification problem is semi-supervised, since the unlabeled pixel data is also available; applying semi-

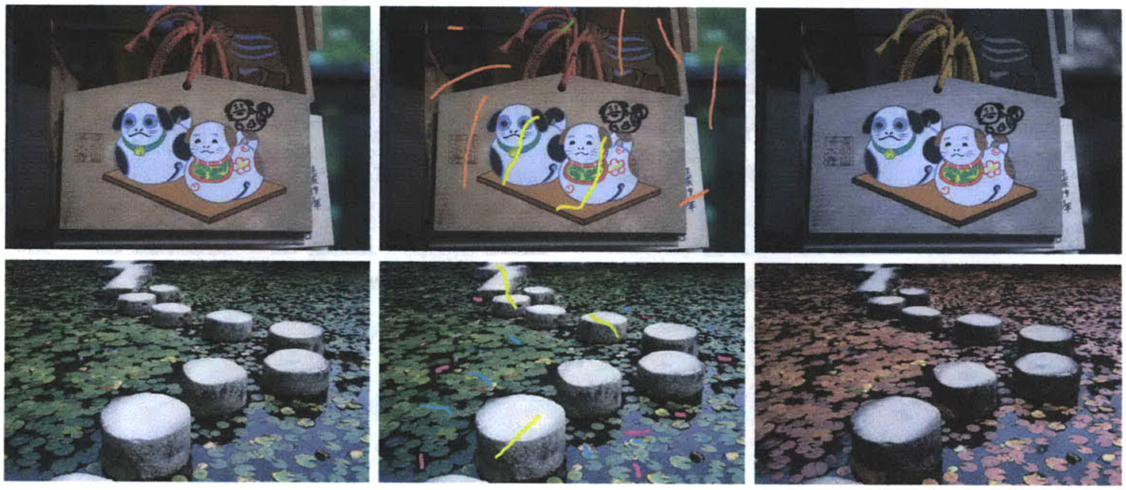


Figure 4-6: Results from our system that involve more than two scribble classes. Left to right: original image, scribbles, and editing result.

supervised methods could significantly improve results. Also, better classifiers may allow the use of extra features, if they can avoid the over-fitting that we sometimes observed.

Edge-aware interpolation offers an attractive and less effort-intensive alternative for local image and video adjustment. By augmenting existing methods with classification we are able to achieve significantly better results than previous work. Our algorithm is quite simple, consisting of a standard and easy-to-implement classifier, the setup and solution of a weighted linear system, and a few flood-fills. We hope to test our system with real users in the near future.

4.9 Discussion

We think an algorithm for interactive segmentation should have some semantic relevance. When two users have identical goals, they should be able to reach essentially identical results. If two users wish to separate the same target from the rest of the image, they should be able to get very similar results regardless of where exactly they lay down their scribbles, how wide they choose their scribbles to be, etc. We conducted a simple experiment to test how our algorithm and the competing algorithms perform in that regard. Five subjects, one of which a co-author, were asked to draw scribbles on three two-class examples: "Buddha" in Figure 4-9, "Birds" in Figure 4-10, and "Girl" in Figure 4-11. All subjects received iden-

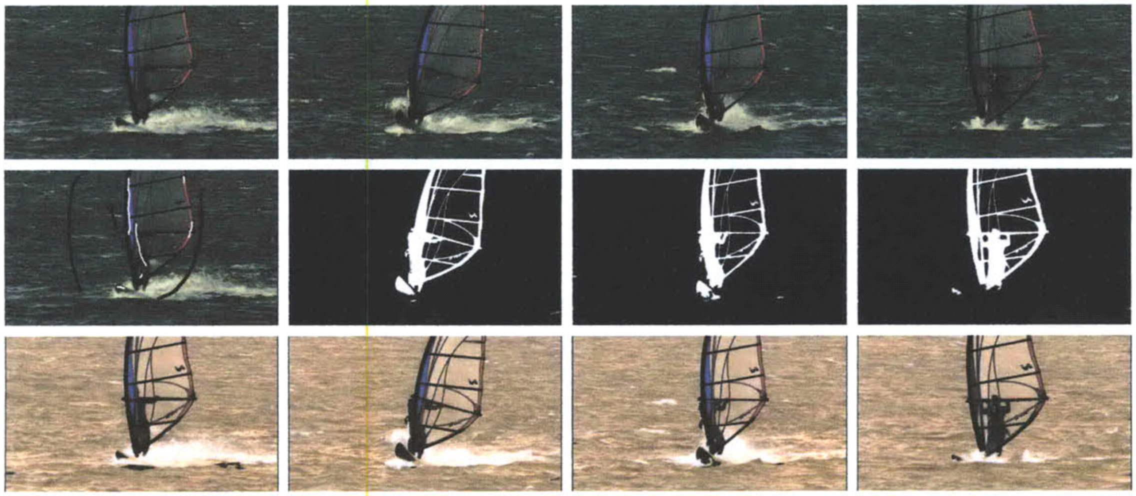


Figure 4-7: A video example, where the water color is adjusted but the windsurfer is unchanged. One out of 123 frames was stroked. Row 1: frames; row 2: scribbles, masks; row 3: adjusted frames.

tical instructions. The instructions take the following form, "draw scribbles of two different colors to separate the *target* from everything else". The "*target*" is, the red coating in "Buddha", the group of birds in "Birds", and the dress in "Girl", respectively. We show the five sets of user scribbles, together with the masks generated using our algorithm, Lischinski et al., Robust Matting, and Lazy Snapping, respectively, in Figure 4-9, 4-10, and 4-11. As shown with the scribbles, the subjects are highly varied in style. Some are very thorough, for example, subject 2 and 3, while others choose to use fewer and thinner scribbles. Based on comparison of results when the same scribbles are fed to both our algorithm (row 2) and the competing algorithms (rows 3-5), ours is shown to be the most robust.

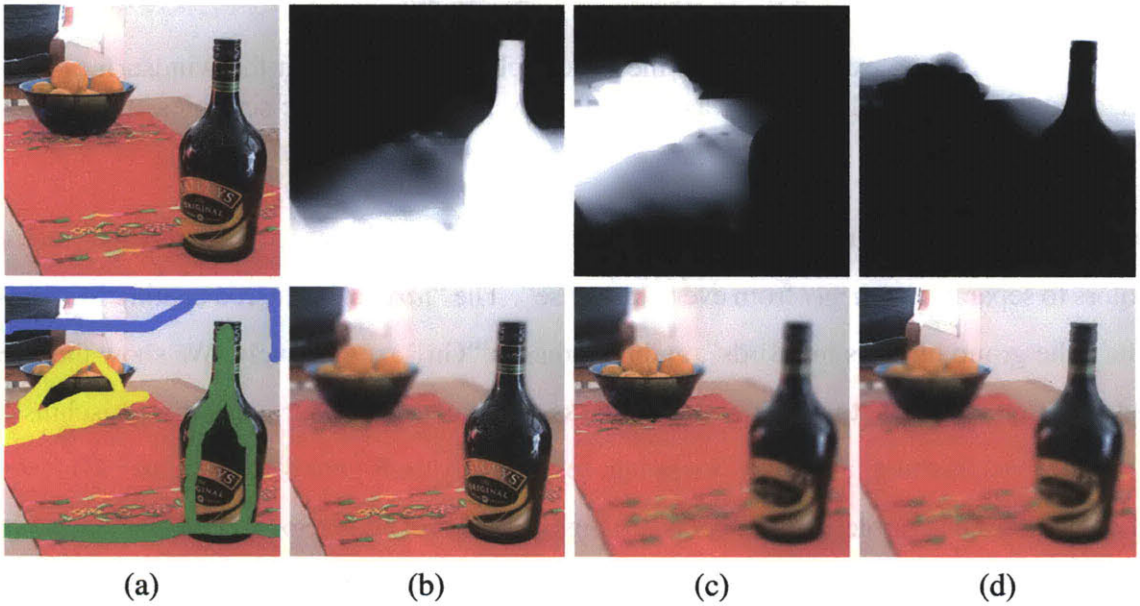


Figure 4-8: An example from the paper of Lischinski et al. that requires long-range, smooth transitions that do not resemble object mattes. (a) Original image (top) and scribbles (bottom) indicating areas for spatially-varying blur. (b-d) Blending weights (top) computed using ScribbleBoost and the depth-of-field effects (bottom) achieved using these masks.

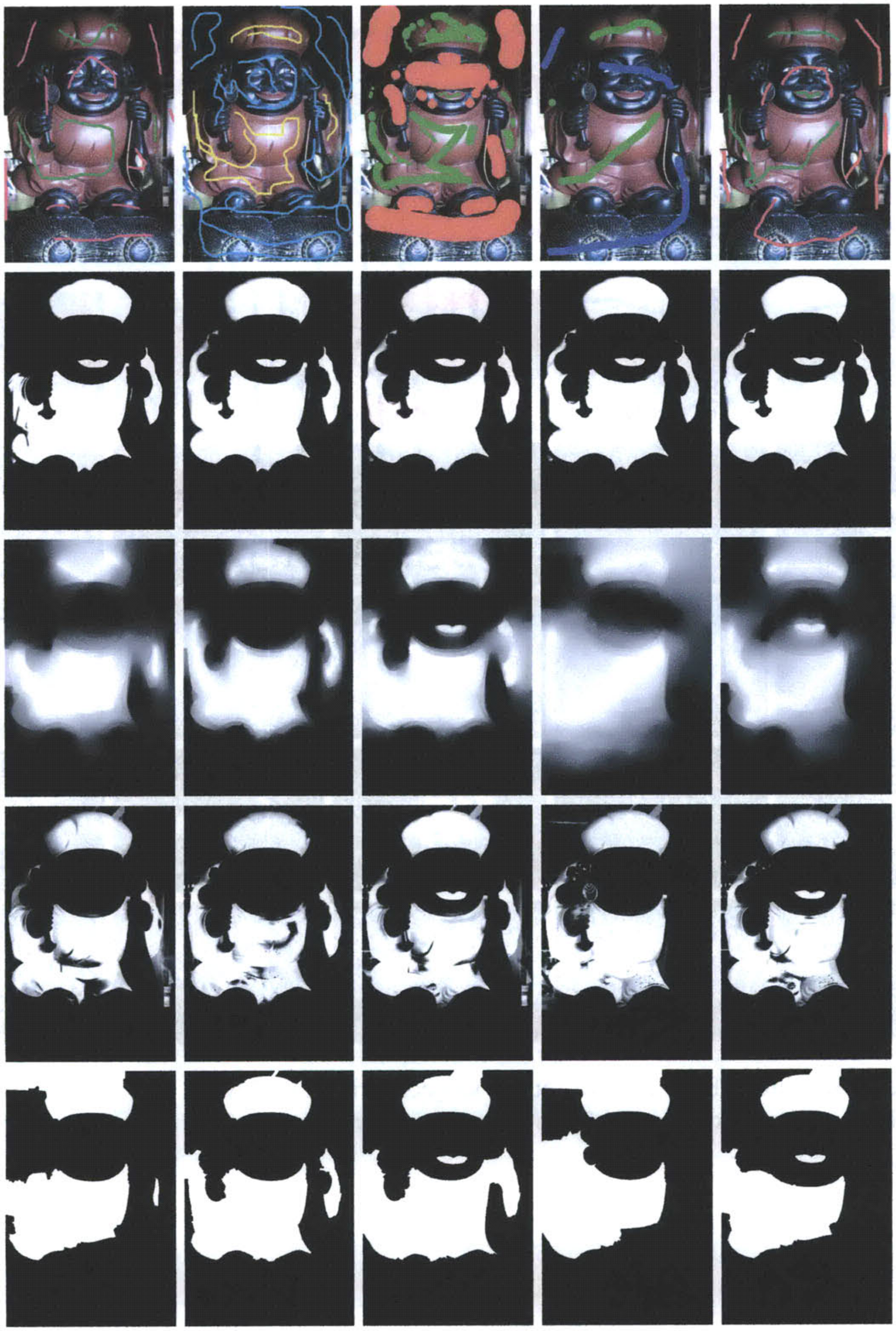


Figure 4-9: Buddha. Row 1: Five sets of subject scribbles. Row 2: Our masks. Row 3: Lischinski et al. masks. Row 4: Robust Matting masks. Row 5: Lazy Snapping masks.

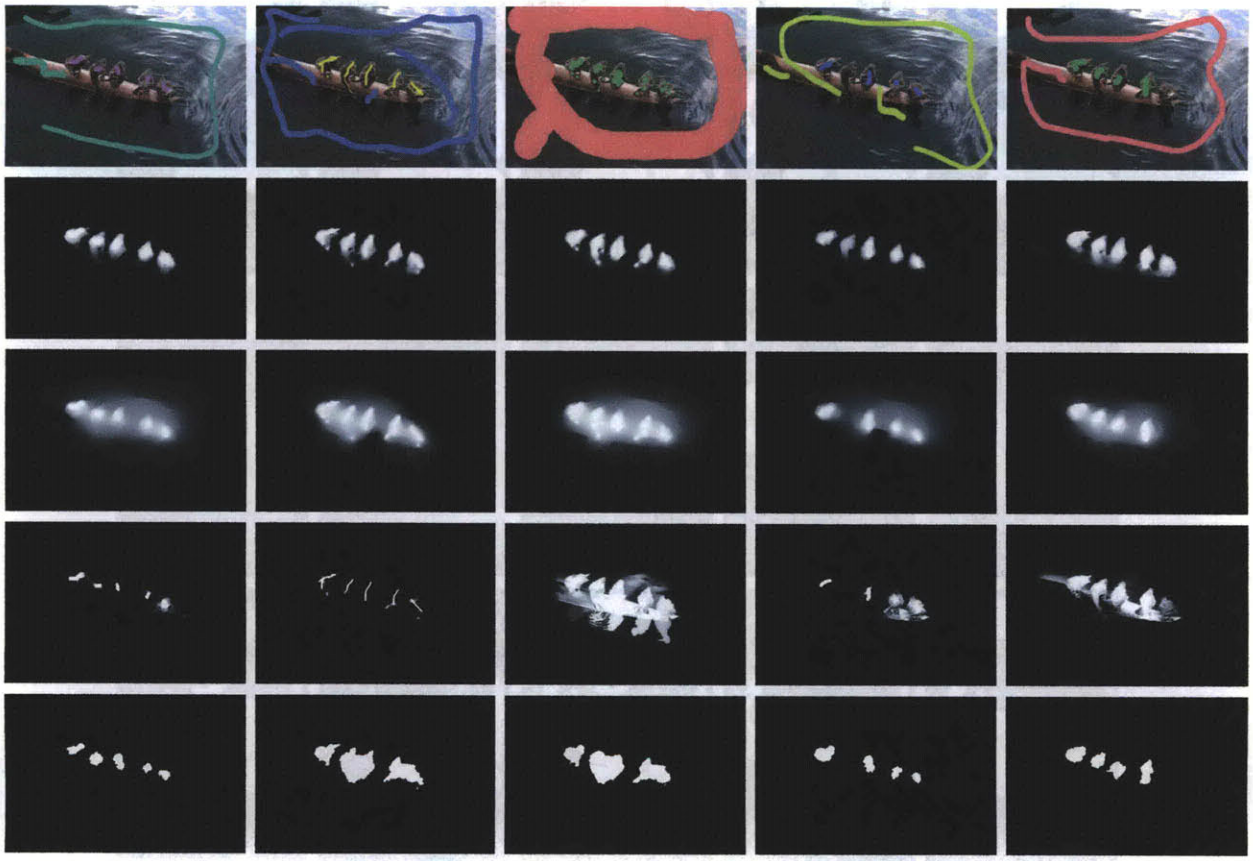


Figure 4-10: Birds. Row 1: Five sets of subject scribbles. Row 2: Our masks. Row 3: Lischinski et al. masks. Row 4: Robust Matting masks. Row 5: Lazy Snapping masks.

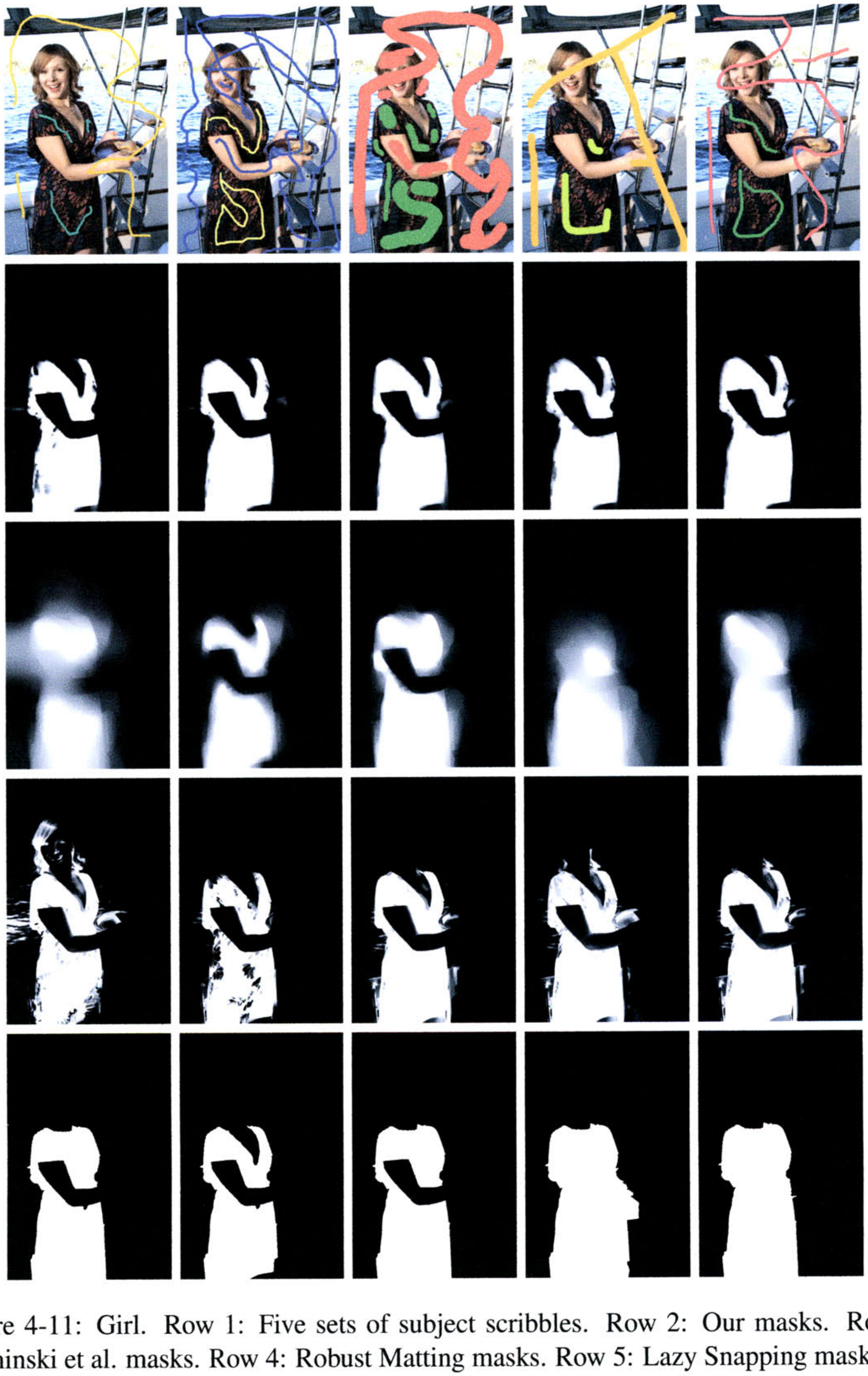


Figure 4-11: Girl. Row 1: Five sets of subject scribbles. Row 2: Our masks. Row 3: Lischinski et al. masks. Row 4: Robust Matting masks. Row 5: Lazy Snapping masks.

Chapter 5

Conclusion

In this thesis, we have presented three image estimation and enhancement algorithms that are inspired by human perception, and that in some cases offer insights into human perception.

In the first part of the thesis, we framed a number of vision and image processing problems as ones of statistical estimation: given an observed image, and the observer's knowledge about the world, estimate an optimal output image. The "knowledge about the world" is learned from a set of training image pairs: noisy vs. clean for denoising, blurry vs. sharp for super-resolution, luminance vs. reflectance for intrinsic image decomposition, etc. We propose methods for capturing local constraints; given an input image patch, what should the output patch be? We do not literally use patches of pixels, but instead use filtered values, i.e., the results of a patch convolved with linear filters, to represent the input patch, and also filtered values to constrain the output patch. Because images when convolved with linear filters display strong regularities, e.g., they demonstrate highly kurtotic distributions, the use of filtered values makes it easier to separate the confounding components, for example, signal and noise. The use of filtered values also makes it possible for us to use a number of heuristics to partition the input feature space into bins that conform to the data density in the space, and are also very easy to query. In addition to local constraints, we propose methods to learn and impose global constraints, represented in the form of subband histograms. Our techniques are demonstrated to be effective for image denoising, super-resolution, and intrinsic image decomposition. They also generate interesting results when used to interpret the relative strengths of lightness illusions.

In the second part of the thesis, we present multi-scale techniques for dynamic range compression. Previous multi-scale techniques for this application acquired the reputation of being hard to use without causing halo artifacts. Taking inspirations from the human visual system, we propose techniques with smooth gain control, an innovation that proves to be crucial in eliminating halo artifacts and producing visually pleasing results with reduced dynamic ranges. Our dynamic range compression scheme can be inverted, so that a high dynamic range image can be compressed to a low dynamic range and quantized into lower bit depths, and later expanded back to high dynamic range with minimal loss of visual quality. This is achieved by iteratively searching for a range-compressed and quantized version of the image, that when range-expanded by inverting the smooth gain control process, gives back a close approximation of the original image. The intermediate, range-compressed version offers a good visual rendition of the high-dynamic-range image on a low dynamic range display. We also compared our companding (compressing-expanding) technique to a related technique called error-diffusion dithering, in terms of how well the original image can be recovered from the quantized versions. Companding is demonstrated to result in much better recovery of the original higher-bit image than dithering.

In the third part of the thesis, we propose a technique that enables a user to easily localize image and video editing by drawing a small number of rough scribbles. Our technique learns a classifier utilizing the user-scribbled pixels as training examples, and then uses the classifier to classify the rest of the pixels into distinct classes. It then uses the classification results as per-pixel data terms, and combines them with smoothness terms that respect edges. The resulting images are better than those produced by state-of-art algorithms for interactive segmentation. We also compare our technique with the competing ones in terms of robustness to user scribbles: when two users wish to separate the same target from the rest of the image, are they able to reach essentially identical results, regardless of the exact way (thoroughly or succinctly, for example) they lay down their scribbles? We conduct an experiment where five users are given the same instructions and asked to provide scribbles. Based on comparison of results when the same scribbles are fed to both our algorithm and the competing ones, our algorithm is shown to be the most robust.

Bibliography

[1] E. H. Adelson and A. P. Pentland. The perception of shading and reflectance. In D. Knill and W. Richards (eds.), Perception as Bayesian Inference (pp. 409-423). New York: Cambridge University Press, 1996.

[2] E. H. Adelson, E. Simoncelli, and R. Hingorani. Orthogonal pyramid transforms for image coding. In *Visual Communications and Image Processing II, Proc. SPIE*, volume 845, pages 50–58, Cambridge, MA, Oct 1987.

[3] E. H. Adelson and E. P. Simoncelli. Hexagonal qmf pyramids. In *Proceedings of the Optical Society of America Topical Meeting on Applied Vision*, pages 5–8, San Fransisco, CA, July 1989.

[4] Edward H Adelson. *M Gazzaniga, M.S., ed., The New Cognitive Neurosciences*, chapter Lightness perception and lightness illusions, pages 339–351. MIT Press, 2 edition, 2001.

[5] L. Arend. Surface colors, illumination, and surface geometry: Intrinsic-image models of human color perception. In A. Gilchrist (Ed.), Lightness, Brightness, and Transparency (pp. 159-213). Hillsdale:Erlbaum., 1994.

[6] Shai Avidan. Spatialboost: Adding spatial reasoning to adaboost. In *ECCV (4)*, pages 386–396, 2006.

[7] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *IEEE International Conference on Computer Vision*, 2007.

[8] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. *A. Hanson and E. Riseman, editors, Computer Vision Systems*, 1978.

[9] T.O. Binford. Inferring surfaces from images. *Artificial Intelligence*, 17:205–244, 1981.

[10] Jeff Bolz, Ian Farmer, Eitan Grinspun, and Peter Schröder. Sparse matrix solvers on the GPU: Conjugate gradients and multigrid. *ACM Transactions on Graphics*, 22(3):917–924, July 2003.

[11] C. R. Carlson, E. H. Adelson, and C. H. Anderson. System for coring an image-representing signal. United States Patent 4,523,230, 1985.

[12] Hong Chen, Ziqiang Liu, Chuck Rose, Yingqing Xu, Heung yeung Shum, and David Salesin. Example-based composite sketching of human portraits. In *Proc. 3rd Int'l Symp. NPAR*, pages 95–153, 2004.

[13] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics*, 26(3):103, 2007.

[14] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 264–271, 2001.

[15] Ingrid Daubechies. *Ten lectures on wavelets.* Society for Industrial and Applied Mathematics, 1992.

[16] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pages 369–378, August 1997.

[17] K. Devlin, A. Chalmers, A. Wilkie, and W. Purgathofer. Star: Tone reproduction and physically based spectral rendering. *In: State of the Art Reports, Eurographics*, pages 101–123, September 2002.

[18] J. M. Dicarlo and B. A. Wandell. Rendering high dynamic range images. In *In Proceedings of the SPIE: Image Sensors*, volume 3965, pages 392–401, 2001.

[19] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[20] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. In *Biometrika*, volume 81(3), pages 425–455, 1994.

[21] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics*, 21(3):257–266, July 2002.

[22] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001.

[23] Katrin Eismann. *Photoshop Masking & Compositing.* Peachpit Press, 2005.

[24] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics*, 21(3):249–256, July 2002.

[25] James A. Ferwerda, Sumant Pattanaik, Peter S. Shirley, and Donald P. Greenberg. A model of visual adaptation for realistic image synthesis. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pages 249–258, August 1996.

[26] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[27] R.W. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale. *Proceedings of the Society of Information Display*, 17(2):75–77, 1976.

[28] W. T. Freeman. Exploiting the generic viewpoint assumption. *International Journal Computer Vision*, 20(3):243–261, 1996.

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

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

[31] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, 1995.

[32] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 2(28):337–374, 2000.

[33] James J. Gibson. *The perception of the visual world.* Houghton Mifflin, Boston, 1950.

[34] L. Grady. Multilabel random walker image segmentation using prior models. In *2005 Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 763–770, 2005.

[35] Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[36] A. Gray and A. Moore. Very fast multivariate kernel density estimation via computational geometry. *Proc. Joint Stat. Meeting*, 2003.

[37] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[38] D. J. Heeger. Half-squaring in responses of cat simple cells. *Visual Neurosci.*, 9:427–443, 1992.

[39] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH*, 1995.

[40] D.J. Heeger. Modeling simple cell direction selectivity with normalized, half-squared, linear operators. *Journal of Neurophysiology*, 70:1885–1898, 1993.

[41] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin. Image analogies. In *SIGGRAPH*, pages 327–340, 2001.

[42] J.E. Hudson. Signal processing using mutual information. *IEEE Signal Processing Magazine*, 23(6):50–58, 2006.

[43] A. Ihler. Kernel density estimation toolbox for matlab. http://www.ics.uci.edu/ ihler/code/, 2003.

[44] IJG. Independent jpeg group. *http://www.ijg.org/files/.*

[45] J. Jarvis, C. Judice, and W. Ninke. A survey of techniques for the display of continuous tone pictures on bilevel displays. *Computer Graphics and Image Processing*, pages 13–40, 1976.

[46] J.L.Bentley. Multidimensional binary search trees used for associative searching. *Comm. ACM*, 18:509–517, 1975.

[47] D. J. Jobson, Z. Rahman, and G. A. Woodell. A multi-scale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing*, 6(7):965–976, July 1997.

[48] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik. A high quality, fast inverse halftoning algorithm for error diffused halftones. In *Proc. of IEEE International Conference Image Processing (ICIP)*, volume 2, pages 59–63, Oct 1998.

[49] T. D. Kite, B. L. Evans, and A. C. Bovik. Modeling and quality assessment of halftoning by error diffusion. *IEEE Transactions on Image Processing*, 9(4):909–922, May 2000.

[50] T. D. Kite, B. L. Evans, A. C. Bovik, and T. L. Sculley. Digital halftoning as 2-d delta-sigma modulation. *Proc. IEEE International Conference on Image Processing*, I:799–802, Oct 1997.

[51] F. Labaere and P. Vuylsteke. Image contrast enhancing method. U.S. Patent no. 5,717,791, 1998.

[52] E. H. Land and J. J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, Jan 1971.

[53] 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.

[54] Hsien-Che Lee. Automatic tone adjustment by contrast gain-control on edges. United States Patent 6,285,798, September 2001.

[55] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[56] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Transactions on Graphics*, 23(3):689–694, August 2004.

[57] Anat Levin, Dani Lischinski, and Yair Weiss. A closed form solution to natural image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 61–68, 2006.

[58] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Transactions on Graphics*, 23(3):303–308, August 2004.

[59] Yuanzhen Li and Edward H. Adelson. Image mapping using local and global statistics. In *Human Vision and Electronic Imaging XIII, Proc. of SPIE-IS&T Electronic Imaging, SPIE*, 2008.

[60] Yuanzhen Li, Edward H. Adelson, and Aseem Agarwala. Scribbleboost: Adding classification to edge-aware interpolation of local image and video adjustments. In *Graphics Forum 27(4), Proceedings of Eurographics Symposium on Rendering*, 2008.

[61] Yuanzhen Li, Lavanya Sharan, and Edward H. Adelson. Compressing and companding high dynamic range images with subband architectures. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 836–844, New York, NY, USA, 2005. ACM.

[62] Dani Lischinski, Zeev Farbman, Matt Uyttendaele, and Richard Szeliski. Interactive local adjustment of tonal values. *ACM Transactions on Graphics*, 25(3):646–653, 2006.

[63] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *Rendering Techniques 2007 (Proceedings Eurographics Symposium on Rendering)*, June 2007.

[64] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Trans. on PAMI*, 14(7):710–732, 1992.

[65] S. Mann and R. W. Picard. Extending dynamic range by combining different exposed pictures. In *Proceedings of IS&T*, pages 442–448, 1995.

[66] D. Marr and T. Poggio. From understanding computation to understanding neural circuitry. *Neurosciences Res. Prog. Bull.*, 15:470–488, 1977.

[67] David Marr. *Vision: A computational investigation into the human representation and processing of visual information.* W. H. Freeman and Co., 1982.

[68] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[69] M. Menendez, D. Morales D, and L. Pardo. Maximum entropy principle and statistical inference on condensed ordered data. *Statistics and Probability Letters*, 34(1):85–93, 1997.

[70] T. Mitsunaga and S. K. Nayar. High dynamic range imaging: Spatially varying pixel exposures. In IEEE CVPR, 472-479, 2000.

[71] David M. Moount and Sunil Arya. Ann: A library for approximate nearest neighbor searching, version 1.1.1. http://www.cs.umd.edu/ mount/ANN/, Aug 2006.

[72] Nakazawa, Masayuki, Tsuchino, and Hisanori. Method of compressing a dynamic range for a radiation image. United States Patent 5,471,987, 1995.

[73] R. Neelamani, R. Nowak, and R. Baraniuk. Winhd: Wavelet-based inverse halftoning via deconvolution. *Submitted to the IEEE Transactions on Image Processing*, September 2002.

[74] Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild, and Donald P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 287–298, July 1998.

[75] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, 1990.

[76] K. Popat and R. W. Picard. Cluster based probability model and its application to image and texture processing. *IEEE Trans. Image Processing*, 6(2):268–284, 1997.

[77] Thomas Porter and Tom Duff. Compositing digital images. In *Computer Graphics (Proceedings of SIGGRAPH 84)*, pages 253–259, July 1984.

[78] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Processing*, 2003.

[79] Alexis Protiere and Guillermo Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Transactions on Image Processing*, 16(4):1046–1057, 2007.

[80] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. *ACM Transactions on Graphics*, 25(3):1214–1220, 2006.

[81] R.C. Reid, R.E. Soodak, and R.M. Shapley. Directional selectivity and spatiotemporal structure of receptive fields of simple cells in cat striate cortex. *J. Neurophysiol*, 66:509–529, 1991.

[82] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM Transactions on Graphics*, 21(3):267–276, July 2002.

[83] S. Roth and M. Black. Fields of experts: A framework for learning image priors. *in Proc. of CVPR, vol. 2, pp. 860–867*, 2005.

[84] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, August 2004.

[85] R. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[86] Helge Seetzen, Wolfgang Heidrich, Wolfgang Stuerzlinger, Greg Ward, Lorne Whitehead, Matthew Trentacoste, Abhijeet Ghosh, and Andrejs Vorozcovs. High dynamic range display systems. *ACM Transactions on Graphics*, 23(3):760–768, August 2004.

[87] E. P. Simoncelli. *Handbook of Image and Video Processing*, chapter 4.7. Statistical modeling of photographic images. Academic Press, 2nd edition, 2005.

[88] E P Simoncelli and E H Adelson. Noise removal via bayesian wavelet coring. In *3rd IEEE Int'l Conf on Image Processing*, Laussanne, Switzerland, Sep 1996.

[89] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D J Heeger. Shiftable multi-scale transforms. *IEEE Trans Information Theory, Special Issue on Wavelets*, 38(2):587–607, 1992.

[90] Pawan Sinha and Edward H. Adelson. Recovering reflectance in a world of painted polyhedra. In *Proceedings of Fourth International Conference on Computer Vision*, pages 156–163, 1993.

[91] T.G. Stockham. Image processing in the context of a visual model. *Proc. IEEE*, 60:828–842, 1972.

[92] G. Strang. *Introduction to Applied Mathematics*, chapter 2.5 Least squares estimation and the Kalman filter, pages 146–148. Wellesley-Cambridge Press, 1986.

[93] Richard Szeliski. Locally adapted hierarchical basis preconditioning. *ACM Transactions on Graphics*, 25(3):1135–1143, July 2006.

[94] M. F. Tappen, E. H. Adelson, and W. T. Freeman. Estimating intrinsic component images using non-linear regression. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[95] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472, 2005.

[96] D.J. Tolhurst and A.F. Dean. Evaluation of a linear model of directional selectivity in simple cells of the cat's striate cortex. *Vis Neurosci.*, 6(5):421–428, 1991.

[97] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *In Proc. IEEE Int. Conf. on Computer Vision*, pages 836–846, 1998.

[98] J. Tumblin. *Three methods of detail-preserving contrast reduction for displayed images*. PhD thesis, College of Computing Georgia Inst. of Technology, 1999.

[99] Jack Tumblin and Holly E. Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics & Applications*, 13(6):42–48, November 1993.

[100] Jack Tumblin and Greg Turk. Lcis: A boundary hierarchy for detail-preserving contrast reduction. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 83–90, August 1999.

[101] P. Vuylsteke and E. Schoeters. Method and apparatus for contrast enhancement. U.S. Patent no. 5,805,721, 1998.

[102] Jue Wang. Discriminative Gaussian mixtures for interactive image segmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 386–396, 2007.

[103] Jue Wang and Michael F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 936–943, 2005.

[104] Jue Wang and Michael F. Cohen. Optimized color sampling for robust matting. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[105] Greg Ward and Maryann Simmons. Subband encoding of high dynamic range imagery. In *APGV '04: Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 83–90. ACM Press, 2004.

[106] Gregory J. Ward. The radiance lighting simulation and rendering system. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, pages 459–472, July 1994.

[107] Liron Yatziv and Guillermo Sapiro. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing*, 15(5):1120–1129, 2006.