MULTIDIMENSIONAL MODELLING OF IMAGE FIDELITY MEASURES

by

Mason Macklem
B.A. (Mathematics), Okanagan University College, 2000

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTERS OF SCIENCE
in the School
of
Mathematics

SIMON FRASER UNIVERSITY
December 2002

APPROVAL

Name: Mason Macklem

Degree: Masters of Science

Title of thesis: Multidimensional Modelling of Image Fidelity Measures

Examining Committee: Dr. Petr Lisonek
Chair

Dr. Jonathan Borwein
Senior Supervisor
Simon Fraser University

Dr. Adrian Lewis
Simon Fraser University

Dr. Torsten Möller
Simon Fraser University

Dr. Philip Loewen
External Examiner
University of British Columbia

Date Approved: ______________________

Abstract

Over the past decade, the JPEG image compression standard has become the benchmark for compressing natural still images for use on the internet. This standard has been extremely successful in maintaining a balance between low file-size and high subjective image quality. However, JPEG-encoded images can exhibit unexpected objectionable artifacts. For example, for images already encoded using compression methods based around the *Discrete Cosine Transform* (DCT), further compression by requantization can create objectionable visual artifacts. This phenomenon, which we termed the *JPEG Perceptual Dipper Effect*, is an example of the more general concept of *visual thresholds*, where two approximations of the same original image, neither of which differ dramatically from each other in terms of a given numerical measure, differ dramatically in the visual quality of their approximation of the original image.

The problem with detecting visual thresholds numerically is that many quality measures detect specific phenomena separately, motivated in part by the observation that subjects, while giving similar results to each other when measuring quality by detecting specific simple artifacts, differ dramatically from each other when the simple artifacts are combined to form more complex artifacts. This problem of the multiple dimensions of errors could be avoided by treating each dimension separately, giving an error representation in multiple dimensions.

In this thesis, we discuss the JPEG Perceptual Dipper Effect, showing how it can be easily avoided by simple calculations in the process of designing the quantization matrices used to requantize the given image. We discuss various commonly used measures of the perceptual quality of digital images, comparing their ability to predict the Dipper Effect. We also discuss and compare two methods of analyzing multidimensional data, Multidimensional Scaling (MDS) and Multicriteria Optimization (MCO), showing approaches of modelling visual thresholds with each method.

To my family.

"Modelling image quality without considering the visual system could be like modelling food quality without considering the digestive system."

— Sergei Yendrikhovskij, Color reproduction and the naturalness constraint [84].

Acknowledgments

The work contained in this thesis was made possible by the generous financial, intellectual, emotional and spiritual support of a great number of people.

I would first like to thank everyone at Packeteer, who provided generous financial and intellectual support. The researchers at Packeteer have helped me immeasurably, answering many of my pressing questions and providing me with directions in which to focus my energy. In particular, I want to thank Nicholas Swart, who first directed me to the problem of measuring perceptual image fidelity, and Heinz Bauschke, whose mentorship I have always valued greatly. As well, special thanks to Paul Archard, Chris Hamilton, Aaron Lees, Mike McGuire, and Justin McMichael.

Secondly, I would like to thank my supervisor, Dr. Jonathan Borwein, for his direction, support, and encouragement during the past two years, as well as for his inspirational leadership in gathering a wide variety of technical resources. My time within the Centre for Experimental and Constructive Mathematics (CECM) has been exciting and stimulating, and the new CoLab environment has enabled new and interesting methods for demonstrating the visual phenomena I have been studying.

In addition, special thanks to Dr. A. Lewis, Dr. P. Loewen, and Dr. T. Möller, whose helpful comments have made the process of finishing this thesis much less stressful. I look forward to continuing to work with each of you in the future.

I would also like to thank my fellow students at Simon Fraser University, for helping to create the optimal work/research environment, especially my officemates (Danny Dyer, Warren Hare, Alan Meichsner, Beth Powell, Tara Stuckless, Jodie Wallis, and Herre Wiersma), and my roommates (Shaun Betts, Reuben Bonaparte, Amit Kumar Goel, Jamie McGough, Claudia Ruitenberg, Pavlina Vagnerova, and Ivan Zuboff). Also, special thanks to Jeff van Dyk for helpful advice in learning the programming languages used to write my software, and to Aaron Peck for providing me with a venting medium when necessary.

In addition, special thanks to the students in Campus Crusade for Christ, whose spiritual passion has been an inspiration for me when my work has discouraged me, especially Jon Decker, Jennifer Duckhorn, and John Kim. Also, thanks to Jennifer and Ben Nyland for helping to lead me back to the Church; as these pages show, God has given me a great many things to be thankful for, and I thank you for helping me realize this fact.

Finally, the work in this thesis was also made possible by financial support from an NSERC PGS-A scholarship, as well as from the Canada Research Chair. Special thanks to both NSERC and the CRC for making my research possible.

Contents

List of Tables

List of Figures

Table of Acronyms

AC	Alternating Current
CSF	Contrast Sensitivity Function
DC	Direct Current
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
EOS	Efficiency of Search
FFT	Fast Fourier Transform
HVS	Human Visual System
IJG	Independent JPEG Group
JPEG	Joint Photographic Experts Group
MCO	Multicriteria Optimization
MDS	Multidimensional Scaling
MSE	Mean-Squared Error
MOP	Multicriteria Optimization Problem
MOS	Mean Opinion Score
MPEG	Motion Picture Experts Group
NISE	Noninferior Set Estimation
PSNR	Peak Signal-to-Noise Ratio
RMS (RMSE)	Root Mean-Squared Error
ROI	Region of Interest
SVD	Singular Value Decomposition
VDP	Visible Differences Predictor
VQEG	Video Quality Experts Group

Chapter 1

Introduction

1.1 Image Compression Model

Conceptually, the problem of designing image compression algorithms follows the general structure of modifying the *representation of the input signal* such that the *imperceptible content* is *easily discarded*. This basic formulation outlines three major areas of focus in the image compression literature:

- designing methods of signal representation, generally by linear transform, to make compact signal content that contributes to the visual appearance of the input image;
- modelling the perceptual fidelity of an image to accurately predict when signal errors will be visible due to the loss of data during encoding;
- designing methods of quantization, to efficiently remove redundant signal content.

This formulation illustrates the need for an accurate image fidelity measure: for an original image X and compressed image $\hat{X}$, a *perceptual fidelity measure* $q(X, \hat{X})$ can provide constraints that can be used in the process of designing an image compression algorithm, namely solving the (ill-defined) optimization problems

$$\min_{q(X,\hat{X}) \geq m} f(\hat{X}) \quad and \quad \max_{f(X) \leq M} q(X, \hat{X}), \tag{1.1}$$

where $f(X)$ is the filesize of X. In other words, given an accurate measure of perceptual image fidelity, the goal of the compression algorithm is to remove as much data as possible from the input image while maintaining its visual appearance (to an acceptable level), or of achieving as high an image fidelity as possible while maintaining a given maximum bit-rate.

1.1.1 Quality vs. Fidelity

In the discussion of the general model, a few useful distinctions should be mentioned. First, *image quality* refers to the presence (or absence) of a given property within a given image, and is measured without a reference image; for example, image quality has been referred to as the image's "naturalness" [84], "usefulness" [31], or more generally as the overall impression created by the image. On the other hand, *image fidelity* refers to the "closeness" of a given image to a reference (ideal, original) image, and thus is a comparative measure of the distance between a pair of images[1].

In general, image quality is much harder to measure, and is specific to the image's intended application. However, many industrial applications of digital imaging, such as medical or astronomical imaging, are limited to considering image quality due to the lack of an existing original or ideal image. The application of compressing images for use on the internet requires a balance between quality and fidelity; although the visitor to a web-site will only be aware of the final compressed version of an image on the site, as a representation of the image creator and/or owner, concern should be taken to best represent the intent and form of the original image.

1.2 Mean-Squared Error

The traditional image fidelity standard is the *mean squared-error* (MSE): for X an $M \times N$-pixel image with approximation $\hat{X}$,

$$\mathrm{MSE}(X, \hat{X}) = \frac{1}{MN} \sum_{i=0}^{M} \sum_{j=0}^{N} (X_{i,j} - \hat{X}_{i,j})^2. \tag{1.2}$$

Also frequently encountered in the literature are measures related to the MSE, namely the *root mean-squared error* (RMS or RMSE), the *signal-to-noise ratio* (SNR), and the *peak*

[1]Thus, an image fidelity measure is also commonly referred to as a *image metric*.

signal-to-noise ratio (PSNR):

$$\mathrm{RMS}(X,\hat{X}) = \sqrt{\frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N}(X_{i,j}-\hat{X}_{i,j})^2} \tag{1.3}$$

$$\mathrm{SNR}(X,\hat{X}) = 10\log\frac{\sum_{i=0}^{M}\sum_{j=0}^{N}(X_{i,j})^2}{\sum_{i=0}^{M}\sum_{j=0}^{N}(X_{i,j}-\hat{X}_{i,j})^2} \tag{1.4}$$

$$\mathrm{PSNR}(X,\hat{X}) = 10\log\frac{(2^B-1)^2}{\sum_{i=0}^{M}\sum_{j=0}^{N}(X_{i,j}-\hat{X}_{i,j})^2}, \tag{1.5}$$

where B is the *color-depth* (the number of bits used to represent a single color index) of the input image. We will refer to the MSE, RMS, SNR and PSNR collectively as the *MSE-variants.*

Some initial observations about the MSE-variants:

- the MSE-variants are all image fidelity measures, or measures of closeness between two images; in particular, they are *local fidelity measures*, or measures that only consider differences between corresponding pixels;
- the MSE-variants are all *scalar-valued measures*, in that each measure returns a single real-number intended to represent the global image fidelity;
- the MSE-variants weight the contributions of errors of equal magnitudes the same, regardless of their location in the image or of their relative luminance values;
- the MSE, RMS and PSNR are *topologically equivalent*, in that given a pair of approximations of a given image, each will return the same approximation as being "closer" to the original image; the SNR is also topologically equivalent to the MSE, provided we fix the reference image as the first argument, as $\mathrm{SNR}(X,Y) \neq \mathrm{SNR}(Y,X)$.

In general, the MSE-variants have been used largely due to their ease of formulation and implementation, as well as their mathematical simplicity. In particular, when treating images solely as matrices, independent of their visual interpretation, the MSE can be interpreted intuitively as the Euclidean distance on $\Re^{MN}$. However, each of the above observations contributes to the inability of the MSE-variants to accurately predict image quality for basic properties of the human visual system, and each will serve as a starting point for the refinements discussed in Chapter 3.

1.3 Mean-Squared Error Counterexamples

Comparisons of the MSE to quality ratings determined through subjective tests have consistently given low correlation values of approximately 0.60-0.65, which is often used both as proof of the limitations of the MSE as a quality measure and as a benchmark for proposed improvements. However, we can demonstrate directly the shortcomings of the MSE using basic properties of the human visual system, specifically luminance masking (Section 1.3.1), contrast sensitivity (Section 1.3.2), and texture masking (Section 1.3.3)[2].

1.3.1 Luminance Masking

Luminance masking, also referred to as *light adaptation* and *amplitude nonlinearity*, is the relationship between the human visual system's ability to detect differences in luminance levels, and the absolute magnitude of the luminance levels. This property is illustrated in Figure 1.1: shown are two sets of four squares, with each set having a common background luminance, $L_o = 128$ for (top) and $L_o = 192$ for (bottom), using 8-bit grey-levels. For each set, the top-left square shows the constant background, while the remaining three squares show a centrally-located smaller square of increasing luminance, in unit increments from left-to-right. It is clear that the patches are more noticeable for the squares with the higher background luminance; the central patches are imperceptible for all three squares at $L_o = 128$, while the central patch is quickly perceivable for $L_o = 192$. The key observation is that each of the three test patches at $L_o = 128$ would return the same MSE value as the corresponding test patches at $L_o = 192$, despite the fact that the patches are indistinguishable from the background in the former, while clearly distinguishable in the latter. This demonstrates the inability of the MSE to account for luminance masking when measuring image quality. Luminance masking will be discussed further in Section 3.1.1.

[2]The following figures lose some of their effect when translated to print media; images illustrating visual phenomena throughout this thesis can be viewed online at `http://www.cecm.sfu.ca/~mmacklem/thesis/thesis.htm`.

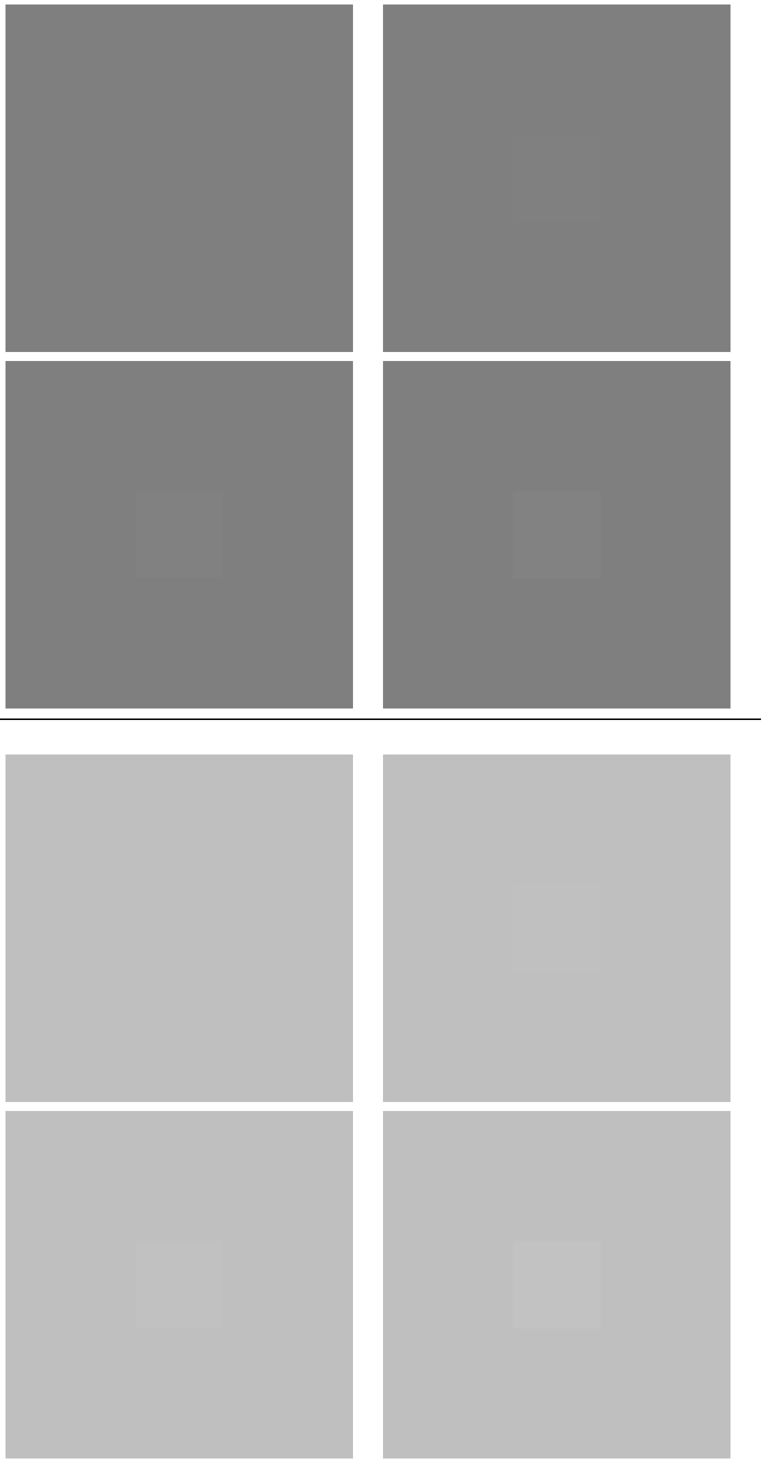

Figure 1.1: Inability of mean-squared error to account for light adaptation: (top) four frames $\{a_i\}_{i=1}^4$, each with constant background luminance $L_o^1 = 128$, with centered smaller squares of increasing luminance, $L_t = 0, \ldots, 3$; (bottom) four frames $\{b_i\}_{i=1}^4$, with constant background luminance $L_o^2 = 192$, with the same increments in luminance $L_t = 0, \ldots, 3$. Note that for each frame, $MSE(L_o^1, a_i) = MSE(L_o^2, b_i)$ for $i = 1, \ldots, 4$.

1.3.2 Contrast Sensitivity

Contrast sensitivity refers to the nonlinear relationship between a given frequency and the human visual system's ability to detect its presence. This is illustrated using a process similar to that described the previous section: a given frequency is added to a constant background, and its amplitude is adjusted until it is detectable.

This is illustrated in Figure 1.2: shown are three sets of three squares, with constant frequency and increasing amplitude across each row, and increasing frequency down each column. Within each row, the amplitude is adjusted so that the MSE of each image compared to the constant background is constant in each column. We can see the nonlinear relationship between the frequency and the detection threshold: the visibility threshold of the low-frequency images and the high-frequency images (top and bottom rows) is lower than that of the middle frequency images (middle row). This phenomenon will be described further in Section 3.1.2.

1.3.3 Texture Masking

Texture masking, also referred to as *content masking* or *contrast masking* [51], refers to the human visual system's decreased sensitivity to a given signal due to the presence of another signal with similar frequency and location. This is illustrated in Figure 1.3: in (a), the left column shows a constant image (top) and random noise image (bottom), normalized such that the average lumina are equal, while the right column shows a sine grating added to both fields. Notice how the presence of the sine grating is masked by the high-frequency noise, but is quite visible in the constant image. Similarly, (b) shows the same experiment applied to the test image `bird` and `mandrill`; notice how the speckled content in the `mandrill` image masks the presence of the sine grating. In both cases, the MSE of the top and bottom image pairs will be equal, despite the bottom image pairs looking almost identical. This illustrates the inability of the MSE to account for texture masking.

1.3.4 JPEG Perceptual Dipper

For our present discussions, the most important phenomenon that is not predicted by the MSE-variants is that of the *JPEG perceptual dipper*, or simply the *perceptual dipper*, which we have discussed in [7, 8]. The Perceptual Dipper refers to the fact that, for images already

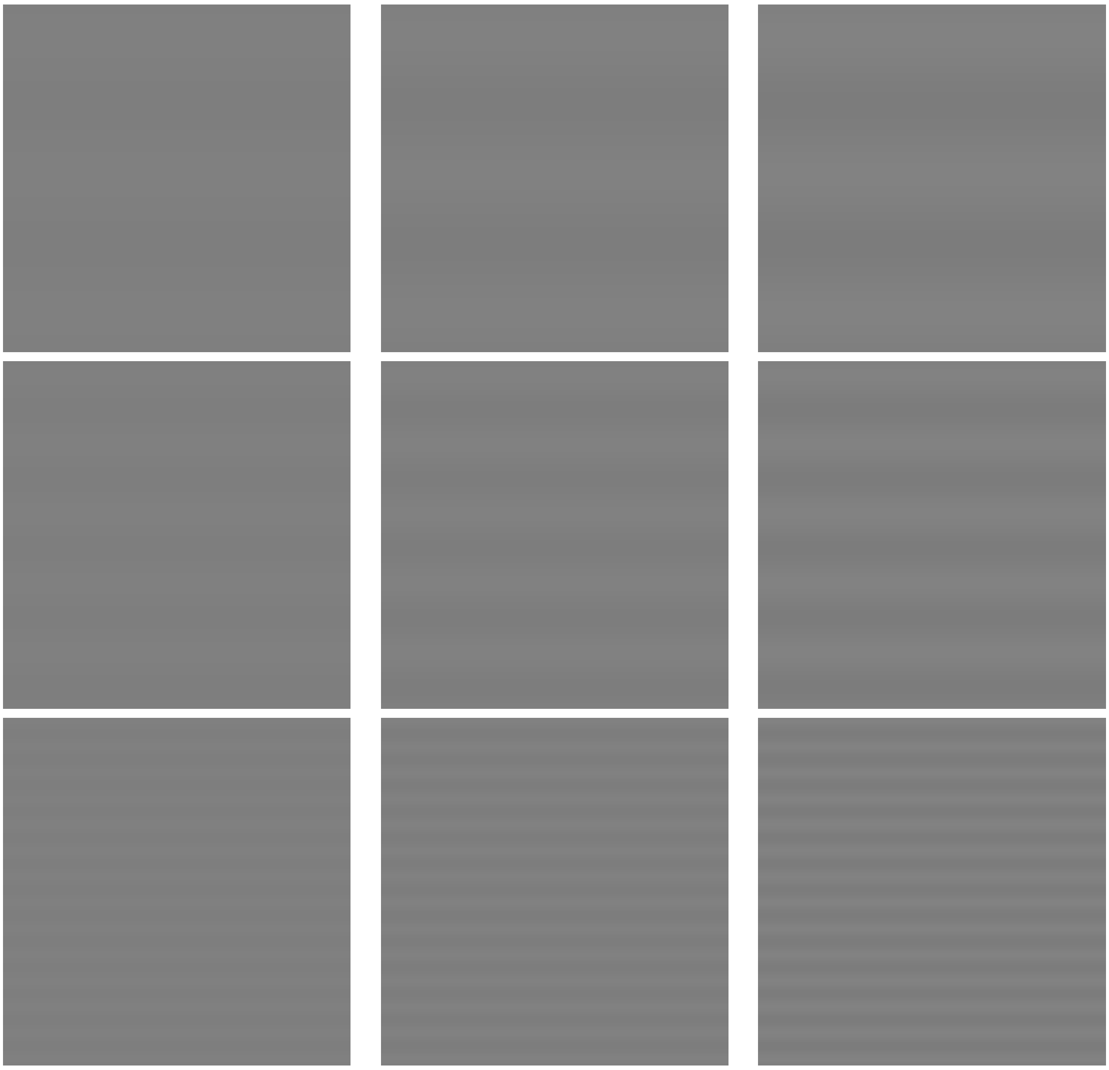

Figure 1.2: Inability of mean-squared error to account for contrast sensitivity: shown is a 3×3 grid, with increasing frequency down each column, and increasing amplitude across each row; the amplitude is adjusted such that the MSE of each image compared to the constant background is the same within each column. Notice that the top and bottom rows are more quickly visible than the middle row, showing the non-linear relationship between visibility threshold and frequency.

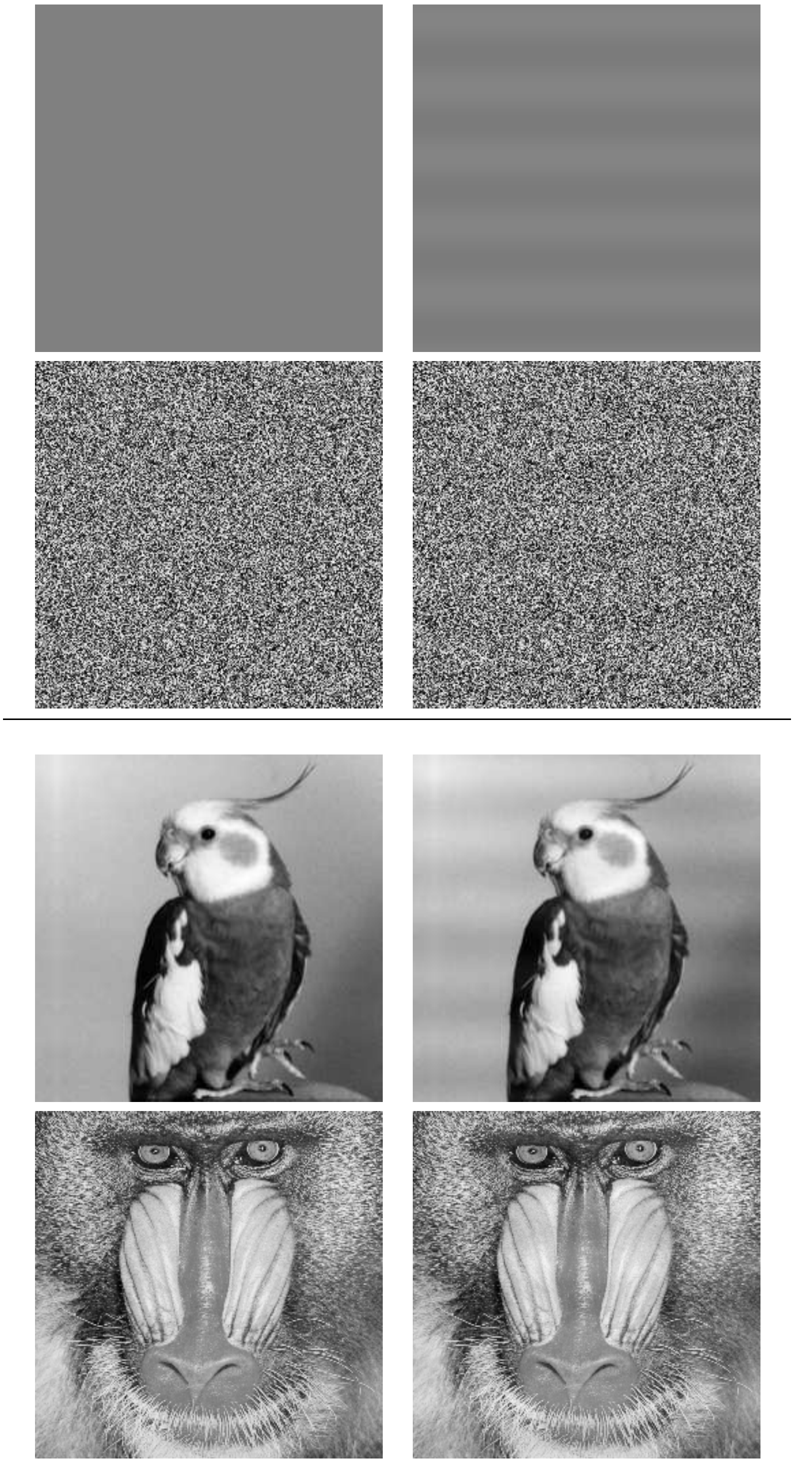

Figure 1.3: Texture masking illustrated using (top) artificial and (bottom) natural images: the sine grating is clearly visible for images with predominantly low-frequency content (top row), while it is masked by the presence of high-frequency image content (bottom row).

encoded using compression methods based on the Discrete Cosine Transform (DCT), blindly requantizing these images to decrease the file-size can create objectionable visual artifacts, which can be avoided by further compressing the image, thus giving an image of a smaller filesize and a closer visual approximation to the original image. However, we need some background to the JPEG image compression standard prior to discussing this phenomenon, and will discuss both in-depth in Chapter 2. We mention this phenomenon here to place our future discussion in the context of the other MSE counterexamples.

1.4 Outline

This thesis outlines a number of ways that properties of the human visual system (HVS) can be used to measure the visual quality of a compressed digital image, which we have seen are not modelled accurately by the MSE. In particular, we will discuss established perceptual quality measures applied to images encoded according the JPEG image compression standard, and in particular to the JPEG Perceptual Dipper. The perceptual dipper is an example of what I call a *visual threshold*, where two approximations of the same original image, neither of which differ dramatically from the other in terms of a given numerical measure, differ dramatically in the visual quality of their approximation of the original image.

The problem with detecting visual thresholds numerically is that many fidelity measures detect specific phenomena separately; this is partly due to the empirical observation that subjects, while giving similar results to each other when measuring quality by detecting specific simple artifacts, will differ dramatically from each other when the simple artifacts are combined to form more complex artifacts [4]. This problem of the multiple dimensionality of perceptual errors, namely that subjects differ from each other in their response to different types of perceptual artifacts, could be avoided by treating each dimension separately, giving an vector-valued fidelity measure; in this way, we keep the underlying structure of each perceptual measure and thus avoid a weighting specific to a single subject.

In this thesis, we will discuss in more detail each section of the situations outlined above. In Chapter 2.1, we discuss the JPEG image compression standard, including a discussion of the Discrete Cosine Transform (DCT) in Section 2.1.1. In Chapter 2.2 we describe methods of post-processing and pre-processing compressed images, including the JPEG Perceptual Dipper in Section 2.3. Next, we discuss established perceptual measures based on the literature on the Human Visual System 3.1 and Human Attention 3.2.1. Finally, we compare two methods of modelling multiple dimensional problems, Multidimensional Scaling in Chapter 4.3 and Vector Optimization in Chapter 4.4.

1.5 Software

In the course of the research contained in this thesis, we have focussed on enabling the reader to repeat all of our experiments. Details on our software are located in Appendix A; throughout this thesis, we will refer to specific demonstrations within the software to highlight the relevant discussions, as many of the phenomena we discuss do not translate to print media.

Chapter 2

JPEG Image Compression and the Perceptual Dipper

Over the past decade, the JPEG image compression standard has become the benchmark for compressing natural still images for use on the internet. This standard has been extremely successful in maintaining a balance between low file-size and high visual fidelity. However, in cases where maintaining high image quality is a higher priority than low file-size, improved image quality can be achieved by various post-processing techniques, where the image is processed in the decoding stage to remove noise added during the encoding stage.

The primary disadvantage of post-processing methods are that they do not conform to the accepted JPEG standard, and therefore the processed files will not be readable by standard JPEG readers. However, staying within the confines of the JPEG standard can often have unexpected visual results. For example, for images already encoded using compression methods based around the *Discrete Cosine Transform* (DCT), further compression by re-quantization can create objectionable visual artifacts. This phenomenon, which we termed the *JPEG Perceptual Dipper Effect*, can easily be avoided by simple pre-calculations ([7, 8]).

In this chapter, we will describe the baseline subset of the JPEG Image Compression Standard (Section 2.1), focussing on the Discrete Cosine Transform (Section 2.1.1) and the IJG family of quantization matrices (Section 2.1.2). We then briefly describe some common pre- and post-processing methods, focussing on the quantization matrix design model of Peterson, Ahumada and Watson (Section 2.2.2) and the de-blocking algorithm of Ahumada and Horng (Section 2.2.1), respectively. Finally, in the key discussion of this Chapter, we introduce the Perceptual Dipper, provide numerical justification for its occurrence, and provide a heuristic to avoid artifacts arising as a result of it (Section 2.3.1); we also describe subjective experiments that validate our heuristic 2.3.4.

2.1 JPEG Compression

The *JPEG* (Joint Photographic Experts Group) *Still-Image Compression Standard*, released in 1991, was designed as a general-purpose image compression format, that could effectively balance compression rate and image fidelity over a wide range of bit-rates, while still being computationally feasible for lower-memory computing environments [42, 53, 79]. The JPEG standard itself has four components:

- *sequential encoding*: the image is partitioned into sub-blocks, which are then encoded and included in the bitstream via a left-to-right, top-to-bottom sequence;
- *progressive encoding*: the image is encoded into layers, which are sent in sequence allowing lower-resolution image approximations to be decoded while the entire bitstream is sent;
- *lossless encoding*: the image is encoded as an exact representation of the original data, resulting in no data loss but in lower compression ratios;
- *hierarchical encoding*: multiple resolutions of the image are encoded sequentially in the bitstream, so that lower resolution versions can be decoded independently of the entire high-resolution image.

In general, JPEG implementations will not include the entire standard, and will frequently implement only the baseline sequential JPEG algorithm [42, 53, 79]:

Algorithm 1 ***(JPEG Baseline sequential compression)***

1. *The given (greyscale or color) image is separated into three color components (luma, chrominance blue and chrominance red) [56].*
2. *Each color component is partitioned into* 8×8 *blocks, extending the image if the dimensions are not multiples of 8; common techniques for extending the image include* zero-padding *(adding zeros to the image),* symmetric extension *(add a mirror-image of the image) and* periodic extension *(add the beginning of the image).*

3. *Each block is transformed using the two-dimensional Discrete Cosine Transform (DCT), described in Section 2.1.1.*

4. *Each transformed block is quantized with respect to an 8-by-8 quantization matrix, described in Section 2.1.2.*

5. *The resulting quantized data is (Huffman or arithmetic) encoded.*

The components of this algorithm of interest for the present discussion are performed in steps 3 and 4, which we will discuss in more detail.

2.1.1 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is designed to take advantage of spatial redundancy that is generally present in natural images, in that pixels located close to each other will frequently contain similar information. In addition, the DCT is designed to provide a joint spatial/frequency representation, in that the location of each 8x8 block within the DCT representation of the image provides spatial information about the location of the corresponding pixels in the decoded image, while the DCT coefficients provide frequency information about the corresponding 8x8 subregion of the image [60]. For an $N \times N$ sub-block $[i_{j,k}]_{i,j=1}^{N}$, the DCT coefficients $[I_{m,n}]_{m,n=1}^{N}$ are defined by

$$I_{m,n} = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} i_{j,k} c_{j,m} c_{k,n} \text{ for } m, n = 0, \ldots, N-1 \tag{2.1}$$

where

$$c_{j,m} = \alpha_m \cos\left(\frac{\pi m}{2N}[2j+1]\right), \quad \text{and} \quad \alpha_m = \begin{cases} \sqrt{\frac{1}{N}} & m = 0 \\ \sqrt{\frac{2}{N}} & m \geq 0. \end{cases} \tag{2.2}$$

The Inverse Discrete Cosine Transform (IDCT) is defined by

$$i_{j,k} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} I_{m,n} c_{j,m} c_{k,n} \text{ for } j, k = 0, \ldots, N-1. \tag{2.3}$$

For an 8×8 block, the recommended block-size, the resulting DCT coefficients measure the weighting of the corresponding 8×8 DCT basis matrix within the original image block; the 64 DCT *basis matrices*, shown in Figure 2.1, are constructed by performing the IDCT on a single 8×8 block with only a single nonzero entry, whose location changes for each of the basis matrices. The (1,1)-coefficient is referred to as the *DC coefficient*, contains the (scaled) average value of the entries in each block, and generally dominates the content within the block; the 63 remaining entries are the *AC coefficients*. Notice that the DCT coefficients corresponding to the basis matrices containing higher-frequency coefficients occur in the bottom-right half of the DCT matrix; this fact will be useful during the quantization and encoding steps of the JPEG compression algorithm. Sample DCT output is shown in Figure 2.2 for the `bird` test image; the partitioning of the image into 8×8 blocks can be observed by the high-magnitude DC coefficients in each block.

2.1.2 Quantization Matrices

The DCT encoding process is lossless, and thus the data representation is still invertible. Most of the data compression (and thus data loss) is achieved during the subsequent quantization process, which is defined by an $N \times N$ quantization matrix (for each color channel), with the same block-size used for both the DCT and quantization stages. For each $N \times N$ block, the DCT coefficients are divided by the corresponding entry of the $N \times N$ quantization matrix, then rounded. During decoding dequantization reverses the steps, multiplying each quantized entry by the corresponding entry of the quantization matrix.

The design of the quantization matrices takes advantage of the human visual system's contrast sensitivity function (Sections 1.3.2 and 3.1.2), using the decreased sensitivity to higher frequencies to remove imperceptible data. The most common class of quantization matrices is defined by the *Independent JPEG Group* (IJG)[1]. This class of matrices is defined by a scalar weighting of a single quantization matrix Q^{IJG} for each channel, and are defined as a function $Q : \{0, \ldots, 100\} \to \mathbb{M}^{N \times N}$ of an integer quality parameter by

$$Q_{i,j}(q) = \text{round}\left(\frac{s(q)}{100} Q_{i,j}^{IJG}\right) \tag{2.4}$$

where Q^{IJG} is the IJG standard matrix, shown in Figure 2.4, and $s(q)$ is a piecewise function,

[1] The original JPEG standard described the structure of JPEG file features, including those listed in the discussion at the beginning of Section 2.1; however, they did not define a file format standard to enable compliance across JPEG codecs. The file format was defined by the IJG, and included the class of matrices discussed in this section. See [53, 79] for more details on the history of the JPEG file format.

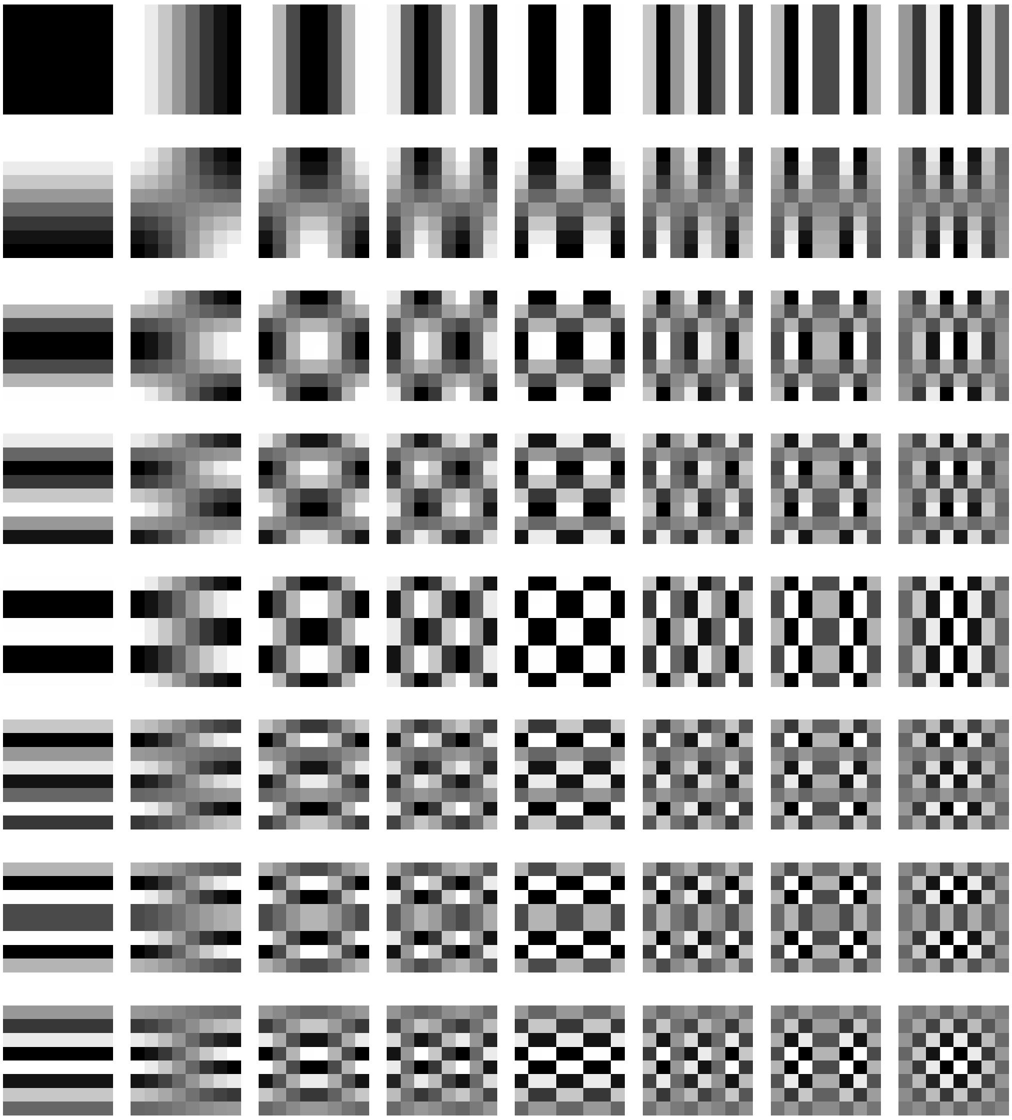

Figure 2.1: Basis matrices for 8×8 Discrete Cosine Transform (DCT)

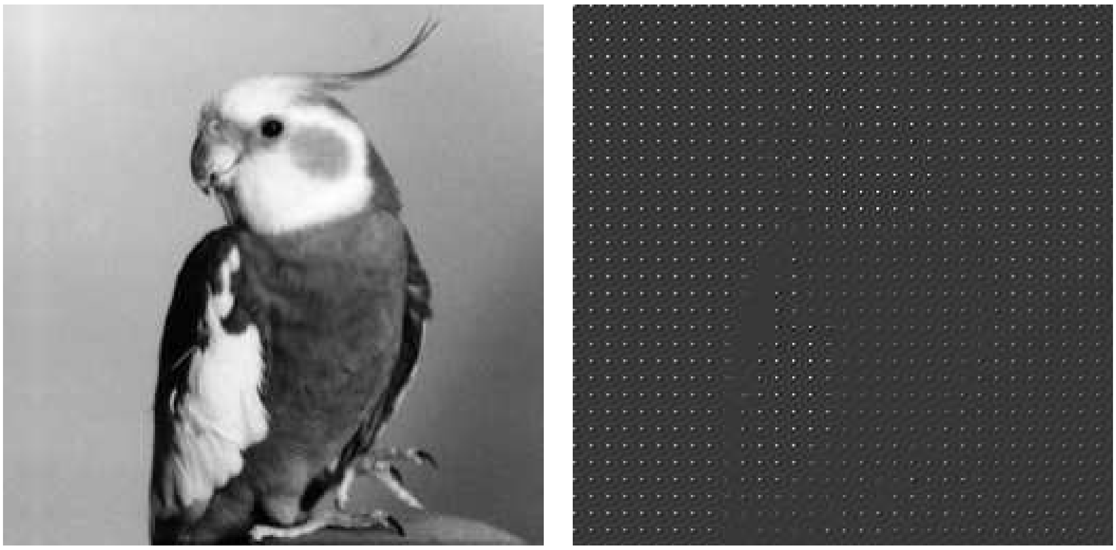

Figure 2.2: JPEG-compression sample: (left) original `bird` test image, (right) DCT-encoded data. Notice the 8×8 sub-blocks within the image, indicated by the DC coefficients.

with hyperbolic and decreasing linear pieces defined by

$$s(q) = \begin{cases} \left\lfloor \frac{5000}{q} \right\rfloor, & 1 \leq n < 50; \\ 200 - 2q, & 50 \leq q \leq 100. \end{cases} \tag{2.5}$$

The graph of $s(q)$ is shown in Figure 2.3; notice the transition between the hyperbolic and linear function-pieces at $q = 50$.

An example of the DCT-encoding and quantization stages is shown in Figure 2.5: shown is the original image data, a single 8×8 sub-block of the `bird` image (top left), with its (rounded) DCT coefficients (top right); notice how the magnitude of the DC coefficient dominates that of the AC coefficients, and how most of the bottom-right half of the coefficients are either zero or close to zero. Also shown is the quantization matrix for $q = 50$ (bottom left) and the quantized DCT coefficients (bottom right); the quantized coefficients are then combined into a single bitstream in a zigzag order within the block (see Figure 2.6), so the dominance of zeros in the lower-right half allows for a shorter encoded bitstream.

One important observation is that the quantization matrices are contained within the JPEG-encoded file, even if they are determined by the IJG quality factor; although this means that additional bits must be used in storing these matrices, it also means that these matrices are free variables within the design stage, a fact we will exploit in Section 2.3.

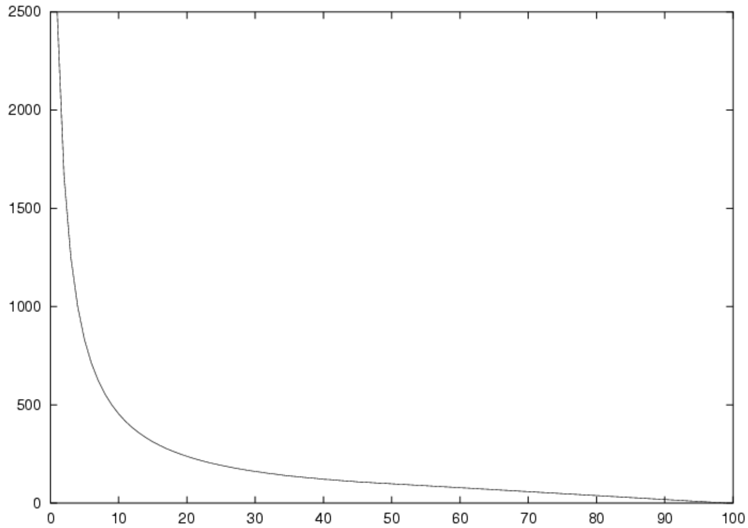

Figure 2.3: Graph of function $s(q)$ defining scaling factor for IJG quality parameter; notice the transition between the hyperbolic and linear functions at $q = 50$.

2.2 Perceptual measures in pre- and post-processing

Given the above description of the JPEG compression algorithm, we now discuss three general approaches for improving the ability of the JPEG compression algorithms to maintain an image's visual appearance, namely *post-processing* to improve perceptual quality (Section 2.2.1), *pre-processing* to optimize image fidelity (Section 2.2.2), and *re-quantization* to optimally balance image quality and bit-rate (Section 2.3.1). Each of the three methods uses knowledge of the block-coding structure inherent in the JPEG model, as well as properties of the human visual response to each of the component 2D-frequencies described by each

$$\begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

Figure 2.4: IJG standard quantization matrices: (top) luminance channel, (bottom) chrominance red/blue channels

$$\underset{(a)}{\begin{pmatrix} 56 & 65 & 75 & 81 & 74 & 65 & 63 & 57 \\ 51 & 65 & 75 & 79 & 77 & 67 & 61 & 58 \\ 53 & 64 & 71 & 78 & 79 & 68 & 61 & 63 \\ 51 & 56 & 61 & 71 & 79 & 75 & 71 & 67 \\ 49 & 50 & 58 & 67 & 71 & 72 & 71 & 68 \\ 51 & 51 & 56 & 57 & 63 & 65 & 68 & 67 \\ 61 & 54 & 57 & 54 & 57 & 63 & 65 & 67 \\ 71 & 57 & 54 & 53 & 53 & 57 & 60 & 64 \end{pmatrix}} \quad \underset{(b)}{\begin{pmatrix} 508 & -23 & -27 & -2 & 3 & 1 & 0 & 2 \\ 26 & 10 & -37 & -12 & -2 & -4 & -6 & -1 \\ -5 & 24 & 11 & -4 & 0 & 3 & 2 & 0 \\ -5 & -8 & -1 & -1 & -4 & 4 & 0 & 1 \\ 2 & -2 & 2 & 5 & 2 & 4 & -1 & -1 \\ 3 & -1 & 2 & 2 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & -1 & 2 & -1 & -1 & 0 \\ 1 & 2 & 0 & -2 & 1 & 1 & 1 & -1 \end{pmatrix}}$$

$$\underset{(c)}{\begin{pmatrix} 8 & 6 & 5 & 8 & 12 & 20 & 26 & 31 \\ 6 & 6 & 7 & 10 & 13 & 29 & 30 & 28 \\ 7 & 7 & 8 & 12 & 20 & 29 & 35 & 28 \\ 7 & 9 & 11 & 15 & 26 & 44 & 40 & 31 \\ 9 & 11 & 19 & 28 & 34 & 55 & 52 & 39 \\ 12 & 18 & 28 & 32 & 41 & 52 & 57 & 46 \\ 25 & 32 & 39 & 44 & 52 & 61 & 60 & 51 \\ 36 & 46 & 48 & 49 & 56 & 50 & 52 & 50 \end{pmatrix}} \quad \underset{(d)}{\begin{pmatrix} 64 & -4 & -5 & 0 & 0 & 0 & 0 & 0 \\ 4 & 2 & -5 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}$$

Figure 2.5: JPEG quantization example: (a) sample 8x8 block taken from `bird` test image; (b) DCT coefficients for sample block (rounded); (c) standard quantization matrix for IJG quality factor Q75; (d) quantized output.

of the DCT basis matrices; the re-quantization approach also uses information about the distribution of the DCT coefficients.

2.2.1 Post-processing: Deblocking JPEG images

As an illustration of the use of perceptual measures in post-processing JPEG-encoded images, we will consider the de-blocking algorithm of Ahumada and Horng [2, 3, 26]. The central problem this algorithm attacks is that of *blocking artifacts* present in some low bit-rate JPEG images, an example of which is shown in Figure 2.7. As the bit-rate decreases and thus fewer bits are used to described the AC coefficients, the edges of each 8×8 block becomes increasingly visible. Edge artifacts arise from the fact that each block is encoded independently of its neighbouring blocks, and thus as the DC coefficient dominates the bits used to encode the block, the transition between blocks provided by the higher-frequency AC coefficients is lost, resulting in noticeable blocking-artifacts.

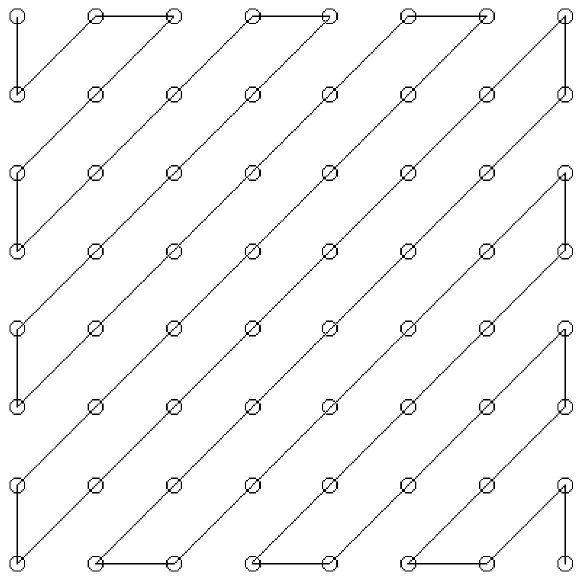

Figure 2.6: Order in which DCT coefficients are encoded: circles represent DCT coefficients within an 8×8 image sub-block.

The approach taken by Ahumada and Horng is to smooth out the differences between pixel-values along a block-edge by using the values one pixel away from the block-edge on either side as an estimate for the values of the pixels along the block edge prior to encoding. The edge-pixels are then modified to provide a smoother transition along the block boundary, resulting in less objectionable blocking artifacts. To this end, the algorithm uses a measure of *global edge-variance* defined by

$$E(X) = \sum_{(X_{i,j}, X_{i_0,j_0}) \in A} \left(X_{i,j} - X_{i_0,j_0}\right)^2, \tag{2.6}$$

where the summation is performed across all blocks, with A the set of adjacent pixel pairs across block boundaries, i.e. for an $m \times n$ block partition,

$$\begin{gathered}(X_{i,j}, X_{i_0,j_0}) \in A \Longleftrightarrow \\ \left\{ \lfloor \tfrac{i}{m} \rfloor \neq \lfloor \tfrac{i_0}{m} \rfloor or \ \lfloor \tfrac{j}{n} \rfloor \neq \lfloor \tfrac{j_0}{n} \rfloor \text{ and } |i - i_0| + |j - j_0| = 1 \right\}.\end{gathered} \tag{2.7}$$

However, modifying the edge-pixels can potentially smooth across blocks at the expense of expanding the blocking artifacts into the interior of a given block. Therefore, the algorithm checks the level of additional noise that has been added, using a *block roughness* measure defined by

$$R(X) = \sum_{\text{DCT blocks}} \sum_{u=0,v=0}^{N-1} \left(u^2 + v^2\right) \hat{I}_{u,v}^2 \tag{2.8}$$

where the summation is again performed over all blocks, and $\hat{I}_{u,v}$ is the (u, v)th DCT coefficient of the smoothed image, as defined in Section 2.1.1. By weighting each AC coefficient by its frequency, this measure will detect when the edge is expanded into the interior of the block by the presence of increased AC coefficients. If modifying the edge pixels increases the block roughness, then the algorithm removes the modifications and leaves the image untouched.

Notice that the block roughness is considered an additional constraint, i.e. the algorithm minimizes the edge variance subject to an upper bound on the block roughness. Therefore, we can approach this problem as an optimization problem with multiple objectives, namely

$$\min\left[E(X), R(X)\right]. \tag{2.9}$$

In other words, we want to find an image approximation that simultaneously minimizes edge variance and block roughness, using our given measures for each. We shall refer back to this problem in our discussions of multicriteria optimization in Chapter 4.4.

The de-blocking algorithm illustrates the relationship between perceptual measures and post-processing algorithms: given an image, the compressed image is processed to optimize a given perceptual measure. In the above discussion, the perceptual measures model a particular type of visual artifact, namely blocking artifacts, and thus is specific to a particular coder that creates this type of artifact; in general, post-processing algorithms require knowledge about the original encoding method. The primary disadvantage of post-processing algorithms is that they require additional modifications after the image-decoding stage, and therefore cannot be used if you only have access to the image encoder; for example, the decoded image, post-deblocking, will not conform to the JPEG standard, and thus will not be readable by a standard JPEG-reader. In addition, the post-processing will frequently improve the image quality at the expense of the image fidelity, as the processing is designed to remove or mask visual artifacts present in the compressed image rather than maintain the visual information present in the original image.

Figure 2.7: DCT blocking artifacts: (left) original `bird` test image, (b) compressed to quality factor $q = 21$.

2.2.2 Pre-processing: Quantization matrix design

Although the JPEG standard provides a recommended class of quantization matrices, the quantization matrix itself is transmitted with the JPEG file, and thus can be chosen freely. Therefore, improvements in the visual fidelity of JPEG-encoded images can be achieved by studying the perceptual visibility of artifacts resulting from quantizing the DCT coefficients, and designing the quantization matrix to minimize the visibility subject to a given measure. To illustrate this, we will discuss Watson's DCTune compression algorithm, and its accompanying image quality metric [82, 83].

Peterson *et. al.* first approached quantization matrix design for DCT-encoding by finding the magnitudes for each DCT coefficient at which the basis matrices, shown in Figure 2.1, are first visible over a constant background; quantization matrices were defined as twice the threshold magnitude for the corresponding DCT basis matrix, calculated using subjective tests for a fixed viewing distance in [54]. These matrices, shown in Figure 2.8, were generalized with varying viewing distances for luminance [5] and luminance-chrominance images [55]. Watson extended these matrices from artificial stimuli to natural images by including additional factors for luminance masking, contrast sensitivity, and texture masking

[82, 83]; the resulting matrices are image-specific, and calculated by solving a separate least-squares problem for each of the 64 DCT coefficients.

Unlike the post-processing algorithms, compression models that focus on pre-processing the image, such as the above models that design quantization matrices to optimize perceptual image fidelity, generally conform to the JPEG standard, and thus can be read by any JPEG-reader without requiring any additional control over the image-delivery environment. The most frequently cited disadvantage to these models is their computational complexity; however, of more interest to our present discussion is the fact that these models assume that the image being quantized has not already been JPEG-encoded, as requantizing an image without consideration of its original quantization matrix can have unanticipated visual results. An example of this is the JPEG Perceptual Dipper, which we will now discuss.

2.3 Compression by Requantization

The design of quantization matrices, as discussed in Section 2.2.2, focuses on finding a quantization matrix that will result in the minimum perceptual error for a target filesize. However, for many internet-based image-encoding algorithms, the target filesize is either unknown or variable. For example, websites that target web browsers with both high-bandwidth (cable, DSL, etc.) and low-bandwidth (PDAs, cellphones, etc.) internet connections can benefit from additional flexibility in the filesizes of images located on their sites; a high-(low-)bitrate, high-(low-)quality version can be presented to high-(low-)bandwidth connections, with the target bitrate determined in real-time. For such multi-bitrate applications, a highly-computational approach to quantization matrix design, such as that of Watson from Section 2.2.2, is overcomplex.

Another approach is to have a single "target" version of an image, with multiple levels of lower-quality compressed versions, one each for each separate target bandwidth. The problem that arises with this *requantization* approach is that unexpected artifacts can arise. We will discuss an example of such artifacts, namely the *JPEG dipper effect*, in Section 2.3.1, and will describe a sample approach to avoiding such artifacts in Section 2.3[2].

[2]This section discusses material originally presented in [7, 8]; special thanks to all of the co-authors for permission to include this work in this thesis.

$$\begin{pmatrix} 5 & 3 & 4 & 7 & 11 & 16 & 24 & 34 \\ 3 & 4 & 4 & 6 & 8 & 12 & 18 & 25 \\ 4 & 4 & 8 & 9 & 11 & 15 & 20 & 28 \\ 7 & 6 & 9 & 14 & 16 & 20 & 26 & 33 \\ 11 & 8 & 11 & 16 & 26 & 28 & 34 & 42 \\ 16 & 12 & 15 & 20 & 28 & 41 & 46 & 54 \\ 24 & 18 & 20 & 26 & 34 & 46 & 63 & 71 \\ 34 & 25 & 28 & 33 & 42 & 54 & 71 & 95 \end{pmatrix}$$

$$\begin{pmatrix} 7 & 9 & 18 & 28 & 43 & 63 & 91 & 128 \\ 9 & 8 & 17 & 23 & 33 & 48 & 68 & 94 \\ 18 & 17 & 31 & 34 & 43 & 58 & 78 & 105 \\ 28 & 23 & 34 & 55 & 63 & 77 & 98 & 126 \\ 43 & 33 & 43 & 63 & 98 & 108 & 128 & 157 \\ 63 & 48 & 58 & 77 & 108 & 154 & 174 & 204 \\ 91 & 68 & 78 & 98 & 128 & 174 & 239 & 255 \\ 128 & 94 & 105 & 126 & 157 & 204 & 255 & 255 \end{pmatrix}$$

$$\begin{pmatrix} 14 & 18 & 46 & 71 & 109 & 161 & 232 & 255 \\ 18 & 17 & 43 & 59 & 85 & 122 & 173 & 240 \\ 46 & 43 & 80 & 87 & 111 & 148 & 200 & 255 \\ 71 & 59 & 87 & 142 & 160 & 196 & 249 & 255 \\ 109 & 85 & 111 & 160 & 251 & 255 & 255 & 255 \\ 161 & 122 & 148 & 196 & 255 & 255 & 255 & 255 \\ 232 & 173 & 200 & 249 & 255 & 255 & 255 & 255 \\ 255 & 240 & 255 & 255 & 255 & 255 & 255 & 255 \end{pmatrix}$$

Figure 2.8: Base threshold quantization matrices for the three channels in YC_bC_r color space[51]

2.3.1 The JPEG Dipper Effect

Intuitively, the JPEG quality factor discussed in Section 2.1.2 is an approximate measure of the visual quality of a JPEG image; as the quality factor decreases, the level of quantization increases, and thus more information is discarded from the signal. However, contrary to intuition, this quality measure is perceptually nonmonotone; for images compressed to particular quantization levels, a particular decrease in the quality factor (and thus also in the filesize) can actually *increase* the perceptual quality of the compressed image. We refer to this nonmonotonicity as the *JPEG Perceptual Dipper*; this effect is of particular interest because it provides a rare example of the usually contradictory requirements of low filesize and high perceptual quality being satisfied simultaneously.

Some examples of the perceptual dipper are shown on the web-site described in Appendix B.2 [3], although the steps used to create these images can be performed using *any original image*. Shown are the original test images `bird`, `boat`, `lena`, `mandrill` and `peppers` (from top) quantized to $q = 75$ (left column); the image quantized to $q = 75$, then quantized again to $q = 50$ (19% smaller than the $q = 75$ image) (center column); and the image quantized to $q = 75$, then quantized again to $q = 48$ (37% smaller) (right column). The $q = 50$ image contains highly objectionable "speckling" noise, particularly noticeable around the edges of the bird, while this noise is noticeably absent from the $q = 48$ image.

This effect, although being quite surprising, is easily predictable using properties of the distribution of the DCT coefficients; we will now outline these properties, and describe an algorithm to take advantage of this behaviour for general quantization matrices.

2.3.2 Numerical Background to Dipper Effect

Distribution of DCT Coefficients

For each of the 63 AC coefficients, the histogram of unquantized coefficients for a fixed AC frequency has been observed to resemble a *Laplacian distribution* [38, 61, 69, 70]. The *probability density function* $f(x)$ of a Laplacian with zero mean and parameter $\lambda > 0$ is $f(x) = (\lambda/2)\exp(-\lambda|x|)$, and the corresponding cumulative distribution function $F(x)$ is

[3] `http://www.cecm.sfu.ca/~mmacklem/thesis/dipper.htm`

given by

$$F(x) = \int_{-\infty}^{x} f(t)\,dt = \begin{cases} \frac{1}{2}\exp(\lambda x), & \text{if } x \leq 0; \\ 1 - \frac{1}{2}\exp(-\lambda x), & \text{if } x \geq 0. \end{cases}$$

Let X be a continuous random variable with probability function f. Then the variance of X is known to be $\mathrm{Var}(X) = 2/\lambda^2$ [69, 70].

Considering the dequantized DCT coefficients for a fixed AC frequency and a fixed color component, let $q \in \mathbb{N}_+ = \{1, 2, 3, \ldots\}$ be the corresponding entry in the quantization matrix and

$$y_1 = q \cdot \eta_1, \quad y_2 = q \cdot \eta_2, \quad \ldots, \quad y_J = q \cdot \eta_J$$

be the *dequantized* DCT coefficients, so that the η_i's are the DCT entries encoded in the JPEG file. For a random variable X as previously defined, the y_i's can be viewed as a random sample taken by the discrete random variable $Y = q \cdot \mathrm{round}(X/q)$, with half-integers rounded up, *i.e.*, $\mathrm{round}\left(\frac{2n+1}{2}\right) = n + 1$.

Proposition 1 *Let p be the probability density function associated with the discrete random variable Y. Then*

$$p(nq) = \begin{cases} 1 - \exp(-\lambda q/2), & \text{if } n = 0; \\ \exp(-\lambda q|n|)\sinh(\lambda q/2), & \text{if } n \in \mathbb{Z} \setminus \{0\}. \end{cases}$$

Moreover, $Var(Y) = \big(q^2\cosh(\lambda q/2)\big)/\big(2\sinh^2(\lambda q/2)\big)$, and the expected value $E(Y)$ equals zero.

Proof: First observe that

$$\begin{aligned} p(0) &= F(q/2) - F(-q/2) \\ &= 1 - \tfrac{1}{2}\exp(-\lambda q/2) - \tfrac{1}{2}\exp(\lambda(-q/2)) \\ &= 1 - \exp(-\lambda q/2). \end{aligned}$$

Now let $n \in \mathbb{Z} \setminus \{0\}$. Suppose $n < 0$. Then

$$\begin{aligned} Y = nq \quad &\Leftrightarrow \quad \mathrm{round}(X/q) = n \\ &\Leftrightarrow \quad n - \tfrac{1}{2} \leq X/q < n + \tfrac{1}{2}. \end{aligned}$$

Therefore,

$$\begin{aligned} p(nq) &= F(qn+q/2)-F(qn-q/2) \\ &= \tfrac{1}{2}\exp(\lambda(qn+q/2))-\tfrac{1}{2}\exp(\lambda(qn-q/2)) \\ &= \exp(\lambda qn)\tfrac{1}{2}\big[\exp(\lambda q/2)-\exp(-\lambda q/2)\big] \\ &= \exp(\lambda qn)\sinh(\lambda q/2). \end{aligned}$$

Similarly, the derivation for $n > 0$ leads $p(nq) = \exp(-\lambda qn)\sinh(\lambda q/2)$, thus giving our desired formula for $p(nq)$.

The expected value, $E(Y) = 0$, follows from the fact that p is an even function. To calculate the variance, we use the following equivalent form:

$$\begin{aligned} \mathrm{Var}(Y) &= E\big((Y-E(Y))^2\big) \\ &= E(Y^2) \\ &= \sum_{n\in\mathbb{Z}}(nq)^2p(nq) \\ &= \sum_{n=1}^{+\infty}2n^2q^2\sinh(\lambda q/2)\exp(-\lambda qn) \end{aligned}$$

Now recall the identity $\sum_{n=1}^{+\infty} n^2\alpha^n = \alpha(1+\alpha)/(1-\alpha)^3$ for $0 \le \alpha < 1$ (which follows readily from differentiating the geometric series twice). Using this, the desired identity follows by setting $\alpha = \exp(-\lambda q)$, and after some algebraic manipulations. □

The Laplacian distribution depends on the parameter λ, and thus we need a method of estimating this parameter from the collected AC coefficients $\{y_i\}$ for a given frequency. Previous papers have used moment-matching techniques [69, 70], but we propose a technique involving the *Maximum Likelihood estimator*, using the following result of Price and Rabbani [58].

Proposition 2 (Method of Maximum Likelihood)
For samples $y_i, i = 1, \ldots, J$, let $\alpha = |\{j : y_j = 0\}|$, $\beta = J - \alpha$, and $\rho = 2\sum_j |\eta_j|$, with $|\cdot|$ denoting the number of elements in a finite set. Then

$$\lambda_{\mathrm{ML}} = \frac{-2}{q}\ln\left(\frac{-\alpha+\sqrt{\alpha^2-4\alpha\beta-4\beta^2+4\rho^2+4\alpha\rho}}{2(J+\rho)}\right)$$

is the Maximum Likelihood estimator *for λ. (If $\beta = 0$, then λ_{ML} becomes $+\infty$.)*

Proof: The corresponding likelihood function is

$$\ell(\lambda) = \prod_{j=1}^{J} \text{prob}(Y = \eta_j q) = \prod_{j=1}^{J} p(\eta_j q).$$

By Proposition 1, $\ell(\lambda)$ is equal to

$$\prod_{j:\eta_j=0} \big(1 - \exp(-\lambda q/2)\big) \prod_{j:\eta_j\neq 0} \exp(-\lambda q|\eta_j|)\sinh(\lambda q/2).$$

Let $L = \ln \circ \ell$ be the log-likelihood function. Then $L(\lambda)$ is

$$\alpha \ln\big(1 - \exp(-\lambda q/2)\big) + \beta \ln\big(\sinh(\lambda q/2)\big) - \rho\lambda q/2.$$

Set $\xi = \exp(-\lambda q/2)$. Now compute the derivative of L with respect to ξ, and set it equal to zero. The resulting equation is equivalent to

$$\frac{\alpha\xi}{1-\xi} + \frac{\beta(1+\xi^2)}{1-\xi^2} = \rho,$$

which leads to a quadratic in ξ. The solution we want is

$$\xi = \frac{1}{2}\frac{-\alpha + \sqrt{\alpha^2 - 4\alpha\beta + 4\alpha\rho - 4\beta^2 + 4\rho^2}}{\alpha+\beta+\rho}.$$

After switching back to original notation, we obtain the formula for λ_{ML}. □

Direct vs Indirect Dequantization

The errors that we observed in Section 2.3.1 as a result of requantizing an already quantized image, using a quantizer value distinct from the previous value, can be described using the following result, which relates the errors due to requantization to the quantizer values.

Proposition 3 *For given positive integers q_0, q_1 with $q_0 < q_1$, let $l = \text{lcm}(q_0, q_1)$. Let x be an integer satisfying $1 \leq x \leq l$, and let $x_1 = q_1 \cdot \text{round}(x/q_1)$, $y_0 = q_0 \cdot \text{round}(x/q_0)$, and $y_1 = q_1 \cdot \text{round}(y_0/q_1)$. Then:*

(i) the only differences possible are $x_1 - y_1 \in \{-q_1, 0, q_1\}$;

(ii) periodicity: if $\tilde{x} = x + l$, then $\tilde{x_1} = x_1 + l$ and $\tilde{y_1} = y_1 + l$;

(iii) odd multiples preserve values: if $m \in \mathbb{N}\setminus\{0\}$ is odd and $q_1 = mq_0$, then $x_1 = y_1$;

(iv) *even multiples over-estimate: let $m \in \mathbb{N} \setminus \{0\}$ be even, $q_1 = mq_0$, and $1 \leq x \leq l$. Then*

$$y_1 = \begin{cases} x_1 + q_1, & \frac{q_1}{2} - \left\lfloor \frac{q_0}{2} \right\rfloor \leq x \leq \frac{q_1}{2} - 1; \\ x_1, & \textit{otherwise}. \end{cases} \tag{2.10}$$

(v) *even-plus-one multiples under-estimate: let $m \in \mathbb{N} \setminus \{0\}$ be even, and $q_1 = 1 + mq_0$. Then*

$$y_1 = \begin{cases} x_1, & x < \frac{q_1+1}{2}; \\ x_1 - q_1, & \frac{q_1+1}{2} \leq x \leq \frac{q_1-3}{2} + \left\lceil \frac{q_0}{2} \right\rceil, \end{cases} \tag{2.11}$$

while the value of y_1 is uncertain for $x > \frac{q_1-3}{2} + \left\lceil \frac{q_0}{2} \right\rceil$.

Proof:

(i) : Clearly, $|x - x_1| \leq q_1/2$, $|x - y_0| \leq q_0/2$, and $|y_0 - y_1| \leq q_1/2$. Hence

$$|x_1 - y_1| \leq |x_1 - x| + |x - y_0| + |y_0 - y_1| \leq q_1/2 + q_0/2 + q_1/2 < 3q_1/2.$$

On the other hand, x_1 and y_1 both belong to $q_1 \cdot \mathbb{Z}$, and so must $|x_1 - y_1|$. Altogether, $|x_1 - y_1| \in \{0, q_1\}$ and item (i) follows.

(ii) : Since $\widetilde{x} = x + l$, we have $\widetilde{x}/q_1 = x/q_1 + l/q_1$. Now $l/q_1 \in \mathbb{Z}$, thus $\text{round}(\widetilde{x}/q_1) = \text{round}(x/q_1) + l/q_1$ and hence $\widetilde{x}_1 = q_1 \cdot \text{round}(x/q_1) + l = x_1 + l$. The proof of $\widetilde{y}_1 = y_1 + l$ is analogous.

(iii) : Define $r_0 = \text{round}\left(\frac{x}{q_0}\right)$. Then $y_0 = q_0 r_0$ and

$$r_0 - \frac{1}{2} \leq \frac{x}{q_0} < r_0 + \frac{1}{2}. \tag{2.12}$$

We can write $r_0 = mr + b$ with $|b| \leq \left\lceil \frac{m-1}{2} \right\rceil$; since m is odd, $\frac{m-1}{2}$ is an integer, and thus $|b| \leq \frac{m-1}{2}$. Dividing both sides by m gives

$$\frac{|b|}{m} \leq \frac{1}{2} - \frac{1}{2m} \tag{2.13}$$

or equivalently

$$\frac{1}{2m} - \frac{1}{2} \leq \frac{b}{m} \leq \frac{1}{2} - \frac{1}{2m}. \tag{2.14}$$

Now divide (2.12) by m to give

$$\frac{r_0}{m} - \frac{1}{2m} \leq \frac{x}{mq_0} < \frac{r_0}{m} + \frac{1}{2m}. \tag{2.15}$$

Substituting in $r_0 = mr + b$ gives

$$r + \frac{b}{m} - \frac{1}{2m} \leq \frac{x}{q_1} < r + \frac{b}{m} + \frac{1}{2m}. \tag{2.16}$$

Furthermore, substituting the bounds on $\frac{b}{m}$ from (2.14) into (2.15) gives

$$r - \frac{1}{2} \leq \frac{x}{q_1} < r + \frac{1}{2}, \tag{2.17}$$

therefore giving $r = \operatorname{round}\left(\frac{x}{q_1}\right)$ and thus $x_1 = q_1 r$.

Now if we consider $\frac{y_0}{q_1}$, we get

$$\frac{y_0}{q_1} = \frac{q_0 r_0}{q_0 m} = \frac{r_0}{m} = r + \frac{b}{m}. \tag{2.18}$$

Combining (2.18) with the fact that from (2.13) we have $\frac{|b|}{m} < \frac{1}{2}$ gives $\operatorname{round}\left(\frac{y_0}{q_1}\right) = r$. Therefore,

$$y_1 = q_1 \operatorname{round}\left(\frac{y_0}{q_1}\right) = q_1 r = x_1, \tag{2.19}$$

and thus $y_1 = x_1$ as claimed.

(iv) : We split the proof into three cases, based on the conditions in the piecewise behaviour of y_1.

(I) $\frac{q_1}{2} - \left\lfloor \frac{q_0}{2} \right\rfloor \leq x \leq \frac{q_1}{2} - 1$: Since $x \leq \frac{q_1}{2} - 1$, then $\operatorname{round}\left(\frac{x}{q_1}\right) = 0$. Now consider when q_0 is even, and thus $\left\lfloor \frac{q_0}{2} \right\rfloor = \frac{q_0}{2}$; then

$$\begin{aligned} x \geq \frac{q_1}{2} - \left\lfloor \frac{q_0}{2} \right\rfloor \quad &\Rightarrow \quad x \geq \frac{q_1 - q_0}{2} \\ &\Rightarrow \quad \frac{x}{q_0} \geq \frac{q_1}{2q_0} - \frac{1}{2} = \frac{m-1}{2} \\ &\Rightarrow \quad \frac{x}{q_0} \geq \frac{m}{2}. \end{aligned}$$

If q_0 is odd, then $\left\lfloor \frac{q_0}{2} \right\rfloor = \frac{q_0 - 1}{2}$, and thus

$$\begin{aligned} x \geq \frac{q_1 - q_0 + 1}{2} \quad &\Rightarrow \quad \frac{x}{q_1} \geq \frac{m}{2} - \frac{1}{2} + \frac{1}{2q_0} \\ &\Rightarrow \quad \frac{x}{q_0} \geq \frac{m}{2} - \frac{1}{2}\left(1 - \frac{1}{q_0}\right) \\ &\Rightarrow \quad \frac{x}{q_0} \geq \frac{m}{2}. \end{aligned}$$

Therefore, round $\left(\frac{x}{q_0}\right) \geq \frac{m}{2}$. Finally, notice that

$$\begin{aligned} x \leq \frac{q_1}{2} - 1 \quad &\Rightarrow \quad \frac{x}{q_0} \leq \frac{q_1}{2q_0} - \frac{1}{q_0} \\ &\Rightarrow \quad \frac{x}{q_0} \leq \frac{m}{2} - \frac{1}{q_0} \\ &\Rightarrow \quad \text{round}\left(\frac{x}{q_0}\right) \leq \text{round}\left(\frac{m}{2}\right) = \frac{m}{2}, \end{aligned}$$

and thus round $\left(\frac{x}{q_0}\right) = \frac{m}{2}$.

Therefore, since half-integers are rounded up,

$$\text{round}\left(\frac{y_0}{q_1}\right) = \text{round}\left(\frac{mq_0/2}{q_1}\right) = \text{round}\left(\frac{q_1/2}{q_1}\right) = \text{round}\left(\frac{1}{2}\right) = 1$$

and thus $y_1 = q_1 = x_1 + q_1$.

(II) $x < \frac{q_1}{2} - \left\lfloor \frac{q_0}{2} \right\rfloor$: Clearly, round $\left(\frac{x}{q_1}\right) = 0$. The remaining argument follows the previous case: if q_0 is even, then

$$\begin{aligned} \left\lfloor \frac{q_0}{2} \right\rfloor = \frac{q_0}{2} \quad &\Rightarrow \quad x < \frac{q_1 - q_0}{2} \\ &\Rightarrow \quad \frac{x}{q_0} < \frac{q_1}{2q_0} - \frac{1}{2} = \frac{m}{2} - \frac{1}{2} \\ &\Rightarrow \quad \text{round}\left(\frac{x}{q_0}\right) < \frac{m}{2}, \end{aligned}$$

while if q_0 is odd then

$$\begin{aligned} \left\lfloor \frac{q_0}{2} \right\rfloor = \frac{q_0 - 1}{2} \quad &\Rightarrow \quad x < \frac{q_1 - q_0 + 1}{2} \\ &\Rightarrow \quad \frac{x}{q_0} < \frac{q_1}{2q_0} - \frac{1}{2} + \frac{1}{2q_0} = \frac{m}{2} - \frac{1}{2}\left(1 - \frac{1}{q_0}\right) \\ &\Rightarrow \quad \text{round}\left(\frac{x}{q_0}\right) < \frac{m}{2}. \end{aligned}$$

Therefore, $y_0 = q_0 \, \text{round}\left(\frac{x}{q_0}\right) < \frac{q_1}{2}$, and therefore $y_1 = x_1 = 0$.

(III) $x > \frac{q_1}{2} - 1$: The proof of this case is entirely analogous to that of Case II, except with both x_1 and y_1 rounded up to q_1.

(v) : The proof of (v) is analogous to that of (iv), and thus is omitted. $\square$

Remark: Notice that the oddness of m is essential for the representation of $r_0 = mr + b$ satisfying the bound $|b| \leq \frac{m-1}{2}$, since the integrality of b forces this bound to be non-tight for even m. For example, if $m = 6$ and $r_0 = 15$, then the smallest value of $|b|$ of this form is 3, while $\frac{m-1}{2} = \frac{5}{2}$; therefore, any representation of the form $r_0 = mr + b$ will not satisfy the required bound. In addition, notice that in Case iv-I, the fact that half-integers are rounded up is crucial to the overestimate of y_1, and that rounding down half-integers would result in preserving the values; we will use this fact in Section 2.3.3. □

Using the notation from Proposition 3, we can interpret these results in the context of JPEG requantization by considering the error resulting from requantizing an integer value; specifically, for $n \in \mathbb{N}^*$, let

$$e_n = q_1 \cdot \text{round}\left(q_0 \cdot \text{round}(n/q_0)/q_1\right) - n.$$

Therefore, the weighted absolute error made by dequantizing the given Laplacian in two steps is

$$e = 2 \sum_{n \in \mathbb{N}^*} p(n)|e_n|,$$

where the value n occurs with frequency $p(n)$. Using the approach of Chan in [14, 15], we break this error up into two components: $e = e_+ + e_-$, where e_+ and e_- are both nonnegative and measure *increase* and *decrease* in amplitude, respectively. The components are given by

$$e_+ = 2 \sum_{n \in \mathbb{N}^* : e_n > 0} p(n)|e_n| \quad \text{and} \quad e_- = 2 \sum_{n \in \mathbb{N}^* : e_n < 0} p(n)|e_n|,$$

where e, e_+, e_- all depend implicitly on λ, q_0, q_1. Using these definitions, within the context of JPEG images, Proposition 3 states that if the component e_+ (resp. e_-) dominates, then the amplitude increases (resp. decreases) and we thus expect the image dequantized in two steps to be *grainier/sharper* (resp. *blockier/smoother*) than the original.

Assuming a fixed λ and q_0 and treating the errors as functions of q_1, one finds that they are not monotone, and thus finding an ideal value for requantizing amounts to finding $q_1 \geq q_0$ such that q_1 is a local minimum of these error measures. In order to find these local minima, we used the Method of Maximum Likelihood from Proposition 2 to calculate the Laplacian parameter for each DCT AC coefficient over eight standard test images: `bird`, `boat`, `camera`, `frog`, `goldhill`, `lena`, `mandrill`, and `peppers` [80]. The statistics are summarized in Table 2.1.

Table 2.1: Statistics of the ML estimates of λ.

minimum	0.0129
1^{st} quartile	0.0710
median	0.1240
3^{rd} quartile	0.2115
maximum	1.3794
mean	0.1788
standard deviation	0.1910

For numerical estimations of the Laplacian distribution, we need a truncation limit; this is provided by the fact that the JPEG image format implicitly constrains the range of values a DCT coefficient can take.

Proposition 4 *Suppose X is an 8-by-8 integer matrix with values in $\{0, 1, 2, \ldots, 255\}$, and let Y be the two-dimensional DCT of X. Then Y is entry-wise bounded: from below by*

$$-\begin{pmatrix} 0 & 925 & 943 & 925 & 1020 & 925 & 943 & 925 \\ 925 & 838 & 854 & 838 & 925 & 838 & 854 & 838 \\ 943 & 854 & 871 & 854 & 943 & 854 & 871 & 854 \\ 925 & 838 & 854 & 838 & 925 & 838 & 854 & 838 \\ 1020 & 925 & 943 & 925 & 1020 & 925 & 943 & 925 \\ 925 & 838 & 854 & 838 & 925 & 838 & 854 & 838 \\ 943 & 854 & 871 & 854 & 943 & 854 & 871 & 854 \\ 925 & 838 & 854 & 838 & 925 & 838 & 854 & 838 \end{pmatrix}, \tag{2.20}$$

and from above by

$$\begin{pmatrix} 2040 & 925 & 943 & 925 & 1020 & 925 & 943 & 925 \\ 925 & 838 & 854 & 838 & 925 & 838 & 854 & 838 \\ 943 & 854 & 871 & 854 & 943 & 854 & 871 & 854 \\ 925 & 838 & 854 & 838 & 925 & 838 & 854 & 838 \\ 1020 & 925 & 943 & 925 & 1020 & 925 & 943 & 925 \\ 925 & 838 & 854 & 838 & 925 & 838 & 854 & 838 \\ 943 & 854 & 871 & 854 & 943 & 854 & 871 & 854 \\ 925 & 838 & 854 & 838 & 925 & 838 & 854 & 838 \end{pmatrix}. \tag{2.21}$$

In particular, the magnitude of an AC DCT coefficient is never larger than 1020.

Proof: Our proof is computational: for a fixed frequency, consider the DCT formula for Y in terms of X, and choose X with values in $\{0, 255\}$ so that the coefficient corresponding to the given frequency is as large (or small) as possible. The `Octave` code to perform this calculation is shown in Appendix A.1.1 [48], and can be used to create the desired matrices. □

Of particular note, the bounds in (2.20) and (2.21) are very conservative, as they are achieved only for artificial black-and-white images, and never simultaneously. Using these numerical results, we considered values of $\lambda \in [0.01, 2]$, and truncated the resulting Laplacian distribution for the corresponding Laplacian distribution with frequency $p(n)$ (for p as defined in Proposition 1) at 1020. We then computed the errors e_+ and e_- numerically, for all q_0, q_1 in $\mathbb{N}^*$ with $1 \leq q_0 \leq q_1 \leq 255$, corresponding to the JPEG baseline standard. These errors were collected in an upper triangular 255-by-255 matrix, where the (q_0, q_1) entry is the error corresponding to the fixed λ. The resulting matrices of errors e_+, e_-, e are shown in Figures 2.9, 2.10, 2.12, respectively, for $\lambda = 0.0710$ (first quartile), chosen because the important features are more pronounced and hence easier to spot[4]. Notice that in each Figure, lighter colors correspond to large entries, with the matrices normalized with the maximum entry being white; hence the same colors have different magnitudes in each of the matrices.

From these images, we can observe that the e,e_+, and e_- error images exhibit ridges when q_1 is approximately an even multiple of q_0 (see Figures 2.12, 2.9 and 2.10, respectively), with the ridges disappearing more quickly as λ increases. This behaviour is more easily observed by taking a cross-section of the three error matrices by fixing a value for q_0; for example, see Figure 11, with fixed $\lambda = 0.1788$ (mean) and $q_0 = 5$.

[4]The corresponding error matrices can have quite different behavior when created using a different distribution, thus underlining the crucial importance of the Laplacian distribution to this phenomenon.

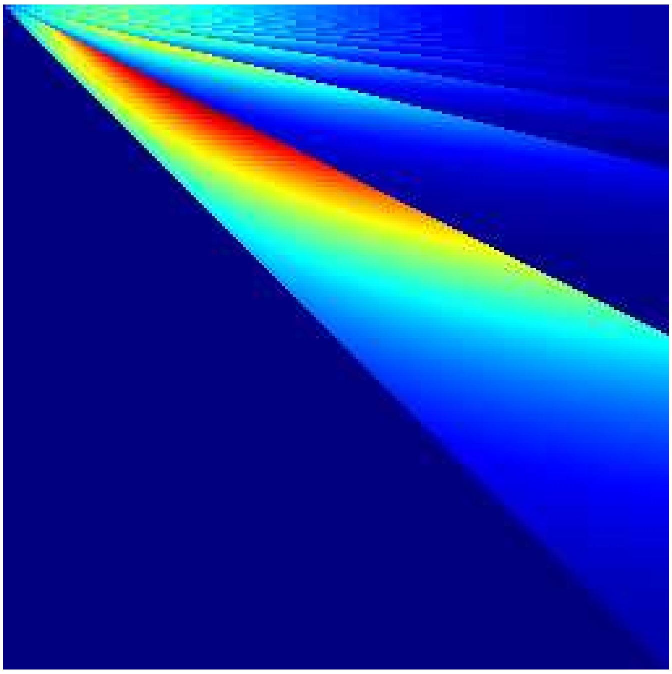

Figure 2.9: Sqrt-plot of the amplitude-intensifying error e_+ for various values of q_0 (horizontal axis) and q_1 (vertical axis), calculated using $\lambda = 0.0710$ (first quartile). Lighter entries denote higher error values, normalized so that white is the maximum matrix entry (11.8).

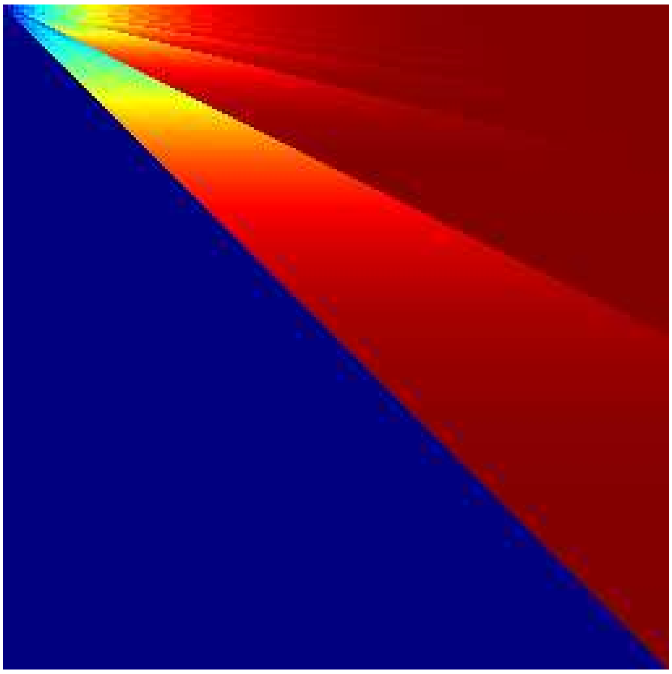

Figure 2.10: Sqrt-plot of the amplitude-decreasing error e_{-} for various values of q_0 (horizontal axis) and q_1 (vertical axis), calculated using $\lambda = 0.0710$ (first quartile). Lighter entries denote higher error values, normalized so that white is the maximum matrix entry (14.1).

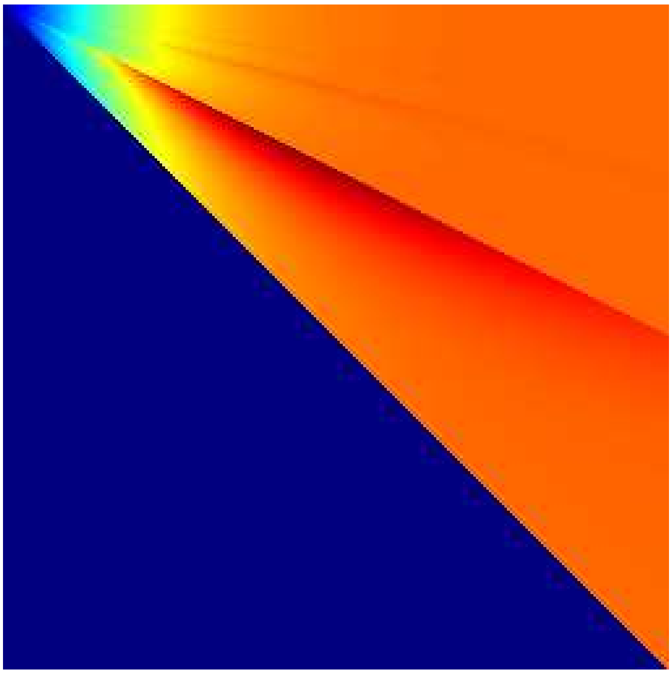

Figure 2.11: Sqrt-plot of the overall error $e = e_+ + e_-$ for various values of q_0 (horizontal axis) and q_1 (vertical axis), calculated using $\lambda = 0.0710$ (first quartile). Lighter entries denote higher error values, normalized so that white is the maximum matrix entry (18.2).

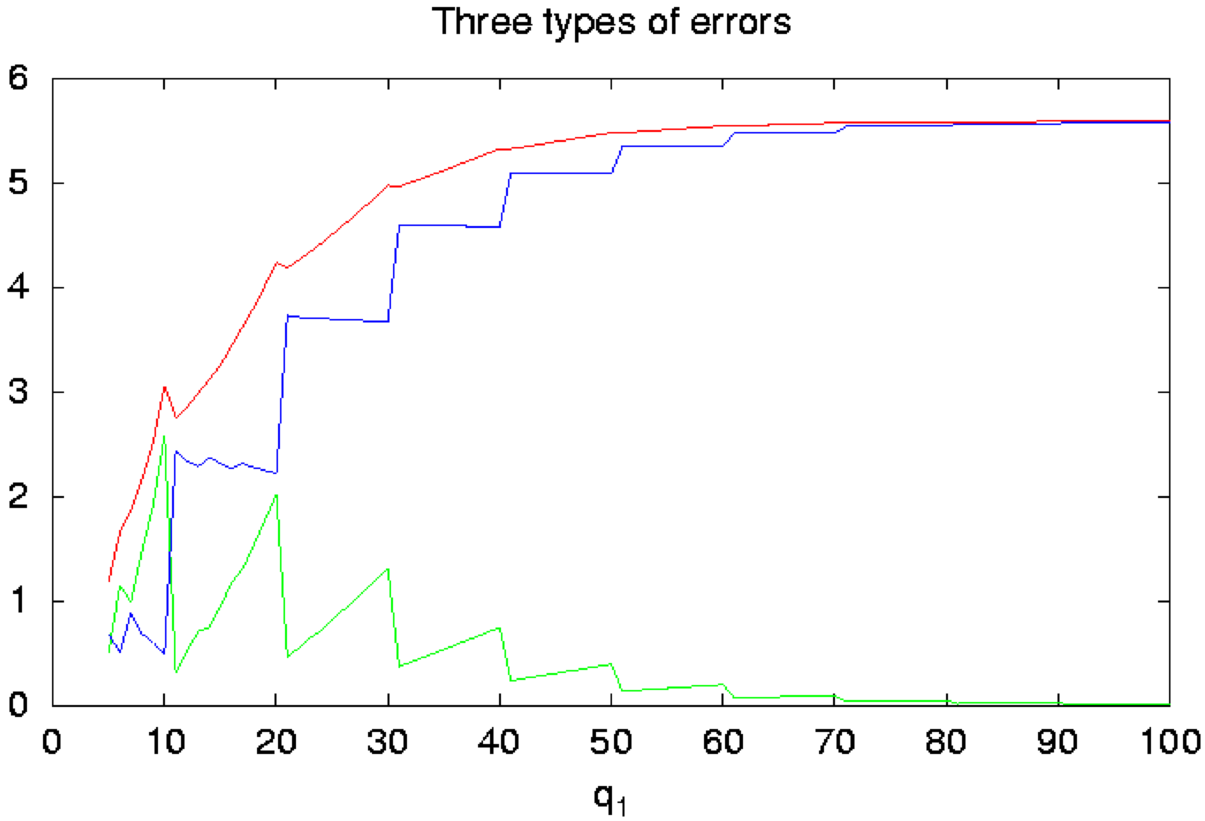

Figure 2.12: Error plots for fixed λ and q_0 and varying q_1

These numerical results lead us to postulate the following: for fixed λ and q_0, consider the errors e, e_+, e_- as functions of q_1. Then:

(i) the total error e is monotone increasing, with the exception of small nonmonotonicities[5] around points of the form $q_1 = mq_0 + 1$, where $m \in \mathbb{N}^*$ is even;

(ii) the (amplitude-intensifying) error e_+ is overall decreasing to 0, but is increasing on intervals $[mq_0 + 1, (m+2)q_0]$ with $m \in \mathbb{N}^*$ even, and drops sharply between mq_0 and $mq_0 + 1$;

(iii) the (amplitude-reducing) error e_- is overall increasing and approaching e, but is nearly constant on intervals $[mq_0+1, (m+2)q_0]$ with $m \in \mathbb{N}^*$ is even, and rises sharply between mq_0 and $mq_0 + 1$.

Interpreting these observations in the context of requantizing JPEG images, we can see that requantizing with $q_1 = mq_0$ (or less) is amplitude intensifying and thus results in sharper/grainier images, while requantizing with $q_1 = mq_0 + 1$ (or more) is amplitude reducing and hence yields smoother/blockier images.

We can apply these hypotheses to our earlier example illustrated using the five standard test images, in which the speckling noise occurred when the image quantized at quality factor 75 was requantized at quality factor 50, but was absent when requantized at quality factor 48. If we denote QN as the standard IJG quantization matrix for quality factor N, then the matrix of entrywise quotients (using `MATLAB` notation) Q50./Q75 is

$$\begin{pmatrix} 2 & 2^- & 2 & 2 & 2 & 2 & 2^- & 2^- \\ 2 & 2 & 2 & 2^- & 2 & 2 & 2 & 2^- \\ 2 & 2^- & 2 & 2 & 2 & 2^- & 2^- & 2 \\ 2 & 2^- & 2 & 2^- & 2^- & 2^- & 2 & 2 \\ 2 & 2 & 2^- & 2 & 2 & 2^- & 2^- & 2^- \\ 2 & 2^- & 2^- & 2 & 2^- & 2 & 2^- & 2 \\ 2^- & 2 & 2 & 2^- & 2^- & 2^- & 2 & 2^- \\ 2 & 2 & 2^- & 2 & 2 & 2 & 2^- & 2^- \end{pmatrix},$$

[5]These nonmonotonicities give the "dipper effect" its name.

whereas Q48./Q75 is

$$\begin{pmatrix}
2^+ & 2^- & 2 & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ \\
2 & 2 & 2^+ & 2 & 2^+ & 2^+ & 2^+ & 2^+ \\
2^+ & 2 & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ \\
2^+ & 2 & 2^+ & 2 & 2^+ & 2^+ & 2^+ & 2^+ \\
2^+ & 2^+ & 2 & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ \\
2^+ & 2 & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ \\
2^+ & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ \\
2^+ & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ & 2^+ & 2^+
\end{pmatrix},$$

where 2^+ and 2^- denote real numbers greater than and less than 2, respectively. Referring to our earlier observations (with $m = 2$), for Q50 we can see that the large number of entrywise quotients less than 2 will result in the error being dominated by amplitude-intensifying errors, and thus the resulting image will appear grainy, while for Q48 the large number of entrywise quotients greater than 2 will result in the error being dominated by amplitude-reducing errors, and thus the resulting image will appear smoother. This is confirmed by our observations about our earlier example in Section 2.3.1 using the five test images.

2.3.3 Application: A Heuristic Algorithm for Requantization

Similar to our discussion of the strategies of post-processing and quantization matrix design, we now present an example of how to use our observations of the perceptual quality of an image to design an image compression algorithm. In this case, our algorithm will requantize a JPEG image while avoiding the "grainy" amplitude-intensifying artifacts and seeking the "smooth" amplitude-reducing artifacts instead.

We have previously observed that when the new quantizer q_1 is (close to) an integer multiple of the old quantizer q_0, say $q_1 = mq_0$ for some $m \in \mathbb{N}^*$, then the resulting image is unaffected for odd m, while the resulting image is extremely sensitive for even m, with strong amplitude increase (resp. decrease) for $q_1 = mq_0$ (resp. $q_1 = mq_0 + 1$). Let Q_{bound} be a quantization matrix that results in a JPEG image with approximately our target filesize; we want to construct a quantization matrix that will give approximately the same image quality as Q_{bound}, while avoiding the aforementioned "grainy" artifacts. For a fixed frequency, denote the corresponding quantizers by q_0, q_{bound}, and q_1, respectively.

Algorithm 2 (Heuristic algorithm for requantization) *Let* $k = \lfloor q_{\text{bound}}/q_0 \rfloor$.

1. *For each entry* q_0 *in the original quantization matrix* Q_0, *define the corresponding entry* q_1 *in the new quantization matrix* Q_1 *by*

$$q_1 = \begin{cases} q_0, & \text{if } k = 0; \\ kq_0, & \text{if } k > 0. \end{cases}$$

2. *Requantize the given image with respect to* Q_1 *but round all positive half-integers* $(\frac{1}{2}, \frac{3}{2}, \ldots)$ *down (towards zero).*

2.3.4 Experimental results

Methods

To validate our heuristic algorithm, we performed subjective tests, comparing the implemented heuristic algorithm of Section 2.3.3 to a blind requantization[6]. We allowed for some flexibility in viewing conditions, where the subjects were given a small range of viewing distances, varying from 3 to 4 times the image height.

The seven test subjects were selected from a larger pool of subjects, according to the consistency of their responses. All subjects had normal or corrected-to-normal vision, and received simple training which included a description of the types of errors that would be occurring within the test images. The test subjects were also given training with regards to the rating scale used for the test, which was the *Mean Opinion Score (MOS) grading scale* from [6], shown in Table 2.2; note that the MOS grading scale also allows half-integer scores; the range of possible scores is $\{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$.

Each subject viewed 73 images, and 63 images were selected for the final results, based on their location within the test, so as to minimize artifacts due to fatigue or to adjusting to the test environment.

Test Images

Since our interest in the heuristic is for Internet applications, the 63 JPEG images were selected from a database containing more than 1 million images assembled from the World Wide Web. The database was created by downloading images using an application that

[6]Special thanks to Packeteer for permission to include the following experimental methodology and results.

Table 2.2: Mean Opinion Score (MOS) guidelines

Rating	Description
5.0	Imperceptible
4.0	Perceptible, but not annoying
3.0	Slightly annoying
2.0	Annoying
1.0	Very annoying

emulated a web-user: a starting web-site was chosen, all images on the starting page were downloaded, a link from this page was randomly selected and followed, all images were downloaded from the new page; this process was continued until a sufficient number of images had been compiled. For all JPEG images contained in the resulting database, the quantization matrices were collected, and lists of frequently-used quantization matrices were compiled. Using the database as a "random sample" of images on the internet, the images used in the experiment were chosen to maintain the percentage that each quantization matrix occurred within the database.

One potential problem in the design of this experiment is that by using more complex images, we are unable to isolate the specific property of the stimulus to which the subjects are responding, whereas using more artificial test stimuli, such as the scaled DCT basis matrices used in the experiments of Peterson et.al. [54, 55], would enable such observations. We chose to use natural images in our experiment for two reasons: first, our heuristic was designed to compress natural images, and thus our primary concern in the experiment was to test its performance in this setting; and second, our claim, for which we have provided extensive numerical justification, is that the observed "speckling" behavior is a property of the distribution of the DCT coefficients, and as such is independent of the original image content.

The images in the selected test-set were recompressed using blind requantization to Q50, and also with the heuristic algorithm described in 2.3.3, with $\mathrm{Q}_{\mathrm{bound}} = \mathrm{Q}45$. The quality level Q50 was selected to give additional compression compared to the original versions of each image, as well as to select a quality level prevalent within the image database. We chose $\mathrm{Q}_{\mathrm{bound}} = \mathrm{Q}45$ in order to approximately match the corresponding file sizes with those obtained from Q50. This also ensured that we captured some of the "dips" described in

Section 2.3.3. Each compressed image was presented to the right of the original, on a 19-inch monitor set to 800-by-600 resolution; this low resolution was chosen to accentuate the differences between the images, and thus to encourage use of the entire MOS scale.

Results

We now discuss the result of the subjective test just described. Table 2.3 shows the average scores over all 63 test images for each subject; from this table, we can see that the subjective ratings for the heuristic algorithm *consistently increase across every subject relative to blind requantization.*

Another method to consider the subjective responses is by looking at the pairwise ratings of the heuristic algorithm and blind requantization, to see the degree to which the subjective ratings for the heuristic algorithm improve for each image. This data is shown in Table 2.4, specifically the frequency of the ratings for the blanket requantization to Q50 relative to the ratings for the heuristic method described in Section 2.3.3 with $Q_{\text{bound}} = Q45$. From this table, we can see that while *the heuristic algorithm at least maintains, and frequently increases, the perceived image quality, it does so while also avoiding the outliers observed in the table.* For example, none of the subjects gave any ratings of 1.0-1.5 to any images produced by the heuristic algorithm, while all of the ratings of 1.0-1.5 for images created using blind requantization improved when the heuristic algorithm was used, to as high a rating as 4.5.

Table 2.3: MOS average test results

Subject	requantize to Q50	$Q_{\text{bound}} = Q45$
1	4.065	4.324
2	4.235	4.470
3	4.130	4.370
4	3.923	4.115
5	3.744	4.256
6	3.667	3.795
7	3.289	3.748

Table 2.4: Number of occurrences of each rating: vertical axis is rating for Q50, horizontal axis is rating for $Q_{\mathrm{bound}} = Q45$

	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.5	1.0
5.0	43	28	5	0	0	0	0	0	0
4.5	26	45	15	4	0	0	0	0	0
4.0	16	38	34	10	8	0	0	0	0
3.5	7	6	23	13	9	2	0	0	0
3.0	4	4	19	7	15	10	2	0	0
2.5	1	1	10	2	5	2	1	0	0
2.0	1	1	5	0	2	0	6	0	0
1.5	0	0	0	0	0	0	1	0	0
1.0	0	3	3	2	1	0	1	0	0

2.4 Conclusion

In this chapter, we have described the JPEG still image compression standard, focussing on the Discrete Cosine Transform and the recommended family of quantization matrices. We have also described methods of pre- and post-processing that try to improve the perceptual quality/fidelity of the compressed image. Finally, we demonstrated the JPEG perceptual dipper, a phenomenon that neither the pre- nor post-processing methods we discussed would avoid, and derived a heuristic to requantize a JPEG-encoded image as much as possible while avoid the observed artifacts. This heuristic conforms to the JPEG standard, and thus will create images readable by any JPEG-compliant image reader, unlike the post-processing algorithms discussed in Section 2.2.1. Additionally, our heuristic algorithm is computationally inexpensive, requiring only $64(\times 3)$ divisions to design the new quantization matrices given the original quantization matrix and a target matrix, unlike the more expensive calculations required to solve $64(\times 3)$ least-squares problems in the pre-processing algorithms described in Section 2.2.2. Furthermore, this heuristic allows us to easily requantize an image to varying quality levels, simply by changing the target quantization matrix; this enables us to create images dynamically for both low- and high-bandwidth connections, an option not available to the pre- and post-processing algorithms.

Of note, the artifacts we observed resulted from quantizing an already quantized Laplacian random variable, and thus depend only on the Laplacian distribution of the Discrete

Cosine Transform; therefore, the Perceptual Dipper will occur for any DCT-based compression method, which includes many flavors of the MPEG moving image compression standards [44, 67]. For further research, it would be interesting to see if gains can be made from using our heuristic for compressing frames of motion images within the MPEG standards, or if the frame changes already mask the "speckling" effect observed in still images.

Chapter 3

General Perceptual Fidelity Measures

In Chapter 2, we discussed a specific image compression algorithm, namely the Discrete Cosine Transform in the context of the JPEG still image compression standard. We also discussed how using properties of the artifacts produced by this compression algorithm can lead to increased compression, by removing information that does not contribute to the visual appearance of the original image or that causes objectionable visual artifacts. When dealing with images that have already been compressed, our ability to use the properties of artifacts present due to the compression algorithm is dependent on our knowledge of which algorithm was used to compress the image. For example, the DCT applied to a block-partitioning of an image results in blocking and "speckling" noise, whose presence is decreased by the post-processing algorithm of Ahumada and Horng (Section 2.2.1) and the preprocessing heuristic algorithm of Bauschke, *et.al.* (Section 2.3.1), respectively.

Without assuming any knowledge of the algorithm used to encode the compressed image, we cannot assume knowledge of the types of artifacts present. Therefore, we are limited in our ability to model the properties of the image, and are restricted to using information about general properties of the visibility of artifacts, of the system used to display the images, and of the human visual system that is perceiving the artifacts. In this chapter, we present properties of the *human Visual System* (HVS), and consider general image quality models based around these properties. We also compare these algorithms to the Mean-Squared Error, both individually and as weights for alternative summation methods.

3.1 Human Visual System Properties

Independent of any knowledge about the compression algorithm used to encode a given image, there are three sources of redundancy inherent in the human visual system [83]:

- *luminance masking*, the relationship between the physical luminance level and the change in luminance required to distinguish a visible difference (see Section 1.3.1);
- *contrast sensitivity*, the nonlinear relationship between a given frequency and the amplitude required for it to be perceivable (see Section 1.3.2); and
- *contrast masking*, the reduction in visibility of a given frequency/pattern due to the presence of another similar pattern, generally due to similar spatial frequencies and orientations (see Section 1.3.3).

Each of these redundancies arises as a result of properties of the physical structure of the human eye and the processing system of the visual cortex. However, the structure of the eye is out of the scope of this thesis; see [18] for an in-depth discussion.

In this section, we will expand our earlier discussions on each of these phenomena, framing our discussions in the context of one of the standard general HVS-based models, Daly's *Visible Differences Predictor* (VDP) [20].

3.1.1 Luminance masking

Luminance masking can be measured using patches of constant luminance displayed over a background of a second constant luminance. Specifically, if a subject is presented with a patch of constant luminance L_o, and asked to vary the luminance of a centrally located square until it becomes distinguishable from the background, then the threshold luminance change $L_o + L_t$ will depend linearly on L_o, i.e.

$$\frac{L_t}{L_o} = k \tag{3.1}$$

for some constant value k [51]; this linear relationship occurs over a subrange of luminance levels which will generally include those in natural images, although additional relationships have been observed for lower luminance levels [18].

Since luminance masking is measured using artificial stimuli, its model must be extended for natural images, which requires an interpretation of the constant background and of the foreground patch. One method is to use each pixel as the constant patch and use an average of surrounding pixels as the constant background, such as the (weighted or unweighted) average of all pixels within a certain “radius” of the given pixel. Another method is to apply a lowpass transform to the image’s frequency spectrum, and use the resulting low frequency image as the pixel-wise "background". A third method is to apply a nonlinear transformation to each pixel such that equal differences in the transformed values can be weighted equally. For a given luminance L, some commonly-used transforms are $\log(L)$ and L^α for $\alpha \in (0,1)$. In the VDP, Daly used the latter model: for a given image of luminance levels $L = [L_{i,j}]_{i,j=1}^{M,N}$, the weighted image is defined by the formula

$$\frac{L(i,j)}{L(i,j) + (cL(i,j))^b} \tag{3.2}$$

where b and c are parameters that vary with the maximum luminance level of the display.

3.1.2 Contrast Sensitivity

Contrast sensitivity describes the relationship between a given frequency and the amplitude required for it to be perceivable. Contrast sensitivity is the inverse of the *contrast threshold*, which is measured using a *(frequency) grating*, the general form of which is defined by

$$I(x,y) = \alpha \cos(2\pi\gamma(x\cos\theta - y\sin\theta)) \tag{3.3}$$

where α, γ and θ are the grating’s amplitude, frequency and orientation, respectively. A grating of a fixed frequency and orientation is added to a constant background, and the amplitude is increased until the grating is distinguishable from the background; performing this experiment over a variety of frequencies gives the *contrast sensitivity function* (CSF), which measures contrast sensitivity as a function of frequency.

A useful description of the CSF is shown in Figure 3.1, which shows a sine frequency added to a constant background, with the frequency increasing linearly from left-to-right and the amplitude decreasing linearly from top-to-bottom. Clearly visible in this Figure is the contrast threshold, found by following the outline of where the sine frequency is first observable from the background; normalizing such that the height of the image is equal to 1, dividing by the contrast threshold for each frequency gives an intuitive sense of the shape of the CSF.

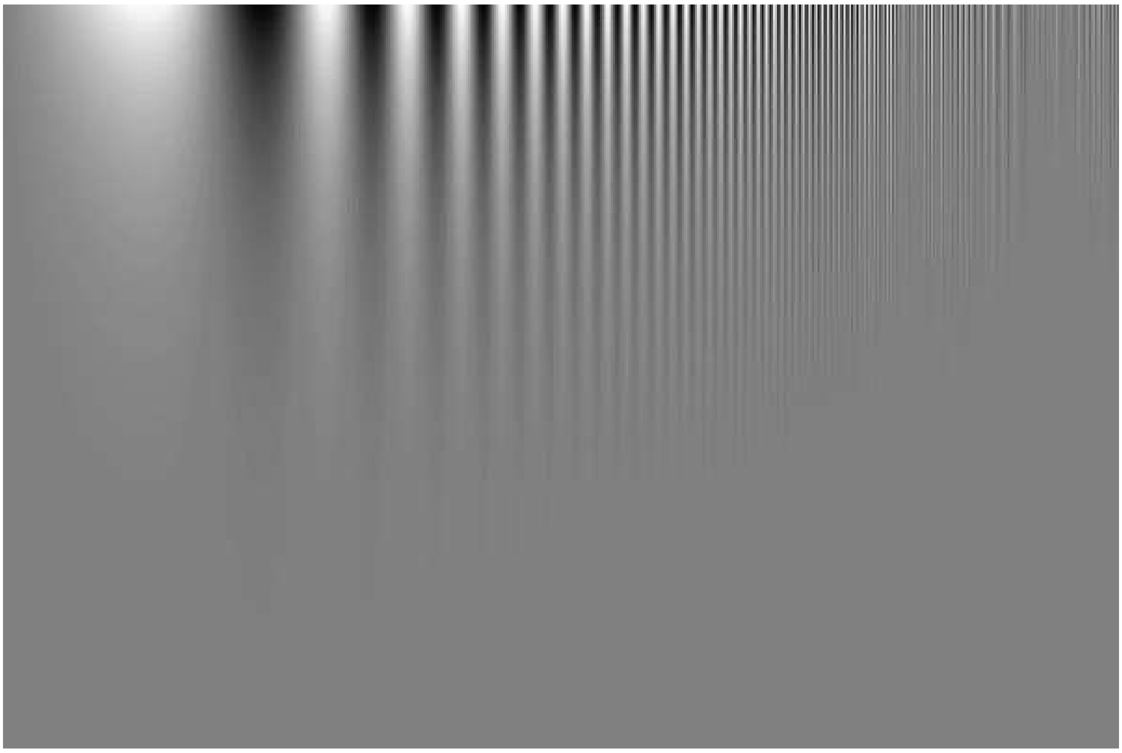

Figure 3.1: Demonstration of contrast threshold: the outline between the frequency grating and the constant background shows the relationship between a given frequency and the contrast threshold [18]

Mannos and Sakrison defined a general form for the CSF by

$$C(\gamma) = a(b + c\gamma)e^{(-d\gamma)^g} \tag{3.4}$$

where a, b, c, d and g are parameters whose values can be calculated experimentally[1] [40]. Some values of these parameters for common CSFs are shown in Table 3.1. To apply these to two-dimensional natural images, the 2-dimensional Discrete Fourier Transform [11] of the image is shifted to center its DC term, and is pointwise transformed by multiplying each frequency f by the CSF evaluated at $|f|$. Therefore, these one-dimensional CSFs assume that the contrast threshold is isotropic. However, although it is nearly isotropic, the contrast threshold has been observed to increase at gratings with $\theta = \pm\pi/2$. The CSF included in

[1]In their original paper, Mannos and Sakrison actually used only four parameters, with c and d treated as the same parameter. However, more recent treatments have used separate parameters, so we will focus our discussion on this more general form.

Authors	a	b	c	d	g	Reference
Mannos & Sakrison	2.6	0.0192	0.114	0.114	1.1	[40]
Nill	1	0.2	0.45	0.18	1	[46]
Ngan	1	0.31	0.69	0.29	1	[45]
Chitraprasert & Rao	2.46	0.1	0.25	0.25	1	[59]

Table 3.1: Values of parameters in (3.4)

the VDP implementation avoids this isotropy assumption, and its discretization over the frequency plane is shown in Figure 3.3, along with those of Mannos & Sakrison, Ngan, and Chitraoprasert & Rao, for comparison; see [59] for a complete discussion of these various CSFs.

3.1.3 Contrast masking

Contrast masking refers to the decrease in sensitivity to a given pattern due to the presence of another similar pattern. Models of contrast masking focus on two operations, namely frequency decompositions and masking functions. The frequency decomposition partitions the frequency plane into windows, considering masking effects separately for each window; it is well-known that masking effects occur when the original input pattern has similar spatial frequency and orientation to the additional artifacts, so the windows are designed to closely model these observed results.

Some commonly used frequency decompositions are the Laplacian pyramid of Burt and Adelson [12], and the Steerable filters of Freeman and Adelson [24] and Simoncelli, Freeman, Adelson and Heeger [68]. We will focus our discussion on the Cortex Transform of Watson [82] and specifically the variation of Daly used in the VDP implementation [20].

Given the need to localize frequency and orientation information, the Cortex Transform performs a sequence of bandpass filters to separate the frequency plane into radial and orientation bands. The orientation bands are formed via what Watson termed *fan filters* defined by a single sinusoidal crest centered on a given orientation, where the centered orientations are evenly divided around the frequency plane. The number of filters is left as a parameter L; these filters are shown in Figure 3.4 for $L = 6$, the value of L recommended by Daly for the VDP.

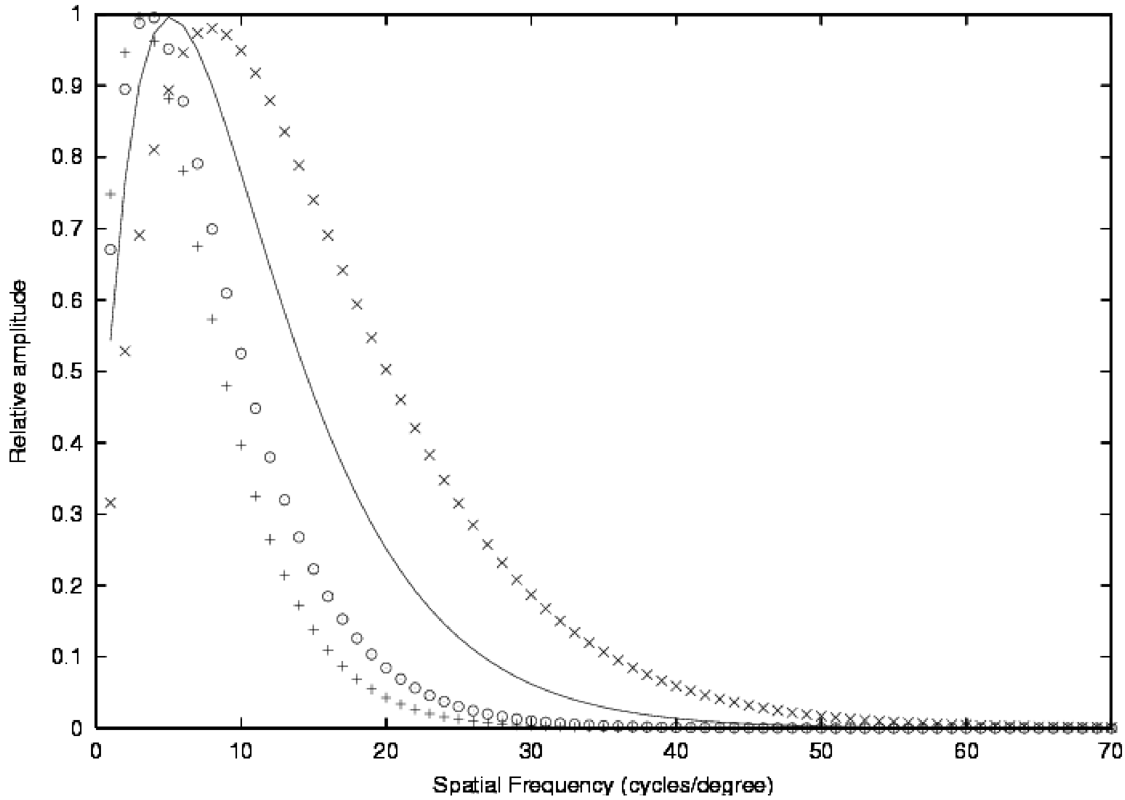

Figure 3.2: Graphs of the CSF functions defined by parameters in Table 3.1 with Mannos & Sakrison ($\times$), Nill (line), Ngan ($+$), and Chitraoprasert & Rao ($\circ$)

The spatial frequency bands are defined using three filters: the *mesa filter*, a sequence of low-pass filters with varying cutoff frequency, consisting of a unit pass-band with a sinusoidal transition window; the *dom filter* which uses the difference between two mesa filters of different width to act as a spatial frequency window; and the *base filter*, which defines the DC and low-frequency window, and consists of a Gaussian with parameter defining the baseband width. As with the fan filters, the number of dom filters is left as a parameter K; the dom filters are defined such that the radius of the mesa filters increases by a factor of two, with the radius of the first mesa filter chosen such that the transition window into the final dom filter reaches the boundary of the frequency spectrum. The dom filters are shown in Figure 3.5 for $K = 6$, the value of K recommended by Daly for the VDP implementation; notice that the dom filter of the smallest radius is the base filter.

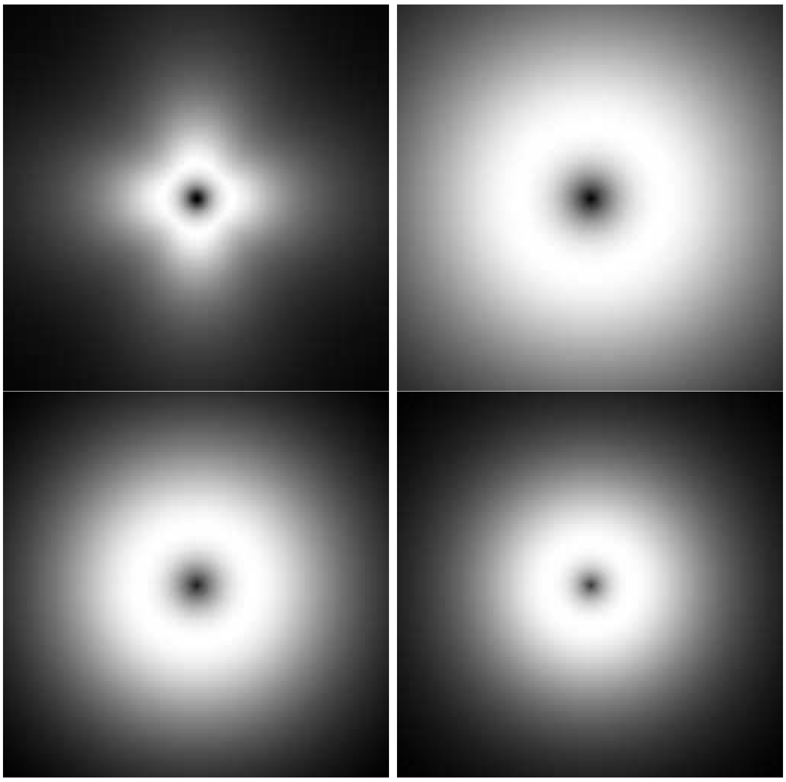

Figure 3.3: The CSFs of Daly (top left), Mannos & Sakrison (top right), Ngan (bottom left), and Chitraprasert and Rao (bottom left); note that only the CSF of Daly is nonisotropic, with an increase in the threshold for mid-frequency signals diagonally oriented.

The *cortex filter* for spatial frequency band k and radial band l is defined as the entrywise product of the kth dom filter and the lth fan filter; the frequencies weighted by the baseband are not separated into orientation bands, resulting in $K(L-1)$ cortex filters. The windows created by the partitioning of the frequency plane defined by the cortex transform are called *cortex bands*; a sample cortex band, with the component spatial and radial bands, is shown in Figure 3.6.

Since contrast masking is an effect local to each cortex band, the comparison of contrast levels is performed in each cortex band, and then combined afterwards. The general shape of the masking function is measured by adding a frequency grating to a background consisting of another frequency grating of the same frequency with a fixed amplitude, with the amplitude of the added grating adjusted until a difference is perceivable; the increase in contrast threshold as a result of the mask is called the *threshold elevation*. The resulting

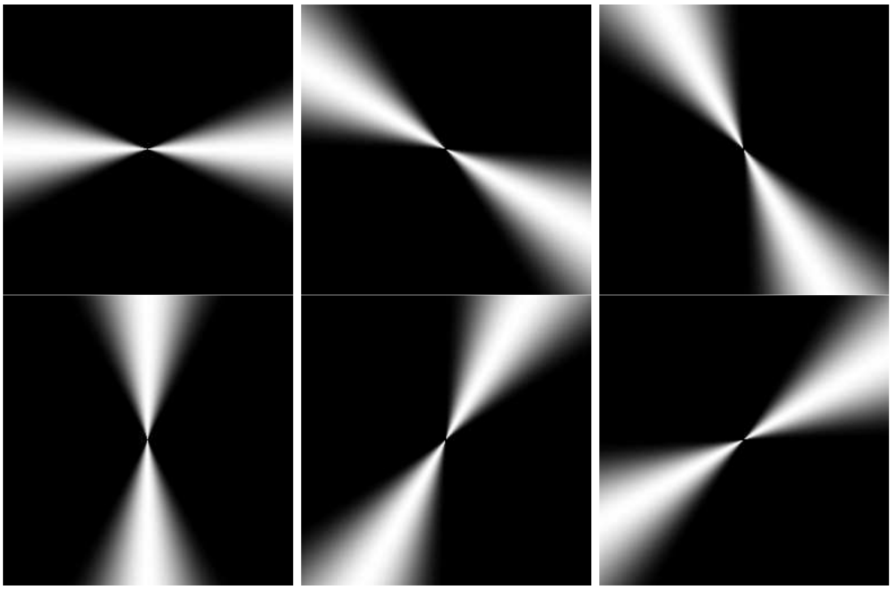

Figure 3.4: Fan filters for $L = 6$ [20, 82]

masking function is a function of the mask contrast, and returns the threshold elevation; the graph of a typical masking function is shown in Figure 3.7, with the linear region for low mask contrasts resulting from the mask itself not being perceivable, and thus no additional masking present [18]. Typically, the results within each band are combined by an entrywise L_p-summation over all of the bands, where the value of p is dependent on the individual measure; see [1] for specific values of p for various perceptual measures.

The resulting matrix is interpreted as an entrywise measure of the degree of dissimilarity at each pixel, and thus an additional pooling stage is necessary if a scalar-valued measure is desired. However, the matrix itself can serve as a useful tool during the process of designing an image compression algorithm, as it can identify within the image areas where the algorithm is creating visible artifacts. Furthermore, the resulting matrix is not dependent on knowledge of the encoding method used to compress the two images being compared, further distinguishing the more general HVS-based measures from perceptual measures specific to comparing DCT- or block-encoded images, such as the models of Ahumada, Watson, and Bauschke discussed in Sections 2.2.1, 2.2.2 and 2.3.

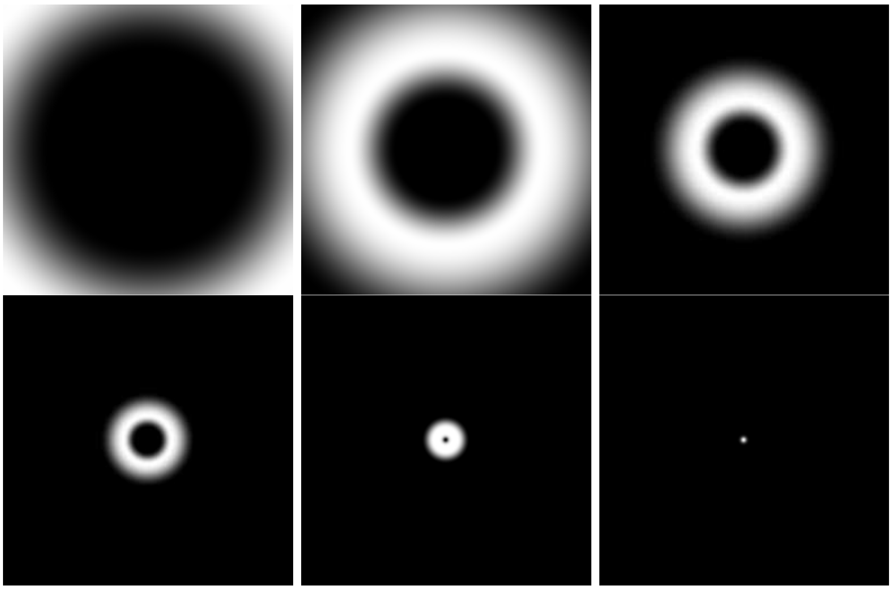

Figure 3.5: Dom filters for $K = 6$ [20, 82]

3.1.4 Other General Models

The specifics of Daly's Visible Differences Predictor, described throughout this section, illustrate the structure of HVS-based models. However, we have given only enough of a description of the HVS-based model to illustrate techniques of modelling the perceptual phenomena discussed at the outset of this section; we invite the reader to our implementation of the VDP, documented in Appendix B.1. The general structure of HVS-based models is illustrated in Figure 3.8; other commonly cited models of this type include those of Lubin [39] and Teo and Heeger [73]. Overviews of these algorithms can be found in [1, 51].

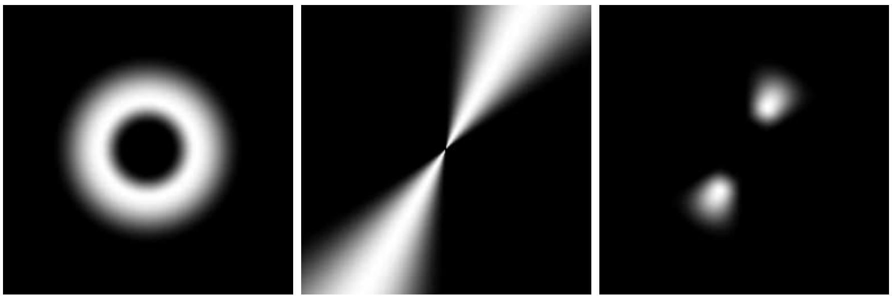

Figure 3.6: Sample cortex filter (right), with component dom and fan filters (left and middle, respectively) [20, 82]

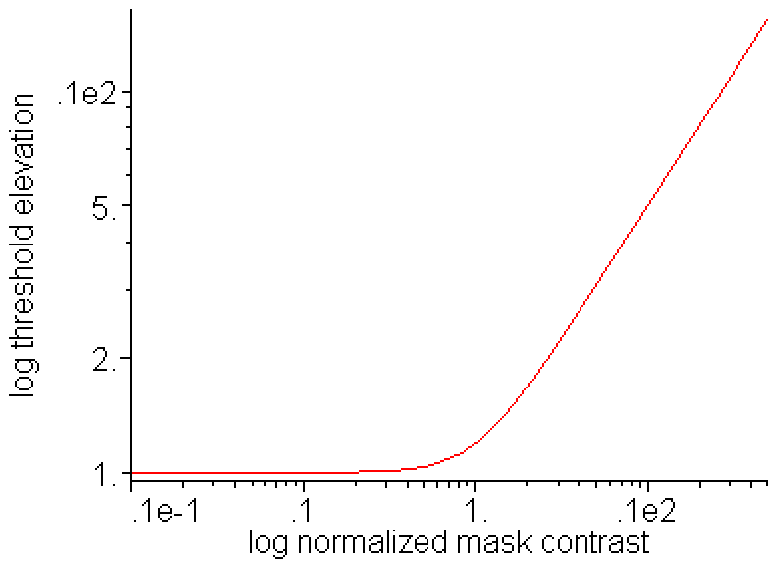

Figure 3.7: Log-log plot of masking function, with low-contrast linear region and high-contrast exponential region

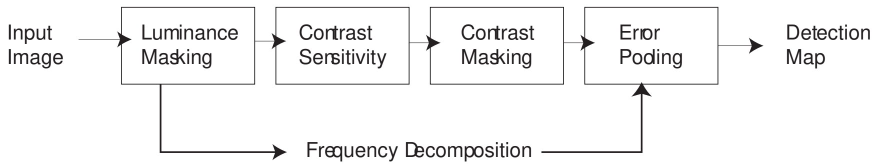

Figure 3.8: Structure of general HVS-based quality metrics [51]

Notice that in Figure 3.8, we have kept the frequency decomposition stage outside of the three main stages to illustrate its accessibility at any stage during the model. In fact, this stage can be viewed as decomposing the overall image quality into separate measures of image quality, measured locally over each subband. Therefore, one potential area of improvement in these models is to forego the first error pooling stage across all frequency windows, and to treat the output from each frequency window as a single component of a multi-component image fidelity measure. For example, if we knew that the given image was JPEG-encoded, then we might want to weight more heavily the frequency windows consisting of horizontal and vertical frequencies, since these would detect block edges; this option is not available if we blindly pool the errors as part of our fidelity measure. We focus on the issue of multifactored measures in Section 4.4.

3.2 High-level Visual Processes

The general perceptual measures described in the previous section focus on properties of the Human Visual System, and thus on measuring errors that are perceptible by the HVS, based on results of subjective experiments using artificial stimuli; the content of the image is considered in terms of its ability to mask errors due to their similar frequency content. However, in these models no consideration is given to the fact that natural images have *cognitive content* as well, and that errors will more likely be detected if they occur in an area of the image of high interest due to its visual content; for example, in the image shown in Figure 3.9, the mountaintop is more likely to be viewed than the clouds, due in part to it being centrally located within the image and providing a high-contrast against the background of the clouds. This relationship between the likelihood of an error being the focus of the viewer's attention and the presence of the error within the areas of high semantic interest is not considered by HVS models.

This section provides an overview of a recent model by Osberger [49, 50], building the discussion towards our implementation in Appendix B.1.

3.2.1 Visual Attention

While the HVS models consider images as combinations of component frequencies, visual attention models focus on images as *carriers of visual/semantic information* [31], thus focussing on errors that are attended by the viewer due to the presence of given visual properties. Visual attention models focus on individual objects within an image as the visual information carriers, and calculate the probability of a viewer attending each object by measuring the presence of visual properties of interest within each object.

The precise properties of interest are determined using visual search experiments: a subject is presented with a collection of identical stimuli, of which a target stimulus may be present that differs by the presence of a given visual property; the amount of time required for the subject to determine whether the target stimulus is present in the collection is used to calculate the *Efficiency-of-Search* (EOS). The total number of stimuli in the collection is varied over subsequent test repetitions; stimuli of interest in the visual attention model are those whose EOS is slow to increase as the total number of distracter stimuli increases.

Figure 3.9: Sample natural image, used to illustrate semantic processing involved in interpreting natural images

Given these *basic stimuli*, the quality model considers each object within an image independently by segmenting the image and measuring the degree to which each stimulus is present in each object. The model of Osberger [49] focusses on three properties:

- the human visual system has high acuity only in the central few degrees of the visual field, and thus is highly sensitive to stimuli located in the centre of the visual field;
- the visual system focusses on *objects* within the image, rather than *regions* of the image;
- the visual system alternates from focussing on the centre of the visual field to searching the peripheral regions for a new object to focus on, with objects of high interest/complexity/uncertainty more likely to be selected.

The general structure of the underlying process is illustrated in Figure 3.10: a centrally-located region of the image is focussed on, and artifacts within this region are more likely noticed; the peripheral areas are then searched for areas of high interest/complexity/uncertainty, and the visual field is shifted to centrally locate the resulting peripheral area. This new centrally-located area is then focussed on, and thus the process repeats.

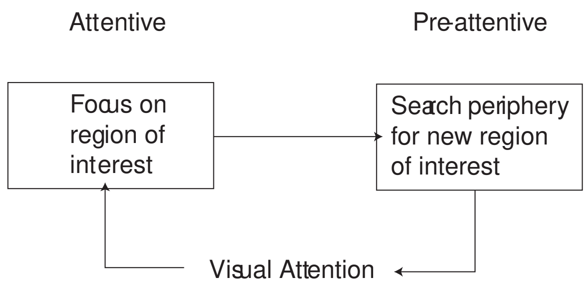

Figure 3.10: Role of visual attention in process of scene interpretation [49]

3.2.2 Properties of stimuli that attract visual attention

Expanding on the visual attention processes described in the previous section, we can consider perceptual image quality as *existing differences that are focussed on by the visual system*, using properties of visual attention to predict what differences will be focussed on. Since pre-attentive processes partition an image into *objects*, we consider properties of objects that increase the probability of them being attended.

For still images, some of the primary properties that have been shown to attract visual attention are as follows:

- *contrast*: objects are more likely attended as contrast from adjacent objects increases;
- *colour*: objects are more likely attended as differences in colour from surrounding regions increases;
- *size*: larger objects are more likely attended;
- *location*: objects located near the center of the picture are more likely attended;
- *shape*: unusual or unpredictable shapes are more likely attended than highly regular/symmetric shapes;
- *spatial depth*: objects located in the foreground of the image are more likely attended than objects located in the background of the image.

Other important factors that attract attention, but which are of less interest in modelling fidelity for still images, include [49]:

- *motion*: moving objects in an image sequence are most frequently attended than stationary objects;
- *eccentricity*: objects located near a given point of fixation are more likely attended;
- *context/instructions*: objects of interest within an image will depend on the instructions given to the subject, or on the context within which the image is being viewed;
- *people*: if an image contains human shapes, then they are most frequently attended;
- *clutter/complexity*: if a region of an image is highly complex, then it will be more frequently attended;
- *unusual/unpredictable objects*: objects that seem out of place within a given scene will be more frequently attended.

Motion, although the primary factor of interest for video compression, is not of interest for still-image compression, and thus will not be included in our discussions. The remaining factors are useful to keep in mind when designing experiments to measure subjective image quality, but are not expected to be implementable.

Visual attention is included in modelling perceptual image fidelity by segmenting an image into *objects*, and weighting each object according to the degree to which each visual factor discussed earlier is present within the given object. Each of the separate factors are then combined into a single perceptual map, called the *Region-of-Interest* (ROI) map, which represents the probability that each object will be the focus of the subject at some point during their viewing of the given image; as such, the ROI map is measured relative to a single image, as opposed to the VDP map, which was measured as a function of two input images. Since Osberger's original algorithm was focussed on modelling the impact of visual attention on perceptual image fidelity for block-encoding algorithms, he defined an additional map, called the *Importance Map* (IM), which is a smoothed version of the *Region-of-Interest* (ROI) map; all of the entries within an $n \times n$ block are defined to be the largest value taken by any of the corresponding entries in the ROI map, the theory being that the importance of a block within the context of the original image is dependent on the most important regions within the given block.

We invite the reader to examine our implementation of Osberger's model, outlined in Appendix B.1: sample output is shown in Figure 3.11 applied to the standard `bird` test image; we follow Osberger's recommended outline in [49], measuring each object's shape, size, location, spatial depth, and contrast, although Osberger left open the specific properties to include in a final implementation; in a more recent paper [50], Osberger included color and the presence of skin tones to the still-image model.[2]

3.3 Alternate Summations

Given the failure of the MSE-variants to predict simple visual phenomena, we now compare the results of the MSE-variants to those of alternate *summation methods*, or functions $f : M^{m,n} \times M^{m,n} \rightarrow \bar{\Re}$, where $M^{m,n}$ is the space of all $m \times n$ non-negative real-valued matrices. In particular, we will consider those functions that make only entry-wise comparisons; our goal in doing so is to determine to what extent the occurrence of the pathologies discussed in the previous section are inherent to entry-wise quality measures, or if an alternate summation can achieve better results.

3.3.1 Topological comparisons

A number of papers have analyzed the structure of the output of various proposed image quality measures, the most notable being the extensive experimental results of the *Video Quality Experts Group* (VQEG) [76]; the standard approach is to perform a single set of subjective experiments, with input images encoded using compression methods that vary the types of artifacts used, such as JPEG compression, wavelet compression [25, 72], or fractal compression [23]. The proposed quality measures were then applied to the compressed images, and the correlation between the output of the measures and the subjective responses is used as a measure of similarity. One such experiment compared nine measures, including PSNR, and found that eight of the nine had very similar correlation to that of PSNR, with the ninth measure having lower correlation [77].

[2]Note that Osberger's model incorporated a combination of an HVS-based entrywise weighting, based on the VDP, in calculating the final error weighting; although we have included this in our implementation, we have separated each model in our discussions for the purposes of illustrating our optimization applications.

One problem with this method is that it stresses the magnitudes of the measures rather than the *rank ordering* given by these measures. For example, consider two measures f and g and two approximations $\hat{X}$ and $\bar{X}$ of a given original image X that satisfy

$$|f(\hat{X},X) - f(\bar{X},X)| < \epsilon \quad \text{and} \quad |g(\hat{X},X) - g(\bar{X},X)| < \epsilon \tag{3.5}$$

$$|f(\hat{X},X) - g(\hat{X},X)| < \epsilon \quad \text{and} \quad |f(\bar{X},X) - g(\bar{X},X)| < \epsilon \tag{3.6}$$

$$g(\bar{X},X) > g(\hat{X},X) \quad \text{and} \quad f(\bar{X},X) < f(\hat{X},X) \tag{3.7}$$

for a small value of $\epsilon > 0$. The correlation will lose the differences between the rank ordering of f and g (namely that f "prefers" $\hat{X}$ while g "prefers" $\bar{X}$) because of the similar magnitude of the quality measures. However, if we are looking for a local minimum for our quality measure, the rank ordering of the measure is the primary quantity of interest, and thus we need a method of quantifying the degree of similarity in the rank ordering of two quality measures.

In the present discussion, we will use the following definition:

Definition 1 *Let X be a given image, and let X_k denote the JPEG-encoded image at IJG quality factor $q = k$. For two functions $f, g : M^{m,n} \times M^{m,n} \to \Re$, we say that f and g* differ topologically *for X_k if there exists l such that either $f(X,X_k) > f(X,X_l)$ and $g(X,X_k) < g(X,X_l)$ or $f(X,X_k) < f(X,X_l)$ and $g(X,X_k) > g(X,X_l)$.*

In other words, f and g differ topologically if there is a pair of approximations of the original image for which the rank order of the fidelity ratings of the two image approximations differs.

The above definition presents a framework with which to analyze summation methods: given the previous similarity found between proposed measures and the MSE, measures are of interest that differ topologically from the MSE-variants over some subset of the IJG quality scale. Table 3.2 contains a list of entry-wise summation methods that will be compared structurally to the MSE-variants; entries (1)-(10) are compiled in [19], (11) is the Signal-to-Noise ratio discussed in Section 1.2, and (12) is a general measure that we have found to be useful in vector quantization applications.

For the purposes of comparing various pixel-based summation methods, we will define three methods of quantifying the degree of similarity of the rank ordering of these summations to the MSE-variants. The underlying assumption is that the aforementioned shortcomings of the MSE-variants will not be avoided by a measure with a similar rank ordering to these measures, and thus we will be interested in summation methods that differ highly from the MSE-variants.

The first structural comparison method is designed to find measures that differ in their rate of decrease/increase along the IJG quality scale; it does not specifically detect topological differences as defined above, but will detect differences in magnitude when topological differences exist. This method normalizes the range of summation values to $[0, 1]$, and compares the normalized values at each entry.

Algorithm 3 *Let X be a given image, and let X_k denote the JPEG-encoded image at IJG quality factor $q = k$, with given upper and lower bounds on q of q_{high} and q_{low}, respectively Then:*

1. *Measure $\hat{M} = [MSE(X, X_i)]_{i=q_{low}}^{q_{high}}$ and $\hat{F} = [f(X, X_i)]_{i=q_{low}}^{q_{high}}$ over given range of IJG values.*

2. *Define M and F as the values of $\hat{M}$ and $\hat{F}$ normalized and rescaled such that $M, F \subset [0, 1]$: if $\hat{M}_{min}$ and $\hat{M}_{max}$ (resp. $\hat{F}_{min}$ and $\hat{F}_{max}$) are the minimum and maximum calculated values of $\hat{M}$ (resp. $\hat{F}$), respectively, then*

$$M_i = \frac{\hat{M}_i - \hat{M}_{min}}{\hat{M}_{max} - \hat{M}_{min}}, \text{ for } i = q_{low}, \ldots, q_{high}, \tag{3.8}$$

with F_i similarly defined in terms of $\hat{F}_i$.

3. *Calculate mean-squared differences $\sum_{i=q_{low}}^{q_{high}} (M_i - F_i)^2$.*

The second method detects topological differences between two measures by calculating each measure along the IJG quality scale, calculating an $n \times n$ matrix denoting for each (i, j)-th entry the "preferred" quality level, and comparing these matrices entry-wise for each measure.

Algorithm 4 *Let X, X_i, q_{low}, and q_{high} be defined as in Algorithm 3, and let $n = q_{high} - q_{low} + 1$. Then:*

1. *Define matrices M and F to be $n \times n$ matrices, with each entry of M and F defined as follows:*

$$M_{i,j} = \begin{cases} 1, & MSE(X, X_i) > MSE(X, X_j); \\ 0, & otherwise, \end{cases} \tag{3.9}$$

and

$$F_{i,j} = \begin{cases} 1, & f(X, X_i) > f(X, X_j); \\ 0, & otherwise, \end{cases} \tag{3.10}$$

for $i, j = q_{low}, \ldots, q_{high}$.

2. *Calculate the probability of* $M_{i,j} = F_{i,j}$ *taken over all* $i, j = q_{low}, \ldots, q_{high}$, *i.e., the probability that MSE and* f *agree topologically over the range of IJG quality factors of interest.*

The third method is a refinement of the second method, in that it calculates the same matrices, but now only compares quality levels within a given distance r from each other, measured in absolute value. This avoids the method getting decreased excessively by irrelevant comparisons (i.e., comparing quality levels 1 and 100, whose rank ordering is (presumably) independent of the summation method), and gives more weight to known topological differences like the perceptual dipper of Section 2.3.1.

Algorithm 5 *Let* X, X_i, q_{low}, *and* q_{high} *be defined as in Algorithm 3,* n *be defined as in Algorithm 4, and let* $r \in \Re$. *Then:*

1. *Define matrices* M *and* F *to be* $n \times n$ *matrices, with each entry of* M *and* F *defined as follows:*

$$M_{i,j} = \begin{cases} 0, & |i - j| \geq r; \\ 1, & |i - j| < r \text{ and } MSE(X, X_i) > MSE(X, X_j); \\ 0, & otherwise, \end{cases} \tag{3.11}$$

and

$$F_{i,j} = \begin{cases} 0, & |i - j| \geq r; \\ 1, & |i - j| < r \text{ and } f(X, X_i) > f(X, X_j); \\ 0, & otherwise, \end{cases} \tag{3.12}$$

for $i, j = q_{low}, \ldots, q_{high}$.

2. *Calculate the probability of* $M_{i,j} = F_{i,j}$ *taken over all* $|i - j| < r$, *i.e., the probability that MSE and* f *agree topologically.*

3.3.2 Results of topological comparisons

The average results of applying Algorithms 3, 4 and 5 to 21 test images are shown in Table 3.4. Since Algorithm 5 depends on the radius used to delimit the difference in the IJG quality scale factor between the two comparison images, we have included in Table 3.3 the results for various radii. Some observations about these results:

- Algorithm 3 responds differently to topological similarities than algorithms 4 and 5. For example, algorithm 3 measures divergence as being more similar to MSE than the other measures (1)-(8), which is the opposite determination made by algorithms 4 and 5. Since Algorithms 4 and 5 strictly measure topological differences, we will consider Algorithm 3 as a refinement of 4 and 5 when used together.

- Algorithms 4 and 5 respond to the same structure in the measure being compared to MSE; however, algorithm 5 is much more sensitive to small differences between the measure and the MSE. For example, the numerical difference between the topological structure of MSE and that of either the Wave-Hedges or correlation measures is substantially more when measured by 5 than 4. This confirms our original prediction that the most topological differences occur when the IJG quality factors of the compressed images being compared are similar, and motivates our focus on the measurements of Algorithm 5 in the remainder of our discussion.

- We have included the results for the L_p-measures for $p = 1, 2, 3$, but we also calculated Algorithms 3-5 for values of $p = 4, \ldots, 10$. However, increasing p further does not add any additional structure, so we have omitted the results.

- The L_2-measure is the closest to MSE in its definition, which is confirmed by the high degree of similarity as measured by Algorithms 4 and 5. However, more surprisingly, most of the other measures are measured as being almost as similar. In particular, the Bray-Curtis, Soergel, Bhattacharyya, angular separation, and correlation measures are almost as similar accordingly to Algorithm 5, while the Canberra and Wave-Hedges measures are only slightly more dissimilar. For this reason, we limit our algorithms in the remainder of our discussion to considering the Canberra, divergence, Wave-Hedges, block roughness, edge variance, and σ-threshold measures.

- We have included the results for the σ-threshold measure for $\sigma = 1, \ldots, 10, 15, 20, 25$. As can be observed by comparing Figures 3.12 and 3.13, this measure starts with a number of very noticeable topological differences from MSE for $\sigma = 1$. This structure is gradually smoothed as σ increases, up to a maximum degree of similarity when $\sigma = 9$; in Figure 3.12, we can see that at $\sigma = 5$ the measure already starts to resemble the MSE, shown in Figure 3.13. As σ increases further, Algorithm 5 decreases as all structure is removed as every pixel is thresholded, thus returning 0 for a large subset of the IJG quality scale; this demonstrates that a smaller value for Algorithm 5 can be misleading. For these reasons, we limit our focus on the σ-threshold measure to $\sigma = 1, 2, 3$.

- Although the results shown in Table 3.4 is averaged across 21 test images, the results for each image did not vary substantially, indicating that the observations made from the average results are specific to the DCT-encoding rather than the image contents.

- Table 3.3 shows the effect of increasing the radius δ on Algorithm 5, calculated with the `bird` test image; not surprisingly, the probability of topological agreement increases as δ increases, with the rate of increase decreasing as δ increases. In the remainder of our discussion, we have chosen to use Algorithm 5 for $\delta = 3$, since this value provides a transition between the faster and slower rates of increase.

- In Table 3.3, notice that Algorithm 5 does not increase monotonically for the block roughness measure. In fact, the block roughness measure maintains a value between 0.2 and 0.26 for $\delta = 7, \ldots, 10$, indicating that the low amount of topological agreement is due to the lack of structure in this measure.

- Tables 3.5 and 3.6 show the results of using each of the 5 perceptual maps used to calculate the ROI perceptual map as pixel-weightings for the measures. These pixel-weightings had only a small effect on the results of Algorithm 3. However, the effects on Algorithm 5 applied to the σ-threshold measure are quite interesting. In particular, the Size, Spatial Depth, ROI and IM maps have minimal effect on the topological similarity, while the Shape and Contrast maps cause a decrease in similarity, likely due to the high weighting of smaller objects consisting of numerous merged blocks split across different 8×8 matrices. The Location map causes a sharp increase in similarity, likely due to the fact that this particular image has a large portion of its content near the center of the image. This central content suffers more data loss than the background due to the presence of higher frequency content, thus increasing the percentage of the image that differs by more than σ; the MSE is similarly increased due to the low weights given to the background content, which suffers less data loss and thus decreases the average squared difference over the entire image. However, it is difficult to determine what impact the use of these component maps have on Algorithm 5 without performing this experiment on a larger set of images.

- Over the entire set of 21 test images, no topological differences were found between the MSE and the IJG quality scale, further confirming the hypothesis that the MSE is always monotonically decreasing along the IJG quality scale. Given the amount of rounding that occurs during the quantization stage of the JPEG-encoding algorithm, this observation is still an open question.

3.4 Conclusion

This chapter presents two techniques of modelling perceptual image fidelity, namely the HVS-based model and the Visual Attention model. We focussed on an example of each type of model, namely the Daly's Visible Differences Predictor (VDP) and Osberger's Region-of-Interest Importance Map (IM). We provide for the reader an implementation of both examples, with documentation available in Appendix B.1.

We also present experiments with measures of the degree of structural similarity between a given measure and the Mean-Squared Error, although our algorithms can be applied to any pair of measures. Our observations with the results of this experiment suggest that the shortcomings of the Mean-Squared Error in detecting phenomena such as the Perceptual Dipper of Section 2.3 are not entirely a result of limiting the comparisons to individual pixels, as opposed to considering the frequency content of the entire image. We base this conclusion on the fact that, while most of the other measures based on pixel-comparisons were very similar in their structure to the MSE, we were able to identify other pixel-based methods with considerable topological differences from MSE, where we measure the differences according to the degree to which the rank-orderings differ. These experiments can also be repeated using the application described in Appendix B.1.

Although we tried using the Region-of-Interest component maps as pixel-weightings in these experiments (with mixed results), a more interesting experiment would be using the VDP in this role. However, this is extremely expensive computationally; while the Region-of-Interest component maps are properties of a single image, and thus only need to be computed once during the experiment, the VDP would need to be computed once for each IJG quality factor value. However, this computation could easily take advantage of a parallel environment; as such, we leave this experiment for a later date.

(1)	L_p metrics	$d(X,Y) = \left(\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{i,j} - Y_{i,j})^p\right)^{1/p}$
(2)	Weighted L_p metrics	$d(X,Y) = \left(\sum_{i=1}^{M}\sum_{j=1}^{N}w_{i,j}(X_{i,j} - Y_{i,j})^p\right)^{1/p}$, for $w_{i,j} \in \mathbb{R}$
(3)	Canberra metric	$d(X,Y) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\lvert X_{i,j} - Y_{i,j}\rvert}{\lvert X_{i,j} + Y_{i,j}\rvert}$
(4)	Divergence	$d(X,Y) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{(\lvert X_{i,j} - Y_{i,j}\rvert)^2}{(X_{i,j} + Y_{i,j})^2}$
(5)	Bray-Curtis	$d(X,Y) = \frac{1}{MN}\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\lvert X_{i,j} - Y_{i,j}\rvert}{\sum_{i=1}^{M}\sum_{j=1}^{N}\lvert X_{i,j} + Y_{i,j}\rvert}$
(6)	Soergel	$d(X,Y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\lvert X_{i,j} - Y_{i,j}\rvert}{\sum_{i=1}^{M}\sum_{j=1}^{N}\max(X_{i,j}, Y_{i,j})}$
(7)	Bhattacharyya	$d(X,Y) = \left(\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{i,j}^{1/2} - Y_{i,j}^{1/2})^2\right)^{1/2}$
(8)	Wave-Hedges	$d(X,Y) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(1 - \frac{\min(X_{i,j}, Y_{i,j})}{\max(X_{i,j}, Y_{i,j})}\right)$
(9)	Angular Separation	$d(X,Y) = 1 - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}X_{i,j}Y_{i,j}}{\left[\left(\sum_{i=1}^{M}\sum_{j=1}^{N}X_{i,j}^2\right)\left(\sum_{i=1}^{M}\sum_{j=1}^{N}Y_{i,j}^2\right)\right]^{1/2}}$
(10)	Correlation	$d(X,Y) = 1 - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{i,j} - \bar{X})(Y_{i,j} - \bar{Y})}{\left(\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{i,j} - \bar{X})^2\sum_{i=1}^{M}\sum_{j=1}^{N}(Y_{i,j} - \bar{Y})^2\right)^{1/2}}$ where $\bar{X} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}X_{i,j}$ and $\bar{Y} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}Y_{i,j}$

Table 3.2: Scalarization methods used in experiments on measuring structural similarity to MSE-variants (entries (1)-(10) compiled in [19]).

(11)	Signal-to-Noise Ratio	$d(X,Y) = 10\log\left(\frac{\sum_{i=0}^{M}\sum_{j=0}^{N}(X_{i,j})^2}{\sum_{i=0}^{M}\sum_{j=0}^{N}(X_{i,j}-Y_{i,j})^2}\right)$
(12)	σ-thresholded	$d(X,Y) = \sum_{i=1}^{M}\sum_{j=1}^{N}\delta\left(X_{i,j},Y_{i,j}\right),$ where $\delta(u,v) = \begin{cases} 1, & \lvert u-v\rvert \geq \sigma; \\ 0, & \text{otherwise.} \end{cases}$

Table 3.2: Scalarization methods used in experiments on measuring structural similarity to MSE-variants.

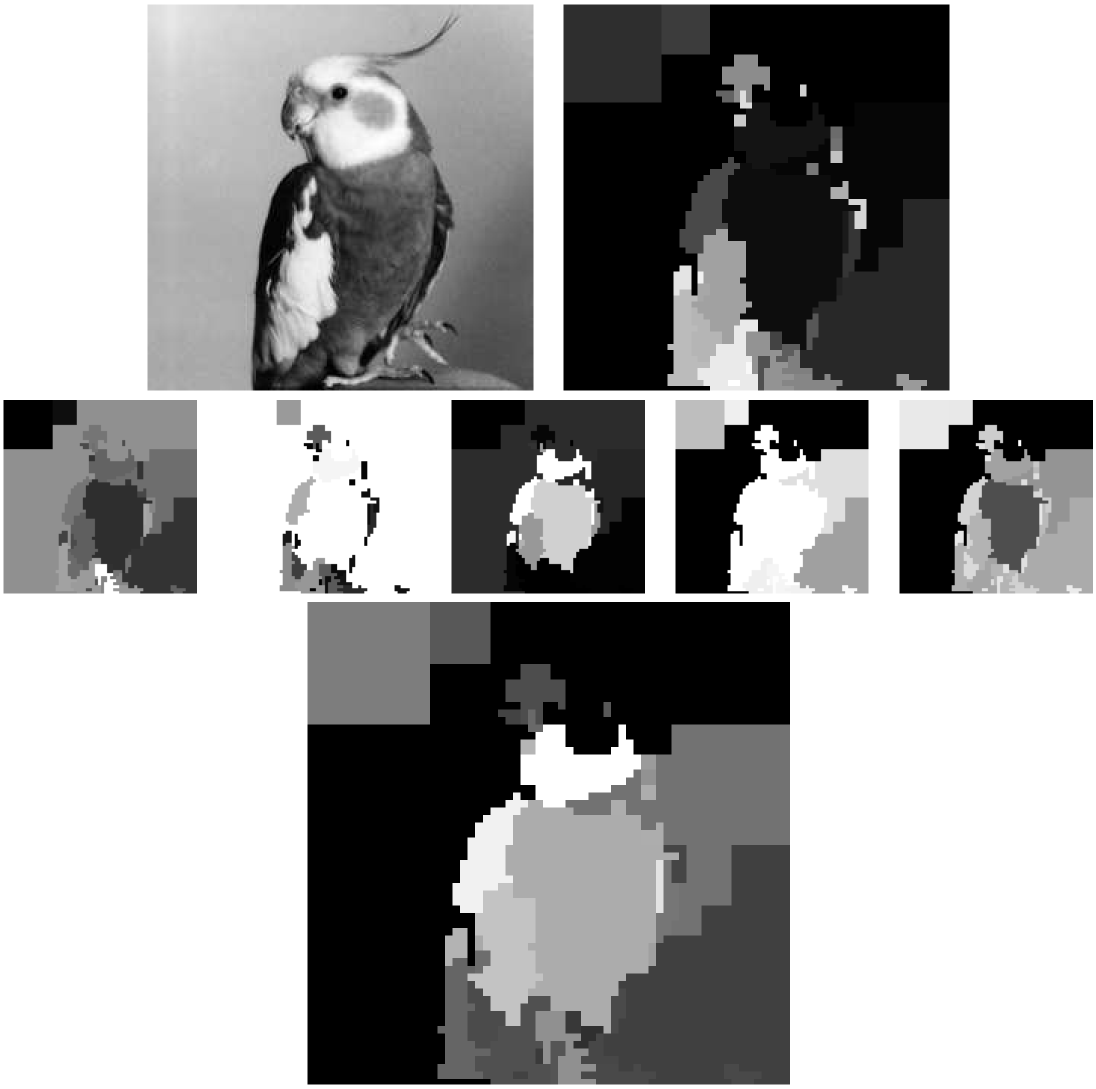

Figure 3.11: Sample entrywise weighting images from visual attention model described in Section 3.2.1; shown are: (from top row, left-to-right) the original image, the image segmented into objects, and the images resulting by weighting each object according to their shape, size, location, spatial depth, and contrast, with the final image (bottom) created by performing an entrywise sum-of-squares the five weighting factors[49].

Summation	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$
(3)	0.598	0.726	0.783	0.817	0.838	0.853
(4)	0.651	0.790	0.849	0.882	0.903	0.917
(8)	0.658	0.794	0.852	0.884	0.905	0.919
Block Roughness	0.216	0.249	0.255	0.254	0.252	0.248
Edge-Variance	0.548	0.681	0.748	0.788	0.814	0.834
(12, $\sigma=1$)	0.551	0.681	0.748	0.790	0.818	0.837
(12, $\sigma=2$)	0.591	0.721	0.780	0.817	0.840	0.856
(12, $\sigma=3$)	0.618	0.749	0.809	0.841	0.862	0.876

Table 3.3: Effects of increasing the radius r on Algorithm 5, for *bird* image: shown are the calculated values for Algorithm 5 for various radius values, $r = 1, \ldots, 6$, for a subset of the summations from Table 3.2; we can observe the increase (in general) of the calculated value as r increases, corresponding to a wider range of quality factors being compared, which confirms that images become more dissimilar as the difference in quality factors increases.

Summation	Algorithm 3	Algorithm 4	Algorithm 5 ($r = 2$)
(1, $p = 1$)	0.0301	0.989	0.846
(1, $p = 2$)	0.0307	0.990	0.852
(1, $p = 3$)	0.0317	0.989	0.847
(3)	0.034	0.986	0.829
(4)	0.012	0.965	0.790
(5)	0.028	0.989	0.845
(6)	0.035	0.989	0.845
(7)	0.031	0.989	0.845
(8)	0.050	0.987	0.834
(9)	3.67E-05	0.990	0.850
(10)	0.005	0.989	0.846
Block Roughness	0.527	0.093	0.255
Edge Variance	0.040	0.965	0.748
(12, $\sigma = 1$)	0.571	0.961	0.696
(12, $\sigma = 2$)	0.460	0.968	0.738
(12, $\sigma = 3$)	0.368	0.977	0.768
(12, $\sigma = 4$)	0.297	0.981	0.778
(12, $\sigma = 5$)	0.241	0.982	0.780
(12, $\sigma = 6$)	0.193	0.983	0.787
(12, $\sigma = 7$)	0.154	0.983	0.785
(12, $\sigma = 8$)	0.124	0.984	0.787
(12, $\sigma = 9$)	0.100	0.983	0.782
(12, $\sigma = 10$)	0.084	0.983	0.780
(12, $\sigma = 15$)	0.030	0.978	0.756
(12, $\sigma = 20$)	0.011	0.967	0.716
(12, $\sigma = 25$)	0.004	0.949	0.675

Table 3.4: Results of structural comparison experiments: shown are the calculated values for Algorithms 3-5 for each of the Summation in Table 3.2, with a range of parameter-values for the two general summation-classes, the L_p and σ-threshold summations.

Summation	Unweighted	Shape	Size	Location	Spatial Depth	Contrast	ROI	IM
(3)	0.019	0.022	0.019	0.018	0.019	0.022	0.019	0.019
(4)	0.001	6.22E-4	0.002	0.002	0.001	5.34E-4	0.001	0.001
(8)	0.033	0.036	0.032	0.031	0.032	0.036	0.032	0.033
(12, $\sigma = 1$)	0.606	0.608	0.607	0.599	0.605	0.612	0.606	0.606
(12, $\sigma = 2$)	0.475	0.481	0.477	0.465	0.474	0.486	0.476	0.476
(12, $\sigma = 3$)	0.375	0.382	0.377	0.363	0.374	0.388	0.376	0.376

Table 3.5: Sample output of Algorithm 3 with visual factors maps as weights for sum, for *bird* test image: shown are the calculated values using each of the maps defined by the presence of five different visual factors (shape, size, location, spatial depth, and contrast), as well as the combined ROI map and the smoothed Importance Map (IM), for a subset of the summations from Table 3.2.

Summation	Unweighted	Shape	Size	Location	Spatial Depth	Contrast	ROI	IM
(3)	0.852	0.852	0.852	0.852	0.852	0.852	0.852	0.852
(4)	0.849	0.840	0.843	0.840	0.849	0.835	0.846	0.843
(8)	0.852	0.852	0.852	0.852	0.852	0.852	0.852	0.852
(12, $\sigma = 1$)	0.748	0.763	0.745	0.806	0.757	0.711	0.754	0.754
(12, $\sigma = 2$)	0.780	0.768	0.774	0.832	0.788	0.728	0.780	0.783
(12, $\sigma = 3$)	0.809	0.783	0.794	0.843	0.809	0.740	0.800	0.806

Table 3.6: Sample output of Algorithm 5 ($r = 3$) with visual factors maps as weights for sum, for *bird* test image: shown are the calculated values using each of the maps defined by the presence of five different visual factors (shape, size, location, spatial depth, and contrast), as well as the combined ROI map and the smoothed Importance Map (IM), for a subset of the summations from Table 3.2
.

Measure	Unweighted	IM
1	0.694	0.694
2	0.065	0.065
3 ($r = 1$)	0.213	0.213
3 ($r = 2$)	0.248	0.248
3 ($r = 3$)	0.256	0.256
3 ($r = 4$)	0.247	0.247
3 ($r = 5$)	0.242	0.242
3 ($r = 6$)	0.238	0.238
3 ($r = 7$)	0.229	0.229
3 ($r = 8$)	0.219	0.219
3 ($r = 9$)	0.210	0.210
3 ($r = 10$)	0.202	0.202

Measure	Unweighted	IM
1	0.001	0.001
2	0.976	0.977
3 ($r = 1$)	0.611	0.605
3 ($r = 2$)	0.733	0.729
3 ($r = 3$)	0.786	0.783
3 ($r = 4$)	0.812	0.810
3 ($r = 5$)	0.833	0.831
3 ($r = 6$)	0.845	0.847
3 ($r = 7$)	0.858	0.861
3 ($r = 8$)	0.868	0.871
3 ($r = 9$)	0.877	0.880
3 ($r = 10$)	0.884	0.887

Table 3.7: Results for Algorithms 3-5 for Ahumada's block roughness and edge variance measures (Section 2.2.1): shown are the calculated values for each Algorithm, both unweighted and using the Importance Map (IM) as entrywise weights. We have not included those of the other visual factors since the Block Roughness and Edge Variance summations are applied to blocks rather than to each pixel, and thus only the IM map is appropriate.

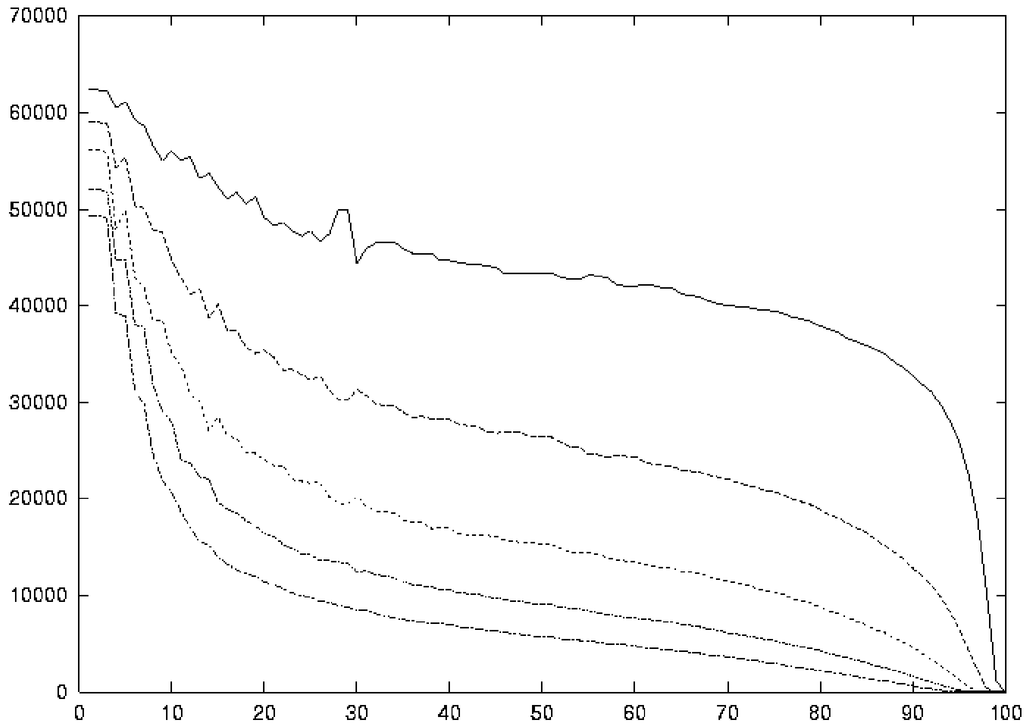

Figure 3.12: Plots of IJG quality factor (horizontal axis) vs. σ-norm, for $\sigma = 1, \ldots, 5$ (from top)

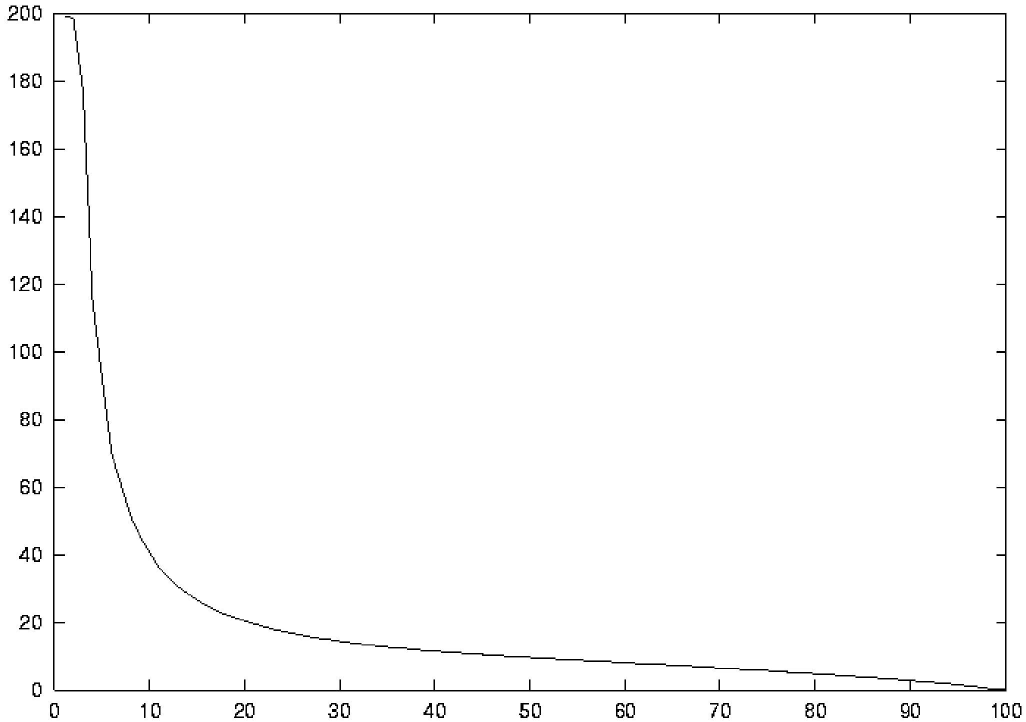

Figure 3.13: Plot of IJG quality factor (horizontal axis) vs. Mean-Squared Error

Chapter 4

Multidimensional Modelling of Visual Thresholds

When compressing files for use on the internet, one often wants to make an image as small as possible while maintaining a desired level of quality. In the ideal case, we would like a large decrease in the given quality measure after the (IJG) quality factor that corresponds to the smallest image that gives an acceptable fidelity; however, in reality the numerical differences of quality measures are quite small over the range of quality levels that provide an acceptable balance between image fidelity and decreased file-size. Therefore, we are limited to predicting thresholds by determining which of these small numerical differences correspond to large differences in observed visual fidelity.

In Chapter 2, we discussed an example of the JPEG Perceptual Dipper: an image encoded at IJG quality factor $q = 75$ is requantized at $q = 50$ and $q = 48$, with the $q = 48$ image having higher visual fidelity despite having smaller file-size. This effect is surprising, not only due to the resulting non-monotonicity of the IJG quality scale, but also because such a small change in the quality scale factors gives a highly noticeable change in visual appearance. In effect, the Dipper Effect describes a *visual threshold*, separating the high-fidelity (q48) image from the lower quality (q50) image.

The ability to model these thresholds can be very useful, as it can provide a method of dynamically choosing which image to serve to different environments. Presently, the increase in available bandwidth combined with the increased use of low-bandwidth environments (for example, web-accessible cellular phones, Palm Pilots, and other hand-held devices) results in contradictory requirements on images: high-bandwidth environments require high quality images while low-bandwidth environments require small image file-sizes. Modelling visual thresholds effectively can provide a way to choose in real-time which image to serve, by choosing the supra-threshold image of the smallest file-size.

In this chapter, we discuss an experimental environment that determines where visual thresholds occur along the IJG quality scale (Section 4.1). We also discuss *Multidimensional Scaling* (MDS), a prominent method of geometrically modelling objects based on information about the relative distances between them (Section 4.3); we also propose algorithms to use the output of MDS methods to model visual thresholds (Section 4.5.2). We finish our discussion by briefly describing some theory and techniques of multicriteria optimization (MCO) in Section 4.4 as an alternative to MDS; we then show some sample applications for modelling visual thresholds using each of MDS and MCO in Section 4.3.5, with subjective experiments to test the performance of each proposed algorithm (Section 4.5).

4.1 Visual Thresholds

As previously mentioned, the Perceptual Dipper Effect is an example of a *visual threshold*: if we partition a set of approximations to a given image into two sets of acceptable and unacceptable approximations, we consider a visual threshold to be modelled in terms of a pair of images that are close to each other in terms of some numerical measure (the IJG quality scale factor in our example) and that well distinguish between the two sets. In the case where the measure used to define the closeness of the two images is monotonic as a function of the image's filesize, the pair of images contains the image of the smallest filesize that is still of acceptable fidelity according to the given measure.

Figure 4.1 shows a screen-capture of the Packeteer `ImgStat` application[1] used to measure subjective visual thresholds. The interface consists of a scrollbar which adjusts the IJG quality factor used to compress the image, and two checkboxes used to fix the values of the IJG quality factor at which a visual threshold is reached. In-context comparisons between the images are made by mouse-clicking on the image, which toggles between the original and compressed images. The subject adjusts the scroll-bar until differences between the

[1] The screen-capture shown is an emulation of the original `ImgStat` application, written by Justin McMichael for use by Packeteer (`www.packeteer.com`). Special thanks to both Justin and Packeteer for permission to include the `ImgStat` discussion in this thesis research.

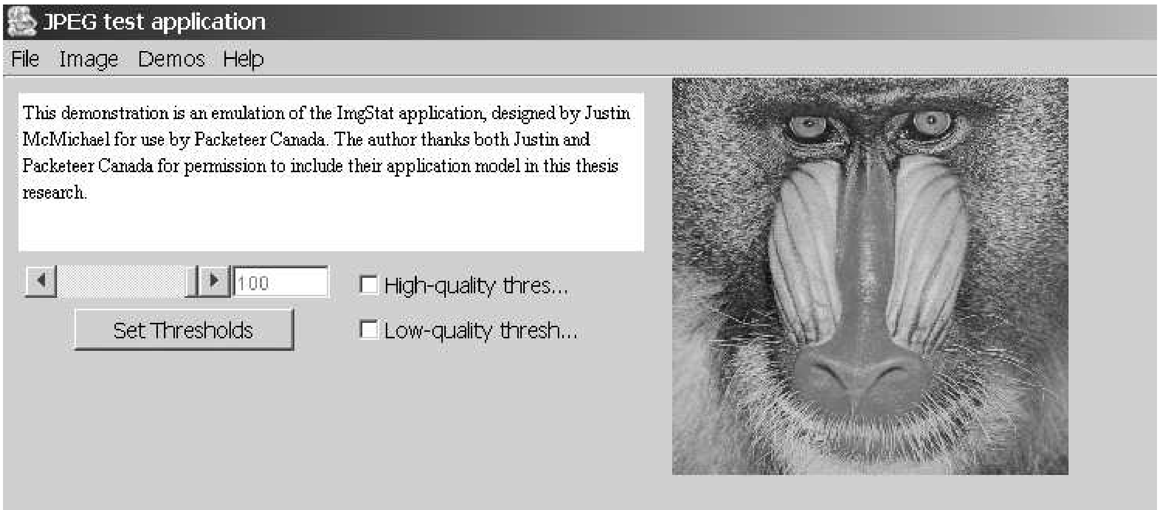

Figure 4.1: Screenshot of application designed to measure visual thresholds

original and compressed images are first noticeable, at which point the subject selects the top checkbox to indicate that the high-quality threshold has been reached. The subject further adjusts the scrollbar until differences between the two images become annoying, then selects the bottom checkbox to indicate that the low-quality threshold has been reached.

The `ImgStat` interface models the problem of varying the compressed image for high- and low-bandwidth environments as described in the beginning of this chapter, as well as conforming to the MOS rating scale discussed in Section 2.3.4; the high-bandwidth image will be the image of the lowest quality level above the high-quality threshold, while similarly the low-bandwidth image will be the image of lowest quality above the low-quality threshold. In addition, this experiment shows the general structure of visual thresholds, as well as showing how the numerical value of the threshold can change depending on the specific property being measured. In Section 4.5, we will discuss subjective experiments we performed using this interface.

4.2 Mathematical Preliminaries

In the following definitions, we denote $\mathbb{M}^{m\times n}$ as the set of all $m \times n$ matrices.

Definition 2 *For $A \in \mathbb{M}^{n\times n}$, the* trace *of A, denoted $tr(A)$, is the sum of the diagonal entries of A.*

Definition 3 *([75], Ch. 2) A square matrix $A \in \mathbb{M}^{n \times n}$ is* orthogonal *if $A^T = A^{-1}$, i.e, if $A^T A = I$.*

Definition 4 *([75], Ch. 24) For $A \in \mathbb{M}^{n \times n}$, a nonzero vector $x \in \mathbb{R}^n$ is an* eigenvector *with corresponding* eigenvalue $\lambda \in \mathbb{R}$ *if*

$$Ax = \lambda x. \tag{4.1}$$

A matrix is positive semidefinite *if all of its eigenvalues are nonnegative, and* positive definite *if they are all positive. The* eigenvalue decomposition *of A is a factorization*

$$A = X \Lambda X^{-1} \tag{4.2}$$

where $X \in \mathbb{R}^{n \times n}$ is nonsingular and $\Lambda \in \mathbb{R}^{n \times n}$ is diagonal, with the main diagonal consisting of the eigenvalues of A in nonincreasing order.

Definition 5 *([75], Ch. 23) For $A \in \mathbb{M}^{n \times n}$ symmetric positive semidefinite, the* Cholesky factorization *of A is the factorization*

$$A = R^T R, \tag{4.3}$$

where $R \in \mathbb{M}^{n \times n}$ is upper-triangular.

Definition 6 *([75], Ch. 4 and 5) For $m, n \in \mathbb{N}_+$ and $A \in \mathbb{M}^{m \times n}$, the* singular value decomposition *(SVD) of A is a factorization*

$$A = U \Sigma V^T \tag{4.4}$$

where $U \in \mathbb{M}^{m \times m}$ and $V \in \mathbb{M}^{n \times n}$ are orthogonal, and $\Sigma \in \mathbb{M}^{m \times n}$ is diagonal with nonnegative diagonal entries in nonincreasing order. (Note that we are not assuming $m = n$.)

4.3 Multidimensional Scaling

Multidimensional scaling (MDS) is a data visualization technique, where measurements of distances between a given set of objects, generally corresponding to existing objects, are used to estimate the locations of each object. The popular analogy is with a map of a country: given the map, one can measure the distances between cities to form a matrix of distances; MDS deals with the inverse problem, namely finding the (relative) locations on the map given the distances between cities [86]. MDS extends this idea beyond notions of physical location, from measures of distance to measures of difference, or the more frequently used term, *dissimilarity*; rather than considering proximities between locations, dissimilarity measures can include properties that are more difficult to measure directly. This makes it an interesting approach in the context of the visual fidelity, or visual similarity, measures discussed in Chapter 3.

The problem MDS deals with can be formulated as follows: let O be a set of n objects, with measured *dissimilarities* $\delta_{r,s}$ for $r, s \in O$; for example, these dissimilarities can be measured via subjective experiments or with a predetermined function. Let E be an n-dimensional Euclidean space with the standard metric denoted by $d : E \times E \to \mathbb{R}$. MDS constructs a mapping $\phi : O \to E$ such that for each pair of points $r, s \in O$, $d(\phi(r), \phi(s)) \approx \delta_{r,s}$. In other words, MDS constructs a set of *object points* $\{\phi(x)\}_{x \in O}$ that structurally approximates a real-life set of objects, in an easily visualized space.

Historically, multidimensional scaling has been divided into two areas: *metric MDS*, where the dissimilarity measure is continuous and the magnitudes of the dissimilarities has a physical interpretation; and *nonmetric scaling*, where the dissimilarity measure is discrete, and only the rank order of the dissimilarities is of interest. We discuss definitions used in multidimensional scaling methods in Section 4.3.1, and illustrate algorithms for metric and nonmetric MDS in Sections 4.3.2 and 4.3.3, respectively. Much of the literature on MDS neglects the theory behind the algorithms, instead referring to the use of software to solve these problems. As such, our discussion focusses on the mathematical theory behind these algorithms, collecting many of the original discussions used to motivate these algorithms. Many of the following algorithms and results can be found in [13, 19, 34, 41, 65, 66, 74, 86].

4.3.1 Definitions

Continuing the discussion of the previous section, the components of the formulation of multidimensional scaling are as follows:

- a set O of n *objects*, each of which represents an existing physical object, person, location, organization, etc.;
- an $N \times N$ matrix of *dissimilarities* $[\delta_{r,s}]_{(r,s) \in O \times O}$, where $\delta_{r,s}$ represents the dissimilarity of object r to object s; for example, these dissimilarities can be acquired by performing subjective experiments or by performing physical measurements using a given function.

Note that in general, the dissimilarity matrix will be symmetric; in fact, quite often only the upper- or lower-triangular entries of the matrix will be measured, with the remaining entries defined assuming symmetry. However, there has been some research on how to handle non-symmetric matrices; see [19] for a brief overview. In our discussion, we will be assuming a symmetric dissimilarity matrix.

Given O and $[\delta_{r,s}]$, the goal of MDS is to find a mapping $\phi : O \to \mathbb{R}^m$ such that for a given distance measure $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$,

$$d(\phi(r), \phi(s)) \approx \delta_{rs}, \tag{4.5}$$

i.e., the set of *distances* $[d(\phi(r), \phi(s))]_{(r,s)\in O\times O}$ between the points in the image of the set of objects approximates the original set of observed dissimilarities between the objects. We represent the points $[\phi(r)]_{r\in O}$ as a *configuration* $X = [x_{ri}] \in \mathbb{M}^{n\times m}$, where the elements $x_{ri},\ i = 1, \ldots, m$ are the coordinates of $\phi(r) \in \mathbb{R}^m$.

4.3.2 Torgerson

Multidimensional scaling was first proposed by Torgerson [74] to extend scaling along a single-dimension, using a result due Young and Householder [87]. This model required the dissimilarities to be defined by exact Euclidean distances. Torgerson's algorithm to construct the object points is described in the following result, originally in [74] and [87]; our proof follows the discussion of [19] and [85].

Given a matrix of dissimilarities $D = [d_{r,s}]_{r,s=1}^n$, we want to construct a configuration $X = [x_{r,i}]_{r=1,i=1}^{N,N}$ such that $d_{r,s} = \sqrt{(x_r - x_s)^T(x_r - x_s)}$. Since the distances between points are not affected by arbitrary translations of the entire configuration, assume that the centroid of the configuration is located at the origin, so that for all $i = 1, \ldots, N$,

$$\sum_{r=1}^{n} x_{ri} = 0. \tag{4.6}$$

Since $d_{rs}^2 = x_r^T x_r + x_s^T x_s - 2x_r^T x_s$, then

$$\begin{aligned}
\frac{1}{n}\sum_{r=1}^{n} d_{rs}^2 &= \frac{1}{n}\sum_{r=1}^{n}\left(x_r^T x_r + x_s^T x_s - 2x_r^T x_s\right) &(4.7)\\
&= \frac{1}{n}\sum_{r=1}^{n}(x_r^T x_r) + x_s^T x_s - \frac{2}{n}\sum_{r=1}^{n}\sum_{i=1}^{p} x_{ri}x_{si} &(4.8)\\
&= \frac{1}{n}\sum_{r=1}^{n}(x_r^T x_r) + x_s^T x_s - \frac{2}{n}\sum_{i=1}^{p} x_{si}\sum_{r=1}^{n} x_{ri} &(4.9)\\
&= \frac{1}{n}\sum_{r=1}^{n}(x_r^T x_r) + x_s^T x_s. &(4.10)
\end{aligned}$$

Similarly,

$$\frac{1}{n}\sum_{s=1}^{n} d_{rs}^2 = \frac{1}{n}\sum_{s=1}^{n}(x_s^T x_s) + x_r^T x_r. \qquad (4.11)$$

Performing the summation over both indices r and s gives

$$\begin{aligned}
\frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^2 &= \frac{1}{n}\sum_{r=1}^{n}\left(\frac{1}{n}\sum_{s=1}^{n} x_s^T x_s + x_r^T x_r\right) &(4.12)\\
&= \frac{1}{n}\sum_{r=1}^{n} x_r^T x_r + \frac{1}{n}\sum_{s=1}^{n} x_s^T x_s &(4.13)\\
&= \frac{2}{n}\sum_{r=1}^{n} x_r^T x_r. &(4.14)
\end{aligned}$$

Solving for $b_{rs} = x_r^T x_s$ from the definition of d_{rs} using the above expressions gives us

$$\begin{aligned}
b_{rs} &= \frac{-1}{2}\left(d_{rs}^2 - x_r^T x_r - x_s^T x_s\right) &(4.15)\\
&= \frac{-1}{2}\left(d_{rs}^2 - \frac{1}{n}\sum_{v=1}^{n} d_{rv}^2 + \frac{1}{n}\sum_{v=1}^{n} x_v^T x_v - \frac{1}{n}\sum_{u=1}^{n} d_{us}^2 + \frac{1}{n}\sum_{u=1}^{n} x_u^T x_u\right) &(4.16)\\
&= \frac{-1}{2}\left(d_{rs}^2 - \frac{1}{n}\sum_{s=1}^{n} d_{rs}^2 - \frac{1}{n}\sum_{r=1}^{n} d_{rs}^2 + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^n\right). &(4.17)
\end{aligned}$$

Claim: If $B = [b_{rs}]$ is positive semidefinite, then there exists a configuration there exists a configuration $X = [x_{r,i}]_{r=1,i=1}^{N,N}$ such that $d_{r,s} = \sqrt{(x_r - x_s)^T(x_r - x_s)}$.

To prove the claim, consider the matrix $X = B^{1/2}$, which exists since B is positive semidefinite; for example, for the eigenvalue decomposition $B = U^{\Lambda}U^{-1}$, $B^{1/2} = U\Lambda^{1/2}$, where $\Lambda^{1/2}$ is the diagonal matrix with entries equal to the square-root of the diagonal entries of Λ^2. Then B is the matrix of inner products of the rows of X, ie., $b_{rs} = x_r^T x_s$ where x_r is the rth row of X. Working backwards from our original definition of $b_{rs} = \frac{-1}{2}\left(d_{rs}^2 - x_r^T x_r - x_s^T x_s\right)$, we have

$$\begin{aligned} & x_r^T x_s = \frac{-1}{2}\left(d_{rs}^2 - x_r^T x_r - x_s^T x_s\right) \\ \Rightarrow\ & -2x_r^T x_s + x_r^T x_r + x_s^T x_s = d_{rs}^2 \\ \Rightarrow\ & (x_r - x_s)^T (x_r - x_s) = d_{rs}^2 \\ \Rightarrow\ & d_{rs} = \sqrt{(x_r - x_s)^T (x_r - x_s)}. \end{aligned}$$

Therefore, the matrix $X = B^{1/2}$ provides us with a set of n points in $\mathbb{R}^N$ that preserve the distances $\{d_{rs}\}$. $\square$

If we want to move our configuration into a smaller dimensional space (preferably $\mathbb{R}^2$ or $\mathbb{R}^3$), we can approximate our configuration X by calculating the singular value decomposition of $X = UDV^T$, and defining our approximation to be the matrix $\hat{X} = U\hat{D}V^T$, where $\hat{D}$ is the diagonal matrix with the 2 (or 3) largest singular values along its diagonal.

The above derivation is legitimate if the inner-product matrix B is positive semidefinite. If D is not positive semidefinite, then the recommended heuristic approaches are:

- instead of using the Cholesky decomposition of B to define the configuration matrix X, calculate the eigenvalue decomposition $B = V\Lambda V^T$, and ignore the negative eigenvalues by defining $X = V\hat{\Lambda}^{1/2}$, where $\hat{\Lambda}$ is the diagonal matrix with the 2 (or 3) largest positive eigenvalues on the diagonal, with the remaining diagonal elements equal to 0; or

- add a sufficient scalar multiple of the identity matrix to make $\hat{B} = B + \delta I$ positive semidefinite (for example, the negative of the smallest eigenvalue of B), and follow all remaining steps.

However, in practice the dissimilarities will not be exact Euclidean distances, and will instead be obtained by subjective experiments; thus, the dissimilarities will either contain

[2]In [19, 74], the Cholesky decomposition of B is used for $B^{1/2}$. However, in practice we will calculate the eigenvalue decomposition of the inner product matrix B, since we gain nothing in terms of our configuration by the upper-triangular property of the Cholesky decomposition of B.

errors or be categorical/discrete variables. Therefore, a configuration that estimates the measured dissimilarities will have to be approximated rather than determined exactly using the above approach.

4.3.3 Shepard and Kruskal

The method of Torgerson described in the previous section is an example of *metric scaling*, where the dissimilarity values are taken from a continuous scale, and where the numerical value has a physical interpretation. In most psychophysical experiments, the dissimilarity data is not continuous, but instead is categorical/discrete. For example, a common technique in conducting subjective experiments is to have subjects choose similarity ratings from a discrete set of values, which may be labelled with text interpretations for each value; we encountered this with the MOS rating scale used in the experiments discussed in Section 2.3.4, and shall encounter it again in Section 4.5. The need to model such dissimilarity data led to *nonmetric scaling*, first proposed by Shepard in 1962 ([65, 66]) and Kruskal in 1964 ([35, 36]), the central proposition of which was to find a configuration whose distances maintain the rank ordering of the measured dissimilarities. Shepard first outlined a general approach emphasizing the use of the rank ordering of the dissimilarities, while Kruskal's work quantifies Shepard's discussion.

In the formulation of Kruskal's original algorithm, he proposed two additional definitions: for a given set of dissimilarities $\{\delta_{rs}\}$ and a configuration $\{x_r\}$ with distances $\{d_{rs}\}$ defined by $d_{rs} = f(x_r, x_s)$ for a given measure f, the *disparities* $\left\{\hat{d}_{rs}\right\}$ are defined to be any set of values that maintains the rank ordering of the dissimilarities; and the *stress* of the configuration is defined as

$$S = \sqrt{\frac{\sum_{r,s}\left(d_{rs} - \hat{d}_{rs}\right)^2}{\sum_{r,s} d_{rs}^2}}. \tag{4.18}$$

The stress of a configuration is a measure of similarity between the rank ordering of the given configuration and the original dissimilarities.

With the above definitions, and for a given set of dissimilarities $\{\delta_{rs}\}$, Kruskal's original algorithm is as follows [19, 35, 36, 85]):

Algorithm 6 *For an initial configuration $X^0 = \left\{x_r^0\right\}$:*

1. *Normalize the current configuration X^i such that the centroid is located at the origin, and the mean distance from the configuration to the origin is 1.*

2. *Calculate the distances* $\{d_{rs}\}$ *for the current configuration.*

3. *Calculate the disparities* $\left\{\hat{d}_{rs}\right\}$ *that maintain the rank ordering of the dissimilarities while minimizing* S.

4. *Adjust the current configuration to fit the dissimilarities, according to the formula*

$$x_{rj}^{i+1} = x_{rj}^{i} - \alpha \frac{\frac{\partial S}{\partial x_{rj}}}{\left|\frac{\partial S}{\partial x_{rj}}\right|}, \tag{4.19}$$

   where $\alpha > 0$ *is the chosen stepsize.*

5. *Check whether the configuration has converged by checking whether the stationary point condition* $\nabla S = 0$ *holds; if not, then normalize the configuration and repeat process.*

We now derive each of the steps in the above algorithm, specifically the method of finding the disparities that minimize stress for a given set of distances, and calculating the partial derivatives of S using a closed form expression.

Derivation of disparities of minimum stress

For any set of points $\{d_i\}_{i=1}^{N}$, define $D_n = \sum_{i=1}^{n} d_i$ for $n = 1, \ldots, N$, with $D_0 = 0$. Then a *convex minorant* of the finite set of values $\{D_n\}_{n=0}^{N}$ is the set $\{\hat{D}_n\}_{n=1}^{N}$ satisfying the following properties:

- $\hat{D}_0 = D_0$, $\hat{D}_N = D_N$, and $\hat{D}_i \leq D_i, i = 1, \ldots, N$;
- $i \geq j \Leftrightarrow \hat{D}_i - \hat{D}_{i-1} \geq \hat{D}_j - \hat{D}_{j-1}$.

The *greatest convex minorant* is the convex minorant with the largest values for each $\hat{D}_i$. Using these definitions, we get the following result [19, 35, 36]:

Theorem 1 *(Kruskal, 1964) Given* $\delta = [\delta_{rs}]$*, define* δ^* *to be the vector consisting of the values of* δ_{rs} *in increasing order, and define* d^* *to be the values of* d_{rs} *in the same reordering as* δ^**.Define* $D_n = \sum_{i=1}^{n} d_i^*$*. Then the disparities* $\hat{d}_i$ *that minimize* S *are*

$$\hat{d}_i = \hat{D}_i - \hat{D}_{i-1} \ (i = 1, \ldots, N) \tag{4.20}$$

with $D_0 = \hat{D}_0 = 0$*, where* $\hat{D}$ *is the greatest convex minorant of* D.

Proof: To prove that the greatest convex minorant minimizes S, consider any vector $\bar{d}$ whose values maintain the rank ordering of δ^*. Then

$$\sum_{i=1}^{n}(d_i-\bar{d}_i)^2 = \sum_{i=1}^{n}\left[(d_i-\hat{d}_i)+(\hat{d}_i-\bar{d}_i)\right]^2 \tag{4.21}$$

$$= \sum_{i=1}^{n}(d_i-\hat{d}_i)^2+\sum_{i=1}^{n}(\hat{d}_i-\bar{d}_i)^2+2\sum_{i=1}^{n}(d_i-\hat{d}_i)(\hat{d}_i-\bar{d}_i). \tag{4.22}$$

Consider $\sum_{i=1}^{n}(d_i-\hat{d}_i)(\hat{d}_i-\bar{d}_i)$. Abel's summation [63] says that

$$\sum_{i=1}^{n}a_ib_i=\sum_{i=1}^{n-1}A_i(b_i-b_{i+1})+A_nb_n \tag{4.23}$$

where $A_i=\sum_{j=1}^{i}a_j$. Applying this to $\sum_{i=1}^{n}(d_i-\hat{d}_i)(\hat{d}_i-\bar{d}_i)$ gives

$$\begin{aligned}\sum_{i=1}^{n}(d_i-\hat{d}_i)(\hat{d}_i-\bar{d}_i) &= \sum_{i=1}^{n-1}\left[(D_i-\hat{D}_i)(\hat{d}_i-\bar{d}_i-\hat{d}_{i+1}+\bar{d}_{i+1})\right] \\ &\qquad + (D_N-\hat{D}_N)(\hat{d}_N-\bar{d}_N) \\ &= \sum_{i=1}^{n-1}(D_i-\hat{D}_i)(\hat{d}_i-\hat{d}_{i+1})-\sum_{i=1}^{n-1}(D_i-\hat{D}_i)(\bar{d}_i-\bar{d}_{i+1})\end{aligned}$$

since $D_N=\hat{D}_N$. Furthermore, $(D_i-\hat{D}_i)(\bar{d}_i-\bar{d}_{i+1})=0$ for all i, since if $D_i>\hat{D}_i$ then $\hat{D}_i$ is a convex combination of $\hat{D}_{i+1}$ and $\hat{D}_{i-1}$, or

$$\begin{aligned}\hat{D}_i=\frac{\hat{D}_{i+1}+\hat{D}_{i-1}}{2} &\Rightarrow \hat{D}_i-\hat{D}_{i-1}=\hat{D}_{i+1}-\hat{D}_i \\ &\Rightarrow \hat{d}_i=\hat{d}_{i+1}.\end{aligned}$$

Therefore,

$$\sum_{i=1}^{n}(d_i-\hat{d}_i)(\hat{d}_i-\bar{d}_i) = -\sum_{i=1}^{n-1}(D_i-\hat{D}_i)(\bar{d}_i-\bar{d}_{i+1}). \tag{4.24}$$

However, $D_i\geq\hat{D}_i$ and $\bar{d}_i\leq\bar{d}_{i+1}$ (since $\bar{d}$ maintains the rank ordering of δ), and thus $-\sum_{i=1}^{n-1}(D_i-\hat{D}_i)(\bar{d}_i-\bar{d}_{i+1})\geq 0$.

Therefore,

$$\begin{aligned}\sum_{i=1}^{n}(d_i-\bar{d}_i)^2 &= \sum_{i=1}^{n}(d_i-\hat{d}_i)^2+\sum_{i=1}^{n}(\hat{d}_i-\bar{d}_i)^2+2\sum_{i=1}^{n}(d_i-\hat{d}_i)(\hat{d}_i-\bar{d}_i) && (4.25)\\ &\geq \sum_{i=1}^{n}(d_i-\hat{d}_i)^2+\sum_{i=1}^{n}(\hat{d}_i-\bar{d}_i)^2 && (4.26)\\ &\geq \sum_{i=1}^{n}(d_i-\hat{d}_i)^2, && (4.27)\end{aligned}$$

and thus the greatest convex minorant $\hat{d}_i$ minimizes S. □

The following algorithm describes how to find the disparities $\hat{d}=\left[\hat{d}_{rs}\right]$ that minimize S for the current configuration; note that this algorithm implements Step 3 of Algorithm 6.

Algorithm 6.1 *For a given configuration* $\{x_r\}$ *and a given set of dissimilarities* $\delta=[\delta_{rs}]$*:*

1. *Calculate the distances* $d=[d_{rs}]$ *for the current configuration.*

2. *Reorder the dissimilarities* $[\delta_{rs}]$ *into a vector* δ^* *with nondecreasing entries.*

3. *Define a vector* d^* *to have the same reordering as* δ^**, and define the partial sums* $D_i:=\sum_{j=1}^{i}d_j^*$.

4. *Find the greatest convex minorant* $\hat{D}$ *of* D*, and define the disparity vector* $\hat{d}$ *as* $\hat{d}_i:=\hat{D}_i-\hat{D}_{i-1}$.

5. *Read the disparities into a disparity matrix* $\hat{D}:=\left[\hat{d}_{rs}\right]$ *using the indices defining each corresponding entry in the vectors* d^* *and* δ^*.

Derivation of configuration adjustment

For a given set of dissimilarities, the configuration at each step must be updated in order to improve the approximation of the inter-point distances to the dissimilarities. We update the points using the steepest descent method, which minimizes a given function f by searching along the direction $-\nabla f(x_k)$ from the current iterate x_k; in our case, the function we want to minimize is $S(\mathbf{x})$, and our current iterate is our present configuration.

For our current configuration $\mathbf{x}_k$, we define our adjusted configuration to be

$$\mathbf{x}_{k+1}=\mathbf{x}_k-\Delta\nabla S(\mathbf{x}_k),$$

for $\Delta > 0$ a stepsize[3] chosen to give $S(\mathbf{x}_{k+1}) < S(\mathbf{x}_k)$. However, note that S is defined as a function of the distances, and for $\nabla S(x_k)$ we need to evaluate the partial derivatives $\frac{\partial S}{\partial \mathbf{x}_{he}}$. We do this in a two-stage application of the chain rule. First,

$$\begin{aligned}
\frac{\partial S^2}{\partial \mathbf{x}_{he}} &= \frac{1}{\sum_{i,j} d_{ij}^2} \frac{\partial \left(\sum_{i,j} (d_{ij} - \hat{d}_{ij} \right)^2)}{\partial x_{he}} && (4.28) \\
&= \frac{2}{\sum_{i,j} d_{ij}^2} \sum_i (d_{ih} - \hat{d}_{ih}) \frac{\partial d_{ih}}{\partial x_{he}} && (4.29)
\end{aligned}$$

gives us an expression of the desired partial derivatives in terms of the partial derivatives of the distances $\frac{\partial d_{ih}}{\partial x_{he}}$.

Second, we need to evaluate $\frac{\partial d_{ih}}{\partial x_{he}}$, which will obviously depend on the metric space X within which our configuration is located; we will treat the case where $X = l_p^n$, and thus

$$d_{ih} = \left[\sum_{j=1}^{n} |x_{ij} - x_{hj}|^p \right]^{1/p}. \qquad (4.30)$$

Therefore,

$$\begin{aligned}
\frac{\partial d_{ih}}{\partial x_{he}} &= \frac{\partial \left(\left[\sum_j |x_{ij} - x_{hj}|^p \right]^{1/p} \right)}{\partial x_{he}} && (4.31) \\
&= \frac{1}{p} \left[\sum_{j=1}^{n} |x_{ij} - x_{hj}|^p \right]^{1/p-1} \frac{\partial \left(\sum_{i,j} |x_{ij} - x_{hj}|^p \right)}{\partial x_{he}} && (4.32) \\
&= \frac{1}{p} d_{ih}^{1-p} \left(\sum_{j=1}^{n} \frac{\partial \left(|x_{ij} - x_{hj}|^p \right)}{\partial x_{he}} \right) && (4.33) \\
&= d_{ih}^{1-p} \left(\sum_{j=1}^{n} |x_{ij} - x_{hj}|^{p-1} \frac{\partial \left(|x_{ij} - x_{hj}| \right)}{\partial x_{he}} \right) && (4.34) \\
&= d_{ih}^{1-p} |x_{ie} - x_{he}|^{p-1} \mathrm{signum}(x_{ie} - x_{he}) \frac{\partial (x_{ie} - x_{he})}{\partial x_{he}} && (4.35) \\
&= -d_{ih}^{1-p} |x_{ie} - x_{he}|^{p-1} \mathrm{signum}(x_{ie} - x_{he}). && (4.36)
\end{aligned}$$

[3]The specific heuristic used by Kruskal to update the stepsize is not included here, although we have used it in our implementation in Appendix B.1. Although we have not done so, it would be interesting to compare the results of this heuristic to stepsizes chosen by a traditional backtracking method using the Armijo-Goldstein-Wolfe conditions [47], to see if smaller local minima can be found consistently.

Substituting this back into $\frac{\partial S^2}{\partial \mathbf{x}_{he}}$ gives

$$\frac{\partial S^2}{\partial \mathbf{x}_{he}} = \frac{-2}{\sum_{i,j} d_{ij}^2} \left[\sum_i \frac{d_{ih} - \hat{d}_{ih}}{d_{ih}^{p-1}} |x_{ie} - x_{he}|^{p-1} \text{signum}(x_{ie} - x_{he}) \right]. \tag{4.37}$$

In our implementation described in Appendix B.1, we use this expression for $\nabla S^2(x_k)$ in the configuration adjustment stage of Kruskal's algorithm. Although using this form fixes our problem to using an l_p^n metric space, this class of spaces is general enough for our purposes to not focus on this limitation; note that for another metric space, we would simply evaluate ∇d and continue our above argument at (4.31).

Remarks

One concern with Kruskal's algorithm is that it lacks convergence results, partly due to the heuristic method in which the stepsize is chosen at each step, although Kruskal has proven partial results in [37]. Since Kruskal's original formulation of the above algorithm, other more advanced algorithms have expanded on Kruskal's model, and specifically on its goal of minimizing stress. Two frequently used algorithms are the ALSCAL (*Alternating Least-squares Scaling*) algorithm of Takane, Young and de Leeuw [85], and the SMACOF (*Scaling by Majorizing a Complicated Function*) algorithm of de Leeuw [21], which uses a more general polynomial minimization approach, and which has convergence results for both metric and nonmetric MDS. Although discussions for both methods fall outside the scope of this thesis, software for each of ALSCAL and SMACOF is publicly available, and is used in the experimental results discussed in Section 4.5.

4.3.4 Procrustes Analysis

Given two configurations $X = [x_r] \in \mathbb{R}^{p \times n}$ and $Y = [y_r] \in \mathbb{R}^{p \times n}$ representing the same set of n objects, *Procrustes Analysis* is a method of quantifying the degree of similarity between the two configurations; for example, the two configurations could either differ in the measure used to generate the original dissimilarities $\{\delta_{rs}\}$, or the method used to calculate the configuration using a fixed measure. This process consists of finding (ρ, A, b) such that the transformed configuration defined by

$$x'_r = \rho A^T x_r + b \tag{4.38}$$

minimizes

$$R^2 = \sum_{r=1}^{n} (y_r - x'_r)^T (y_r - x'_r), \tag{4.39}$$

where $\rho \in \mathbb{R}$ is the *dilation factor*, $A \in \mathbb{M}^{p \times p}$ is an orthogonal matrix representing the linear transformation of the configuration, and $b \in \mathbb{R}^p$ is the *translation factor*. We solve this by finding separately each of ρ, b, and A; the following discussion follows [19].

(i) To find the optimal b, consider each configuration relative to their centroids

$$x_0 = \sum_{r=1}^{n} x_r \quad \text{and} \quad y_0 = \sum_{r=1}^{n} y_r, \tag{4.40}$$

so that

$$\begin{aligned} R^2 &= \sum_{r=1}^{n} ((y_r - y_0) - \rho A^T (x_r - x_0) + y_0 - \rho A^T x_0 - b)^T \\ &\qquad ((y_r - y_0) - \rho A^T (x_r - x_0) + y_0 - \rho A^T x_0 - b) \\ &= \sum_{r=1}^{n} ((y_r - y_0) - \rho A^T (x_r - x_0))^T ((y_r - y_0) - \rho A^T (x_r - x_0)) \\ &\qquad + n(y_0 - \rho A^T x_0 - b)^T (y_0 - \rho A^T x_0 - b). \end{aligned}$$

The minimum of R^2 will occur when $b = y_0 - \rho A^T x_0$, which gives $x'_r = \rho A^T x_r + y_0 - \rho A^T x_0$. Therefore,

$$\begin{aligned} \frac{1}{n} \sum_{r=1}^{n} x'_r &= \frac{1}{n} \sum_{r=1}^{n} \rho A^T x_r + y_0 - \rho A^T x_0 && (4.41) \\ &= y_0 - \rho A^T x_0 + \frac{1}{n} \sum_{r=1}^{n} \rho A^T x_r && (4.42) \\ &= y_0 - \frac{1}{n} \sum_{r=1}^{n} \rho A^T x_r + \frac{1}{n} \sum_{r=1}^{n} \rho A^T x_r && (4.43) \\ &= y_0, && (4.44) \end{aligned}$$

and thus the centroid of the transformed coefficients of X is equal to the centroid of Y. (This condition is easily satisfied by centering both coefficients at the origin prior to the dilation and transformation steps.)

(ii) : To find the optimal ρ, consider the original expression for R^2, with $x_0 = y_0 = 0$:

$$\begin{aligned} R^2 &= \sum_{r=1}^{n}(y_r - \rho A^T x_r)^T(y_r - \rho A^T x_r) && (4.45) \\ &= \sum_{r=1}^{n} y_r^T y_r + \rho^2 \sum_{r=1}^{n} x_r^T x_r - 2\sum_{r=1}^{n} \rho x_r^T A y_r && (4.46) \\ &= \mathrm{tr}(YY^T) + \rho^2 \mathrm{tr}(XX^T) - 2\rho \mathrm{tr}(XAY^T). && (4.47) \end{aligned}$$

Minimizing with respect to ρ gives

$$\begin{aligned} \frac{\partial R^2}{\partial \rho} &= 2\rho \mathrm{tr}(XX^T) - 2\mathrm{tr}(XAY^T) \;=\; 0 && (4.48) \\ &\Rightarrow \quad \rho = \frac{\mathrm{tr}(XAY^T)}{\mathrm{tr}(XX^T)}. && (4.49) \end{aligned}$$

(iii) : To find the optimal transformation A, since $R^2 = \mathrm{tr}(YY^T) + \rho^2\mathrm{tr}(XX^T) - 2\rho\mathrm{tr}(XAY^T)$, we want to maximize $\mathrm{tr}(XAY^T) = \mathrm{tr}(AY^TX)$. We begin by calculating the SVD of Y^TX, so that

$$Y^TX = U\Lambda V^T \tag{4.50}$$

where U and V are orthogonal and Λ is diagonal. Then

$$\begin{aligned} \mathrm{tr}(AY^TX) &= \mathrm{tr}(AU\Lambda V^T) && (4.51) \\ &= \mathrm{tr}(V^TAU\Lambda). && (4.52) \end{aligned}$$

Since V, A,and U are all orthogonal, then so is V^TAU; furthermore, since the maximum entry in an orthogonal matrix is 1, then $\mathrm{tr}(V^TAU\Lambda) \leq \mathrm{tr}(\Lambda)$. Therefore, the maximum of $\mathrm{tr}(AY^TX)$ occurs when $V^TAU\Lambda = \Lambda$. We can now solve for A directly as follows:

$$\begin{aligned} V^TAU\Lambda &= \Lambda \\ \Rightarrow \quad AU\Lambda V^T &= V\Lambda V^T \\ \Rightarrow \quad AY^TX &= V\Lambda V^T \\ &= (V\Lambda^2V^T)^{(1/2)} \\ &= (V\Lambda U^TU\Lambda V^T)^{(1/2)} \\ &= ((Y^TX)^T(Y^TX))^{(1/2)} \\ &= (X^TYY^TX)^{(1/2)} \\ \Rightarrow \quad A &= (X^TYY^TX)^{(1/2)}(Y^TX)^{-1}. \end{aligned}$$

Therefore, for configurations X and Y, normalized such that the centroid of both is located at the origin, the values of (ρ, A) that minimize R^2 are

$$(\rho, A) = \left(\frac{\operatorname{tr}\left((X^T Y Y^T X)^{(1/2)}\right)}{\operatorname{tr}(XX^T)}, (X^T Y Y^T X)^{(1/2)} (Y^T X)^{-1} \right). \qquad (4.53)$$

□

Note that this derivation of (ρ, A) assumes that the configurations are matched such that the points representing each of the n objects are matched in the same row. If the original objects corresponding to each object are not known, then the optimal match can still be estimated by calculating (ρ, A) separately for each permutation of the indices of one of the configurations, and using the configuration with the smallest value of R^2 as the optimal transformed configuration. Although this exhaustive process can quickly become computationally expensive as the dimension of the configuration increases, the fact that MDS is broadly used to more easily visualize data indicates that most configurations are already either 2 or 3 dimensional. The increase in computations resulting from an increase in the number of objects being modelled is of more concern, and is likely not avoidable in general. However, this is not a concern in our present discussion, since we have full control over the order of the objects within each of our configurations.

4.3.5 Application to Image Fidelity Measures

If we treat as our object set O the set of images created by quantizing an original image using various IJG quality factors, multidimensional scaling can be used to further evaluate the structural similarities between a given pair of image fidelity measures $f : O \times O \to \mathbb{R}$ and $g : O \times O \to \mathbb{R}$. It can also be used to measure the ability of a given measure to capture observed visual thresholds, despite the earlier observation that for most measures there are only small numerical differences over the range of quality factors of interest.

We can model the structural similarity of a pair of measures as outlined in the following algorithm:

Algorithm 7 *For a given pair of measures f and g:*

1. *For an original image I, calculate $f(I_m, I_n)$ and $g(I_m, I_n)$ for $m, n = 1, \ldots, 100$, where I_n is the JPEG-encoded image, quantized to IJG quality factor n.*

2. *Calculate configurations X and Y using as the dissimilarity matrices $[f(I_m, I_n)]$ and $[g(I_m, I_n)]$, respectively, using any of the methods described in Section 4.3.*

3. *Apply Procrustes Analysis to find the optimal transformation of X, as described in Section 4.3.4; the resulting value of R^2 is a measure of the structural similarity of the two measures f and g.*

To measure the ability of a single measure to capture observed visual thresholds, recall that a visual threshold is defined in terms of two sets of images, namely sub-threshold and supra-threshold images, which are separated by the observed threshold. We can model visual thresholds as follows:

Algorithm 8 *For a single measure f, a given set of images O, and an observed threshold σ:*

1. *Define S = the set of all sub-threshold images in O, and $\bar{S}$ = the set of all supra-threshold images in O.*

2. *Define a function $Q : O \rightarrow \{0, 1\}$ by*

$$Q(I) = \begin{cases} 0, & I \text{ is supra-threshold;} \\ 1, & I \text{ is sub-threshold.} \end{cases} \tag{4.54}$$

3. *Define a dissimilarity matrix $\Delta = [\delta_{rs}]$ by*

$$\delta_{rs} = |Q(I_r) - Q(I_s)| \tag{4.55}$$

   for $I_r, I_s \in O$.

4. *Define a second dissimilarity matrix $\Delta^* = [f(I_r, I_s)]_{I_r, I_s \in O}$. Using any of the methods discussed in Section 4.3.3, construct configurations X and Y for Δ and Δ^*, respectively.*

5. *Apply Procrustes Analysis to find the optimal transformation of X, as described in Section 4.3.4; the resulting value of R^2 is a measure of the ability of f to model the observed visual threshold σ.*

Notice that for the dissimilarity matrix Δ, there is an obvious configuration of minimum stress, namely where the configuration consists of two distinct points, one each for all of the sub-threshold and supra-threshold images. Therefore, matching the configuration defined using f for the dissimilarities to this optimal configuration provides a method of quantifying the degree to which f clusters the sub-threshold and supra-threshold together. We return to both of these algorithms in the context of the subjective experiments in Section 4.5.

4.4 Multicriteria Optimization

The approach of Multidimensional Scaling (MDS) discussed in Section 4.3 extracts structure from existing data; the data objects are represented by points in a "visible" space (such as $\mathbb{R}^2$ or $\mathbb{R}^3$), with additional structure derived from finding axes that reflect the layout of the points. Two papers on the use of MDS in modelling image quality (not fidelity) defend their use of MDS on the following central observation: when measuring the quality of an image based on the presence of a single "simple" artifact, there is a high correlation in the observations across different subjects; however, when measuring the quality of an image based on the presence of multiple "simple" artifacts, there is a lower correlation due to differences in the way each subject weights each simple artifact in the combined complex artifact [4, 41]. Therefore, for a given measure that does not model a specific type of simple visible artifact, MDS is used to extract components that correspond to the weighted amount of each simple artifact in the original measure.

However, this approach is redundant if the given measure already models a simple artifact; scaling each object in multiple dimensions is not necessary since a single axis corresponding to the simple artifact being measured would suffice. If our given measures each model a specific type of simple artifact, then we can model the general problem of fidelity as a weighted combination of distinct artifacts, with the weights varying across different subjects. This approach can be performed using *multicriteria optimization* (MCO) (or vector optimization), which adds structure to the problem by combining multiple "simple" measures into a vector-valued measure, while maintaining the distinction between each measure. Using multicriteria optimization, our approach is to measure each simple artifact separately and construct a set of *potentially optimal* images, with different subjects choosing different preferred images from this set. Looking at this approach a slightly different way, we are translating the problem of a general fidelity measure to the language of decision analysis, with the decision in question one of choosing an image of smallest filesize with sufficient fidelity, where fidelity is modelled as a combination of measures of simple artifacts.

We begin our discussion with a case study of an image fidelity measure which uses this approach of modelling different artifacts separately (Section 4.4.1); we follow this discussion with a summary of general multicriteria optimization definitions and theory, as well as algorithms to construct the set of potentially optimal images (Section 4.4.3)). Finally, we conclude our discussion with an application of multicriteria optimization to modelling visual thresholds (Section 4.4.4).

4.4.1 Case Study: Picture Quality Scale (PQS)

In Chapter 3, we described two classes of measures, based on models of the HVS and visual attention, and described how each class of measures is inherently a vector-valued measure: the HVS-based measures partitioned the frequency plane into frequency windows, and modelled image fidelity separately within each window; and the visual attention models partitioned the image into objects, and measured separately the presence of each of a set of properties that attract visual attention.

Both of these classes of models defined an error-pooling stage, in which each of the separate quality factors was combined to form a single *perceptual map*. One motivation for the error pooling stage is that the compression design problem implicitly balances two contradictory requirements, namely low bitrate and high visual fidelity, and thus the fidelity measure is implicitly used in the form of a scalar-valued optimization problem in which the optimal approximation of a given image is that which achieves the minimum filesize while maintaining a specified level of image fidelity, or which achieves maximum image fidelity for a given filesize. The error-pooling stage allows us to interpret each of these problems as one of optimizing a scalar-valued function: the two problems can be described respectively as

$$\min f(X_0) \quad \text{subject to} \quad q(X, X_0) \geq \sigma, \tag{4.56}$$

and

$$\min q(X, Y) \quad \text{subject to} \quad f(Y) \leq \nu, \tag{4.57}$$

where $f : \mathbb{M}^{m\times n} \to \mathbb{R}$ measures the filesize of the image $X \in \mathbb{M}^{m\times n}$, $q : \mathbb{M}^{m\times n} \times \mathbb{M}^{m\times n} \to \mathbb{R}$ is a given real-valued image fidelity measure, and $\sigma, \nu \in \mathbb{R}$ are the minimum fidelity and maximum filesize, respectively.

The error-pooling stage requires a static definition for the summation across every factor, the goal of which is for the weighted sum to approximate the weighting that is performed by the human visual system. One method of finding weights for the summation is to perform a linear regression analysis to fit the resulting vector-valued measure to a real-valued measure defined by the Mean Opinion Score (MOS) obtained via subjective tests. For example, in the Picture Quality Scale (PQS) model of Miyahara, Kotani and Algazi [43], separate measures were designed to consider the effects resulting from:

- traditional mean-squared error (F_1) ;
- mean-squared error above a given threshold magnitude (F_2);
- errors due to block artifacts (F_3);
- correlated errors across larger regions (F_4); and
- masked errors near high-contrast boundaries (F_5).

(Each of the calculations of F_i was performed on images preprocessed to consider brightness non-linearity and contrast sensitivity.) These error measures were then combined using traditional mean-squared error techniques,

$$F_n = \frac{\sum_{i,j} f^n_{i,j}}{\sum_{i,j} X_{i,j}},$$

where f^n denotes the error map corresponding to the nth error measure, and X denotes the preprocessed original image.

One problem with this approach is that the specific weights found are dependent on the viewing conditions and test images used in the subjective experiments; these viewing conditions will vary widely across different monitors, web browsers and imaging applications, thus limiting the wider use of the combined measure[4]. A larger problem, however, is the fact that this approach ignores the earlier observation that low correlation is observed across subjects when judging quality according to the combination of multiple types of simple artifacts; this observation indicates that the observed weighting will vary widely across subjects, and thus the set weighting resulting from fitting the results of the subjective experiments can just as easily be a property of the specific subjects rather than a more

[4]The original PQS software included warnings when the magnitude of any of the five factors fell outside of the range of values that had been used to calibrate the scalar weighting. When applying this software to the `mandrill` test image quantized to IJG quality factor 98, the results fell outside of the specified range of values at IJG quality factor 80, well above the high-quality threshold measured in the experiments in Section 4.5.

general property of the human visual system. In an ideal situation, we would design a subject-specific image-metric; however, this is not implementable, so we need to explore alternate approaches.

The problems with the addition of the error-pooling stage can be avoided by changing the underlying optimization model by using as the objective function the vector-valued function $q(X, \hat{X}) = \left[F_1(X, \hat{X}), \ldots, F_5(X, \hat{X})\right]$, and generating a set of potentially optimal images; this set of images then provides options for choosing an image that is optimal for a given subject and a given set of viewing conditions, but without fixing the specific weighting used to select the optimal image across subjects and viewing conditions. We will discuss the theory and algorithms required to generate these points in the next section.

4.4.2 Definitions

Optimization problems with real-valued objective functions compare points using the total ordering defined on the real line: for an objective function $f : \mathbb{R}^n \to \mathbb{R}$, a point $x \in \mathbb{R}^n$ is optimal if $f(x) \le f(y)$ for all $y \in \mathbb{R}^n$. When we consider vector-valued objective functions $f : \mathbb{R}^n \to \mathbb{R}^m$, we have only a partial ordering, and thus will frequently have *incomparable points* where for $x, y \in \mathbb{R}^n$, none of $f(x) < f(y)$, $f(x) > f(y)$ and $f(x) = f(y)$ hold. We use the following series of definitions in the remainder of this chapter.

Definition 7 *[9, 10, 71] For $x \in \mathbb{R}^N$, m constraints $g_i(x)$ $(i = 1, \ldots, m)$ and n objectives $f_i(x)$ $(i = 1, \ldots, n)$:*

- *the **feasible region** (or **decision space**) is the set $X \subset \mathbb{R}^N$ defined by*

$$X = \left\{x \in \mathbb{R}^N : g_i(x) \le 0 \ \forall i = 1, \ldots, m\right\}; \tag{4.58}$$

- *the **multiobjective function** is the function $f : \mathbb{R}^N \to \mathbb{R}^n$ defined by*

$$f(x) = (f_1(x), \ldots, f_n(x)); \tag{4.59}$$

- *the **objective space** is the set $F \subset \mathbb{R}^n$ defined by*

$$F = \{f(x) : x \in X\}; \tag{4.60}$$

- *a **multicriteria optimization problem** (MOP) is formulated as*

$$\min_{x \in X} f(x), \tag{4.61}$$

or equivalently

$$\min_{x \in X} (f_1(x), \ldots, f_n(x))\,; \tag{4.62}$$

notice that this definition is not precise, as we need to define in what sense we can minimize a multiobjective function, since in general solutions will not exist that simultaneously minimize every objective f_i, $i = 1, \ldots, n$.

Remark: As mentioned earlier, when finding solutions for a given MOP, rather than finding a single optimal solution, we will construct a set of potentially optimal solutions. Therefore, an additional stage is necessary to find an optimal solution: the *analysis stage* of optimizing a MOP constructs a set of potentially optimal solutions, and the *decision-making stage* defines the preference structure used to determine which of the solutions from the constructed set is the optimal solution.

The following definitions formalize the above discussion.

Definition 8 *[64] For* $x \in \mathbb{R}^N$, *m constraints* $g_i(x)$ $(i = 1, \ldots, m)$, *n objectives* $f_i(x)$ $(i = 1, \ldots, n)$ *with corresponding multiobjective function* $f(x)$, *and the resulting MOP as defined in (4.61):*

- *a* ***cone*** *is a set* C *such that* $rC = C$ *for all* $r \in \mathbb{R}$;
- *given a set of vectors* $V = \{v_i\}_{i=1}^{p}$, *the* ***convex cone generated by*** V *is the set defined by*

$$cone(V) = \left\{ \sum_{i=1}^{p} w_i v_i : w_i \in \mathbb{R}_+ \right\}; \tag{4.63}$$

- *for a given convex cone* C, *a* ***partial ordering*** $\geq_C$ ***generated by*** C *is defined by*

$$x \geq_C y \quad \Leftrightarrow \quad x - y \in C; \tag{4.64}$$

- *for* $x, y \in \mathbb{R}^N$, *a cone* $C \subset \mathbb{R}^n$ *and a partial ordering* $\geq_C$, x ***dominates*** y *if* $f(y) \geq_C f(x)$;
- *for a given convex cone* $C \subset \mathbb{R}^n$, *a* ***domination set*** *is a set* $D(f)$ *(for* $f \in F$*) defined by*

$$D(f) = \{f\} \oplus C; \tag{4.65}$$

- *for a given convex cone $C \subset \mathbb{R}^n$, a point $x \in X$ is* ***nondominated*** *if $\nexists y \in X$ such that $f(x) \geq_C f(y)$, or equivalently if $D(f) \cap F = \{f\}$;*
- *the* ***nondominated set*** *is the set of all nondominated $f(x) \in F$.*

Remarks and Observations

Some additional remarks about the above definitions:

- the problem of balancing different conflicting objectives is of interest to a wide variety of areas of study, and thus many fields have their own terminology independent of each other;
- MOPs are often referred to as *multiobjective optimization* or *vector optimization problems*;
- nondominated points are also referred to as *noninferior*, *efficient*, *Pareto-optimal*, or *Pareto* points [28];
- nondominated points are often distinguished from noninferior points by limiting nondominated points to the objective space and noninferior points to the decision space [71].

In addition, two examples of cones C frequently used in modelling MOPs are [64]:

- *coordinate-order*: $C = \mathbb{R}^n_+$, i.e.

$$y \leq_C x \quad \Leftrightarrow \quad y_i \leq x_i \ (i = 1, \ldots, N); \tag{4.66}$$

- *lexicographic order*: order two points $x, y \in \mathbb{R}^N$ by first coordinate k satisfying $x_k \neq y_k$, i.e.

$$C = \{d \in \mathbb{R}^n : \exists k \in \{1, \ldots, n\} \text{ such that } d_i = 0 \ (i < k) \text{ and } d_k > 0\} \cup \{0\}\,. \tag{4.67}$$

4.4.3 Methodology

The central problem in solving multicriteria optimization problems is generating the nondominated set; since MOPs are extensions of single-objective optimization problems, a natural approach is to modify the given MOP in order to have a single objective that approximates some of the structure of the original multiobjective function. Once this translation has been performed, we can apply standard techniques in solving single-objective problems, such as the simplex method (for linear objectives), steepest descent or quasi-Newton methods.

Two classes of methods for constructing the nondominated set are generating methods and preferential methods. The distinction between these two classes of methods is analogous to that of the distinction between the analysis and decision-making stages of solving a MOP: generating methods focus on the analysis stage, solving multiple single-objective problems to construct an approximation of the nondominated set, and leaving the decision-making stage as a separate task; preferential methods focus on the decision-making stage, defining one (or a small number of) single-objective optimization problem whose objective function makes explicit use of the preferences used in the decision-making task. We will limit our discussion to two generating methods, the weighting method and the constraint method. Information on preferential methods can be found in [17, 28, 64, 71].

Weighting Method

The weighting method combines the multiple objectives into a single objective function using a weighted sum, and solves the resulting single-objective optimization problem repeatedly for various weights; the modified single-objective optimization problem that the weighting method solves is

$$\min_{x \in X} f(w, x) = \sum_{i=1}^{n} w_i f_i(x), \tag{4.68}$$

where X and $f_i(x)$ are as defined in the original formulation (4.62).

Notice that our single-objective is now a function of both the decision variables x and the weights w, as the specific problem being solved is dependent on the values of the weights. Also, for $w = e_i$, the ith basis vector in $\mathbb{R}^N$, the problem (4.68) finds the solution that optimizes the ith objective. Formulated as an algorithm, the weighting method can be described as:

Algorithm 9 *For a feasible set X, a multiobjective function $f(x)$, and a given MOP (4.62):*

1. *Optimize each objective $f_i(x)$ individually, to find a range of nondominated extreme points.*

2. *Vary each of the weights to find additional nondominated extreme points.*

When optimizing each objective separately, if the optimal solution x^* is the same for all objectives, then X^* is obviously optimal for the multiobjective problem; the nondominated set would be a singleton. Otherwise, the different solutions found define bounds on the nondominated set, and modifying the weight w directs the cone ordering across the nondominated set in search of new nondominated points.

Constraint Method

As with the weighting method, the constraint method first considers each of the multiple objectives $f_k(x)$ individually as a single objective; however, rather than combining the objectives, the constraint method forces bounds on all remaining objectives $f_i(x)$ for $i \neq k$, and modifies the bounds to force movement across the nondominated set. The optimization problem that the constraint method solves is

$$\min_{x \in C_k(l,u,X)} f_k(x), \tag{4.69}$$

where

$$C_k(l,u,X) = \{x \in X : l_i \leq f(x_i) \leq u_i \ \forall i = 1, \ldots, k-1, k+1, \ldots, m\}. \tag{4.70}$$

Notice that our single-objective is now fixed, while the feasible region is changing as a function of the lower and upper bounds l and u. However, since $C_k(l,u,X) \subset X$, the bounds for each single-objective problem should be chosen to avoid bounding the problem outside of X, ie. $C_k(l,u,X) = \emptyset$. Formulated as an algorithm, the constraint method can be described as [17]:

Algorithm 10 *For a feasible set X, a multiobjective function $f(x)$, and a given MOP (4.62):*

1. *Optimize each objective $f_i(x)$ individually, to form an initial collection of nondominated extreme points $\boldsymbol{x}_i$.*

2. *Evaluate each objective $f_i(x)$ at the optimal solution vectors for every optimal solution vector $\boldsymbol{x}_k$ $(k = 1, \ldots, n)$.*

3. *For each objective $f_k(x)$, set m_k and M_k equal to the smallest and largest values $f_k(\boldsymbol{x}_i)$ respectively, taken over the evaluations performed in Step 2.*

4. *Evaluate the modified problems of the form shown in equations (4.69)-(4.70), for various values of l_k satisfying $l_k \geq m_k$ and $l_k \leq M_k$.*

Adding additional constraints limits the set of feasible vectors to a subset of X; calculating m_k and M_k ensures that the bounds chosen for the new constraints do not make the problem infeasible (ie. $X \neq \emptyset$). Changing the constraints forces new nondominated points to be found by forcing the points already found outside the modified feasible region.

4.4.4 Application to visual thresholds

The weighting method generates the entire nondominated set by varying the values of the weights of each objective. A specific weight represents preference information of the decision-maker; the weighting defines a priority on the objectives, in that an objective receiving a higher weighting is of a higher priority than one receiving a lower weighting. We can use this fact to define a method for dynamically choosing an image for both high- and low-bandwidth environments. For high-bandwidth environments, high image quality is a higher priority than low-filesize, while the opposite is true for low-bandwidth environments. Therefore, we can define a set of weightings that are adjusted for each environment based on these conflicting priorities, by setting two objectives f and g as filesize and a given summation method, respectively. For original image X and X_n the image compressed using IJG quality factor n, we then solve

$$\min_{n \in \{0,\ldots,100\}} f(X_i) + \lambda g(X_i, X) \tag{4.71}$$

where λ takes high values for high-bandwidth environments, and low values for low-bandwidth environments.

We apply a similar approach to this problem using the constraint method; in this case, the constraints are used to define the maximum filesize and minimum visual fidelity for low- and high-bandwidth environments, respectively. We define a pair of optimization problems by defining two objectives f and g as filesize and a given summation method, respectively; for original image X and X_n the image compressed using IJG quality factor n, we then solve

$$\min_{n \in \{0,\ldots,100\}} f(X_i) \quad \text{subject to} \quad g(X_i, X) \geq \sigma \tag{4.72}$$

for high-bandwidth environments, and

$$\max_{n \in \{0,\ldots,100\}} g(X_i, X) \quad \text{subject to} \quad f(X_i) \leq \sigma^* \tag{4.73}$$

for low-bandwidth environments. We return to both of these algorithms in Section 4.5.

4.5 Experiments

In order to explore the algorithm proposed in Section 4.3.5, we need dissimilarities that correspond to observed visual thresholds. To measure these dissimilarities, we performed subjective experiments using the `ImgStat` application described in Section 4.1[5].

4.5.1 Methods

Using the `ImgStat` application, each subject was instructed to decrease the quality level until differences between the original and test images were first noticeable. Comparisons were made in-context by right-clicking on the image, which toggled between the original and compressed images; when differences became noticeable, the subject was instructed to select the *High-quality threshold* checkbox. The subject was then instructed to decrease the quality level until differences became objectionable; at this point the subjects selected the *Low-quality threshold* checkbox, and continued to the next image in the sequence.

The five test subjects were selected for varying levels of experience in viewing images on the internet, although each had a minimum level of experience using the internet. All subjects had normal or corrected-to-normal vision, and received the same simple training as in the experiments in Section 2.3.4. This included a description of the types of errors that would be occurring within the test images[6]. Although this experiment did not use the *Mean Opinion Scale* (MOS) rating explicitly, we presented the MOS ratings and their text descriptions as guidelines for determining the two thresholds: we presented the high-quality threshold as MOS = 4, and the low-quality threshold as MOS = 2.

The test images were taken from a subset of 254 images of a larger database of images, previously described in Section 2.3.4; the first 73 of the 254 images were used for the experiments described in Section 2.3.4. The images selected were square images (necessary for the ROI model) with original IJG quality factors that occurred most frequently in the larger image database; the images were then resized to 256×256 pixels, converted to grey-scale, and recompressed using their original IJG quality factor[7] The images were presented

[5]Special thanks to Packeteer for permission to include the following experimental methodology and results.

[6]`http://www.cecm.sfu.ca/~mmacklem/thesis/MOS.html`

[7]In hindsight, this is the one design choice that I would have done differently. The recompression using the original IJG quality factor was done in order to maintain the original proportions in the larger database. By resizing and then compressing using the 8×8 DCT transform, we created additional blocking noise to an image that could have had blocking in the original image; however, the resizing process shrinks the original

to the subjects on a 19-inch monitor at 1024×768 display resolution, at a viewing distance of 3 times the image height, a constant distance since all images were resized to have the same dimensions.

4.5.2 Results

The measured high- and low-quality thresholds are shown in Tables 4.2 and 4.3, respectively. Each table shows the displayed-value at each threshold (on the slider-bar within the `ImgStat` interface), with the actual IJG quality factor value in parentheses; throughout the remaining discussion, we will refer to the measured thresholds using the actual IJG quality factors. Each subject took approximately 25-30 minutes to complete the experiment, so there was little variation due to fatigue. In addition, the training resulted in consistent responses across subjects at the start of the experiment. Furthermore, although there were still a number of images that had a wide variety in the responses (13 of 21 had differences of more than 20 between the 5 responses), most of the images had enough consistency to estimate the location of the high-quality threshold (15 of 21 had differences of less than 15 between the closest 4 of the 5 responses). Not surprisingly, the widest variation in responses was for the `mandrill` image: the high-frequency speckling was compressed imperfectly for relatively high quality factors, resulting in increased high-quality threshold rating for subject 4; however, the presence of the speckling masked existing errors throughout the image, resulting in decreased high-quality threshold rating for subject 2.

The low quality threshold ratings had considerably more variance between subjects: 13 of 21 images had differences in the IJG quality scaling factor of more than 30, while 9 of 21 had differences of more than 20 between the closest 4 of 5 subjects. However, the structure of the responses of each subject is self-consistent, in that each subject has a consistent rate of decrease between their high- and low-quality threshold ratings; this indicates that the subjects use different artifacts as "triggers" for determining their low-quality threshold. Furthermore, 17 of 21 images have differences of less than 15 between the closest 3 of 5 ratings.

8×8 block partition, thus creating two different sizes of blocking artifacts as we decrease the quality of the resized image.

Image Number	Image Name	Original quality level
1	105	59
2	14	90
3	145	98
4	bird	98
5	154	89
6	167	74
7	17	76
8	175	37
9	109	88
10	mandrill	98
11	176	39
12	200	49
13	Tower	88
14	peppers	98
15	232	74
16	234	85
17	27	89
18	36	76
19	42	85
20	lena	98
21	57	93

Table 4.1: Order of image presentation in *ImgStat* experiment, with reference names for use in the demonstration application

According to these observations, we determine the high- and low-quality thresholds as follows:

- the high-quality threshold Λ_n is determined to be the average of the closest 4 of 5 subject responses;
- the low-quality threshold λ_n is determined to be the median of the closest 3 of 5 subject responses.

4.5.3 Modelling visual thresholds using multidimensional scaling

We model the low- and high-quality thresholds using a variation of Algorithm 8, using the threshold values as defined in Section 4.5.2.

Image Number	1	2	3	4	5
1	91 (54)	76 (45)	81 (48)	91 (54)	82 (48)
2	93 (84)	58 (52)	66 (59)	74 (67)	76 (68)
3	71 (70)	35 (34)	53 (52)	79 (77)	71 (70)
4	72 (71)	38 (37)	48 (47)	78 (76)	69 (68)
5	95 (85)	49 (44)	49 (44)	90 (80)	73 (65)
6	94 (70)	78 (58)	83 (61)	84 (62)	88 (65)
7	92 (70)	85 (65)	86 (65)	86 (65)	84 (64)
8	89 (33)	76 (28)	87 (32)	90 (33)	89 (33)
9	94 (83)	70 (62)	69 (61)	89 (78)	92 (81)
10	71 (70)	35 (34)	58 (57)	91 (89)	61 (60)
11	95 (37)	80 (31)	90 (35)	90 (35)	84 (33)
12	89 (44)	60 (29)	79 (39)	91 (45)	83 (41)
13	63 (55)	70 (62)	60 (53)	90 (79)	70 (62)
14	62 (61)	40 (39)	58 (57)	61 (60)	59 (58)
15	77 (57)	75 (56)	76 (56)	90 (67)	90 (67)
16	68 (58)	33 (28)	87 (74)	84 (71)	89 (76)
17	68 (61)	60 (53)	69 (61)	90 (80)	74 (66)
18	86 (65)	85 (65)	80 (61)	82 (62)	86 (65)
19	94 (80)	57 (48)	87 (74)	91 (77)	92 (78)
20	81 (79)	55 (54)	62 (61)	91 (89)	64 (63)
21	83 (77)	45 (42)	40 (37)	84 (78)	59 (55)

Table 4.2: Results of *ImgStat* experiment, for High-quality threshold; the test images were presented to each subject, who then adjusted the quality parameter until differences first became noticeable. The quality parameter was normalized to have values from 1 to 100, depending on the original IJG quality parameter; shown are the selected un-normalized quality parameters, with the corresponding actual IJG quality parameter in parentheses.

Image Number	1	2	3	4	5
1	62 (37)	32 (19)	59 (35)	41 (24)	65 (38)
2	66 (59)	16 (14)	33 (30)	50 (45)	54 (49)
3	52 (51)	15 (15)	41 (40)	48 (47)	49 (48)
4	56 (55)	22 (22)	34 (33)	50 (49)	44 (43)
5	64 (57)	34 (30)	40 (36)	63 (56)	48 (43)
6	73 (54)	65 (48)	78 (58)	67 (50)	76 (56)
7	84 (64)	45 (34)	82 (62)	61 (46)	73 (55)
8	69 (26)	65 (24)	77 (28)	71 (26)	80 (30)
9	86 (76)	35 (31)	50 (44)	68 (60)	55 (48)
10	44 (43)	18 (18)	35 (34)	65 (64)	39 (38)
11	71 (28)	47 (18)	79 (31)	73 (28)	70 (27)
12	49 (24)	30 (15)	65 (32)	76 (37)	63 (31)
13	18 (16)	32 (28)	42 (37)	72 (63)	50 (44)
14	40 (39)	18 (18)	32 (31)	45 (44)	39 (38)
15	60 (44)	29 (21)	72 (53)	81 (60)	77 (57)
16	35 (30)	25 (21)	57 (48)	71 (60)	57 (48)
17	48 (43)	50 (45)	57 (51)	74 (66)	56 (50)
18	66 (50)	26 (20)	74 (56)	73 (55)	76 (58)
19	48 (41)	20 (17)	57 (48)	70 (60)	56 (48)
20	60 (59)	25 (25)	42 (41)	76 (74)	42 (41)
21	53 (49)	25 (23)	35 (33)	68 (63)	35 (33)

Table 4.3: Results of *ImgStat* experiment, for Low-quality threshold; the test images were presented to each subject, who then adjusted the quality parameter until differences first became objectionable. The quality parameter was normalized to have values from 1 to 100, depending on the original IJG quality parameter; shown are the selected un-normalized quality parameters, with the corresponding actual IJG quality parameter in parentheses.

Consider the set of images $\{X_q^n\}$, where X_q^n is the nth test image compressed to IJG quality factor q. We partition this set into *perfect approximations* P_n, *acceptable approximations* A_n, and *objectionable approximations* B_n, where

$$X_q^n \in \begin{cases} P_n, & q \geq \Lambda_n; \\ A_n, & \lambda_n \leq q < \Lambda_n; \\ B_n, & q < \lambda_n. \end{cases} \tag{4.74}$$

We then define the dissimilarities $[\delta_{rs}]$ as

$$\delta_{rs} = |Q(X_r^n) - Q(X_s^n)| \tag{4.75}$$

where

$$Q(X_r^n) = \begin{cases} 0, & X_r^n \in P_n; \\ 1, & X_r^n \in A_n; \\ 2, & X_r^n \in B_n. \end{cases} \tag{4.76}$$

Using these dissimilarity values, we construct a configuration using the MDS methods discussed in Section 4.3, and use Procrustes Analysis to compare it to the configuration constructed by the dissimilarities created using the summations in Section 3.3.

The results of our experiments using this method are shown in Tables 4.4 and 4.5. The numbers shown are averages of the lowest three of four ratings taken from three repetitions on each of the `bird`, `lena`, `mandrill`, and `peppers` test images; we start each repetition using a randomly generated initial configuration, so multiple repetitions are performed to remove the influence of an unfortunate starting configuration[8].

In each repetition, we use a subset of the compressed images, corresponding to IJG quality factors 25 to 80; this avoids large values of R^2 dominated by variations in the locations of the points in the configuration representing the extreme ends of the quality scale. When computing the configurations iteratively for the experiments comparing the structural similarity of each summation to MSE, we stop each repetition once the stress of both configurations stabilizes to six decimal places. For the experiments comparing each summation dissimilarity matrix to the observed threshold matrix, we stop each repetition

[8] In the experiments comparing the configuration for each measure to that of the MSE, we occasionally encountered initial configurations for which R^2 did not converge, continually increasing at each iteration. This behaviour occurred whenever a large value of R^2 occurred at the first iteration, at which point neither configuration accurately models its respective data. The cause of this behaviour is still unknown, and was the motivation for multiple iterations; in the near future, we would like to perform additional repetitions of these experiments on a larger set of test images.

when the value of R^2 stabilizes to six decimal places; as previously discussed, the ideal configuration for the threshold matrix has zero stress, so we construct a configuration for the given measure whose stress stabilizes to six decimal places, and perform additional iterations on the construction of the threshold matrix configuration until R^2 stabilizes.

Prior to comparing the results of the two algorithms, notice that the two algorithms measure opposite effects. Since the mean-squared error is observed to be monotonic (as in Section 3.3.2), limiting our focus to the subrange of the IJG quality scale that decreases slowly will result in a configuration whose points are approximately collinear and evenly distributed. Therefore, a configuration that is clustered, as with the optimal configuration as defined by the threshold-matrix, will differ structurally from the configuration constructed using the MSE as the underlying distance measure.

Given this observation, the output from both algorithms almost agree with each other, with one major exception: the structure of the σ-threshold measure for $\sigma = 2$ is determined to be both similar to the structure of the MSE configuration and the threshold configuration, a result which appears to be contradictory based on our previous discussion. However, the two algorithms agree on the remaining results: each algorithm rates the Canberra and divergence measures as more effective in capturing the threshold-clustering, and σ-threshold measures as more effective in capturing the threshold-clustering than the σ-threshold measure for $\sigma = 1$, which is somewhat unexpected given the results of the topological comparison in Section 3.3.1. Some additional observations about the results of this experiment:

- The value of R^2 gives a slightly different assessment of the summation from the topological comparison algorithms. The most obvious example is the better structural similarity of the σ-threshold measure for $\sigma = 2$ than for $\sigma = 1$, the opposite of the measurements of topological comparison Algorithm 5.

- In assessing the accuracy of either algorithm, as well as the topological comparison algorithms, the ideal technique would be to define the target matrix as the measured subjective ratings for each image, with the dissimilarities δ_{rs} defined as the difference in the subjective fidelity ratings given for each image I_r and I_s; this would improve the present definition of the threshold-matrix since the JPEG Perceptual Dipper illustrates that the set of *acceptable approximations* is not a connected subset of the IJG quality scale. However, the subjective experiments required to create this matrix are prohibitively expensive (in both time and cost) to be computed at this time.

Summation Index	R^2
3	0.577848
4	0.592305
8	0.565501
12 ($\sigma = 1$)	0.531005
12 ($\sigma = 2$)	0.556161
12 ($\sigma = 3$)	0.581490

Table 4.4: Structural similarity of the summations from Table 3.2 to MSE, averaged over test images `bird`, `lena`, `mandrill`,and `peppers`

Summation Index	R^2
3	5.2107972
4	4.618106778
8	29.2039994
12 ($\delta = 1$)	29.342036
12 ($\delta = 2$)	3.050825
12 ($\delta = 3$)	20.310053

Table 4.5: Structural similarity of the summations from Table 3.2 to matrix defined by observed thresholds, averaged over test images `bird`, `lena`, `mandrill`,and `peppers`

These observations can benefit from confirmation by additional repetitions on a larger subset of images, although three repetitions gave some consistency in the results for each image.

4.5.4 Modelling visual thresholds using multicriteria optimization

We experiment with modelling the observed thresholds using multicriteria optimization by a combination of the weighting and constraints methods: for each measure, we create a set of data points in $\mathbb{R}^2$, with entries equal to the image filesize and fidelity, as measured by the given summation. Taking the convex hull of the resulting points, we define a weighted sum of the filesize and fidelity (defined by a given summation) by normalizing both the filesize and fidelity to the interval $[0, 1]$; we then find the image that optimizes the weighted sum, and vary the weights to get the image that provides the closest match to both thresholds.

The weights are defined differently for the high- and low-quality thresholds: the weighted sum for modelling the high-quality threshold is first found by adjusting the weights to find the closest fit to this threshold; the resulting weight is multiplied by an additional factor, which is adjusted to find the closest fit to the low-quality threshold. This weighting can then be used for other images to measure the consistency of the match to the observed threshold.

Table 4.6 shows the results of applying this method to the `bird` test image. Of particular interest is the fact that the MSE matched both thresholds with a higher degree of accuracy than 6 of the 7 measures we selected as a result of the topological comparison experiments; this is likely due to the fact that the non-monotonicities found in each of these measures cause the convex hull of the epigraph of the given measure to be a smaller subset of the IJG quality scale than the MSE, thus giving a smaller range of choices in finding the optimal match to the observed thresholds. This can potentially be avoided by adding the selected measures to the optimization problem by adding an additional criterion, although the time factor involved can be prohibitively expensive depending on the number of measures used in the comparison.

Given the results shown in Table 4.6, we can refine this method by not taking the convex hull of the epigraph of the given summation, and instead searching along the IJG quality scale for a non-monotonicity; the idea is that if the summation accurately measures the perceptual quality of the given compressed image, then the existing numerical non-monotonicity will correspond to a non-dominated point with smaller filesize and decreased error, as observed with the JPEG Perceptual Dipper Effect of Section 2.3.1. To do this, we decrease the IJG quality factor, and measure the quality using a given summation, repeating this process until the summation output becomes non-monotonic; we define this local minimum to be the *measured high-quality threshold*. We then repeat this process starting at this measured threshold, with an offset in the quality factor λ designed to ensure that we find a different local minimum; we define the resulting local minimum to be the *measured low-quality threshold*.

The results of this experiment are shown in Tables 4.7-4.10, with a separate table for each of the divergence and σ-threshold measures, for $\sigma = 1, 2, 3$. Some general observations about the results:

- No results of interest were found for either the Canberra or Wave-Hedges measures; the only consistent non-monotonicities in either measure occurred at IJG quality factor 2, although the Canberra measure had additional non-monotonicities for the `peppers` test image.

- With the exception of the `lena` test image, the divergence and σ-threshold ($\sigma = 1$) measures tend to over-estimate both thresholds for images originally compressed using higher IJG quality factor values (≥ 90). More promising results were found for images

Summation Index	Measured Thresholds (`bird`)
MSE	76/38
3	87/33
4	76/42
8	87/28
12 ($\sigma = 1$)	90/21
12 ($\sigma = 2$)	90/21
12 ($\sigma = 3$)	90/21

Table 4.6: Thresholds measured by solving multicriteria optimization problem, using the `bird` test image with observed thresholds 66/49. The summation index refers to Table 3.2.

encoded at lower quality settings, although the close proximity of the two thresholds for test image 17 were not detected by either measure due to the offset defined from the measured high-quality threshold.

- With the exception of `lena`, the results for high-quality originals improve for $\sigma = 2$. However, as observed in Section 3.3.2, increasing σ smooths the resulting summation values, thus removing non-monotonicities occurring at higher and lower IJG values. Consequently, some thresholds are severely under-estimated, such as the low-quality threshold for `lena` and both thresholds for `Tower`. Increasing σ further can remove all non-monotonicities, as observed in Table 4.10 for the `lena` and `Tower` images.

Based on these observations, Table 4.11 shows the measured thresholds, defined piecewise based on the original IJG quality factor: for original quality factor greater than 90, we use the thresholds calculated using $\sigma = 2$, otherwise we use the thresholds calculated using $\sigma = 1$. We can see that this piecewise calculation matches the observed thresholds quite closely, with the exception of the low-quality threshold for the `lena` test image. Although these results show some promise in this method, they can benefit from a larger original set of images; to this end, we are performing these experiments on the original set of 254 test images, although the results are not yet known.

4.6 Conclusions and further research

In this chapter, we presented an interface used to measure visual thresholds, and discussed two ways to model visual thresholds by splitting the problem across multiple dimensions. Multidimensional scaling splits a single measure into multiple dimensions in order to allow

Image number	Original quality	Observed thresholds	Measured thresholds
bird	98	66/49	73/56
lena	98	64/41	57/31
mandrill	98	69/38	n/a
peppers	98	59/39	81/58
Tower	88	58/37	76/63
105	59	51/38	50/39
14	90	62/49	75/53
145	98	67/48	82/70
154	89	69/56	62/49
167	74	62/56	68/43
17	76	65/62	56/44

Table 4.7: Observed thresholds, with thresholds measured by checking for non-monotonicities in the *divergence* measure over the IJG quality scale.

Image number	Original quality	Observed thresholds	Measured thresholds
bird	98	66/49	75/63
lena	98	64/41	60/46
mandrill	98	69/38	72/61
peppers	98	59/39	72/58
Tower	88	58/37	47/37
105	59	51/38	53/42
14	90	62/49	75/61
145	98	67/48	75/58
154	89	69/56	78/65
167	74	62/56	62/50
17	76	65/62	64/52

Table 4.8: Observed thresholds, with thresholds measured by checking for non-monotonicities in the σ-*threshold* measure ($\sigma = 1$) over the IJG quality scale.

Image number	Original quality	Observed thresholds	Measured thresholds
bird	98	66/49	58/46
lena	98	64/41	51/19
mandrill	98	69/38	61/41
peppers	98	59/39	72/58
Tower	88	58/37	40/8
105	59	51/38	52/41
14	90	62/49	71/53
145	98	67/48	59/46
154	89	69/56	75/65
167	74	62/56	58/45
17	76	65/62	65/47

Table 4.9: Observed thresholds, with thresholds measured by checking for non-monotonicities in the σ-*threshold* measure ($\sigma = 2$) over the IJG quality scale.

Image number	Original quality	Observed thresholds	Measured thresholds
bird	98	66/49	51/30
lena	98	64/41	n/a
mandrill	98	69/38	55/43
peppers	98	59/39	50/36
Tower	88	58/37	n/a
105	59	51/38	50/39
14	90	62/49	65/52
145	98	67/48	53/37
154	89	69/56	68/52
167	74	62/56	60/41
17	76	65/62	40/25

Table 4.10: Observed thresholds, with thresholds measured by checking for non-monotonicities in the σ-*threshold* measure ($\sigma = 3$) over the IJG quality scale.

Image number	Original quality	Observed thresholds	Measured thresholds
bird	98	66/49	58/46
lena	98	64/41	51/19
mandrill	98	69/38	61/41
peppers	98	59/39	72/58
Tower	88	58/37	47/37
105	59	51/38	53/42
14	90	62/49	71/53
145	98	67/48	59/46
154	89	69/56	78/65
167	74	62/56	62/50
17	76	65/62	64/52

Table 4.11: Observed thresholds, with thresholds measured by combining the results of $\sigma = 1, 2$ based on the original IJG quality factor.

the data to be easily visualized, as well as extracting structure across multiple dimensions from a single scalar-valued measure. Multicriteria optimization considers multiple measures separately by finding a set of potentially optimal solutions, depending on the desired properties of the final solution.

The results of the multicriteria optimization approach highlights the usefulness in considering the topological similarities between a given measure and the Mean-Squared Error; since the MSE is monotonically decreasing for increasing IJG quality factor, in order to define cutoffs for quantization levels according to the nonmonotonicities in a given quality measure, the measure must differ topologically from the MSE. Therefore, comparing the topological structure of a given measure to the MSE gives us additional information about the measure that can be applied to a requantization approach, such as described earlier in Section 2.3.1.

One advantage of each of these methods is that they can easily scale to larger problems: advanced techniques in multidimensional scaling can consider dissimilarities that consist of multiple components, each of which cannot be compared to each other due to differences in the type of data used (i.e., discrete/categorical vs. continuous measurements), while modelling larger problems with multicriteria optimization consists of simply adding additional dimensions to the multiobjective function. Both techniques are of interest for possible improvements in the results of the experiments discussed in this chapter.

Chapter 5

Conclusions and Further Research

This thesis discusses the general problem of measuring image fidelity, and describes some of the properties of the human visual system that must be considered by an accurate perceptual fidelity measure. We present the JPEG Perceptual Dipper Effect, a phenomenon that is not predicted by the Mean-Squared Error; this effect is extremely useful in recompressing JPEG-encoded images to multiple bitrates, in order to present high-fidelity images to environments of varying bandwidths.

We presented the Dipper Effect in the larger context of visual thresholds, which can be useful in determining images that balance the contradictory requirements of high-fidelity and small filesize. Our discussion has focussed on a series of methods to analyze the structure of a given image fidelity measure:

- three topological comparison algorithms, used to determine the degree to which the rank-ordering of a given measure is non-monotonic, thus potentially being able to detect the Dipper Effect;

- two methods of using the output of multidimensional scaling algorithms to determine the degree to which the clustering behaviour of a given measure is similar to the partition of a set of images into low- and high-fidelity images;

- two methods of using multicriteria optimization to detect images that provide a best balance between small filesize and high quality.

The three topological comparison algorithms present a compelling argument that the problems encountered with the Mean-Squared Error, such as its monotonicity as a function of the IJG quality factor, are not necessarily implicit in the pixel-based approach for comparing images, where comparisons are limited to corresponding pixels within the two images. By comparing the topological differences between the MSE and a set of alternate summations, we found that different rank orderings can be found using only pixel-based comparisons. Using this fact, we achieved promising results in fitting measured visual thresholds by matching them with nonmonotonicities for a specific set of summations. These results provide justification for considering the rank ordering as a criteria for comparing two summation methods.

Although each of the algorithms presented in this thesis performed with varying success, each provides an additional use in their traditional context:

- the topological comparisons can be used to compare the rank ordering of a measure to a known rank ordering, such as that for the set of images that contain the speckling artifacts caused by the Dipper Effect;

- the perceptual measures provide a method of presenting the appearance of a compressed image, so that during the process of a designing a compression algorithm, the artifacts created by the data loss will be detected;

- the multidimensional scaling algorithms can be used to visualize the set of images in terms of their distances from each other, as measured using a given image fidelity measure.

Throughout this thesis, our discussions have been built on a series of assumptions; for example, we have assumed that JPEG is the compression algorithm being used to generate the set of images. Potentially, this model can be useful in more general settings by experimenting with different conditions; some examples include:

- *change of domain*: The alternate summation measures in Tables 3.2 and 3.2 could have different structures when applied in different domains, such as the DCT or wavelet domain. In fact, James S. Walker states in [78] that improved results over MSE were found by applying the L_1 norm to the low frequency images in the wavelet domain (using D2 wavelets); however, we were unable to replicate his results when we tried this experiment ourselves.

- *perceptual map*: in Section 3.3.1, we used Osberger's ROI component maps as pixel-weightings, with limited effect. In addition to the VDP, some additional perceptual maps to use as pixel-weightings include the PIC [51], Lubin [39], and Teo & Heeger perceptual models [73].

- *segmentation criteria*: in defining the ROI perceptual map, a segmentation of the image into component objects is required. We have used the object's variance as the segmentation criteria, but others are also frequently used, such as soft-threshold, hard-threshold, and entropy. Of additional interest, a general treatment of segmentation in [27] allows for arbitrary criterion to define segmentation method at each stage.

- *different visual threshold criteria*: in measuring the high- and low-quality visual thresholds, we used the existence of noticeable differences as the threshold criteria. Another possible criteria is the presence of blocking.

- *additional MDS methods*: as discussed earlier, including the more complicated ALSCAL and SMACOF models are of interest to compare the configurations constructed using these methods.

It would also be of interest to explore how these methods work on current fidelity measures for color images, as well as for video, but each has additional visual factors to consider that raise the complexity outside the scope of our present discussions.

Appendix A

Code Listing

A.1 Octave code

The following code calculates the upper and lower bounds on the DCT coefficients (matrices (2.20) and (2.21), respectively).

A.1.1 Calculate bounds on DCT entries

Lower bounds on DCT coefficients

```
function [r] = paper_mindct()
   r = zeros(8,8);
   for v=1:8
      for u=1:8
         for i=1:8
            for j=1:8
               temp = cos((2*i-1)*(u-1)*pi/16)*cos((2*j-1)*(v-1)*pi/16);
               if (temp < 0)
                  x(j,i) = 255;
               else
                  x(j,i) = 0;
               end;
            end;
         end;
```

```
         temp = dct(x);
         r(v,u) = temp(v,u);
      end;
   end;
end;
```

Upper bounds on DCT coefficients

```
function [r] = paper_maxdct()
   r = zeros(8,8);
   for v=1:8
      for u=1:8
         for i=1:8
            for j=1:8
               temp = cos((2*i-1)*(u-1)*pi/16)*cos((2*j-1)*(v-1)*pi/16);
               if (temp > 0)
                  x(j,i) = 255;
               else
                  x(j,i) = 0;
               end;
            end;
         end;
         temp = dct(x);
         r(v,u) = temp(v,u);
      end;
   end;
end;
```

Appendix B

Supplemental Material

This chapter describes additional material not contained in the printed version of this thesis. Many of the phenomena we have discussed in this thesis are specific to displays, and thus do not translate to print media. As such, we have prepared a supplementary application and web-site.

B.1 JPEGApp

This chapter describes the `JPEGApp` application designed to serve as a general interface for the algorithms described throughout this thesis. The intention of this application is to provide a means to experiment with the effects that varying the parameters in each algorithm have on the algorithm's output/performance. In addition, many of the phenomena that we have described in this thesis do not easily translate to print media, and thus this application provides a means to see these phenomena directly on a computer display.

The `JPEGApp` application was originally designed to serve as a front-end for a large collection of disparate software pieces that I had written in the course of the research in this thesis. The development of this application was motivated by the discovery of the *Java Advanced Imaging* (JAI) API [32], which solved a number of image processing and displaying issues that had plagued previous attempts at writing similar front-ends. As such, the `JPEGApp` has grown out of its original role of front-end into a more ambitious research project, which means that a number of interesting potential pieces are still in development. Included in the descriptions for each of the interfaces is additional pieces that (time permitting) will eventually be included, described as opportunities for further research.

The images in the `JPEGApp` software were read and displayed using the *Java Advanced Imaging* (JAI) API [32]; all other additional software is described in the context of the interface in which they are used.

B.1.1 Visual Perception

Michelson Contrast

- *Figure*: B.1
- *Section*: 1.3.1, 3.1.1, 3.1.2
- *Description*: This interface demonstrates the process of measuring Michelson contrast, which is defined to be the amplitude of a sine grating when it is first visible on a constant background of a fixed luminance. The top square is a constant background, with luminance level set by the top scroll-bar. The middle square is a sine grating, with two adjustable parameters set by the middle two scroll-bars: the top and bottom scroll-bars set the frequency and amplitude of the grating, respectively. The bottom square shows the combination of the background and grating.

  This interface provides a means of defining your *contrast sensitivity function*. By adjusting the amplitude until the grating first becomes visible in the bottom square, you can set your contrast threshold for the given frequency. By pushing the `Set threshold` button, the contrast threshold is plotted in the bottom window, frequency (horizontal) vs. threshold (vertical); repeating this experiment for various frequencies will give the shape of the contrast sensitivity function observed in Figure 3.2.
- *Parameters*: Background luminance (top scroll-bar), grating frequency (centre scroll-bar) and amplitude (bottom scroll-bar)
- *Additional software*: `XYPlot` class to graph thresholds (Dennis Sigel, Sun Microsystems)
- *To do*: Add frequency orientation, to measure the observed anisotropy of the contrast sensitivity function

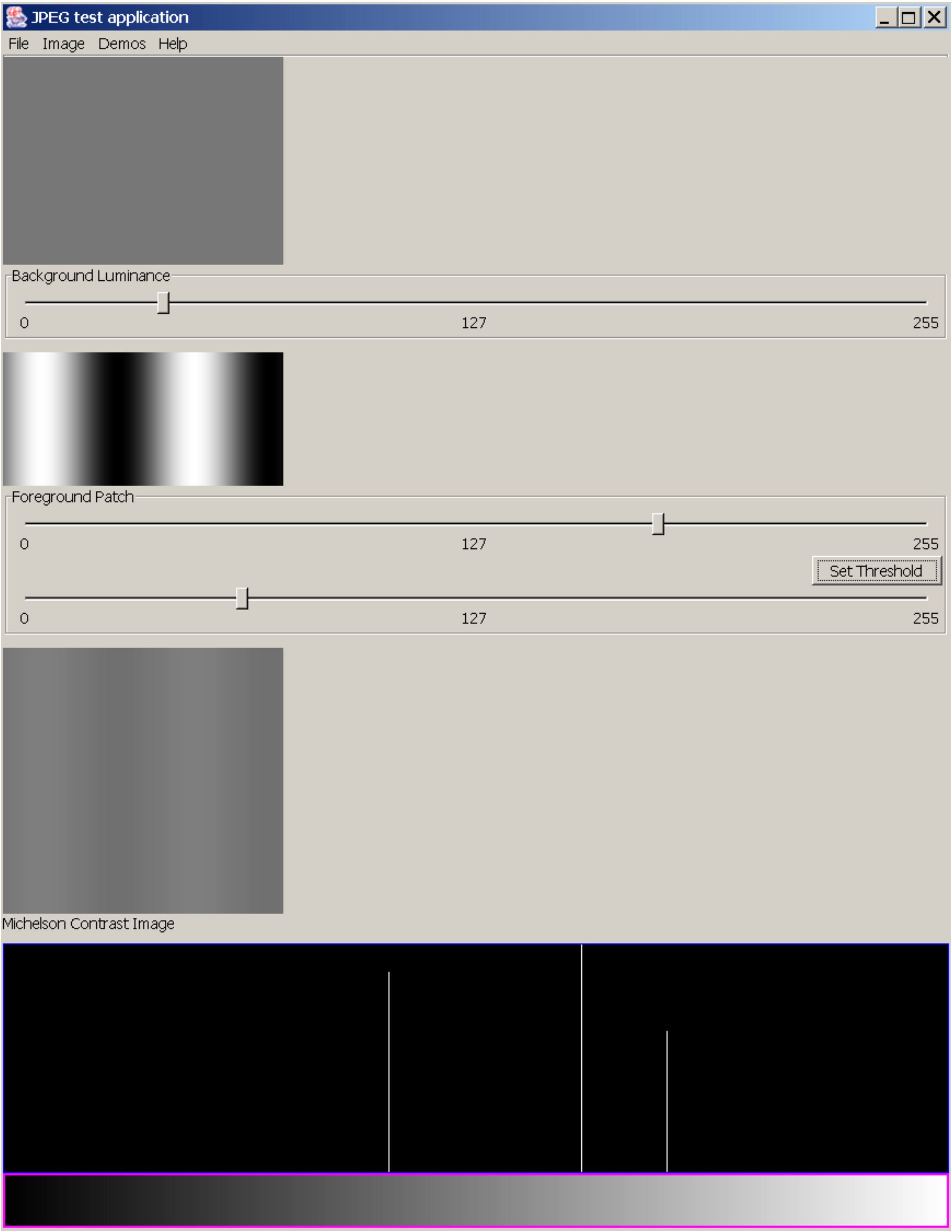

Figure B.1: Michelson contrast interface

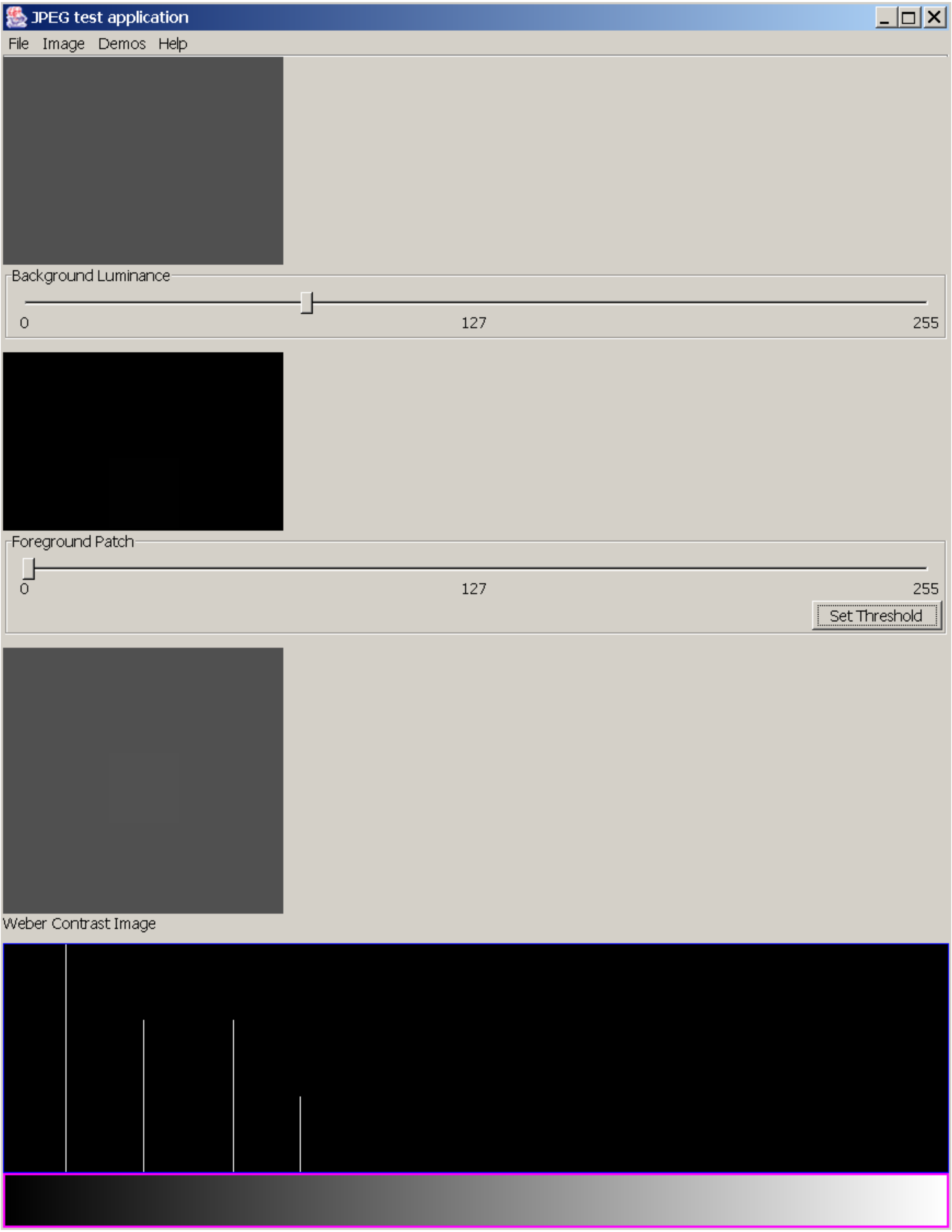

Figure B.2: Weber contrast interface

Weber Contrast

- *Figure*: B.2
- *Section*: 1.3.1, 3.1.1
- *Description*: This interface demonstrates the process of measuring Weber contrast, defined as the difference in luminance between a constant square patch and a constant background, when the patch is first visible over the background. The top square shows a constant background, whose luminance can be adjusted by the top scroll-bar. The middle square shows a constant square patch in the center of a constant background, whose luminance can be adjusted by the middle scroll-bar. The bottom square shows the combination of the background and the patch. As with the Michelson contrast interface, the threshold is plotted by adjusting the luminance of the patch until it is first visible in the combined image, then pushing the `Set threshold` button; repeating this experiment for various background luminance levels will give a plot of the Weber contrast, described by equation (3.1).
- *Parameters*: background luminance (top scroll-bar), luminance of foreground patch (bottom scroll-bar)
- *Additional software*: `XYPlot` class

Contrast Sensitivity Function

- *Figure*: B.3
- *Section*: 1.3.2, 3.1.2
- *Description*: This interface demonstrates the general shape of the contrast sensitivity function (CSF). Mannos and Sakrison [40] give a general form for the CSF, given in equation (3.4); this general form is given in terms of four parameters, whose values have varied dramatically in different experiments requiring CSF measurements.

  This interface provides two methods of displaying a CSF: the four scroll-bars (left) can be used to adjust the value of each parameter in the general form given by Mannos and Sakrison; and the combo-box (right) gives a number of built-in CSFs, some of which are shown in Table 3.1. Notice that the built-in CSF of Daly is not isotropic,

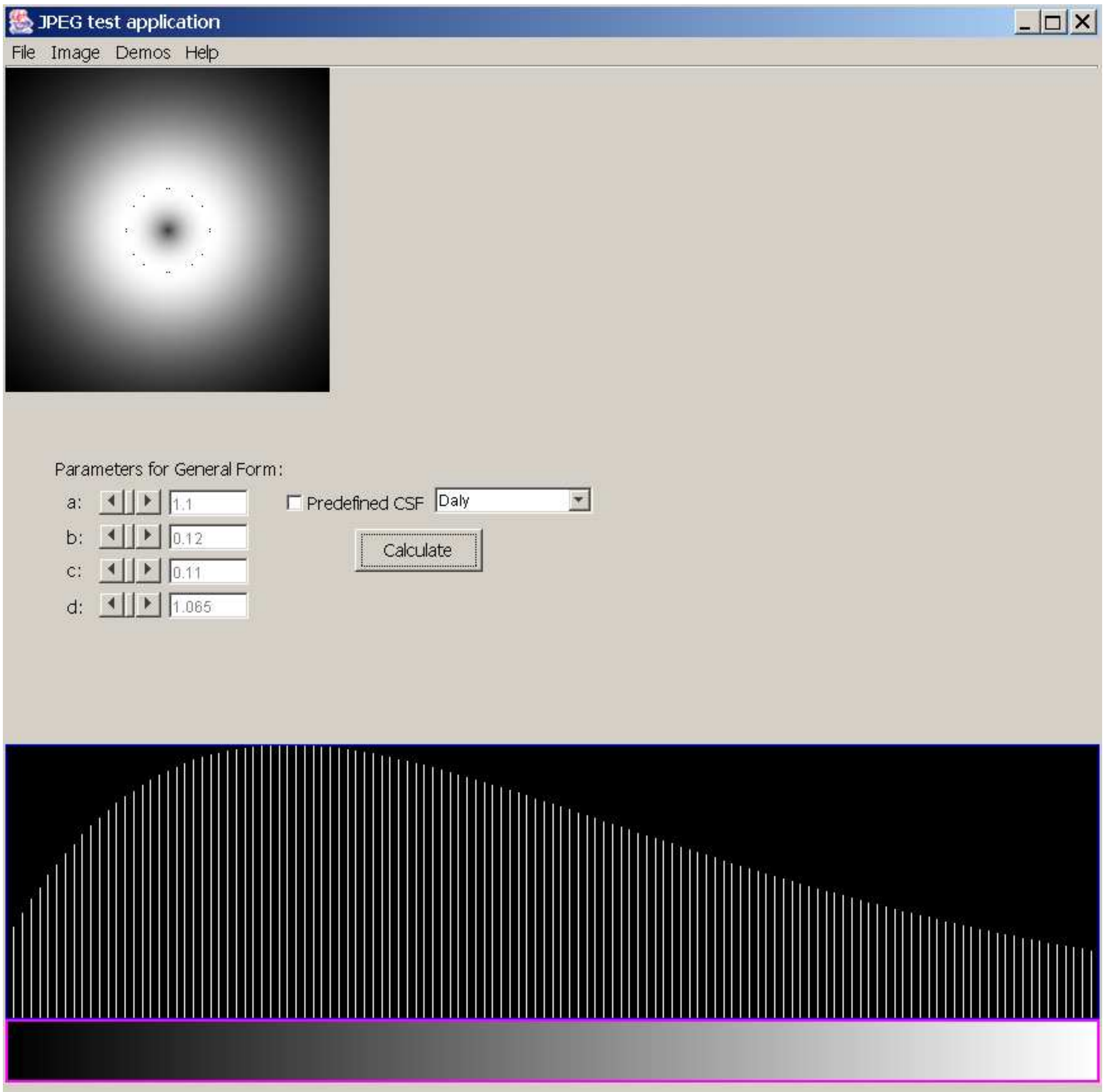

Figure B.3: Contrast sensitivity function interface

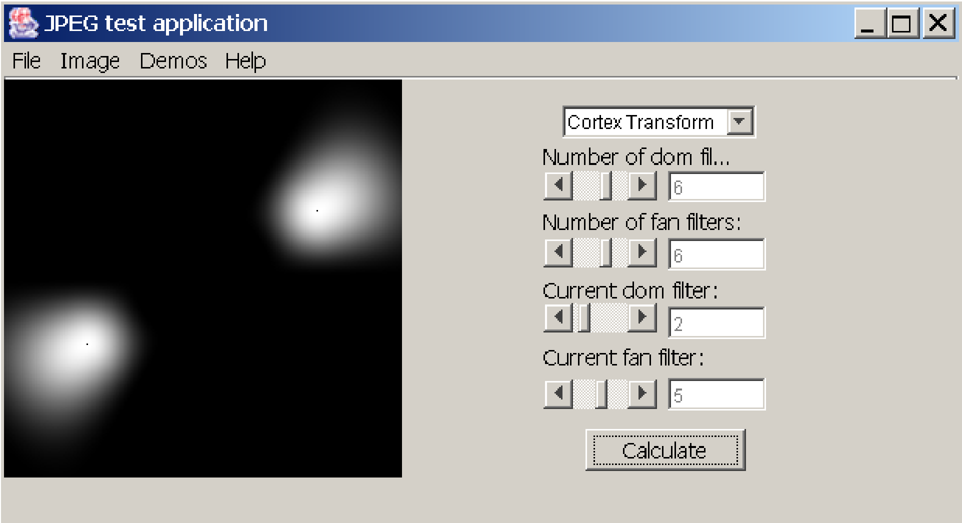

Figure B.4: Frequency decomposition interface

and thus cannot be created by adjusting the four parameters. For the isotropic CSFs, the one-dimensional plot for the CSF is displayed in the bottom plot window.

- *Parameters*: four parameters from (3.4) (four scroll-bars), and choice of built-in CSF (combo-box)
- *Additional software*: `XYPlot` class
- *To do*: provide additional adjustable parameters for diagonal frequencies to model non-isotropy of CSF

Frequency Decompositions

- *Figure*: B.4
- *Section*: 1.3.3, 3.1.3
- *Description*: This interface shows frequency decompositions commonly used in modelling contrast masking. Contrast masking occurs due to content in the original image having similar frequency and orientation as the artifacts present in the compressed image. As such, the process of measuring contrast masking involves partitioning the frequency plane into windows of similar frequency and orientation. Every method of

performing this partitioning includes parameters to specify the window, and most use parameters for the number of frequency and orientation windows within the partition.

This interface demonstrates a number of common frequency partitioning models; the parameters within each model specifying the number of frequency and orientation windows can be adjusted to observe the effect that each parameter has on the resulting partition.

- *Parameters*: Choice of partitions (top combo-box), number frequency and orientation windows (top two scroll-bars), and current frequency and orientation window (bottom two scroll-bars)
- *Additional software*: None
- *To do*: At present, only Daly's variation on Watson's Cortex Transform [81] is included in this interface; we would like to include other common frequency partitions, specifically the Laplacian pyramid of Burt and Adelson [12], and the Steerable filters of Freeman and Adelson [24] and Simoncelli, Freeman, Adelson and Heeger [68]. Additionally, we would like to include an image in this interface to show the frequency content of each image within each frequency window.

B.1.2 JPEG Demos

DCT

- *Figure*: B.5
- *Section*: 2.1.1
- *Description*: This interface demonstrates the effect that each DCT coefficient has on the visual appearance of an image. Shown are three images: the original image (top left), a compressed version of the original image (top right), and the absolute difference between the two images (bottom left). The compressed image is quantized using the elements of the included 8×8 quantization table (bottom right); the elements of this table can either be set manually by changing the entries within each textbox, or if the checkbox is selected then the scrollbar can be used to adjust the entire quantization table along the IJG quality scale. Of particular interest, changing a single entry of the table to a large number while leaving the remaining elements small shows the

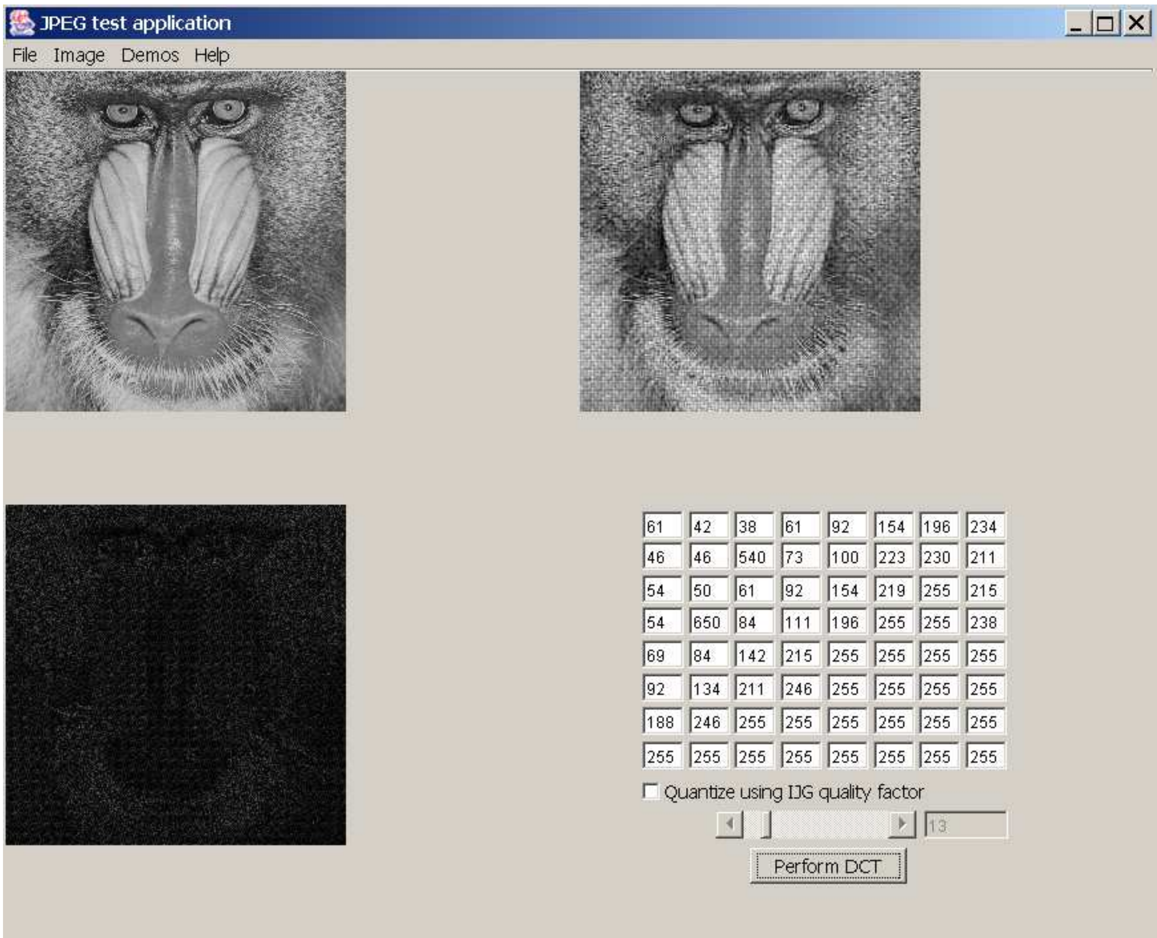

Figure B.5: DCT interface

effect that each DCT coefficient has on the visual appearance of the entire image; setting a large entry corresponds to removing all content of a particular frequency and orientation within each block.

- *Parameters*: The 8×8 quantization coefficients (bottom right), IJG quality factor (bottom scroll-bar)
- *Additional software*: None
- *To do*: An extension to the JPEG standard allows for *adaptive quantization*, where the quantization matrix can change throughout the image; each different quantization matrix throughout the image is a (rounded) scalar multiple of a given base quantization matrix, and scalar factors are used to define the change in the scalar multiple between adjacent blocks. The quantization matrix design method in Section 2.2.2 has been extended to adaptive JPEG encoding, so an interface that enables experimentation with the adaptive encoding model is of great interest. An extension of the present interface could use mouse input to increase or decrease the quantization per block.

Dipper

- *Figure*: B.6
- *Section*: 2.3.1
- *Description*: This interface demonstrates the heuristic algorithm presented in Section 2.3.1 as a solution to avoiding the speckling artifacts caused by the *JPEG Perceptual Dipper Effect*. This effect is caused by the over-estimations in the quantized DCT coefficients, resulting from requantizing an image that has already been JPEG-encoded at a higher bitrate.

  This interface shows three images and three quantization matrices to demonstrate the JPEG Perceptual Dipper: the first quantization matrix is the matrix used to encode a given original image; the second is the target quantization matrix for requantizing the image compressed using the first matrix; and the third is the matrix defined by the heuristic algorithm of Section 2.3.3, used to approximate the target matrix while avoiding the speckling artifacts caused by the Dipper Effect. The fourth matrix shows the entry-wise quotients between the original and target quantization matrices, which

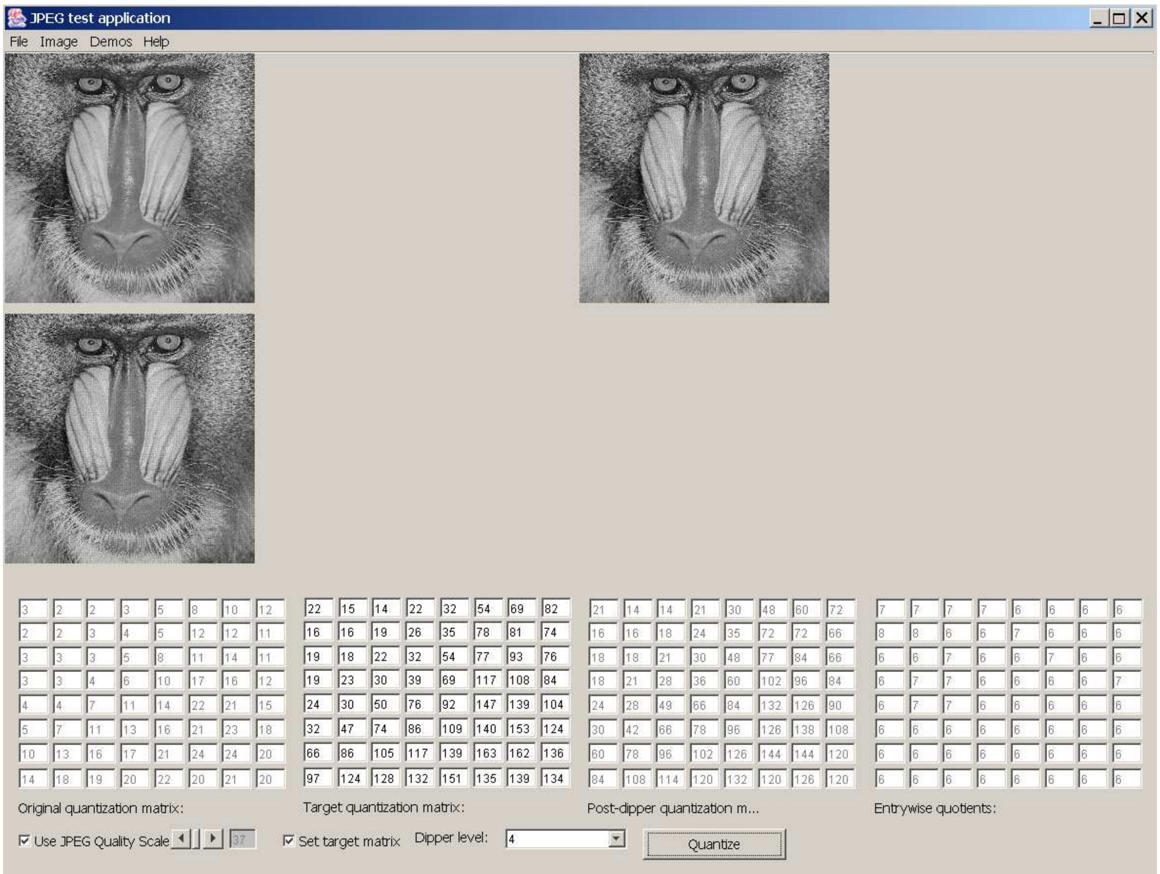

Figure B.6: Dipper interface

is used to define the third quantization matrix. The three images displayed are the first JPEG-encoded image (top left), the compressed image requantized using the matrix defined by the heuristic algorithm (top left), and the image quantized using the matrix with each entry 1 more than the corresponding entry in the matrix defined by the heuristic. We show this third image to demonstrate how specific the Dipper Effect is: differences are clearly visible between the two images despite the fact that they are compressed from the same source image using quantization matrices that are very similar.

- *Parameters*: Three 8×8 quantization matrices, and an integer used to define the dipper multiple (combo-box at bottom)
- *Additional software*: None
- *To do*: None

Summation

- *Figure*: B.7
- *Section*: 3.3
- *Description*: This interface plots each of the alternate summations described in Table 3.2. Each of the summations can be chosen from the combo-box, with additional parameters set when necessary using the top two scrollbars. Once a summation method is selected, pushing the `Perform summation` button displaying the plot in the bottom window, where the points plotted are the values of the measure evaluated for the original image and the images compressed using each value of the IJG quality scale. Since the vertical axis of the plot is normalized between 0 and the largest value, if more detail of a subsection of the IJG quality scale is desired, then the endpoints of the desired subsection can be selected using the two bottom scrollbars.
- *Parameters*: Summation method (top combo-box) with additional parameters for l_p- and σ-measure (top two scroll-bars), and endpoints for IJG quality scale (bottom two scrollbars, if checkbox is selected)
- *Additional software*: `XYPlot` class

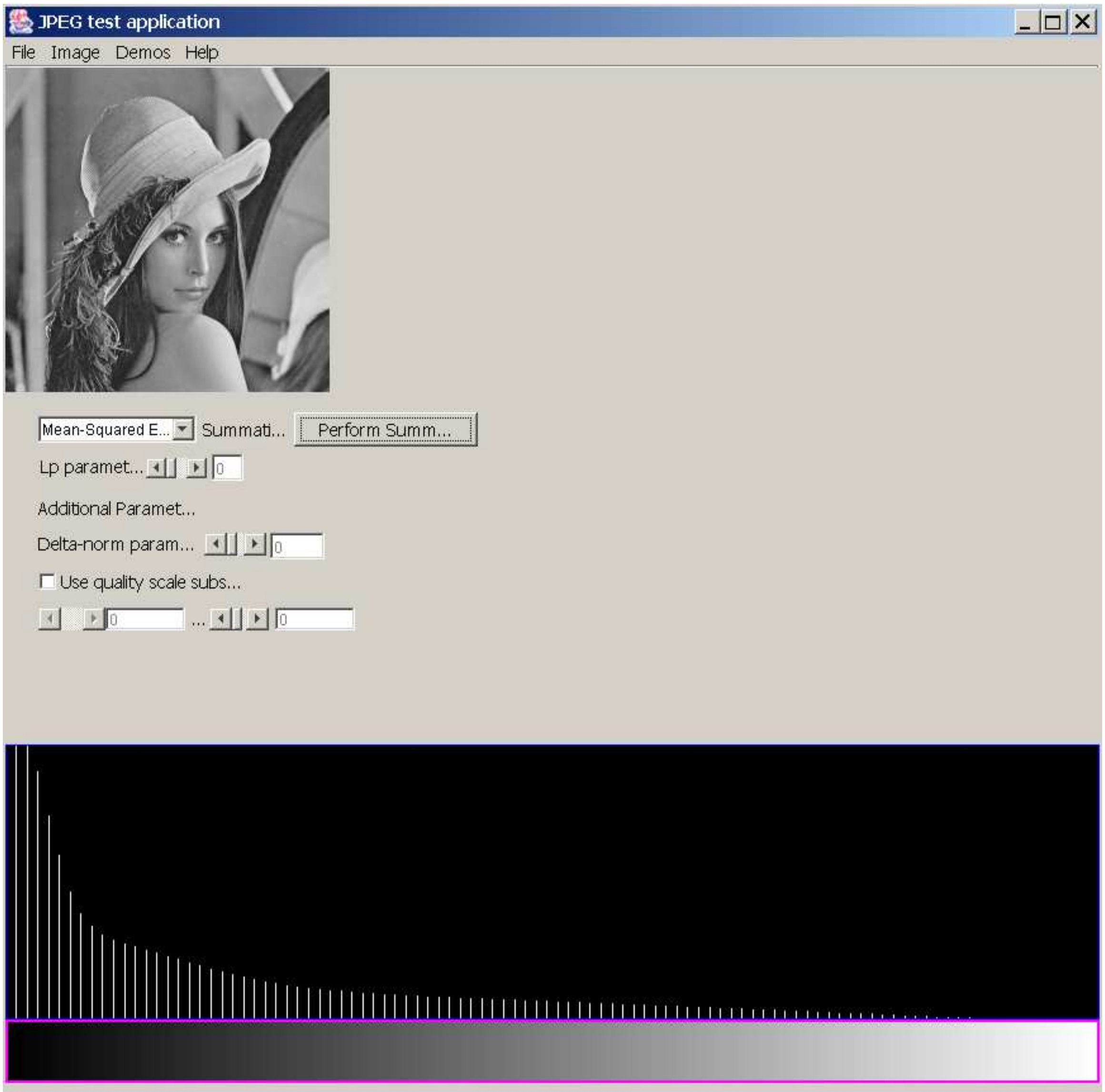

Figure B.7: Summation interface

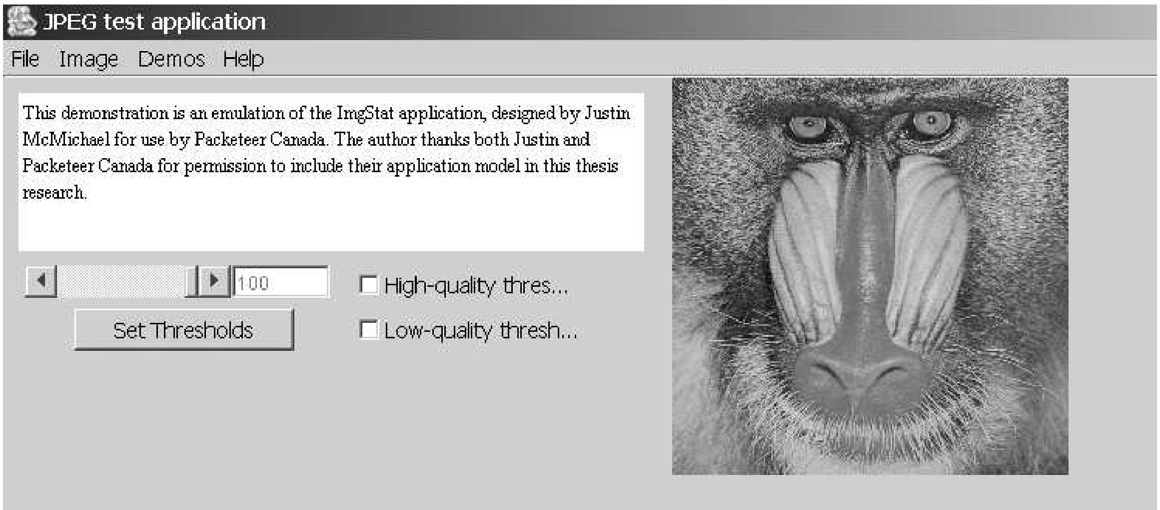

Figure B.8: Screenshot of application designed to measure visual thresholds

- *To do*: Include alternative domains, such as DCT and wavelet domains; in wavelet domain, include choice of summation over low-pass or high-pass coefficients

ImgStat

- *Figure*: B.8
- *Section*: 4.1
- *Description*: This interface demonstrates the method of measuring visual thresholds used in Section 4.1. The interface consists of a scrollbar which adjusts the IJG quality factor used to compress the image, and two checkboxes used to fix the values of the IJG quality factor at which a visual threshold is reached. In-context comparisons between the images are made by mouse-clicking on the image, which toggles between the original and compressed images; this allows the user to directly compare the visual appearance of the original and compressed images. The subject adjusts the scroll-bar until differences between the original and compressed images are first noticeable, at which point the subject selects the top checkbox to indicate that the high-quality threshold has been reached. The subject further adjusts the scrollbar until differences between the two images become annoying, then selects the bottom checkbox to indicate that the low-quality threshold has been reached. Pushing the `Set thresholds` button

writes the IJG quality factors corresponding to both thresholds into a text-file in the image's directory.

- *Parameters*: IJG quality scale (scroll-bar), high- and low-quality thresholds (top and bottom checkboxes)
- *Additional software*: None
- *To do*: Allow for mouse-input to be captured at points where the subject indicates the errors are noticeable.
- *Note*: This interface is a Java-emulation of the original `ImgStat` application, written by Justin McMichael for use by Packeteer (`www.packeteer.com`). Special thanks to both Justin and Packeteer for permission to include the `ImgStat` discussion in this thesis research.

B.1.3 Multidimensional Scaling

Kruskal algorithm

- *Figure*: B.9
- *Section*: 4.3.3
- *Description*: This interface demonstrates Kruskal's algorithm for calculating a configuration with inter-point distances approximating a given dissimilarity matrix. For a given original image (not shown in interface) and a given summation method (see Table 3.2), four matrices are calculated: the *dissimilarity* matrix (top left) is generated by the summation method applied to a connected subset of length 10 of the IJG quality scale; the *configuration* matrix (center left) is the matrix of coordinates of the current iterate, of dimension at most 10; the *distance* matrix (center right) is the matrix of distances between the points in the current configuration; and the *disparity* matrix (bottom left), is the set of points closest in structure to the distances while preserving the rank ordering of the dissimilarities.

  For a selected summation method, when the `Calculate error` button is pressed, the dissimilarities are calculated, as is the initial configuration, while the stress of the current configuration is displayed in the top left. Subsequent iterates are calculated by

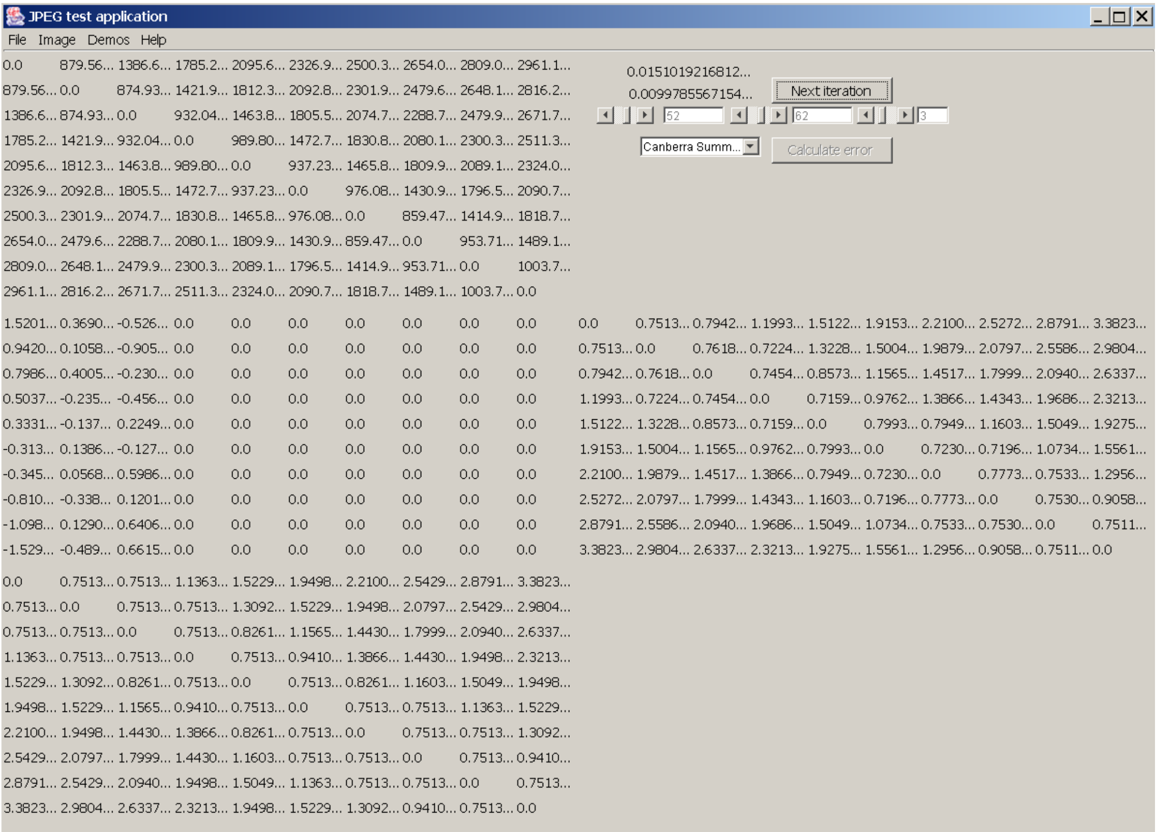

Figure B.9: Kruskal's algorithm interface pressing the `Next iteration` button; the stepsize used to update the current iterate is also displayed in the top left.

- *Parameters*: Lower- and upper-bounds on the IJG quality factor (left and middle scrollbars), summation method (combo-box) and number of dimensions in configuration (right scrollbar).
- *Additional software*: None
- *To do*: None

Procrustes Analysis

- *Figure*: B.10
- *Section*: 4.3.4
- *Description*: This interface demonstrates both of the algorithms proposed in Section 4.3.5. For a given original image, the top experiment constructs the disimilarity matrix generated by the selected summation method, as well as the matrix defined by (4.76) for the observed high- and low-quality thresholds. The configurations for both matrices are calculated separately by pushing either of the `Next Iter` buttons; once a configuration of sufficiently small stress constructed, pressing the `Compare` button performs a Procrustes Analysis to determine the degree of similarity between the two configurations.

  The bottom half of this interface performs the same experiment, using as the dissimilarity matrices two user-selected summation methods. The same procedure is performed to compare the resulting configurations.

- *Parameters*: (Threshold experiment, top): summation method (combo-box), with parameters for l_p and σ-threshold summations (top two scroll-bars); lower- and upper-bounds on the IJG quality factor (bottom-left scrollbars) and number of dimensions in configuration (right scroll-bar).

  (MSE experiment, bottom): summation methods for structural comparison (top two comboboxes), with remaining parameters the same as the threshold experiment.

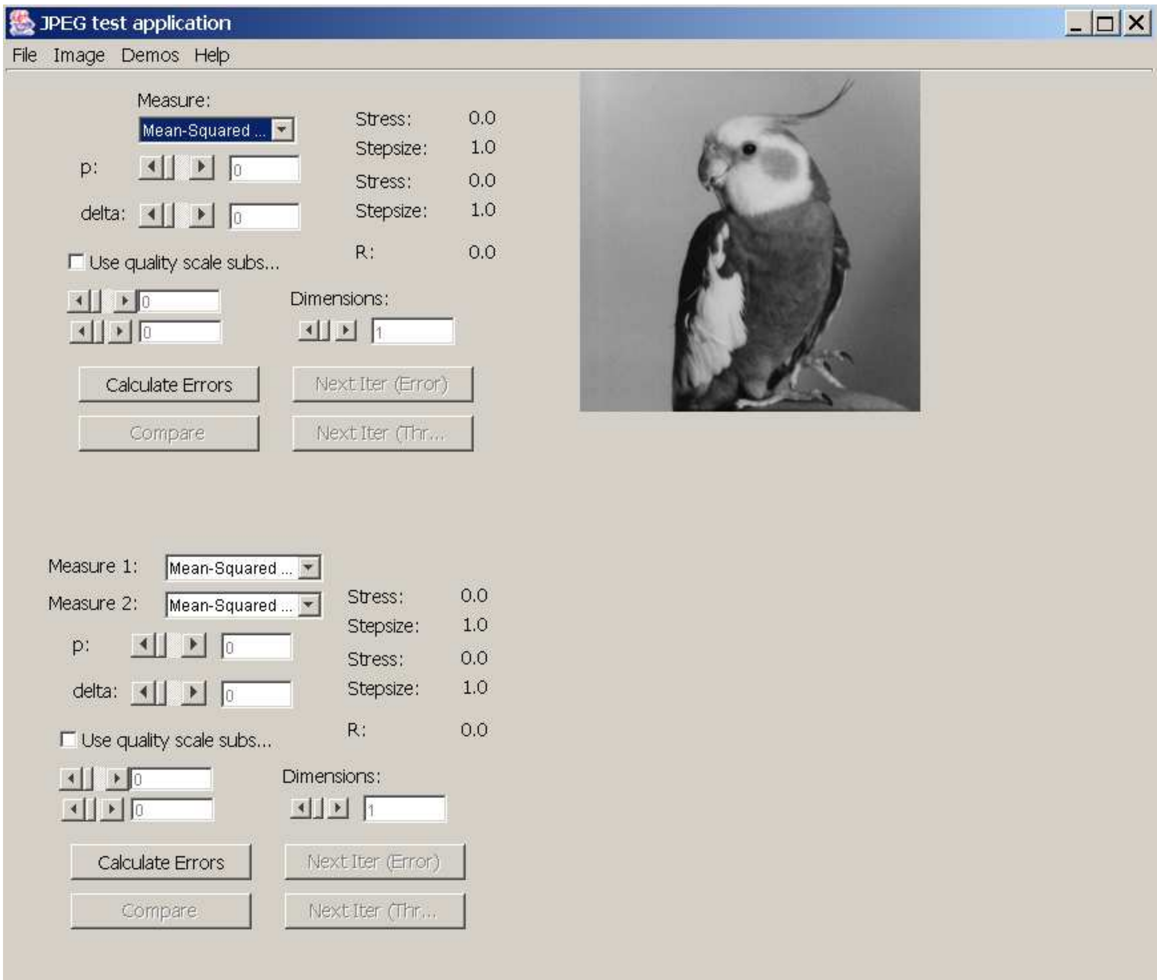

Figure B.10: Procrustes Analysis interface

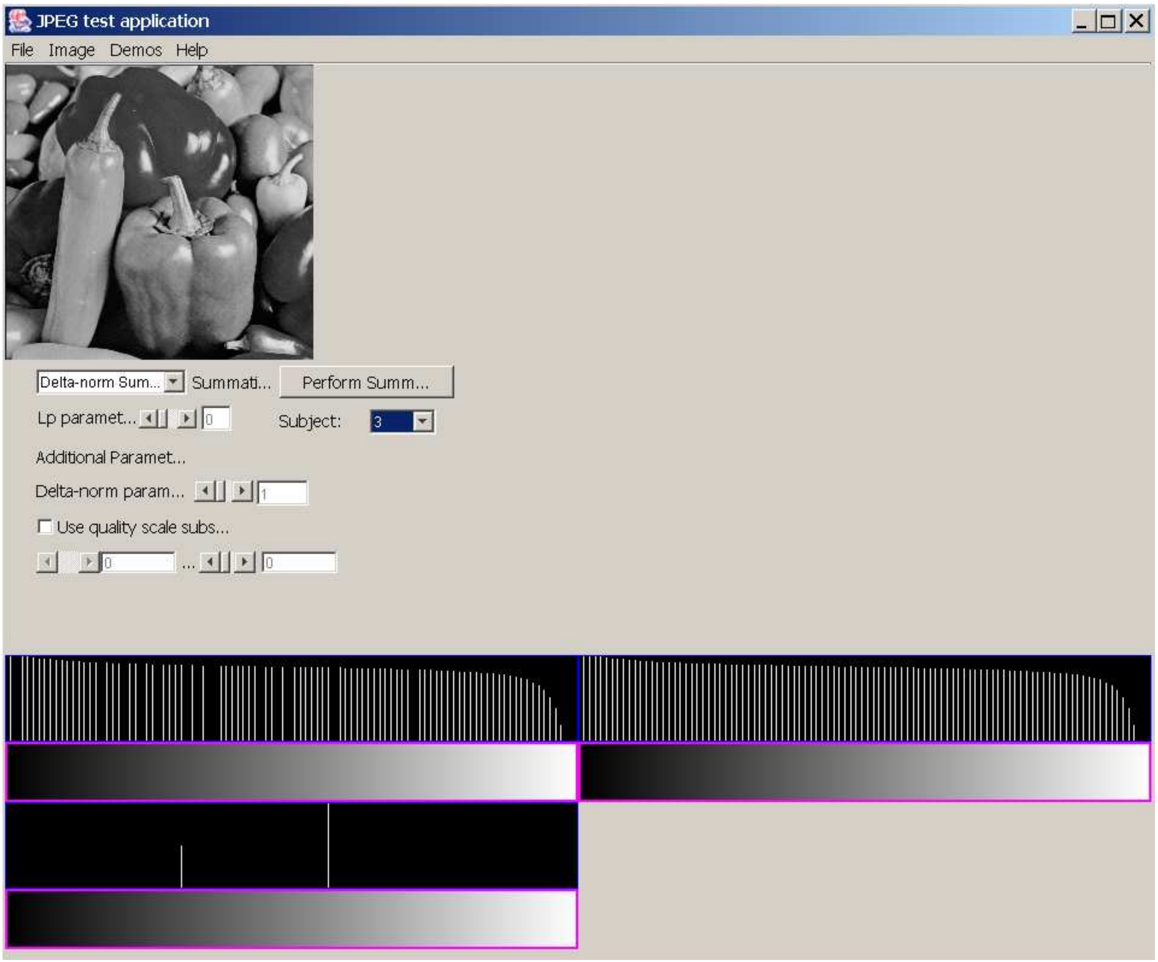

Figure B.11: Non-dominated set generation interface

- *Additional software*: Jama Matrix package for Java [30], used to invert the matrices and calculate the singular value decompositions, as described in Section 4.3.4.
- *To do*: None

B.1.4 Multicriteria Optimization

Nondominated Set (Pareto)

- *Figure*: B.11
- *Section*: 4.4

- *Description*: This interface generates the nondominated set of a given summation method, using as the multiple objectives the given summation and the IJG quality scale. The left-hand graph shows the graph of the method applied to the given image compressed over the IJG quality scale. The top right-hand graph shows the nondominated set of this summation method, with spaces in the graph corresponding to dominated points on the graph. The bottom graph shows the high- and low-quality thresholds for a selected subject, to enable in-context comparison between the observed threshold and the nondominated set.
- *Parameters*: summation method (top combo-box), choice of subject (right-hand combo-box), and IJG quality scale subset (bottom two scroll-bars)
- *Additional software*: None
- *To do*: None

Weighting Method

- *Figure*: B.12
- *Section*: 4.4.3
- *Description*: This interface demonstrates the weighting method used to generate the nondominated set of a group of three summation methods. The three graphs to the left of the screen are the graphs of three user-selected summations. The top right-hand graph is the weighted sum of the three graphs, with weights defined using the bottom scroll-bars under each summation combo-box. The middle right-hand graph is the nondominated set of the weighted sum; the bottom graph shows the high- and low-quality thresholds for a selected subject, to enable in-context comparison between the observed threshold and the nondominated set.
- *Parameters*: three summation methods (top combo-boxes), weights for each summation method (bottom scrollbars), choice of subject (bottom combo-box)
- *Additional software*: `XYPlot` class
- *To do*: Need to normalize the summations, to avoid one summation dominating the other two due to it having a larger value.

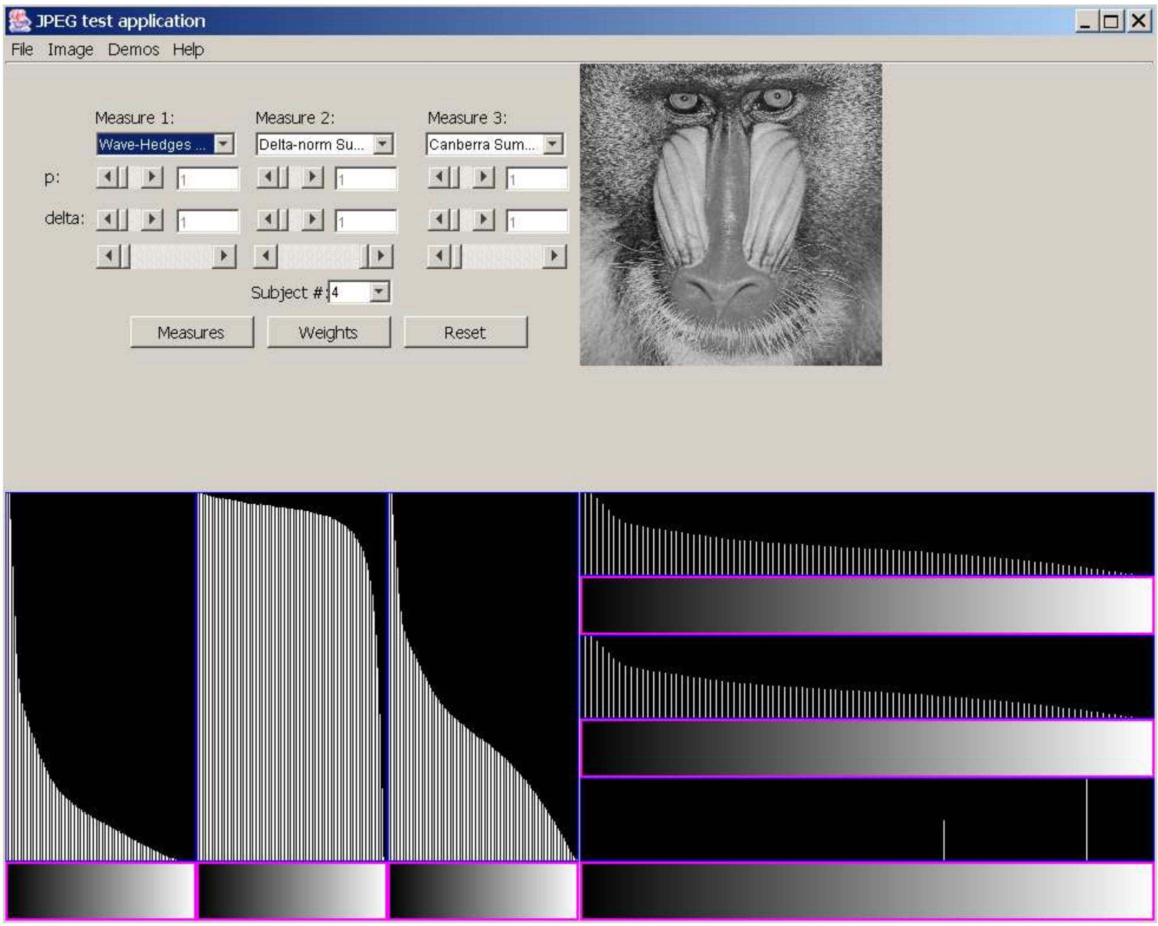

Figure B.12: Weighting method interface

Constraint Method

- *Figure*: B.13
- *Section*: 4.4.3
- *Description*: This interface demonstrates both the weighting and constraint methods of calculating the nondominated set. For a given summation method, the two objectives are the summation and the IJG quality scale. For the weighting method, adjusting the top scroll-bar adjusts the weight applied to the summation objective for calculating the high-quality threshold, while the bottom scrollbar defines the percentage of the top scrollbar value used to weight the combined objective function to calculate the low-quality threshold. Selecting the `inverse` radio-button defines the weight as the reciprocal of the scrollbar value, with the low-quality threshold weight defined similarly.

  For the constraint method, the top two scrollbars set the upper and lower bounds on the summation method, while the bottom two scrollbars set the upper and lower bounds on the IJG quality factor. Pressing the `Calculate threshold` button calculates both the high- and low-quality thresholds, which are shown at the bottom of the menu next to the observed thresholds for the given image.
- *Parameters*: Summation method (top scrollbar), weights for combined objective function for high- and low-quality threshold calculations (top and bottom scrollbars in `Weights` submenu), upper and lower bounds on the summation method and IJG quality factor (top and bottom pairs of scrollbars in `Constraints` submenu).
- *Additional software*: `XYPlot` class
- *To do*: Use filesize rather than IJG quality scale as second objective.

B.1.5 Quality Models

Visible Differences Predictor (VDP)

- *Figure*: B.14
- *Section*: 3.1

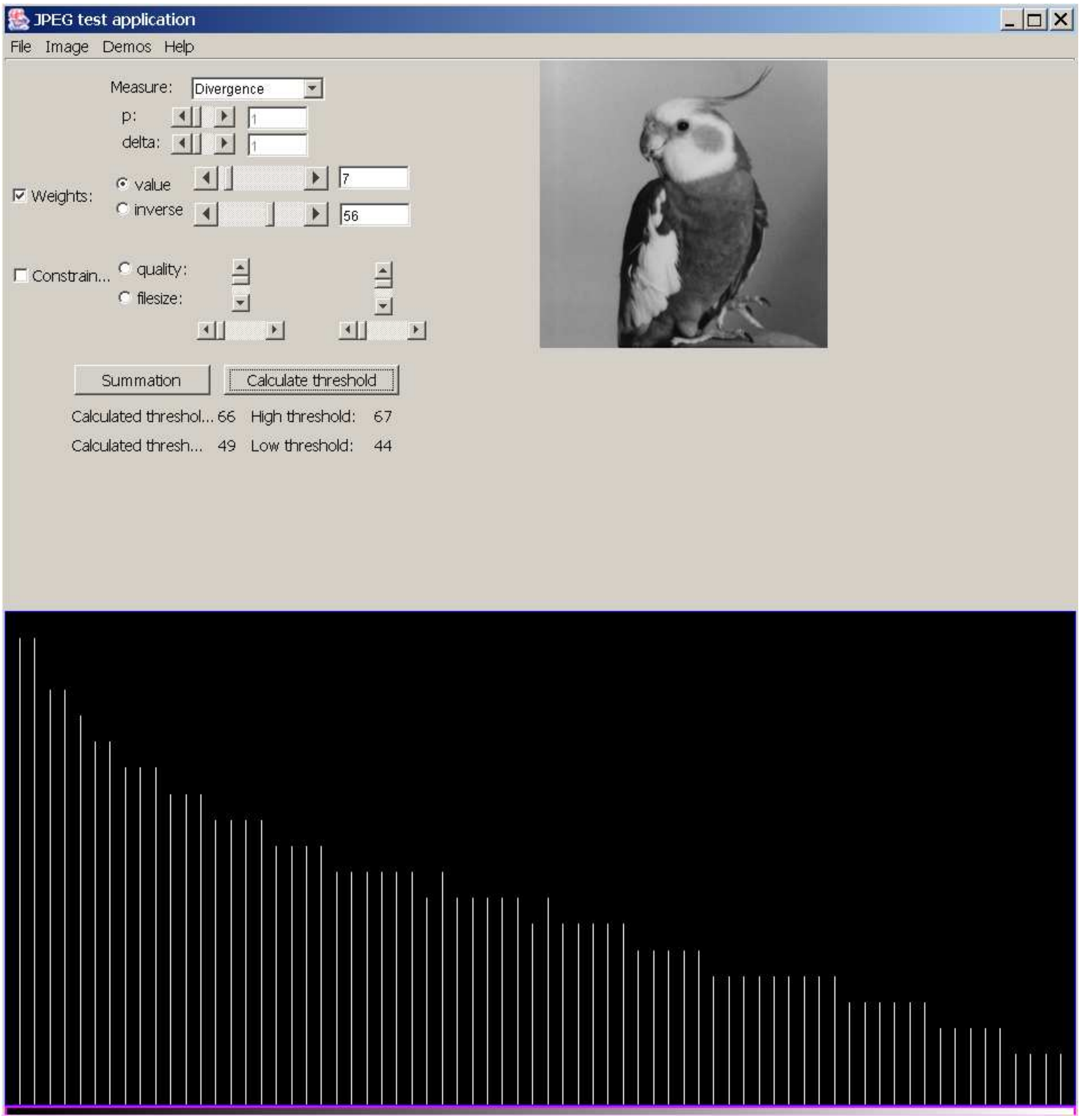

Figure B.13: Constraint method interface

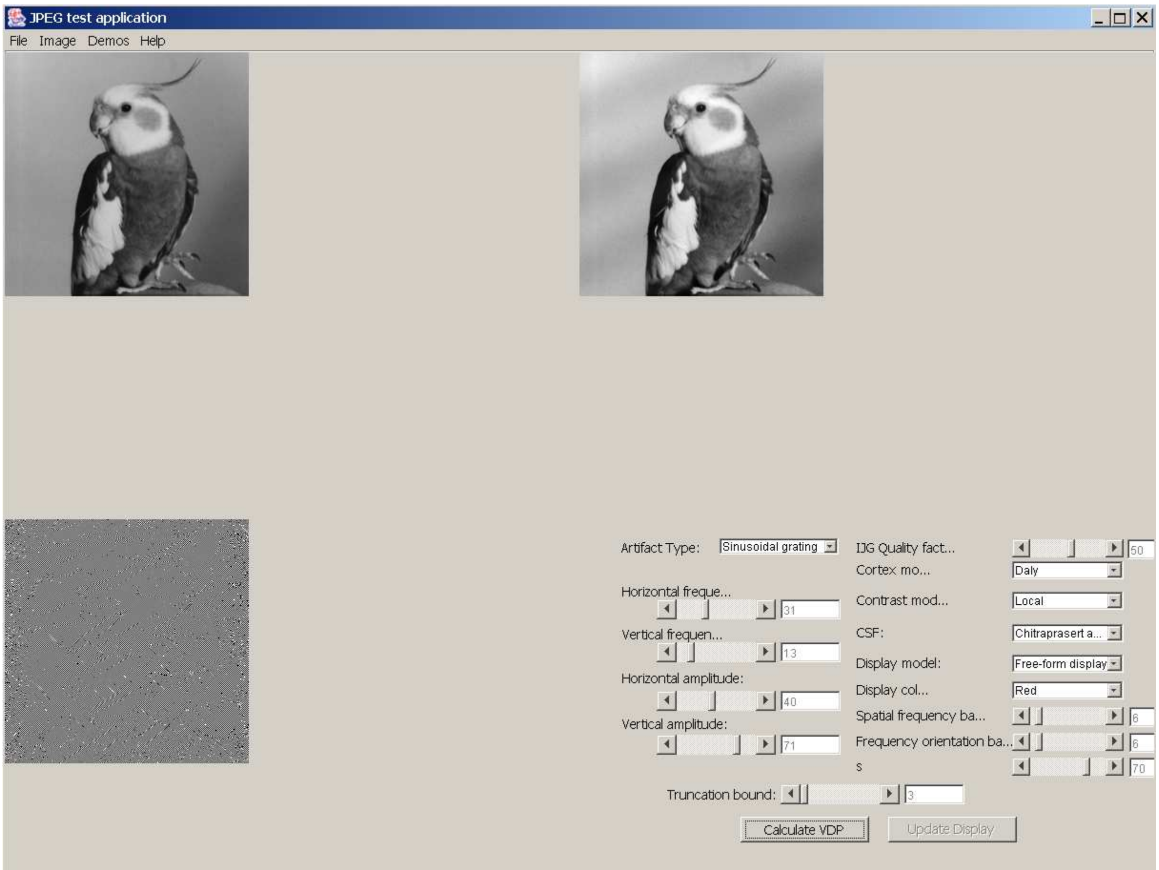

Figure B.14: Visible Differences Predictor interface

- *Description*: This interface demonstrates Daly's Visible Differences Predictor [20]. For a given image, there are two types of artifacts available, sine gratings and JPEG artifacts. For the sine gratings, the three scrollbars adjust the horizontal and vertical frequencies, and the grating amplitude. For the JPEG artifacts, the IJG quality factor is set using the top-right scrollbar.

  The remaining combo-boxes set the calculation and display parameters for the VDP. The combo-boxes set the cortex model used to partition the frequency plane, the contrast model used to convert luminance values into units of contrast, the CSF, the display model used to show the VDP map, and the color-plane used to display the in-context map (if appropriate). The scrollbars set the number of spatial frequency and frequency orientation bands, and the parameter s used to set the slope of the transition band in the psychometric function.

- *Parameters*: See Description for list of parameters.

- *Additional software*: FFT routine from Numerical Recipes [57]

- *To do*: Include Watson and Osberger cortex models.

Region-of-Interest (ROI)

- *Figure*: B.15

- *Section*: 3.2.1

- *Description*: This interface demonstrates the Region-of-Interest model of Osberger, based on a model of visual attention. The four scroll-bars adjust the split and merge variance-thresholds, and the minimum split and merge blocksizes; the split variance thresholds defines the maximum thresholds within a block for that block to be considered a distinct object, while the merge threshold defines the maximum variance for two distinct blocks to define a single merged object. The minimum split blocksize defines the minimum size of a block during the segmentation stage (used to set stopping depth for split stage), while the minimum merge blocksize defines the minimum size an object must be to remain distinct if small objects are merged (all blocks below the minimum blocksize are merged with their largest neighbouring object).

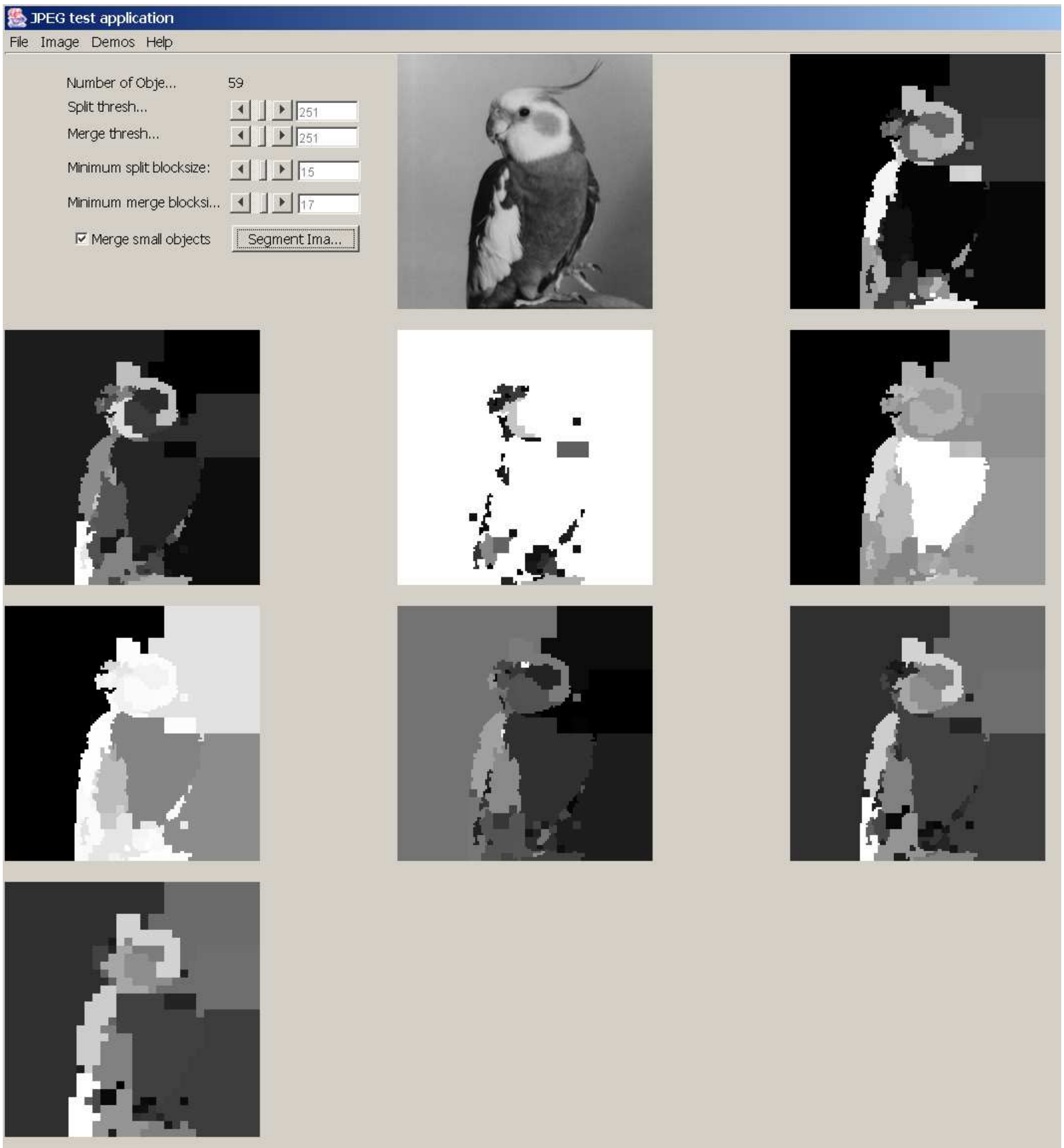

Figure B.15: Region-of-Interest interface

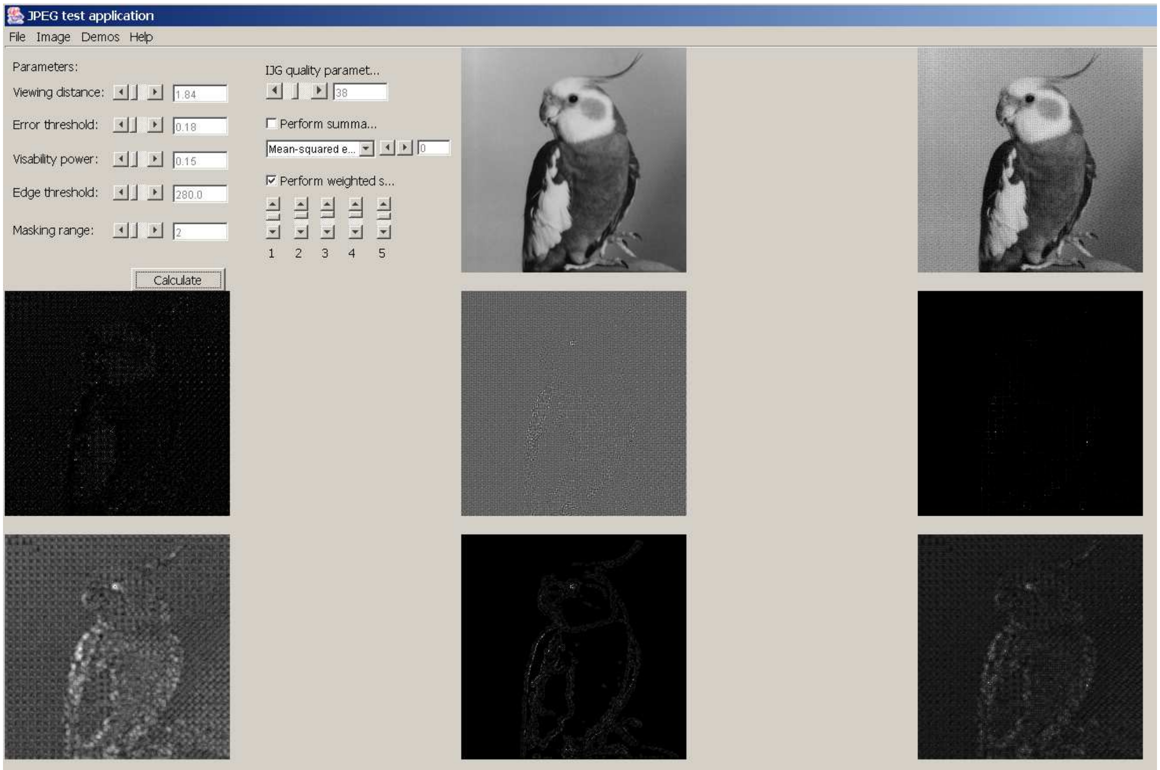

Figure B.16: Picture Quality Scale interface

- *Parameters*: Split and merge variances (top two scrollbars), minimum split and merge blocksizes (bottom two scrollbars), setting for merging small objects with largest neighbouring object (bottom checkbox).
- *Additional software*: None
- *To do*: Include additional segmentation criteria, as described in Chapter 5.

Picture Quality Scale (PQS)

- *Figure*: B.16
- *Section*: 4.4.1
- *Description*: This interface demonstrates the Picture Quality Scale of Miyahara,

Kotani and Algazi [43]. The image displayed are (from top, left to right): the original image, the compressed image using the selected IJG quality factor, error images containing the pixel-wise mean-squared error, thresholded mean-squared error, blocking-artifacts, error correlated across adjacent pixel-regions, and errors masked by high-contrast edges, along with a combined error map. The weights of the five maps in the combined map can be set using the five vertical scrollbars. The five horizontal scrollbars set values for parameters used to calculate the five error maps.

- *Parameters*: IJG quality factor (top right scrollbar), viewing distance, error threshold, visability power, edge threshold and masking range (five left-hand scrollbars), summation method used to combine five maps into single perceptual map (right-hand combo-box), weights used to combine five error maps into single error map (five right-hand vertical scrollbars)
- *Additional software*: Original code to calculate PQS factors [43], FFT routine.
- *To do*: None

B.1.6 Other

Topological Comparisons

- *Figure*: B.17
- *Section*: 3.3.1
- *Description*: This interface performs the topological comparisons described in Section 3.3.1. For each of the summations in Table 3.2, selecting the corresponding checkbox will evaluate the three topological measures, while selecting a perceptual map will use it as a pixel-wise weighting. Note that the VDP measure takes quite a while to calculate.
- *Parameters*: Summation method (left-hand checkboxes) and perceptual map (right-hand combo-box).
- *Additional software*: None
- *To do*: None

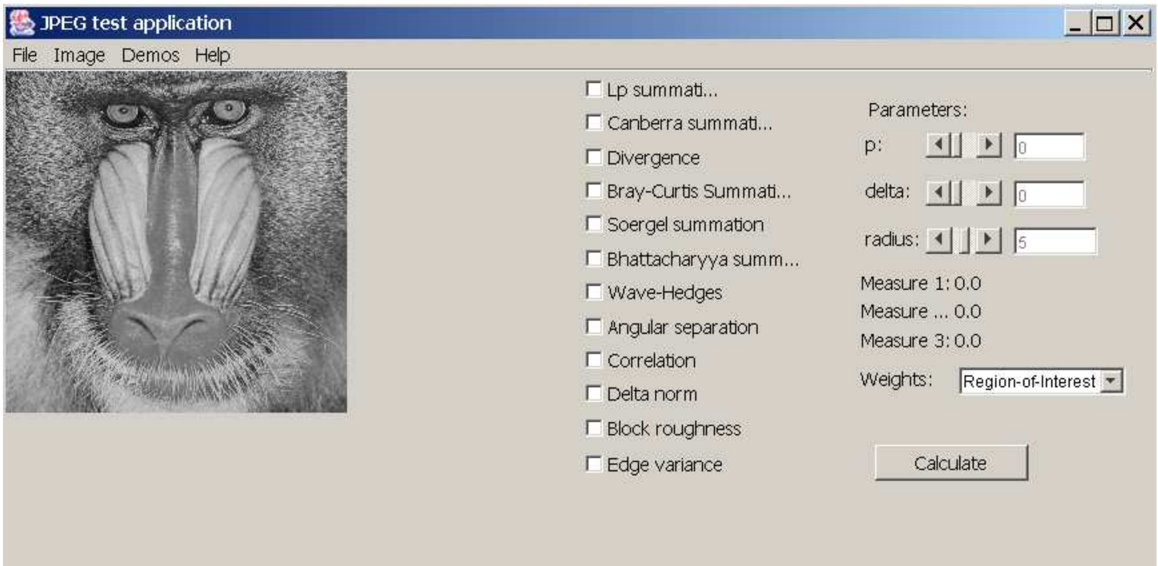

Figure B.17: Topological Comparisons interface

B.2 Webpages

The webpages showing phenomena discussed in this thesis that do not easily translate to print can be found at `http://www.cecm.sfu.ca/~mmacklem/thesis`. In particular, these include the images that illustrate the JPEG Perceptual Dipper Effect, which are located at `http://www.cecm.sfu.ca/~mmacklem/thesis/dipper.htm`.

Bibliography

[1] Albert J. Ahumada. Computational image quality metrics: A review. In J. Morreale, editor, *Society for Information Display International Symposium Digest of Technical Papers*, volume 24, pages 305–308, 1993.

[2] Albert J. Ahumada and Rensheng Horng. De-blocking DCT compressed images. In B. Rogowitz and J. Allebach, editors, *Human Vision, Visual Processing, and Digital Display V*, volume SPIE Proc. Vol. 2179, pages 109–116, 1994.

[3] Albert J. Ahumada and Rensheng Horng. Smoothing DCT compression artifacts. In B. Rogowitz and J. Allebach, editors, *Human Vision, Visual Processing, and Digital Display V*, volume SPIE Proc. Vol. 2179, pages 109–116, 1994.

[4] Albert J. Ahumada and Cynthia H. Null. Image quality: A multidimensional problem. In Albert B. Watson, editor, *Digital Images and Human Vision*, pages 141–148. MIT Press, 1993.

[5] Albert J. Ahumada and Heidi A. Peterson. Luminance-model-based DCT quantization for color image compression. In *Human Vision, Visual Processing, and Digital Display III*, pages 365–374, 1992.

[6] J. Allnatt. *Transmitted-Picture Assessment.* Wiley and Sons, 1983.

[7] H.H. Bauschke, C.H. Hamilton, M.S. Macklem, J.S. McMichael, and N.R. Swart. Recompression of JPEG images by requantization. 2001, submitted to IEEE Transactions on Image Processing.

[8] H.H. Bauschke, C.H. Hamilton, M.S. Macklem, J.S. McMichael, and N.R. Swart. A requantization-based method for recompressing JPEG images. In *Proceedings of the 27th IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2489–2492, Orlando, Florida, May 13-17 2001.

[9] J. Borwein. Proper efficient points for maximizations with respect to cones. *SIAM Journal of Control and Optimization*, 15:57–63, 1977.

[10] J. Borwein. On the existence of pareto efficient points. *Mathematics of Operations Research*, 8:64–73, 1983.

[11] W.L. Briggs and V.E. Henson. *The DFT: an owner's manual for the discrete fourier transform.* SIAM, Philadelphia, 1995.

[12] P.J. Burt and E.H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Information Theory*, 31:532–540, 1983.

[13] J.Douglas Carroll and Myron Wish. Multidimensional perceptual models and measurement methods. *Handbook of Perception*, 2:395–412, 1974.

[14] Syin Chan. Recompression of still images. Technical Report 2-92*, University of Kent, Canterbury, UK, March 1992.

[15] Syin Chan. *The use of the JPEG Image Compression Standard and the Problems of Recompression.* PhD thesis, University of Kent, December 1992.

[16] Vira Chankong and Yacov Y. Haimes. *Multiobjective Decision Making: Theory and Methodology.* North Holland: New York, 1983.

[17] J.L. Cohon. *Multiobjective Programming and Planning.* Academic Press, New York, 1978.

[18] T.N. Cornsweet. *Visual Perception.* Academic Press, New York, 1970.

[19] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling.* Chapman & Hall, New York, 2000.

[20] S. Daly. The visible differences predictor: an algorithm for the assessment of image fidelity. In A.B. Watson, editor, *Digital Images and Human Vision*, pages 179–206. MIT Press, Cambridge, 1993.

[21] J. de Leeuw. Applications of convex analysis to multidimensional scaling. In J.R. Barra, F. Brodeau, G. Romier, and B. van Cutsem, editors, *Recent developments in Statistics*, pages 133–145. Amsterdam, The Netherlands: North-Holland, 1977.

[22] M.J. Fabian, P. Habala, P. Hajek, J. Pelant, and V.M. Santalucia. *Functional Analysis and Infinite Dimensional Geometry.* Number 8 in CMS Books in Mathematics. Springer Verlag, New York, 2001.

[23] Y. Fisher, editor. *Fractal Image Compression: Theory and Application.* Springer-Verlag, New York, 1995.

[24] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[25] M. L. Hilton, Bjorn D. Jawerth, and A. Sengupta. Compressing still and moving images with wavelets. *Multimedia Systems*, 2(5):218–227, 1994.

[26] Rensheng Horng and Jr. Albert J. Ahumada. A fast DCT block smoothing algorithm. In *Visual Communication and Image Processing Annual Meeting Proceedings*, volume 2501, 1995.

[27] Steven L. Horowitz and Theodosis Pavlidis. Picture segmentation by a directed split-and-merge procedure. In *Proceedings of 2nd International Joint Conference on Pattern Recognition*, pages 424–433, Aug. 13-15 1974.

[28] C-L Hwang and A.S.M. Masud. *Multiple Objective Decision Making - Methods and Applications.* Springer-Verlag, New York, 1979.

[29] Independent JPEG Group. *Various documents.* `http://www.ijg.org`, accessed December 6, 2002.

[30] Jama Java Matrix Package. *API.* `http://math.nist.gov/javanumerics/jama/`, accessed December 6, 2002.

[31] T.J.W.M. Janssens. *Computational Image Quality.* Ph.D. Thesis, IPO, Center for User-System Interaction, Eindhoven, 1999.

[32] Java Advanced Imaging (JAI). `http://java.sun.com/products/java-media/jai`, accessed December 6, 2002.

[33] Joint Photographic Experts Group. *Various documents and links.* `http://www.jpeg.org`, accessed December 6, 2002.

[34] J.B. Kruskal and M. Wish. *Multidimensional Scaling.* Sage Publications, London, 1978.

[35] Joseph B. Kruskal. Multidimensional scaling by optimizing goodness-of-fit to a non-metric hypothesis. *Psychometrika*, 29:1–27, 1964.

[36] Joseph B. Kruskal. Non-metric multidimensional scaling: a numerical method. *Psychometrika*, 29:115–129, 1964.

[37] Joseph B. Kruskal. Monotone regression: continuity and differentiability properties. *Psychometrika*, 36:57–62, 1971.

[38] Edmund Y. Lam and Joseph W. Goodman. A Mathematical Analysis of the DCT Coefficient Distributions for Images. *IEEE Transactions on Image Processing*, 9(10):1661–1666, October 2000.

[39] J. Lubin. The use of psychophysical data and models in the analysis of display system performance. In A.B. Watson, editor, *Digital Images and Human Vision*, pages 179–206. MIT Press, Cambridge, 1993.

[40] J.L. Mannos and D.J. Sakrison. The effects of a visual fidelity criterion on the encoding of images. *IEEE Trans. Information Theory*, 20:525–536, 1974.

[41] Jean-Bernard Martens. Multidimensional modeling of image quality. *Proceedings of the IEEE*, 90(1):133–153, January 2002.

[42] John Miano. *Compressed image file formats: JPEG, PNG, GIF, XBM, BMP*. ACM Press, 1999.

[43] M. Miyahara, K. Kotani, and V.R. Algazi. Objective picture quality scale (PQS) for image coding. *IEEE Trans. Comm.*, 46:1216–1226, 1998.

[44] Moving Pictures Experts Group. *Various documents and links.* `http://mpeg.telecomitalialab.com`, accessed December 6, 2002.

[45] K.N. Ngan, K.S. Leong, and H. Singh. Adaptive cosine transform coding of images in perceptual domain. *IEEE Transactions on ASSP*, 37:1743–1750, 1989.

[46] N.B. Nill. A visual model weighted cosine transform for image compression and quality assessment. *IEEE Trans. Comm.*, 33:551–557, 1985.

[47] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Verlag: New York, 1999.

[48] Octave programming language. *Homepage*. `http://www.octave.org`, accessed December 6, 2002.

[49] W. Osberger. *A Computational model of the human visual system for image quality assessment*. Ph.D. Thesis, Queensland University of Technology, 1999.

[50] W. Osberger and A.M. Rohaly. Automatic detection of regions of interest in complex video sequences. In B.E. Rogowitz and T.N. Pappas, editors, *Human Vision and Electronic Imaging VI*, volume 4299 of *Proc. SPIE*, pages 361–372, 2001.

[51] T.N. Pappas and R.J. Safranek. Perceptual criteria for image quality evaluation. In A. Bovik, editor, *Handbook of Image and Video Processing*, pages 669–684. Academic Press, London, 2000.

[52] E. Peli. Contrast in complex images. *Journal of the Optical Society of America A*, 7:2032–2040, 1990.

[53] William B. Pennebaker and Joan L. Mitchell. *JPEG still image data compression standard*. Van Nostrand Reinhold, 1993.

[54] H. A. Peterson, H. Peng, J. H. Morgan, and W. B. Pennebaker. Quantization of color image components in the DCT domain. In B. E. Rogowitz, M. H. Brill, and J. P. Allebach, editors, *Proceedings of SPIE*, volume 1453 of *Human Vision, Visual Processing, and Digital Display*, pages 210–222, 1991.

[55] Heidi A. Peterson, Albert J. Ahumada, and Andrew B. Watson. An improved detection model for DCT coefficient quantization. In *SPIE Proceedings*, volume 1913, pages 191–201, 1993.

[56] Charles Poynton. *Various colour-related documents.* `http://www.inforamp.net/~poynton`, accessed December 6, 2002.

[57] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1988.

[58] J.R. Price and M. Rabbani. Biased reconstruction for JPEG encoding. *IEEE Signal Processing Letters*, 6(12):297–299, December 1999.

[59] K.R. Rao and P. Yip. *Discrete Cosine Transform - Algorithms, Advantages, Applications.* Academic Press, New York, 1990.

[60] Todd R. Reed. Local frequency representations for image sequence processing and coding. In A.B. Watson, editor, *Digital Images and Human Vision*, pages 3–12. MIT Press, Cambridge, 1993.

[61] Randall C. Reininger and Jerry D. Gibson. Distributions of the Two-Dimensional DCT Coefficients for Images. *IEEE Transactions on Communications*, COM-31(6):835–839, June 1983.

[62] Ruth Rosenholtz and Andrew B. Watson. Perceptual adaptive JPEG coding. In *IEEE International Conference on Image Processing*, volume 1, pages 901–904, 1996.

[63] Walter Rudin. *Principles of Mathematical Analysis.* McGraw-Hill, 1953.

[64] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization.* Academic Press, 1985.

[65] Roger N. Shepard. The analysis of proximities: multidimensional scalign with an unknown distance function I. *Psychometrika*, 27:125–140, 1962.

[66] Roger N. Shepard. The analysis of proximities: multidimensional scalign with an unknown distance function II. *Psychometrika*, 27:219–246, 1962.

[67] T. Sikora. MPEG digital video–coding standards. *IEEE Signal Processing Magazine*, 14(5):82–100, September 1997.

[68] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38:587–607, 1992.

[69] Stephen R. Smoot and Lawrence A. Rowe. Laplacian Model for AC DCT Terms in Image and Video coding. In *Proceedings of the Ninth IEEE Image and Multidimensional Digital Signal Processing Workshop*, 1996.

[70] Stephen R. Smoot and Lawrence A. Rowe. Study of DCT Coefficient Distribution. In B. E. Rogowitz and T. N. Pappas, editors, *Human Vision and Electronic Imaging*, volume 2657. SPIE — The International Society for Optical Engineering, 1996.

[71] R.E. Steuer. *Multiple Criteria Optimization: Theory, computation, and Application.* John Wiley and Sons, New York, 1986.

[72] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, Wellesley, MA, 1995.

[73] P.C. Teo and D.J. Heeger. Perceptual image distortion. volume II of *Proc. SPIE*, pages 982–986, 1994.

[74] W.S. Torgerson. *Theory and Method of Scaling.* Wiley, New York, 1958.

[75] L.N. Trefethen and D. Bau III. *Numerical Linear Algebra.* SIAM: Philadelphia, 1997.

[76] Video Quality Experts Group. *Various documents.* `http://www.vqeg.org`, accessed December 6, 2002.

[77] Video Quality Experts Group. Current results and future directions. In King N. Ngan, Thomas Sikora, and Ming-Ting Sun, editors, *Visual Communications and Image Processing*, volume SPIE Proc. Vol. 4067, pages 742–753, 2000.

[78] James S. Walker. *A Primer on Wavelets and their Scientific Applications.* Chapman & Hall/CRC, Boca Raton, FL, 1999.

[79] G.K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[80] Waterloo BragZone. *Standard* `bird`, `boat`, `lena`, `mandrill` *and* `peppers` *images.* `http://links.waterloo.ca/bragzone.base.html`, accessed December 6, 2002.

[81] A.B. Watson. The cortex transform: Rapid computation of simulated neural images. *Computer Vision, Graphics and Image Processing*, 39:311–327, 1987.

[82] A.B. Watson. DCTune: A technique for visual optimization of DCT quantization matrices for individual images. *Society for Information Display Digest of Technical Papers*, XXIV:946–949, 1993.

[83] Andrew B. Watson. DCT quantization matrices visually optimized for individual images. In J. P. Allebach and B. E. Rogowitz, editors, *Proceedings of SPIE*, volume 1913 of *Human Vision, Visual Processing, and Digital Display*, pages 202–216, 1993.

[84] Sergei N. Yendrikhovskij. *Color reproduction and the naturalness constraint.* Ph.D. Thesis, IPO, Center for User-System Interaction, Eindhoven, 1998.

[85] Forrest W. Young. *ALSCAL Multidimensional Scaling software.* `http://forrest.psych.unc.edu/research/alscal.html`, accessed December 6, 2002.

[86] Forrest W. Young. Multidimensional scaling. In S. Kotz and N.L. Johnson, editors, *Encyclopedia of Statistical Sciences*. John Wiley & Sons, 1985.

[87] G. Young and A.S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, (3):19–22, 1938.

Index