An investigation of the use of gradients in imaging, including best approximation and the Structural Similarity image quality measure

by

Amelia Kunze

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The L^2-based mean squared error (MSE) and its variations continue to be the most widely employed metrics in image processing. This is most probably due to the fact that (1) the MSE is simple to compute and (2) it possesses a number of convenient mathematical properties, including differentiability and convexity. It is well known, however, that these L^2-based measures perform poorly in terms of measuring the visual quality of images. Their failure is partially due to the fact that the L^2 metric does not capture spatial relationships between pixels. This was a motivation for the introduction of the so-called Structural Similarity (SSIM) image quality measure [32] which, along with is variations, continues to be one of the most effective measures of visual quality. The SSIM index measures the similarity between two images by combining three components of the human visual system—luminance, contrast, and structure. It is our belief that the structure term, which measures the correlation between images, is the most important component of the SSIM.

A considerable portion of this thesis focusses on adapting the L^2 distance for image processing applications. Our first approach involves inserting an intensity-dependent weight function into the integral such that it conforms to generalized Weber's model of perception. We solve the associated best approximation problem and discuss examples in both one- and two-dimensions. Motivated by the success of the SSIM, we also solve the Weberized best approximation problem with an added regularization term involving the correlation.

Another approach we take towards adapting the MSE for image processing involves introducing gradient information into the metric. Specifically, we study the traditional L^2 best approximation problem with an added regularization term involving the L^2 distance between gradients. We use orthonormal functions to solve the best approximation problem in both the continuous and discrete setting. In both cases, we prove that the Fourier coefficients remain optimal provided certain assumptions on the orthonormal basis hold.

Our final best approximation problem to be considered involves maximizing the correlation between gradients. We obtain the relevant stationarity conditions and show that an infinity of solutions exists. A unique solution can be obtained using two assumptions adapted from [2]. We demonstrate that this problem is equivalent to maximizing the entire SSIM function between gradients. During this work, we prove that the discrete derivatives of the DCT and DFT basis functions form an orthogonal set, a result which has not appeared in the literature to the best of our knowledge.

Our study of gradients is not limited to best approximation problems. A second major focus of this thesis concerns the development of gradient-based image quality measures. This was based on the idea that the human visual system may also be sensitive to distortions

in the magnitudes and/or direction of variations in greyscale or colour intensities—in other words, their gradients. Indeed, as we show in a persuasive simple example, the use of the L^2 distances between image gradients already yields a significant improvement over the MSE. One naturally wonders whether a measure of the correlation between image gradients could yield even better results—in fact, possibly "better" than the SSIM itself! (We will define what we mean by "better" in this thesis.) For this reason, we pursue many possible forms of a "gradient-based SSIM".

First, however, we must address the question of how to define the correlation between the gradient vectors of two images. We formulate and compare many novel gradient similarity measures. Among those, we justify our selection of a preferred measure which, although simple-minded, we show to be highly correlated with the "rigorous" canonical correlation method. We then present many attempts at incorporating our gradient similarity measure into the SSIM. We finally arrive at a novel gradient-based SSIM, our so-called "gradSSIM1", which we argue does, in fact, improve the SSIM. The novelty of our approach lies in its use of SSIM-dependent exponents, which allow us to seamlessly blend our measure of gradient correlation and the traditional SSIM.

To compare two image quality measures, e.g., the SSIM and our "gradSSIM1", we require use of the LIVE image database [24]. This database contains numerous distorted images, each of which is associated with a single score indicating visual quality. We suggest that these scores be considered as the independent variable, an understanding that does not appear to be have been adopted elsewhere in the literature. This work also necessitated a detailed analysis of the SSIM, including the roles of its individual components and the effect of varying its stability constants. It appears that such analysis has not been performed elsewhere in the literature.

Acknowledgements

Firstly, I would like to thank my primary supervisor, Prof. Edward R. Vrscay, Department of Applied Mathematics, University of Waterloo, and my co-supervisor, Prof. Siv Sivaloganathan, Department of Applied Mathematics, University of Waterloo, for their support during both my undergraduate and graduate studies at UW. Although Prof. Vrscay and Prof. Sivaloganathan have always been friendly faces on campus (thanks in part to their friendship with my dad which predates me), I did not expect them to become meaningful figures in my own life. For this I am sincerely grateful. Indeed, I became interested in studying under Prof. Vrscay after he kindly taught me a reading course on mathematical imaging during my undergraduate degree. He continues to be a wealth of both kindness and knowledge now as my supervisor, and I am extremely grateful for our many exchanges, about mathematics and much else, which I thoroughly value.

I would also like to thank Prof. Giang Tran, Department of Applied Mathematics, University of Waterloo, and Prof. Zhou Wang, Department of Electrical and Computer Engineering, University of Waterloo, for devoting their time and expertise as readers of my thesis. In addition to their valuable participation related to my defense, I was fortunate enough to have taken a course from both Prof. Tran and Prof. Wang during my graduate studies. I am further grateful to them for fostering especially encouraging learning environments which enriched my studies considerably. My sincerest thanks and appreciation go to Prof. Wayne Oldford of the Department of Statistics and Actuarial Sciences, UW for introducing to us the method of "canonical correlation".

I would like to acknowledge partial financial support from the Faculty of Mathematics, UW. Finally, I gratefully acknowledge partial support in the form of Graduate Research Scholarships provided by the Natural Sciences and Engineering Research Council (ERV's Discovery Grant).

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

We entered into this work equipped with a few simple ideas. A major initial interest of ours concerned best approximation problems involving images. Measuring how well a given image approximates another image depends entirely on the choice of distance function. It is well known that using conventional, i.e., L^2-based, metrics on images produces distances that are not in agreement with their visual closeness according to a human observer. By developing metrics based on simple considerations of the human visual system—e.g., its nonlinear luminance perception described by Weber's law, its aptitude for extracting edges and structural information—one can produce mathematically optimal image approximations which are more likely to be visually "optimal" according to a human observer.

Our study of image-based best approximation also involved extending previous work which maximized the Structural Similarity (SSIM) Index and its individual components using orthonormal functions [3, 2]. The SSIM [32] is a well-known image quality measure which attempts to predict the visual quality of distorted images in a manner consistent with the human visual system. Some evaluations showed that the SSIM outperformed alternative prediction algorithms at the time of its debut [23], and it continues to be a popular choice in many image processing applications today. The SSIM consists of three component parts: a luminance component, a contrast component, and a correlation component. It was our guiding belief that the correlation is the most important component of the SSIM. Throughout our work, the correlation was a lodestone to which we returned time and time again.

At the same time, we were also highly motivated to pursue the use of gradients in image processing. For years, gradients have been used in a variety of image processing applications; A particularly canonical example is the use of gradient filters in edge detection. For

our introduction to using gradients, we naturally pursued the best approximation problem which maximizes the correlation between gradients. During this work, we were able to show that the discrete derivatives of the DCT and DFT basis functions form an orthogonal set, a result which has not appeared in the literature to the best of our knowledge.

Our study of gradients, however, was not limited to best approximation problems: We also undertook a computational exploration of the gradient, which culminated in many attempts to "improve" the SSIM by incorporating gradient information. (We will define what is meant by "improve" later in this thesis.) This was based on the idea that the human visual system may also be sensitive to distortions in the magnitudes and/or direction of variations in greyscale or colour intensities—in other words, their gradients. Indeed, some image quality measures centered on gradient information have recently been proposed in [17, 21]. The mathematical tractability which characterizes our approach, however, differentiates our gradient-based measures from those existing in the literature. Ultimately, we obtain a novel gradient-based measure which, based on our results using the LIVE image database [24], we argue "improves" the traditional SSIM. This pursuit also necessitated a structured and detailed examination of the SSIM which, to the best of our knowledge, has not appeared in the literature.

We could not have predicted how our initial interests would continually renew a rich fountain of ideas upon which our study evolved. Indeed, we were often developing multiple compelling problems at once, our attention shifting from one task to another and back again with great interest. Only after surveying this progression in retrospect could we grasp a complete and satisfying sense in its manner of unfolding. To honour this natural sequence of events, this thesis is presented as a narrative. Its chapters are organized chronologically and are outlined briefly below.

In Chapter 2, we present a discussion of some important foundational ideas and the mathematical preliminaries of our work. Our first best approximation problem is then presented in Chapter 3, where Weber's model motivates our initial attempt at adapting the L^2-based distance for image processing applications. Our computational investigation of the SSIM begins in Chapter 4, where we develop a simple example to investigate whether correlation is, in fact, the most important component of the SSIM. We return to our "Weberized" best approximation problem in Chapter 5, adding a regularization term involving the correlation between signals.

We provide a brief introduction to gradients and their applications in image processing in Chapter 6. We also consider a simple best approximation problem involving the L^2-distance between gradients. In Chapter 7, we explore the best approximation problem which maximizes the correlation between gradients. The contents of Chapters 8 and 9

pertain to our efforts towards engineering a gradient-based image quality measure. In Chapter 8, we develop and compare different measures of gradient similarity. We then incorporate those measures into the SSIM in Chapter 9. In order to assess the performance of our gradient-based measures, we analyze experiments using the LIVE image database. In this chapter, we also investigate the effect of varying the stability constants in the SSIM. In Chapter 10, we present some concluding remarks which include some natural avenues for future research.

Chapter 2

Mathematical Preliminaries

2.1 The Need for Image Quality Measures

In order to be able to characterize how well a given image is approximated by another image, or perhaps how much an image has been distorted after being subjected to some procedure, we need to be able to define some kind of "distance" between images. In many applications, one simply considers images to be elements of a metric space (more details are provided below) and then uses the metric space to define distances. However, as we discuss below, such metrics do not necessarily capture visual quality very well. As a result, an entire field of research is devoted towards developing perceptually-meaningful image quality measures. A perceptually-meaningful image quality measure computes distances between images in a manner consistent with their perceived visual closeness according to a human observer.

Such image quality measures may be based on known properties of images, of the human visual system, or both. A given image quality measure is typically classified as being either a full-reference, reduced-reference, or no reference method. Full-reference methods assume that a perfect reference image is available for making pixel-by-pixel comparisons against the distorted image, while no reference methods assume no such availability and use only the distorted image when inferring its quality. Existing somewhere in the middle are reduced-reference methods, which require some small set of attributes of the reference image. In each case, image quality measures can be further characterized by distinguishing between general-use and application-specific methods. One popular full-reference, general-use perceptual quality measure which will be discussed below is the Structural Similarity (SSIM) Index [32].

The application scope of image quality measures goes far beyond simply quantifying or predicting image quality. Most, if not all, image processing tasks rely somehow, either implicitly or explicitly, on an image quality measure. They provide a framework not only for assessing the performance of an image processing system, but also for optimizing its performance [30]. In short, developing perceptually-meaningful image quatlity measures not only motivates many mathematical problems explored in this thesis, but it is has a profound impact on the integrity of many real-world engineering processes.

2.2 Mathematical Representation of Images

A *greyscale image* can be defined as a real-valued function $f(x, y)$, where x and y are continuous real variables. Realistically, as is the case for a photograph, we expect the image to be defined over a bounded, and typically rectangular, domain $D \subset \mathbb{R} \times \mathbb{R}$. The value $f(x, y)$ indicates the intensity or grey level at a point (x, y) of the image. In practice, we also expect the range of f on D, called the greyscale range, to be bounded and non-negative. As such, the greyscale range is some interval $[A, B]$, where A will be black and B will be white. Of course, any intermediate value $A < f(x, y) < B$ will represent some shade of grey. In many applications, it is convenient to normalize the greyscale range $[A, B]$ to $[0, 1]$.

A *digital image* can be obtained by a discrete sampling of the continuous image $f(x, y)$. Such a greyscale digital image is represented by an $N \times M$ matrix u, where N and M are determined by the sampling frequency. We would like to choose N and M sufficiently large in order to preserve the visual information of the image. In this way, the sampling frequency contributes to the perceptual quality of a digital image. Visual distortions and artifacts can also be introduced in digital images by other processes; some distortion effects and their impact on perceptual quality are explored in our main work.

The entry u_{ij} of the matrix u, often written $u[i, j]$ in image processing applications, indicates the greyscale value of the digital image at the $(i, j)^{\text{th}}$ pixel, for $1 \leq i \leq N$ and $1 \leq j \leq M$. The intensity range of the digital image takes on discrete values to be stored in digital memory by an irreversible process known as quantization. An n-bit greyscale image usually has an intensity range with 2^n possible grey values. A particularly popular case is that of 8-bit images with possible greyscale values $\{0, 1, 2, \cdots, 255\}$. The largest possible pixel value, here $L = 255$, is also called the *dynamic range* of the image.

Beyond greyscale images, most images one encounters today are coloured. Colour images are represented mathematically by a vector-valued function. One possibility is to

employ red, green, and blue components, often referred to as an "RGB" image. At each point (x, y) in the domain, the entries in the vector $f(x, y) = (r(x, y), g(x, y), b(x, y))$ define the red, green, and blue intensities, respectively. For digital images, this translates to three matrices, one for each of the three colour channels. There are other ways to code colour images but a discussion of these methods is beyond the scope of this thesis.

Many popular quality measures have been developed on greyscale images. In some applications, these measures are first applied to each colour channel separately and subsequently aggregated. In other applications, colour images are somehow converted to greyscale images in order to asses their quality computationally. Neither of these approaches are completely satisfying. Indeed, there are many interesting open problems when one considers quality assessment for coloured media. In this work, our focus is restricted to perceptual quality assessment of greyscale images.

In practice, one is often concerned with the perceptual quality of digital images. Indeed, this is one of the main subjects of this thesis. Still, we will make use of the continuous image function $f(x, y)$ to establish some theoretical results. In the one-dimensional case, i.e, when $f(x) : \mathbb{R} \to \mathbb{R}$, $f(x)$ is usually called a *signal*. In the following sections, we will present some mathematical background for 1D and 2D domains D, and for both continuous and discrete dependent variables, to be well-equipped for any setting.

2.3 Generalized Weber's Model of Perception

Weber's Law is among the first recorded efforts to describe human perception in quantitative terms. In words, Weber's Law hypothesizes that the first just-noticeable increase in a stimulus is proportional to the pre-existing stimulus [20]. The statement applies to a stimulation of any of our five senses: hearing, taste, touch, smell, and, of particular relevance here, vision.

Indeed, Weber's Law can be stated mathematically for greyscale images. In this case, the pre-existing stimulus is a greyscale background intensity $I > 0$, while a similar grey level ΔI is the first noticeable deviation according to the human visual system. Weber's Law states that I and ΔI are related as follows,

$$\frac{\Delta I}{I} = C,$$

where C is constant over a significant range of intensities I.

There are also situations (see, e.g., [18]) in which the following generalization of Weber's Law is more applicable,

$$\frac{\Delta I}{I^a} = C, \tag{2.1}$$

where $a > 0$ and C is constant, or at least roughly constant, over a significant range of intensities I. In previous work, this relationship has been referred to as a "generalized" Weber model of perception in order to distinguish it from the standard Weber model, i.e., Weber's Law, where $a = 1$ [27]. The generalization can be adapted to conform to more complicated behaviours by adjusting a, with limiting value $a = 0$ corresponding to an absence of Weber's model. Indeed, Weber's law is also known to fail at low and high intensities [18]. As will be seen in our discussion of the Structural Similarity image quality index, one way to accommodate the failure at low intensities is to emply a "stability constant" A so that Eq. (2.1) becomes

$$\frac{\Delta I}{I^a + A} = C. \tag{2.2}$$

All of the above mentioned complications, however, are beyond the scope of this thesis. Here, we focus on the model in Eq. (2.1) with the understanding that our analysis and methods can be adapted to conform to more complicated behaviours.

In essence, Weber's model of perception implies that the human visual system will be less (more) sensitive to a given change in intensity ΔI in regions of an image at which the local image intensity I is high (low).

2.4 Noteworthy Image Quality Measures

2.4.1 Mean Squared Error (MSE)

In many discussions, we will work with signals $f(x) \in L^2[0,1]$. Recall that the space of real-valued, squared-integrable functions on [0,1] is defined by

$$L^2[0,1] = \left\{ f : [0,1] \to \mathbb{R} \,\middle|\, \int_0^1 |f(x)|^2 dx < \infty \right\}.$$

More precisely, the signals of interest to us will be further limited to functions in $L^2[0,1]$ with bounded range defined by the greyscale values. Regardless, the inner product in this space is given by

$$\langle f, g \rangle = \int_0^1 f(x)g(x)dx,$$

where, recognizing our restriction to real-valued functions, we have omitted the complex conjugate. The inner product induces a norm, denoted $||\cdot||_2$, and together $(L^2[0,1], ||\cdot||_2)$ form a Hilbert space. The 2-norm can be used to define a metric on $L^2[0,1]$. For two functions $f, g \in L^2[0,1]$, a distance between f and g can be computed by

$$d_2(f,g) = ||f-g||_2 = \left[\int_0^1 |f(x)-g(x)|^2 dx\right]^{1/2}.$$

For 2D images, the natural extension is to consider functions $f(x,y)$ in the space $L^2(D), D = [0,1]^2$. In this case, $d_2(f,g)$ involves a double integral over D.

For discrete signals, the relevant space is simply $\mathbb{R}^N$ equipped with the dot product, as follows, for $u, v \in \mathbb{R}^N$,

$$\langle u, v\rangle = \sum_{i=1}^{N} u_i v_i.$$

This inner product yields the l^2 norm (also called the Euclidean norm) for vectors on $\mathbb{R}^N$, stated below,

$$||u||_2 = \left(\sum_{i=1}^{N} |u_i|^2\right)^{1/2},$$

to which corresponds the following metric

$$d_2(u,v) = ||u-v||_2 = \left(\sum_{i=1}^{N} |u_i - v_i|^2\right)^{1/2}.$$

$(\mathbb{R}^N, ||\cdot||_2)$ is a Hilbert space.

The mean squared error (MSE), obviously related to $d_2(u,v)$, can now be defined below,

$$\text{MSE}(u,v) = \frac{1}{N}[d_2(u,v)]^2 = \frac{1}{N}\sum_{i=1}^{N}(u_i - v_i)^2.$$

The mean squared error is one of the most widely used signal fidelity measures. The MSE has many advantages, including: it is simple, it is easy to implement, and it satisfies the properties of convexity, symmetry, and differentiability [31].

For two digital images $u, v \in \mathbb{R}^{N\times M}$, the MSE is a double sum, as follows,

$$\text{MSE}(u,v) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(u_{ij} - v_{ij})^2.$$

Sometimes it is useful to compare the root mean squared error (RMSE) between images, defined by,

$$\mathrm{RMSE}(u,v) = \left[\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(u_{ij}-v_{ij})^2\right]^{1/2}.$$

A peak signal-to-noise ratio (PSNR) can also be computed from the MSE by

$$\mathrm{PSNR} = 10\log_{10}\frac{L^2}{\mathrm{MSE}},$$

where L is the dynamic range of the pixel intensities (recall from Section 2.2 that $L = 255$ for 8-bit images). The PSNR, although just a rescaling of the MSE, is useful for comparing error across pairs of images having different dynamic ranges.

Finally, having established that our images belong to a Hilbert space, we can state the following theorem relating to best approximation problems [7], [12].

Theorem 1. Let $\{\phi_1, \phi_2, \cdots, \phi_n\}$ be an orthonormal set in a Hilbert space H. Define $Y = \mathrm{span}\{\phi_i\}_{i=1}^n$. Y is a subspace of H. Then for any $x \in H$, the best approximation of x in Y is given by the unique element

$$y = P_Y(x) = \sum_{k=1}^{n} c_k\phi_k \qquad \text{(projection of } x \text{ onto } Y\text{)}$$

where

$$c_k = \langle x, \phi_k\rangle, \qquad k = 1, 2, \cdots, n.$$

In words, this well-known result is that the MSE is minimized by the Fourier coefficients of x with respect to the set $\{\phi_k\}$.

2.4.2 Structural Similarity (SSIM) Index

Although the MSE has many attractive qualities for both computation and analysis, it does not accurately predict image quality according to the human visual system. Many implicit assumptions made by the MSE and its resulting limitations have been discussed in [31]. A particularly compelling example in [31] shows a set of images obtained by applying different visual distortions to a single reference image, namely, the well-known "Einstein

image". Many of the resulting degraded images have nearly equal MSE, yet strikingly different degrees of visual quality. In Chapter 4 we present a set of these equal-MSE "Einstein images" [32] and use them in some experiments.

The Structural Similarity (SSIM) Index was proposed as an alternative to the MSE for image quality assessment. The underlying philosophy motivating the SSIM is that natural images are very structured, which suggests that the human visual system is highly adapted to extract those visual structures. These observations imply that the relationships between neighbouring pixels in an image are very important. Accordingly, the SSIM evaluates an image in chunks to preserve the relationships between neighbouring pixels. After extracting structural information between adjacent pixels, the SSIM approach seeks to penalize structural distortions more than distortions affecting other attributes of the image. By comparison, the MSE only measures the error at individual pixels in isolation; it does not consider neighbourhood dependencies or structures.

In order to emphasize the importance of structures, they must somehow be isolated from the other visual qualities in an image. The SSIM approach decomposes the similarity measurement into three distinct computations: a luminance comparison, a contrast comparison, and a structure comparison. Because the pixel intensities involved in the computation can exhibit large variations across an entire visual scene, these comparisons are performed in small, local image patches. (An "image patch" is an $n_1 \times n_2$ neighbourhood of greyscale values.)

Let $x, y \in \mathbb{R}^N$ denote local image patches taken from the same location in two images to be compared. The luminance in a scene is interpreted by the human visual system as the degree of perceived brightness, i.e., the predominance of light or dark greyscale intensities. The local luminance similarity between the patches x and y is denoted by $S_1(x, y)$. Image contrast is characterized by the range of brightness levels differentiating the objects in the image. A high-contrast image is one where there is a large degree of separation between the brightness levels of different objects. The local contrast similarity is performed by $S_2(x, y)$. Finally, structural information defines the form of the objects, independent of their perceived brightness. The structural comparison is denoted by $S_3(x, y)$. The local SSIM for the pair of patches x, y is usually defined as the product of these three terms, as follows,

$$\mathrm{SSIM}(x, y) = S_1(x, y) \cdot S_2(x, y) \cdot S_3(x, y).$$

The three local elements comprising the SSIM are defined by simple statistics. The

luminance of the signal is estimated using the mean intensity

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Then the luminance comparison is measured by

$$S_1(x, y) = \frac{2\bar{x}\bar{y} + C_1}{\bar{x}^2 + \bar{y}^2 + C_1}$$

where the presence of the stability constant C_1 prevents numerical instability when the denominator is close to 0.

The signal contrast is estimated by the standard deviation

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2\right)^{1/2},$$

so that the contrast comparison can be defined by

$$S_2(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}.$$

The correlation coefficient between the patches x and y gives a rating of the similarity of local structures, written below,

$$S_3(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3},$$

where σ_{xy} is the covariance of x and y,

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}).$$

In detail, the local SSIM is

$$\begin{aligned}\mathrm{SSIM}(x, y) &= S_1(x, y) \cdot S_2(x, y) \cdot S_3(x, y) \\ &= \frac{2\bar{x}\bar{y} + C_1}{\bar{x}^2 + \bar{y}^2 + C_1}\ \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\ \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}.\end{aligned}$$

The SSIM is bounded, $-1 \leq \text{SSIM}(x, y) \leq 1$, and symmetric in its input arguments, $\text{SSIM}(x, y) = \text{SSIM}(y, x)$. An $\text{SSIM}(x, y)$ closer to 1 indicates better perceptual similarity between the image patches x and y, and $\text{SSIM}(x, y) = 1$ if and only if $x = y$. If the stability constants are chosen so that $C_3 = C_2/2$, then the $\text{S}_2(x, y)$ and $\text{S}_3(x, y)$ terms collapse into a single term, as shown below,

$$\text{SSIM}(x, y) = \frac{2\bar{x}\bar{y} + C_1}{\bar{x}^2 + \bar{y}^2 + C_1} \; \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}.$$

This form of the SSIM has been widely adopted in the literature.

A global SSIM value between two images is often obtained by computing local SSIM values between sets of corresponding image patches which cover the image and then taking the mean. Such image patches are often obtained by "sliding" an $n_1 \times n_2$ pixel "window" in an overlapping manner. If M such sliding windows are employed, then the global, mean SSIM, denoted as MSSIM, is given by

$$\text{MSSIM} = \frac{1}{M} \sum_{i=1}^{M} \text{SSIM}(x, y).$$

For improved performance, it is suggested to use a Gaussian-weighted vector to compute the local statistics [32]. In this formulation, the central pixels in each window contribute more than those located around the edge. The relative weights of the local SSIM values as they contribute to the global MSSIM can also be generalized similarly.

The luminance component $S_1(x, y)$ is connected to Weber's Law. $S_1(x, y)$ can easily be rewritten to obtain

$$S_1(\bar{x}, \bar{y}) = \frac{2(\bar{y}/\bar{x}) + C_1'}{1 + (\bar{y}/\bar{x})^2 + C_1'}, \tag{2.3}$$

where $C_1' = C_1/\bar{x}^2$. The dependence on the ratio $\bar{y}/\bar{x}$ already suggests a connection to Weber's Law. To make it clear, let x be taken from a perfect reference image, while the patch y is obtained from applying a small distortion to x. We expect y to approximate x, i.e., $y = x + \Delta x$ for some residual term Δx. By linearity, we have $\bar{y} = \bar{x} + \overline{\Delta x}$. Substitution into Equation 2.3 yields

$$S_1(\bar{x}, \bar{y}) = \frac{2(1 + \overline{\Delta x}/\bar{x}) + C_1'}{1 + (1 + \overline{\Delta x}/\bar{x})^2 + C_1'}$$

If the ratio $\overline{\Delta x}/\bar{x}$ is a constant from Weber's Law, then as the mean intensity $\bar{x}$ increases, a greater deviation $\overline{\Delta x}$ would be required to keep the perceptual luminance similarity index

$S_1(\bar{x}, \bar{y})$ constant. This is consistent with Weber's Law, which states that the human visual system can tolerate greater discrepancies at high intensity regions of an image.

We conclude this section with one final observation: The correlation between two vectors $x, y \in \mathbb{R}^N$ is related to the angle between them. Indeed, the correlation

$$S_3(x,y) = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{[\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2]^{1/2}[\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2]^{1/2}}$$

can be rewritten using

$$x_0 = x - \bar{x} \quad \text{and} \quad y_0 = y - \bar{y},$$

to obtain

$$S_3(x,y) = \frac{\sum_{i=1}^{N}(x_0)_i(y_0)_i}{[\sum_{i=1}^{N}((x_0)_i)^2]^{1/2}[\sum_{i=1}^{N}((y_0)_i)^2]^{1/2}} = \frac{x_0 \cdot y_0}{\|x_0\|\|y_0\|}. \tag{2.4}$$

Eq. (2.4) shows that the correlation $S_3(x, y)$ may be interpreted as the cosine of the angle θ between the two zero-mean vectors x_0 and y_0.

2.5 Discrete Fourier Transform (DFT)

The discrete Fourier transform (DFT) provides a frequency-domain representation of a discrete spatial signal $u \in \mathbb{R}^N$ [4]. The DFT treats the signal u as an entire period obtained by evenly sampling a periodic sequence. Mathematically, it assumes $u[j + N] = u[j]$ for $j \in \mathbb{Z}$, which enforces a periodic extension of the data. As such, the basis functions used in the DFT representation are N-periodic vectors. In particular, the following set of complex N-periodic vectors, ϕ_k, $0 \le k \le N-1$, forms an orthonormal basis in $\mathbb{R}^N$,

$$\phi_k[n] = \frac{1}{\sqrt{N}} \exp\left(\frac{i2\pi kn}{N}\right), \quad 0 \le n \le N-1. \tag{2.5}$$

Any signal $u \in \mathbb{R}^N$ will admit an expansion of the form

$$u = \sum_{k=0}^{N-1} c_k \phi_k,$$

where $c_k = \langle u, \phi_k \rangle$ denotes the Fourier coefficients of f in the $\{\phi_k\}$ basis. Substitution gives

$$\begin{aligned} c_k = \langle u, \phi_k \rangle &= \sum_{n=0}^{N-1} u[n]\overline{\phi_k[n]} \\ &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} u[n] \exp\left(-\frac{i2\pi kn}{N}\right), \quad 0 \le k \le N-1. \end{aligned}$$

The vector of coefficients $c \in \mathbb{R}^N$ defines the DFT of the signal $u \in \mathbb{R}^N$. The original signal can be retrieved by applying the inverse discrete Fourier Transform (IDFT), defined by,

$$u[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} c[k] \exp\left(\frac{i2\pi kn}{N}\right), \quad 0 \le n \le N-1.$$

Other versions of the DFT and IDFT exist. Some common variations arise when the basis functions ϕ_k are not normalized, i.e., ϕ_k form an orthogonal set, but not an orthonormal one.

2.6 Discrete Cosine Transform (DCT)

For a signal $u \in \mathbb{R}^N$, the DFT can exhibit poor convergence near the endpoints $u[jN]$ for $j \in \mathbb{Z}$. Because the DFT assumes an N-point periodic extension of the data u, convergence problems occur when $u[N-1]$ is not close to $u[0] = u[N]$. To address this issue, one can think of reflecting and repeating the data in order to produce an even $2N$-point periodic extension of the signal $u \in \mathbb{R}^N$. This idea is the key assumption made by the discrete cosine transform (DCT) [5].

The following set of real N-vectors ϕ_k, $0 \le k \le N-1$, forms an orthonormal basis in $\mathbb{R}^N$,

$$\begin{aligned} \phi_0[n] &= \frac{1}{\sqrt{N}}, \quad 0 \le n \le N-1 \\ \phi_k[n] &= \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right), \quad 1 \le k \le N-1, \quad 0 \le n \le N-1. \end{aligned}$$

For convenience, let

$$\lambda_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & k \neq 0 \end{cases}$$

so that the basis functions ϕ_k, $0 \leq k \leq N-1$, can be compactly written

$$\phi_k[n] = \lambda_k \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right), \quad 0 \leq n \leq N-1.$$

As before, any signal $u \in \mathbb{R}^N$ can be represented by a linear combination of these basis functions. Then the DCT is defined by the coefficients

$$c_k = \langle u, \phi_k \rangle = \sum_{n=0}^{N-1} u[n] \lambda_k \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right), \quad 0 \leq k \leq N-1.$$

The original signal can be retrieved using the inverse discrete cosine transform (IDCT), defined by,

$$u[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c_k \lambda_k \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right), \quad 0 \leq n \leq N-1.$$

Like the DFT, there are many forms of the DCT and IDCT. The version provided here is called the "DCT-II" and is widely used—in fact, it is the one employed in the JPEG compression standard.

Chapter 3

Intensity-based Weight Functions in Generalized Weber's Model of Perception

3.1 Intensity-dependent Weight Functions Which Produce "Weberized" Distance Functions

In this section, we present our first approach towards adapting the MSE for image processing applications. We first observe that L^2-based distances, including the MSE, do not accommodate Weber's model of perception because they integrate point-wise intensity differences, $|u(x) - v(x)|^2$, with no consideration of the magnitudes of $u(x)$ or $v(x)$. By contrast, as discussed in Section 2.4.2, the SSIM index is connected to Weber's Law. In addition, other classical image processing methods have been adapted to conform to Weber's model, including total variation (TV) restoration [25] and Mumford-Shah segmentation [26]. In this work, the idea is to "Weberize" the L^2 distance by introducing a simple intensity-dependent weighting function into the integral.

In the following discussion, we consider signals supported on $D = [0, 1]$ with bounded greyscale range $\mathbb{R}_g = [A, B] \subset (0, \infty)$. The restriction $A > 0$ is necessary due to the form of the weighting function, as will become clear below. We denote this relevant set of signals by $\mathcal{F} = \{u \in L^2[0, 1] \,|\, u : [0, 1] \to \mathbb{R}_g\}$. Before continuing, we mention that our discussion can be extended to higher dimensional cases, including image functions such that $D \subset \mathbb{R} \times \mathbb{R}$. It also easily extends to the discrete case encountered in practice.

Recall that Weber's model of perception states that the human visual system will be less (more) sensitive to a given change in intensity in regions of an image at which the local image intensity is high (low). As such, a Weberized distance between two functions $u(x)$ and $v(x)$ should tolerate greater (lesser) differences over regions in which they assume higher (lower) intensity values. The degree of toleration as the intensity varies will be determined by the Weber parameter a.

A Weberized weighting function could take on many interesting forms. One possibility, which will be the focus of this work, is to consider weight functions which are dependent upon one or both of the intensities of the image functions $u(x), v(x) \in \mathcal{F}$. The general form of such an intensity-based weighted L^2 distance is

$$d_{2W}(u,v) = \left[\int_D g(u(x),v(x))[u(x)-v(x)]^2 dx\right]^{1/2}, \tag{3.1}$$

where $g : \mathbb{R}_g \times \mathbb{R}_g \to \mathbb{R}_+$ denotes the intensity-dependent weight function.

Eq. (3.1) was first presented and thoroughly investigated in [11] and [10] for Weber's standard model with $a = 1$. The ideas explored in [11] and [10] provide a significant foundation upon which the main contribution of this section is based, and further inform other work presented later in this thesis.

A natural first question is that of properties that should be satisfied by the weight function g as well as possible functional forms that it could assume. As discussed in [11], for d_{2W} to satisfy the properties of a metric, $g(u,v)$ should be symmetric in its arguments, i.e., $g(u,v) = g(v,u)$. Furthermore, for d_{2W} to be Weberized, it is desirable that $g(u,v)$ be decreasing in each of its arguments. These requirements are satisfied by the family of weight functions, $g(u,v) = |uv|^{-q}$, where $q > 0$, resulting in weighted L^2 metrics of the form,

$$d_{2W,q}(u,v) = \left[\int_D \frac{1}{u(x)^q v(x)^q}[u(x)-v(x)]^2\right]^{1/2}$$

The appearance of both functions in the denominator, however, complicates matters when we consider the approximation problem $u \approx v$ where v is a linear combinator of basis functions—see Sect. (3.2). In [11], two unsymmetric weight functions, $g_1(u(x),v(x)) = u(x)^{-2}$ and $g_2(u(x),v(x)) = v(x)^{-2}$, were employed to produce two integral distance functions. The resulting distance functions were shown to conform to Weber's standard model for $a = 1$. We now present an extension of that result to produce distance functions which conform to Weber's model for any $a > 0$.

For any $a > 0$, consider the nonsymmetric weight function $g_1(u(x), v(x)) = u(x)^{-2a}$ so that the weighted L^2 distance in Eq. (3.1) becomes

$$\Delta_a(u, v) = \left[\int_D \frac{1}{u(x)^{2a}}[u(x) - v(x)]^2 dx\right]^{1/2}. \tag{3.2}$$

Now consider the nonsymmetric weight function $g_2(u(x), v(x)) = v(x)^{-2a}$ so that the weighted L^2 distance in Eq. (3.1) becomes

$$\Delta_a(v, u) = \left[\int_D \frac{1}{v(x)^{2a}}[u(x) - v(x)]^2 dx\right]^{1/2}. \tag{3.3}$$

Note that in general, $\Delta_a(u, v) \neq \Delta_a(v, u)$, which implies that Δ_a is not a metric in the strict mathematical sense of the term. This is once again the price paid for employing weight functions $g(u, v)$ which are not symmetric in the functions u and v. This complication is not a serious limitation because of the following results that apply to our space $\mathcal{F}$ of image functions.

Theorem 2. Let u,$v \in \mathcal{F}$, once again recalling the assumption that the greyscale range $\mathbb{R}_g = [A, B]$ is bounded away from zero, i.e., $A > 0$. Then for $\Delta_a(u, v)$ and $\Delta_a(v, u)$ defined in Equations (3.2) and (3.3) respectively,

$$\frac{1}{B^a} d_2(u, v) \leq \left\{\begin{matrix}\Delta_a(u, v) \\ \Delta_a(v, u)\end{matrix}\right\} \leq \frac{1}{A^a} d_2(u, v), \tag{3.4}$$

where d_2 denotes the L^2 metric, from which it follows that

$$\left(\frac{A}{B}\right)^a \Delta_a(u, v) \leq \Delta_a(v, u) \leq \left(\frac{B}{A}\right)^a \Delta_a(v, u). \tag{3.5}$$

Proof. We first proceed to obtain Eq. (3.4). Our restriction of the range $\mathbb{R}_g = [A, B]$ immediately gives

$$0 < A \leq u(x) \leq B.$$

Because y^{2a} is increasing for $y > 0$ and for any $a > 0$, we obtain

$$\frac{1}{B^{2a}} \leq \frac{1}{u(x)^{2a}} \leq \frac{1}{A^{2a}}. \tag{3.6}$$

Multiply the inequality (3.6) by $[u(x) - v(x)]^2 \geq 0$ and integrate to get

$$\int_D \frac{1}{B^{2a}}[u(x) - v(x)]^2 dx \leq \int_D \frac{1}{u(x)^{2a}}[u(x) - v(x)]^2 dx \leq \int_D \frac{1}{A^{2a}}[u(x) - v(x)]^2 dx.$$

Finally, taking the square root,

$$\frac{1}{B^a}\left[\int_D [u(x)-v(x)]^2 dx\right]^{1/2} \leq \left[\int_D \frac{1}{u(x)^{2a}}[u(x)-v(x)]^2 dx\right]^{1/2} \leq \frac{1}{A^a}\left[\int_D [u(x)-v(x)]^2 dx\right]^{1/2},$$

which—with reference to the definition in Eq. (3.2)—we recognize as

$$\frac{1}{B^a} d_2(u,v) \leq \Delta_a(u,v) \leq \frac{1}{A^a} d_2(u,v). \tag{3.7}$$

Noting that we can equally write Eq. (3.6) for $v(x)$ in place of $u(x)$, the argument proceeds as before to obtain

$$\frac{1}{B^a} d_2(u,v) \leq \Delta_a(v,u) \leq \frac{1}{A^a} d_2(u,v), \tag{3.8}$$

which, together with the previous inequality, establish Eq. (3.4).

Using Eq. (3.7) from the preceding result, we have

$$d_2(u,v) \leq B^a \Delta_a(u,v),$$

which can be combined with the upper bound in Eq. (3.7) to get the first half of the desired inequality,

$$\Delta_a(v,u) \leq \left(\frac{B}{A}\right)^a \Delta_a(u,v). \tag{3.9}$$

Similarly, Eq. (3.8) gives

$$d_2(u,v) \leq B^a \Delta_a(v,u),$$

which can be used with

$$A^a \Delta_a(u,v) \leq d_2(u,v)$$

to obtain

$$A^a \Delta_a(u,v) \leq d_2(u,v) \leq B^a \Delta_a(v,u),$$

or, after rearranging,

$$\left(\frac{A}{B}\right)^a \Delta_a(u,v) \leq \Delta_a(v,u).$$

Putting Eq. (3.9) and Eq. (3.1) together, we obtain Eq. (3.5),

$$\left(\frac{A}{B}\right)^a \Delta_a(u,v) \leq \Delta_a(v,u) \leq \left(\frac{B}{A}\right)^a \Delta_a(v,u),$$

which completes the proof. ■

The following example illustrates how the weighting function accommodates generalized Weber's model of perception. Consider the "flat" reference image $u(x) = I$, where $I \in \mathbb{R}_g$. For an $a > 0$, let $v(x) = I + \Delta I$ be the constant approximation to $u(x)$, where $\Delta I = CI^a > 0$ is the minimum perceived change in intensity corresponding to I, according to Weber's model. The L^2 distance between u and v is

$$\begin{aligned} d_2(u,v) &= \left[\int_D [u(x) - v(x)]^2 dx\right]^{1/2} \\ &= \left[\int_D [I - (I + \Delta I)]^2 dx\right]^{1/2} \\ &= \left[\int_D (\Delta I)^2 dx\right]^{1/2} \\ &= \left[\int_D dx\right]^{1/2} \cdot \Delta I \\ &= KCI^a, \qquad \text{where } K = \left[\int_D dx\right]^{1/2}. \end{aligned} \tag{3.10}$$

If we impose $D = [0, 1]$, then $K = 1$. In general, K is a constant independent of $u(x) = I$.

The weighted L^2 distance in Eq. (3.2) is

$$\begin{aligned} \Delta_a(u,v) &= \left[\int_D \frac{1}{u(x)^{2a}} [u(x) - v(x)]^2 dx\right]^{1/2} \\ &= \left[\int_D \frac{1}{I^{2a}} [I - (I + \Delta I)]^2 dx\right]^{1/2} \\ &= \left[\int_D \frac{1}{I^{2a}} (\Delta I)^2 dx\right]^{1/2} \\ &= \left[\int_D \frac{1}{I^{2a}} C^2 I^{2a} dx\right]^{1/2} \\ &= \left[\int_D dx\right]^{1/2} \cdot C \\ &= KC. \end{aligned} \tag{3.11}$$

Note that the L^2 distance in Eq. (3.10) increases with the intensity level I, which is expected since ΔI increases with I. However, the weighted L^2 distance in Eq. (3.11)

remains constant. As such, we claim that $\Delta_a(u, v)$ accommodates, or "conforms to", Weber's model of perception for $a > 0$: Perturbations ΔI of image intensities I according to Weber's model, $\Delta I = CI^a$, yield the same distance measure, independent of I.

Before concluding this section, it is necessary to mention another related method that has been devised to Weberize L^p-based metrics, namely, the use of appropriate measures that are supported on the (positive) range space $\mathbb{R}_g = [A, B]$ of functions to reformulate the integrals which normally define the L^p distance between two functions. Some foundational ideas in this section also appeared in [15] and [16]; This related method is discussed in greater detail in [14]. That being said, this method will not be discussed any further in this thesis.

3.2 Best Approximation in Terms of Weberized Distance Functions

In what follows, let $\{\phi_k\}_{k=1}^{\infty}$ denote a set of real-valued functions that form a complete basis in $L^2([0, 1])$. Now let $u \in \mathcal{F} \subset L^2[0, 1]$ denote the reference function to be approximated. We are interested in best approximations to u having the form,

$$u \approx v_n = \sum_{k=1}^{n} c_k \phi_k,$$

for some $n \geq 1$.

We wish to find the "best Weberized" approximation to u which corresponds to minimizing the following weighted L^2 distance,

$$\Delta_a(u, v_n) = \left[\int_D \frac{1}{u(x)^{2a}} \left[u(x) - \sum_{k=1}^{n} c_k \phi_k \right]^2 dx \right]^{1/2}$$

for a given $n \geq 1$ and $a > 0$. In practice, it is more convenient to work with the squared distance function,

$$[\Delta_a(u, v_n)]^2 = \int_D \frac{1}{u(x)^{2a}} \left[u(x) - \sum_{k=1}^{n} c_k \phi_k \right]^2 dx := f(c). \tag{3.12}$$

Letting $g(x)$ denote the intensity-dependent weighting function, we can expand Eq. (3.12) to get,

$$\begin{aligned} f(c) &= \int_D g(x) \left[u(x) - \sum_{k=1}^{n} c_k \phi_k \right]^2 dx \\ &= \int_D g(x) u(x)^2 dx - 2 \sum_{k=1}^{n} c_k \int_D g(x) u(x) \phi_k dx + \sum_{k=1}^{n} \sum_{j=1}^{n} \int_D g(x) \phi_k \phi_j dx. \end{aligned} \tag{3.13}$$

While I was not involved in establishing the relevant theoretical results, a proof of the existence and uniqueness of the minimizer $v_n \in \mathcal{F}$ can be found in [27]. Moreover, in [27], the space of functions $\mathcal{F}$ allows for more general behaviour than what is considered in this thesis.

We may now pursue the unique optimizer v_n by way of the stationarity conditions,

$$\frac{\partial f}{\partial c_p} = 0, \quad 1 \leq p \leq n.$$

Differentiating Eq. (3.13), we obtain the following linear system of equations in terms of the unknown coefficients c_p,

$$\int_D g(x) u(x) \phi_p dx = \sum_{k=1}^{n} c_k \int_D g(x) \phi_k \phi_p dx \quad , \text{ for } 1 \leq p \leq n. \tag{3.14}$$

Note that in the special case $g(x) = 1$, we have the usual L^2-based distance and Eq. (7.14) simplifies to

$$\langle u, \phi_p \rangle = \sum_{k=1}^{n} c_k \langle \phi_k, \phi_p \rangle \qquad \text{for } 1 \leq p \leq n,$$

which, using the orthonormality of the basis functions, reduces to the Fourier coefficients as expected,

$$c_p = \langle u, \phi_p \rangle \qquad \text{for } 1 \leq p \leq n.$$

3.2.1 Selected Examples in Best Weberized Approximations

Example 1: Consider the following step function on $D = [0, 1]$,

$$u(x) = \begin{cases} 2, & 0 \leq x \leq 1/2 \\ 4, & 1/2 < x \leq 1. \end{cases} \tag{3.15}$$

We use the following set of functions

$$\begin{aligned}\phi_1(x) &= 1 \\ \phi_k(x) &= \sqrt{2}\cos((k-1)\pi x), \quad k \geq 2,\end{aligned} \tag{3.16}$$

which form an orthonormal basis in the space of functions $L^2[0,1]$.

In Fig. 3.1 are presented the plots of the best Weberized approximations v_n to u using $N = 5$, $N = 10$, and $N = 20$ basis functions for the cases $a = 0.25, 0.5, \cdots, 2.0$. The best L^2 approximations, u_n, corresponding to the case $a = 0$, are also shown for comparison. To obtain the unknown coefficients in each case, we solve the system of equations described by Eq. (7.14) in Maple using the '`int`' command. As expected, the best Weberized approximations v_n yield better approximations of $u(x)$ than u_n over $[0, 0.5]$ and poorer approximations over $[0.5, 1]$. Also as expected, the degree of "betterness" over $[0, 0.5]$ and "worseness" over $[0.5, 1]$ of the Weberized approximations increases with the Weber exponent a since the weight function $g(u) = u^{-2a}$ decreases more rapidly with increasing a.

Example 2: Consider the 512×512-pixel, 8 bit-per-pixel image *Boat*. Recall the DCT basis functions defined previously for vectors in $\mathbb{R}^N$,

$$\phi_k[n] = \lambda_k \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right), \quad 0 \leq n \leq N-1, \tag{3.17}$$

where

$$\lambda_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & k \neq 0. \end{cases}$$

Eq. (3.17) can be used to define the 2-dimensional DCT basis,

$$\phi_{kl}[n, m] = \phi_k[n]\phi_l[m], \quad 0 \leq n, m \leq N-1, \tag{3.18}$$

for $0 \leq k \leq K-1$ and $0 \leq l \leq L-1$, where K and L are the numbers of basis functions used in the best approximation problem. In the following experiment, we compute the best approximation in 8×8 image blocks, using $2 \times 2 = 4$ 2D-DCT basis functions for each block. This choice corresponds to $N = 8$ and $K = L = 2$. To obtain the unknown coefficients for each block, we minimize the distance function in Eq. (3.12) in Maple using gradient descent. After processing each block, we combine the block-wise best approximations to construct the image. Theoretically, we can perfectly reconstruct the *Boat* image by using 8×8 2D-DCT basis functions for each block.

Figure 3.2 depicts (a) the original *Boat* image, (b) the best L^2 approximation, and the best Weberized approximations for (c) $a = 0.5$ and (d) $a = 1$. Overall, there is not much perceptual difference between the approximations in (b), (c), and (d); Although the sky is reconstructed fairly well, many of the visual details of the reference image are obscured by the blockiness of the approximations. Moreover, there are many annoying ringing artifacts caused by the distortion over regions with edges separating high and low greyscale intensities. For example, the diagonal posts coming off the central mast exhibit an obvious ringing effect.

Despite the general similarity of the approximations, upon closer inspection, the dark shadows throughout the images appear more "shaded in" in (c) and (d) when compared to (b). For example, the central mast appears more uniformly black in (c) and (d); As a result, the ringing artifacts are slightly less pronounced here compared to the mast in the best L^2 approximation. Similarly, the name of the boat also appears thicker and darker in (c) and (d). However subtle, these observations correspond with the expected effect of the intensity-based weighting function: They suggest that the Weber-based approximations tolerate lesser differences over lower intensity regions of the image.

The opposite observation can be made in the high intensity regions of the image. In particular, the original image features a bright rope hanging off the front of the central boat. The bright end of rope strongly stands out against the dark underside of the boat. As expected, this line is slightly brighter and more pronounced in (b) than it is in (c) and (d). This observation complements those made previously, suggesting that the Weber-based approximations tolerate greater differences over high intensity regions of the image.

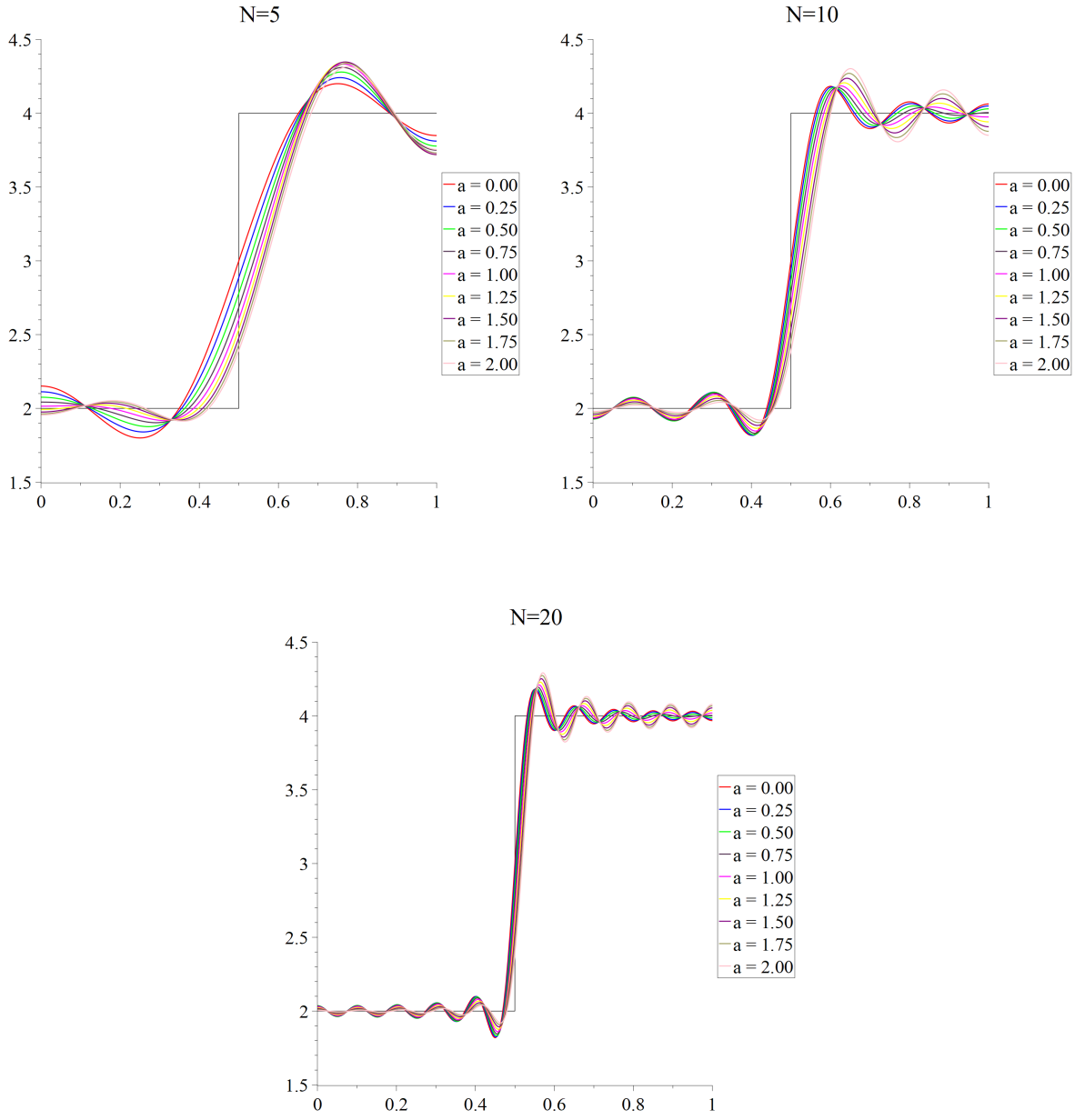


Figure 3.1: Best L^2 ($a = 0$) approximation and best Weberized approximations for $a = 0.25, \cdots, 2.00$ to the step function $u(x)$ using $N = 5$, $N = 10$, and $N = 20$ basis functions.

(a) Original Image

(b) $a = 0$ (best L^2)

(c) $a = 0.5$

(d) $a = 1$

Figure 3.2: (a) Original *Boat* image and best approximations for (b) $a = 0$ (best L^2), (c) $a = 0.5$, and (d) $a = 1$ using $2 \times 2 = 4$ 2D-DCT basis functions for each 8×8 image block.

Chapter 4

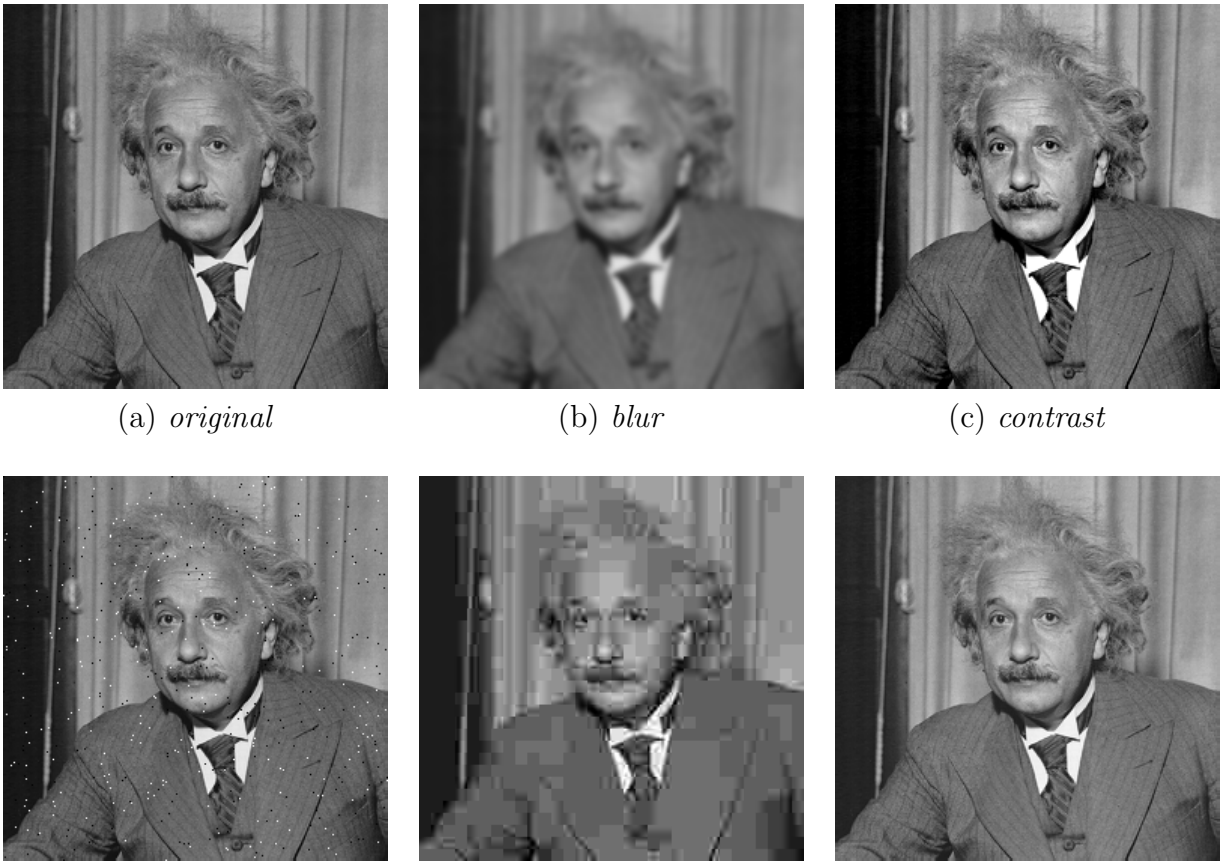
The Einstein Images

4.1 The Einstein Images

We now present our first set of experiments on the well-known Einstein images. These experiments, the first of many to be presented in this thesis, begin our detailed investigation of the SSIM. A main goal of the experiments presented in this section is to explore whether the correlation alone is sufficient to characterize the visual closeness between images.

Figure 4.1 shows six Einstein portraits, each of which is a 256×256 pixel, 8 bits-per-pixel greyscale image. We are grateful to have obtained these Einstein images from Prof. Z. Wang, Department of Electrical and Computer Engineering, University of Waterloo. They have been used in several of his papers to illustrate the inadequacy of the traditional L^2 metric for applications in signal and image processing; To the best of our knowledge, these images were first presented by Wang, Bovik, and Sheikh in [32].

The images *blur*, *contrast*, *impulse*, *jpg*, and *meanshift* (Figure 4.1 (b)-(f)) are perturbations of the Einstein image *original* (Figure 4.1 (a)). Each image (b)-(f) has been degraded by a particular distortion as indicated by its title. The *blur* image results from applying a blurring filter to *original*. *contrast* is obtained by performing a "contrast stretch" on *original*. ("Constrast stretching" describes an intensity transformation which "stretches" a dominant interval of greyscale values to span a larger range of values.) *impulse* is obtained by contaminating *original* with randomly occurring white and black pixels, usually called "impulsive salt-and-pepper noise". *jpg* is produced by a JPEG compression of *original* using a rather low quality factor. *meanshift* is obtained by increasing all of the greyscale values in *original* by a small constant value, resulting in an overall lighter image.



(a) *original* (b) *blur* (c) *contrast*

(d) *impulse* (e) *jpg* (f) *meanshift*

Figure 4.1: The reference Einstein image *original* and its perturbations.

(Greyscale values which, once shifted, would exceed the dynamic range of the image are simply assigned the highest possible value.)

All of the distortions were adjusted to yield nearly equal MSE relative to *original*. Because the degraded images differ significantly and obviously in perceptual quality, they are a striking example of the failure of the MSE to measure perceptual quality. Table 4.1 reports the RMSE values of the Einstein images in Figure 4.1. It can be seen from the table that the *jpg* image is somewhat of an exception, with a slightly lower RMSE than the others.

The MSSIM scores, reported in [32] and presented in Table 4.3, are more indicative of the apparent perceptual quality of the images. Unlike the MSE, the MSSIM scores are well

blur	*contrast*	*impulse*	*jpg*	*meanshift*
11.9962	12.0091	11.9975	11.9144	11.9998

Table 4.1: RMSE between the original *Einstein* images and its perturbations.

separated, which reflects the significant variations in quality among the degraded images relative to *original*.

blur	*contrast*	*impulse*	*jpg*	*meanshift*
0.6940	0.9133	0.8317	0.6624	0.9884

Table 4.2: MSSIM between the original *Einstein* images and its perturbations.

Perhaps more pertinent than the actual MSSIM values are their relative values. The values in Table 4.2 imply the following ordering of the distorted images in decreasing order of quality:

$$\textit{meanshift} > \textit{contrast} > \textit{impulse} > \textit{blur} > \textit{jpg}. \tag{4.1}$$

Ideally the ranking in Eq. (4.1) is consistent with the reader's own subjective preferences of the distorted images in Figure 4.1. While personal preferences among the poorer images may vary between individiuals, it should not be controversial to assert that the *meanshift* and *contrast* images clearly "look" the closest to *original*. It would only be reasonable to choose *meanshift* as the most similar to *original* visually, as does the MSSIM. It is interesting that while the "errors" in the *contrast* image are evident, it is perhaps the most visually appealing of all the images. Of course, the MSSIM is concerned with visual faithfulness to a reference image and not visual enhancement of images.

4.2 Correlation Between the Einstein Images

We now present our first set of so-called "Einstein experiments". Although simple-minded, their inclusion preserves the natural unfolding of our explorations, which—we have come to feel in retrospect—progressed in a satisfying and highly interconnected manner. In the same way they served us, we believe these experiments may provide an instructive first example for some readers. This section also prepares us to revisit the Einstein images in Chapter 8, where we compare different and novel measures of gradient similarity. Moreover, these two sets of "Einstein experiments" jointly introduce many ideas to be re-explored on a larger scale in Chapter 9, where we present our experiments on the LIVE database.

Recall that the SSIM is composed of three component terms,

$$\begin{aligned}\text{SSIM(x,y)} &= S_1(x,y)\cdot S_2(x,y)\cdot S_3(x,y)\\ &= \frac{2\bar{x}\bar{y}+C_1}{\bar{x}^2+\bar{y}^2+C_1}\ \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2}\ \frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3},\end{aligned} \tag{4.2}$$

one of which is the correlation (namely, the S_3 term). It is our belief that the correlation is the most important component of the SSIM. Because the S_3 term is thought to perform the structural comparison, this hypothesis certainly aligns with the emphasis on structural integrity which underlies the entire SSIM approach. Our primary interest in this section is to explore whether the correlation alone is a sufficient image quality measure. To that end, we perform a simple experiment which computes the SSIM and correlations between the *original* Einstein image and each of its five perturbations.

Our process can be summarized as follows:

1. For a given $n > 0$, partition the images into $n \times n$ nonoverlapping patches. In this experiment, we employed the following values for n: 8, 16, 32, and 64.

2. Compute the SSIM and correlation between all corresponding pairs of patches in *original* and each of its perturbations. Then combine the patchwise scores to obtain the average SSIM and average correlation for each perturbed image.

Relatively large values of the stability constants C_1, C_2, and C_3 push the three quotients in Eq. (4.2) towards a perfect similarity index of 1. We seek to limit the influence of the stability constants on the S_3 and SSIM values computed in our experiment. For this reason, we omit the stability constants in the numerators of the S_1, S_2, and S_3 terms. To protect against numerical instabilities, we are obliged to include small stability constants $C_1 = C_2 = C_3 = 10^{-7}$ in the denominator only. This discussion leads one to question the extent to which variations in the stability constants affect the SSIM. The sensitivity of the SSIM to changes in the stability constants will be a focus of Chapter 9.

Our SSIM scores, obtained using the simplistic method described above, are reported in Table 4.3. Table 4.3 also presents the MSSIM values for comparison, which are computed as discussed in Chapter 2, p. 12. Before performing a comparison with the correlation, a few observations can be made on these results alone. For each perturbed image, the SSIM values increase as the patch size n increases, with one exception: the *impulse* image from patches sized 8×8 to 16×16. Across all four values of n, a particularly significant increase is observed for the poorest images, i.e., *blur* and *jpg*. The increase is not as pronounced for

	8×8	16×16	32×32	64×64	MSSIM
blur	0.5247	0.6891	0.8215	0.9119	0.6940
contrast	0.8962	0.9124	0.9310	0.9640	0.9133
impulse	0.7951	0.7352	0.8210	0.9082	0.8317
jpg	0.3786	0.5966	0.8003	0.9056	0.6624
meanshift	0.9877	0.9887	0.9899	0.9927	0.9884

Table 4.3: Average SSIM values between the *original* Einstein image and its degradations for various patch sizes.

contrast and *meanshift*, both of which look very similar to *original*, because their minimum initial SSIM values are already relatively high.

The above discussion leads to the question, "What is the best value of n in this simplistic method?" If we consider the MSSIM values included in the final column of Table 4.3 as some kind of reference values, then a patch sized somewhere between 16×16 and 32×32 pixels should perform well. The "impulse" image is still the exception, once again due to its "dip" in SSIM values which occurs at 16×16 patches. Despite this issue, the values obtained using 16×16 patch size do preserve the ordering implied by the MSSIM values, as previously stated in Eq. (4.1),

$$\textit{meanshift} \;>\; \textit{contrast} \;>\; \textit{impulse} \;>\; \textit{blur} \;>\; \textit{jpg}.$$

The SSIM values obtained using 8×8 patches, although overall lower than their MSSIM counterparts, preserve this ordering as well and may also be a suitable choice.

The correlation scores are reported in Table 4.4. As observed for the SSIM scores, the correlation values increase with patch size n—with, once again, the exception of the *impulse* image from patches sized 8×8 to 16×16. The *contrast* and *meanshift* images, which are visually close to *original* according to the MSSIM, are essentially constant at the value 1.0 for all patch sizes. At this point, one might suspect that the correlation is not working as well as the SSIM in ascertaining visual differences between image sublocks.

That being said, the correlation *is* able to produce a ranking of the distorted images that is consistent with the ordering imposed by the MSSIM in Eq. (4.1). As in the case for the SSIM values, for all block sizes, the correlation-based ordering of the images places *meanshift* and *contrast* as first and second, respectively, and *jpg* in last place. The *impulse* and *blur* images change positions with patch size as before. For 8×8 patches, *impulse* is greater than *blur*; But for all higher patch sizes, *impulse* is less than *blur*. At a minimum,

	8×8	16×16	32×32	64×64
blur	0.6698	0.7815	0.8648	0.9260
contrast	0.9936	0.9957	0.9996	0.9997
impulse	0.8218	0.7673	0.8394	0.9126
jpg	0.3881	0.6078	0.8081	0.9074
meanshift	1.0000	1.0000	1.0000	1.0000

Table 4.4: Average S_3 values between the *original* Einstein image and its degradations.

using 8×8 patches, the correlation discerns sufficient differences in visual quality among the degraded images to assign the same relative ordering as the MSSIM.

To see if there is any correlation between the SSIM values in Table 4.3 and the correlation values in Table 4.4, we plot the ordered pairs of values (SSIM, correlation) for all images and patch sizes. The resulting plot is shown in Figure 4.2. The pairs from *contrast*

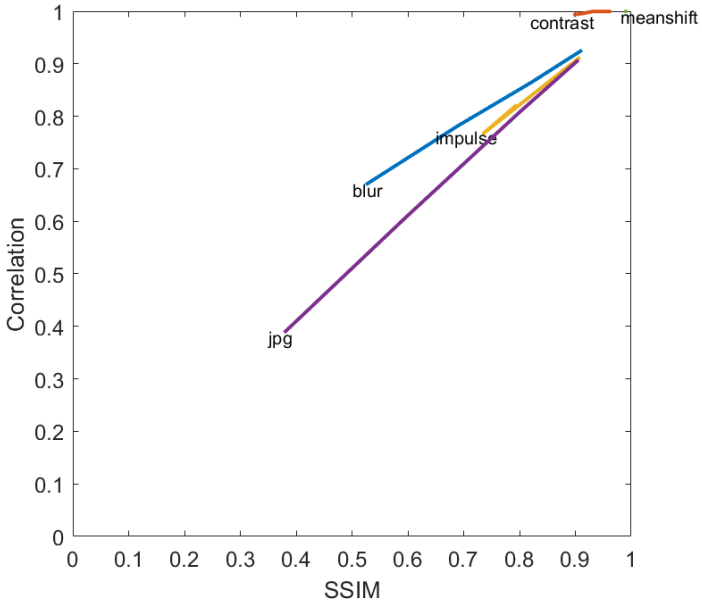


Figure 4.2: Plot of the ordered pairs (SSIM, correlation) for all degraded images and patch sizes.

and *meanshift* are concentrated near the top right of the plot, i.e., near $(1, 1)$, from which little can be concluded. As for the other three images, there is a general correlation between the SSIM and correlation values which is the result of their simultaneous increase

with block size. Indeed, even in the case of the exceptional *impulse* image, the simultaneous decrease of the SSIM and correlation values from 8×8 patches to 16×16 patches produces ordered pairs which still lie very roughly along a straight line.

All told, the results of our simple experiment suggest that there is reason to be hopeful that correlation alone may be used in place of the entire SSIM to assess visual quality. But before any such conclusions should be drawn and the first two components of the SSIM potentially tossed aside, it is only prudent to examine what information the S_1 and S_2 terms contribute to the computation.

In Table 4.5 are presented the S_1 scores of the degraded Einstein images, computed using the same method previously described. Recall that the S_1 values are dependent upon the mean intensity of the individual image patches.

	8×8	16×16	32×32	64×64
blur	0.9995	0.9999	1.0000	1.0000
contrast	0.9275	0.9433	0.9616	0.9952
impulse	0.9997	0.9999	1.0000	0.9952
jpg	0.9983	0.9993	0.9998	1.0000
meanshift	0.9877	0.9887	0.9899	0.9927

Table 4.5: Average S_1 values between the *original* Einstein image and its degradations for various patch sizes.

For each fixed patch size n, the S_1 values of all images lie very close to 1.0—with the exception of the *contrast* image, whose S_1 scores are slightly lower. Because the "contrast stretch" exaggerates the greyscale intensities across the image, the mean intensities, especially in small image patches, are more likely to be perturbed away from those in *original*. Thus the S_1 scores for *contrast* are understandably more penalized compared to the other distortions, such as impulse noise and blurring, which have less impact on local image intensity. *constrast* aside, the S_1 scores of all other images for all patch sizes belong in the interval $[0.99, 1.00]$ when rounded to two decimal places. As such, no ranking can be performed. Indeed, we would not expect much, if any, ranking capacity since the S_1 values are determined only by the mean values of image blocks.

In Table 4.6 are presented the values of the second component, S_2, of the SSIM as computed for the Einstein images. Recall that the S_2 is formulated to measure the similarities of local patch contrasts.

In the case of 8×8 patches, a significant differentiation between the degraded images is achieved, from which a ranking consistent with Eq. (4.1) can be assigned. The degree

	8×8	16×16	32×32	64×64
blur	0.7481	0.8622	0.9433	0.9845
contrast	0.9679	0.9689	0.9686	0.9689
impulse	0.9083	0.9108	0.9699	0.9948
jpg	0.4862	0.8113	0.9840	0.9980
meanshift	1.0000	1.0000	1.0000	1.0000

Table 4.6: Average S_2 values between the *original* Einstein image and its degradations for various patch sizes.

of differentiation significantly lessens as the block size n increases. Moreover, for 16×16 patches, this ranking falters due to the reversal of the *jpg* and *blur* images. For both patches of size 32×32 and 64×64, the ranking of the *jpg* image has increased further, surpassing even the *impulse* and *contrast* images.

From the above results, it appears that the S_1 term of the SSIM function performs a very minor role, if any, in differentiating between images. Although the S_2 term does differentiate between images, the range of the S_3 scores is more spread. This is especially true as the blocksize n increases: For large n, the S_2 values differ little between the images, while the S_3 scores are still reflecting more significant differences in visual quality. To complete our analysis, we now present the values of the product S_2S_3 in Table 4.7.

	8×8	16×16	32×32	64×64	MSSIM
blur	0.5249	0.6892	0.8216	0.9119	0.6940
contrast	0.9618	0.9646	0.9682	0.9686	0.9133
impulse	0.7951	0.7352	0.8211	0.9082	0.8317
jpg	0.3792	0.5969	0.8005	0.9056	0.6624
meanshift	1.0000	1.0000	1.0000	1.0000	0.9884

Table 4.7: Average product values S_2S_3 between the *original* Einstein image and its degradations for various patch sizes. The MSSIM is included in the last column for comparison.

There is, as expected, a high degree of similarity between the S_2S_3 entries in Table 4.7 and the SSIM values in Table 4.3. The difference between the first three entries of the *contrast* row, which have elevated S_2S_3 values compared to SSIM values, can of course be explained by their low corresponding S_1 values in Table 4.5.

4.3 Weberized Distance Between the Einstein Images

Before concluding this chapter, we would like to examine the Weberized L^2 distances between the *original* Einstein image and its perturbations. We are interested to see if they are roughly constant as observed for the L^2 distances presented in Table 4.1. We have used the following formula, which is the discrete $2D$ analogue of our distance function discussed in Chapter 3 with $a = 1$,

$$d_W(u,v) = \frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{[u_{ij}+1]^2}[u_{ij}-v_{ij}]^2\right]^{1/2}, \quad 0 \le u_{ij}, v_{ij} \le 2^M - 1, \tag{4.3}$$

where $N = 256$ and $M = 8$. Note from the denominator that the greyscale values have been artificially shifted upward by 1 (this shift is cancelled in the numerator) since the value of zero is included in the greyscale range 0 to 255. The resulting Weberized distances are presented in Table 4.8.

blur	*contrast*	*impulse*	*jpg*	*meanshift*
0.5161	0.3041	0.2222	0.2466	0.2011

Table 4.8: Weberized L^2 distances between the *original* Einstein image and its perturbations.

The much lower values of these distances are due to the normalizing influence of the denominator in Eq. (4.3). Interestingly, these distances are *not* roughly equal to each other. In fact, the rather significant deviation of the distance of the *blur* image from the others is quite interesting and worthy of more investigation.

Finally, we note that there is little, if any, correlation between the Weberized distances and the visual quality of the distorted images as characterized by, say, the MSSIM values. Yes, the *meanshift* image, which, according to the MSSIM, was closest to *original* in quality, is also closest in terms of the Weberized distance. But the worst quality image, *jpg*, lies closer to *original* in Weberized distance than the next best quality image, *contrast*. The Weberized distance is a modified L^2 distance which contains no explicit information about correlation. As such, we expect it to play a secondary role to correlation in image quality assessment.

Chapter 5

SSIM-based Best Approximation Using Orthonormal Functions

5.1 Previous Work on SSIM-based Approximation of Functions Using Orthonormal Functions

We concluded in the preceding chapter that our Weberized distance cannot alone characterize the visual quality of the Einstein images because it suffers from an absence of correlation-based information. This motivates us to explore, in this section, the possibility of incorporating the correlation into our Weberized best-approximation problem from Chapter 3.

The formalism of this chapter follows closely from the mathematics presented in [3], where the entire SSIM function is maximized using orthonormal basis functions. We also benefit from important results in [2], which investigates the best approximation problems seeking to maximize the individual component functions S_1, S_2, and S_3. Below, we will re-establish a special case of the main result in [3] for the reader's benefit. While [3] and [2] are both concerned with the discrete case (i.e., the M-dimensional best approximation of a signal $x \in \mathbb{R}^N$), we will re-establish the result for functions of a continuous real variable.

Indeed, consistent with Chapter 3, we will once again be considering function approximation over the space $L^2[0,1]$. The result of interest pertains to maximizing the local SSIM function, recalled below,

$$\mathrm{SSIM}(u,v) = \frac{2\bar{u}\bar{v} + C_1}{\bar{u}^2 + \bar{v}^2 + C_1} \; \frac{2\sigma_u\sigma_v + C_2}{\sigma_u^2 + \sigma_v^2 + C_2} \; \frac{\sigma_{uv} + C_3}{\sigma_u\sigma_v + C_3}. \tag{5.1}$$

Although the local SSIM is usually computed in overlapping sliding windows, we will analyze the simpler case where the SSIM function is defined over non-overlapping blocks. This is a simplifying assumption which we have inherited from [3], as it ensures that a simple analytic solution for the unique global maximum of the SSIM function is admitted. Very briefly, the result to be re-established states that the optimal SSIM-based approximation may be determined from the optimal L^2 approximation as follows: The first-order coefficients are the same, and the higher order SSIM coefficients are obtained from their Fourier counterparts by scaling.

In re-establishing the above result, our focus will be restricted to a special case which corresponds to setting $C_1 = C_2 = C_3 = 0$ in Eq. (5.1). This choice yields the "collapsed" form of the SSIM, as described on p. 12 and stated below,

$$\mathrm{SSIM}(u, v) = \frac{4\bar{u}\bar{v}\sigma_{uv}}{(\bar{u}^2 + \bar{v}^2)(\sigma_u^2 + \sigma_v^2)}. \tag{5.2}$$

The more general result provided in [3] allows for arbitrary stability constants and, as a result, looks messier. That being said, the proof is no more complicated in nature and follows the same process pursued below.

When introducing the SSIM on p. 12, we defined the quantities in Eq. (7.39) in terms of discrete statistics. Here, we will work with their continuous analogues, defined for $u, v \in L^2[0, 1]$ as follows,

$$\bar{u} = \int_0^1 u(x)dx$$

and

$$\begin{aligned}
\sigma_u^2 &= \int_0^1 (u(x) - \bar{u})^2 dx \\
&= \int_0^1 u(x)^2 dx - 2\bar{u}\int_0^1 u(x)dx + \bar{u}^2 \int_0^1 dx \\
&= \overline{u^2} - 2\bar{u}^2 + \bar{u}^2 \\
&= \overline{u^2} - \bar{u}^2.
\end{aligned}$$

Similar expressions hold for v. Finally,

$$\begin{aligned}\sigma_{uv} &= \int_0^1 (u(x)-\bar{u})(v(x)-\bar{v})dx \\ &= \int_0^1 u(x)v(x)dx - \bar{v}\int_0^1 u(x)dx - \bar{u}\int_0^1 v(x)dx + \bar{u}\bar{v}\int_0^1 dx \\ &= \overline{uv} - \bar{v}\bar{u} - \bar{u}\bar{v} + \bar{u}\bar{v} \\ &= \overline{uv} - \bar{u}\bar{v}.\end{aligned}$$

Let $\{\phi_k\}_{k=1}^{\infty}$ denote a complete orthonormal basis in $L^2[0,1]$. We once again assume that u is a reference signal to be approximated and v_M is an M-dimensional best approximation to u. This implies the following expansions,

$$u(x) = \sum_{k=1}^{\infty} a_k\phi_k \qquad \text{and} \qquad v_M(x) = \sum_{k=1}^{M} c_k\phi_k, \tag{5.3}$$

where

$$a_k = \langle u, \phi_k\rangle = \int_0^1 u(x)\phi_k(x)dx, \quad k \geq 1,$$

and the c_k, $1 \leq k \leq M$, are to be determined. We further require that the first element of the basis is "flat", i.e., $\phi_1(x) = 1$, $0 \leq x \leq 1$, and that all other basis functions are zero-mean, i.e., $\overline{\phi_k} = \int_0^1 \phi_k(x)dx = 0$, $k \geq 2$.

We can express the quantities in the SSIM function in terms of the series expansions in Eq. (5.3). Beginning with the mean value $\bar{u}$, we have

$$\bar{u} = \int_0^1 u(x)dx = \int_0^1 \sum_{k=1}^{\infty} a_k\phi_k(x)dx = \sum_{k=1}^{\infty} a_k \int_0^1 \phi_k(x)dx = a_1, \tag{5.4}$$

where the final simplification is due to the assumptions on the basis functions ϕ_k. Similarly,

$$\bar{v}_M = \int_0^1 v_M(x)dx = \int_0^1 \sum_{k=1}^{M} c_k\phi_k(x)dx = c_1. \tag{5.5}$$

The orthonormality of the basis functions ϕ_k is used to simplify the variance, as shown

below,

$$\begin{aligned}
\sigma_u^2 &= \overline{u^2} - \bar{u}^2 \\
&= \int_0^1 \left[\sum_{k=1}^{\infty} a_k \phi_k(x)\right]^2 dx - a_1^2 \\
&= \sum_{k=1}^{\infty}\sum_{l=1}^{\infty} a_k a_l \int_0^1 \phi_k \phi_l dx - a_1^2 \\
&= \sum_{k=1}^{\infty} a_k^2 - a_1^2 \\
&= \sum_{k=2}^{\infty} a_k^2. \qquad (5.6)
\end{aligned}$$

In the same way, one will also obtain

$$\sigma_{v_M}^2 = \sum_{k=2}^{M} c_k^2. \qquad (5.7)$$

And finally, the covariance simplifies similarly,

$$\begin{aligned}
\sigma_{uv_M} &= \overline{uv}_M - \bar{u}\bar{v}_M \\
&= \int_0^1 \left[\sum_{k=1}^{\infty} a_k \phi_k(x) \sum_{l=1}^{M} c_l \phi_l(x)\right] dx - a_1 c_1 \\
&= \sum_{k=1}^{\infty}\sum_{l=1}^{M} a_k c_l \int_0^1 \phi_k \phi_l dx - a_1 c_1 \\
&= \sum_{k=1}^{M} a_k c_k - a_1 c_1 \\
&= \sum_{k=2}^{M} a_k c_k. \qquad (5.8)
\end{aligned}$$

We will obtain an expression for the optimal SSIM-based coefficients by way of the stationarity conditions. To simplify the exercise, we will use logarithmic differentiation on Eq. (7.39). First, taking the logarithm yields

$$\log(\mathrm{SSIM}(u, v_M)) = \log(4\bar{u}) + \log(\bar{v}_M) + \log(\sigma_{uv_M}) - \log(\bar{u}^2 + \bar{v}_M^2) - \log(\sigma_u + \sigma_{v_M}).$$

Now differentiate the above expression with respect to each coefficient c_p, $1 \le p \le M$,

$$\frac{1}{\mathrm{SSIM}(u, v_M)} \frac{\partial \mathrm{SSIM}(u, v_M)}{\partial c_p} = \frac{1}{4\bar{u}} \frac{\partial \bar{u}}{\partial c_p} + \frac{1}{\bar{v}_M} \frac{\partial \bar{v}_M}{\partial c_p} + \frac{1}{\sigma_{uv_M}} \frac{\partial \sigma_{uv_M}}{\partial c_p} - \frac{1}{\bar{u}^2 + \bar{v}_M^2} \frac{\partial \bar{u}^2}{\partial c_p} - \frac{1}{\bar{u}^2 + \bar{v}_M^2} \frac{\partial \bar{v}_M^2}{\partial c_p} - \frac{1}{\sigma_u^2 + \sigma_{v_M}^2} \frac{\partial \sigma_u^2}{\partial c_p} - \frac{1}{\sigma_u^2 + \sigma_{v_M}^2} \frac{\partial \sigma_{v_M}^2}{\partial c_p}. \tag{5.9}$$

Clearly, the partial derivatives involving only u vanish, namely,

$$\frac{\partial \bar{u}}{\partial c_p} = \frac{\partial \bar{u}^2}{\partial c_p} = \frac{\partial \sigma_u^2}{\partial c_p} = 0.$$

After multiplying by $\mathrm{SSIM}(u, v_M)$, we have already simplified Eq. (5.9) greatly, written below for $1 \le p \le M$,

$$\frac{\partial \mathrm{SSIM}}{\partial c_p} = \mathrm{SSIM} \left[\frac{1}{\bar{v}_M} \frac{\partial \bar{v}_M}{\partial c_p} + \frac{1}{\sigma_{uv_M}} \frac{\partial \sigma_{uv_M}}{\partial c_p} - \frac{1}{\bar{u}^2 + \bar{v}_M^2} \frac{\partial \bar{v}_M^2}{\partial c_p} - \frac{1}{\sigma_u^2 + \sigma_{v_M}^2} \frac{\partial \sigma_{v_M}^2}{\partial c_p} \right]. \tag{5.10}$$

The remaining partial derivatives can be computed easily using the series expansions in Eqs. (5.4)-(5.8), as follows,

$$\frac{\partial \bar{v}_M}{\partial c_p} = \frac{\partial}{\partial c_p} c_1 = \begin{cases} 1, & p = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{\partial \bar{v}_M^2}{\partial c_p} = \frac{\partial}{\partial c_p} c_1^2 = \begin{cases} 2c_0, & p = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{\partial \sigma_{uv_M}}{\partial c_p} = \frac{\partial}{\partial c_p} \sum_{k=2}^{N} a_k c_k = \begin{cases} 0, & p = 1 \\ a_p, & \text{otherwise} \end{cases}$$

$$\frac{\partial \sigma_{v_M}^2}{\partial c_p} = \frac{\partial}{\partial c_p} \sum_{k=2}^{N} c_k^2 = \begin{cases} 0, & p = 1 \\ 2c_p, & \text{otherwise} \end{cases}$$

Once again, the partial derivatives simplify nicely due to the assumptions on the basis functions ϕ_k. In general, the RHS of Eq. (5.9) is a complicated nonlinear expression in the coefficients c_p and a solution of the equations $\frac{\partial \mathrm{SSIM}}{\partial c_p} = 0$ is intractable.

At a relative minimum or maximum of $\mathrm{SSIM}(u, v_M)$, the M partial derivatives described by Eq.(5.10) must vanish. We substitute the expression obtained above in Eq. (5.10) to obtain the following stationarity conditions.

First, for $p = 1$:

$$\frac{\partial \text{SSIM}(u, v_M)}{\partial c_1} = \text{SSIM}(u, v_M) \left[\frac{1}{\bar{v}_M} - \frac{2c_1}{\bar{u}^2 + \bar{v}_M^2} \right] = 0.$$

Assuming SSIM $\neq 0$ and recalling that $\bar{u} = a_1$ and $\bar{v}_M = c_1$, we get

$$2c_1^2 = a_1^2 + c_1^2 \implies c_1 = \pm a_1. \tag{5.11}$$

We rule out the solution $c_1 = -a_1$ because it will yield an image whose mean is $-a_1 = -\bar{u}$.

For $p \neq 1$:

$$\frac{\partial \text{SSIM}(u, v_M)}{\partial c_p} = \text{SSIM}(u, v_M) \left[\frac{a_p}{\sigma_{uv_M}} - \frac{2c_p}{\sigma_u^2 + \sigma_{v_M}^2} \right] = 0$$

Again assume SSIM $\neq 0$, and observe that when $a_p = 0$, then we must have $c_p = 0$. For $c_p \neq 0$, we can rearrange to obtain

$$\frac{a_p}{c_p} = \frac{2\sigma_{uv_M}}{\sigma_u^2 + \sigma_{v_M}^2}, \quad 2 \leq p \leq M. \tag{5.12}$$

Notice that the RHS of Eq. (5.13) is constant for all p. In other words, for some constant $\alpha \neq 0$, we have

$$\frac{a_2}{c_2} = \frac{a_3}{c_3} = \cdots = \frac{a_M}{c_M} = \frac{1}{\alpha}$$

or, after inverting,

$$c_p = \alpha a_p, \quad 2 \leq p \leq M. \tag{5.13}$$

At this point, we have determined all M SSIM-based coefficients in terms of their Fourier counterparts. It only remains to find α. To accomplish this, we return to Eq. (5.12) and substitute the relation in Eq. (5.13) to get

$$\frac{a_p}{\alpha a_p} = \frac{2 \sum_{k=2}^{M} \alpha a_k a_k}{\sigma_u^2 + \sum_{k=2}^{M} (\alpha a_k)^2}$$

Which can be rearranged as follows,

$$\sigma_u^2 + \alpha^2 \sum_{k=2}^{M} a_k^2 = 2\alpha^2 \sum_{p=2}^{M} a_k^2$$

$$\alpha^2 = \sigma_u^2 \left[\sum_{k=2}^{M} a_k^2 \right]^{-1}$$

$$\alpha_{1,2} = \pm \sigma_u \left[\sum_{k=2}^{M} a_k^2 \right]^{-1/2}.$$

After substituting the expression for σ_u, this becomes

$$\alpha_{1,2} = \pm \left[\sum_{k=2}^{\infty} a_k^2 \right]^{1/2} \left[\sum_{k=2}^{M} a_k^2 \right]^{-1/2}.$$

We choose $\alpha = \alpha_1 > 0$ to be the admissible root, noting that $\alpha_1 \to 1$ as $M \to \infty$. In words, the optimal SSIM-based solution approaches the best L^2 expansion in this limit.

To summarize, we have shown that the M-dimensional approximation v_M which maximizes the SSIM function can be computed from the best-L^2 approximation by taking

$$c_1 = a_1 \tag{5.14}$$

and

$$c_k = \alpha a_k, \quad 2 \le k \le M, \quad \text{where} \quad \alpha = \left[\sum_{k=2}^{\infty} a_k^2 \right]^{1/2} \left[\sum_{k=2}^{M} a_k^2 \right]^{-1/2} \ge 1. \tag{5.15}$$

Our arrival at this simple relation between the best-SSIM coefficients and the best-L^2 coefficients is both surprising and interesting. The ease with which one can obtain the best-SSIM coefficients directly from their L^2 counterparts is undoubtedly attractive from a computational perspective. This practical consideration aside, the result also has an aesthetic appeal. Since $a_1 = c_1$, it follows that the means of the reference signal and the best-SSIM approximation are equal, i.e., $\bar{u} = \bar{v}_M$. Regarding the other coefficients, the scaling factor $\alpha \ge 1$ produces coefficients c_k which can only be equal to or larger in magnitude than the a_k. As long as $\alpha \ne 1$, the SSIM-based approximation v_M represents a *contrast-enhanced* version of the reference signal u.

There is one other important property of the SSIM-based approximation v_M—its norm is equal to the norm of u, the function it is approximating. This is easy to show:

$$\begin{aligned}
\|v_M\|^2 &= c_1^2 + \sum_{k=2}^{M} c_k^2 \\
&= a_1^2 + \alpha^2 \sum_{k=2}^{M} a_k^2 \\
&= a_1^2 + \left[\sum_{k=2}^{\infty} a_k^2\right] \left[\sum_{k=2}^{M} a_k^2\right]^{-1} \sum_{k=2}^{M} a_k^2 \\
&= a_1^2 + \left[\sum_{k=2}^{\infty} a_k^2\right] \\
&= \|u\|^2
\end{aligned} \tag{5.16}$$

Let us summarize these two important properties since they will reappear later in this thesis. The target function u and its SSIM-based best approximation v_M are related as follows:

1. Equal means, i.e., $\overline{u} = \bar{v}_M$
2. Equal norms, i.e., $\|u\| = \|v_M\|$

Indeed, the relations $\bar{u} = a_1$ and $\bar{v}_M = c_1$ also enable the neat elimination of the first coefficient in each of the sums σ_u^2, $\sigma_{v_M}^2$, σ_{uv_M}, and α. Although we didn't draw attention to this simplification during our derivation above, it is of course very satisfying. But in the context of our new work in this thesis, where we have been concerned with accommodating Weber's model of perception, the absence of a_1 from α is potentially of concern. By immediate consequence of the much recalled relation $\bar{u} = a_1$, it follows that the parameter α is also independent of the mean of the reference signal u. In a Weberized approach, the parameter α might be expected to decrease in magnitude as the mean value $\bar{u}$ is increased.

On the other hand, the absence of the coefficient a_1 in the expression for α in Eq. (5.15) may be viewed as a kind of "blessing" in terms of Weberization. Suppose that the expres-

sion for α were instead as follows,

$$\alpha = \left[\sum_{k=1}^{\infty} a_k^2\right]^{1/2} \left[\sum_{k=1}^{M} a_k^2\right]^{-1/2} \geq 1.$$

Now for a fixed M, consider the case that a_1 becomes very large. In that case, Weber's model states that the human visual system will be tolerant of greater deviations between u and its approximation v_M. One would expect that α would be allowed to increase. But as $a_1 \to \infty$, $\alpha \to 1^+$. This is rather "anti-Weberization". In this way, the fact that a_1 does not appear in the expression for α could be viewed as a counter measure against "anti-Weberization".

5.2 The Quest for Another Weberized Image Quality Measure

Having re-established the related result, we are now prepared to present our new contributions. They follow closely from the preceding chapter, and are still driven by our belief that the correlation is the most important component of the SSIM. Although our simple experiments on the Einstein images were never intended to be sufficient to convince ourselves or the reader totally of our guiding claim, the data presented in Chapter 4 does suggest that together the S_2 and S_3 components are almost entirely responsible for the total differentiation by the SSIM. In other words, at the very least, our experiments strongly suggest that the S_1 term is the least important component of the SSIM.

A principled defense of the S_1 term is that it is alone responsible for any accommodation of Weber's model of perception achieved by the SSIM function. Of course, in practice, any "Weberization" effect is not overwhelmingly pronounced in the S_1 scores (at least for the albeit limited set of Einstein images). Moreover, as discussed in the previous section, the SSIM-based best approximation result displays a limited capacity to accommodate Weber's model of perception. At best, we find an ambivalent commitment to Weber's model by way of the insurance against "anti-Weberizing" in the form of the parameter α.

At this point, our discussion may be arising speculation that we could simply discard the S_1 term in the above SSIM-based best approximation problem with little consequences. But that is not at all the case. The best approximation problems using only one or either of the S_2 or S_3 terms was investigated in [2]. It was found that at least two of the three components of the SSIM, one of which must be the luminance term, S_1, must be present in order to provide a unique solution for the best approximation problem.

So the focus of this section is not on discarding the S_1 term, but rather on replacing it with something else. In the best approximation problem discussed above, the "Weberizing" term of the SSIM function,

$$S_1(u,v) = \frac{2\bar{u}\bar{v}}{\bar{u}^2 + \bar{v}^2},$$

is maximized to the value of 1 if the means of u and v are made equal, i.e., if $c_1 = a_1$ (see Eq. (5.15)). The resulting Weberization is an average Weberization involving means. It is not enforced pointwise over individual elements of the vector.

By comparison, in Chapter 3 we developed a distance function that accommodates Weber's model on a more pointwise level. As we observed during our experiments on the Einstein images, when used for image quality assessment, this metric suffers from an absence of correlation-based information. In what follows, we will incorporate the correlation into our Weberized best approximation problem.

With these ideas in mind, we will consider the following distance function between u and v in $L^2[0,1]$,

$$G_{a,\lambda}(u,v) = [\Delta_a(u,v)]^2 + \lambda\,[1 - S_3(u,v)]\,, \qquad a, \lambda \geq 0, \tag{5.17}$$

where the Weberized metric, previously defined in Chapter 3, is recalled below,

$$\Delta_a(u,v) = \left[\int_0^1 \frac{1}{u(x)^{2a}}[u(x) - v(x)]^2 dx\right]^{1/2}.$$

Once again, the two terms which comprise the objective function $G_{a,\lambda}(u,v)$ both play their own important role in the minimization:

1. $[\Delta_a(u,v)]^2$: For $a > 0$, it will impose a kind of pointwise Weberization to the best approximation problem, as opposed to the mean-value Weberization imposed by the $S_1(u,v)$ term in the SSIM function.

2. $[1 - S_3(u,v)]$: By seeking to minimize this function, we try to maximize the correlation $S_3(u,v)$ between the approximation v and the target function u.

The parameters a and λ are both important:

1. The parameter a is the Weber exponent. Recall that the traditional form of "Weber's Law" corresponds to $a = 1$.

2. As λ increases, we are enforcing a greater pressure on the best approximation v to be correlated with u.

Recall that in the special case $a = 0$, the Weberized metric $\Delta_a(u, v)$ is the usual L^2 distance between u and v, i.e.,

$$\Delta_0(u, v) = \left[\int_0^1 [u(x) - v(x)]^2 dx\right]^{1/2} = \|u - v\|_2.$$

As such, we may consider the distance function $G_{0,0}(u, v)$ as a kind of reference distance function since

$$G_{0,0}(u, v) = \|u - v\|_2^2.$$

Note that we have chosen the parameter λ to multiply the correlation term $[1 - S_3(u, v)]$ in Eq. (5.17). We consider this term to be a perturbation of the squared Weberized L^2 distance term $[\Delta_a(u, v)]^2$. The reason for this is that the "unperturbed problem", i.e., that which corresponds to $\lambda = 0$, is well-behaved: It is simply the best Weberized L^2 approximation, for which a unique solution exists [27]. Moreover, the solution of the unperturbed problem involves solving a system of M equations which depend *linearly* on the unknown coefficients c_k. On the other hand, as we shall see below, the correlation $S_3(u, v)$ is a rather complicated function of the coefficients c_k. Once again, in [2], it was shown that the best approximation problem using only the $S_3(u, v)$ portion of the SSIM function is not unique.

5.2.1 Solving the Weberized Best Approximation Problem with Correlation as a Regularization Term

A mathematical investigation of the modified best approximation problem can proceed easily thanks to our work in Section 5.1 re-establishing the result from [3]. As before, let $\{\phi_k\}_{k=1}^{\infty}$ denote a complete orthonormal basis in $L^2[0, 1]$. The following assumptions on the basis functions are still in effect: $\phi_1 = 1$ and $\bar{\phi}_k = 0$ for all $k \geq 2$.

For given values of $a \geq 0$ and $\lambda \geq 0$, we wish to find the function of the following form,

$$v_M = \sum_{k=1}^{M} c_k \phi_k(x),$$

which minimizes the distance function $G_{a,\lambda}(u, v_M)$ in Eq. (5.17). The next step is to express each of the two terms in the objective function $G_{a,\lambda}(u, v_M)$ in terms of the coefficients c_k, $1 \leq k \leq M$.

The Weberization term,

$$[\Delta_a(u, v_M)]^2 = \int_0^1 \frac{1}{u(x)^{2a}}[u(x) - v_M(x)]^2 dx,$$

can easily be differentiated with respect to each coefficient c_p, for $1 \le p \le M$,

$$\begin{aligned}
\frac{\partial[\Delta_a(u, v_M)]^2}{\partial c_p} &= -2\int_0^1 \frac{1}{u(x)^{2a}}\left[u(x) - \sum_{k=1}^{M} c_k\phi_k(x)\right]\phi_p(x)dx \\
&= 2\sum_{k=1}^{M} c_k \int_0^1 \frac{1}{u(x)^{2a}}\phi_k(x)\phi_p(x)dx - 2\int_0^1 \frac{1}{u(x)^{2a}}u(x)\phi_p(x)dx \\
&= 2\sum_{k=1}^{M} A_{kp}c_k - 2b_p.
\end{aligned}$$

In the "perturbed" case $\lambda > 0$, the solution to these linear equations may provide a suitable starting point for some iterative method, e.g., gradient descent, designed to find solutions to the perturbed problems.

The quantities σ_u, σ_{v_M}, and σ_{uv_M} were previously computed in Section 5.1. In terms of these expressions, the correlation becomes

$$\begin{aligned}
S_3(u, v_M) &= \frac{\sigma_{uv_M}}{\sigma_u\sigma_{v_M}} \\
&= \left[\sum_{k=2}^{M} a_k c_k\right]\left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2}\left[\sum_{k=2}^{M} c_k^2\right]^{-1/2}. \qquad (5.18)
\end{aligned}$$

As expected, $S_3(u, v_M)$ is a nonlinear function of the coefficients c_k. We'll use logarithmic differentiation to compute the partial derivatives of S_3. First,

$$\log S_3(u, v_M) = \log\left[\sum_{k=2}^{M} a_k c_k\right] - \frac{1}{2}\log\left[\sum_{k=2}^{\infty} a_k^2\right] - \frac{1}{2}\log\left[\sum_{k=2}^{M} c_k^2\right].$$

Taking the partial derivative with respect to c_p, $1 \le p \le M$,

$$\frac{1}{S_3}\frac{\partial S_3}{\partial c_p} = \left[\sum_{k=2}^{M} a_k c_k\right]^{-1} a_p - \left[\sum_{k=2}^{M} c_k^2\right]^{-1} c_p.$$

Multiplication by S_3 yields,

$$\frac{\partial S_3}{\partial c_p} = a_p \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \left[\sum_{k=2}^{M} c_k^2\right]^{-1/2} - c_p \left[\sum_{k=2}^{M} a_k c_k\right] \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \left[\sum_{k=2}^{M} c_k^2\right]^{-3/2} \tag{5.19}$$

or

$$\frac{\partial S_3}{\partial c_p} = \frac{a_p}{\sigma_u \sigma_{v_M}} - \frac{c_p \sigma_{u v_M}}{\sigma_u \sigma_{v_M}^3}.$$

We now combine the results from the equation to compute the gradient of the objective function

$$G_{a,\lambda}(u, v_M) = \left[\Delta_a(u, v_M)\right]^2 + \lambda \left[1 - S_3(u, v_M)\right], \quad a, \lambda \geq 0.$$

For $1 \leq p \leq M$,

$$\frac{\partial G_{a,\lambda}(u, v_M)}{\partial c_p} = 2 \sum_{k=1}^{M} A_{kp} c_k - 2 b_p - \lambda \left[\frac{a_p}{\sigma_u \sigma_{v_M}} - \frac{c_p \sigma_{u v_M}}{\sigma_u \sigma_{v_M}^3}\right] = 0. \tag{5.20}$$

Before looking for solutions to Eq (5.20), let us first investigative the correlation in Eq. (5.18). This discussion is related to following theorem for vectors in $\mathbb{R}^N$, stated in [2].

Theorem 3. Let $x \in \mathbb{R}^N$ and $y = ax + b1_N$, where $a, b \in \mathbb{R}$, $a \neq 0$ and $1_N = (1, 1, \cdots, 1) \in \mathbb{R}^N$. Then

$$S_3(x, y) = \operatorname{sgn}(a) = \begin{cases} 1, & a > 0 \\ -1, & a < 0. \end{cases} \tag{5.21}$$

This result cannot be directly applied to our discussion of functions of a continuous real variable. In particular, the expression for σ_u is an infinite series.

In the special case that $c_k = a_k$, $1 \leq k \leq M$, v_M is simply a finite-dimensional truncation of u. The correlation in Eq. (5.18) becomes

$$S_3(u, v_M) = \left[\sum_{k=2}^{M} a_k^2\right]^{1/2} \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \leq 1. \tag{5.22}$$

In this case, v_M is the best L^2-based approximation to u. We expect that it is also the maximizer of $S_3(u, v)$. Indeed, in the case $c_k = a_k$, $1 \leq k \leq M$, the partial derivatives of

S_3 are zero. This is easy to see by substitution into Eq. (5.19):

$$\frac{\partial S_3}{\partial c_p} = a_p \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \left[\sum_{k=2}^{M} a_k^2\right]^{-1/2} - a_p \left[\sum_{k=2}^{M} a_k^2\right] \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \left[\sum_{k=2}^{M} a_k^2\right]^{-3/2} = 0.$$

On the $(M-1)$-dimensional sphere S_R of radius R, i.e.,

$$S_R = \left\{ (c_1, \cdots, c_M) \left| \sum_{k=1}^{M} c_k^2 = R^2 \right. \right\},$$

S_3 assumes its maximum value given in Eq. (5.22) above at the point $c_k = a_k$, $1 \le k \le M$. As $R \to \infty$, this maximum value approaches the value 1.

In the case $c_k = -a_k$, $1 \le k \le M$,

$$S_3(u, v_M) = -\left[\sum_{k=2}^{M} a_k^2\right]^{1/2} \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \ge -1. \tag{5.23}$$

In this case, v_M is the minimizer of $S_3(u, v)$. This is once again easy to show by substitution into Eq. (5.19):

$$\begin{aligned}\frac{\partial S_3}{\partial c_p} &= a_p \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \left[\sum_{k=2}^{M} (-a_k)^2\right]^{-1/2} - (-a_p) \left[\sum_{k=2}^{M} a_k(-a_k)\right] \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \left[\sum_{k=2}^{M} (-a_k)^2\right]^{-3/2} \\ &= a_p \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \left[\sum_{k=2}^{M} a_k^2\right]^{-1/2} - a_p \left[\sum_{k=2}^{M} a_k^2\right] \left[\sum_{k=2}^{\infty} a_k^2\right]^{-1/2} \left[\sum_{k=2}^{M} a_k^2\right]^{-3/2} = 0.\end{aligned}$$

The functions v_M and u are at their maximum anticorrelation.

On the $(M-1)$-dimensional sphere S_R of radius R defined above, S_3 assumes its minimum value given in Eq. (5.23) at the point $c_k = -a_k$ $1 \le k \le M$. As $R \to \infty$, this minimum value approaches the value -1.

We now explore Eq. (5.20) further to see if any analytical solutions are possible. In the special non-Weber case $a = 0$, the matrix A is diagonal. When $\lambda = 0$, the above equations yield the following result, as expected,

$$c_p = b_p = a_p, \quad 1 \le p \le M.$$

Let us now consider the slightly more general case $a = 0$ and $\lambda > 0$. This represents an L^2 best approximation problem with an additional correlation term. With an eye to the re-estalished result from [3], let us assume a solution to Eq. (5.20) of the form,

$$c_k = \alpha a_k, \quad 1 \leq k \leq M. \tag{5.24}$$

Substitution into Eq. (5.20) yields the following equation,

$$2\alpha \sum_{k=1}^{M} A_{kp} a_k - 2b_p - \lambda a_p \left[\sum_{k=2}^{\infty} a_k^2 \right]^{-1/2} \left[\sum_{k=2}^{M} a_k^2 \right]^{-1/2} (\alpha^{-1} - \alpha^{-1}) = 0.$$

The entire second term corresponding to the S_3 term vanishes which seems remarkable. In retrospect, however, this is to be expected since the c and a vectors are perfectly correlated by Eq. (5.24), which represents a stationary point (maximum) in terms of the correlation function S_3. (Because $c = (c_1, \cdots, c_M) \in \mathbb{R}^M$ and $a = (a_1, \cdots, a_M) \in \mathbb{R}^M$, this result is guaranteed by Theorem 3.) The equation above becomes

$$2\alpha \sum_{k=1}^{M} A_{kp} a_k - 2b_p = 0.$$

This may appear to be disconcerting since the above equation is independent of λ. As such, it might seem that the assumption in Eq. (5.24) is invalid. But recall that we have assumed that $a = 0$, i.e., the non-Weberized case. We have not yet applied this assumption with regard to the matrix A. When $a = 0$, the matrix A is diagonal so that the above equation becomes

$$\alpha a_p = b_p, \quad 1 \leq p \leq M.$$

But from $a = 0$ we know that $a_p = b_p$, so $\alpha = 1$. This is actually to be expected. We know that the best L^2 approximation to u is provided by the Fourier coefficients a_k of u. The M-vector of c_k coefficients which is most correlated with the M-vector of a_k coefficients is given by Eq. (5.24) for arbitrary α. But if the L^2 distance $\|u - v\|_2$ is included in the objective function, $\alpha = 1$.

5.2.2 Selected Examples in Correlation-based Weberized Distance

Example 1: Consider the following step function,

$$u(x) = \begin{cases} 2, & 0 \leq x \leq 1/2 \\ 4, & 1/2 < x \leq 1. \end{cases} \tag{5.25}$$

We use the following set of functions

$$\begin{aligned}\phi_1(x) &= 1\\ \phi_k(x) &= \sqrt{2}\cos((k-1)\pi x), \quad k \geq 2,\end{aligned} \tag{5.26}$$

which form an orthonormal basis in the space of functions $L^2[0,1]$. Note that this choice accommodates our additional assumptions on the basis functions (ϕ_1 is "flat" and $\overline{\phi_k} = 0$ for all $k \neq 1$).

In Fig. 5.1 are presented the plots of the best approximations v_M to u using $M = 5$ basis functions for $a = 0.00, 0.50, 1.00$ where, in each case, $\lambda = 0.00, 0.25, 0.50, 0.75, 1.00$. The best L^2 approximation, corresponding to the case $a = 0.00$ and $\lambda = 0.00$, is also shown for comparison. To obtain the unknown coefficients in each case, we minimize the distance function in Eq. (5.17) in Maple using gradient descent, where the starting guess is initialized as the best Weberized coefficients (which corresponds to the appropriate power of a and $\lambda = 0$). The black box routine `'fsolve'` in Maple was also able to solve the stationarity conditions described by Eq. (5.20). We verified that, in each case, the coefficients obtained using the two methods matched to at least 6 decimal places.

As expected, Figure 5.1 a) confirms that when $a = 0$, for any $\lambda > 0$, the best approximation v_M is simply determined by the Fourier coefficients. Interestingly, for non-zero a, it appears that increasing λ "undoes" the Weberization. For example, Figure 5.1 b) corresponds to the case $a = 0.50$: As λ increases, the best approximations move away from the best-Weberized case ($\lambda = 0$) and towards the best-L^2 approximation. For $a = 1.00$ in Figure 5.1 c), this effect is even more pronounced: While the curves around $u = 2$ are hugging the best-L^2 approximation quite closely, around $u = 4$ the correlation-based approximation is *below* the best-L^2 curve. The Weberization at this higher intensity $u = 4$ is more than "undone", it is reduced past the unWeberized L^2 best approximation. Interestingly, the curve for $\lambda = 1$ does not stay closest to the target function for long: In fact, all correlation-based best approximations appear to meet the Weberized best approximation at $x = 1$.

For interest, we have tabulated the l^2 distances between the first M Fourier coefficients of the target function u, $(a_1, a_2, \cdots, a_M)$, and the M coefficients defining v_M, $(c_1, c_2, \cdots, c_M)$, i.e.,

$$D_{a,\lambda} = \left[\sum_{k=1}^{M}(a_k - c_k)^2\right]^{1/2}. \tag{5.27}$$

Table 5.1 reports the distances $D_{a,\lambda}$ between u and all the best approximations pictured in Figure 5.1. (Note that this l^2 distance represents the L^2 distance between the approximation v_M and the best M-dimensional L^2 approximation to v, which is sufficient for this

discussion. The L^2 distance between v_M and u would be obtained by including the sum of the squares of the infinite "tail" of coefficients c_k in Eq. (5.27).)

	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.50$	$\lambda = 0.75$	$\lambda = 1.00$
$a = 0$	0.000000	0.000000	0.000000	0.000000	0.000000
$a = 0.5$	0.100568	0.077881	0.065221	0.057478	0.052438
$a = 1.0$	0.181498	0.114293	0.097834	0.093465	0.092495

Table 5.1: l^2 distances computed according to Eq. (5.27) for $u(x)$ and its best approximations v_M, for $M = 5$ pictured in Figure 5.1.

Unsurprisingly, the row in Table 5.1 corresponding to $a = 0$ reiterates that $D_{0,\lambda} = 0$, i.e., for any $\lambda \geq 0$, there is no difference between the Fourier and correlation-based coefficients when $a = 0$. For non-zero a, the distance $D_{a,\lambda}$ decreases as λ increases towards 1. This observation agrees with our earlier remarks on Figure 5.1. Moreover, for a given λ, increasing a also increases the distance $D_{a,\lambda}$. Because a determines the degree of Weberization, this trend is also to be expected.

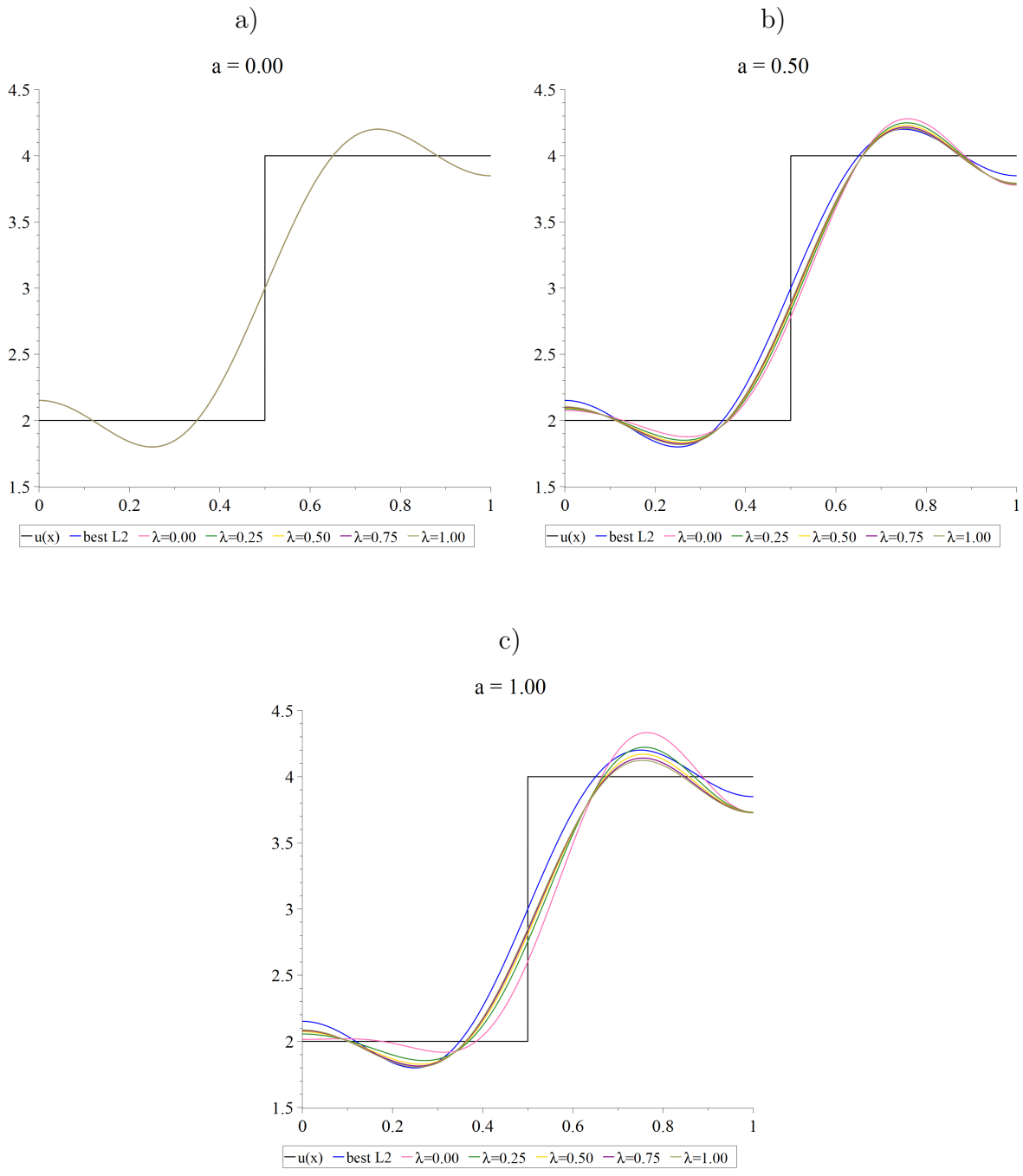


Figure 5.1: Best approximations using $a = 0$, $a = 0.5$, and $a = 1.0$ to the step function $u(x)$ in Eq. (5.25) using $M = 5$ basis functions, where in each case $\lambda = 0, 0.25, 0.5, 0.75, 1$. The best L^2 approximation (which corresponds to setting $a = 0$, $\lambda = 0$) has also been plotted for comparison.

Example 2: Consider the following "bumpy" step function,

$$w(x) = \begin{cases} 1.5, & 0 \leq x \leq 0.05 \\ 2, & 0.05 < x \leq 0.10 \\ 0.3, & 0.10 < x \leq 0.15 \\ 3, & 0.15 < x \leq 0.20 \\ 1, & 0.20 < x \leq 0.25 \\ 2.5, & 0.25 < x \leq 0.75 \\ 3.5, & 0.75 < x \leq 0.80 \\ 4, & 0.80 < x \leq 0.85 \\ 2.3, & 0.85 < x \leq 0.9 \\ 5, & 0.90 < x \leq 0.95 \\ 3, & 0.95 < x \leq 1.00 \end{cases} \tag{5.28}$$

We are interested to see if increasing the weighting λ on the correlation term will force v_M to conform more to this bumpy signal.

We will once again be employing the following basis functions,

$$\begin{aligned} \phi_1(x) &= 1 \\ \phi_k(x) &= \sqrt{2}\cos((k-1)\pi x), \quad k \geq 2. \end{aligned} \tag{5.29}$$

Figure 5.2 shows the best approximations v_M to w using $M = 20$ basis functions for $a = 0.00, 0.50, 1.00$, where $\lambda = 0.00, 0.50, 1.00$. As before, Figure 5.2 a) illustrates little but to reiterate that, when $a = 0$, all best approximations are equal to the case $\lambda = 0$.

Figure 5.2 c) shows the best approximations for $a = 1$, which is the highest degree of Weberization explored in this experiment. First consider the pink curve corresponding to $\lambda = 0$, i.e., having no correlation term. Because the Weberized distance tolerates lesser deviations at lower intensity regions, this pink curve hovers very low to match the troughs of the bumpy step function. This behaviour is especially pronounced for the low-intensity cycle at $0 \leq x \leq 0.25$. It is also in stark contrast to the much greater amplitudes of the best-L^2 curve in this region. The curves corresponding to non-zero λ move more freely in $0 \leq x \leq 0.25$, conforming less tightly to those troughs of the target function. This is to be expected, as increasing λ should "undo" the Weberization. The behaviour in $0.5 < x < 0.75$ is remarkable. As one would expect, the best-L^2 and $\lambda = 0$ solutions oscillate around the constant value $w = 2.5$. On the other hand, the correlation-based

solutions, corresponding to $\lambda = 0.50$ and $\lambda = 1.00$, hover below the target function for most of this region. This may be due to the particular test function examined. A slight perturbation of this test function could produce approximations which lie above the middle plateau. We do notice, however, that the approximations lying below the middle plateau have slightly lower amplitudes of oscillation—apparently the non-zero correlation term in the cost function is forcing them to at least correlate somewhat with the flat middle region of $w(x)$. Correlation does not imply closeness in value.

Once again, we have computed the distances between the first M coefficients a_k and c_k according to Eq. (5.27). We have computed the distances for more choices of λ than those which are plotted in Figure 5.2. The results are listed in Table 5.2.

	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.50$	$\lambda = 0.75$	$\lambda = 1.00$
$a = 0$	0.000000	0.000000	0.000000	0.000000	0.000000
$a = 0.5$	0.221589	0.193961	0.178218	0.167961	0.160842
$a = 1.0$	0.398554	0.342105	0.322702	0.310653	0.302803

Table 5.2: l^2 distances computed according to Eq. (5.27) $w(x)$ and its best approximations v_M, for $M = 20$ pictured in Figure 5.2.

It should not come as a surprise that we observe the same trends as before. At the risk of repetition, $D_{0,\lambda} = 0$ for any $\lambda \geq 0$. For a fixed $a \neq 0$, $D_{a,\lambda}$ decreases as λ increases towards 1. For a fixed λ, $D_{a,\lambda}$ increases with a towards 1.

In closing this section, we mention that the approach reported above can be extended to the two-dimensional case, i.e., images, in a straightforward way. Indeed, work in this direction was started but interrupted shortly thereafter by seemingly promising new ideas. Unfortunately, we never returned to complete this investigation and recommend that it be considered by others in the future.

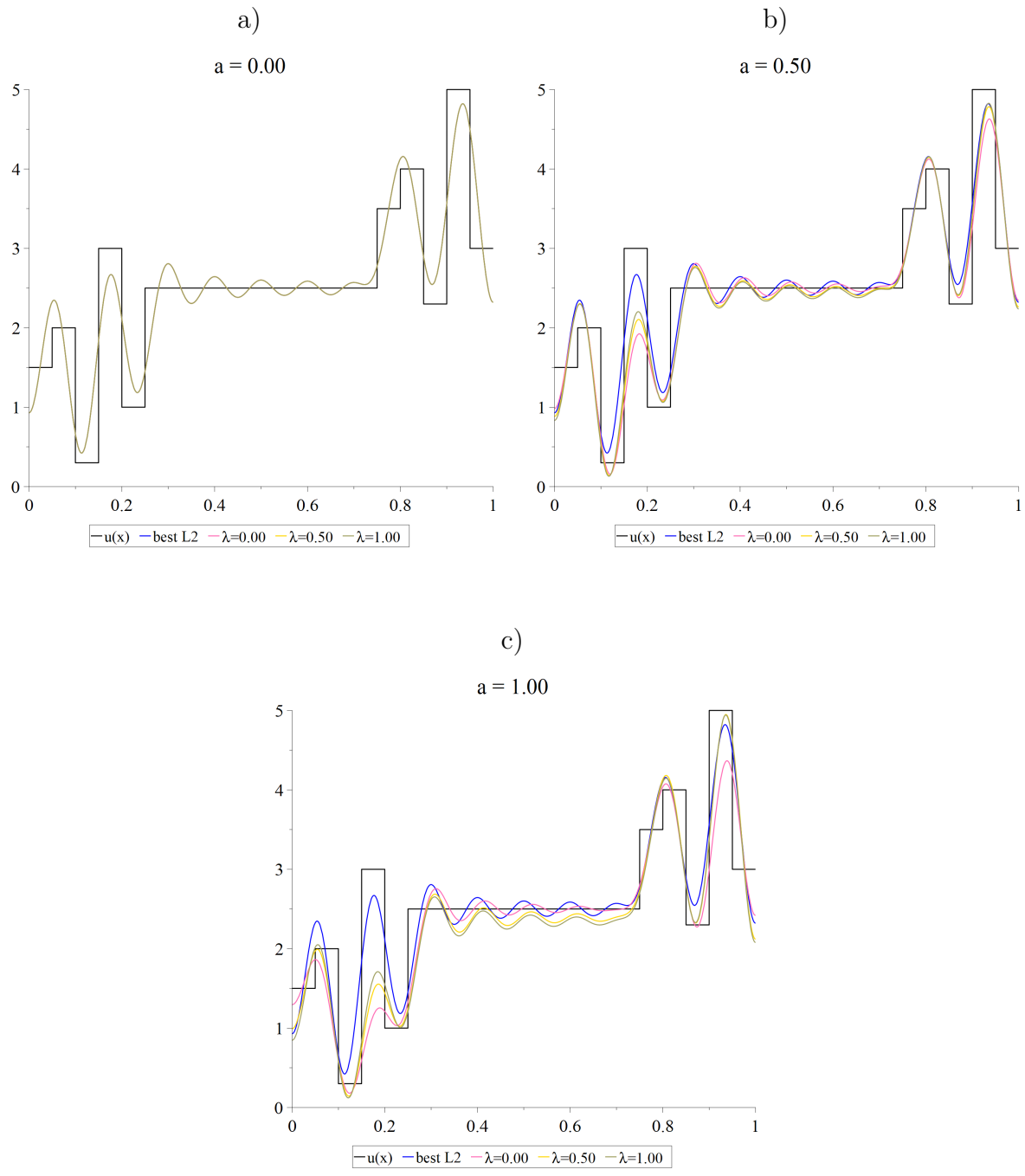


Figure 5.2: Best approximations using $a = 0$, $a = 0.5$, and $a = 1.0$ to the bumpy step function $w(x)$ in Eq. (5.28) using $M = 20$ basis functions, where in each case $\lambda = 0, 0.5, 1$. The best L^2 approximation (which corresponds to setting $a = 0$, $\lambda = 0$) has also been plotted for comparison.

5.3 Other Distance Functions Involving the Weberized Distance and Correlation

The distance function $G_{a,\lambda}$ in Eq. (5.17) contains a variable parameter λ. The presence of such a parameter always raises the question, "What is the optimal value of λ"? In order to avoid this question, we have also considered distance functions composed of the Weberized metric $\Delta_a(u, v)$ and the correlation term $[1 - S_3(u, v)]$ which have no such variable parameters.

One possibility is to consider a product of the two terms, as is done in the SSIM. To simplify the algebra, we explored the squares of the components so that the objective function to be minimized assumed the form,

$$H_a(u, v) = [\Delta_a(u, v)]^2[1 - S_3(u, v)]^2. \tag{5.30}$$

We also explored the following objective function which is linear in the second term,

$$J_a(u, v) = [\Delta_a(u, v)]^2[1 - S_3(u, v)]. \tag{5.31}$$

In both cases, we performed a similar investigation as presented for $G_{a,\lambda}(u, v)$. The partial derivatives of Eq. (5.30) and Eq. (5.31) can be easily computed from the previous sections. Using Maple, we found that best approximations to the simple step function $u(x)$ in Eq. (5.25) which minimize either $H_a(u, v_M)$ or $J_a(u, v_M)$ are very similar to those obtained for $G_{a,\lambda}(u, v_M)$. There appears to be little qualitative difference between the best approximations obtained using any of the three objective functions, $H_a(u, v)$, $J_a(u, v)$, and $G_a(u, v)$. As such, we have omitted a detailed discussion of these investigations and results.

Chapter 6

Best Approximation Methods for Signals Which Involve Their Gradients: Part 1

6.1 An Introduction to the Application of Gradients in Mathematical Imaging

So far, our efforts have been solely directed towards exploring the use of correlation in applications relating to signal fidelity and image quality assessment. This chapter introduces a second topic of interest which will maintain our attention for the remainder of this thesis. In other words, we now pursue the second of two main guiding themes—that being, of course, the use of gradients in image quality assessment. First, in the following pages, we explore a simple best approximation method for signals involving their gradients. Later, in Chapter 7, we explore a problem involving both of our two main topics, by maximizing the correlation between gradient vectors.

For functions of continuous real variables, the gradient is defined in the usual way. (Recall that, for a greyscale image $f \in D \subset \mathbb{R}^2$, the gradient of f at a point $(x_0, y_0) \in D$ is defined as the vector of partial derivatives $\nabla f(x_0, y_0) = (\frac{\partial f}{\partial x}(x_0, y_0), \frac{\partial f}{\partial y}(x_0, y_0)))$. The vector of values $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$ evaluated along the entire domain of definition $D \subset \mathbb{R}^2$ represents the *gradient image*. Clearly, the *gradient image* is a higher dimensional image than we can visualize. We can, however, produce a conventional 2D image depicting the change along the x-direction by displaying the set of real values $\frac{\partial f}{\partial x}$ evaluated on D. Similarly,

we can produce another image depicting the change along the y-direction by displaying $\frac{\partial f}{\partial y}$ evaluated on D.

When f is instead a greyscale digital image, the intensity values are only known at discrete points. In this case, one must define a discrete derivative in both of the x- and y-directions to obtain a discrete gradient image. (Of course, if we assume that f is obtained by discrete sampling of a continuous image function, then one hopes that the discrete derivative accurately approximates the actual gradient values at each sample point.) In our applications, we will use a simple forward difference scheme to define the discrete derivative (this will be defined in detail in Section 6.2.1 below). Difference equations, like our simple forward differences, can easily be implemented as sliding window filters. As such, the discrete approximation to the gradient is commonly obtained from the original digital image by convolution with an appropriate filter, such as a Sobel or Scharr filter [6].

In general, the gradient indicates changes in intensity across adjacent pixels. This information is very useful in many applications in image processing. One of the most common uses of gradients in imaging is edge detection. In a typical scene, an edge separates objects which likely have very different intensities. In other words, we expect the gradient to be large at edges. Edge detection is useful in many applications, including image sharpening, computer vision, and medical imaging [6, 1]. Another popular use of the gradient is total variation (TV) regularization, which computes a best approximation seeking to minimize a balance of terms, one of which involves the gradient [13]. In essence, the gradient term to be minimized measures total variability across the entire signal.

More recently, gradient information has also been incorporated into image quality measures. It is known that the human visual system does not weigh all visual information equally; In particular, edges communicate very important information in a visual scene. Distortions affecting boundaries and edges are more bothersome to the human visual system than distortions in textural regions [19]. As such, these new measures are based on the notion that edge similarity, i.e., gradient similarity, is of particular importance when assessing the visual closeness of images. In [17], an SSIM-like "gradient similarity" measure is proposed, which seeks to quantitatively rate the presence of edges in an image patch using gradient operators. Haar wavelet filters are used to detect edges in [21]; then, a few equations based on the so-called Feature Similarity Index [33] are required to quantify the similarity of these edge maps.

Our work presented so far in this thesis has also incorporated gradient information. Recall our exploration in Section 5.1 of the SSIM-based best approximation problem using orthonormal functions. We found that the unknown coefficients c_k which maximize the SSIM are related to their Fourier counterparts a_k by a simple scaling, i.e., $c_k = \alpha a_k$ where

$\alpha \geq 1$. We previously observed that the SSIM-based best approximation is characterized by the *contrast-enhancement* $c_k = \alpha a_k$; In light of this section, this contrast-enhancement can also be viewed as an amplification of the gradient.

While the result $c_k = \alpha a_k$ does involve gradient information, as do other methods like TV regularization, the problem of *matching* gradients has been far less studied. In the remainder of this thesis, our focus will be to match the gradients of two related signals using various mathematical formulations. Our work which follows is also quite different from the "gradient similarity" measures proposed in both [17] and [21]. Both [17] and [21] feed gradient-based information into rather complicated mathematical machinery. As a result, it is difficult to understand exactly how the edge data informs the resulting similarity index. More to the point, it is unclear to what degree these complicated methods of collecting and aggregating gradient information reflect an honest matching of the gradient vectors. On the other hand, our approach, based on first principles, is mathematically tractable and clearly seeks to *match* the gradients of two signals according to a given similarity/distance measure.

Behind our approach is the question of how sensitive the human visual system could be with respect to changes in the gradient vectors of an image, in terms of either direction or magnitude, or perhaps both. Here we recall the simple model of blurring of an image by means of convolution with respect to an appropriate operator, e.g., a Gaussian filter. The blurring clearly modifies the gradient of the image by dampening it. The question is whether an image quality measure employing gradients can characterize the degradation of such blurring as well as, or even better, than current image quality measures, such as the Structural Similarity Index, which do not use gradient information.

6.2 Squared L^2 Distance Between Gradients as a Regularization Term in the L^2-based Best Approximation Problem

In the following sections, we present our first approach to matching gradients between signals. Below, we present an L^2-based best approximation problem which, similar to TV regularization, features a regularization term involving the gradient. We hope that the introduction of our gradient term will enforce a higher rate of convergence at the edges of u, where errors are particularly bothersome to the human visual system. The work in this section may thus be viewed as our second attempt to adapt the MSE for image

processing applications, following from our first attempt in Chapter 3 which involved the intensity-dependent Weberized distance.

We will once more be working in the Hilbert space $H = L^2[0,1]$. As usual, let $\{\phi_k(x)\}_{k=1}^{\infty}$ denote an orthonormal basis of H. We shall assume, as is often the case, that the orthonormal basis functions $\phi_k(x)$ are at least C^1 functions on $[0,1]$. (The standard trigonometric basis is, of course, a special case.)

Given a target function $u \in L^2[0,1]$, we know that there is a unique representation of the form

$$u = \sum_{k=1}^{\infty} a_k \phi_k,$$

where

$$a_k = \langle u, \phi_k \rangle, \quad k \geq 1,$$

are the Fourier coefficients of u.

We will be looking for an M-dimensional best approximation v_M to u, where

$$v_M(x) = \sum_{k=1}^{M} c_k \phi_k(x)$$

for some unknown coefficients c_k. From our assumptions on the basis functions ϕ_k, it follows that $v_M \in C^1[0,1]$ and

$$v'_M(x) = \sum_{k=1}^{M} c_k \phi'_k(x)$$

in the classical sense, i.e., each function $v'_M(x)$ is the classical derivative of $v_M(x)$.

For a given $M \geq 1$ and a fixed $\lambda \in [0,1]$, we consider the following approximation problem. Find coefficients $(c_1, \ldots, c_M)$ which minimize the following squared distance function involving both v_M and v'_M,

$$\Delta_M^2(\lambda) = \left\| u - \sum_{k=1}^{M} c_k \phi_k \right\|_2^2 + \lambda \left\| Du - \sum_{k=1}^{M} c_k \phi'_k \right\|_2^2. \tag{6.1}$$

When $\lambda = 0$, we have the standard L^2 best-approximation problem. By "Du", we mean a function which, in some way, represents a "derivative" of u. For example, if $u \in C^1$, then Du could be $u'(x)$. If u is a discrete (i.e., pixellated) approximation of a signal or image,

then Du could be a "discrete derivative" obtained by some kind of difference scheme. The discrete setting will be explored in Section 6.2.1.

We express the squared L^2 norms in terms of inner products on H,

$$\Delta_M^2(\lambda) = \left\langle u - \sum_{k=1}^{M} c_k \phi_k, u - \sum_{l=1}^{M} c_l \phi_l \right\rangle + \lambda \left\langle Du - \sum_{k=1}^{M} c_k \phi_k', Du - \sum_{l=1}^{M} c_l \phi_l' \right\rangle.$$

We now expand the inner products (which is permissible since the sums are finite). Because the inner product is symmetric for real-valued functions, we are able to combine the cross terms,

$$\begin{aligned}\Delta_M^2(\lambda) = \langle u, u\rangle - 2\sum_{k=1}^{M} c_k \langle u, \phi_k\rangle + \sum_{k=1}^{M}\sum_{l=1}^{M} c_k c_l \langle \phi_k \phi_l\rangle \\ + \lambda \left[\langle Du, Du\rangle - 2\sum_{k=1}^{M} c_k \langle Du, \phi_k'\rangle + \sum_{k=1}^{M}\sum_{l=1}^{M} c_k c_l \langle \phi_k', \phi_l'\rangle \right].\end{aligned}$$

Taking into account the orthonormality of the basis functions ϕ_k,

$$\begin{aligned}\Delta_M^2(\lambda) = \|u\|_2^2 - 2\sum_{k=1}^{M} c_k \langle u, \phi_k\rangle + \sum_{k=1}^{M} c_k^2 \\ + \lambda \left[\|Du\|_2^2 - 2\sum_{k=1}^{M} c_k \langle Du, \phi_k'\rangle + \sum_{k=1}^{N}\sum_{l=1}^{M} c_k c_l \langle \phi_k', \phi_l'\rangle \right].\end{aligned}$$

Now for each integer $p \in \{1, \ldots, M\}$, we take the partial derivative,

$$\frac{\partial \Delta_M^2}{\partial c_p} = -2\langle u, \phi_p\rangle + 2c_p - 2\lambda \langle Du, \phi_p'\rangle + 2\lambda \sum_{k=1}^{M} c_k \langle \phi_p', \phi_l'\rangle. \tag{6.2}$$

The stationarity conditions

$$\frac{\partial \Delta_M^2}{\partial c_p} = 0$$

yield a set of M simultaneous linear equations in the unknowns c_p, $1 \le p \le M$. In the special case $\lambda = 0$, Eq. (6.2) yields the standard result, as expected,

$$c_p = \langle u, \phi_p\rangle, \quad 1 \le p \le M.$$

The system in Eq. (6.2) can be rewritten in the following form,

$$c_p + \lambda \sum_{k=1}^{M} A_{pk} c_k = \langle u, \phi_p \rangle + \lambda \langle Du, \phi_p' \rangle, \quad 1 \leq p \leq M, \tag{6.3}$$

where

$$A_{ij} = \langle \phi_i', \phi_j' \rangle, \quad 1 \leq i, j \leq M. \tag{6.4}$$

The symmetry of the inner product implies that $A_{ij} = A_{ji}$, i.e., $\mathbf{A}$ is a symmetric matrix. Eq. (6.3) can be written as the matrix-vector system

$$[\mathbf{I} + \lambda \mathbf{A}]\mathbf{c} = \mathbf{b} + \lambda \mathbf{d},$$

where $\mathbf{A}$ is the $M \times M$ matrix defined in Eq. (6.4) above, $\mathbf{I}$ is the $M \times M$ identity matrix, and $\mathbf{c}$, $\mathbf{b}$, and $\mathbf{d}$ are M-vectors.

Note: Here we mention that our work is motivated by practical applications, i.e, relatively low values of M, and not with the theoretical question of what happens in the limit $M \to \infty$. The latter, certainly a most interesting question, lies in the realm of functional analysis (possibly involving Sobolev spaces) and is beyond the scope of this thesis.

A Special Case of Gradient-based Best-approximation with Examples

In the following example, we will once again be considering the set of functions,

$$\begin{aligned} \phi_1(x) &= 1 \\ \phi_k(x) &= \sqrt{2}\cos((k-1)\pi x), \quad k \geq 2, \end{aligned} \tag{6.5}$$

which forms an orthonormal basis in the Hilbert space $H = L^2[0,1]$.

Given the well-known orthogonality property of the sine and cosine functions on $[-\pi, \pi]$, one might expect that the scaled derivative functions,

$$\begin{aligned} \phi_1'(x) &= 0 \\ \phi_k'(x) &= -\sqrt{2}\pi(k-1)\sin((k-1)\pi x), \quad k \geq 2, \end{aligned} \tag{6.6}$$

also satisfy a set of orthogonality conditions. Indeed, we will demonstrate their orthogonality below.

First consider the case $k = l$. For $k \geq 2$, we have

$$\begin{aligned}\langle \phi_k', \phi_k' \rangle &= \int_0^1 2\pi^2 (k-1)^2 \sin^2((k-1)\pi x) dx \\ &= 2\pi^2 (k-1)^2 \int_0^1 \left[\frac{1}{2} - \frac{1}{2}\cos(2(k-1)\pi x) \right] dx \\ &= \pi^2 (k-1)^2.\end{aligned}$$

Also notice that when $k = 1$, $\langle \phi_1, \phi_1 \rangle = 0 = \pi^2(1-1)^2$. Hence $\langle \phi_k, \phi_k \rangle = \pi^2(k-1)$ for $k \geq 1$.

Now consider the case $k \neq l$. Clearly, if either $k = 1$ or $l = 1$, then $\langle \phi_k, \phi_l \rangle = 0$. When $k, l \geq 2$, we have

$$\begin{aligned}\langle \phi_k', \phi_l' \rangle &= 2\pi^2 (k-1)(l-1) \int_0^1 \sin((k-1)\pi x) \sin((l-1)\pi x) dx \\ &= \frac{2}{\pi^2 (k-1)(l-1)} \int_0^1 \sin((k-1)\pi x) \sin((l-1)\pi x) dx,\end{aligned}$$

where the final line is obtained after integrating by parts twice. Rearranging shows

$$\left(\pi^4 (k-1)^2 (l-1)^2 - 1 \right) \int_0^1 \sin((k-1)\pi x) \sin((l-1)\pi x) dx = 0.$$

For any $k, l \in \mathbb{Z}$, $(\pi^4 (k-1)^2 (l-1)^2 - 1) \neq 0$. Hence,

$$\int_0^1 \sin((k-1)\pi x) \sin((l-1)\pi x) dx = 0,$$

which implies that $\langle \phi_k', \phi_l' \rangle = 0$ for $k \neq l$.

The orthogonality of the derivative functions ϕ_k' implies that the matrix $\mathbf{A}$ with elements A_{ij} defined in Eq. (6.4) is diagonal, with

$$A_{kk} = \pi^2 (k-1)^2, \quad k \geq 1. \tag{6.7}$$

As such, the system in Eq. (6.3) becomes

$$c_p(\lambda) = \frac{1}{1 + \lambda (p-1)^2 \pi^2} [\langle u, \phi_p \rangle + \lambda \langle Du, \phi_p' \rangle], \quad 1 \leq p \leq M. \tag{6.8}$$

Note the rather special nature of the first coefficient, c_1: Given that $\phi_1' = 0$,

$$\begin{aligned} c_1 &= \langle u, \phi_1 \rangle \\ &= a_1. \end{aligned}$$

In other words, c_1 is always the first Fourier coefficient of u in the ϕ_k basis, and independent of λ. Furthermore, since $\phi_1(x) = 1$,

$$c_1 = \int_0^1 u(x)dx, \tag{6.9}$$

the mean value of u on $[0, 1]$.

We will now consider a simple example to explore the effects of the regularization term in the objective function Δ_M^2 in Eq. (6.1).

Example 1: Consider $u(x) = x^2$ on $[0, 1]$. It is easy to obtain the first coefficient of the best approximation,

$$c_1 = \int_0^1 x^2 dx = \frac{1}{3}.$$

To obtain the coefficients $c_p(\lambda)$ for $p \geq 2$, we need to compute the two inner products in Eq. (6.8). For the first inner product, we integrate by parts twice,

$$\begin{aligned} \langle u, \phi_p \rangle &= \int_0^1 x^2 \sqrt{2} \cos((p-1)\pi x) dx \\ &= x^2 \frac{\sqrt{2}\sin((p-1)\pi x)}{(p-1)\pi}\bigg|_0^1 - \int_0^1 2x \frac{\sqrt{2}\sin((p-1)\pi x)}{(p-1)\pi} dx \\ &= 2x \frac{\sqrt{2}\cos((p-1)\pi x)}{(p-1)^2\pi^2}\bigg|_0^1 - \int_0^1 2 \frac{\sqrt{2}\cos((p-1)\pi x)}{(p-1)^2\pi^2} dx \\ &= \frac{(-1)^{p-1} 2\sqrt{2}}{(p-1)^2\pi^2}. \end{aligned} \tag{6.10}$$

For the second inner product, we will first use the fact that $u(x) = x^2 \in C^2$ to write

$$\begin{aligned}\langle Du, \phi_p' \rangle &= \langle u', \phi_p \rangle \\ &= \int_0^1 u'(x)\phi_p'(x)dx \\ &= u'(x)\phi_p(x)\big|_0^1 - \int_0^1 u''(x)\phi_p(x)dx. \end{aligned} \tag{6.11}$$

Now substituting $u(x) = x^2$, this becomes

$$\begin{aligned}\langle Du, \phi_p' \rangle &= 2x\sqrt{2}\cos((p-1)\pi x)\Big|_0^1 - \int_0^1 2\sqrt{2}\cos((p-1)\pi x)dx \\ &= (-1)^{p-1}2\sqrt{2}.\end{aligned}$$

Substitution of these two results into Eq. (6.8) yields

$$c_p(\lambda) = \frac{(-1)^{p-1}2\sqrt{2}}{1+\lambda(p-1)^2\pi^2}\left[\frac{1}{(p-1)^2\pi^2} + \lambda\right], \quad p \geq 2.$$

Although the solution appears complete, the above result becomes much more informative if one thinks to perform the following simple rearrangement,

$$\begin{aligned}c_p(\lambda) &= \frac{(-1)^{p-1}2\sqrt{2}}{1+\lambda(p-1)^2\pi^2}\left[\frac{1+\lambda(p-1)^2\pi^2}{(p-1)^2\pi^2}\right] \\ &= \frac{(-1)^{p-1}2\sqrt{2}}{(p-1)^2\pi^2} \\ &= c_p(0).\end{aligned}$$

In other words, for any $\lambda \geq 0$, the gradient-based best approximation is given by the standard best-L^2 approximation obtained for $\lambda = 0$. This result was verified numerically.

At this point, the above result has been obtained only for the particular choice $u(x) = x^2$. One may wonder if the result holds in general, perhaps provided that the target function u is sufficiently "nice", i.e., sufficiently smooth. Indeed, we can obtain this most interesting result if the integration by parts performed in Eq. (6.11) is reversed. Let us

consider any C^1 function u so that $Du = u'$. Then,

$$\begin{aligned}
\langle Du, \phi_p' \rangle &= \langle u', \phi_p' \rangle \\
&= \int_0^1 u'(x)\phi_p'(x)dx \\
&= \phi_p'(x)u(x)\big|_0^1 - \int_0^1 u(x)\phi_p''(x)dx \\
&= (p-1)^2\pi^2 \int_0^1 u(x)\phi_p(x)dx \qquad (\text{since } \phi_p'(0) = \phi_p'(1) = 0) \\
&= (p-1)^2\pi^2 \langle u, \phi_p \rangle.
\end{aligned}$$

Substitution of this result into Eq. (6.8) yields

$$\begin{aligned}
c_p(\lambda) &= c_p(0) \\
&= a_p.
\end{aligned} \tag{6.12}$$

Although the above result is very interesting, it is perhaps not too surprising. Indeed, when u and v_M are close in terms of the L^2 distance, their derivatives u' and v_M' should understandably also be close in terms of the L^2 distance. In the following section, we will explore whether the result in Eq. (6.12) also holds in the discrete case, i.e., for digital signals/images.

6.2.1 Discrete Formulation of the Gradient-based Best Approximation Problem

In this section, we will be considering digital signals belonging to the Hilbert space $\mathbb{R}^N$, where $N > 1$. Let $\{\phi_k\}_{k=1}^N$ denote an orthonormal basis of $\mathbb{R}^N$. A signal $u \in \mathbb{R}^N$ admits an expansion of the form

$$u = \sum_{k=1}^{N} a_k \phi_k \tag{6.13}$$

where

$$a_k = \langle u, \phi_k \rangle, \quad 1 \leq k \leq N,$$

are the Fourier coefficients of u.

We will once again be looking for M-dimensional approximations v_M to u having the form,

$$v_M = \sum_{k=1}^{M} c_k \phi_k, \quad 1 \leq M < N.$$

We now consider the discrete version of the approximation problem associated with Eq. (6.1) in the previous section. Namely, we are searching for coefficients $(c_1, \ldots, c_M)$ which minimize the following squared distance function,

$$\Delta_M^2(\lambda) = \left\| u - \sum_{k=1}^{M} c_k \phi_k \right\|_2^2 + \lambda \left\| Du - \sum_{k=1}^{M} c_k D\phi_k \right\|_2^2, \tag{6.14}$$

for a fixed $\lambda \in [0, 1]$. By "Du" and "$D\phi_k$", we mean functions which again represent a derivative of v and ϕ_k, respectively, but in the discrete domain. The discrete derivative employed throughout this thesis will be a simple forward difference scheme, i.e.,

$$u = (u_1, \ldots, u_N) \implies Du = (u_2 - u_1, u_3 - u_2, \ldots, u_N - u_{N-1}, u_{N+1} - u_N).$$

The term u_{N+1} will be determined by the orthonormal basis that we employ which, in turn, indicates the periodic extension of the data which is assumed. For example,

1. If we use the DCT basis functions, then an even extension of N data points is assumed, implying that $u_{N+1} = u_N$.
2. If we use the DFT basis functions, then a periodic extension of the N data points is assumed, implying that $u_{N+1} = u_1$.

After expanding the inner products as done for the continuous case, we arrive at the following expression for our discrete distance function,

$$\Delta_M^2 = \|u\|_2^2 - 2\sum_{k=1}^{M} c_k \langle u, \phi_k \rangle + \sum_{k=1}^{M} c_k^2$$
$$\lambda + \left[\|Du\|_2^2 - 2\sum_{k=1}^{M} c_k \langle Du, D\phi_k \rangle + \sum_{k=1}^{M} \sum_{l=1}^{M} c_k c_l D\phi_k D\phi_l \right].$$

Imposing the stationarity conditions,

$$\frac{\partial \Delta_M^2}{\partial c_p} = 0, \quad 1 \leq p \leq M,$$

produces a set of linear equations in the unknown coefficients c_p having the form,

$$c_p + \lambda \sum_{k=1}^{M} A_{pk} c_k = \langle u, \phi_p \rangle + \lambda \langle Du, D\phi_p \rangle, \quad 1 \leq p \leq M, \tag{6.15}$$

where

$$A_{ij} = \langle D\phi_i, D\phi_j \rangle, \quad 1 \leq i, j \leq M. \tag{6.16}$$

The system of equations in Eq. (6.15) can be written in the following matrix-vector form,

$$[\mathbf{I} + \lambda \mathbf{A}]\mathbf{c} = \mathbf{b} + \lambda \mathbf{d},$$

where $\mathbf{A}$ is the $M \times M$ matrix defined in Eq. (6.16) above, $\mathbf{I}$ is the $M \times M$ identity matrix, and $\mathbf{c}$, $\mathbf{b}$, and $\mathbf{d}$ are M-vectors.

From Eq. (6.13), it follows that

$$Du = \sum_{k=1}^{N} a_k D\phi_k.$$

Substitution of this result into Eq. (6.15) yields the following,

$$\begin{aligned} c_p + \lambda \sum_{k=1}^{M} A_{pk} c_k &= \langle u, \phi_p \rangle + \lambda \sum_{k=1}^{N} a_k \langle D\phi_k, D\phi_p \rangle \\ &= a_p + \lambda \sum_{k=1}^{M} a_k A_{kp} + \lambda \sum_{k=M+1}^{N} a_k A_{kp}, \quad 1 \leq p \leq M. \end{aligned} \tag{6.17}$$

Special Cases of the Discrete Gradient-based Best Approximation Problem

We now briefly explore the solution to the discrete problem under simplifying assumptions on the matrix $\mathbf{A}$ defined in Eq. (6.16). Provided either of the special cases listed below holds, the gradient-based best approximation is once again determined by the Fourier coefficients of the reference signal.

1. When $\mathbf{A}$ is a diagonal matrix—which we encountered in the continuous case when an orthonormal cosine basis was used—the final summation in Eq. (6.17) vanishes and the system of equations becomes,

$$c_p - \lambda A_{pp} c_p = a_p + \lambda A_{pp} a_p.$$

Factoring both sides,

$$(1 + \lambda A_{pp})c_p = (1 + A_{pp})a_p, \quad 1 \leq p \leq M.$$

Because this equation must hold for all $\lambda \geq 0$, and

$$A_{pp} = \langle D\phi_p, D\phi_p \rangle = \|D\phi_p\|_2^2 \geq 0,$$

it follows that

$$(1 + \lambda A_{pp}) \geq 0$$

and hence

$$c_p(\lambda) = a_p = c_p(0), \quad 1 \leq p \leq M.$$

2. Suppose that for all $1 \leq p \leq M$,

$$A_{kp} = 0, \quad M + 1 \leq k \leq N.$$

Then the final summation in Eq. (6.17) once again vanishes so that the system of equations becomes,

$$c_p + \lambda \sum_{k=1}^{M} A_{pk}c_k = a_p + \lambda \sum_{k=1}^{M} a_k A_{kp}, \quad 1 \leq p \leq M.$$

Using the fact that $\mathbf{A}$ is a symmetric matrix, let us rewrite the above equation as follows,

$$\lambda \sum_{k=1}^{M} A_{pk}(c_k - a_k) = a(c_p - a_p), \quad 1 \leq p \leq M.$$

This equation has the form

$$\mathbf{Ad} = \mu \mathbf{d},$$

where

$$\mathbf{d} = \mathbf{c} - \mathbf{a} \text{ and } \mu = -\frac{1}{\lambda}.$$

This looks like an eigenvector-eigenvalue problem. The problem is that λ, hence μ, must assume a continuous set of values. This suggests that $\mathbf{d} = \mathbf{0}$ which implies that $\mathbf{c} = \mathbf{a}$, i.e.,

$$c_p(\lambda) = a_p = c_p(0), \quad 1 \leq p \leq M.$$

6.3 Orthogonality of the Discrete Derivatives of the DCT an DFT Basis Functions

We would now like to draw the reader's attention to the following two results. To the best of our knowledge, they have not appeared in the literature. We were naturally guided, during the completion of our work in the previous section, to uncover the following remarkable property: The discrete derivatives of the DCT and DFT basis functions each form, respectively, an orthogonal set. These results are formally stated below. Note that, throughout this section, we employ the standard index notation for the DCT/DFT basis functions (i.e., a signal $x \in \mathbb{R}^N$ is denoted $x = (x_0, \ldots, x_{N-1})$), which differs from the notation used previously.

Theorem 4. Let $\{\phi_0, \cdots, \phi_{N-1}\}$ denote the following DCT basis functions,

$$\begin{aligned} \phi_0[n] &= \frac{1}{\sqrt{N}}, \\ \phi_k[n] &= \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right), \quad 1 \leq k \leq N-1, \quad 0 \leq n \leq N-1. \end{aligned} \tag{6.18}$$

Let the discrete derivative of these functions be defined by simple forward differences so that

$$D\phi_k[n] = \phi_k[n+1] - \phi_k[n], \quad 0 \leq n \leq N-1. \tag{6.19}$$

Then for a given $N > 0$, $\{D\phi_0, \cdots, D\phi_{N-1}\}$ forms an orthogonal set in $\mathbb{R}^N$.

Theorem 5. Let $\{\phi_0, \cdots, \phi_{N-1}\}$ denote the following DFT basis functions,

$$\phi_k[n] = \frac{1}{\sqrt{N}} \exp\left(\frac{i2\pi kn}{N}\right), \quad 0 \leq n \leq N-1. \tag{6.20}$$

Let the discrete derivative of these functions be defined by simple forward differences so that

$$D\phi_k[n] = \phi_k[n+1] - \phi_k[n], \quad 0 \leq n \leq N-1. \tag{6.21}$$

Then for a given $N > 0$, $\{D\phi_0, \cdots, D\phi_{N-1}\}$ forms an orthogonal set in $\mathbb{C}^N$.

In Appendix A can be found a proof of both Theorem 4 and Theorem 5. These two results have also been verified numerically. To the best of our knowledge, at the time of writing this thesis, these properties have not previously been published.

These results provide a very tidy and satisfying completion to our discussion in the preceding section. With reference to the discrete problem discussed above, the DCT and

DFT basis functions both yield a diagonal matrix $\mathbf{A}$, where $A_{ij} = \langle D\phi_k, D\phi_j \rangle$. In particular, employing the DCT basis functions in the discrete problem behaves in an analogous manner to using the orthonormal cosine basis in the continuous case, as one might hope. The results stated in Theorem 4 and Theorem 5 will also be required for our work in the next sections. Before moving on, however, we would like to make a few more important comments about the DCT derivative functions.

The DCT functions ϕ_k for $N = 8$ are plotted in Figure 6.3. Their derivative functions $D\phi_k$ for $N = 8$ are plotted in Figure 6.3. Looking at the plots, it is apparent that the DCT derivative functions $D\phi_k[n]$ are either symmetric or antisymmetric about the point $n = 3$. By comparison, the DCT functions $\phi_k[n]$ are either symmetric or antisymmetric about the point $n = 3.5$.

Recall that both the DCT basis functions ϕ_k and their derivative functions $D\phi_k$ are not N-periodic, but $2N$-periodic. If we were to extend the n-values of the $\phi_k[n]$ for $n > N-1$, the resulting plot would be even-symmetric with respect to the point $n = N - \frac{1}{2}$. In other words, we have

$$\begin{aligned} \phi_k[N] &= \phi_k[N-1], \\ \phi_k[N+1] &= \phi_k[N-2], \end{aligned}$$

and, in general,

$$\phi_k[N+p] = \phi_k[N-p-1], \qquad \text{for } 0 \leq p \leq N-1.$$

It can be seen in Figure 6.3 that, for all k, $D\phi_k[7] = 0$. In general, for all k, $D\phi_k[N-1] = 0$ since $\phi_k[N-1] = \phi_k[N]$ from the even extension assumed in the DCT case. Moreover, thinking back to the continuous case, one might expect the derivative functions $D\phi_k$ to be sine functions. While the $D\phi_k[n]$ are "sine-like", they are not pure sine functions. In fact, the derivative functions $D\phi_k$ contain both a sine and cosine component; This property is fully demonstrated by the expansion $D\phi_k = C_k\phi_k - D_k\psi_k$ derived in the proof in Appendix A.2.

Finally, note that the DCT derivative functions $D\phi_k$ form an orthogonal set for $1 \leq k \leq 7$. Since the dimensionality of this space is $N = 8$, this set does not form a basis—we are essentially one vector short because $k = 0$ corresponds to the zero vector. For the same reason, the DFT derivative functions do not form a basis either. As demonstrated in the proof in Appendix A.1, $k = 0$ once again corresponds to the zero vector.

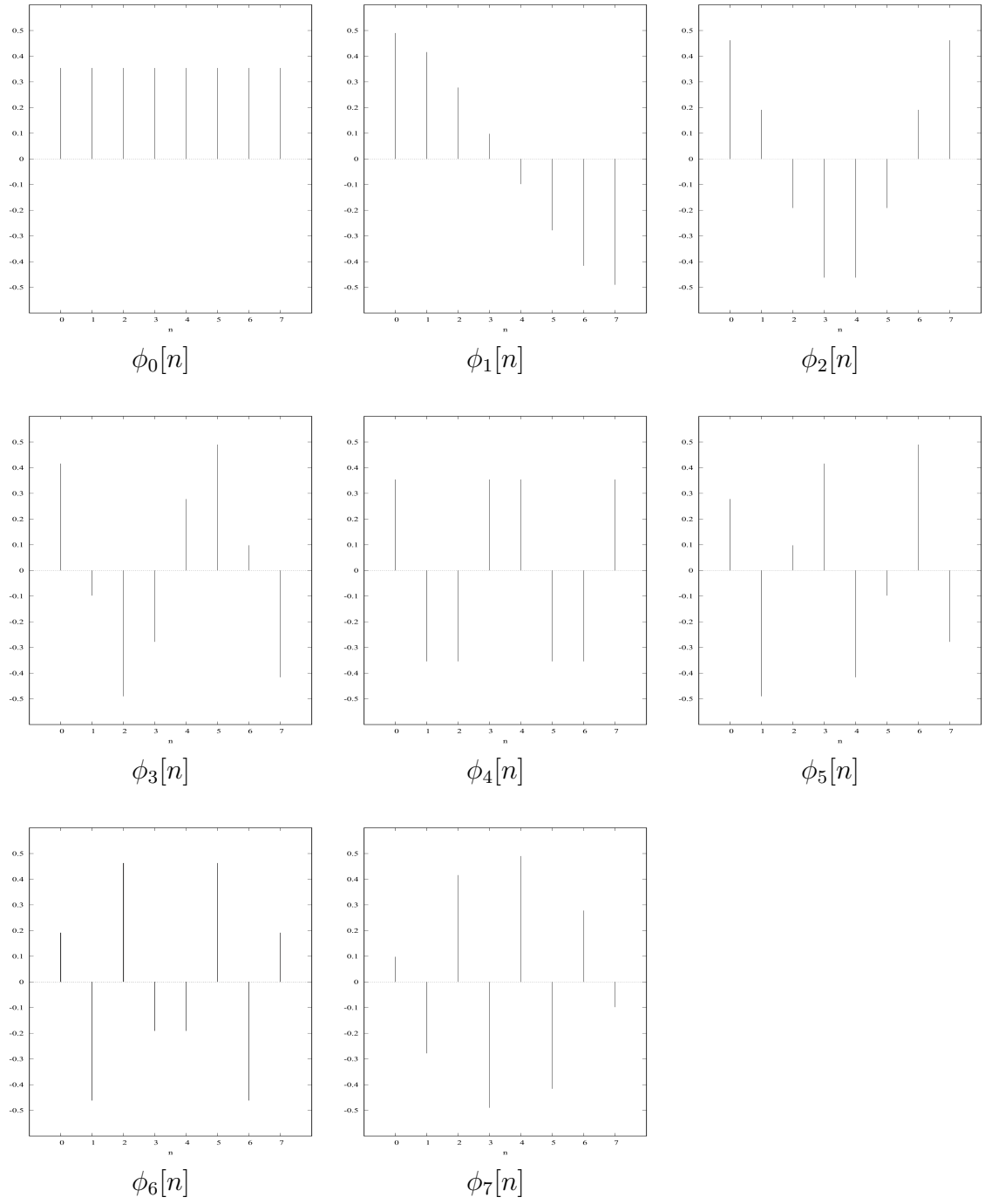
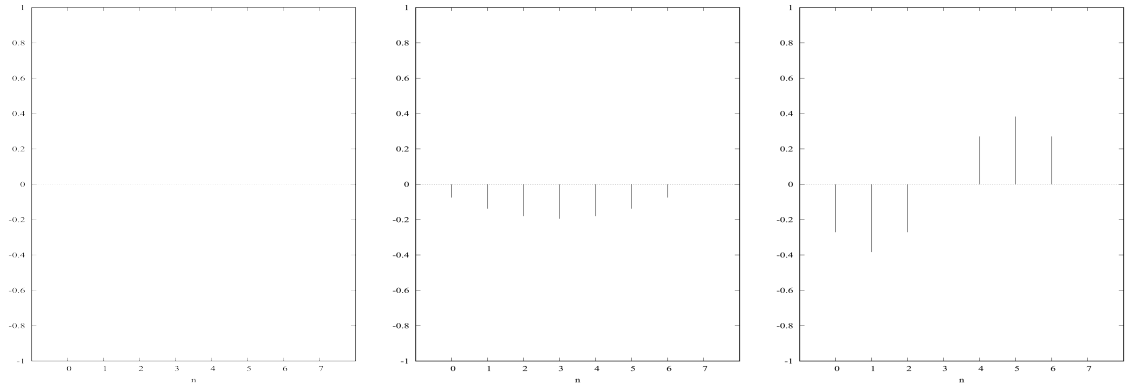
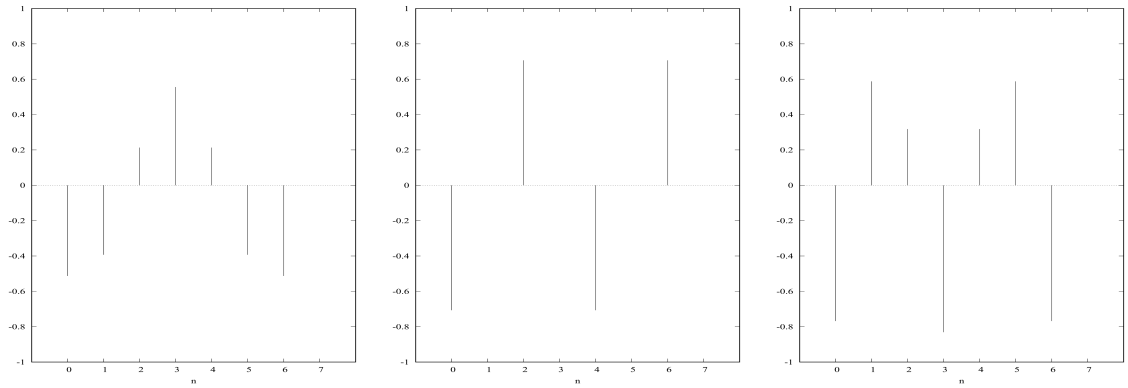


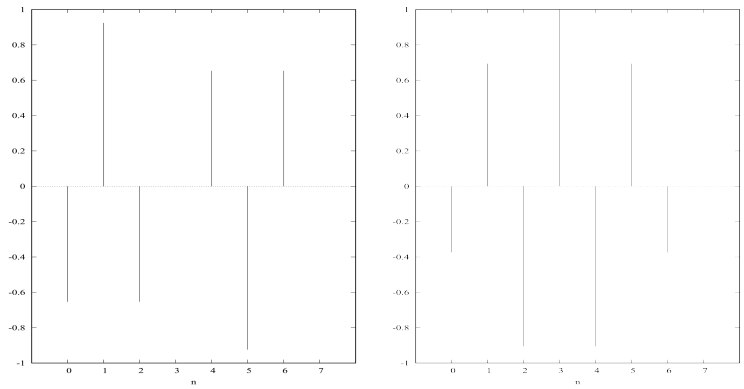
Figure 6.1: The N=8-point DCT orthonormal function $\phi_k[n]$, $k = 0, 1, \ldots, 7$.



$D\phi_0[n]$ $D\phi_1[n]$ $D\phi_2[n]$



$D\phi_3[n]$ $D\phi_4[n]$ $D\phi_5[n]$



$D\phi_6[n]$ $D\phi_7[n]$

Figure 6.2: The N=8-point DCT derivative functions $D\phi_k[n] = \phi_k[n+1] - \phi_k[n]$, $k = 0, 1, \cdots, 7$. The functions $D\phi_k$, $k = 0, 1, \ldots, 7$ form an orthogonal set.

Chapter 7

Best Approximation Methods for Signals Which Involve Their Gradients: Part 2

7.1 Best Approximation by Maximizing the Correlation Between Gradient Vectors

Still motivated by our guiding belief that the correlation is the most important component of the SSIM, we now investigate a problem which combines both the correlation and the gradient. In our second approach to matching gradients, presented below, we seek to maximize the correlation between gradient vectors. Having previously maximized the SSIM between two signals in Chapter 5.1, we are now interested to see if maximizing the correlation between their gradients will yield different results. And, if so, we wonder if these gradient-based results will perhaps yield better approximations according to the human visual system.

Following from our previous formulation, we will once more be employing orthonormal basis functions; However, unlike our work in Chapter 5, we will consider the discrete case below. In the following work, we will make the natural assumption that the discrete orthonormal basis in use is either the DCT or DFT. Some simplifications will be permitted throughout our derivation due to known properties of these two sets. In particular, we will once again be applying Theorem 4 and Theorem 5 which were stated at the conclusion of the preceding chapter.

As is well-known to us by now, the correlation is the third component of the SSIM function. Below, we will once more be considering the special case where the stability constant $C_3 = 0$. Then, for $u, v \in \mathbb{R}^N$, the correlation is denoted by $S_3(u, v)$ and computed as follows:

$$S_3(u, v) = \frac{s_{uv}}{s_u s_v}, \tag{7.1}$$

where

$$s_{uv} = \frac{1}{N-1} \sum_{k=1}^{N} (u_k - \bar{u})(v_k - \bar{v}), \tag{7.2}$$

and

$$s_u = \sqrt{s_{uu}}, \tag{7.3}$$

with

$$\bar{u} = \frac{1}{N} \sum_{k=1}^{N} u_k. \tag{7.4}$$

Let $\{\phi_k\}_{k=1}^N$ denote an orthonormal basis in $\mathbb{R}^N$. For both $u, v \in \mathbb{R}^N$, we shall denote the components of their gradient functions as follows,

$$Du = (Du_1, \ldots, Du_N) \quad \text{and} \quad Dv = (Dv_1, \ldots, Dv_N).$$

We now wish to consider the following best approximation problem in $\mathbb{R}^N$ in terms of the correlation between two gradient vectors: For a given target function $u \in \mathbb{R}^N$, with Fourier expansion

$$u = \sum_{i=1}^{N} a_i \phi_i, \quad \text{where} \quad a_k = \langle u, \phi_k \rangle, \quad 1 \leq k \leq N,$$

and an $1 \leq M < N$, find the approximation of the form,

$$v_M = \sum_{i=1}^{M} c_i \phi_i,$$

which maximizes the correlation between the gradients of u and v, i.e.,

$$c = \arg\max_{d \in \mathbb{R}^M} S_3(Du, Dv_M).$$

Here,

$$Du = \sum_{i=1}^{N} a_i D\phi_i \quad \text{and} \quad Dv_M = \sum_{i=1}^{M} c_i D\phi_i.$$

We will go forward with the understanding that v denotes our best approximation v_M, i.e., we will omit the subscript, to avoid confusion when extracting its component parts $v = (v_1, \ldots, v_M)$. The mean values of the gradient vectors will be also important,

$$\overline{Du} = \frac{1}{N} \sum_{k=1}^{N} Du_k \quad \text{and} \quad \overline{Dv} = \frac{1}{N} \sum_{k=1}^{N} Dv_k.$$

Note that to when defining $\overline{Dv}$, we consider Dv to be an N-vector with the last $N - M$ of its elements being zero instead of an M-vector. (We have to consider both vectors as N-vectors in order to be able to compute their correlation.)

If the discrete derivative is defined by forward differences, then the above mean values are telescopic sums that reduce to simple differences, i.e.,

$$\overline{Du} = \frac{1}{N}(u_{N+1} - u_1) \quad \text{and} \quad \overline{Dv} = \frac{1}{N}(v_{N+1} - v_1).$$

If we further assume an N-periodic extension of the N-vectors—as is the case when using the DFT basis functions—then,

$$\overline{Du} = 0 \quad \text{and} \quad \overline{Dv} = 0.$$

We are now ready to compute the correlation between gradients, i.e.,

$$S_3(Du, Dv) = \frac{s_{DuDv}}{s_{Du} s_{Dv}}.$$

For simplicity, we will first consider the square of the term in the denominator $s_{DuDu} = s_{Du}^2$,

$$s_{DuDu} = \frac{1}{N-1} \sum_{k=1}^{N} (Du_k - \overline{Du})(Du_k - \overline{Du}).$$

Now expand and substitute the above expressions to get

$$\begin{aligned} s_{DuDu} &= \frac{1}{N-1}\left[\sum_{k=1}^{N}(Du_k)^2 - 2\overline{Du}\sum_{k=1}^{N}Du_k + (\overline{Du})^2\sum_{k=1}^{N}1\right] \\ &= \frac{1}{N-1}\left[\sum_{k=1}^{N}(Du_k)^2 - 2\overline{Du}(N\overline{Du}) + N(\overline{Du})^2\right] \\ &= \frac{1}{N-1}\left[\sum_{k=1}^{N}(Du_k)^2 - N(\overline{Du})^2\right]. \end{aligned} \tag{7.5}$$

Similarly, one can obtain

$$s_{DvDv} = \frac{1}{N-1}\left[\sum_{k=1}^{N}(Dv_k)^2 - N(\overline{Dv})^2\right]. \tag{7.6}$$

Finally, we also have

$$\begin{aligned} s_{DuDv} =& \frac{1}{N-1}\sum_{k=1}^{N}(Du_k - \overline{Du})(Dv_k - \overline{Dv}) \\ &= \frac{1}{N-1}\left[\sum_{k=1}^{N}(Du_k)(Dv_k) - \overline{Dv}\sum_{k=1}^{N}Du_k - \overline{Du}\sum_{k=1}^{N}Dv_k + (\overline{Du})(\overline{Dv})\sum_{k=1}^{N}1\right] \\ &= \frac{1}{N-1}\left[\sum_{k=1}^{N}(Du_k)(Dv_k) - \overline{Dv}(N\overline{Du}) - \overline{Du}(N\overline{Dv}) + N(\overline{Du})(\overline{Dv})\right] \\ &= \frac{1}{N-1}\left[\sum_{k=1}^{N}(Du_k)(Dv_k) - N(\overline{Du})(\overline{Dv})\right]. \end{aligned} \tag{7.7}$$

We now express these summations in terms of the expansion coefficients a_i and c_i. First

consider the mean values,

$$\begin{aligned}
\overline{Dv} &= \frac{1}{N}\sum_{k=1}^{N} Dv_k \\
&= \frac{1}{N}\sum_{k=1}^{N}\left(\sum_{i=1}^{M} c_i D\phi_i\right)_k \\
&= \sum_{i=1}^{M} c_i \left(\frac{1}{N}\sum_{k=1}^{N}(D\phi_i)_k\right) \\
&= \sum_{i=1}^{M} c_i \overline{D\phi_i}.
\end{aligned}$$

Recall that $\overline{D\phi_1} = 0$ for both the DCT and DFT basis functions. Hence, the summation will run from $i = 2$ to $i = M$, i.e.,

$$\overline{Dv} = \sum_{i=2}^{M} c_i \overline{D\phi_i}. \tag{7.8}$$

Similarly, we also find that

$$\overline{Du} = \sum_{i=2}^{N} a_i \overline{D\phi_i}. \tag{7.9}$$

We will now deal with the remaining summation in Eq. (7.6), rewritten below,

$$\begin{aligned}
\sum_{k=1}^{N}(Dv_k)^2 &= \sum_{k=1}^{N}\left(\sum_{i=1}^{M} c_i D\phi_i\right)_k \left(\sum_{j=1}^{M} c_j D\phi_j\right)_k \\
&= \sum_{i=1}^{M}\sum_{j=1}^{M} c_i c_j \sum_{k=1}^{N}(D\phi_i)_k (D\phi_j)_k \\
&= \sum_{i=1}^{M}\sum_{j=1}^{M} c_i c_j \langle D\phi_i, D\phi_j\rangle \\
&= \sum_{i=1}^{M}\sum_{j=1}^{M} c_i c_j A_{ij}.
\end{aligned}$$

Due to the orthogonality of the discrete derivatives, as stated in Theorem 4 and Theorem 5 in the previous chapter, the matrix $\mathbf{A}$ is diagonal for both the DCT and DFT basis functions. Thus,

$$\sum_{k=1}^{N}(Dv_k)^2 = \sum_{i=1}^{M} c_i^2 A_{ii},$$

where the values of the diagonal elements A_{ii} will depend on the choice of basis. For either the DCT or DFT basis functions, $A_{11} = 0$ so that the above summation runs from $i = 2$ to $i = M$. The resulting expressing for s_{DvDv} in Eq. (7.6) is

$$\begin{aligned} s_{DvDv} &= \frac{1}{N-1}\left[\sum_{k=1}^{N}(Dv_k)^2 - N(\overline{Dv})^2\right] \\ &= \frac{1}{N-1}\left[\sum_{i=2}^{M} c_i^2 A_{ii} - N\left(\sum_{i=2}^{M} c_i \overline{D\phi_i}\right)^2\right]. \end{aligned} \tag{7.10}$$

In a similar manner, we can find that the term s_{DuDu} in Eq. (7.5) becomes

$$s_{DuDu} = \frac{1}{N-1}\left[\sum_{i=2}^{N} a_i^2 A_{ii} - N\left(\sum_{i=2}^{N} a_i \overline{D\phi_i}\right)^2\right]. \tag{7.11}$$

Finally, we must consider the first summation in the expression for s_{DuDv} written in Eq. (7.7),

$$\begin{aligned} \sum_{k=1}^{N}(Du_k)(Dv_k) &= \sum_{k=1}^{N}\left(\sum_{i=1}^{N} a_i D\phi_i\right)_k \left(\sum_{j=1}^{M} c_j D\phi_j\right)_k \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} a_i c_j \langle D\phi_i, D\phi_j\rangle \\ &= \sum_{i=1}^{M} a_i c_i A_{ii}, \end{aligned}$$

where the simplification in the final line once again results from the orthogonality of the discrete derivatives of the DCT and DFT basis functions. Once again, $A_{11} = 0$ so that the

above summation runs from $i = 2$ to $i = M$. Thus,

$$\begin{aligned} s_{DuDv} &= \frac{1}{N-1}\left[\sum_{k=1}^{N}(Du_k)(Dv_k) - N(\overline{Du})(\overline{Dv})\right] \\ &= \frac{1}{N-1}\left[\sum_{i=2}^{M} a_i c_i A_{ii} - N\left(\sum_{i=2}^{N} a_i \overline{D\phi_i}\right)\left(\sum_{j=2}^{M} c_j \overline{D\phi_j}\right)\right]. \end{aligned} \tag{7.12}$$

The final result for the correlation $S_3(Du, Dv)$ expressed in terms of the unknown coefficients c_k is as follows,

$$\begin{aligned} S_3(Du, Dv) &= \frac{s_{DuDv}}{s_{Du}s_{Dv}} \\ &= \frac{\frac{1}{N-1}\left(\sum_{i=2}^{M} a_i c_i A_{ii} - N\left(\sum_{i=2}^{N} a_i \overline{D\phi_i}\right)\left(\sum_{j=2}^{M} c_j \overline{D\phi_j}\right)\right)}{s_{Du}\left[\frac{1}{N-1}\left(\sum_{i=2}^{M} c_i^2 A_{ii} - N\left(\sum_{i=2}^{M} c_i \overline{D\phi_i}\right)^2\right)\right]^{1/2}}. \end{aligned} \tag{7.13}$$

We are now prepared to find the unknown coefficients c_i which maximize the correlation $S_3(Du, Dv)$. Note, however, that $S_3(Du, Dv)$ does not depend on the first coefficient c_1. (This was also the case when maximizing the SSIM between two discrete N-vectors in [3].) As such, we'll have to impose additional conditions in order to determine c_1. For $2 \leq p \leq M$, we'll impose the stationarity conditions

$$\frac{\partial S_3(Du, Dv)}{\partial c_p} = 0 \tag{7.14}$$

in an effort to find the optimal coefficients c_i.

Of course, we are required to compute the partial derivatives of the components s_{DuDv} and s_{Dv} in Eq. (7.13). (The term s_{Du} contains only the a_k coefficients and therefore behaves as a constant.) Firstly, differentiating Eq. (7.12),

$$\begin{aligned} \frac{\partial s_{DuDv}}{\partial c_p} &= \frac{1}{N-1}\left[a_p A_{pp} - N\left(\sum_{i=2}^{N} a_i \overline{D\phi_i}\right)\overline{D\phi_p}\right] \\ &= \frac{1}{N-1}\left[a_p A_{pp} - N\overline{Du D\phi_p}\right], \end{aligned} \tag{7.15}$$

where the final line simply results from the resubstitution of $\overline{Du}$.

Now consider the expression for s_{DvDv} written in Eq. (7.10). First notice that $s_{Dv} = (s_{DvDv})^{1/2}$, so that

$$\frac{\partial s_{Dv}}{\partial c_p} = \frac{1}{2}(s_{DvDv})^{-1/2}\frac{\partial s_{DvDv}}{\partial c_p} = \frac{1}{2s_{Dv}}\frac{\partial s_{DvDv}}{\partial c_p}.$$

Hence, differentiating Eq. (7.10) yields,

$$\frac{\partial s_{DvDv}}{\partial c_p} = \frac{1}{N-1}\left[2c_pA_{pp} - 2N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i\overline{D\phi_i}\right)\right].$$

And therefore,

$$\frac{\partial s_{Dv}}{c_p} = \frac{1}{s_{Dv}(N-1)}\left[c_pA_p - N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i\overline{D\phi_i}\right)\right].$$

Let us now examine the structure of the partial derivative of $S_3(Du, Dv)$ with respect to the unknown coefficients c_p for $2 \leq p \leq M$. Letting the prime sign denote the partial derivative with respect to c_p, we have

$$\frac{\partial S_3(Du, Dv)}{\partial c_p} = \frac{1}{s_{Du}}\frac{s_{Dv}s'_{DuDv} - s_{DuDv}s'_{Dv}}{s_{Dv}^2}.$$

The stationarity condition in Eq. (7.14) implies that the numerator of the expression on the right be zero, i.e.,

$$s_{Dv}s'_{DuDv} = s_{DuDv}s'_{Dv}.$$

We now proceed as in Chapter 5.1 by rearranging the above equation as follows,

$$\frac{s'_{Dv}}{s'_{DuDv}} = \frac{s_{Dv}}{s_{DuDv}}.$$

Substitution of the appropriate expressions into the left hand side of the equation yields,

$$\frac{\frac{1}{s_{Dv}(N-1)}\left[c_pA_{pp} - N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i\overline{D\phi_i}\right)\right]}{\frac{1}{N-1}\left[a_pA_{pp} - N\overline{Du}, \overline{D\phi_p}\right]} = \frac{s_{Dv}}{s_{DuDv}}.$$

Multiply by s_{Dv} and cancel the repeated factor $\frac{1}{N-1}$ to get

$$\frac{c_pA_{pp} - N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i\overline{D\phi_i}\right)}{a_pA_{pp} - N\overline{Du}, \overline{D\phi_p}} = \frac{s_{Dv}^2}{s_{DuDv}}, \quad 2 \leq p \leq M. \tag{7.16}$$

This represents a set of $M-1$ equations that must be satisfied by the coefficients c_p, $2 \leq p \leq M$. Using the expansions in Eq. (7.10) and Eq. (7.12), Eq. (7.16) becomes,

$$\frac{c_p A_{pp} - N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i \overline{D\phi_i}\right)}{a_p A_{pp} - N\overline{Du}, \overline{D\phi_p}} = \frac{\sum_{i=2}^{M} c_i^2 A_{ii} - N\left(\sum_{i=2}^{M} c_i \overline{D\phi_i}\right)^2}{\sum_{i=2}^{M} a_i c_i A_{ii} - N\left(\sum_{i=2}^{N} a_i \overline{D\phi_i}\right)\left(\sum_{j=2}^{M} c_j \overline{D\phi_j}\right)}. \quad (7.17)$$

Notice that both the numerator of the left hand side and the denominator of the right hand side are linear in the coefficients c_k. The numerator of the right is a quadratic form in the c_k. This leads to the following important result: For a given nonzero $\alpha \in \mathbb{R}$, if we replace all of the coefficients c_p, $2 \leq p \leq M$, with αc_p, then Eq. (7.17) remains unchanged. In other words, if the particular set of coefficients $\{c_k\}_{k=1}^{M}$ is a solution to the stationarity conditions in Eq. (7.17), then for any $\alpha \neq 0$, the set of coefficients $\{\alpha c_k\}_{k=1}^{M}$ is also a solution. This implies that the solutions to Eq. (7.17) lie on a ray in $\mathbb{R}^{M-1}$.

This remarkable result can be understood when we recall that we are maximizing the correlation between functions. In particular, for any constant $\alpha \neq 0$, the following two correlations are equal,

$$S_3(Du, \alpha Dv) = S_3(Du, Dv). \quad (7.18)$$

This fact can be easily seen from the definition,

$$\begin{aligned} S_3(Du, \alpha Dv) &= \frac{\frac{1}{N-1}\sum_{k=1}^{M}(Du_k - \overline{Du})(\alpha Dv_k - \alpha\overline{Dv})}{s_{Du}\left[\frac{1}{N-1}\sum_{k=1}^{M}(\alpha Dv_k - \alpha\overline{Dv})^2\right]^{1/2}} \\ &= S_3(Du, Dv). \end{aligned}$$

The right hand sides of Eq. (7.16) and Eq. (7.17) are independent of p. This implies that the ratio on the left hand side is constant for all p, i.e.,

$$c_p A_{pp} - N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i \overline{D\phi_i}\right) = \beta_M \left[a_p A_{pp} - N\overline{Du D\phi_p}\right], \quad 2 \leq p \leq M, \quad (7.19)$$

for a constant β_M (where the subscript M indicates the dependency on the number of basis functions used). This tactic is reminiscent of what occurred when we maximized the SSIM in Chapter 5.1. Indeed, recall that stationarity conditions for the SSIM function yielded the following result,

$$c_k = \alpha a_k, \quad 1 \leq k \leq M, \quad (7.20)$$

where the a_k are the Fourier coefficients of the target function. Once again, the solutions are seen to lie on a ray in $\mathbb{R}^{M-1}$. In Chapter 5.1, we were then able to obtain an analytical expression for α in terms of the known Fourier coefficients a_k. Unfortunately, the complicated relationship between the c_i and a_i coefficients in Eq. (7.19) does not permit any kind of substitution of the c_i in terms of the a_i to yield a condition on β_M. We will have to try a different approach to determine a value, or a range of values, for β_M.

Firstly, note that in special case that $\overline{D\phi_p} = 0$ for $2 \leq p \leq M$—which would apply in the continuous case—these equations reduce to

$$c_p = \beta_M a_p, \quad 2 \leq p \leq M. \tag{7.21}$$

In this special case, we retrieve an expression similar to the contrast-enhancement $c_k = \alpha a_k$. The hypothesis $\overline{D\phi_p} = 0$ holds for the DFT basis functions when the gradient operator is defined by forward differences. As one might expect, this assumption is not true in general. In particular, the same property for the DCT basis functions does not carry over from the continuous case: When the gradient operator is defined by forward differences, the derivative DCT basis functions are not necessarily zero mean. In Figure 6.3 at the end of the previous chapter, we see that some DCT derivative basis functions, but not all, have zero mean. (For those basis functions with zero mean, Eq. (7.21) will apply.)

For the general case, observe that using Eq. (7.16) and Eq. (7.19), we also have,

$$\beta_M = \frac{s_{DvDv}}{s_{DuDv}}. \tag{7.22}$$

Using the expression for the correlation $S_3(u, v)$ between two N-vectors given in Eq. (7.1), we can rewrite the expression for β_M as follows,

$$\beta_M = \frac{S_3(Dv, Dv)s_{Dv}s_{Dv}}{S_3(Du, Dv)s_{Du}s_{Dv}} = \frac{1}{S_3(Du, Dv)}\frac{s_{Dv}}{s_{Du}}, \tag{7.23}$$

where we have used the fact that $S_3(Dv, Dv) = 1$. We know that $0 \leq |S_3(Du, Dv)| \leq 1$. Moreover, because v should approximate u, we are expecting a positive correlation, i.e., $0 \leq S_3(Du, Dv) \leq 1$. This implies that

$$\beta_M \geq 0. \tag{7.24}$$

Unfortunately, this seems to be as far as we can go at this moment since the term s_{Dv} contains the unknown coefficients c_k. If, however, we consider the following condition on the approximation v,

$$s_{Du} = s_{Dv}, \tag{7.25}$$

then from Eq. (7.23), it follows that

$$\beta_M \geq 1. \tag{7.26}$$

The condition in Eq. (7.25) implies that the standard deviations (or variances) of the N-vectors of Du and Dv are equal. It is unclear whether or not such a condition is reasonable but it does serve as a kind of reference point in our analysis.

We also attempted another approach in our search for valid β_M values. For each $2 \leq p \leq M$ in Eq. (7.19), multiply both sides of the equation by c_p and then sum over $2 \leq p \leq M$ to yield the following equation,

$$\begin{aligned}\sum_{p=2}^{M} c_p^2 A_{pp} - N\left(\sum_{p=2}^{M} c_p \overline{D\phi_p}\right)\left(\sum_{i=2}^{M} c_i \overline{D\phi_i}\right) \\ = \beta_M\left[\sum_{p=2}^{M} a_p c_p A_{pp} - N\overline{Du}\left(\sum_{p=2}^{M} c_p \overline{D\phi_p}\right)\right].\end{aligned} \tag{7.27}$$

Because the above equation has been obtained from the stationarity condition in Eq. (7.16), it also satisfies the homogeneity property: If $\{c_k\}_{k=1}^{M}$ is a solution of Eq. (7.27), then $\{\alpha c_k\}_{k=1}^{M}$ is also a solution for any $\alpha \neq 0$. From our earlier discussion, each of the summations in the above equation may be identified with a particular quantity, e.g., $\overline{Dv}$. As such, the above equation may be written as follows,

$$\|Dv\|^2 - N(\overline{Dv})^2 = \beta_M[\langle Du, Dv\rangle - N\overline{Du}\overline{Dv}]. \tag{7.28}$$

Note that if $M = N$, then $u = v$ which implies that $Du = Dv$. The above equation would dictate that $\beta_M = 1$. That being said, it is not clear that we can simply use Eq. (7.28). So let us try something else, namely, multiplying both sides of Eq. (7.19) by a_p and summing over $2 \leq p \leq M$, with the understanding that $c_p = 0$ for $M+1 \leq p \leq N$. We arrive at the result,

$$\begin{aligned}\sum_{p=2}^{M} a_p c_p A_{pp} - N\left(\sum_{p=2}^{N} a_p \overline{D\phi_p}\right)\left(\sum_{i=2}^{M} c_i \overline{D\phi_i}\right) \\ = \beta_M\left[\sum_{p=2}^{N} a_p^2 A_{pp} - N\overline{Du}\left(\sum_{p=2}^{N} a_p \overline{D\phi_p}\right)\right].\end{aligned} \tag{7.29}$$

Note that Eq. (7.29) also satisfies the homogeneity property. As before, Eq. (7.29) is equivalent to the following equation,

$$\langle Du, Dv\rangle - N\overline{Du}\overline{Dv} = \beta_M\left[\|Du\|^2 - N(\overline{Du})^2\right]. \tag{7.30}$$

Let us now exploit the fact that Eq. (7.28) and Eq. (7.30) share a common term, namely,

$$\langle Du, Dv\rangle - N\overline{Du}\,\overline{Dv}.$$

We'll multiply Eq. (7.30) and equate the appropriate terms to arrive at the following result,

$$\|Dv\|^2 - N(\overline{Dv})^2 = \beta_M^2[\|Du\|^2 - N(\overline{Du})^2],$$

which can be rewritten as follows,

$$\|Dv\|^2 - \beta_M^2\|Du\|^2 = N\left[(\overline{Dv})^2 - \beta_M^2(\overline{Du})^2\right]. \tag{7.31}$$

Once again, it appears that we can go no farther. In [2], it was shown that a unique solution to the problem of best approximation of vectors using correlation can be obtained only if two additional constraints are applied. Two such constraints, adapted from [2] for our gradient approximation problem, are as follows,

1. Equal gradient norms, i.e., $\|Dv\| = \|Du\|$ and
2. Equal gradient means, i.e., $\overline{Dv} = \overline{Du}$.

Using these constraints, Eq. (7.31) becomes

$$\|Du\|^2(1 - \beta_M^2) = N(\overline{Du})^2(1 - \beta_M^2). \tag{7.32}$$

We see that Eq. (7.32) is satisfied by

$$\beta_M = \pm 1 \quad \text{or} \quad \|Du\| = \sqrt{N}\|\overline{Du}\|. \tag{7.33}$$

Returning to the result in Eq. (7.26), we can discard the negative root. Given that the right side of the second equation above depends only on $u[1]$ and $u[N]$, it is not guaranteed that all functions $u[n]$ will satisfy it. Therefore we conclude that the application of these two constraints, i.e., equal gradient norms and equal gradient means, yields the result $\beta_M = 1$. In the following section, we shall explore the case $\beta_M = 1$ in a computational example to see if "good" approximations are produced. We shall also examine some cases $\beta_M > 1$.

Examples in Correlation-based Best Approximation Between Gradients

Example 1: Consider the discrete reference signal $u[n] = x[n]^2$ evenly sampled $N = 100$ times on the domain of support $[0, 1]$. We will use the usual DCT functions to define the basis $\{\phi_k\}_{k=1}^N$ of $\mathbb{R}^N$, i.e.,

$$\phi_0[n] = \frac{1}{\sqrt{N}}, \quad 0 \le n \le N-1,$$

$$\phi_k[n] = \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right), \quad 1 \le k \le N-1, \ 0 \le n \le N-1, \tag{7.34}$$

where $N = 100$ throughout. The discrete derivative will be defined by simple forward differences, so that

$$D\phi_k[n] = \phi_k[n+1] - \phi_k[n], \quad 0 \le k, n \le N-1. \tag{7.35}$$

An even extension of the data will be assumed, i.e., $\phi_k[N] = \phi_k[N-1]$, for $0 \le k \le N-1$.

From Theorem 4, the derivatives of the DCT basis functions $\{\phi_k\}_{k=1}^N$ defined above form an orthogonal set. The orthogonality relation is

$$\langle D\phi_k, D\phi_l \rangle = \begin{cases} 0, & k \neq l \\ A_k, & k = l, \end{cases} \tag{7.36}$$

where $A_0 = 0$ and $A_k = 4\sin^2\left(\frac{k\pi}{2N}\right) > 0$ for $1 \le k \le N-1$. Once more, we refer the reader to the proof in Appendix A.2 for details.

For various choices of M and a given β_M, we are now interested in exploring the nature of solutions v_M to Eq. (7.19). Note that when β_M is fixed, Eq. (7.19) is a linear system of coefficients. (When we fix β_M, we are fixing a unique solution on the ray in $\mathbb{R}^{M-1}$ which satisfies Eq. (7.19).) To find the optimal coefficients c_p, we will solve Eq. (7.19), which holds for $2 \le p \le M$, in Maple. For the remaining coefficient, we will simply let $c_1 = a_1$, the first Fourier coefficient of the reference signal $u[n]$ in the $\{\phi_k\}$ basis.

In our preceding analysis, we found that the constant β_M, the reference signal u, and the best approximation v_M should satisfy Eq. (7.31), rewritten below,

$$\|Dv_M\|^2 - \beta_M^2 \|Du\|^2 = N\left[(\overline{Dv_M}^2 - \beta_M^2(\overline{Du}^2)\right].$$

We also observed that imposing (1) equal gradient norms and (2) equal gradient means in the above relation implies that $\beta_M = 1$.

In our experiments, we would like to verify that Eq. (7.31) is in fact being satisfied. Bringing everything to one side and taking the absolute value, we tabulate the value of the following expression,

$$\left| \|Dv_M\|^2 - \beta_M^2 \|Du\|^2 - N\left[(\overline{Dv_M})^2 - \beta_M^2(\overline{Du})^2\right] \right|, \tag{7.37}$$

which, once again, we expect to be close to 0. We also tabulate the values of $\|Dv_M\|$ and $\|Du\|$, whose equality would satisfy condition (1), along with the values of $\overline{Dv_M}$ and $\overline{Du}$, whose equality would satisfy condition (2). If both condition (1) and (2) are met, then it follows that $\beta_M = 1$. However, in some experiments we fix $\beta_M = 1$, from which it is not guaranteed that the norms and means are pairwise equal.

In this first example, we fix $\beta_M = 1$ and vary M, the number of basis functions used in the approximation v_M. In Figure 7.1 are plotted the solutions v_M to the stationarity conditions in Eq. (7.19) for (a) $M = 5$, (b) $M = 20$, (c) $M = 40$ and (d) $M = 80$ basis functions. In all cases, the best-SSIM approximation and best-L^2 approximations, both also using M basis functions, are plotted for comparison. (In each of Figure 7.1 (a)-(d), to the resolution of the plot, the best-SSIM and best-L^2 approximations are virtually equal.) As one would expect, v_M offers a good approximation to the target function $u[n]$ for large M. And while we do expect the approximations to get poorer as M decreases, there appears to be a critical value of M where the poorness of the results is quite staggering. Indeed, v_5, depicted in Figure 7.1 (a), offers a strikingly poor approximation of the target function, hovering around the average value of $u[n]$. By comparison, the best-SSIM and best-L^2 curves, restricted to same dimensionality, are already performing significantly better.

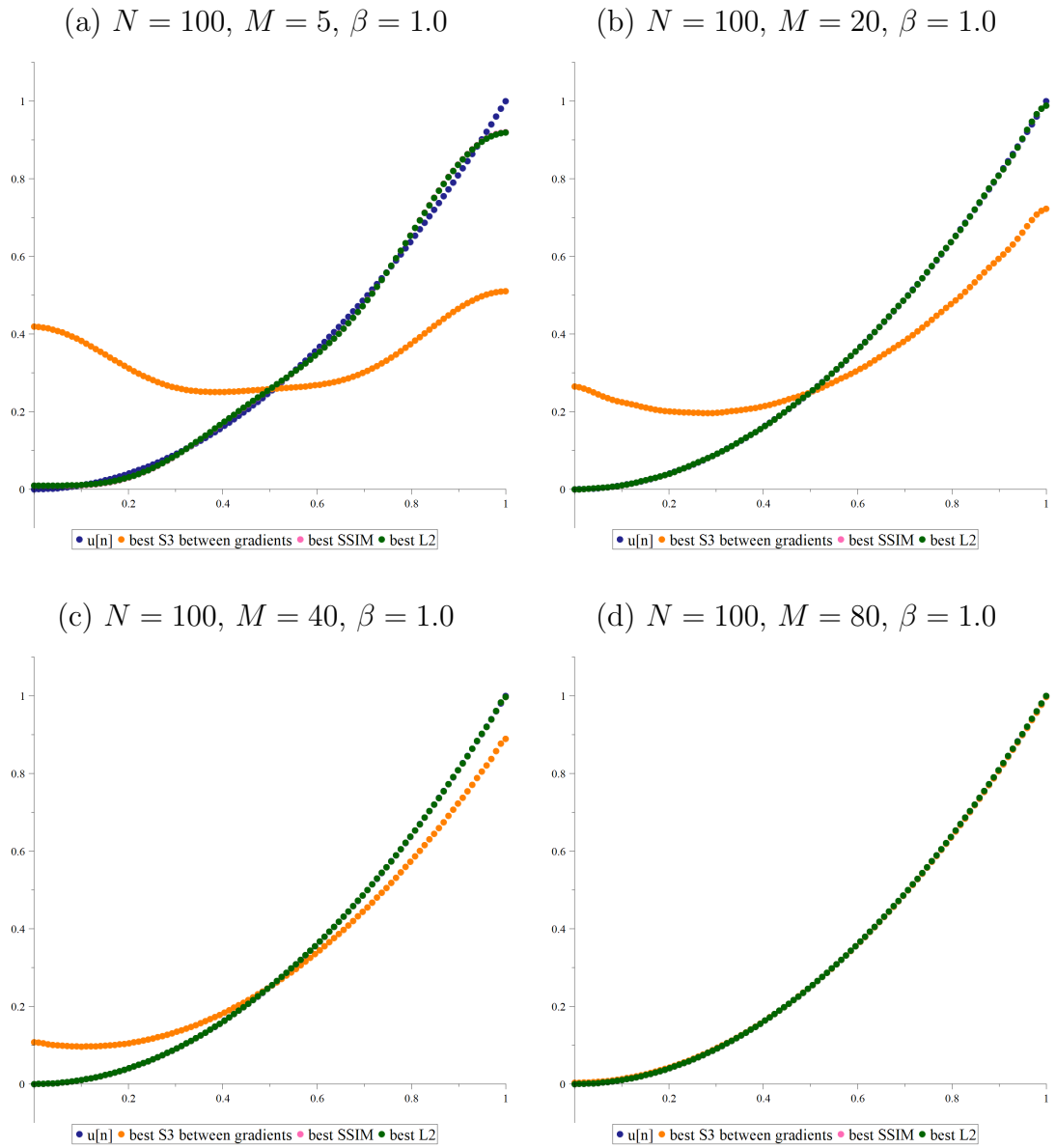


Figure 7.1: Correlation-based approximations between gradients to the function $u(x) = x^2$ on $[0, 1]$ using (a) $M = 5$, (b) $M = 20$, (c) $M = 40$ and (d) $M = 80$ basis functions. In all cases, $\beta = 1$. For each M, the M-dimensional best-SSIM approximation and the M-dimensional best-L^2 approximation are plotted for comparison. To the resolution of the plots, the best-SSIM and best-L^2 approximations are virtually equal for all 4 values of M.

	$M = 5$	$M = 20$	$M = 40$	$M = 80$
$\overline{Du}$	0.01000	0.01000	0.01000	0.01000
$\overline{Dv}$	0.00091	0.00458	0.00781	0.00993
$\|Du\|$	0.11605	0.11605	0.11605	0.11605
$\|Dv\|$	0.05222	0.07335	0.09758	0.11545
Eq. (7.37)	8.242×10^{-4}	1.856×10^{-4}	5.284×10^{-5}	1.52×10^{-6}

Table 7.1: Values of $\overline{Dv_M}$ and $\|Dv_M\|$ for variable M where $\beta = 1$. The values of $\overline{Du}$ and $\|Du\|$, which do not vary with M, are included for comparison. We also include the value of the expression in Eq. (7.37), which we expect to be near 0.

In Table 7.1, we present the values of $\overline{Dv_M}$, $\|Dv_M\|$ and Eq. (7.37) for each of $M = 5, 20, 40, 80$ as depicted in Figure 7.1. The values $\overline{Du}$ and $\|Du\|$, which depend only on $N = 100$ and are constant with respect to changes in M, are also included for comparison. We find that the condition described by Eq. (7.37) is roughly satisfied; for each value of M, this number is near 0. As M increases, the value of Eq. (7.37) gets even smaller. We know that $v_M \to u$ as M increases towards $N = 100$, hence it is not surprising that $\overline{Du} \approx \overline{Dv_{80}}$ and $\|Du\| \approx \|Dv_{80}\|$ according to the entries in Table 7.1. On the other hand, conditions (1) and (2) are not roughly met for the smaller values of M. The fact that Eq. (7.37) is roughly 0 indicates that some other balance of terms is occurring.

The poorness of the approximations for low M in Figure 7.1 is somewhat surprising. What is perhaps even more surprising, looking at the plots, is that the correlations between the gradients of these approximations and that of the target function are greater than the correlations between the best L^2 gradients and SSIM approximation gradients and that of the target function! (In order to understand this better, one would need to examine the derivatives, which we have done but have not reported.)

In this example, our value of $N = 100$ is quite high. By contrast, when approximating a digital image, we would likely be using such an approximation for rather low values of N, e.g., the 8×8 blocks that the JPEG standard uses. The candidate values of M will still be rather moderate by comparison, and may still produce reasonable approximations. It may be worthwhile to investigate how these approximations differ from their Fourier and best-SSIM counterparts. However, because this method did not seem too promising, we did not pursue the 2D case while completing this thesis.

For our second example, we explore the role of the parameter β_M in Eq. (7.19). While we know that Eq. (7.19) holds for some constant β_M, we were not able to fully determine its value. Our analysis in the previous section suggests that we may expect $\beta_M \geq 1$. In

Figure 7.2, we fix $M = 60$ basis functions and plot the best approximations v_M for (a) $\beta_M = 1.0$, (b) $\beta_M = 1.05$, (c) $\beta_M = 1.1$ and (d) $\beta_M = 1.3$. In Table 7.2, we present the values of $\overline{Dv_M}$, $\|Dv\|$ and Eq. (7.37) for each of the four plots in Figure 7.2. For the values of $\beta = 1.0, 1.05$, the gradient norms and means are closest to being equal. The values described by Eq. (7.37) are also at their smallest. As β increases, this value grows away from 0 as expected. The gradient means and norms also grow further away from each other, respectively.

	$\beta_M = 1.0$	$\beta_M = 1.05$	$\beta_M = 1.1$	$\beta_M = 1.3$
$\overline{Du}$	0.01000	0.01000	0.01000	0.01000
$\overline{Dv_M}$	0.00941	0.00988	0.01035	0.01223
$\|Du\|$	0.11605	0.11605	0.11605	0.11605
$\|Dv_M\|$	0.11091	0.11646	0.07335	0.09758
Eq. (7.37)	1.307×10^{-5}	1.44×10^{-5}	1.582×10^{-5}	2.209×10^{-5}

Table 7.2: Values of $\overline{Dv_M}$ and $\|Dv_M\|$ for $M = 60$ basis functions and variable β. The values of $\overline{Du}$ and $\|Du\|$, which do not vary with β_M, are included for comparison. We also include the value of the expression in Eq. (7.37), which we expect to be near 0.

In this section, we have derived the stationarity conditions for the maximization of $S_3(Du, Dv_M)$, the correlation between the gradient of a target function $u \in \mathbb{R}^N$ and the gradient of its M-dimensional approximation v_M. An infinity of solutions exists: If $\{c_k\}_{k=2}^M$ is a solution, then so is $\{\alpha c_k\}_{k=2}^M$ for any $\alpha \in \mathbb{R}$. This follows from the fact that $S_3(Du, Dv_M) = S_3(Du, \alpha Dv_M)$. A unique solution can be obtained by imposing two constraints employed in best SSIM-based approximation but adapted to this problem, namely, (1) equal means, $\overline{Du} = \overline{Dv}$, and (2) equal norms, $\|Du\| = \|Dv_M\|$. In "nice" cases, i.e., when M is sufficiently large, the approximations yielded by this method are reasonably "good", as compared to best L^2-based and best-SSIM-based approximations. But when M is small, the approximations yielded by this method can be quite poor. A most likely reason for this problem is the lack of connection between the values of the approximation and the target function since the method deals only with derivatives. One way to overcome this difficulty, which also avoids the need to impose the equal norms constraint, is to include a "regularization term" in the cost function which penalizes large distances between u and its approximation v_M and which can be expressed in terms of their expansion coefficients, e.g.,

$$\lambda \|u - v_M\|_2^2 = \lambda \sum_{k=2}^{M} (a_k - c_k)^2 + \lambda \sum_{k=2}^{N} a_k^2 \,. \tag{7.38}$$

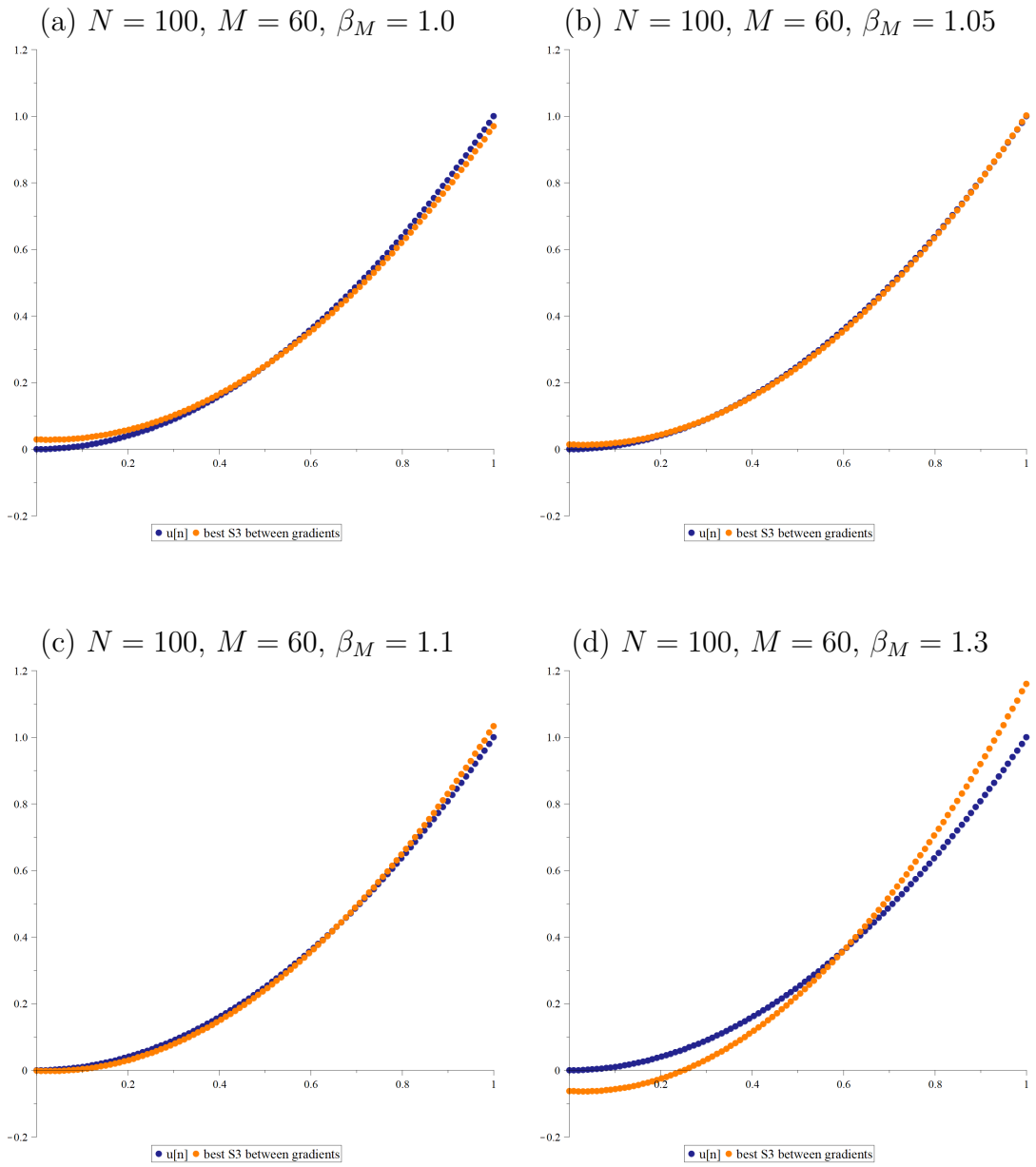


Figure 7.2: Correlation-based approximations between gradients to the function $u(x) = x^2$ on $[0, 1]$ using $M = 60$ basis functions, with (a) $\beta_M = 1.0$, (b) $\beta_M = 1.05$, (c) $\beta_M = 1.1$ and (d) $\beta_M = 1.3$.

(Of course, the final term can be ignored.) We have performed a few experiments and found that for very small values of $\lambda > 0$, the approximations yielded by this regularized scheme are very much improved. That being said, these methods will not be discussed in this thesis since (1) only a few experiments were performed and (2) a proper discussion and analysis of the method—which is an extremely interesting one—would require at least another chapter.

7.2 Best Approximation by Maximizing the SSIM Between Gradient Vectors

Recall that in Chapter 5.1, we maximized the following SSIM function between two continuous-time signals,

$$\mathrm{SSIM}(u, v) = \frac{4\bar{u}\bar{v}\sigma_{uv}}{(\bar{u}^2 + \bar{v}^2)(\sigma_u^2 + \sigma_v^2)}. \tag{7.39}$$

Eq. (7.39) corresponds to a special case of the SSIM in which the S_2 and S_3 terms have collapsed into a single term. (This combination is made possible by choosing stability constants such that $C_2 = 2C_3$.) In Eq. (7.39), we have also made the simplifying assumption that all three stability constants C_1, C_2, and C_3 are equal to 0.

In this section, we will maximize the SSIM between gradient vectors. In the following problem, we will omit the S_1 term of the SSIM since it involves only mean values. We expect, as occurred in [3], that we will need to use the S_1 term in order to determine the first coefficient c_1 of the approximation.

With that understanding, we define the following SSIM-based similarity function for vectors in $\mathbb{R}^N$,

$$S(u, v) = S_2(u, v)S_3(u, v) = \frac{2s_{uv}}{s_u^2 + s_v^2} = \frac{2s_{uv}}{s_{uu} + s_{vv}}. \tag{7.40}$$

We now consider the following best approximation problem in $\mathbb{R}^N$ in terms of this SSIM-based similarity between two gradient vectors: For a given target function $u \in \mathbb{R}^N$, with Fourier expansion

$$u = \sum_{i=1}^{N} a_i \phi_i, \quad \text{where} \quad a_k = \langle u, \phi_k \rangle, \quad 1 \leq k \leq N,$$

and an $1 \leq M < N$, find the approximation of the form,

$$v_M = \sum_{i=1}^{M} c_i \phi_i,$$

which maximizes the similarity between the gradients of u and v, as measured by $S(u, v)$ in Eq. (4.2),

$$c = \arg\max_{d \in \mathbb{R}^M} S(Du, Dv_M).$$

Here,

$$Du = \sum_{i=1}^{N} a_i D\phi_i \quad \text{and} \quad Dv_M = \sum_{i=1}^{M} c_i D\phi_i.$$

As before, we will let v denote our best approximation going forward, omitting the subscript, to avoid confusion when extracting its component parts.

The statistics in the objective function,

$$S(Du, Dv) = \frac{2 s_{DuDv}}{s_{DuDu} + s_{DvDv}}, \tag{7.41}$$

were already expressed in terms of the expansion coefficients in the previous section. As before, s_{DuDu} does not depend on the unknown coefficients c_k. For the remaining terms, we substitute the expansions obtained in the previous section to get

$$S(Du, Dv) = \frac{2\left[\sum_{i=2}^{M} a_i c_i A_{ii} - N\left(\sum_{i=2}^{N} a_i \overline{D\phi_i}\right)\left(\sum_{j=2}^{M} c_j \overline{D\phi_j}\right)\right]}{s_{DxDx} + \left[\sum_{i=2}^{M} c_i^2 A_{ii} - N\left(\sum_{i=2}^{M} c_i \overline{D\phi_i}\right)^2\right]}.$$

We now examine the structure of the partial derivative of $S(Du, Dv)$ with respect to c_p for $2 \leq p \leq M$. As before, letting the prime sign denote the partial derivative with respect to c_p,

$$\frac{\partial S(Du, Dv)}{\partial c_p} = \frac{2\left[(s_{DuDu} + s_{DvDv}) s'_{DuDv} - s_{DuDv} s'_{DvDv}\right]}{(s_{DuDu} + s_{DvDv})^2}, \quad 2 \leq p \leq M.$$

The stationarity condition,

$$\frac{\partial S(Du, Dv)}{\partial c_p} = 0,$$

implies that the expression on the right be zero, i.e.,

$$(s_{DuDu} + s_{DvDv}) s'_{DuDv} - s_{DuDv} s'_{DvDv} = 0,$$

which implies that,

$$(s_{DuDu} + s_{DvDv})s'_{DuDv} = s_{DuDv}s'_{DvDv}.$$

We now proceed as before by rearranging the above equation to obtain,

$$\frac{s'_{DvDv}}{s'_{DuDv}} = \frac{s_{DuDu} + s_{DvDv}}{s_{DuDv}}.$$

Substituting the expressions for the appropriate derivatives from the previous section into the LHS of the above equation, we have

$$\frac{2c_pA_{pp} - 2N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i\overline{D\phi_i}\right)}{a_pA_{pp} - N\overline{DxD\phi_p}} = \frac{s_{DuDu} + s_{DvDv}}{s_{DuDv}}.$$

Dividing both sides by 2, we arrive at the following result,

$$\frac{c_pA_{pp} - N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i\overline{D\phi_i}\right)}{a_pA_{pp} - N\overline{DxD\phi_p}} = \frac{s_{DuDu} + s_{DvDv}}{2s_{DuDv}}, \quad 2 \leq p \leq M.$$

Once again, the RHS of each of the above $M-1$ equations is independent of p, implying that the ratio on the LHS is constant for all p, i.e.,

$$c_pA_{pp} - N\overline{D\phi_p}\left(\sum_{i=2}^{M} c_i\overline{D\phi_i}\right) = \beta_M\left[a_pA_{pp} - N\overline{DxD\phi_p}\right], \quad 2 \leq p \leq M,$$

for some constant β_M. But we note that this is the same condition as Eq. (7.19) found in the previous section. In other words, our use of the SSIM function as a measure of the similarity of two gradient functions yields nothing new.

Chapter 8

The Einstein Images Revisited

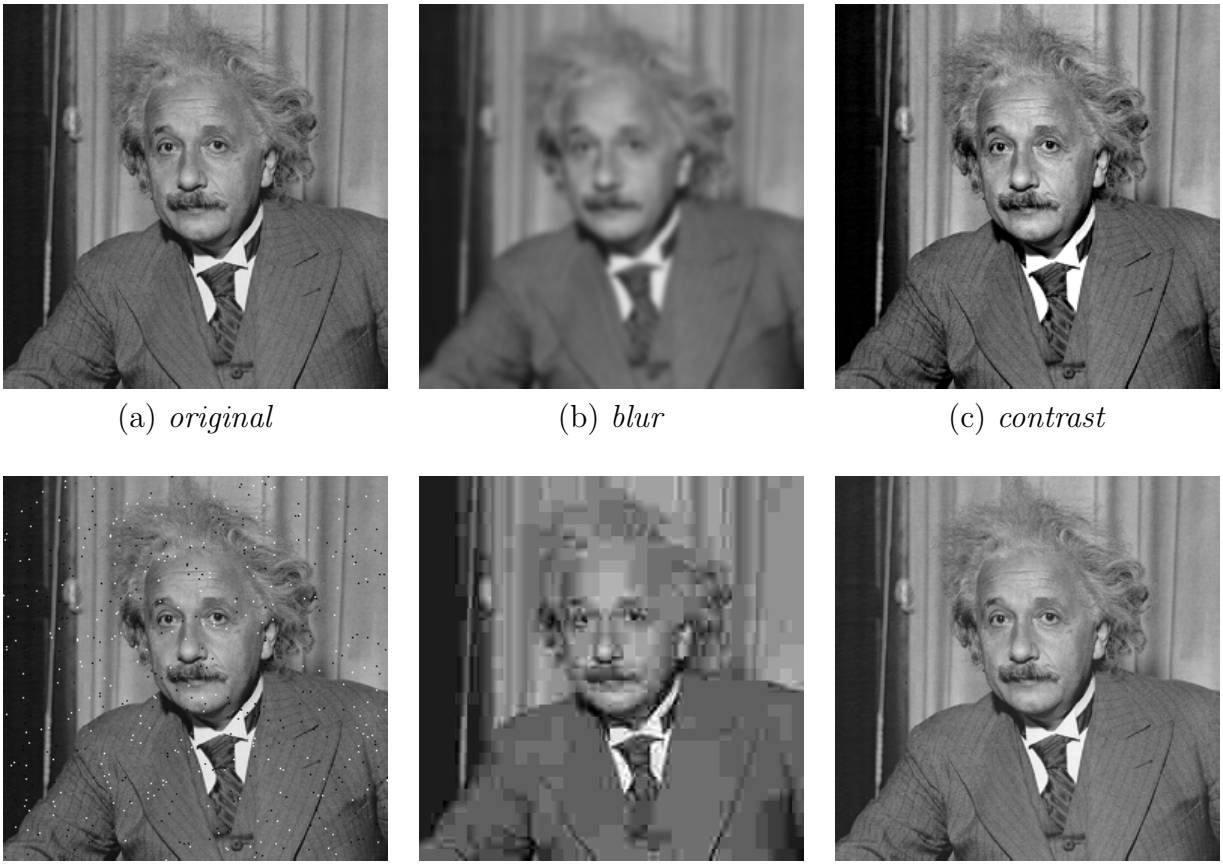
8.1 MSE Between Gradient Vectors

This chapter marks a transition in our treatment of gradients in image processing applications. Following our introduction to gradients via best approximation methods, we now embark on our in-depth investigation of gradient similarity measures and their incorporation into the SSIM.

We arrived, at this moment in our winding explorations, newly afaced with those Einstein images recalled in Figure 8.1. It may be necessary to restate some details with which we had been previously well-acquainted: The six Einstein portraits are each a 256×256 pixel, 8 bits-per-pixel greyscale image. They are notorious for having nearly-equal MSE, despite exhibiting varying degrees of perceptual quality. In Chapter 4, the Einstein images allowed us to analyze the SSIM and, in particular, the respective roles of each of its three component parts. Although no definitive conclusions could be drawn from our simple experiments, those results did support our belief that the correlation is the most important component of the SSIM.

We now present our second set of so-called "Einstein experiments". Although simple-minded, the following experiments highlight once again our inclination towards a stripped back approach based on first principles, a position which characterizes our work throughout this thesis.

Our focus in this section is to begin our computational exploration of the gradient. The first step is, of course, deciding on the manner in which to compute gradients. There are numerous reasonable choices: Indeed, we acknowledge that in the image processing



(a) *original* (b) *blur* (c) *contrast*

(d) *impulse* (e) *jpg* (f) *meanshift*

Figure 8.1: The reference Einstein image *original* and its perturbations.

literature, many different formulas—which can be expressed in terms of "filters" operating on the matrices representing the images—are employed. Furthermore, different gradient filters will most probably yield different computational results. That being said, our primary purpose here is to introduce the idea of using gradients in image quality measures. As such, we continue to employ simple forward differences to compute our gradients and consider an exploration of other methods to be beyond the scope of this thesis.

Let x denote the reference image *original* and y denote the appropriate degraded image. For $1 \leq i, j \leq 256$, the gradient of the reference image x at the $(i,j)^{\text{th}}$ pixel is defined by,

$$\nabla x_{ij} = (x(i+1,j) - x(i,j),\ \ x(i,j+1) - x(i,j)), \tag{8.1}$$

except at the edges of the image, i.e., when $i = 256$ or $j = 256$. Since the index $256 + 1$ lies outside the region over which the image is defined, the image must be extended. We have assumed an even extension of the image, so that $x(256 + 1, j) = x(256, j)$ and $x(i, 256 + 1) = x(i, 256)$. This produces a zero value for the appropriate component of the gradient vector. A similar expression holds for a degraded image y.

Underlying the following explorations is the question of how sensitive the human visual system is to changes in the gradient vector. For our first simple attempt at addressing this question, we will compute the usual L^2 distance *between gradients*. If the structural information encoded in our simplistic gradient allows the MSE, which does not otherwise consider spatial relationships between pixels, to differentiate between the Einstein images, then this would be substantial evidence in favour of applying gradients in image quality assessment.

For the *original* image x and each of the 5 perturbations y, we compute the following distances,

$$\|\nabla x - \nabla y\|_2 := \frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{N}\|\nabla x_{ij} - \nabla y_{ij}\|_2^2\right]^{1/2}, \tag{8.2}$$

where $N = 256$ and the gradients ∇x and ∇y are computed using simple forward differences as previously described. The results of the computation are presented in Table 8.1 below.

blur	*contrast*	*impulse*	*jpg*	*meanshift*
17.0349	5.5426	23.9110	17.6135	0.1309

Table 8.1: RMSE between the gradients of the original *Einstein* images and its perturbations.

We can immediately observe a significant deviation in the values of these distances. It is clear that computing the MSE *between gradients* already offers a marked improvement over the traditional MSE. However, the results in Table 8.1 become even more encouraging after a closer look. Recall that in Chapter 4, we discussed the following ranking of the Einstein images as prescribed by the MSSIM,

$$\textit{meanshift} \;>\; \textit{contrast} \;>\; \textit{impulse} \;>\; \textit{blur} \;>\; \textit{jpg}. \tag{8.3}$$

In terms of the distances reported in Table 8.1, the *meanshift* image is once again closest to the *original* Einstein image. Since the *meanshift* image is produced from *original* by merely adding a constant to the greyscale values of the latter, the gradient of the two images are identical. (The observed deviation is due the presence of a few pixels whose *meanshift*

value is restricted by limits on the greyscale range, i.e, restricted to the maximum possible value $L = 255$.) The *contrast* image is next lowest in terms of gradient distance. These two results are in agreement with the ordering of image quality dictated by the MSSIM and reported in Eq. (8.3). Indeed, if we let the gradient distances define an ordering of image quality, it would be as follows,

$$meanshift \;>\; contrast \;>\; blur \;>\; jpg > \;\; impulse \;. \tag{8.4}$$

We see that the relative ordering of *blur* and *jpg* is also the same as for the MSSIM. In fact, the only difference between Eq.(8.4) and Eq. (8.3) is the placement of the *impulse* image. According to the gradient distance, *impulse* is moved to the end of the line, the furthest from *original*. The addition of impulse noise strongly affects not only the gradient at each contaminated pixel, but also the gradients of its adjacent neighbours. The increased disparity between adjacent intensity values is evidently, and understandably, bothersome to the gradient-based MSE. It is possible that some preprocessing methods, such as blurring or downsampling of the image, could mitigate this effect, perhaps even to the extent that one could retrieve a ranking of gradient distance in agreement with Eq. (8.3).

8.1.1 The Quest for a Gradient Similarity Measure

The results of the previous section suggest that there is a compelling relationship between the visual closeness of images and the distances between their gradients. The question still remains of how best to measure gradient similarity. In the following section, we will formulate a few different measures of gradient similarity, all of which clearly seek to match gradient vectors. We will compare the performance of our various formulations by applying them to the Einstein images.

Our foremost source of inspiration on which to model a gradient similarity measure was, of course, the correlation. Indeed, in the same way that the SSIM computes the correlation between two $M \times M$ patches x and y, one might be tempted to compute the correlation between their gradients ∇x and ∇y. However, ∇x and ∇y will both be an $M \times M$ block of vectors, i.e., each component of the $M \times M$ matrix will be a 2-vector as written in Eq. (8.1). These details illuminate our motivating question. We are, in fact, faced with the following: How does one measure the similarity between two blocks of vectors?

For our first approach, recall that the correlation between two image patches $x, y \in \mathbb{R}^N$ can be interpreted as the cosine of an angle θ, i.e.,

$$S_3(x, y) = \frac{s_{xy}}{s_x s_y} = \frac{x_0 \cdot y_0}{\|x_0\|\|y_0\|} = \cos\theta,$$

where

$$x_0 = x - \bar{x} \qquad \text{and} \qquad y_0 = y - \bar{y},$$

as discussed on p. 13.

In a similar fashion, our first gradient similarity measure is based on the angle between gradient vectors. Given the two blocks of vectors ∇x and ∇y, we compute the cosines of the angles between their respective components $\nabla x_{ij}, \nabla y_{ij} \in \mathbb{R}^2$,

$$\cos\theta_{ij} = \frac{\nabla x_{ij} \cdot \nabla y_{ij}}{\|\nabla x_{ij}\|\|\nabla y_{ij}\| + C_4}, \qquad 1 \le i, j \le M, \tag{8.5}$$

and then take the average of these cosines over the entire block,

$$\overline{\cos\theta} = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \cos\theta_{ij}. \tag{8.6}$$

The gradient vectors ∇x_{ij} and ∇y_{ij} are both defined by two pieces of information: the orientation angle and the magnitude. The above approach is, of course, only taking into consideration the former of these two qualities. It is not yet clear if this is actually disadvantageous. It is unknown at this point if the human visual system is sensitive to changes in gradient angle, changes in gradient magnitude, or some weighting of the two. Testing Eq. (8.6) on the Einstein images should give us an indication of the efficacy of using angle information only.

Notice that in Eq. (8.5), we have included a small stability constant C_4 in the denominator only. This is, of course, to protect against numerical instabilities should either ∇x_{ij} or ∇y_{ij} be close to $(0, 0)$. However, it is quite possible that a given pixel location (i, j), *both* gradient vectors ∇x_{ij} and ∇y_{ij} are close to $(0, 0)$. In such a case, one can make a claim that the two vectors are quite similar—in fact, identical. It may be tempting, in light of this possibility, to insert a stability constant in both the numerator and denominator of Eq. (8.5). On the other hand, it may well be the case that the dot product produces a *bona fide* zero value—in which case we would prefer to revert to having a stability constant in the denominator only. This dilemma involving the stability constants will be further explored in Chapter 9. For the moment, we simply consider the formulation stated in Eq. (8.5). In our computations, we set $C_4 = 10^{-5}$.

As discussed in Chapter 4, p.30, we will compute our similarity measures, including Eq. (8.6), using non-overlapping patches of various sizes. After computing the value of Eq. (8.6) for each patch, we compute the mean over the entire image. The results of this computation are presented in Table 8.2 below.

	8×8	16×16	32×32	64×64
blur	0.3052	0.3052	0.3052	0.3052
contrast	0.9766	0.9766	0.9766	0.9766
impulse	0.9787	0.9787	0.9787	0.9787
jpg	0.1977	0.1977	0.1977	0.1977
meanshift	1.0000	1.0000	1.0000	1.0000

Table 8.2: Average $\overline{\cos\theta}$ values between the *original* Einstein image and its degradations for various patch sizes using Eq. (8.6).

We not that for each degraded image, the values of $\overline{\cos\theta}$ reported in Table 8.2 do not differ with patch size. This is to be expected. Similar to the computation of the RMSE, we are first finding the average angle in each block, then averaging over all blocks, so that the final result is just an average of angles over all pixels in the image.

We also note that the gradient correlations for the *blur* and *jpg* images are significantly lower than the scores for the other degraded images. Blurring and strong JPEG compression affect the gradients of an image significantly—blurring dampens the gradients while, in JPEG compression, many blocks with zero gradient are produced.

On the other hand, the $\overline{\cos\theta}$ values for *contrast*, *meanshift*, and *impulse* are all quite high. In the *contrast* image, the gradients are scaled and therefore correlation is preserved. In the *meanshift* image, we expect the gradient to be identical because the degraded image is simply a shifted version of the *original* image. In the *impulse* image, the modification of a few isolated pixels contributes little to the averages taken. Reflecting on these results, it may be of concern that the generous *impulse* value in Table 8.2, which exceeds even the *contrast* value, disrupts the ranking of the images imposed by the MSSIM in Eq. (8.3).

A natural variation of this method leads us to our second proposed gradient similarity measure. In this second related method, we first compute the mean vector of each block,

$$\overline{\nabla x} = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\nabla x_{ij} \quad \text{and} \quad \overline{\nabla y} = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\nabla y_{ij}, \tag{8.7}$$

and then compute the correlation between these two mean vectors as follows,

$$\cos\overline{\theta} = \frac{\overline{\nabla x}\cdot\overline{\nabla y}}{\|\overline{\nabla x}\|\|\overline{\nabla y}\| + C_4}. \tag{8.8}$$

Once again, we have set $C_4 = 10^{-5}$ in our computation. In Table 8.3 we present the average correlation values obtained by applying this method to the *original* Einstein image and its degradations for various block sizes.

	8×8	16×16	32×32	64×64
blur	0.8245	0.8877	0.9568	0.9979
contrast	0.9938	0.9932	1.0000	0.9999
impulse	0.9055	0.9267	0.9527	0.9969
jpg	0.5252	0.6762	0.8814	0.9834
meanshift	1.0000	1.0000	1.0000	1.0000

Table 8.3: Average $\cos\overline{\theta}$ values between the *original* Einstein image and its degradations for various patch sizes using Eq. (8.8).

The values in this table are generally quite high, with many individual entries exceeding 0.9. The only exceptions are the *blur* entries for 8×8 and 16×16 blocks and the *jpg* entries for all blocks but those sized 64×64 pixels. By first taking the average value of the gradient vectors of a block and then computing a correlation, one might argue that a great deal of information about the block—including any similarities of lack of similarities of component gradient vectors—is lost. Indeed, the rather large values of the *blur* entries—and even the less inflated *jpg* entries—suggest that the method described by Eq. (8.8) may not be a good way to compute gradient similarity.

With all of the above work done involving angles of gradient vectors, we haven't actually determined any correlation between the entire sets of data ∇x and ∇y. Rather, we have so far only considered averaging the cosine of angles existing between their individual components. For our next formulation, we wanted to compute, in some way, the correlations of the angles of the corresponding gradient vectors of the two images.

Let the gradient vectors of the images x and y be expressed in both Cartesian and polar form as follows,

$$\begin{aligned}\nabla x_{ij} &= (a_{ij}, b_{ij}) = r_{ij}e^{i\theta_{ij}}, \text{ where } r_{ij} = \sqrt{a_{ij}^2 + b_{ij}^2} \text{ and } \theta_{ij} = \tan^{-1}(b_{ij}/a_{ij}) \\ \nabla y_{ij} &= (c_{ij}, d_{ij}) = r_{ij}e^{i\theta_{ij}}, \text{ where } r_{ij} = \sqrt{c_{ij}^2 + d_{ij}^2} \text{ and } \phi_{ij} = \tan^{-1}(c_{ij}/d_{ij}). \end{aligned} \tag{8.9}$$

We then compute the correlation between the two M^2 vectors defined by the angles θ_{ij} and ϕ_{ij}, i.e.,

$$S_4(\theta, \phi) = \frac{s_{\theta\phi}}{s_\theta s_\phi + C_4} \tag{8.10}$$

where the covariance $s_{\theta\phi}$ and the variances s_θ and s_ϕ are defined in the usual way.

The average correlations obtained by the above method as applied to the *original* Einstein image and its degradations for various block sizes are presented in Table 8.4.

	8×8	16×16	32×32	64×64
blur	0.2039	0.2388	0.2529	0.2561
contrast	0.9839	0.9849	0.9852	0.9855
impulse	0.9819	0.9826	0.9828	0.9828
jpg	0.1321	0.1756	0.1982	0.2082
meanshift	1.0000	1.0000	1.0000	1.0000

Table 8.4: Average block-correlation values $S_4(\theta, \phi)$ between the *original* Einstein image and its degradations for various patch sizes using Eq. (8.10).

All of this being said, we will not discuss the method defined by Eq. (8.10) at length in this thesis. Given the periodic nature of the angle, complications are introduced based on the fundamental interval of definition. The values in Table 8.4 correspond to $\theta_{ij} \in [-\pi, \pi]$, the default interval in Matlab. In this scheme, the values $-\pi + \epsilon$ and $\pi + \delta$ are nearly 2π apart in the absolute sense, while we would prefer to interpret those angles periodically as being very close together. Shifting the fundamental interval of definition has significant impact on the values in Table 8.4, to the extent of changing the relative ordering of the images, and is based on the unpredictable distribution of the angles of the gradient vectors in the given images. Although we did explore this issue, we will be omitting any further discussion relating to Eq. (8.10).

To devise yet another gradient similarity measure, we turned our focus to addressing a different concern. Up to this point, we had been bothered by the relatively high *impulse* scores in Table 8.2 and Table 8.3. It is likely that the impulse noise affects the magnitudes of the gradient vectors more than their angles, which suggests that we should also examine the correlations of the magnitudes of the gradient vectors, i.e.,

$$S_4(r, s) = \frac{s_{rp}}{s_r s_p + C_4}, \tag{8.11}$$

where the magnitudes r, p are computed according to the formulas in Eq. (8.9). In Table 8.5 below we present the average correlation of gradient magnitudes between the *original* Einstein image and its degradations for various block sizes.

We see that the *impulse* correlation values are noticeably lower than their counterparts in both Table 8.2 and Table 8.3. This indicates that the magnitudes of the gradients vectors are providing at least some way of detecting the distortion produced by impulse noise. The relative ordering of the distorted images imposed by the MSSIM and described by Eq. (8.3) is also maintained for all attempted block sizes. Based on these two observations, it appears that the gradient magnitudes provide a better indication of image quality than the gradient angles alone.

	8×8	16×16	32×32	64×64
blur	0.2298	0.3108	0.4127	0.4649
contrast	0.9807	0.9849	0.9975	0.9974
impulse	0.6971	0.4823	0.4518	0.4737
jpg	0.1361	0.2351	0.3801	0.4565
meanshift	1.0000	1.0000	1.0000	1.0000

Table 8.5: Average block-correlation values $S_4(r, s)$ between the *original* Einstein image and its degradations for various patch sizes using Eq. (8.11).

For our final proposed measure, we step away from methods using one of either the gradient angle or magnitude to instead consider the individual x and y components of the vector. For this final method, we compute the correlation between the components of the gradient vector separately. For two corresponding image blocks, one taken from the reference image and one from a degradation, we let a denote the correlation between the x-components of the blocks and b denote the correlation of the y-component of the blocks. In this way, we obtain two correlations for each block, which we present as an ordered pair (a, b). The question is then how to extract one number from this pair of values. There are a number of reasonable possibilities, e.g., taking the maximum, minimum, or average. We have chosen to compute the "normalized magnitude" of the vector (a, b), i.e., the quantity

$$S_4(a, b) = \frac{1}{\sqrt{2}}(a^2 + b^2)^{1/2}, \tag{8.12}$$

which lies in the range $[0, 1]$. As usual, we then compute the average $S_4(a, b)$ across all blocks in the image. For the *original* Einstein image and each of its perturbations, we present those average "normalized magnitudes" for various block sizes in Table 8.6 below. (When computing the correlations a and b, we have once again included the small stability constant C_4 in the denominator only.)

As was the case for the correlation of magnitudes presented in Table 8.5, these *impulse* values are more punished, although not to the same extent. For all block sizes, these values also maintain the relative ordering imposed by the MSSIM in Eq. (8.3). At this point, it seems that both the correlation of magnitudes and "normalized magnitude" methods are performing at least reasonably well.

This completes the presentation of our novel methods of computing gradient similarity. However, during the course of this work, we became aware of a pre-existing method to compute the correlation between vectorial data sets. It is known as the "canonical correlation" method, introduced by mathematician Harold Hotelling in 1936 [8]. It is, in a sense,

	8×8	16×16	32×32	64×64
blur	0.3551	0.3969	0.4391	0.4619
contrast	0.9902	0.9922	0.9964	0.9975
impulse	0.7600	0.5858	0.5573	0.5715
jpg	0.2142	0.2735	0.3571	0.3977
meanshift	1.0000	1.0000	1.0000	1.0000

Table 8.6: Average "normalized magnitude" of the block-correlation between gradient vectors for the *original* Einstein image and its degradations for various patch sizes.

a more rigorous approach than our simplistic methods explored above. To conclude this section, we will compute the canonical correlation of the Einstein images. Before doing so, we first present a brief description of the method.

First of all, we flatten the original image patches $x, y \in \mathbb{R}^{M \times M}$ to produce column vectors $x, y \in \mathbb{R}^{M^2}$. The individual elements of x, y are K-vectors. In our application to gradient vectors of images, $K = 2$. We now represent this data by two $M^2 \times K$ matrices, i.e.,

$$\mathbf{A} = \begin{pmatrix} a_{11} & \dots & a_{1K} \\ \vdots & \vdots & \vdots \\ a_{M^2 1} & \dots & a_{M^2 K} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & \dots & b_{1K} \\ \vdots & \vdots & \vdots \\ b_{M^2 1} & \dots & b_{M^2 K} \end{pmatrix}.$$

We shall let $\mathbf{a}_i$ and $\mathbf{b}_i$ denote the i^{th} column vectors of $\mathbf{A}$ and $\mathbf{B}$ respectively, i.e.,

$$\mathbf{a}_i = \begin{pmatrix} a_{1i} \\ \vdots \\ a_{M^2 i} \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} b_{1i} \\ \vdots \\ b_{M^2 i}, \end{pmatrix}, \quad 1 \leq i \leq K.$$

(We acknowledge that this notation may be rather dangerous but hope that it will serve its purpose.) In our application, the column M^2-vectors $\mathbf{a}_1$ and $\mathbf{b}_1$ are composed of the x-components of the gradient vectors of the two images being compared and the column vectors $\mathbf{a}_2$ and $\mathbf{b}_2$ are composed of the y-components of the gradient vectors.

Throughout, we assume that all column vectors $\mathbf{a}_i$ and $\mathbf{b}_i$ are zero-mean. (In our earlier computations, the formula defining regular correlations included the subtraction of the means. This is not the case for the canonical correlation. In any applications, we'll have to manually subtract the appropriate means from our original data.)

We now require the following definitions from linear algebra. The collection of column vectors, $\{\mathbf{a}_i\}_{i=1}^K$, define the column vector space of the matrix $\mathbf{A}$, to be denoted as

"colsp($\mathbf{A}$)", as follows,

$$\text{colsp}(\mathbf{A}) = \text{span}\{\mathbf{a}_1, \cdots, \mathbf{a}_K\},$$

i.e., the set of all linear combinations of the $\mathbf{a}_i$ vectors. Likewise, the column vector space of $\mathbf{B}$ is defined as

$$\text{colsp}(\mathbf{B}) = \text{span}\{\mathbf{b}_1, \cdots, \mathbf{b}_K\}.$$

By definition, colsp($\mathbf{A}$) and colsp($\mathbf{B}$) are subspace of $\mathbb{R}^{M^2}$.

We now come to the main point: The canonical correlation, $C(\mathbf{A}, \mathbf{B})$, between the matrices $\mathbf{A}$ and $\mathbf{B}$ is the maximum correlation that can be found between a vector $\mathbf{u} \in$ colsp($\mathbf{A}$) and a vector $\mathbf{v} \in$ colsp($\mathbf{B}$). One way to state this is as follows,

$$C(\mathbf{A}, \mathbf{B}) = \max_{\mathbf{c}\in\mathbb{R}^K, \mathbf{d}\in\mathbb{R}^K} C\left(\sum_{k=1}^{K} c_k \mathbf{a}_k, \sum_{l=1}^{K} d_l \mathbf{b}_l\right). \tag{8.13}$$

Very fortunately, there is a Matlab routine which can be used to compute the canonical correlation between two matrices $\mathbf{A}$ and $\mathbf{B}$. The file "subspacea.m" can be downloaded from the Matlab file exchange [9].

There is one important note to be made regarding the code. The "subspacea.m" routine tries to find K angles between vectors of both column spaces. Because $K = 2$ in our applications, we will ideally be returned two angles. The first angle is the "best", i.e., the angle associated with the vectors yielding the highest correlation. Sometimes, the program can find only one angle. And in some very extreme cases, no angles are found. In this case, we manually assign the value of 0 to the canonical correlation. Such failures were found, but only in the case of the *jpg* image. We have not investigated this problem in detail, but suspect that it may be due to the flatness of the blocking artifacts in the *jpg* image.

In Table 8.7 below are presented the average blockwise canonical correlations obtained by the above method applied to the *original* Einstein image and its degradation for various patch sizes.

The values in Table 8.7 seem to be more forgiving than their counterparts in Table 8.5, which considered the correlation of magnitudes, and Table 8.6, which considered the "normalized magnitude" of the correlations between x- and y-components. This may be due to the fact that the canonical correlation method is considering the complete gradient vector as opposed to only one of its components, thereby giving more opportunity to find good correlation. Moreover, recall that the canonical correlation method searches for the *best* correlation. One may well wonder if this is really what we want. Indeed, although this is perhaps a more rigorous way to approach computing the correlation between gradients,

	8×8	16×16	32×32	64×64
blur	0.4879	0.5095	0.5459	0.5548
contrast	0.9935	0.9945	0.9989	0.9988
impulse	0.8185	0.6694	0.6512	0.6698
jpg	0.3126	0.3744	0.4690	0.5082
meanshift	1.0000	1.0000	1.0000	1.0000

Table 8.7: Average blockwise canonical correlation values between the *original* Einstein image and its degradations for various patch sizes.

we may prefer the results using one of our more simplistic approaches. Answering this question necessitates the testing of our simplistic methods, as well as the canonical correlation, over a much larger range of distortions. This observation ushers in the arrival of our next chapter, where we graduate to experimentations on the so-called "LIVE Database".

Chapter 9

Incorporating Gradient Correlation Into the SSIM

We now present what should be considered as the culmination of our efforts throughout this thesis: The experiments on the LIVE database. We wish to investigative whether the incorporation of gradient information could, in some way, "improve" the MSSIM. Of course, what is meant by "improve" will have to be discussed, and we do so later in this chapter. It naturally arose, during the completion of this work, to perform a detailed examination of the MSSIM, with particular attention paid to the role of the stability constants. To the best of our knowledge, such a systematic analysis of the MSSIM has not appeared in the literature.

9.1 Introducing the LIVE Database

The Einstein images, which had been greatly useful to us, were no longer sufficient for our exploits. Little could be said, using the Einstein images alone, about the individual differences between the gradient similarity measures surveyed in the previous section. Moreover, our hope now was to improve the SSIM by incorporating gradient information—should our naïve measures, or perhaps the canonical correlation, be up to the task. In order to perform these types of comparisons between measures, we required a large collection of digital images and some indication of their relative quality according to a typical human observer.

Of course, this is not a new need for the larger image quality assessment community. A variety of application-specific and general use databases have been assembled over the

recent years. For our concerns, we naturally referred to a study performed by the Laboratory of Video Engineering (LIVE) at the University of Texas at Austin, in which they obtained quality scores from human participants for hundreds of distorted images. In what follows, we use their "LIVE Image Quality Assessment Database"—in particular, the most recent version, "Release 2"—which is freely available for download at [24].

A brief introduction to the LIVE database is required. Details beyond those summarized here can be found in [23] and [22]. To produce the database, twenty-nine high-resolution 24 bits-per-pixel RGB colour images were collected. These reference images were then distorted using one of five distortion types: JPEG 2000, JPEG, white noise in the RGB components, Gaussian blur, and transmission errors in the JPEG 2000 bit stream using a fast-fading Rayleigh channel model.

Hundreds of degraded images were generated for each of the distortion types listed above. In all cases, the level of distortion was slowly increased such that the resulting collection of degraded images occupied a broad range of visual quality, from imperceptible levels of distortion to high levels of impairment.

Human participants were then shown the distorted images and asked to provide a subjective quality score. (They were not shown the associated reference image for comparison.) It is reported that about 20 to 29 observers rated each image. Participants were asked to rate the quality of the image on a continuous linear scale that was divided into five equal regions marked by the following adjectives: "Bad", "Poor", "Fair", "Good", "Excellent". The images were shown to each participant in a unique randomized order. More details about the experiment, e.g., the lighting and viewing conditions, are reported in [23]. In this way, 982 images were evaluated by human subjects over seven sessions. For reasons that will become clear, the reference images were mixed in with the distorted images and also received a subjective quality rating during each of the seven sessions. As such, 203 of the 982 images in the LIVE database correspond to repeated instances of the 29 reference images. The distribution of these 982 images over the five distortion types is reported in Table 9.1.

JPEG	JPEG 2000	White Noise	Gaussian Blur	Fast-Fading	Total
233	227	174	174	174	982

Table 9.1: The distribution of the 982 total images comprising the LIVE database across the five types of distortions.

The subjective ratings obtained from different observers had to be somehow combined so that each image in the LIVE database could be associated with a single perceptual

quality score. These amalgamated values are known as the "Difference Mean Opinion Scores", or simply DMOS. The DMOS values were computed as follows. Let r_{ij} denote the i^{th} participant's raw score of the j^{th} image. This raw score was then converted to a raw difference score d_{ij} according to the following formula,

$$d_{ij} = r_{i\text{ref}(j)} - r_{ij},$$

where $r_{i\text{ref}(j)}$ denotes the raw score given by the i^{th} participant to the reference image corresponding to the j^{th} distorted image. At this point, outliers were identified according to conditions specified in [23] and removed from the data. The raw difference scores were then converted into Z-scores. The Z-score for the i^{th} subject and the j^{th} image is

$$z_{ij} = \frac{d_{ij} - \overline{d_i}}{\sigma_i},$$

where $\overline{d_i}$ is the mean of all raw difference scores associated with participant i and σ_i is the standard deviation of those scores (both computed after outlier remover). The Z-scores were then averaged over all participants i, yielding a single score $\overline{z_j}$ associated with the j^{th} image.

The mean Z-scores $\overline{z_j}$ were then realigned to produce the DMOS scores. The realignment study was intended to calibrate the quality scales of the seven different sessions. Once more, we refer the reader to [23] for details related to the realignment study. Here we will only state that a linear relationship between the DMOS scores and Z-scores was assumed, and the parameters were obtained according to an error minimization scheme. The range of the DMOS scores is reported in [23] to be 0 to 100, but we find that the DMOS scores repackaged in Release 2 range from -2.64 to 111.77. It is unclear what a negative DMOS score implies about the perceptual quality of the image.

Although these details are rather interesting, the point of this discussion is largely to understand that perceptual quality decreases as DMOS increases. A "perfect" score DMOS $= 0$ does not imply that the corresponding image is an undistorted reference image; rather, it indicates an imperceptible level of distortion (if at all present) in the image. At the same time, due to the test methodology described above, the 203 uncorrupted reference images are not guaranteed to possess a perfect score DMOS $= 0$.

The parameters associated with the distortions, such as the JPEG quality factor Q or the standard deviation σ associated with the white noise, were slowly varied to generate the degraded images. The relationship between changes in these parameters and perceptual quality is unlikely to be linear. (Later in this chapter, we will investigate the relationship between the JPEG quality factor Q and the SSIM.) An even distribution across the range

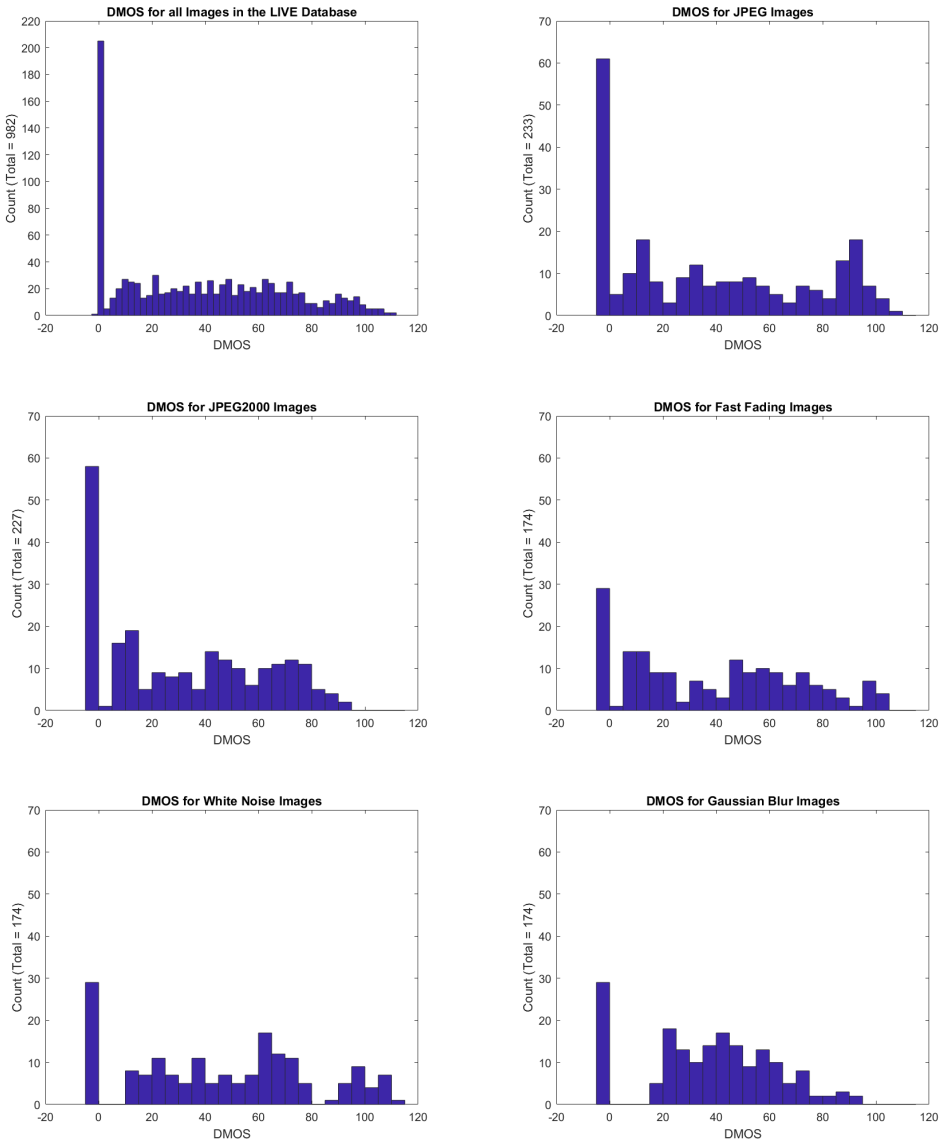


Figure 9.1: Histograms of the DMOS scores by image type.

of perceptual quality is therefore not guaranteed. For this reason, we were interested to see the distribution of the DMOS scores in the LIVE database. In Figure 9.1, we have plotted histograms of the DMOS scores for the entire database and for each individual distortion type. The tall spikes at DMOS = 0 can likely be attributed, in large part, to the reference images. For the white noise and Gaussian blur images, the gap between the

first bar, capturing DMOS near 0, and the second next bar is rather interesting. It appears that small levels of these distortions are immediately more bothersome to the human visual system than the others. Overall, the DMOS scores are well-distributed for all five distortion types.

Figure 9.2, depicting a small set of distorted images from the LIVE database, allows us to quickly peek behind the curtain. We have selected one example image from each of the five distortion types. Although a sample of this size is by no means comprehensive, it does convey the variety in subject and composition among the 29 reference images. The reference images include portraits, photos of animals, nature, and man-made objects, among others; In terms of composition, there are images with an obvious subject anchored in the foreground and images with no obvious focal point.

The representative images in Figure 9.2 were all chosen within a relatively small range of DMOS values. According to the assigned DMOS scores, these images should all have essentially equal perceptual quality. In this way, Figure 9.2 is also meant to demonstrate the difficulty inherent in comparing entirely different images and distortion types on a single scale, whether one is asking this of humans or a mathematical formula.

A final note should be made about how the distortions affect the colour images in Figure 9.2. In particular, high levels of JPEG distortion corrupt the colour channels, yielding greyscale blocks as exhibited in Figure 9.2 (b). This corruption is obvious, and likely very bothersome, to a human's eye. However, as we know, the SSIM was formulated to be computed on greyscale images. The built-in Matlab function "rgb2gray" is typically used to obtain the required greyscale image. This convention is adopted by us in the following experiments. One does wonder if the colour corruptions, like in Figure 9.2, are captured by such a process or if the resulting greyscale image is not as obviously distorted. This question, and an investigation of the best practices for computing the SSIM on colour images, is outside the scope of this thesis.

9.2 Experiments on the LIVE database

9.2.1 Analyzing Different S_4 Formulations and Introducing a First "gradSSIM" Measure

Our first practical experience with the LIVE database was to simply compute the MSSIM for all of its 982 images. The MSSIM [29] of each image was computed using the official code publicly available at [28]. (For the moment, we use the provided code "ssim_index.m"

(a) JPEG 2000

(b) JPEG

(c) Fast Fading

(d) Gaussian Blur

(e) White Noise

Figure 9.2: Example distorted images in the LIVE database. (a) DMOS = 67.8906, (b) DMOS = 63.3076, (c) DMOS = 62.5892, (d) DMOS = 63.3819, (e) DMOS = 67.6265

which does not perform any preprocessing of the images. The suggested downsampling procedure included in "ssim.m" will be explored towards the end of this chapter.) Unlike our simplistic experiments on the Einstein images, the provided code computes the MSSIM in overlapping sliding windows of size 11×11 pixels. When computing the statistics in the local window, a Gaussian-weighted vector is used. In this way, the pixels located centrally in the window contribute more to the local means and variances than those towards the window's border. Further discussion of the Gaussian-weighted vector can be found in [32]. Unless otherwise indicated, we will use a Gaussian-weighted vector in all of our computations on the LIVE database.

Unlike our earlier experiments, the official MSSIM we now compute includes stability constants in both the numerator and denominator of its components. We used the suggested values, as reported in [32]: $C_1 = (K_1 \times L)^2$, $C_2 = (K_2 \times L)^2$, and $C_3 = C_2/2$, where $K_1 = 0.01$ and $K_2 = 0.03$. The greyscale images produced by "rgb2gray" have dynamic range $L = 255$, yielding $C_1 = 6.5$, $C_2 = 58.5$ and $C_3 = 29.2$ when rounded to one decimal place.

Interested to see if we could corroborate the findings of our naïve Einstein experiments, we next computed the individual components of the SSIM on the LIVE database. In each component, we used the appropriate stability constant listed above (in both the numerator and denominator) and an 11×11 Gaussian-weighted vector when computing the local statistics.

Finally, we also computed one of our gradient similarity measures. Our first choice was to compute the normalized magnitudes of the correlation between components, i.e.,

$$S_4 = \frac{1}{\sqrt{2}}(a^2 + b^2)^{1/2}, \tag{9.1}$$

where a denotes the correlation between the x-components of the gradients and b denotes the correlation between the y-components of the gradients. Our work with the Einstein images suggested that this is one of the more promising of our simplistic methods. In the spirit of the MSSIM, we included a stability constant, denoted C_4, in both the numerator and denominator of the correlations a and b. Moreover, because we think of these correlations as gradient-based analogues of the S_3 component, we made C_4 (practically) equal to C_3, i.e., we let $C_4 = 30$. This reasoning is justified by our observation that the gradient correlations were similar in magnitude to the regular correlations, at least for the Einstein images.

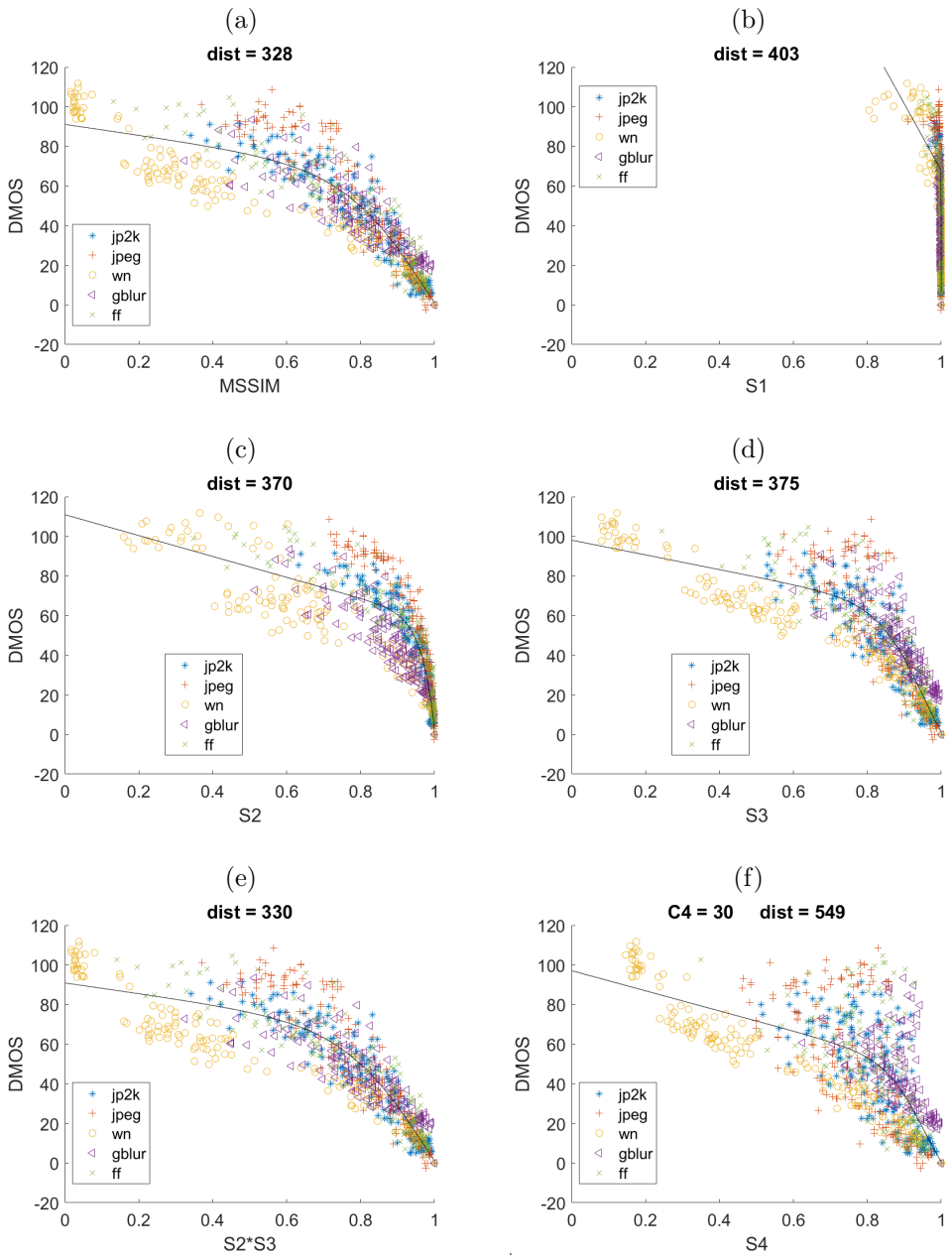


Figure 9.3: Plots illustrating the performance of the MSSIM, its individual components, and our "normalized magnitude" S_4 measure on the LIVE database. The points have been colour-coded according to degradation type: In the legend, "jp2k" indicates JPEG 2000 distortions, "jpeg" for JPEG, "wn" for white noise, "gblur" for Gaussian blur, and "ff" for fast-fading. The "dist" measure indicates the variance in the data according to the curve of best fit.

The results for this first set of experiments are displayed in Figure 9.3. The convention in the literature, which we have adopted here (at least for the moment), is to plot the DMOS scores along the vertical axis. Included in each of the plots in Figure 9.3 is a curve of best fit obtained using nonlinear regression performed by the Matlab function "fitnlm". We use the following nonlinear relationship given in [23] for the regression,

$$\begin{aligned} \text{Quality}(x) &= \beta_1 \text{logistic}(\beta_2, (x - \beta_3)) + \beta_4 x + \beta_5 \\ \text{logistic}(\tau, x) &= \frac{1}{2} - \frac{1}{1 + \exp(\tau x)}. \end{aligned} \tag{9.2}$$

To be able to evaluate the performance of the different measures, we compute the RMSE between the DMOS and the algorithm score after nonlinear regression. (This is one of a few statistical evaluation techniques explored in [23].) For example, consider the plot of MSSIM scores in Figure 9.3 (a). For the j^{th} image in the LIVE database, we compute

$$\text{dist} = \left[\sum_{j=1}^{982} (\text{DMOS}(j) - \text{Quality}(\text{MSSIM}(j)))^2 \right]^{1/2}. \tag{9.3}$$

In Figure 9.3, this value is provided in the "dist" measure displayed in the title of each plot. Because the curve is obtained through least squares regression, our "dist" measure can be thought of as the variance between the fitted curve and the DMOS values as a function of the algorithm score. From the plot of the MSSIM and its regression curve in Figure 9.3, we see that the data points are quite concentrated about the curve at the bottom right region of the plot, which may be considered as the "low DMOS" region (i.e., under 20) or "high MSSIM" region (i.e., near 1). As we move leftward and upward, however, the data points are distributed more diffusely about the regression curve. This led us to think of one possible criterion for "improving" MSSIM, namely, decreasing the diffusiveness of the data points in the lower DMOS region.

The plots of the MSSIM and its individual components in Figure 9.3 reflect our earlier conclusions made using the Einstein images. First looking at Figure 9.3 (b), there is an incredible amount of compression along the S_1 axis towards $S_1 = 1$. This illustrates once again our earlier conclusion that the S_1 term differentiates little between distorted images. Conversely, the S_2 and S_3 components are both performing well, with S_2 exhibiting a slightly lower "dist" value than S_3. Figure 9.3 (e) shows the result of computing $S_2 \times S_3$ in the local sliding windows. Unsurprisingly, the $S_2 \times S_3$ plot and the MSSIM plot are visually indistinguishable from one another, an attribute which is also reflected in their nearly equal "dist" values.

Figure 9.3 (f) illustrates the performance of our first attempted S_4 measure as defined in Eq. (9.1). This plot is, quite plainly, disappointing. Overall, the S_4 plots looks like a more diffuse version of the S_3 one pictured above. In particular, the clump of points ballooning above the fitted curve between $S_4 = 0.6$ and $S_4 = 0.9$ is rather disturbing. Moreover, both this bothersome clump and the yellow circles (denoting white noise images) near $S_4 = 0.2$ appear to be slipping down the fitted curve, cascading towards $S_4 = 1.0$. The obvious culprit responsible for overinflating the scores is the stability constant, which appears in both the numerator and denominator of the gradient correlations. It appears that the value $C_4 = 30$ is too high, artificially pushing the quotients a and b towards 1.

As mentioned earlier, this value of C_4 was chosen to agree with the MSSIM's stability constant C_3. However, during our research, we were unable to find in the literature any clear description of how or why this particular C_3 value was chosen. In fact, the suggested values of all three stability constants C_1, C_2, and C_3 were much larger than we had expected. In [32], the stability constants are described as "small" constants to protect against instabilities in the denominator; although "small" and "large" can only be understood as relative terms, their raison d'être could be satisfied by significantly smaller values. For our second experiment, we were thus motivated to vary the value of C_4, in the hope that a lower value would yield a significant increase in performance.

We were also eager at this time to make a first attempt a modifying the SSIM using gradient information. We considered this following very simple formulation, which we called "gradSSIM" to indicate the addition of gradients. For two image patches x and y, we define the following local similarity measure,

$$\text{gradSSIM}(x, y) = \text{SSIM}(x, y) \cdot S_4(x, y) = S_1(x, y) \cdot S_2(x, y) \cdot S_3(x, y) \cdot S_4(x, y). \qquad (9.4)$$

As is done in the MSSIM, we obtain a global gradSSIM by taking the mean of all the local values in Eq. (9.4). Because $S_4(x, y) \leq 1$, gradSSIM $\leq$ MSSIM. It was not lost on us that this formulation may be overly punitive. However, a test run of Eq. (9.4) on the LIVE database was required to be certain.

The results of applying our gradSSIM measure on the LIVE database, for various choices of C_4, are depicted in Figure 9.4. In all cases, the stability constant C_4 appears in the numerator and denominator of the gradient correlations a and b. The stability constants C_1, C_2, C_3 remain unchanged from their suggested values listed previously.

Unfortunately, the gradSSIM is performing noticeably worse than the MSSIM for all five C_4 values shown in Figure 9.4. For the moderate C_4 values considered, i.e., $C_4 = 1, 10, 30, 50$, the corresponding sequence of $\text{dist}_{\text{gradSSIM}}$ values fluctuates in the low-400s. These "dist" values are significantly elevated compared to $\text{dist}_{\text{SSIM}} = 328$.

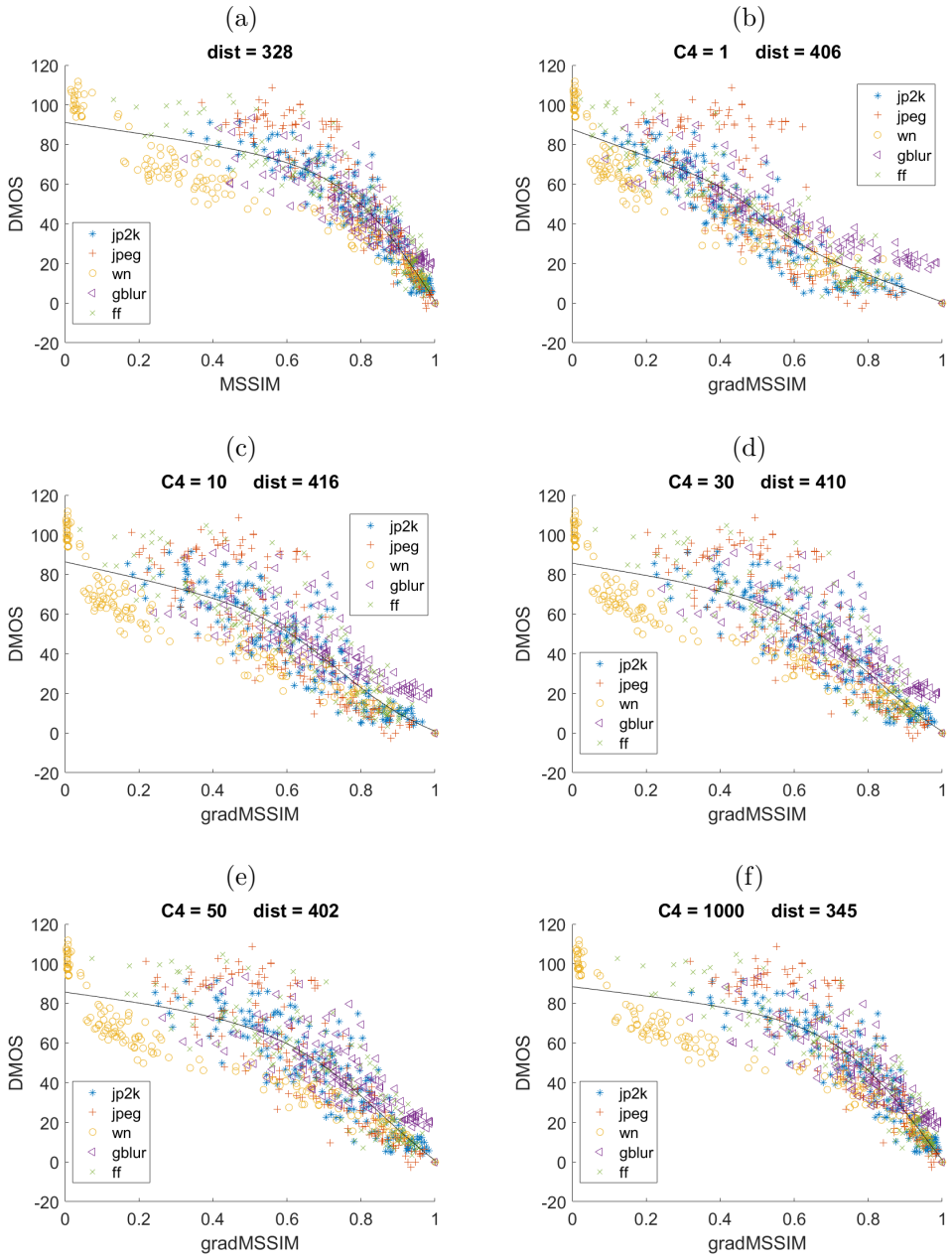


Figure 9.4: Plots of the MSSIM and our first simple "gradSSIM" measure for various choices of C_4. None of the gradSSIM are performing as well as the MSSIM. For $C_4 = 1000$, the gradSSIM plot is visually close (but not yet equal) to that of the MSSIM.

Reflected in all plots is the punitive nature of Eq. (9.4). The MSSIM plot in Figure 9.4 (a) is characterized by the increasingly tighter concentration of points narrowing towards the "tip" at MSSIM = 1. On the other hand, the gradSSIM plots appear much more diffuse. This observation is consistent with our impression of the S_4 plot in Figure 9.3 (f). Moreover, there are significantly more images with gradSSIM $\in [0, \frac{1}{2}]$, i.e., in the lower half of possible scores, compared to the MSSIM plot.

Indeed, that bothersome clump of points is still ballooning atop the fitted curve in the gradSSIM with $C_4 = 30$, although there is more spread between points due to the influence of the SSIM's components. By comparison, the gradSSIM with $C_4 = 1$ deals with most of these points quite nicely, sliding many of them back towards gradSSIM = 0. Figure 9.4 (a) thus achieves a nearly linear relationship. Unfortunately, the red crosses (denoting JPEG images) in this clump, roughly between DMOS = 80 to DMOS = 100, are the exception: They shift little between all six plots in Figure 9.4, appearing to be almost invariant to changes in C_4.

For sufficiently large C_4, $S_4 \approx 1$. Hence, we expect $\text{dist}_{\text{gradSSIM}} \to \text{dist}_{\text{SSIM}}$ as the gradSSIM function approaches the SSIM function. Indeed, the gradSSIM plot with $C_4 = 1000$ is visually similar to the MSSIM plot; However, there are still visible differences. It may be surprising that this C_4 value is not large enough to yield a closer agreement in their "dist" values. This disparity indicates that the quantities involved in the S_4 computation can be very large.

Overall, Figure 9.4 does invite criticism of the punitive nature of the gradSSIM. This admission aside, we are still first and foremost concerned with the stability constant C_4. None of the configurations just explored are entirely satisfying. To continue our investigation of C_4, we returned to our initial instinct during the Einstein experiments: removing C_4 from the numerator and using a value of $C_4 \ll 1$. For the moment, we step away from the gradSSIM to see the effect of this choice on the S_4 term alone.

This idea led to the production of six new plots, the presentation of which is split over Figure 9.5 and Figure 9.6. In the left columns of Figure 9.5 and Figure 9.6 are shown the result of including the indicated C_4 value in both the numerator and denominator. The right columns show the result of having the corresponding C_4 in the denominator only.

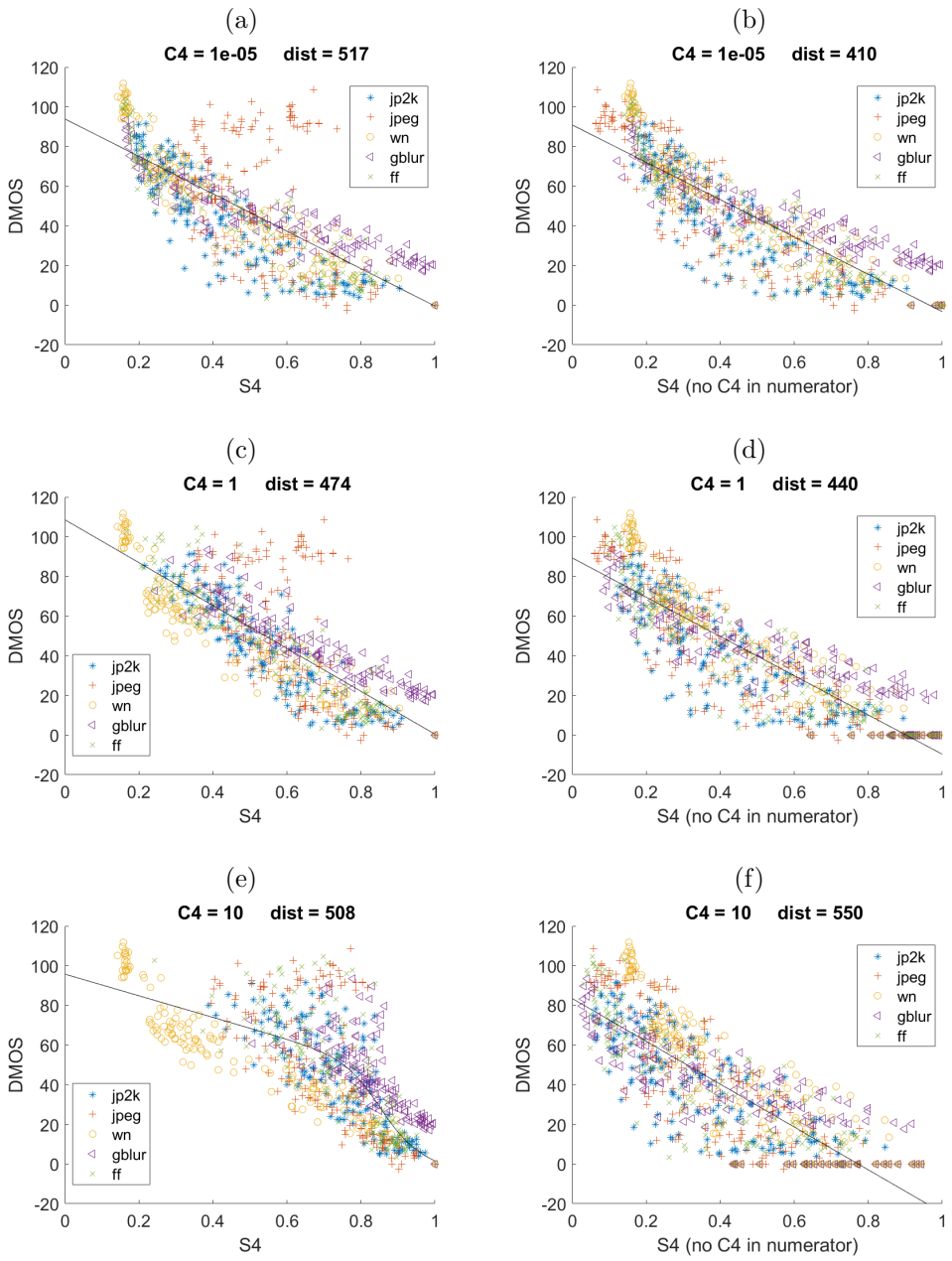


Figure 9.5: Results of varying C_4 in our "normalized magnitude" S_4. Plots in the left column correspond to C_4 in both the numerator and denominator; The right column corresponds to C_4 in the denominator only.

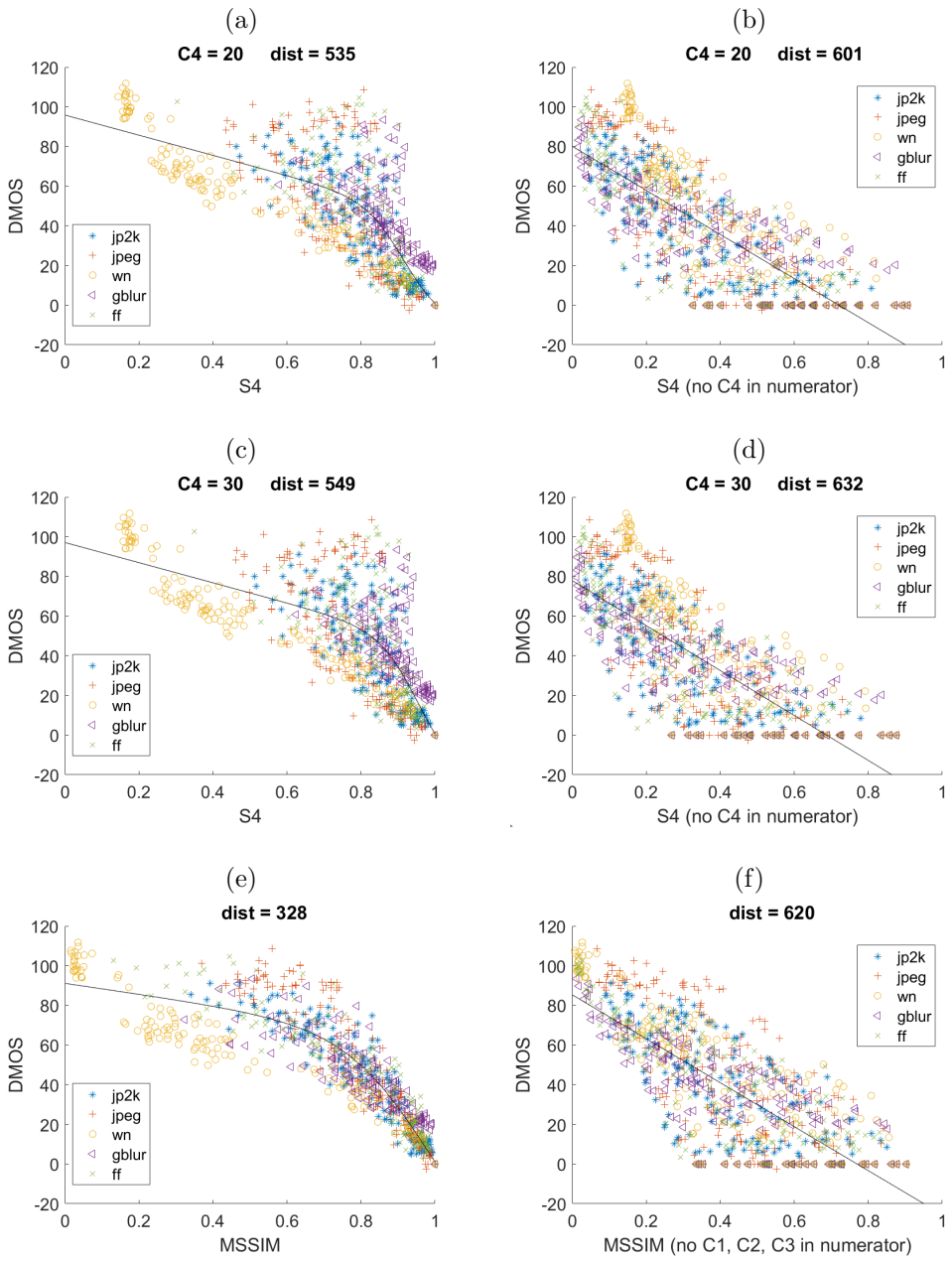


Figure 9.6: Results of varying C_4 in our "normalized magnitude" S_4. Plots in the left column correspond to C_4 in both the numerator and denominator; The right column corresponds to C_4 in the denominator only. The MSSIM has also been plotted for comparison.

Once again, that invariant group of JPEG images persists throughout the plots in the left columns of Figures 9.5 and 9.6. The placement of these points is particularly attention grabbing in Figure 9.5 (a), in which they really stand out from the crowd. Conversely, in Figure 9.5 (b), which corresponds to having $C_4 \ll 1$ in the denominator only, these problematic crosses have been nicely swept towards $S_4 \in [0, 0.2]$. This placement is in much better agreement with their DMOS values and the fitted curve. We also like the overall shape of Figure 9.5 (b): Although it loses the sharp "tip" of the MSSIM, the points are reasonably close to the fitted curve along its entire length.

Before moving on, there is one more worthwhile observation to make about the plots in the right-hand columns of Figures 9.5 and 9.6. Note that, as C_4 increases, there is a growing "trail" of points at DMOS $= 0$. This phenomenon is also exhibited in the MSSIM when the stability constants are removed from the numerator (see Figure 9.6 (f)). We believe this is likely due to instabilities in the numerator, i.e., a situation where $s_{DxDy} \approx 0$ and $\|s_{Dx}\|\|s_{Dy}\| \approx 0$.

Going forward, we are tempted to adopt the S_4 having $C_4 = 10^{-5}$ in the denominator only. This is, however, a large departure from the conventions set by the MSSIM. We performed a couple of tasks to validate our preferred S_4. Having noted that our preferred S_4 in Figure 9.5 (b) deals quite nicely with those problematic JPEG images, we were interested to take a closer look at the relationship between JPEG compression and the S_4 scores.

For our experiment, we saved many degraded instances of the well-known "Boat" image by slowly increasing the level of JPEG compression. We obtained a similar sequence of increasingly degraded "Boat" images using the JPEG 2000 compression method as well. For both JPEG and JPEG 2000 distortions, we were then able to investigate the relationship between the S_4 scores and image quality. The resulting plots are shown in Figure 9.7. In all cases, the horizontal axis indicates "bpp", i.e., "bits-per-pixel". Here we simply state that the bits-per-pixel is a measure of compression in the image. The pure reference "Boat" image is 8 bits-per-pixel; As the degree of compression increases, the bits-per-pixel decreases. (The determination of these bits-per-pixel values represents a significant amount of work, but we won't discuss it here.)

In Figure 9.7 (c), we plot the relationship between our preferred S_4 and the bits-per-pixel for both JPEG images (see the blue curve) and JPEG 2000 images (see the red curve). Figure 9.7 (b) considers instead the S_4 with $C_4 = 30$ in the numerator and denominator. For comparison, the SSIM is used in Figure 9.7 (a). Our preferred S_4 penalizes both JPEG and JPEG 2000 compression much more harshly than the SSIM. A visual examination of the degraded "Boat" images revealed that the heavily-compressed

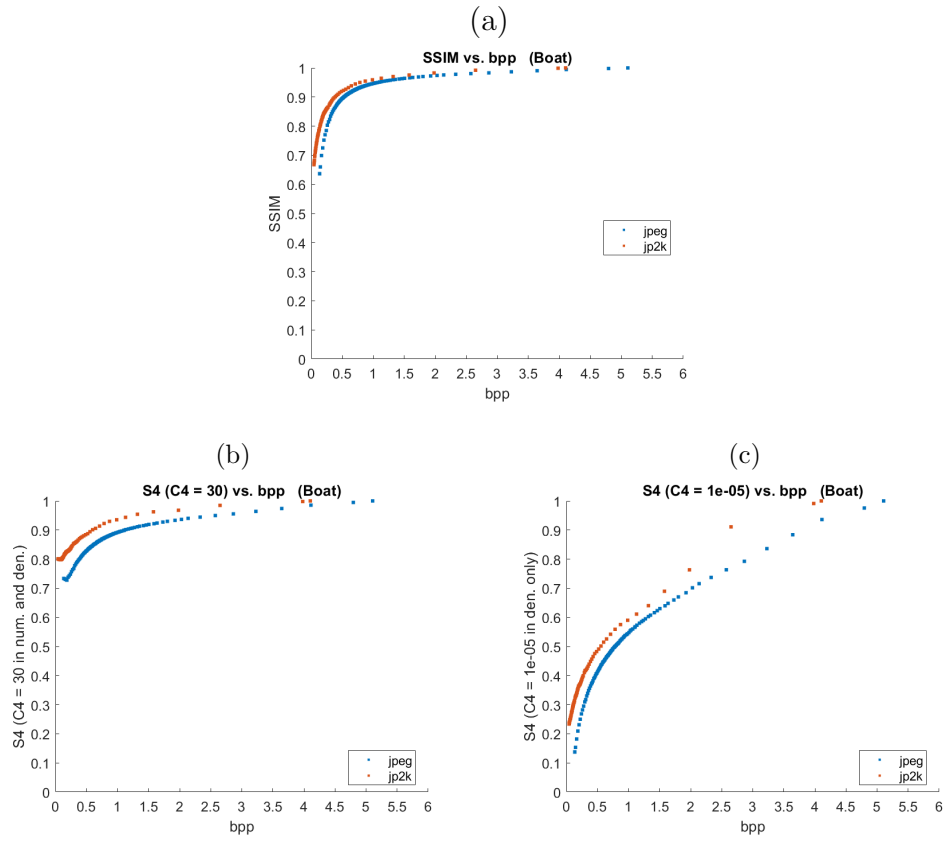


Figure 9.7: SSIM and S_4 vs. bits-per-pixel for JPEG and JPEG 2000 compressed "Boat".

images, i.e., having bpp less than 0.5, were very degraded. We felt that the SSIM in this lower range was undeservedly elevated, while the low scores of our preferred S_4 were arguably more reasonable. This preference is also in agreement with our earlier appreciation of the manner in which this S_4 handles those problematic JPEG crosses in Figure 9.5 (b). On the other hand, the treatment of these distortion types by the S_4 with $C_4 = 30$ is very similar to, although even worse than, that of the SSIM. Overall, this experiment did justify the choice of our preferred S_4 over the S_4 with $C_4 = 30$ in the numerator and denominator.

Seeking further justification of our preferred S_4, we also returned to the other formulations of gradient similarity explored using the Einstein images. In particular, we will now revisit $\overline{\cos\theta}$ (see Eq. (8.6)), $\cos\overline{\theta}$ (see Eq. (8.8)), and the canonical correlation (see Eq. (8.13)), all discussed in the previous section.Figure 9.8 depicts the relationship between the different gradient similarity measures as they perform on the LIVE database. The plots

reveal the degree of correlation between two different S_4 measures, as indicated in the axis titles. In all cases, the horizontal axis corresponds to our preferred S_4. Figure 9.8 (a), which considers the canonical correlation against our preferred S_4, is particularly heartening. These two methods are evidently highly correlated. Recalling that the canonical correlation is arguably a more rigorous approach, we consider Figure 9.8 (a) to be a big endorsement of our preferred S_4. It is also impressive that our efficient and simple-minded S_4 is able to produce comparable results to the computationally-costly canonical correlation.

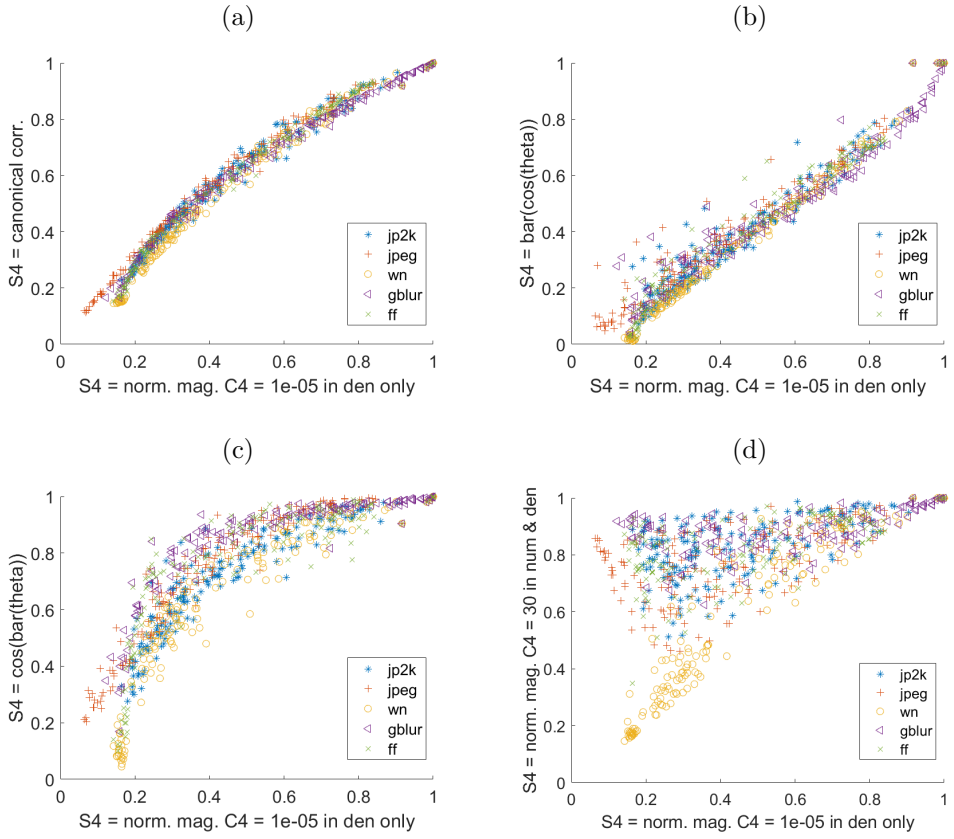


Figure 9.8: Comparison of the different S_4 measures.

Also exhibiting a nearly linear relationship with our preferred S_4 is $\overline{\cos\theta}$, plotted in Figure 9.8 (b). It is interesting to note the sparse, secondary diagonal line which has formed above the primary linear relationship existing in the plot. On the other hand, Figure 9.8 (d) considers our initial choice, S_4 with $C_4 = 30$ in the numerator and denominator, against our preferred S_4. As one might expect from our previous experiments, these two formulations show little correlation.

9.2.2 Investigating the Stability Constants in the MSSIM and Presenting an Improved gradSSIM

The experiments surveyed in the previous section culminated in the determination of our preferred S_4 measure. We had devoted much effort to justify its breaking of tradition set by the MSSIM. Having reached a comfortable level of familiarity with the LIVE database, and after many hours spent pored over plots, we were emboldened at this point in our explorations to take yet another step away from convention.

Going forward, we will flip the orientation of the axes such that the DMOS scores are plotted along the horizontal axis. Because the DMOS scores are obtained from the subjective experiments, they are, or at least could be considered to be, the independent values. When developing a new quality measure, we are seeking to map the subject assessment of an image, i.e., its DMOS value, into a number, e.g., an S_4 value.

Given this understanding, it is actually the scatter in the algorithm scores relative to the curve that characterizes goodness of fit. In order to compute this new "dist" value, we first have to obtain a best fit curve for the flipped data.

The nonlinearity provided in [23] and written in Eq. (9.2) is a transcendental expression and cannot be inverted. One could drop the linear term to obtain an invertible expression, but the resulting inverse function exhibits undesirable behaviour for our application. It is unclear how the authors of [23] came to expect the complicated relationship in Eq. (9.2). A simple polynomial fits the flipped data well. In particular, we will proceed using a quadratic function for the fitted curve.

As discussed, our previous work had also led us to question how the stability constants in the SSIM were chosen. To attempt to answer this question ourselves, we decided to vary the stability constants in the SSIM. In keeping with our new rebellious spirit, we were cautiously wondering, however unlikely, whether the MSSIM could be "improved" by simply changing the value of these constants. Based on our previous work, we were particularly interested to attempt the case with $C_1, C_2, C_3 \ll 1$ in the denominator only.

The relevant plots are shown in Figure 9.9. (Many other values of the stability constant were considered, but have not been included here.) Figure 9.9 (a) considers the case $C_1, C_2, C_3 \ll 1$ in the denominator only. In Figure 9.9 (b)-(f), all three stability constants appear in both the numerator and denominator. In these cases, $C_1 = 6.5$ remains unchanged from its suggested value and $C_2 = 2C_3$ for the value of C_3 indicated by the title. The "distDMOS" value, also displayed in the title of each plot, indicates the variance of the flipped data. The flipped distance has been renamed "distDMOS" in order to distinguish it from the "dist" value computed previously.

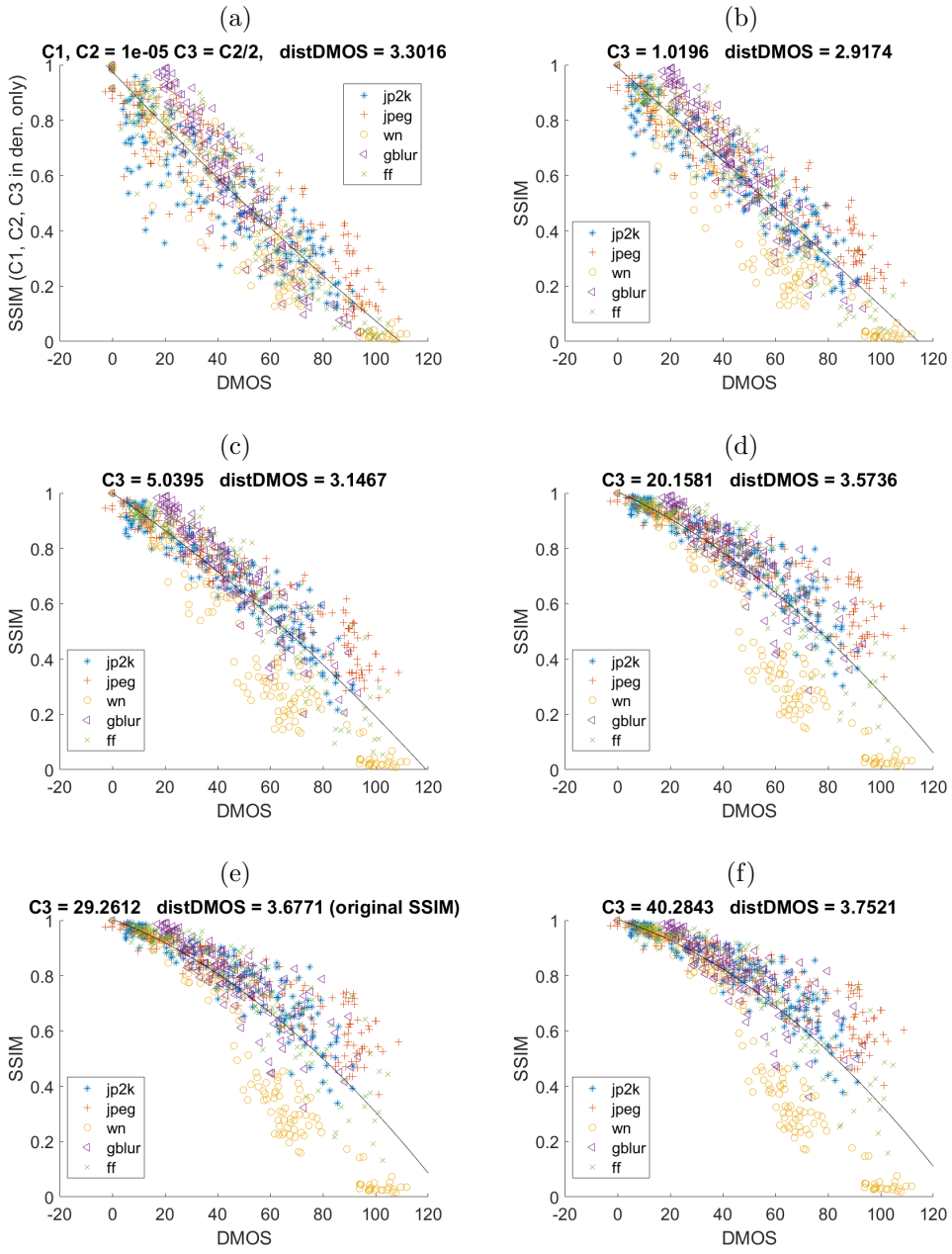


Figure 9.9: Vary the stability constants C_3 and $C_2 = 2C_3$ in the SSIM. The value of $C_1 = 6.5$ is kept constant.

For Figure 9.9 (b)-(f), where the stability constants appear in both the numerator and denominator, increasing C_2 and C_3 appears to sharpen the tip at DMOS = 0. At the same time, the diffusiveness in the high-DMOS region worsens as the fit in the low-DMOS region improves. Based on these observations, we suspect that the suggested values of the MSSIM stability constants were to chosen to yield a good fit for low-DMOS images. It does not appear to be advantageous to increase C_3 past $C_3 = 30$. According to Figure 9.9 (f), there is even more bulging in the high-DMOS region when C_3 is increased past its suggested value.

On the other hand, Figure 9.9 (b) has a uniform spread relative to the best fit curve across its entire length. The bulging for high-DMOS images is better controlled, but at the expense of the great fit in the low-DMOS region. Figure 9.9 (b) also features the lowest "distDMOS" value. For the cases we considered, "distDMOS" continues to decrease as C_3 decreases toward $C_3 = 1$. Interestingly, there is no "critical value" of C_3 (and C_2) at which this distance increases, which could have suggested an "optimal" value for the stability constants.

Finally, Figure 9.9 (a) considers the case $C_1, C_2, C_3 \ll 1$ in the denominator only. This plots looks like a more diffuse, and ultimately worse, version of Figure 9.9 (b). Indeed, the "distDMOS" is also elevated compared to the value in Figure 9.9 (b).

After these experiments, we were able to offer reasonable speculation on the reasoning underlying the choice of stability constants in the MSSIM. At the same time, we liked the plot in Figure 9.9 where $C_3 = 1$. To try and understand which of these two formulations was working better, we undertook our own detailed subjective evaluation of the images in the LIVE database. At this point, it must be mentioned that the subjective evaluation described over the next few paragraphs could be viewed as a kind of "mini-version" of the evaluation performed by the inventors of the Structural Similarity index in order to come up with "reasonable" values of its stability constants. We are making use of subjective assessments, i.e., the DMOS scores, over an image database, i.e., the LIVE database, to come up with a "reasonable" functional form of our S_4 as well as its stability constant.

For our evaluation, we identified various groups of points whose placement in Figure 9.9 (b) and (e) interested us. We then visually inspected the corresponding distorted images, as well as their reference images, to determine which treatment of the points was more "fair".

Over time, the subjective evaluation grew to be a very large set of explorations. We explored not only many groups of images, but also metrics not discussed here, e.g., our S_4 measure and the RMSE. Below, we will provide only a very brief summary of our explorations.

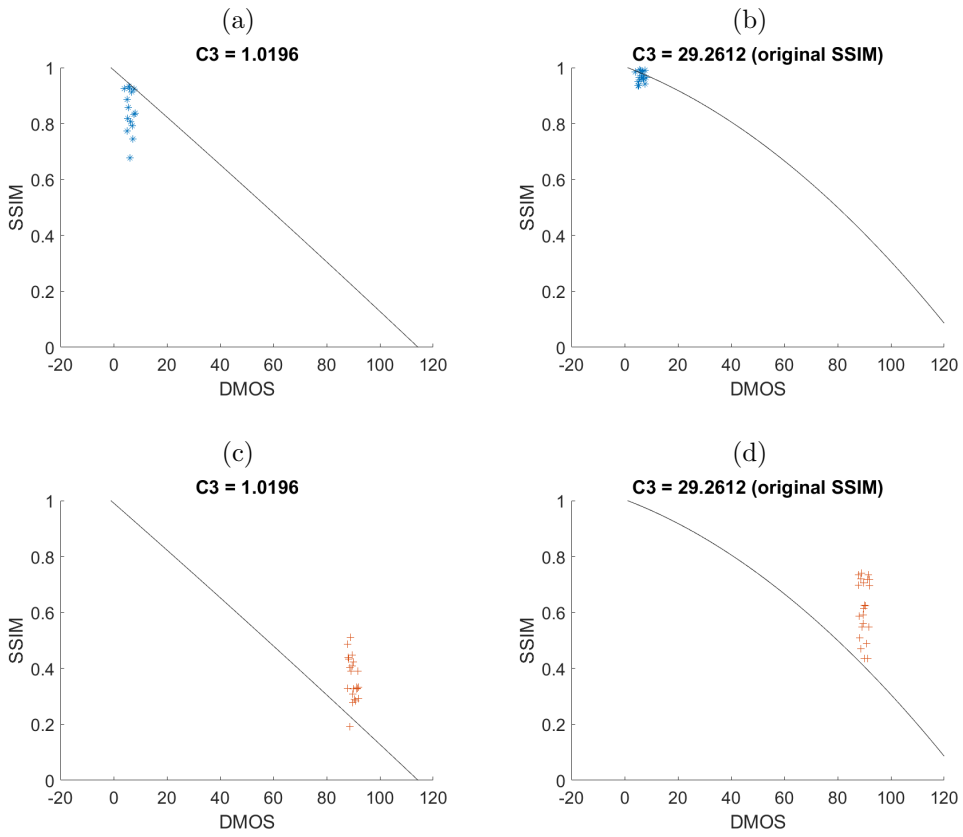


Figure 9.10: Subjective Evaluation.

A few examples of the investigated groups are provided in Figure 9.10. For example, Figure 9.10 (a) and (b) show the same group of JPEG 2000 distorted images, but their MSSIM values are shifted depending on the chosen constant. We were interested to see if these images were all distorted to a similar degree, as indicated by the default MSSIM, or if they covered a wider range of perceptual qualities, as suggested by the MSSIM with $C_3 = 1$. Ultimately, we concluded that these images were all of very high visual quality (as one would expect from their placement near DMOS = 0). We could not discern any visible evidence to justify the spread of scores in Figure 9.10 (a).

We also looked at the JPEG compressed images in Figure 9.9 (c) and (d). Our goal here was to assess which of the default MSSIM scores or the harsher scores of the modified MSSIM were more "fair". We found these images to be heavily distorted, such that it could be argued that the placement in Figure 9.9 (c) was more appropriate.

Our evaluation did reiterate many issues with which we had already been wrestling. Firstly, we found it hard to justify the placement of such vastly different images occupying an exceptionally large range of visual quality on a single quality scale. When faced with an image and asking ourselves, "Is this score fair?" we could sometimes only shrug and conclude, "Maybe?". Moreover, when a preference between regimes was clearer, it was not consistent. When something was working well in one regime (e.g., the concentration in the low-DMOS region for the unmodified MSSIM), something else was going awry (e.g., the scatter in the high-DMOS region of the unmodified MSSIM). We began to wonder if it was possible to expect a single formula to properly handle all distortions and all degrees of visual quality.

Be that as it may, we put those concerns aside for a moment to revisit our gradSSIM. For our second attempt at a gradSSIM measure, we maintained our simple definition, i.e,

$$\text{gradSSIM}(x, y) = \text{SSIM}(x, y) \cdot S_4(x, y),$$

but this time using our preferred S_4 with $C_4 = 10^{-5}$ in the denominator only. The results are provided in Figure 9.11 (b). Figure 9.11 (a) also provides the MSSIM for comparison.

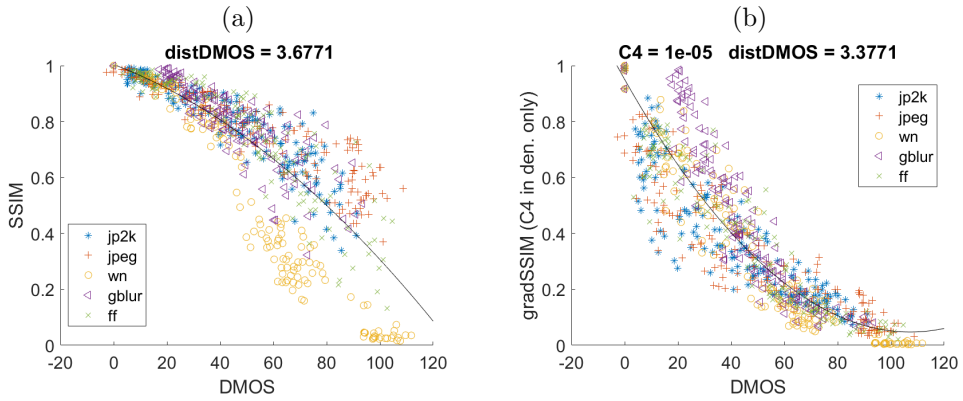


Figure 9.11: Our second attempt at a gradSSIM, which uses our preferred S_4 with $C_4 = 10^{-5}$ in the denominator only. The MSSIM is shown for comparison.

As we had hoped to do, this version of the gradSSIM has a much improved fit in the high-DMOS region compared to the MSSIM. Aligning with our earlier impressions, this success is balanced by a loss in the low-DMOS region, where the gradSSIM is exhibiting its diffuseness. The overall spread in the Figure 9.11 (b) appears to be slightly less than Figure 9.11 (a). This observation is also reflected by the gradSSIM's lower "distDMOS" value.

Ultimately, this gradSSIM offers a better choice for applications dealing with heavily degraded images. Due to the overall lesser scatter, one may also be tempted to claim that it offers a better choice for applications concerning a wide range of qualities. However, our subjective evaluation revealed that the high-DMOS images are extremely degraded. Based on our study, it would be reasonable to assume that the low-to-mid DMOS region captures reasonable distortion levels for most practical applications. In this light, the spread in the low-DMOS region of the gradSSIM is probably much more problematic that the high-DMOS spread of the MSSIM. In other words, a gain in the high-DMOS region may not be worthwhile if it comes at the expense, as it does in this case, of the good fit for low-DMOS. These observations led us to wonder if even better results can be achieved using a different approach.

9.2.3 Blended Image Quality Measures

For some time, we had been questioning the validity of demanding that a single mathematical formula should accommodate all types of distortions across the entire quality spectrum. We often speculated that the human visual system may access a variety of processes when judging different aspects of an image. This thought led us to consider "blending" multiple formulas in a manner that might be more aligned with how the human visual system aggregates information. For example, one could combine the values from multiple measures by taking the maximum, minimum, or a convex combination.

We also considered the following "blended" approach. Recall that the MSSIM performs well in the low-DMOS (high-MSSIM) range, while the gradSSIM performs well in the high-DMOS (low-gradSSIM) range. In practice, we don't have access to the DMOS score of a given image. We can, however, make some inferences based on the MSSIM score, which can be easily computed. If the MSSIM score is high, we expect the MSSIM to accurately predict the image quality. If the MSSIM score is low, we expect the gradSSIM to perform better. Using this idea, we formulate the following piecewise-defined function based on the MSSIM score: For a given threshold $T \in [0, 1]$, define

$$\text{HybridSSIM} = \begin{cases} \text{MSSIM}, & \text{if MSSIM} \geq T \\ \text{gradSSIM}, & \text{if MSSIM} < T. \end{cases} \tag{9.5}$$

If the threshold T made is appropriately close to 1, Eq. (9.5) chooses the score that we expect to be more indicative of the quality of a given image. (Note that this choice occurs on the global level, not in the local image patches.) This approach is in the spirit of

asymptotic analysis, where one has two solutions, both accurate in an isolated region, and attempts to combine or "stitch" them together to yield a single solution working across the entire range of values.

The result of using Eq. (9.5) for various thresholds T is provided in Figure 9.12. Varying the parameter T determines the transition point between regimes. Unfortunately, the plots for all values of T look strange due to the large gaps between regimes. Perhaps the highest threshold $T = 0.85$ in Figure 9.12 (f) looks the best, and it does exhibit the lowest "distDMOS". Overall, one would prefer for the points to be positioned such that a seamless transition is achieved.

We attempted the following to reduce the gaps between regimes in Figure 9.12. Instead of using the gradSSIM, one could instead simply use the S_4 measure in its place, i.e.,

$$\text{HybridSSIM} = \begin{cases} \text{MSSIM}, & \text{if MSSIM} \geq T \\ S_4, & \text{if MSSIM} < T. \end{cases} \tag{9.6}$$

By omitting the SSIM components contained in the gradSSIM, the S_4 measure on its own is less punitive. This should reduce the gap between its values and those of the MSSIM.

The result of using Eq. (9.6) on the LIVE database is shown in Figure 9.13. The threshold $T = 0.85$ shown in Figure 9.13 (f) once again appears to working the best. The "distDMOS" is reduced compared to Figure 9.12 (f), so some improvement has been made. Although the gap between regimes is reduced, the transition is still not seamless. In particular, the horizontal line produced by "cutting" the MSSIM is very obvious.

For our final attempt, we use the modified SSIM with $C_3 = 1$, i.e.,

$$\text{HybridSSIM} = \begin{cases} \text{MSSIM}, & \text{if MSSIM} \geq T \\ \text{modified MSSIM}, & \text{if MSSIM} < T. \end{cases} \tag{9.7}$$

While the modified SSIM does not exhibit a particularly good fit in the high-DMOS region, it does occupy a uniform spread across the entire curve of best fit. The plots produced using Eq. (9.7) are shown in Figure 9.14. As before, the highest attempted threshold $T = 0.85$ in Figure 9.14 (f) appears to be working the best. While this "distDMOS" is the lowest yet, there is still an unsightly gap between regimes.

Although our piecewise definitions such as Eq. (9.5) are well-motivated, in practice cutting off the MSSIM at T produces a stark horizontal line which is rather unappealing. After some reflection, we were able to develop another "blended" image quality measure. By inferring a preference between measures as before, an appropriate weighting between

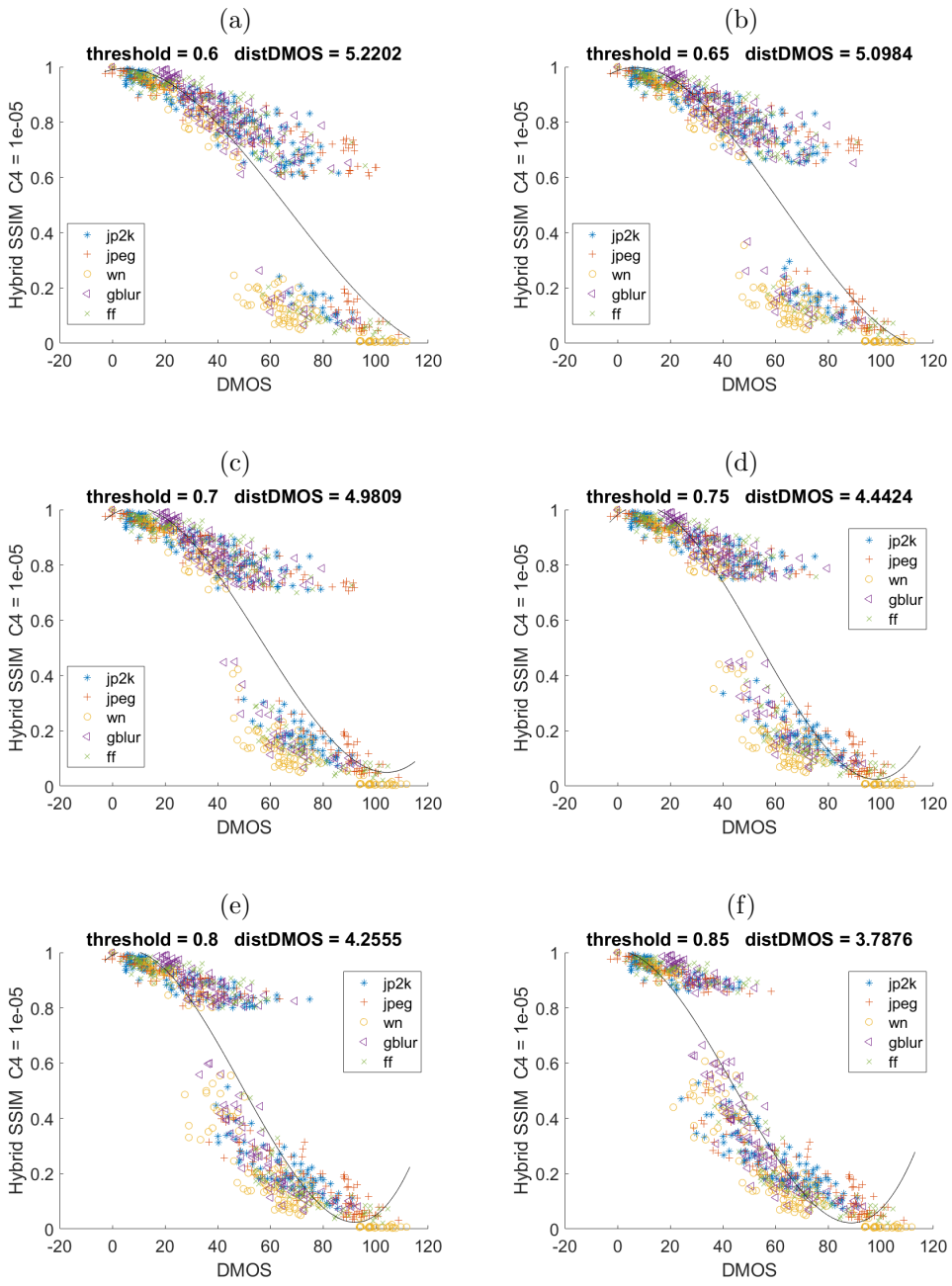


Figure 9.12: A "HybridSSIM" using the gradSSIM as defined in Eq. (9.5).

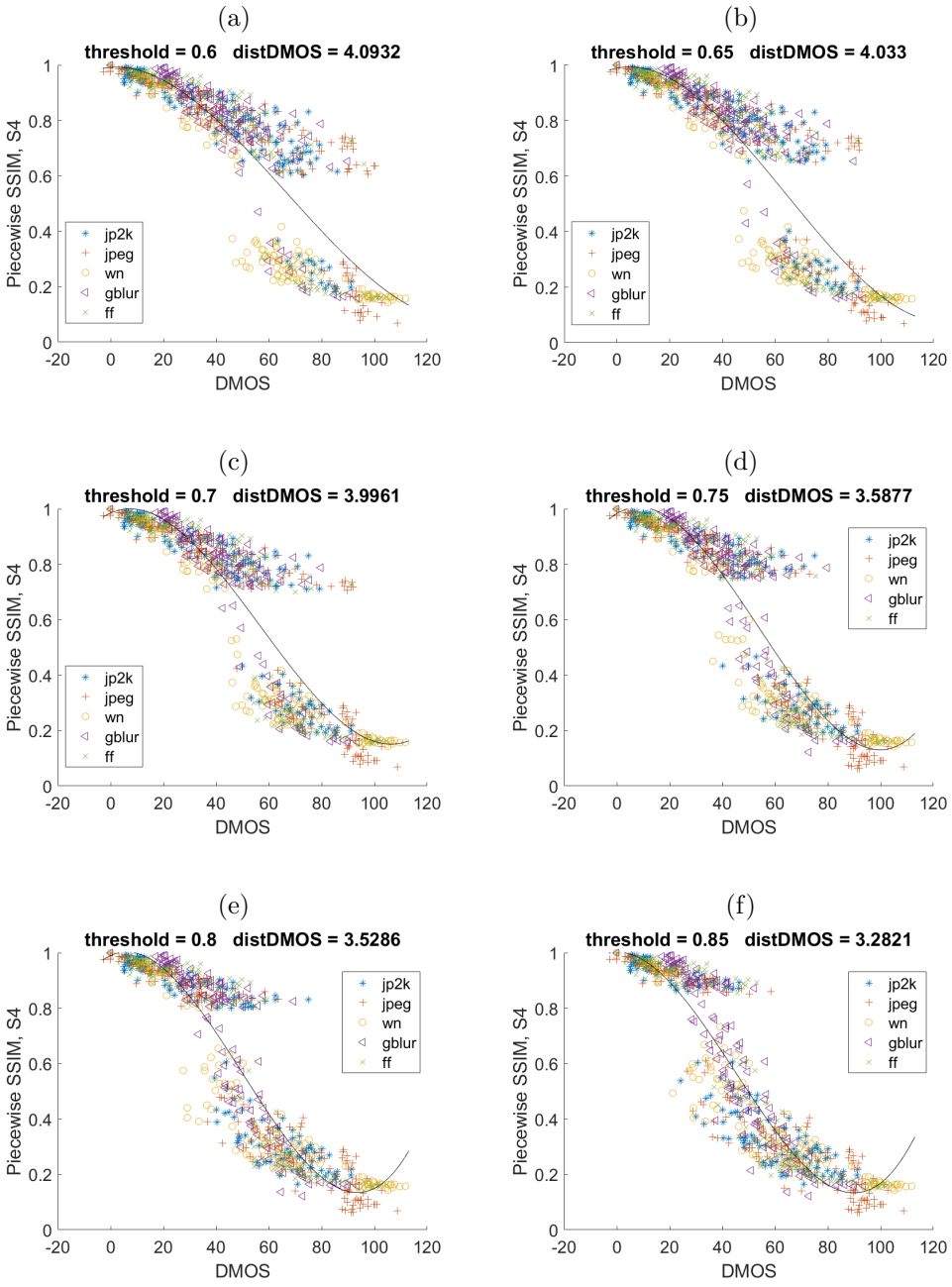


Figure 9.13: A "HybridSSIM" using S_4 as defined in Eq. (9.6).

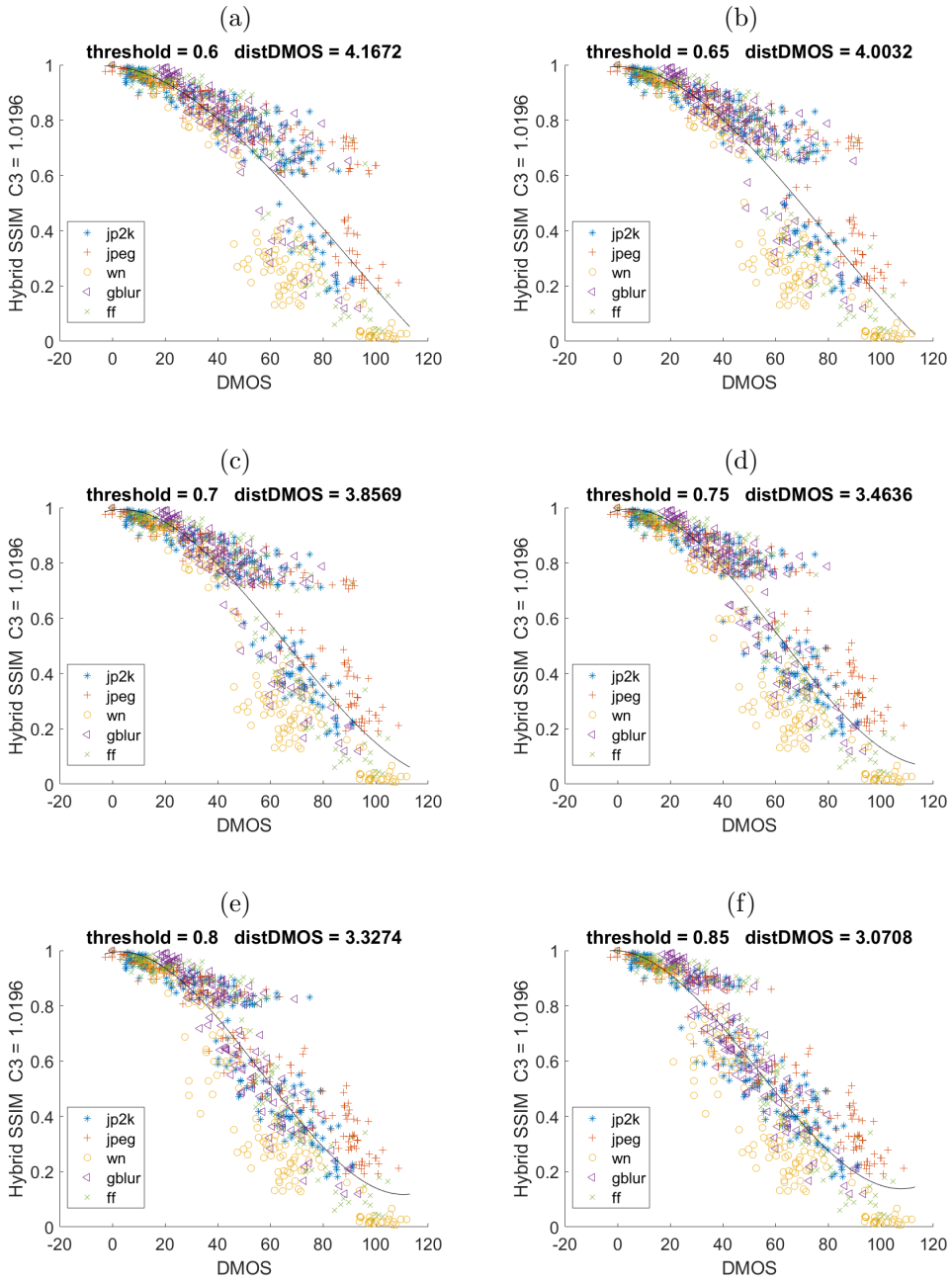


Figure 9.14: A "HybridSSIM" using the modified SSIM with $C_3 = 1$ as defined in Eq. (9.7).

measures is achieved without relying on piecewise definitions. The presentation of this image quality measure, the most successful of our attempted formulations, will conclude this chapter.

For two images patches x and y, we define the following "blended" local similarity function,

$$\text{gradSSIM1}(x, y) = \text{SSIM}(x, y) \cdot S_4(x, y)^{1-\text{SSIM}(x,y)}, \tag{9.8}$$

where the name "gradSSIM1" differentiates this measure from our previous "gradSSIM". For SSIM near 1, $S_4(x, y)^{1-\text{SSIM}(x,y)} \approx 1$ and gradSSIM1 is close to the SSIM function. For SSIM near 0, $S_4(x, y)^{1-\text{SSIM}(x,y)}$ is close to S_4. In this way, we reduce the effect of S_4 in the low-DMOS region, where it exhibits a great deal of scatter, and increase the effect of S_4 in the high-DMOS region, where it is more successful than the MSSIM. Most importantly, the combination of terms in Eq. (9.8) leverages their strengths in a seamless way.

If the effect of the S_4 is not sufficiently strong for large DMOS (low MSSIM) images, we may prefer the following definition,

$$\text{gradSSIM1}(x, y) = \text{SSIM}(x, y) \cdot S_4(x, y)^{1-\text{SSIM}(x,y)^2}. \tag{9.9}$$

On the other hand, this might introduce the problem of the S_4 term having too much influence, and hence introducing too might scatter, in the high MSSIM points. In other words, for low DMOS (high MSSIM) images, $S_4(x, y)^{1-\text{SSIM}(x,y)^2}$ may no longer be sufficiently close to 1.

The results of applying both versions of the gradSSIM1, i.e., Eq. (9.8) and Eq. (9.9), are shown in Figure 9.15. The MSSIM is included in Figure 9.15 (a) for comparison. Both versions of the gradSSIM1 in Figure 9.15 (a) and (b) appear to be working well, with little visible difference between the two plots. According to the "distDMOS" value, Eq. (9.9) is performing slightly better. In general, the gradSSIM1 improves the scatter in the mid-to-high DMOS without sacrificing the good fit for low-DMOS. At a glance, the central cluster of points bulging in the MSSIM plot is nicely pulled down by the gradSSIM1 towards to the lower end of the vertical axis. At the same time, the low-DMOS range of the gradSSIM1 is only somewhat more diffuse compared to the MSSIM.

Throughout our work we had been paying close attention to some problematic groups of points in this migh-to-high DMOS bulge; Eventually, these concerning points were investigated during our subjective evaluation. By pulling this central cluster downwards, the gradSSIM1 handles these problematic groups particularly nicely. Perhaps the most significant difference between the MSSIM and our blended approach in Figure 9.15 is the treatment of those (red) JPEG points with high DMOS, i.e., positioned between 90 to 110

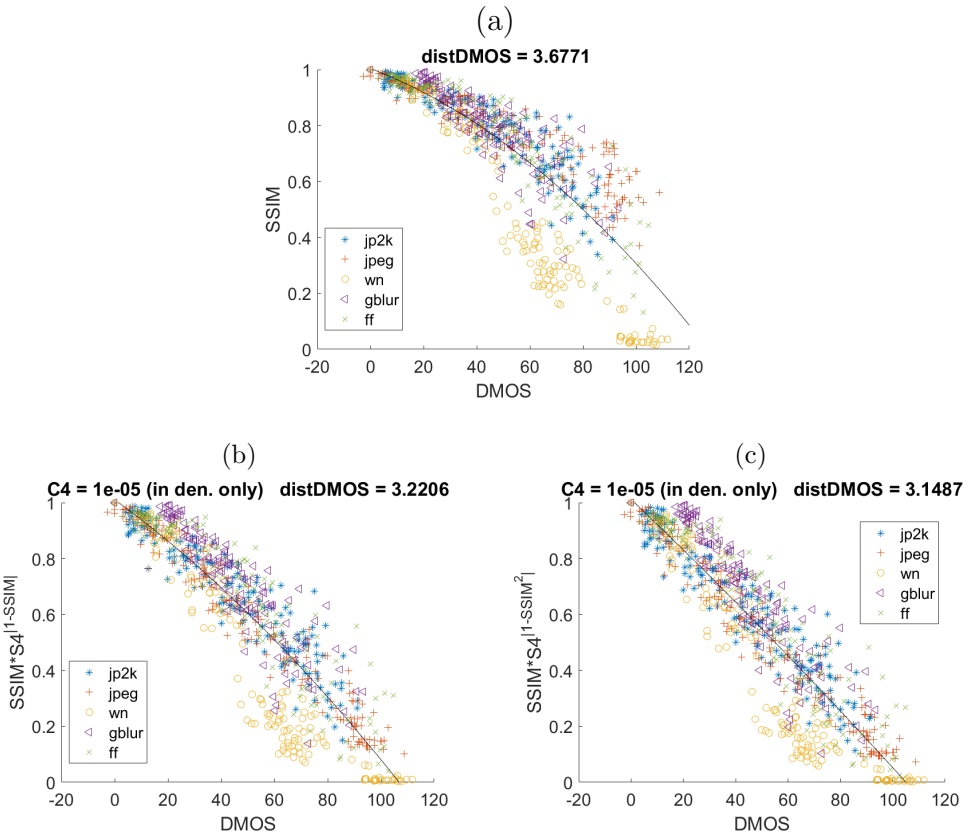


Figure 9.15: The gradSSIM1 in Eqs. (9.8) and (9.9). The MSSIM is shown for comparison.

on the horizontal axis. While the MSSIM of these points is rather high, between 0.4 and 0.7 (see Figure 9.15 (a)), the gradSSIM1 reduces these values significantly, down to 0.1 to 0.4. As mentioned when summarizing our subjective evaluation, these lower values seem to be more appropriate for these images. The same can be said for the similarly placed (blue) JPEG 2000 points, i.e., with DMOS in the range of 90 to 110.

Our gradSSIM1 also penalizes the (yellow) white noise points with slightly higher DMOS, i.e., in the range 50 to 80. Because the JPEG and JPEG 2000 points also are pulled downward, these white noise points are effectively closer to the regression line for our method. Similarly, the white noise points with in the range 90 to 110 are pulled even nearer to 0 on the vertical axis. This placement agrees with our subjective evaluation, during which we observed these images to be extremely degraded.

For the high-DMOS images, i.e., near 100, the attempt to cluster the data points may

be considered as more of an academic exercise than one of practical value. After all, and as alluded earlier, these images are heavily distorted, and this is especially true for those white noise degraded images just discussed. However, it may still be of practical concern to improve the fit in the mid-DMOS range, i.e., 40 to 80. As we have discussed, the MSSIM assigns a relatively high value to most images in the mid-DMOS range. Indeed, with the exception of the white noise images, the great majority of the mid-DMOS points lie above the fitted curve in Figure 9.15 (a). Importantly, these points remain similarly elevated for most values of the stability constants. Even when $C_3 = 1$ and the highest degree of penalty to these points is achieved (see Figure 9.9 (b)), the mid-DMOS MSSIM is still generally higher than the mid-DMOS gradSSIM1. Our method can produce smaller quality values for both the mid-DMOS and high-DMOS range which, on the basis of the subjective evaluation, are warranted. Once again, this placement cannot be achieved simply by varying the stability constants in the MSSIM.

The Effects of Downsampling

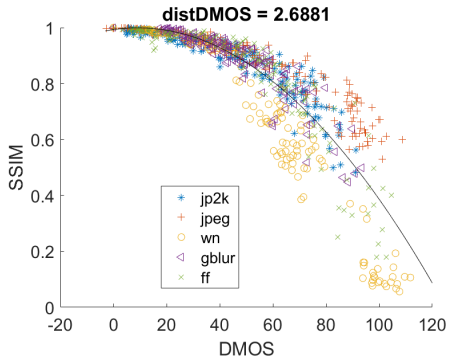
We had finally formulated a gradient-based SSIM-like function which could be seen to "improve" the MSSIM. However, the optimal application of the MSSIM involves a downsampling procedure which we had not yet explored. In order to fully asses the improvements made by our method, we needed to investigate the effect of the downsampling procedure.

The suggested downsampling procedure is described at [28]. For our gradient-based measure, the question of when to perform the downsampling is important. One could either first downsample the images before computing their gradients, or compute the gradients before downsampling the gradient images. We attempt both of these possibilities in Figure 9.16. The MSSIM with downsampling is also shown in Figure 9.16 (a) for comparison. Most interestingly, according to Figure 9.16 (b)-(e), it seems to make little difference when one downsamples the images relative to the gradient computation. This observation is reflected in the similar "distDMOS" scores between all four gradSSIM1 plots. Unlike before, Eq. (9.8) now appears to have the edge, however small, over Eq. (9.9) when downsampling is included.

It is striking how much the MSSIM is improved by the downsampling procedure. The "distDMOS" is significantly decreased between Figure 9.15 (a) and Figure 9.16 (a). By comparison, the gradSSIM1 doesn't profit as much from the additional preprocessing of the images. While the gradSSIM1 is still outperforming the MSSIM in terms of the "distDMOS" metric, its lead is much decreased. That being said, this preprocessing method has been tailored with the MSSIM in mind. It is possible that a different preprocessing method

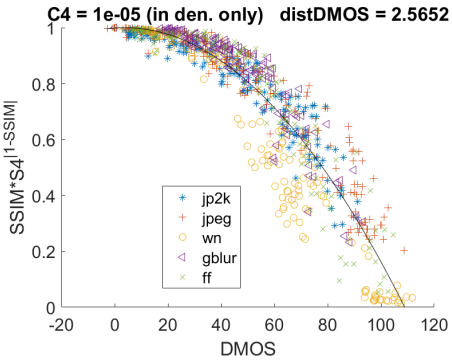
could be more advantageous for the gradSSIM1. This question is, of course, outside of the scope of this thesis.

(a)

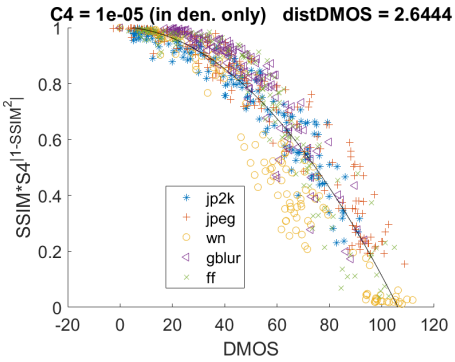


First downsample the images, then compute the gradients:

(b)

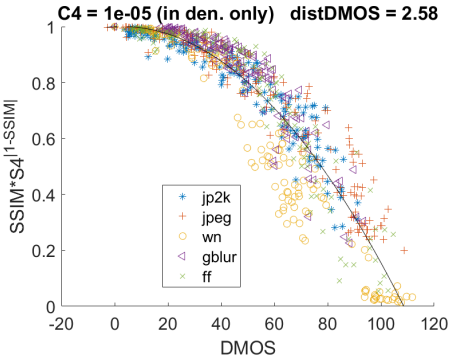


(c)



First compute the gradients, then downsample the gradient images:

(d)



(e)

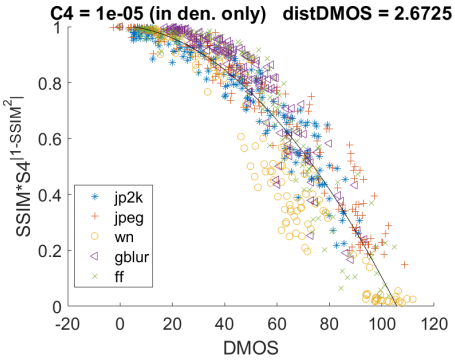


Figure 9.16: Compute the MSSIM and the gradSSIM1 with downsampling. The suggested downsampling procedure is included in the file "ssim.m" available at [28].

Chapter 10

Concluding Remarks

We studied many topics over the course of our work. In this section, we summarize the main contributions of each chapter in this thesis. Many of these topics invite worthwhile extensions which, at the time of writing, have not been fully explored due to time limitations. Throughout this discussion, we also highlight some natural avenues for future work.

The first half of this thesis focusses on the Weberized distance, the correlation, and the SSIM. In Chapter 3, we inserted an intensity-dependent weight function into the L^2-based distance in order for it to conform to generalized Weber's model of perception. We solved the resulting best approximation problem and examined the "Weberization" effect of the weighting function on solutions in both one- and two-dimension(s). An open question worthy of exploration is whether there is any relationship between the approximations yielded by this method and those of the related measure-based approach discussed in [15, 14]. Moreover, the range-based weight function is in no way limited to Weber's model: its behaviour may be tailored to other applications in image processing and beyond.

In Chapter 4, we performed a simple experiment using the set of equal-MSE "Einstein" images, in which we investigated the roles of each individual component of the SSIM. Our experiment suggested that jointly the S_2 and S_3 terms are largely responsible for the total discernment by the SSIM function. On the other hand, we observed that the Weberized distance, which was similar for all distortion types except blurring, exhibits no obvious relationship with perceptual quality. It would be interesting to explore the reason for its exceptional sensitivity to blurring distortion.

In response to these observations, Chapter 5 explored adding the correlation as a regularization term in our Weberized best approximation problem. We analytically determined

solutions for the simple case $a = 0$ and $\lambda \geq 0$. For general values of these parameters, we showed that increasing λ "undoes" the Weberization in the simple one-dimensional examples considered. In the future, it would be worthwhile to extend this approach to the two-dimensional case.

The second half of this thesis is devoted to our exploration of the gradient. In Chapter 6, we incorporated the squared L^2-distance between gradients as a regularization term in the traditional L^2-based best approximation problem. We showed that the Fourier coefficients remain optimal in some special cases—in particular, if the orthogonal cosine basis is used in the continuous setting or if the DCT/DFT basis functions are used in the discrete setting. An interesting open question is whether this result holds for other orthonormal basis sets. During this work, we also proved that the discrete derivatives of the DCT and DFT basis functions form an orthogonal set, a result which has not appeared in the literature to the best of our knowledge.

In Chapter 7, we studied the best approximation problem which maximizes the correlation between gradients. We derived the related stationarity conditions and found that an infinity of solutions exists. We showed that a unique solution can be obtained by imposing the conditions of equal gradient means and equal gradient norms. Using some simple examples, we saw that a significant number of basis functions are required to yield reasonable approximations. Finally, we showed that the same results are obtained if one considers maximizing the entire SSIM function between gradients. Although time constraints regrettably prevented its inclusion in this thesis, we also explored some initial results of using gradient ascent to maximize the stationarity conditions. These results are very interesting and worthy of discussion. Furthermore, we started to explore the addition of a regularization term in the correlation-based problem. This method, yielding much improved results, is undoubtedly worthy of future study.

Chapter 8 begins by showing that, as measures of perceptual quality, the simple L^2-distance between gradients already offers a marked improvement over the conventional MSE. We then formulated various gradient similarity measures and compared their performance on the "Einstein" images. Our proposed gradient similarity measures differ from those existing in the literature due to their mathematical simplicity. An interesting open question which arose during this work was how to compute the correlation between periodic data, i.e., angles. In the future, it would also be interesting to explore different methods for computing the image gradients beyond our simple forward differences employed throughout this thesis.

In Chapter 9, we discussed our numerous experiments using the LIVE image database. We suggest that the DMOS be considered as the independent (input) variable, an un-

derstanding which, to the best of our knowledge, has not been adopted elsewhere in the literature. Of course, this approach can also be applied to other image databases and associated subjective measures. We also performed a detailed investigation of the SSIM which involved varying its stability constants. Once again, it appears that such an analysis has not been performed elsewhere in the literature.

A major goal of Chapter 9 was to explore if the SSIM could be "improved" by incorporating gradient information. To that end, we justified the selection of our preferred gradient similarity measure, which was demonstrated to be highly correlated with the canonical correlation method. We used this to define a "gradSSIM1" image quality measure which, compared to the MSSIM, improves the fit in the mid-to-high DMOS range. The novelty in our approach lies in its ability to seamlessly blend the behaviour of different measures using SSIM-dependent exponents. Regrettably, time did not permit an examination of other possible "blended" methods. Even still, our approach introduces a new way of considering image quality measures. Indeed, it is our hope that this novel framework will stimulate future research in this area.

References

[1] Gonzalo R. Arcé, Jose L. Paredes, and John Mullan. *Handbook of Image and Video Procressing*, chapter 3.2 Nonlinear Filtering for Image Analysis and Enhancement, pages 97–99. Academic Press, 2000.

[2] Paul Bendevis and Edward R. Vrscay. Structural Similarity-Based Approximation over Orthogonal Bases: Investigating the Use of Individual Component Functions $S_k(x, y)$. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition - 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part I*, volume 8814 of *Lecture Notes in Computer Science*, pages 55–64, 2014.

[3] Dominique Brunet, Edward R. Vrscay, and Zhou Wang. Structural Similarity-Based Approximation of Signals and Images Using Orthogonal Bases. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition - 7th International Conference, ICIAR 2010, Póvoa de Varzim, Portugal, June 21-23, 2010. Proceedings, Part I*, volume 6111 of *Lecture Notes in Computer Science*, pages 11–22, 2010.

[4] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*, chapter 4.4, pages 225–229. Pearson, fourth edition, 2018.

[5] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*, chapter 7.6, pages 487–490. Pearson, fourth edition, 2018.

[6] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*, chapter 3.6 Sharpening (Highpass) Spatial Filters, pages 184–188. Pearson, 2018.

[7] Edwin Hewitt and Karl Stromberg. *Real and Abstract Analysis*, chapter IV: Function Spaces and Banach Spaces, pages 238–239. Springer, 1969.

[8] Harold Hotelling. Relations Between Two Sets of Variates. *Biometrika*, 28(3-4):321–277, 1936.

[9] Andrew Knyazev. subspacea—Angles between subspaces. Matlab Central File Exchange, https://www.mathworks.com/matlabcentral/fileexchange/55-subspacea-angles-between-subspaces, 2023.

[10] Ilona Anna Kowalik-Urbaniak. *The Quest for "Diagnostically Lossless" Medical Image Compression Using Objective Image Quality Measures*. PhD thesis, Department of Applied Mathematics, University of Waterloo, 2014.

[11] Ilona Anna Kowalik-Urbaniak, Davide La Torre, Edward R. Vrscay, and Zhou Wang. Some "Weberized" L^2-based Methods of Signal/Image Approximation. In Aurélio J. C. Campilho and Mohamed S. Kamel, editors, *Image Analysis and Recognition - 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part I*, volume 8814 of *Lecture Notes in Computer Science*, pages 20–29, 2014.

[12] Erwin Kreyszig. *Introductory Functional Analysis with Applications*, chapter 6 Further Applications: Approximation Theory, pages 327–334. Wiley, 1978.

[13] Reginald L. Lagendijk and Jan Biemond. *Handbook of Image and Video Processing*, chapter 3.6 Regularization in Image Restoration and Reconstruction, pages 150–151. Academic Press, 2000.

[14] Dongchang Li. A Novel Class of Intensity-based Metrics for Image Functions Which Accomodate a Generalized Weber's Model of Perception. Master's thesis, Department of Applied Mathematics, University of Waterloo, 2020.

[15] Dongchang Li, Davide La Torre, and Edward R. Vrscay. The Use of Intensity-based Measures to Produce Image Function Metrics Which Accomodate Weber's Models of Perception. In Aurélio Campilho, Fakhri Karray, and Bart M. ter Haar Romeny, editors, *Image Analysis and Recognition - 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27-29, 2018, Proceedings*, volume 10882 of *Lecture Notes in Computer Science*, pages 326–335, 2018.

[16] Dongchang Li, Davide La Torre, and Edward R. Vrscay. Existence, Uniqueness and Asymptotic Behaviour of Intensity-based Measures Which Conform to a Generalized Weber's Model of Perception. In Fakhri Karray, Aurélio Campilho, and Alfred C. H. Yu, editors, *Image Analysis and Recognition - 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27-29, 2019, Proceedings, Part I*, volume 11662 of *Lecture Notes in Computer Science*, pages 297–208, 2019.

[17] Anmin Liu, Weisi Lin, and Manish Narwaria. Image Quality Assessment Based on Gradient Similarity. *IEEE Transactions on Image Processing*, 21(4):1500–1512, 2012.

[18] John A. Michon. Note on the Generalized Form of Weber's Law. *Perception and Psychophysics*, 1966.

[19] EePing Ong, Weisi Lin, Zhongkang Lu, Susu Yao, and Minoru Etoh. Visual Distortion Assessment with Empahsis on Spatially Transitional Regions. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):559–566, 2004.

[20] Thrasyvoulos N. Pappas and Robert J. Safranek. *Handbook of Image and Video Processing*, chapter 2.1.1 Amplitude Nonlinearity, page 671. Academic Press, 2000.

[21] Rafael Reisenhofer, Sebastian Bosse, Gitta Kutyniok, and Thomas Wiegand. A Haar Wavelet-based Perceptual Similarity Index for Image Quality Assessment. *Signal Processing: Image Communication*, 61:33–43, 2018.

[22] Hamid R. Sheikh. *Image Quality Assessment Using Natural Scene Statistics*. PhD thesis, The University of Texas at Austin, 2004.

[23] H.R. Sheikh, M.F. Sabir, and A.C. Bovik. A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–2451, November 2006.

[24] H.R. Sheikh, Z.Wang, L. Cormack, and A.C. Bovik. LIVE Image Quality Assessment Database Release 2. http://live.ece.utexas.edu/research/quality.

[25] Jackie Shen. On the Foundations of Vision Modeling: I. Weber's Law and Weberized TV Restoration. *Physica D: Nonlinear Phenomena*, 175:241–251, 2003.

[26] Jackie Shen and Yoon-Mo Jung. Weberized Mumford-Shah Model with Bose-Einstein Photon Noise. *Applied Mathematics and Optimization*, 53:331–358, 2006.

[27] Ilona A. Urbaniak, Amelia Kunze, Dongchang Li, Davide La Torre, and Edward R. Vrscay. The Use of Intensity-dependent Weight Functions to "Weberize" L^2-based Methods of Signal and Image Approximation. *Optimization and Engineering*, 22:2349–2365, 2021.

[28] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. The SSIM Index for Image Quality Assessment. https://ece.uwaterloo.ca/~z70wang/research/ssim/.

[29] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[30] Zhou Wang. Applications of Objective Image Quality Assessment Methods [Applications Corner]. *IEEE Signal Processing Magazine*, 28:137–142, 2011.

[31] Zhou Wang and Alan Bovik. Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measure. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

[32] Zhou Wang, Alan C. Bovik, and Hamid R. Sheikh. *Digital Video Image Quality and Perceptual Coding*, chapter 7: Structural Similarity Based Image Quality Assessment. CRC Press, 2005.

[33] Lin Zhang, Lei Zhang, Xuaqin Mou, and David Zhang. FSIM: A Feature Similairty Index for Image Quality Assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.

APPENDICES

Appendix A

Orthonormality of the Discrete Derivatives of the DCT and DFT Basis Functions

A.1 Proof for the DFT Basis Functions

Theorem 5. Let $\{\phi_0, \cdots, \phi_{N-1}\}$ denote the following DFT basis functions,

$$\phi_k[n] = \frac{1}{\sqrt{N}} \exp\left(\frac{i2\pi kn}{N}\right), \quad 0 \leq n \leq N-1. \tag{6.20}$$

Let the discrete derivative of these functions be defined by simple forward differences so that

$$D\phi_k[n] = \phi_k[n+1] - \phi_k[n], \quad 0 \leq n \leq N-1. \tag{6.21}$$

Then for a given $N > 0$, $\{D\phi_0, \cdots, D\phi_{N-1}\}$ forms an orthogonal set in $\mathbb{C}^N$.

Proof. First use Euler's formula to rewrite the basis functions

$$\phi_k[n] = \frac{1}{\sqrt{N}} \left[\cos\left(\frac{2\pi kn}{N}\right) + i \sin\left(\frac{2\pi kn}{N}\right)\right], \quad 0 \leq n \leq N-1.$$

Using this form, the discrete derivatives become

$$\begin{aligned} D\phi_k[n] &= \phi_k[n+1] + \phi_k[n] \\ &= \frac{1}{\sqrt{N}} \left(\left[\cos\left(\frac{2\pi k(n+1)}{N}\right) - \cos\left(\frac{2\pi kn}{N}\right) \right] \right. \\ &\qquad \left. + i \left[\sin\left(\frac{2\pi k(n+1)}{N}\right) - \sin\left(\frac{2\pi kn}{N}\right) \right] \right). \end{aligned} \tag{A.1}$$

Use the cosine addition formula to rewrite the first cosine function above,

$$\cos\left(\frac{2\pi k(n+1)}{N}\right) = \cos\left(\frac{2\pi kn}{N}\right)\cos\left(\frac{2\pi k}{N}\right) - \sin\left(\frac{2\pi kn}{N}\right)\sin\left(\frac{2\pi k}{N}\right).$$

Similarly, use the sine addition formula to rewrite the first sine function above,

$$\sin\left(\frac{2\pi k(n+1)}{N}\right) = \sin\left(\frac{2\pi kn}{N}\right)\cos\left(\frac{2\pi k}{N}\right) + \cos\left(\frac{2\pi kn}{N}\right)\sin\left(\frac{2\pi k}{N}\right).$$

Substitute both expressions in Eq. (6.21) to get

$$\begin{aligned} D\phi_k[n] = \frac{1}{\sqrt{N}} \left(\left[\cos\left(\frac{2\pi kn}{N}\right) \left(\cos\left(\frac{2\pi k}{N}\right) - 1 \right) - \sin\left(\frac{2\pi kn}{N}\right)\sin\left(\frac{2\pi k}{N}\right) \right] \right. \\ \left. + i \left[\sin\left(\frac{2\pi kn}{N}\right) \left(\cos\left(\frac{2\pi k}{N}\right) - 1 \right) + \cos\left(\frac{2\pi kn}{N}\right)\sin\left(\frac{2\pi k}{N}\right) \right] \right). \end{aligned} \tag{A.2}$$

We will denote the constants with respect to n by

$$C_k = \left(\cos\left(\frac{2\pi k}{N}\right) - 1 \right) \qquad \text{and} \qquad D_k = \sin\left(\frac{2\pi k}{N}\right). \tag{A.3}$$

Now Eq. (A.2) can be written

$$\begin{aligned} D\phi_k[n] = \frac{1}{\sqrt{N}} \left(\left[C_k \cos\left(\frac{2\pi kn}{N}\right) - D_k \sin\left(\frac{2\pi kn}{N}\right) \right] \right. \\ \left. + i \left[C_k \sin\left(\frac{2\pi kn}{N}\right) + D_k \cos\left(\frac{2\pi kn}{N}\right) \right] \right). \end{aligned}$$

We can use Euler's formula to rewrite the above expression as

$$\begin{aligned} D\phi_k[n] &= \frac{1}{\sqrt{N}}\left(C_k \exp\left(\frac{i2\pi kn}{N}\right) + iD_k \exp\left(\frac{i2\pi kn}{N}\right)\right) \\ &= \frac{1}{\sqrt{N}}\left((C_k + iD_k)\exp\left(\frac{i2\pi kn}{N}\right)\right). \end{aligned}$$

Which, with reference to the definition in Eq. (6.20), we recognize as

$$D\phi_k[n] = (C_k + iD_k)\phi_k[n], \quad 0 \leq n \leq N-1. \tag{A.4}$$

We can finally compute the inner product using the simplified expression in Eq. (A.4).

$$\begin{aligned} A_{kl} &= \langle D\phi_k, D\phi_l\rangle \\ &= \langle (C_k + iD_k)\phi_k, (C_l + iD_l)\phi_l\rangle \\ &= (C_k + iD_k)(C_l - iD_l)\langle \phi_k, \phi_l\rangle \\ &= (C_k^2 + iD_k^2)\delta_{kl} \end{aligned}$$

Recall that $\langle \phi_k, \phi_l\rangle = \delta_{kl}$. Substitute the constants definitions given in Eq. (A.3),

$$\begin{aligned} A_{kl} &= \left(\left(\cos\left(\frac{2\pi k}{N}\right) - 1\right)^2 + \sin\left(\frac{2\pi k}{N}\right)\right) \\ &= \left(2 - 2\cos\left(\frac{2\pi k}{N}\right)\right)\delta_{kl} \\ &= 4\sin^2\left(\frac{\pi k}{N}\right)\delta_{kl} \end{aligned}$$

This completes the proof.

To summarize, we have shown that for a given $N > 0$, $0 \leq k, l \leq N-1$,

$$\langle D\phi_k, D\phi_l\rangle = \begin{cases} 0, & k \neq l \\ 4\sin^2\left(\frac{\pi k}{N}\right), & k = l \end{cases}$$

■

A.2 Proof for the DCT Basis Functions

Theorem 4. Let $\{\phi_0, \cdots, \phi_{N-1}\}$ denote the following DCT basis functions,

$$\begin{aligned}\phi_0[n] &= \frac{1}{\sqrt{N}},\\ \phi_k[n] &= \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right), \quad 1 \le k \le N-1, \quad 0 \le n \le N-1. \end{aligned} \tag{6.18}$$

Let the discrete derivative of these functions be defined by simple forward differences so that

$$D\phi_k[n] = \phi_k[n+1] - \phi_k[n], \quad 0 \le n \le N-1. \tag{6.19}$$

Then for a given $N > 0$, $\{D\phi_0, \cdots, D\phi_{N-1}\}$ forms an orthogonal set in $\mathbb{R}^N$.

Proof. There are two cases.

Case 1: $k = 0$. Then,

$$D\phi_0[n] = \frac{1}{\sqrt{N}} - \frac{1}{\sqrt{N}} = 0, \quad 0 \le n \le N-1.$$

Clearly, for any $l \in \{0, \cdots, N-1\}$,

$$\langle D\phi_0, D\phi_l \rangle = \sum_{n=0}^{N-1} 0 \cdot D\phi_l[n] = 0. \tag{A.5}$$

Case 2: $1 \le k \le N-1$. The discrete derivatives defined in Eq. (6.19) involve a difference of cosine functions, i.e., for $0 \le n \le N-1$,

$$\begin{aligned} D\phi_k[n] &= \phi_k[n+1] - \phi_k[n]\\ &= \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n+1+\frac{1}{2}\right)\right) - \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right). \end{aligned} \tag{A.6}$$

Omitting for the moment the factor $\sqrt{\frac{2}{N}}$, we can rewrite $\phi_k[n+1]$ using the cosine addition formula as follows,

$$\cos\left(\frac{k\pi}{N}\left(n+1+\frac{1}{2}\right)\right) = \cos\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right)\cos\left(\frac{k\pi}{N}\right) - \sin\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right)\sin\left(\frac{k\pi}{N}\right).$$

After substituting the difference written above into Eq. (A.6) and factoring, the discrete derivative is

$$D\phi_k[n] = \sqrt{\frac{2}{N}}\left[\cos\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right)\left(\cos\left(\frac{k\pi}{N}\right)-1\right) - \sin\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right)\sin\left(\frac{k\pi}{N}\right)\right].$$

This equation can be more compactly written as

$$D\phi_k = C_k\phi_k - D_k\psi_k, \tag{A.7}$$

where the vector ψ_k is given by

$$\psi_k[n] = \sqrt{\frac{2}{N}}\sin\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right), \quad 1 \le k \le N-1, \quad 0 \le n \le N-1, \tag{A.8}$$

and the constants with respect to n are denoted by

$$C_k = \cos\left(\frac{k\pi}{N}\right) - 1, \quad D_k = \sin\left(\frac{k\pi}{N}\right), \quad 1 \le k \le N-1. \tag{A.9}$$

Notice the subtle difference between these coefficients and those from proof using the DFT basis functions.

We may now use the expansion in Eq. (A.7) to compute the following inner product,

$$\begin{aligned} A_{kl} &= \langle D\phi_k, D\phi_l\rangle \\ &= \langle C_k\phi_k - D_k\psi_k, C_l\phi_l - D_l\psi_l\rangle \\ &= C_kC_l\langle\phi_k,\phi_l\rangle - C_kD_l\langle\phi_k,\psi_l\rangle - D_kC_l\langle\psi_k,\phi_l\rangle + D_kD_l\langle\psi_k,\psi_l\rangle. \end{aligned} \tag{A.10}$$

We know that the DCT basis functions are orthonormal, i.e., $\langle\phi_k,\phi_l\rangle = \delta_{kl}$. We claim that Eq. (A.10) will be further simplified by the following two properties:

(i) The sine functions are also orthonormal, i.e., $\langle\psi_k,\psi_l\rangle = \delta_{kl}$, and

(ii) The cross terms vanish, i.e., $C_kD_l\langle\phi_k,\psi_l\rangle + D_kC_l\langle\psi_k,\phi_l\rangle = 0$ for any $1 \le k,l \le N-1$.

We will first establish (i) by demonstrating that $\langle\psi_k,\psi_l\rangle = \delta_{kl}$. (Should the reader wish to verify the assumption on the DCT basis functions, the same argument can be used to prove that $\langle\phi_k,\phi_l\rangle = \delta_{kl}$.)

The inner product in (i) is

$$\begin{aligned}\langle \psi_k, \psi_l \rangle &= \sum_{n=0}^{N-1} \psi_k \overline{\psi_l} \\ &= \sum_{n=0}^{N-1} \sqrt{\frac{2}{N}} \sin\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right) \sqrt{\frac{2}{N}} \sin\left(\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\right),\end{aligned}$$

Rewrite $\psi_k[n]$ using complex exponentials,

$$\begin{aligned}\psi_k[n] &= \sqrt{\frac{2}{N}} \sin\left(\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right) \\ &= \sqrt{\frac{1}{2N}} \frac{1}{i} \left[\exp\left(i\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right) - \exp\left(-i\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right)\right], \quad 0 \le n \le N-1.\end{aligned}$$

Using this form, the inner product becomes

$$\begin{aligned}\langle \psi_k, \psi_l \rangle &= \frac{1}{2N} \sum_{n=0}^{N-1} \frac{1}{i} \left[\exp\left(i\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right) - \exp\left(-i\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right)\right] \\ &\qquad \cdot \frac{1}{i} \left[\exp\left(i\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\right) - \exp\left(-i\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\right)\right] \\ &= \frac{1}{2N} \sum_{n=0}^{N-1} \left[\exp\left(i\frac{(k-l)\pi}{N}\left(n+\frac{1}{2}\right)\right) + \exp\left(-i\frac{(k-l)\pi}{N}\left(n+\frac{1}{2}\right)\right)\right. \\ &\qquad \left. - \exp\left(i\frac{(k+l)\pi}{N}\left(n+\frac{1}{2}\right)\right) - \exp\left(-i\frac{(k+l)\pi}{N}\left(n+\frac{1}{2}\right)\right)\right].\end{aligned}$$

Now strategically add 0 to the decaying exponentials,

$$\begin{aligned}\langle \psi_k, \psi_l \rangle = \frac{1}{2N} \sum_{n=0}^{N-1} &\left[\exp\left(i\frac{(k-l)\pi}{N}\left(n+\frac{1}{2}\right)\right) + \exp\left(-i\frac{(k-l)\pi}{N}\left(n+\frac{1}{2}+\frac{1}{2}-\frac{1}{2}\right)\right)\right. \\ &\left. - \exp\left(i\frac{(k+l)\pi}{N}\left(n+\frac{1}{2}\right)\right) - \exp\left(-i\frac{(k+l)\pi}{N}\left(n+\frac{1}{2}+\frac{1}{2}-\frac{1}{2}\right)\right)\right].\end{aligned}$$

Then separate the terms independent of the summation,

$$\langle \psi_k, \psi_l \rangle = \frac{1}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \sum_{n=0}^{N-1} \left[\exp\left(i\frac{(k-l)\pi}{N}n\right) + \exp\left(-i\frac{(k-l)\pi}{N}(n+1)\right)\right]$$
$$- \frac{1}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \sum_{n=0}^{N-1} \left[\exp\left(i\frac{(k+l)\pi}{N}n\right) + \exp\left(-i\frac{(k+l)\pi}{N}(n+1)\right)\right].$$

Reindex the terms involving $(n+1)$ in the exponent by letting $m = n+1$,

$$\langle \psi_k, \psi_l \rangle = \frac{1}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \left[\sum_{n=0}^{N-1} \exp\left(i\frac{(k-l)\pi}{N}n\right) + \sum_{m=1}^{N} \exp\left(-i\frac{(k-l)\pi}{N}m\right)\right]$$
$$- \frac{1}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \left[\sum_{n=0}^{N-1} \exp\left(i\frac{(k+l)\pi}{N}n\right) + \sum_{m=1}^{N} \exp\left(-i\frac{(k+l)\pi}{N}m\right)\right].$$

To allow us to combine the exponentials, reindex again letting $n = -m$,

$$\langle \psi_k, \psi_l \rangle = \frac{1}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \left[\sum_{n=0}^{N-1} \exp\left(i\frac{(k-l)\pi}{N}n\right) + \sum_{n=-N}^{-1} \exp\left(i\frac{(k-l)\pi}{N}n\right)\right]$$
$$- \frac{1}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \left[\sum_{n=0}^{N-1} \exp\left(i\frac{(k+l)\pi}{N}n\right) + \sum_{n=-N}^{-1} \exp\left(i\frac{(k+l)\pi}{N}n\right)\right].$$
$$= \frac{1}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \sum_{n=-N}^{N-1} \exp\left(i\frac{(k-l)\pi}{N}n\right)$$
$$- \frac{1}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \sum_{n=-N}^{N-1} \exp\left(i\frac{(k+l)\pi}{N}n\right).$$

Reindex a final time, letting $m = n+N+1$,

$$\langle \psi_k, \psi_l \rangle = \frac{1}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \sum_{m=1}^{2N} \exp\left(i\frac{(k-l)\pi}{N}(m-N-1)\right)$$
$$- \frac{1}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \sum_{m=1}^{2N} \exp\left(i\frac{(k+l)\pi}{N}(m-N-1)\right).$$

After some rearranging, we have

$$\begin{aligned}\langle \psi_k, \psi_l \rangle &= \frac{1}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \exp\left(-i(k-l)\pi\right) \sum_{m=1}^{2N} \exp\left(i\frac{(k-l)\pi}{N}(m-1)\right) \\ &\quad - \frac{1}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \exp\left(-i(k+l)\pi\right) \sum_{m=1}^{2N} \exp\left(i\frac{(k+l)\pi}{N}(m-1)\right) \\ &= \frac{1}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) (-1)^{k-l} \sum_{m=1}^{2N} \exp\left(i\frac{(k-l)\pi}{N}\right)^{m-1} \\ &\quad - \frac{1}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) (-1)^{k+l} \sum_{m=1}^{2N} \exp\left(i\frac{(k+l)\pi}{N}\right)^{m-1} \end{aligned} \tag{A.11}$$

Once more, there are two cases.

Case 1: When $k = l$, the expression simplifies so that we have

$$\begin{aligned}\langle \psi_k, \psi_k \rangle &= \frac{1}{2N} \sum_{m=1}^{2N} 1 - \frac{1}{2N} \exp\left(i\frac{2k\pi}{2N}\right) (-1)^{2k} \sum_{m=1}^{2N} \exp\left(i\frac{2k\pi}{N}\right)^{m-1} \\ &= 1 - \frac{1}{2N} \exp\left(i\frac{2k\pi}{2N}\right) (-1)^{2k} \sum_{m=1}^{2N} \exp\left(i\frac{2k\pi}{N}\right)^{m-1}.\end{aligned}$$

The remaining summation is a finite geometric series with $r = \exp\left(i\frac{2k\pi}{N}\right) \neq 1$. The sum is given by,

$$\begin{aligned}\sum_{m=1}^{2N} \exp\left(i\frac{2k\pi}{N}\right)^{m-1} &= \frac{1 - \exp\left(i\frac{2k\pi}{N}\right)^{2N}}{1-r} \\ &= \frac{1 - \exp\left(i4k\pi\right)}{1-r} \\ &= \frac{1-1}{1-r} \\ &= 0.\end{aligned}$$

Hence $\langle \psi_k, \psi_k \rangle$ =1 for $1 \leq k \leq N-1$.

Case 2: Now consider $k \neq l$. Return to Eq. (A.11),

$$\langle \psi_k, \psi_l \rangle = \frac{(-1)^{k-l}}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \sum_{m=1}^{2N} \exp\left(i\frac{(k-l)\pi}{N}\right)^{m-1} - \frac{(-1)^{k+l}}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \sum_{m=1}^{2N} \exp\left(i\frac{(k+l)\pi}{N}\right)^{m-1}.$$

This time, we have two geometric series with $r_1 = \exp\left(i\frac{(k-l)\pi}{N}\right)$ and $r_2 = \exp\left(i\frac{(k+l)\pi}{N}\right)$. The inner product becomes,

$$\langle \psi_k, \psi_l \rangle = \frac{(-1)^{k-l}}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \left[\frac{1 - \exp\left(i2(k-l)\pi\right)}{1 - r_1}\right] - \frac{(-1)^{k+l}}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \left[\frac{1 - \exp\left(i2(k+l)\pi\right)}{1 - r_2}\right].$$

Note that $2(k \pm l)$ is even for any $1 \leq k, l \leq N-1$. The reindexing performed above allowed for this simplification, which gives

$$\langle \psi_k, \psi_l \rangle = \frac{(-1)^{k-l}}{2N} \exp\left(i\frac{(k-l)\pi}{2N}\right) \left[\frac{1-1}{1-r_1}\right] - \frac{(-1)^{k+l}}{2N} \exp\left(i\frac{(k+l)\pi}{2N}\right) \left[\frac{1-1}{1-r_2}\right] = 0.$$

All together, we have shown that $\{\psi_k, \psi_l\} = \delta_{kl}$ for $1 \leq k, l \leq N-1$, which completes the proof of claim (i).

We will now proceed to establish claim (ii). We are required to show that $C_k D_l \langle \phi_k, \psi_l \rangle + D_k C_l \langle \psi_k, \phi_l \rangle = 0$ for all k, l. First consider the second inner product, written below,

$$\langle \psi_k, \phi_l \rangle = \sum_{n=0}^{N-1} \psi_k \overline{\phi_l} = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N}} \sin\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right) \sqrt{\frac{2}{N}} \cos\left(\frac{l\pi}{N}\left(n + \frac{1}{2}\right)\right).$$

As before, we express the trigonometric functions using complex exponentials,

$$\langle \psi_k, \phi_l \rangle = \frac{1}{2Ni} \sum_{n=0}^{N-1} \left[\exp\left(i\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right) - \exp\left(-i\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right)\right] \cdot \left[\exp\left(i\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\right) + \exp\left(-i\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\right)\right]$$

Expand and separate the terms independent of n to get

$$\begin{aligned}\langle \psi_k, \phi_l \rangle = \frac{1}{2Ni} \Bigg[&\exp\left(i\frac{(k+l)\pi}{2N}\right) \sum_{n=0}^{N-1} \exp\left(i\frac{(k+l)\pi}{N}\right)^n \\ &- \exp\left(-i\frac{(k+l)\pi}{2N}\right) \sum_{n=0}^{N-1} \exp\left(-i\frac{(k+l)\pi}{N}\right)^n \Bigg] \\ + \frac{1}{2Ni} \Bigg[&\exp\left(i\frac{(k-l)\pi}{2N}\right) \sum_{n=0}^{N-1} \exp\left(i\frac{(k-l)\pi}{N}\right)^n \\ &- \exp\left(-i\frac{(k-l)\pi}{2N}\right) \sum_{n=0}^{N-1} \exp\left(-i\frac{(k-l)\pi}{N}\right)^n \Bigg]\end{aligned}$$

If $k = l$, then the difference of terms involving $k-l$ simplifies to 0 immediately. Taking the sum of the geometric series involving $k+l$ shows that $\langle \psi_k, \phi_k \rangle = 0$.

For $k \neq l$, $\exp\left(\pm i\frac{(k-l)\pi}{N}\right) \neq 1$. We have four geometric series, whose sums are written below,

$$\begin{aligned}\langle \psi_k, \phi_l \rangle = \frac{1}{2Ni} \Bigg[&\exp\left(i\frac{(k+l)\pi}{2N}\right) \frac{1-\exp\left(i(k+l)\pi\right)}{1-\exp\left(i\frac{(k+l)\pi}{N}\right)} \\ &- \exp\left(-i\frac{(k+l)\pi}{2N}\right) \frac{1-\exp\left(-i(k+l)\pi\right)}{1-\exp\left(-i\frac{(k+l)\pi}{N}\right)} \Bigg] \\ + \frac{1}{2Ni} \Bigg[&\exp\left(i\frac{(k-l)\pi}{2N}\right) \frac{1-\exp\left(i(k-l)\pi\right)}{1-\exp\left(i\frac{(k-l)\pi}{N}\right)} \\ &- \exp\left(-i\frac{(k-l)\pi}{2N}\right) \frac{1-\exp\left(-i(k-l)\pi\right)}{1-\exp\left(-i\frac{(k-l)\pi}{N}\right)} \Bigg]. \end{aligned} \tag{A.12}$$

The inner product is symmetric because our basis functions are real. In particular, $\langle \phi_k, \psi_k \rangle = \langle \psi_k, \phi_k \rangle = 0$. Therefore, $C_k D_k \langle \phi_k, \psi_k \rangle + D_k C_k \langle \psi_k, \phi_k \rangle = 0$, i.e., (ii) is proven for the case $k = l$.

For $k \neq l$, we can obtain $\langle \phi_k, \psi_l \rangle$ by interchanging k and l in the expression for $\langle \psi_k, \phi_l \rangle$ written in Eq. (A.12),

$$\begin{aligned}
\langle \phi_k, \psi_l \rangle = \frac{1}{2Ni} \Bigg[& \exp\left(i\frac{(k+l)\pi}{2N}\right) \frac{1-\exp\left(i(k+l)\pi\right)}{1-\exp\left(i\frac{(k+l)\pi}{N}\right)} \\
& - \exp\left(-i\frac{(k+l)\pi}{2N}\right) \frac{1-\exp\left(-i(k+l)\pi\right)}{1-\exp\left(-i\frac{(k+l)\pi}{N}\right)} \Bigg] \\
+ \frac{1}{2Ni} \Bigg[& -\exp\left(i\frac{(k-l)\pi}{2N}\right) \frac{1-\exp\left(i(k-l)\pi\right)}{1-\exp\left(i\frac{(k-l)\pi}{N}\right)} \\
& + \exp\left(-i\frac{(k-l)\pi}{2N}\right) \frac{1-\exp\left(-i(k-l)\pi\right)}{1-\exp\left(-i\frac{(k-l)\pi}{N}\right)} \Bigg] . \qquad \text{(A.13)}
\end{aligned}$$

Using these expansions, the cross terms in (ii), rewritten below,

$$\text{LHS} = D_k C_l \langle \psi_k, \phi_l \rangle + D_l C_k \langle \psi_l, \phi_k \rangle$$

become

$$\begin{aligned}
\text{LHS} = \frac{1}{2Ni}(D_k C_l + C_k D_l) \Bigg[& \exp\left(i\frac{(k+l)\pi}{2N}\right) \frac{1-\exp\left(i(k+l)\pi\right)}{1-\exp\left(i\frac{(k+l)\pi}{N}\right)} \\
& - \exp\left(-i\frac{(k+l)\pi}{2N}\right) \frac{1-\exp\left(-i(k+l)\pi\right)}{1-\exp\left(-i\frac{(k+l)\pi}{N}\right)} \Bigg] \\
+ \frac{1}{2Ni}(D_k C_l - C_k D_l) \Bigg[& \exp\left(i\frac{(k-l)\pi}{2N}\right) \frac{1-\exp\left(i(k-l)\pi\right)}{1-\exp\left(i\frac{(k-l)\pi}{N}\right)} \\
& - \exp\left(-i\frac{(k-l)\pi}{2N}\right) \frac{1-\exp\left(-i(k-l)\pi\right)}{1-\exp\left(-i\frac{(k-l)\pi}{N}\right)} \Bigg] . \qquad \text{(A.14)}
\end{aligned}$$

If $k+l$ is even, i.e. if $k+l=2m$ for some integer m, then $k-l=2(m-l)$ is also even. Similarly, if $k+l$ is odd, then $k-l$ must also be odd. We will consider those two cases separately below.

Case 1: $k+l$, $k-l$ are even. Then each of $\exp\left(\pm i(k-l)\pi\right)=\exp\left(\pm i(k+l)\pi\right)=1$ in Eq. (A.14), so that we immediately obtain the desired result LHS $=0$.

Case 2: $k+l=p$ is odd and $k-l=q$ is odd. Then Eq. (A.14) becomes

$$\begin{aligned}
\text{LHS} &= \frac{1}{2Ni}\left[\exp\left(i\frac{p\pi}{2N}\right)\frac{1-\exp\left(ip\pi\right)}{1-\exp\left(i\frac{p\pi}{N}\right)}-\exp\left(-i\frac{p\pi}{2N}\right)\frac{1-\exp\left(-ip\pi\right)}{1-\exp\left(-i\frac{p\pi}{N}\right)}\right](D_kC_l+C_kD_l)\\
&\quad+\frac{1}{2Ni}\left[\exp\left(i\frac{q\pi}{2N}\right)\frac{1-\exp\left(iq\pi\right)}{1-\exp\left(i\frac{q\pi}{N}\right)}-\exp\left(-i\frac{q\pi}{2N}\right)\frac{1-\exp\left(-iq\pi\right)}{1-\exp\left(-i\frac{q\pi}{N}\right)}\right](D_kC_l-C_kD_l)\\
&=\frac{1}{Ni}\left[\exp\left(i\frac{p\pi}{2N}\right)\frac{1}{1-\exp\left(i\frac{p\pi}{N}\right)}-\exp\left(-i\frac{p\pi}{2N}\right)\frac{1}{1-\exp\left(-i\frac{p\pi}{N}\right)}\right](D_kC_l+C_kD_l)\\
&\quad+\frac{1}{Ni}\left[\exp\left(i\frac{q\pi}{2N}\right)\frac{1}{1-\exp\left(i\frac{q\pi}{N}\right)}-\exp\left(-i\frac{q\pi}{2N}\right)\frac{1}{1-\exp\left(-i\frac{q\pi}{N}\right)}\right](D_kC_l-C_kD_l)
\end{aligned}$$

Divide to normalize the numerators,

$$\begin{aligned}
\text{LHS} &= \frac{1}{Ni}\left[\frac{1}{\exp\left(-i\frac{p\pi}{2N}\right)-\exp\left(i\frac{p\pi}{2N}\right)}-\frac{1}{\exp\left(i\frac{p\pi}{2N}\right)-\exp\left(-i\frac{p\pi}{2N}\right)}\right](D_kC_l+C_kD_l)\\
&\quad+\frac{1}{Ni}\left[\frac{1}{\exp\left(-i\frac{q\pi}{2N}\right)-\exp\left(i\frac{q\pi}{2N}\right)}-\frac{1}{\exp\left(i\frac{q\pi}{2N}\right)-\exp\left(-i\frac{q\pi}{2N}\right)}\right](D_kC_l-C_kD_l)\\
&=\frac{1}{Ni}\left[\frac{1}{-2i\sin\left(\frac{p\pi}{2N}\right)}-\frac{1}{2i\sin\left(\frac{p\pi}{2N}\right)}\right](D_kC_l+C_kD_l)\\
&\quad+\frac{1}{Ni}\left[\frac{1}{-2i\sin\left(\frac{q\pi}{2N}\right)}-\frac{1}{2i\sin\left(\frac{q\pi}{2N}\right)}\right](D_kC_l-C_kD_l)\\
&=\frac{1}{N}\left[\frac{1}{\sin\left(\frac{p\pi}{2N}\right)}\right](D_kC_l+C_kD_l)+\frac{1}{N}\left[\frac{1}{\sin\left(\frac{q\pi}{2N}\right)}\right](D_kC_l-C_kD_l). \qquad \text{(A.15)}
\end{aligned}$$

We will now consider the remaining coefficients. Using the definitions in Eq. (A.9), we have

$$D_kC_l=\sin\left(\frac{k\pi}{N}\right)\left(\cos\left(\frac{l\pi}{N}\right)-1\right)=\sin\left(\frac{k\pi}{N}\right)\cos\left(\frac{l\pi}{N}\right)-\sin\left(\frac{k\pi}{N}\right)$$

and, similarly,

$$C_k D_l = \sin\left(\frac{l\pi}{N}\right)\cos\left(\frac{k\pi}{N}\right) - \sin\left(\frac{l\pi}{N}\right).$$

Then, the remaining coefficients are

$$\begin{aligned} D_k C_l - C_k D_l &= \sin\left(\frac{k\pi}{N}\right)\cos\left(\frac{l\pi}{N}\right) - \sin\left(\frac{l\pi}{N}\right)\cos\left(\frac{k\pi}{N}\right) - \sin\left(\frac{k\pi}{N}\right) + \sin\left(\frac{l\pi}{N}\right) \\ &= \sin\left(\frac{(k-l)\pi}{N}\right) - \left[\sin\left(\frac{k\pi}{N}\right) - \sin\left(\frac{l\pi}{N}\right)\right] \end{aligned} \tag{A.16}$$

and

$$\begin{aligned} D_k C_l + C_k D_l &= \sin\left(\frac{k\pi}{N}\right)\cos\left(\frac{l\pi}{N}\right) + \sin\left(\frac{l\pi}{N}\right)\cos\left(\frac{k\pi}{N}\right) - \sin\left(\frac{k\pi}{N}\right) - \sin\left(\frac{l\pi}{N}\right) \\ &= \sin\left(\frac{(k+l)\pi}{N}\right) - \left[\sin\left(\frac{k\pi}{N}\right) + \sin\left(\frac{l\pi}{N}\right)\right]. \end{aligned} \tag{A.17}$$

Using $p = k + l$ and $q = k - l$, we obtain $k = \frac{1}{2}(p+q)$ and $l = \frac{1}{2}(p-q)$. With an eye to Eq. (A.16), we use these equations and the sine addition formula to write

$$\begin{aligned} \sin\left(\frac{k\pi}{N}\right) - \sin\left(\frac{l\pi}{N}\right) &= \sin\left(\frac{(p+q)\pi}{2N}\right) - \sin\left(\frac{(p-q)\pi}{2N}\right) \\ &= 2\sin\left(\frac{q\pi}{2N}\right)\cos\left(\frac{p\pi}{2N}\right). \end{aligned}$$

Then Eq. (A.16) becomes

$$D_k C_l - C_k D_l = \sin\left(\frac{q\pi}{N}\right) - 2\sin\left(\frac{q\pi}{2N}\right)\cos\left(\frac{p\pi}{2N}\right).$$

Similarly, we can also write

$$\begin{aligned} \sin\left(\frac{k\pi}{N}\right) + \sin\left(\frac{l\pi}{N}\right) &= \sin\left(\frac{(p+q)\pi}{2N}\right) + \sin\left(\frac{(p-q)\pi}{2N}\right) \\ &= 2\sin\left(\frac{p\pi}{2N}\right)\cos\left(\frac{q\pi}{2N}\right) \end{aligned}$$

to obtain

$$D_k C_l + C_k D_l = \sin\left(\frac{p\pi}{N}\right) - 2\sin\left(\frac{p\pi}{2N}\right)\cos\left(\frac{q\pi}{2N}\right).$$

We are finally ready to return to Eq. (A.15). We obtain

$$\begin{aligned}
\text{LHS} &= \frac{1}{N}\left[\frac{1}{\sin\left(\frac{p\pi}{2N}\right)}\right](D_kC_l + C_kD_l) + \frac{1}{N}\left[\frac{1}{\sin\left(\frac{q\pi}{2N}\right)}\right](D_kC_l - C_kD_l) \\
&= \frac{1}{N}\left[\frac{1}{\sin\left(\frac{p\pi}{2N}\right)}\right]\left(\sin\left(\frac{p\pi}{N}\right) - 2\sin\left(\frac{p\pi}{2N}\right)\cos\left(\frac{q\pi}{2N}\right)\right) \\
&\quad + \frac{1}{N}\left[\frac{1}{\sin\left(\frac{q\pi}{2N}\right)}\right]\left(\sin\left(\frac{q\pi}{N}\right) - 2\sin\left(\frac{q\pi}{2N}\right)\cos\left(\frac{p\pi}{2N}\right)\right) \\
&= \frac{1}{N}\left(\frac{\sin\left(\frac{p\pi}{N}\right)}{\sin\left(\frac{p\pi}{2N}\right)} - 2\cos\left(\frac{q\pi}{2N}\right)\right) + \frac{1}{N}\left(\frac{\sin\left(\frac{q\pi}{N}\right)}{\sin\left(\frac{q\pi}{2N}\right)} - 2\cos\left(\frac{p\pi}{2N}\right)\right). \qquad \text{(A.18)}
\end{aligned}$$

Using the sine addition formula,

$$\sin 2A = 2\sin A\cos B \implies \frac{\sin 2A}{\sin A} = 2\cos A,$$

we can re-express the quotients in Eq. (A.18) in terms of cosine to obtain

$$\begin{aligned}
\text{LHS} &= \frac{1}{N}\left(2\cos\left(\frac{p\pi}{2N}\right) - 2\cos\left(\frac{q\pi}{2N}\right)\right) + \frac{1}{N}\left(2\cos\left(\frac{q\pi}{2N}\right) - 2\cos\left(\frac{p\pi}{2N}\right)\right) \\
&= 0.
\end{aligned}$$

This completes the proof of (ii), i.e., $D_kC_l\langle\psi_k,\phi_l\rangle + D_lC_k\langle\psi_l,\phi_k\rangle = 0$ for $1 \le k,l \le N-1$.

Having proven properties (i) and (ii), Eq. (A.10) simplifies as follows,

$$\begin{aligned}
A_{kl} &= \langle D\phi_k, D\phi_l\rangle \\
&= C_kC_l\langle\phi_k,\phi_l\rangle - C_kD_l\langle\phi_k,\psi_l\rangle - D_kC_l\langle\psi_k,\phi_l\rangle + D_kD_l\langle\psi_k,\psi_l\rangle \\
&= (C_k^2 + D_k^2)\delta_{kl} \\
&= \left[\left(\cos\left(\frac{k\pi}{N}\right) - 1\right)^2 + \sin^2\left(\frac{k\pi}{N}\right)\right]\delta_{kl} \\
&= 4\sin^2\left(\frac{k\pi}{2N}\right)\delta_{kl}, \qquad \text{(A.19)}
\end{aligned}$$

where the final simplification results from applying the cosine double angle formula.

Together, Eqs. (A.5) and (A.19) establish the orthogonality result for all $0 \leq k \leq N-1$. This completes the proof. Once again, notice that this orthogonality constant is slightly different from that of the DFT case.

To summarize, we have shown that for a given $N > 0$ and $0 \leq k, l \leq N-1$,

$$\langle D\phi_k, D\phi_l \rangle = \begin{cases} 0, & k \neq l \\ A_k, & k = l, \end{cases}$$

where $A_0 = 0$ and $A_k = 4\sin^2\left(\frac{k\pi}{2N}\right) > 0$ for $1 \leq k \leq N-1$. ■