A Study of the Structural Similarity Image Quality Measure
with Applications to Image Processing

by

Dominique Brunet

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

Waterloo, Ontario, Canada, 2012

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

Since its introduction in 2004, the Structural Similarity (SSIM) index has gained widespread popularity as an image quality assessment measure. SSIM is currently recognized to be one of the most powerful methods of assessing the visual closeness of images. That being said, the Mean Squared Error (MSE), which performs very poorly from a perceptual point of view, still remains the most common optimization criterion in image processing applications because of its relative simplicity along with a number of other properties that are deemed important. In this thesis, some necessary tools to assist in the design of SSIM-optimal algorithms are developed. This work combines theoretical developments with experimental research and practical algorithms.

The description of the mathematical properties of the SSIM index represents the principal theoretical achievement in this thesis. Indeed, it is demonstrated how the SSIM index can be transformed into a distance metric. Local convexity, quasi-convexity, symmetries and invariance properties are also proved. The study of the SSIM index is also generalized to a family of metrics called normalized (or M-relative) metrics.

Various analytical techniques for different kinds of SSIM-based optimization are then devised. For example, the best approximation according to the SSIM is described for orthogonal and redundant basis sets. SSIM-geodesic paths with arclength parameterization are also traced between images. Finally, formulas for SSIM-optimal point estimators are obtained.

On the experimental side of the research, the structural self-similarity of images is studied. This leads to the confirmation of the hypothesis that the main source of self-similarity of images lies in their regions of low variance.

On the practical side, an implementation of local statistical tests on the image residual is proposed for the assessment of denoised images. Also, heuristic estimations of the SSIM index and the MSE are developed.

The research performed in this thesis should lead to the development of state-of-the-art image denoising algorithms. A better comprehension of the mathematical properties of the SSIM index represents another step toward the replacement of the MSE with SSIM in image processing applications.

Résumé

Depuis son introduction en 2004, l'indice de Similarité Structurelle (SSIM) a gagné en popularité comme mesure perceptuelle de la qualité d'image. En effet, SSIM est maintenant reconnu comme l'une des meilleures méthode pout comparer la distance perceptuelle entre deux images. Cependant, l'erreur quadratique moyenne reste le critère le plus répandu pour l'optimisation d'algorithmes de traitement d'image. Car malgré sa pauvre performance comme mesure perceptuelle, l'erreur quadratique moyenne est relativement simple et possède plusieurs propriétés mathématiques généralement jugées importantes. Dans cette thèse, plusieurs outils nécessaires pout aider la création d'algorithmes optimaux au sens de SSIM sont développés. Ce travail combine des résultats théoriques avec une recherche expérimentale ainsi que des algorithmes concrets.

La description des propriétés mathématiques de l'indice SSIM représente la principale contribution théorique dans cette thèse. En effet, il est démontré comment l'indice SSIM peut être transformé en une métrique. La convexité locale, la quasi-convexité ainsi que des symétries et des propriétés d'invariance sont aussi prouvées. L'indice SSIM est aussi généralisé en une famille de métriques dites normalisées ou M-relatives.

Plusieurs techniques analytiques pour différentes sortes d'optimisation au sens de SSIM sont ensuite introduites. Par exemple, la meilleure approximation au sens de SSIM pour des bases orthogonales ou redondantes est décrite. Les chemins géodésiques selon l'indice SSIM avec une paramétérisation selon la longueur d'arc sont aussi tracés entre deux images. Finalement, des formules pour trouver des estimateurs ponctuels selon SSIM sont obtenues.

L'étude de l'auto-similarité structurelle des images représente la facette plus expérimentale de la recherche. Cela permet de confirmer l'hypothèse que la principale source d'auto-similarité dans les images se situe dans les régions de faible variance.

Du côté pratique, l'implémentation de tests statistiques locaux sur l'image résiduelle est proposé pour l'évaluation de la qualité d'images débruitées. Aussi, des estimés heuristiques de l'indice SSIM et de l'erreur quadratique moyenne sont développés.

La recherche effectuée dans cette thèse devrait conduire au développement d'algorithmes de débruitage à la fine pointe de la technologie. La meilleure compréhension des propriétés mathématiques de l'indice SSIM représente une autre étape vers le remplacement de l'erreur quadratique moyenne par l'indice SSIM dans les applications de traitement d'image.

Acknowledgements

I am most grateful to all the persons who made this work possible.

First, I would like to thank my supervisor, Edward R. Vrscay, for his constant support through all the stages of production. He patiently helped to improve my writing style and the clarity of my ideas. His enthusiasm, his availability and his overall guidance are greatly appreciated. His approach to research in applied mathematics and in teaching simply represents an inspiration for me.

I am also grateful to have had the opportunity to be co-supervised by Zhou Wang. He shared is deep knowledge of image processing and image quality assessment with me and was always very resourceful. He pushed me to answer mathematical questions that arose from applications and to always find rigorous and elegant solutions to problems.

Finally, I am thankful to the examining committee, Jeff Orchard, Justin Wan, Paul Marriott and Hongmei Zhu, for accepting to review this thesis. Their various insights were much appreciated.

I must also gratefully acknowledge the financial assistance received from NSERC, OGS and the University of Waterloo in the form of scholarships. This material help is necessary to carry on the research without worries.

A special thank is dedicated to the incredible staff of the Department of Applied Mathematics. I will remember in particular Ms. Helen Warren for her smile and readiness to help even on short notice.

I would also like to thank Mrs. Alen Salvador for her help in the proofreading of the thesis. I am sure that it reduced much the workload of my supervisors and reviewers and allowed them to concentrate in the content. Salamat!

On a more personal note, I need to thank my dearest wife for enduring and encouraging me these last two years. I much appreciate her support and her patience with me. Gihigugma Tika!

Finally, I take this opportunity to acknowledge my Creator Who gave me everything: life, intelligence, strength, love and forgivenes. It gives me the true meaning of life and a purpose to do research that is not merely for myself and my glory, but rather to contemplate and share the beauty of the creation and, hopefully, to serve humanity.

Dedication

For His Glory...

Now God grant I speak suitably
and value these endowments at their worth:
For he is the guide of Wisdom
and the director of the wise.
For both we and our words are in his hand,
as well as all prudence and knowledge of crafts.
For he gave me sound knowledge of existing things,
that I might know the organization of the universe
and the force of its elements.

Ws 7:15-17

Table of Contents

List of Tables xi

List of Figures xii

1 Introduction 1

1.1 A Short Introduction to Digital Images 2

1.2 Noise Distortion Models 3

1.3 L^2-based Image Denoising 6

1.3.1 Deterministic Methods 8

1.3.2 Stochastic Methods 9

1.4 Image Quality Assessment 10

1.4.1 Applications of IQA 10

1.4.2 The SSIM Index 11

1.4.3 What's Wrong with L^2? 16

1.5 Thesis Outline 19

2 On Normalized Metrics 23

2.1 Ptolemaic Metric Spaces 24

2.2 Characterization of Normalized Metrics 26

2.2.1 Literature Review 27

2.2.2 Necessary and Sufficient Conditions 30

2.3 Vector-valued Metrics . . . 36
2.3.1 On Cone Metric Spaces . . . 36
2.3.2 Scalarization of Vector-valued Metrics . . . 37
2.4 The SSIM Index as a Normalized Metric . . . 39
2.4.1 Psycho-visual Validation of Normalized Metrics . . . 44

3 Properties of the SSIM Index 46
3.1 Invariance and Symmetry Properties . . . 46
3.2 Level Sets of the SSIM Index . . . 49
3.3 Convexity Properties . . . 53
3.3.1 Quasi-Convexity . . . 54
3.3.2 Region of Convexity . . . 56
3.3.3 Convexity of Cone Metrics . . . 60
3.4 Projection onto a Convex Set . . . 62

4 Geodesics of the SSIM Index 66
4.1 Geodesics: Preliminaries . . . 66
4.1.1 Definitions . . . 66
4.1.2 The Euler-Lagrange Equation . . . 67
4.2 Geodesics of Normalized Metrics . . . 69
4.2.1 General Case . . . 69
4.2.2 Case $M(r,r) = r$ . . . 72
4.2.3 Case $M(r,r) = r^q$ with $0 < q < 1$ . . . 76
4.2.4 Case $M(r,r) = \sqrt{r^2 + \epsilon^2}$ . . . 79
4.3 Geodesics of the SSIM Index . . . 86
4.3.1 SSIM-geodesic Distance . . . 88

5 Maximum SSIM Estimators 91

5.1 Classical Point Estimators 92

5.2 SSIM-Mean 94

5.2.1 One Dimensional Case 94

5.2.2 N-Dimensional Case 99

6 SSIM-based Approximation using Bases 109

6.1 L^2-based Approximation 110

6.1.1 Orthogonal Basis 110

6.1.2 Linear Redundant Basis 111

6.1.3 Non-Linear Approximation 113

6.2 SSIM-based Approximation 116

6.2.1 Linear Approximation 116

6.2.2 Non-Linear Approximation 121

6.3 Applications 123

6.3.1 Image and Video Coding 123

6.3.2 Image Denoising by Sparse Representation 125

7 A Study of Self-Similarity of Images 128

7.1 L^2-based Affine Model of Self-Similarity 128

7.2 SSIM-based Affine Self-Similarity Model 131

7.2.1 Optimal SSIM-based Affine Approximation 131

7.2.2 Relation between Optimal L^2- and SSIM-based Coefficients 135

7.3 Self-Similarity of Natural Images vs. Pure Noise Images 136

8 The Use of Residuals in Image Denoising 141

8.1 No-Reference Estimation of MSE, PSNR and SSIM 142

8.1.1 Estimating the Mean and Variance of the Signal and Noise 143

8.1.2 Estimating the Covariance of the Noise and the Residual 144

8.1.3 Partial-Reference IQA Estimates 146

8.1.4 Analysis of the Estimate of s_x 148

8.2 Statistical Tests on the Image Residual 151

8.2.1 Goodness-of-Fit Tests 154

8.2.2 Independence Tests 155

8.3 Image Denoising Boosting 160

8.3.1 Residual Filtering 160

8.3.2 Iterative Schemes and Statistical Tests 162

9 Conclusion and Future Work **163**

9.1 Summary of Contributions 163

9.2 Future Work 168

9.2.1 Fundamental Research 168

9.2.2 Applications 170

APPENDICES **177**

A Psychovisual Testing **177**

A.1 Methods for the comparison of IQA measures 179

A.2 IQA databases 181

B Matlab Codes **183**

References **201**

List of Tables

1.1 Prediction measures of MOS in the LIVE and TID-2008 databases for the PSNR and the SSIM index. 19

2.1 Correlation between MOS and image quality assessment measures. 45

7.1 Mean of the SSIM measure distributions of natural images, pure noise images with matched statistics and uniform pure noise images for self-similarity models Case 1 to 3 of *Lena* and *Mandrill* images. 138

9.1 Region of validity for the properties of the components of SSIM index and of the SSIM metric. 165

A.1 Description of the size of psycho-physical various databases 182

A.2 Type of distortion in various psycho-physical databases. 182

List of Figures

1.1 Original noiseless image, noise image, noisy image, denoised image and residual image. 7

1.2 Comparison between distorted images with the same L^2-distance. 17

1.3 Comparison of the SSIM maps between distorted images with the same L^2-distance. 18

1.4 Results of psychovisual testing of the Peak-to-Noise Signal Ratio (PSNR) and of the Structural Similarity (SSIM) index. 22

2.1 Comparison of the SSIM index and of the metric $D_{2,2}$ for the distorted images of the TID-2008 database 45

3.1 Level sets of MSE. 51

3.2 Level sets of SSIM with $\beta = 1$. 51

3.3 Level sets of SSIM with $\beta = 2$. 51

3.4 Level sets of SSIM with $\beta = 0.5$. 51

3.5 Level sets of pq-normalized metric with $p = 16$, $q = 1$ and $c = 0$. 52

3.6 Level sets of pq-normalized metric with $p = 1/4$, $q = 1$ and $c = 0$. 52

3.7 Points $x_{(1)}$, $x_{(2)}$, . . . , $x_{(6)}$ and z in the proof of Lemma 3.3.9. 57

3.8 Boundary of the region where $g(x) = \mathrm{NMSE}_c(x, y)$ is convex with $y = (1, 0)$ and $x = (x_1, x_2)$ for $c = 0$, $c = 0.1$ and $c = 1$. 59

4.1 Geodesics for $M(r, r) = r^q$. 80

4.2 Geodesics and mid-points for $M(r, r) = \sqrt{r^2 + C^2}$. 87

4.3 Geodesics for $M(r,r) = \sqrt{r^2+\epsilon^2}$ and for $M(r,r) = r^q$. 88

4.4 Image geodesics between Barbara (low luminance and contrast) and Lena according to the SSIM index and the Euclidean metric (linear path). 90

5.1 Plot of the one dimensional sum of the structural similarity between two points x_1 and x_2 for stability constant $\epsilon_1 = 0$ 97

5.2 The function $f(z) = \frac{2x_1z+C}{x_1^2+z^2+C} + \frac{2x_2z+C}{x_2^2+z^2+C}$ with $x_1 = 0.1$ and $x_2 = 0.4$ and its extrema. 104

5.3 The function $f(z) = \frac{2x_1z+C}{x_1^2+z^2+C} + \frac{2x_2z+C}{x_2^2+z^2+C}$ with $x_1 = 0.05$ and $x_2 = 0.8$ and its extrema. 104

5.4 Optimal path between $x_1 = 0.1$ and $x_2 = 0.4$ according to different metrics. 105

5.5 Optimal path between $x_1 = 0.05$ and $x_2 = 0.8$ according to different metrics. 105

5.6 Comparison between the arithmetic mean and the S_2-mean (SSIM without the mean distortion component) of points in $\mathbf{R}^2$. 106

5.7 Global optimization of the norm r in the computation of the S_2-mean. . . 106

5.8 Comparison between the arithmetic mean and the SSIM-mean of positively correlated points in $\mathbf{R}^2$. 107

5.9 Global optimization of the norm r and the mean component b in the computation of the SSIM-mean for the points in Fig. 5.8. 107

5.10 Comparison between the arithmetic mean and the SSIM-mean of points in $\mathbf{R}^2$ that do not have positively correlated zero-mean components. 108

5.11 Global optimization of the norm r and the mean component b in the computation of the SSIM-mean for the points in Fig. 5.10. 108

7.1 Case 1 to 3 RMS Δ-error distributions for normalized *Lena* and *Mandrill* images over the interval $[0, 0.5]$. 132

7.2 Case 1 to 3 SSIM measure distributions for normalized *Lena* and *Mandrill* images over $[-1, 1]$. 134

7.3 Images made of uniform noise with statistics matching the local mean and variance of natural images: (a) *Lena* (b) *Mandrill* (c) Noise image with each pixel value taken from an uniform distribution on [0,1]. 137

7.4 Comparison of RMS Δ-error distribution of *Lena* and *Mandrill* for Case 1 to 3 with the RMS Δ-error distribution of pure noise images for which the local mean and local variance are matched and with the RMS Δ-error distribution of an i.i.d. uniform pure noise image. 139

7.5 Comparison of SSIM measure distribution of *Lena* and *Mandrill* for Case 1 to 3 with the SSIM measure distribution of pure noise images for which the local mean and local variance are matched and the SSIM measure distribution of an i.i.d. uniform pure noise image 140

8.1 Histograms of global covariance for the images in the training set. (a) $s_{x,n}$ (b) $s_{r,x}$ (c) $s_{r,x} - s_{r,d}$ (d) $s_n^2 - s_{r,n}$. 145

8.2 (a) Scatter plot of the PSNR and its no-reference estimates. (b) Scatter plot of the SSIM index and its no-reference estimates. 149

8.3 (a) SSIM map between the original image *Lena* and its denoised version. (b) Partial-Reference Estimate of the SSIM map. 150

8.4 (a) Local variance of upper left corner of the original *Barbara* image. (b) Local variance of noisy *Barbara* image. (c) Rough estimate of the local variance of the original *Barbara* image. (d) Estimation of the local variance of the original *Barbara* image from random sampling. (e) Absolute error map of the rough estimate. (f) Absolute error map of the random sampling estimate. 152

8.5 Scalar estimate of the local standard deviation of the original *Barbara* image from the local standard deviation of the noisy *Barbara* image 153

8.6 Residual image, its local Kolmogorov-Smirnov statistics and the result of the K-S test 155

8.7 Local statistics for the computation of local independence tests between **d** and **r** and result of local independence tests between **d** and **r** at a significance level $\alpha = 0.05$. 158

8.8 Result of local autocorrelation test between **r** and its horizontal and vertical shifts 159

8.9 Local mutual information between **r** and **d** and result of the *G*-test 160

8.10 (a): *Boat* denoised by Total Variation Minimization. (b): Residual image. (c): Residual “denoised” by Adaptive Wiener Filter. (d): Enhanced denoised image. 161

Chapter 1

Introduction

Images. They convey information, they convey emotions. But what happens if these images are distorted? Image distortion could be from a noisy channel over the internet or from an overzealous compression. Or simply, it could be caused by the inevitable effect of time. When this happens the crucial spatial data becomes useless. The enjoyment is gone. But hopefully not everything is lost.

By the power of mathematics, several algorithms have been devised to restore images, i.e. to attempt to recover an image that will be visually pleasant. Formally, they have been divided into categories according to the kind of distortion they take care of: noise, blur, blotching, down-sampling or compression artifacts.

One kind of distortion that is of particular interest to this thesis is noise. Similar to the notion of audible noise, visual noise is the spurious variation of intensity and colors in an image. It is casually referred as “snow” in an analog television.

The first reason why we are particularly interested in noise is because it is always present in the physical world; any measurement is subject to some kind of uncertainty and variation. Moreover, many subsequent image processing and analysis are severely impeded by the presence of noise. For example, there are the problems of image segmentation or image registration.

Another reason for studying image de-noising, i.e. the restoration of noisy images, is because it provides a benchmark to test natural image models. Indeed, several years of research have not been enough to bring a complete understanding of what images are. It is understood that they are acquired from the real world by some digital cameras, but the question is how to fully describe the image statistics. That is, given any combination of pixel values (supposing an already sampled image), can we find the probability that this image could be taken by a digital camera at any time in history?

Finally, given that image denoising is probably the easiest of the image restoration problems, it is a natural starting point before attacking more involving problems like image deblurring for example. Easiest does not necessary mean easy, as there has been limited success up to now from any research to design good denoising algorithms.

This limited success could be explained in part by the fact that most image denoising algorithms are either *ad hoc* or based on the optimization of the mean squared error (MSE). The problem is that MSE does not simulate well the perception of quality of the human visual system (HVS). The structural similarity (SSIM) index proposed by Wang et al. [114] in 2004 appears to be a better alternative. This leads to the question on how to perform SSIM-optimal image denoising and processing.

The goal of this thesis is thus to provide the necessary mathematical tools for the performance of perceptually optimal image processing. To do so, we will need to study the properties of the SSIM index and show how this perceptual measure can be optimized. As an example of application, we will propose image denoising algorithms with a perceptual criterion based on the SSIM index.

The remainder of this introduction is divided as follows: First, we will state and define the basic concepts of mathematical imaging. Second, two standard models of noise will be developed. After, we will briefly sketch how the MSE is used in image denoising problems. The SSIM index will then be presented and compared to the MSE as a model of image perception through examples and psycho-physical experiments. Finally, we will outline the content of the whole thesis.

1.1 A Short Introduction to Digital Images

A *natural image* is the spatial measurement of some feature in the real world. By "real world", we mean the physical environment that surrounds us. Of particular interest is the measurement of electromagnetic waves of different wavelengths (e.g. X-rays, infrared, ultraviolet, etc.). Here, we are mainly interested in measuring the visible light acquired from a digital camera.

We briefly summarize the acquisition process of digital images in the remaining of this section. For a more thorough introduction the reader is referred to [54].

In the current design of digital cameras, an array of charge-coupled device (CCD) receptors measure the amount of incoming photons as a difference of voltage. This array is generally organized in a mosaic of receptors for red, green and blue (RGB) components of the light. A demoisacking algorithm then interpolates the missing values into an array of pixels (picture elements).

An Analog-to-Digital converter quantizes the amount of possible values for each components to a finite number. With today's digital cameras, the convention is 8 bits-per-pixel for each component, i.e. 256 possible values for the RGB components and a total of 24 bits-per-pixel for the set of the three components. This is approximately 16 million different colors. At the end, we obtain three matrices representing the amount of RGB at each location of the image. To simplify the problem, each RGB component of the digital image is transformed into a luminance and two chrominance channels. The value of each image is normalized in the interval $[0, 1]$.

For most of the work in this thesis, we will be interested solely in greyscale images. That is, we will only consider the luminance channel. Thus, we will manipulate matrices with values between 0 and 1 where 0 is black and 1 is white. Note that these values for black and white are not necessarily universal but depend on the dynamic range of a given image. With indoor images for example, the given value for white might be much lower than with outdoor images.

Let $I = \{1, 2, \ldots, L_1\} \times \{1, 2, \ldots, L_2\}$ be the set of indices of the pixel values for an image of $L_1 \times L_2$ pixels. Let $\mathbf{x}(I)$ be a $L_1 \times L_2$ real-valued matrix representing the intensity values (or *greyscale*) at each pixel. For a mega-pixel (one million pixels) image, the dimension of a square image will be $L_1 = L_2 = 1024$. This is considered a medium-sized image. Today digital cameras have up to sixteen mega-pixels. That is, $L_1 = L_2 = 4096$ for every square image.

Here, we will consider small images, i.e. $L_1 = L_2 = 256$ or $L_1 = L_2 = 512$. Note that the only practical difference between images of different sizes is the computation and memory cost to process them. For example, if the covariance between neighbouring pixels is needed at each location, then we have to compute and store $(L_1 L_2)^2$ values, which represent one trillion values for a mega-pixel image.

1.2 Noise Distortion Models

Noise in images has several sources. Given the quantum nature of light, a discrete amount of photons excites the photovoltaic receptor of the CCD camera. A natural variation occurs particularly in the low luminance region of an image and thus creates noise. This source of distortion is called *shot noise* or *quantum noise*, and is related to the physics of light itself and not on the particular device that acquires an image. Also, within the camera, the interference of its electronic circuit can influence the accuracy of the measurements. This interference is referred as the *read out noise*. Finally, the heating of the photovoltaic cells can also lead to a variation in measurements. This is the effect of *thermal noise*.

The most common models in literature for noise in photographic images (see e.g. [5]) are *Poisson noise* and *additive white Gaussian noise* (AWGN). The main source of distortion is the shot noise which is well modeled by a Poisson distribution. Other sources of noise in digital cameras are modeled either as a Gaussian or Poisson distributions, but are negligible on high illumination situations.

The *Poisson distribution* describes the probability of occurrence of discrete and rare events. For photographic images, it represents well the amount of incoming photons on a photo-receptor. The conditional probability law of a measured pixel y given a pixel of mean illumination x is written as

$$\mathrm{Prob}(\mathscr{Y} = y | \mathscr{X} = x) = \frac{x^y e^{-x}}{y!}. \tag{1.1}$$

Note that both the mean and the variance of this distribution are x. For the whole image, assuming independence between adjacent pixels, the probability distribution is

$$\mathrm{Prob}(\mathscr{Y} = \mathbf{y} | \mathscr{X} = \mathbf{x}) \;=\; \prod_{k=1}^{N} \frac{x_k^{y_k} e^{-x_k}}{y_k!}, \tag{1.2}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is the mean illumination of the image and $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ is the measured image written in a vector form.

In high illumination situations, the Poisson distribution is well approximated by a Gaussian distribution. We can thus consider an AWGN model, i.e. an independent and identically distributed Gaussian noise. Assume that $\mathbf{x}$ belongs to the set of natural images $\mathscr{X}$, i.e. images that are likely to be seen by the human visual system (HVS). Let I be the index set of the pixels of an image and let $N = L_1 L_2$ be its number of pixels. Let $\mathscr{N}$ be an independent Gaussian noise of zero-mean and of variance σ^2. More precisely, the following assumptions can be made:

1. $\mathscr{X}$ and $\mathscr{N}$ are independent random fields;
2. $\mathscr{N}(i)$ and $\mathscr{N}(j)$ are independent for $i \neq j$ $(i, j \in I)$;
3. The noise follows a *Gaussian distribution*.

Given a fixed level of noise σ, we thus have for a single pixel

$$\text{Prob}(\mathscr{Y} = y|\mathscr{X} = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-x)^2}{2\sigma^2}} \tag{1.3}$$

and

$$\begin{aligned} \text{Prob}(\mathscr{Y} = \mathbf{y}|\mathscr{X} = \mathbf{x}) &= \frac{1}{(2\pi\sigma^2)^{N/2}} \prod_{k=1}^{N} e^{\frac{-(y_k-x_k)^2}{2\sigma^2}} & (1.4) \\ &= \frac{1}{(2\pi\sigma^2)^{N/2}} e^{\frac{-\|\mathbf{y}-\mathbf{x}\|_2^2}{2\sigma^2}} & (1.5) \end{aligned}$$

for the whole image. We can then write the noise distortion model as $\mathscr{Y} = \mathscr{X} + \mathscr{N}$; a particular outcome will be represented by $\mathbf{y} = \mathbf{x} + \mathbf{n}$.

Notice that we assumed that natural images are bounded but that the effect of the noise can lead some pixels to be outside the range of $[0, 1]$. This is called the saturation effect. To reduce the effect of clamping of these extreme values, one normally processes the noisy images without correcting the values outside the normal range. It is only for visualization and storage that clamping will be necessary. In fact, we will save the value of the seed parameter in the pseudo-random number generator to allow completely reproducible results and avoid any distortion due to the storage of a noisy image affected by the saturation effect.

The following Matlab code function `gennoisy` loads an original image ’name’ and then generates a noisy image with AWGN.

```
function [Y]  = gennoisy(name);
%% Generate noisy image
X = imread(name); % load the image ’name’
X = double(X)/255; % convert to double representation and normalize
seed = 0; % initialize seed
randn(‘state’, seed); % set the state of the pseudo-random number generator
sigma = 0.25; % standard deviation of the noise
N = randn(size(X))*sigma; % Gaussian noise image
Y = X + N; % noisy image
```

1.3 L^2-based Image Denoising

Given a noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}$, the goal of a *denoiser* is to find a good approximation to the noiseless image $\mathbf{x}$. For any noisy image $\mathbf{y}$, the action of a denoiser may be viewed as separating $\mathbf{y}$ into the sum of a denoised image $\mathbf{d} = \mathscr{D}(\mathbf{y})$ and a *residual image* defined by $\mathbf{r} = \mathbf{y} - \mathbf{d}$. To illustrate this particular outcome, in Fig. 1.1 we show the original noiseless *Barbara* image $\mathbf{x}$ (512 × 512 pixels) along with a noise image $\mathbf{n}$ and the resulting noisy image $\mathbf{y} = \mathbf{x}+\mathbf{n}$. A denoised image $\mathbf{d} = \mathscr{D}(\mathbf{y})$ obtained by Gaussian filtering is then shown along with the residual image $\mathbf{r} = \mathbf{y} - \mathbf{d}$.

The Matlab code is pretty straightforward and very high level. Note that the cropping allows us to work around the problem of boundary handling.

```
function [D] = gdenoise(Y)
%% Denoise by Gaussian filtering
g = fspecial('gaussian',[31 31],2.5); % create the Gaussian filter
D = filter2(g,Y,'same');   % filter the noisy image
R = Y - D;  % residual image

%% Crop the image
crop1 = 100:400; crop2 = 100:400; % cropping window
D = D(crop1,crop2);
```

It is then possible to compute the error of approximation between the original and the denoised image. The most popular way to do this is with the *Peak-Signal-to-Noise Ratio* (PSNR), which is simply a remapping of the *Mean Squared Error* (MSE) between two normalized images:

$$\mathrm{MSE}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{L_1 L_2} \sum_{i_1=1}^{L_1} \sum_{i_2=1}^{L_2} (\mathbf{x}(i_1, i_2) - \hat{\mathbf{x}}(i_1, i_2))^2, \tag{1.6}$$

$$\mathrm{PSNR}(\mathbf{x}, \hat{\mathbf{x}}) = -10 \log_{10} \mathrm{MSE}(\mathbf{x}, \hat{\mathbf{x}}). \tag{1.7}$$

Although digital images are discrete, we sometimes assume in our mathematical construction that they are continuous. It simplifies the treatment of the theory and makes better sense as the real world appears continuous and is generally modeled as such in classical physics. For example, we will refer to the L^p-*norm* between two images on $\Omega \subset \mathbf{R}^2$,

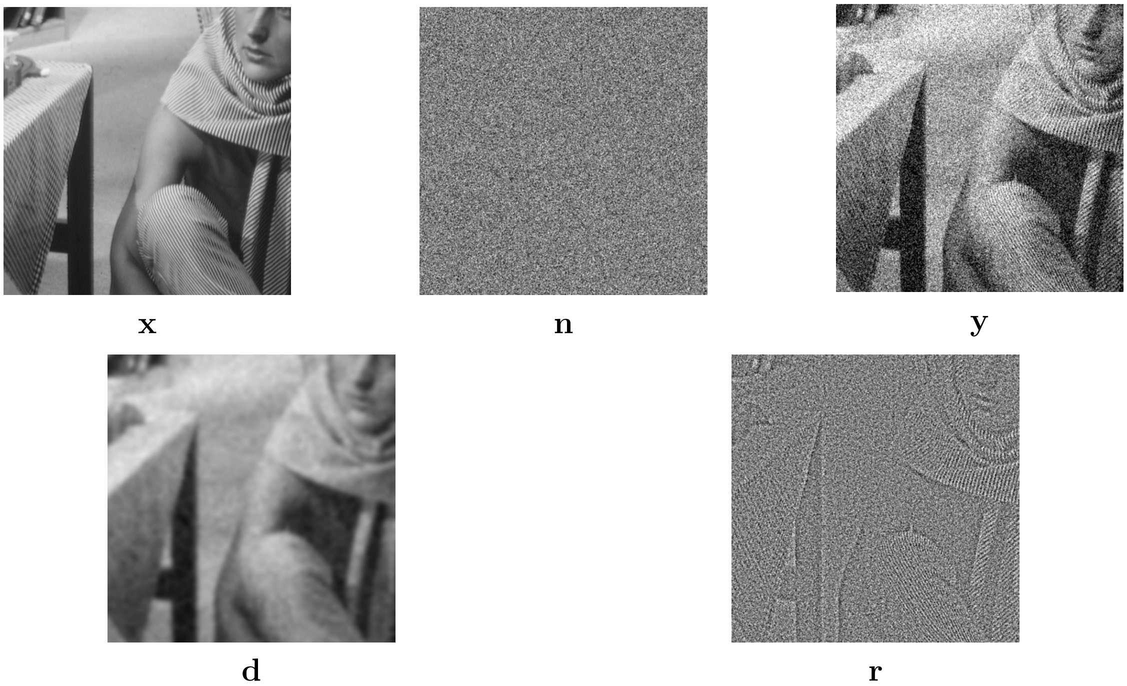


Figure 1.1: $\mathbf{x}$: Original noiseless *Barbara* image. $\mathbf{n}$: a noise image ($\sigma = 15\%$). $\mathbf{y} := \mathbf{x} + \mathbf{n}$, the noisy image. $\mathbf{d} := \mathscr{D}(\mathbf{y})$, the denoised image (Gaussian filtering). $\mathbf{r} := \mathbf{y} - \mathbf{d}$, the residual image. All images are cropped between $(100, 100)$ and $(400, 400)$ for better visualization and to escape boundary handling problems. Notice that $\mathbf{y} = \mathbf{x} + \mathbf{n}$ and $\mathbf{y} = \mathbf{d} + \mathbf{r}$ are two different decompositions of the noisy image $\mathbf{y}$.

which is

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_p = \left(\int_\Omega |\mathbf{x}(s,t) - \hat{\mathbf{x}}(s,t)|^p ds dt \right)^{1/p}. \tag{1.8}$$

Note that the squared L^2-norm divided by the size of the domain Ω is equivalent to the MSE.

The link between the discrete and the continuous world is a piecewise constant function on small square regions. In what follows, we will use L^2-norm and the MSE interchangeably, but we will generally assume that images belong to $\mathbf{R}^N$ (with $N = L_1 L_2$) rather than $L^2(\mathbf{R}^2)$. More precisely, we will assume in several instances that images belong to the positive octant

$$\mathbf{R}_+^N := \{\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbf{R}^N | x_k \geq 0 \text{ for all } 1 \leq k \leq N\} \tag{1.9}$$

since the amount of light coming from each location cannot be negative. To summarize, natural images are well approximated by functions in $L^2(\mathbf{R}^2)$ but we will consider only their sampled version in $\mathbf{R}_+^N$. For a particular instance of an image, we will further normalize this into the hypercube $[0, 1]^N$ for display and storage.

Since the L^2-norm or one of its equivalent formulations is the traditional measure of quality of an image denoising procedure, it is natural to attempt to optimize algorithms in function of this norm. We divide L^2-based image denoising algorithms in two categories depending on whether they are deterministic or stochastic.

1.3.1 Deterministic Methods

Several deterministic algorithms are based on variational models. Variational models assume that images belong to some space of functions $\mathscr{F}$ with norm $\|\mathbf{x}\|_{\mathscr{F}}$. The smaller the norm, the more natural the image is considered. Depending on how much noise is tolerated, a constraint of the type $\|\mathbf{x} - \mathbf{y}\|_p \leq s$ is imposed, with $p = 2$ for Gaussian noise. This inequality reflects that the denoised image is expected to be close to the noisy data. Using a suitable Lagrange multiplier, the problem is thus formulated as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathscr{F}} \|\mathbf{x}\|_{\mathscr{F}} + \lambda \|\mathbf{x} - \mathbf{y}\|_2. \tag{1.10}$$

The Lagrange multiplier λ weighs between the "smoothness" of the reconstructed image and its goodness-of-fit to the data. The method generally employed to find the optimal solution is to solve the associated Euler-Lagrange equations. This often leads to a partial differential equation that has to be solved numerically.

Examples of image denoising based on variational models include spline smoothing [108] and total variation minimization [94]. In these formulations, the L^2-norm accounts for the Gaussian nature of the noise, not for the model of perception, so we cannot really call these methods true L^2-based denoising algorithms.

On the other hand, any model involving an economical representation of an image can be adapted into a L^2-based variational model by computing the error of approximation and adding a regularization term. Given $T(\mathbf{x})$ a representation operator, the variational problem is formulated as follows:

$$\mathbf{d} = \mathscr{D}(\mathbf{y}) \quad = \quad \arg\min_{\mathbf{x}} \|\mathbf{x} - T(\mathbf{x})\|_2^2 + \lambda\|\mathbf{x} - \mathbf{y}\|_2^2. \tag{1.11}$$

Note that term $\|\mathbf{x} - T(\mathbf{x})\|_2$ is commonly called the collage error in fractal imaging literature.

Examples of possible operator T include sparse dictionary representation [40], non-local means [17] and fractal transforms [39]. It is to be remarked that some of these operators can also be used for image compression. It could be explained by the fact that to be compressible is an intrinsic property of images whereas a fundamental discriminating factor of noise is that it is not compressible by nature.

1.3.2 Stochastic Methods

In contrast to deterministic methods, stochastic methods assume that the original image and image distortion follow a certain probability density function (p.d.f.). In Bayesian language, the p.d.f. of the original image $\mathscr{P}(\mathbf{x})$ is the prior whereas the conditional p.d.f. of the distorted image knowing the original image $\mathscr{P}(\mathbf{y}|\mathbf{x})$ is the likelihood distribution. For Poisson or Gaussian noise, this distribution was described in the previous section. By Bayes' formula, the posterior distribution of $\mathbf{x}$ knowing the data $\mathbf{y}$ is proportional to the prior distribution multiplied by the likelihood:

$$\mathscr{P}(\mathbf{x}|\mathbf{y}) \propto \mathscr{P}(\mathbf{y}|\mathbf{x})\mathscr{P}(\mathbf{x}).$$

A Bayesian risk minimizer will then be the best estimate of the denoised image according to a certain loss function $d(\cdot,\cdot)$. It is measured with a conditional expectation:

$$\begin{aligned}\mathbf{d} = \mathscr{D}(\mathbf{y}) \quad &= \quad \arg\min_{\hat{\mathbf{x}}} \mathbf{E}[d(\mathbf{x}, \hat{\mathbf{x}})|\mathbf{y}] &(1.12)\\ &:= \quad \arg\min_{\hat{\mathbf{x}}} \int d(\mathbf{x}, \hat{\mathbf{x}})\mathscr{P}(\mathbf{x}|\mathbf{y})d\mathbf{x}. &(1.13)\end{aligned}$$

For L^2-based denoising, the loss function is quadratic and, as it is well known, the best estimator will be the expectation of the posterior distribution $\mathbf{E}[\mathbf{x}|\mathbf{y}]$. This is called the Minimum Mean Square Estimator (MMSE).

Examples of stochastic L^2-based methods are the Wiener filter, wavelet shrinkage [35], wavelet coring [103] and Gaussian Scale Mixture (GSM) [81]. A less known fact is that the non-local means algorithm [16] can also enter into this category.

1.4 Image Quality Assessment

The goal of *image quality assessment* (IQA) is to model the perception of the human visual system (HVS) in order to evaluate the subjective quality of images. IQA can be divided into *full-reference* and *no-reference*, depending on the availability of the original (ground truth) image. For full-reference IQA, we should speak of image fidelity rather than image quality, since the goal is to measure the similarity between two images. For no-reference IQA, one models the subjective IQA from only the knowledge of a distorted image. This is closer to real-world situations where the original image is not available.

1.4.1 Applications of IQA

In this work, we will be mainly interested in full-reference IQA. This will involve design, optimization and comparison of various image restoration algorithms.

For example, given several proposed denoising algorithms, a simulation can be run:

1. Generate a pseudo-random noise on several original images for different levels of noise;
2. Apply different image denoising algorithms;
3. Compute the quality of the denoised images versus the original images for each combination of images, noise level and denoising algorithms;
4. For each noise level, perform a statistical test to determine if the difference in performance between each pair of algorithms is significant.

Also, IQA measures can be used to train the denoising parameters. This will boil down to multi-valued optimization of an objective function. The objective function will be a combination of all the IQA measures for a training set of images. For example, if $q(\cdot,\cdot)$ represents the full-reference IQA measure, objective functions having the following form may be considered:

$$\mathrm{obj}_{\{\mathbf{x}_j\}_{j=1}^J,\{\mathbf{y}_j\}_{j=1}^J}(c_1,c_2,\ldots,c_k)=\sum_{j=1}^{J} q^r(\mathbf{x}_j,\mathscr{D}_{c_1,c_2,\ldots,c_k}(\mathbf{y}_j)),$$

where $r > 0$ is a fixed exponent, J is the number of images in the training set, $\mathbf{x}_j$ is the j-th original image and $\mathbf{y}_j$ is the j-th distorted image, where $(c_1, c_2, \ldots, c_k)$ are the parameters of the denoising operator $\mathscr{D}$.

Finally, and this will be our principal interest in this thesis, we can take advantage of a full-reference IQA measure to help in the design of image restoration algorithms. For variational problems, take $q(T\mathbf{x}, \mathbf{x})$ as the *collage error*:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathscr{F}} q(T\mathbf{x}, \mathbf{x}) - \lambda\|\mathbf{x} - \mathbf{y}\|_2. \tag{1.14}$$

For the Bayesian formulation, we would maximize the expected image quality measure:

$$\hat{\mathbf{x}} = \arg\max_{\tilde{\mathbf{x}}} \mathrm{risk}(\mathbf{x}, \tilde{\mathbf{x}}) = \arg\max_{\tilde{\mathbf{x}}} \mathbf{E}[q(\mathbf{x}, \tilde{\mathbf{x}})|\mathbf{y}]. \tag{1.15}$$

1.4.2 The SSIM Index

In the classical approach for IQA, researchers attempt to model every single component of the HVS. This is called a bottom-up approach: we start with simple features of the HVS and combine them together until we reach a realistic model of image quality perception.

The two most famous models following this approach are Daly's Visual Difference Predictor [32] and Lubin's Just Noticeable Difference [66]. Both models follow the same general framework as described in [114] and summarized in [8].

The *Structural Similarity* (SSIM) index was recently proposed by Wang et al. [114] as an alternative to the bottom-up approach. It combines simplicity of computation, accuracy of prediction and intuitiveness of design.

The SSIM allows to quantify local image quality. Given a pixel location $i = (i_1, i_2)$ in the set of coordinates $\mathbf{Z}^2$, its local neighbourhood of radius r is defined as $\mathscr{N}_i = \{j \in \mathbf{Z}^2 : \|i - j\|_1 \leq r\}$. The pixels on a local neighbourhood are referred as a sub-block or an image patch.

For two image patches $\mathbf{x}$ and $\mathbf{y}$ in $\mathbf{R}^n_+$, the SSIM is defined as

$$\begin{aligned} \mathrm{SSIM}(\mathbf{x}, \mathbf{y}) &= \left(\frac{2\bar{x}\bar{y} + \epsilon_1}{\bar{x}^2 + \bar{y}^2 + \epsilon_1}\right)^\alpha \left(\frac{2s_x s_y + \epsilon_2}{s_x^2 + s_y^2 + \epsilon_2}\right)^\beta \left(\frac{s_{x,y} + \epsilon_3}{s_x s_y + \epsilon_3}\right)^\gamma, & (1.16)\\ &= (l(\mathbf{x}, \mathbf{y}))^\alpha (c(\mathbf{x}, \mathbf{y}))^\beta (s(\mathbf{x}, \mathbf{y}))^\gamma, & (1.17)\end{aligned}$$

where $\alpha, \beta, \gamma > 0$ are weights on the luminance (l), contrast (c) and structure term (s) of the fidelity measure and ϵ_1, ϵ_2 and ϵ_3 are small positive stability constants. The scores are computed locally with a weighted average:

$$\bar{x} = \sum_{j=1}^{n} w_j x_j, \qquad \bar{y} = \sum_{j=1}^{n} w_j y_j, \tag{1.18}$$

$$s_x^2 = \sum_{j=1}^{n} w_j (x_j - \bar{x})^2, \qquad s_y^2 = \sum_{j=1}^{n} w_j (y_j - \bar{y})^2, \tag{1.19}$$

$$s_{x,y} = \sum_{j=1}^{n} w_j (x_j - \bar{x})(y_j - \bar{y}), \tag{1.20}$$

with the weights $\{w_j\}_{j=1}^{n}$ such that $\sum_{j=1}^{n} w_j = 1$ taken from a (truncated) circular Gaussian window of total size n. The SSIM map is the matrix composed of all these local scores. It gives information on the location of distortion in images. A single index can then be obtained by averaging the local scores. For example, starting from the original *Barbara* image and the distorted images found in Fig. 1.2 various SSIM maps and indices are obtained in Fig. 1.3.

A commonly used simplification of the structural similarity index is to take $\alpha = \beta = \gamma = 1$ and $\epsilon_3 = \epsilon_2/2$. The formula then reduces to

$$\begin{aligned} \mathrm{SSIM}(\mathbf{x}, \mathbf{y}) &= \left(\frac{2\bar{x}\bar{y} + \epsilon_1}{\bar{x}^2 + \bar{y}^2 + \epsilon_1} \right) \left(\frac{2 s_{x,y} + \epsilon_2}{s_x^2 + s_y^2 + \epsilon_2} \right), & (1.21) \\ &= S_1(\mathbf{x}, \mathbf{y}) S_2(\mathbf{x}, \mathbf{y}). & (1.22) \end{aligned}$$

We ought to make a few comments about this formula, either to point out its link with the HVS or to constructively criticize its weaknesses. A general introduction to the SSIM index and its extensions can be found in [113].

The general philosophy behind the SSIM index is to distinguish structural distortions from non-structural distortions. Structural distortions are the ones that affect the most the perception of degradation of an image, whereas non-structural distortions only slightly affect the perception of degradation. The two first terms of the SSIM index (l and c) thus account for non-structural distortions, while the structural distortion (s) is represented by a loss of linear correlation.

As it is explained in [114], the luminance term of the SSIM index is related to *Weber's Law* (see e.g. [110]): the perception of the change of a stimulus is proportional to the intensity of the stimulus. Weber's Law may therefore be expressed as follows:

$$\frac{\Delta I}{I} \;=\; C, \tag{1.23}$$

where I is the intensity, ΔI is the change of intensity and C is a constant. Given an image $\mathbf{x}$ of luminance intensity I_x, the intensity I_y of the luminance of an image $\mathbf{y}$ for the difference to be just perceptible must be

$$\frac{I_y - I_x}{I_x} \;=\; C\;\;. \tag{1.24}$$

This implies that for any *just noticeable distortion* (JND), the ratio I_y/I_x is constant. For zero stability constant, this property applies for the first component of the SSIM index. Indeed,

$$\frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} = k \iff \frac{\bar{y}}{\bar{x}} = \frac{1 \pm \sqrt{1-k^2}}{k}, \tag{1.25}$$

thus the ratio of luminance intensity is constant for each distortion level. Since the property holds not only for JND, but also for suprathreshold distortions, the SSIM index extends Weber's Law.

Weber's Law not only applies when $I_x = \bar{x}$, the luminance, but also with $I_x = s_x$, the image contrast. We reach the same conclusion that, for zero stability constants, the ratio of contrasts s_x/s_y is constant for a constant SSIM distortion level.

The constants ϵ_1, ϵ_2 and ϵ_3 account for the threshold of perception. Indeed, Weber's Law breaks down for small intensities, since there is a minimum absolute change of intensity that needs to observed. Note the similarity with the divisive normalization model [109], where there is a constant in the denominator. In [114], the constants are set empirically as $\epsilon_1 = 0.01^2$, $\epsilon_2 = 0.03^2$ and $\epsilon_3 = \epsilon_2/2$ for an image normalized to $[0,1]$.

The sliding window $\{w_j\}_{j=1}^n$ can be related to the functioning of the HVS where most of the receptors are concentrated in the fovea and eye movement is necessary to acquire a complete picture of the surrounding environment. In [114], a 11 × 11 truncated Gaussian window of standard deviation of 1.5 pixels is taken. In order to model more closely the functions of the HVS, it might be better to use a sliding window following the true distribution of the cones which can be modelled as a Laplacian distribution.

For the pooling of the local scores, averaging over the whole image might not lead to an accurate prediction. Indeed, it was observed that the eyes scan images with quick movements, gazing in the direction of the most relevant features (see e.g. [123]). In [115], a saliency map was derived from an information theoretical approach. Combined with the local SSIM score, this is called the *Information Weighted Structural Similarity* (IW-SSIM) map.

The SSIM index as defined in (1.16) must assume positive correlation between the two signals $\mathbf{x}$ and $\mathbf{y}$, otherwise the third term would be a power of a negative number, leading possibly to an imaginary-valued SSIM score. This issue seems to be ignored in the original contribution [114] and in more recent publication on the subject [24]. Since it is only the relative weights that matter, one can easily come around this issue by setting new weights as $\alpha' = \alpha/\gamma$, $\beta' = \beta/\gamma$ and $\gamma' = 1$. In the original paper [114], the weights α, β and γ are set empirically to 1, thus escaping the problem.

The actual form of the SSIM index also implies that the worst image is the one that is negatively correlated, i.e. $\mathbf{y} = 2\bar{\mathbf{x}} - \mathbf{x}$. Obviously, this image retains a lot of structure. An alternative to the SSIM index would be to take the absolute value of the correlation term and to multiply this term by a factor A accounting for the loss of perception for negative correlation:

$$c_{\pm}(\mathbf{x}, \mathbf{y}) \quad = \quad A(s_{x,y}) \left| \frac{s_{x,y} + \epsilon_3}{s_x s_y + \epsilon_3} \right|^{\gamma}, \tag{1.26}$$

with $A = 1$ if $s_{x,y} \geq 0$ and $A < 1$ otherwise. A psycho-visual experiment could be necessary to train for the values of the factor A.

The formula of the SSIM index does not directly take into account the visual distance of the observer. In practice, it is proposed to first downsample the image to a resolution between 256 and 511 pixels in each direction. A simple way to account for the visual distance would be to adjust the resolution of the image so that the viewing angle per pixel stays constant. That is, if the viewing distance is doubled, then the resolution should be halved.

The *Multi-Scale Structural Similarity* (MS-SSIM) [119] is a multi-scale extension of the SSIM index. It combines the SSIM index computed at several scales with different weighting parameters. In its most general form, it can be written as

$$MS - SSIM(\mathbf{x}, \mathbf{y}) = \prod_{m=1}^{M} (l(\mathbf{x}_m, \mathbf{y}_m))^{\alpha_m} (c(\mathbf{x}_m, \mathbf{y}_m))^{\beta_m} (s(\mathbf{x}_m, \mathbf{y}_m))^{\gamma_m}, \tag{1.27}$$

where $\mathbf{x}_m$ and $\mathbf{y}_m$ are the image $\mathbf{x}$ and $\mathbf{y}$, respectively, at resolution m. In [24], the weight parameters were trained from a psycho-visual database.

Another issue with the SSIM index is that it assumes perfectly registered (aligned) images. If the slightest misalignment occurs, then the correlation term can give erroneous results. As it is the case with luminance shift and contrast distortion, the rigid transformations and scaling, i.e. zooming, of an image are non-structural distortions, thus not affecting much the visual perception of quality. The problem was resolved in [95] by introducing a *Complex Wavelet Structural Similarity* (CW-SSIM). Transforming the images $\mathbf{x}$ and $\mathbf{y}$ via a complex-valued steerable pyramid decomposition we obtain the coefficients $c_{x,i}$ and $c_{y,i}$ for $1 \leq i \leq N$. The CW-SSIM is given by

$$\text{CW-SSIM}(\mathbf{x}, \mathbf{y}) = \frac{2|\sum_{i=1}^{N} c_{x,i} c_{y,i}^{*}| + C}{\sum_{i=1}^{N} |c_{x,i}|^2 + \sum_{i=1}^{N} |c_{y,i}|^2 + C}, \tag{1.28}$$

where C is a stability constant and $c_{y,i}^{*}$ is the complex conjugate of $c_{y,i}$. In this formula, the non-structural geometric changes is accounted in the phase shift of the complex wavelet coefficients.

Another possible weakness of the SSIM index is its handling of textures and noise. Indeed, it is empirically observed that the HVS cannot discriminate well between similar textures or noisy patches, even though the linear correlation between these patches might be weak. A statistical model of image perception could resolve the problem partly, but there is still an open question on how to combine the deterministic model of perception of the SSIM index with a statistical model that is fit for textures and noise.

The SSIM index was designed to measure distortion in greyscale still images. Some authors proposed an extension to colour images [74] (and recently [59]) and to videos [117]. The extra challenges are many. For colour images, one has to describe the perception of chrominance distortion and determine how to gather the errors from luminance and chrominance channels. For videos, one has to describe the perception of temporal distortions and to consider the possibility of temporal misalignment, the effect of motion for visual attention and to factor memory in the pooling of errors.

We described in this section different features of the SSIM index while outlining several strengths and weaknesses. Although it is clear that this IQA measure is not perfect, it is still state-of-the-art today. Particularly, the three extensions of the SSIM index – MS-SSIM, IW-SSIM and CW-SSIM – each resolve a serious weakness and offer an excellent performance in subjective database ratings. This will be exemplified in the next section.

1.4.3 What's Wrong with L^2?

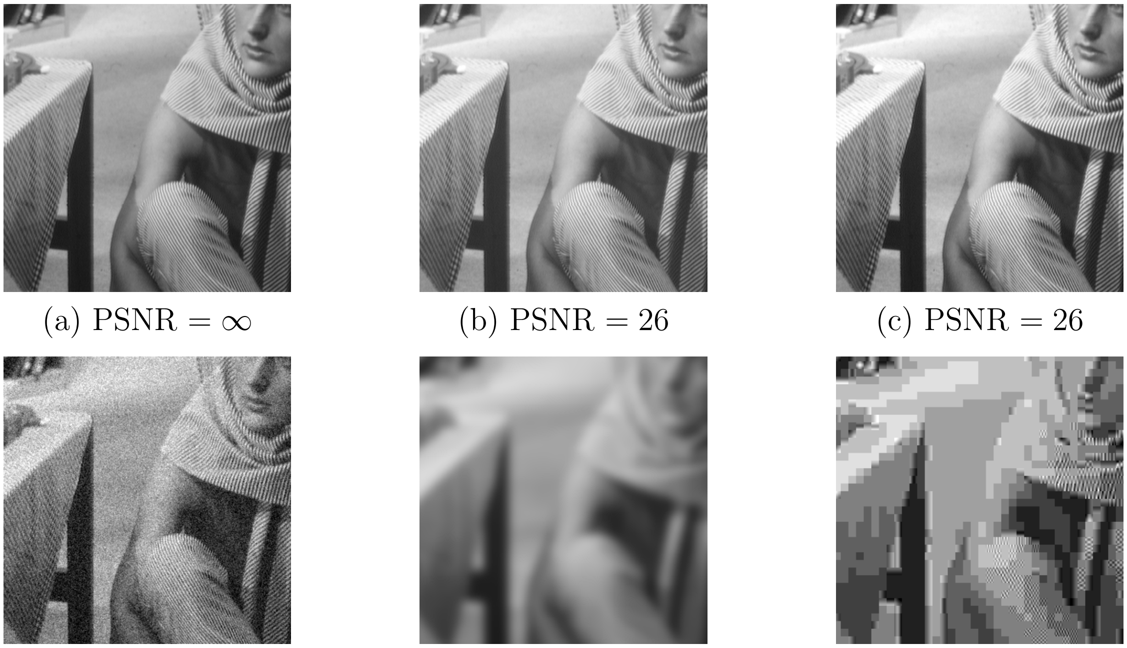
The legitimate question is, do we really need to develop special IQA measures for images? Would a simple L^p-norm not be sufficient? Indeed, L^2 has several mathematical properties that makes it a very attractive choice as a design criterion for optimal image processing algorithms (see [113]): it is simple and inexpensive to compute, it is a valid distance metric, it is invariant under orthogonal transforms, it is convex and differentiable and it is additive for independent sources of distortions. But Gauss himself, the inventor of the least square estimators, has warned against the arbitrariness of this choice of criterion [45]: "Si l'on objecte que cette convention est arbitraire et ne semble pas nécessaire, nous en convenons volontiers."[1] Nevertheless, it seems that this warning has been forgotten at the expense of mathematical convenience.

More recently, there has been a growing amount of voices expressing their concerns against the use of L^2-error in image processing applications. "What's wrong with mean squared error?" [48] is the most famous title, but an extended list of references point out in the same direction (see [8]). In particular, the Video Quality Expert Group [106] reached the conclusion that "the PSNR model fits significantly worse than the best models".

A striking example (see e.g. [114]) that demonstrates the point of the blatant weakness of L^2-error for IQA purposes is the following: Take several distorted images with all the same L^2-distance from the original but generated from different types of distortions. Their perceived quality will generally differ greatly, although L^2 wrongly predicts a similar quality (see Fig. 1.2). On the other hand, it can be seen that the SSIM index better predicts the human rating for these same images (see Fig. 1.3).

This is only an example, although very illustrative, and one could argue, as it is done in [37], that it is possible to devise another example with well chosen images to prove exactly the opposite point. Nevertheless, the SSIM index still performs better than the L^2-error on the average. To prove this point, it was proposed in [118] to push further this idea of comparing images with the same predicted quality according to one IQA model. A procedure was proposed to make two candidate IQA measures compete against each other by finding the image with minimal/maximal quality according to the first IQA measure over the set of all images with equal quality according to second IQA measure, and vice versa. By comparing subjectively these optimized images it was argued that one can choose between the two models, or at least invalidate one or both of the models. A comparison of the evolution of the correct discrimination rate between the MSE and the SSIM index

[1] If someone objects that this convention is arbitrary and does not seem necessary, we willingly agree. (free translation)



(a) PSNR = ∞ (b) PSNR = 26 (c) PSNR = 26

(d) PSNR = 26 (e) PSNR = 26 (f) PSNR = 26

Figure 1.2: Comparison between distorted images with the same L^2-distance. (a) Original image. (b) Global luminance shift of 20/255. (c) Global contrast stretch of a factor $a = 1.1422$. (d) AWGN of standard deviation 20/255. (e) Gaussian blur with a filter of standard deviation of 5 pixels. (f) JPEG compression with a quality factor $Q = 2$. All the images have the same error, although they look visually very different. This thus invalidates the L^2-error as a good IQA measure.

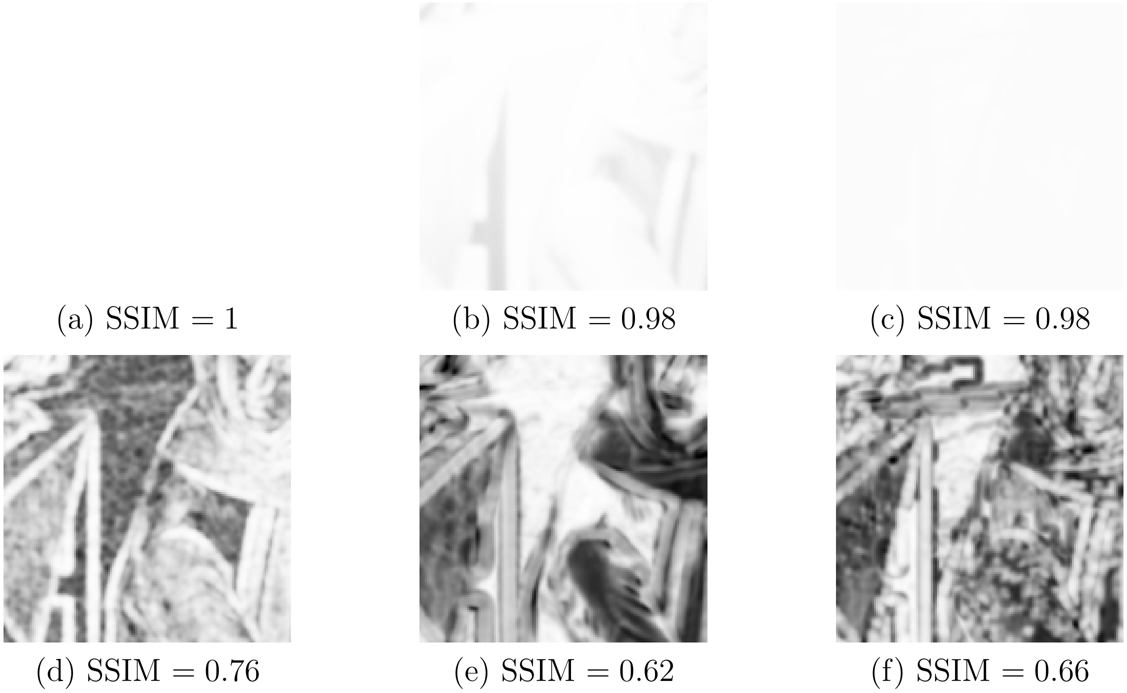
for different distortion levels then showed that the SSIM index outperforms the MSE for moderate to large distortions.

The more traditional way to compare IQA measures is through psycho-physical testing. We present in Fig. 1.4 the results for two subjective evaluation databases: LIVE (second release) and TID-2008. The procedure used and the subjective databases are described in details in Appendix A.

Note that the SSIM index was rescaled according to the formula

$$\text{rescaled SSIM}(\mathbf{x}, \mathbf{y}) = -10 \log_{10}(1 - \text{SSIM}(\mathbf{x}, \mathbf{y})) \tag{1.29}$$

in order to distribute the objective scores more uniformly. It can be seen as an analog to



(a) SSIM = 1

(b) SSIM = 0.98

(c) SSIM = 0.98

(d) SSIM = 0.76

(e) SSIM = 0.62

(f) SSIM = 0.66

Figure 1.3: Comparison of the SSIM maps between distorted images with the same L^2-distance. (a) Original image. (b) Global luminance shift. (c) Global contrast stretch. (d) Additive white Gaussian noise. (e) Gaussian blur. (f) JPEG compression. The SSIM maps represent the predicted error at different location of the distorted images. White means good predicted quality (SSIM close to 1) whereas black represents a low SSIM score (SSIM close to 0). Observe how the SSIM maps and scores follow more closely what a human observer would except.

the PSNR for the MSE. This rescaling will not change the rank correlation, but it allows a better fitting of the logistic curve. It will thus increase Pearson's correlation coefficient and decrease the ratio of outliers and the Root Mean Square Error (RMSE) between the predicted and observed values.

The quality of prediction between the PSNR or the SSIM index and the MOS can be compared with several measures (see Appendix A). These are summarized in Table 1.1 for the scores obtained from LIVE database and TID-2008 database. It is clear from these tests that the SSIM index outperforms the MSE as a predictor of subjective image quality. In fact, according to various subjective experiments, the SSIM index and its multi-scale extension are state-of-the-art IQA measures [61, 80, 101].

Table 1.1: Prediction measures of MOS in the LIVE and TID-2008 databases for the PSNR and the SSIM index.

	LIVE		TID-2008	
Prediction Measure	PSNR	SSIM (rescaled)	PSNR	SSIM (rescaled)
Spearman's ρ	0.8197	0.8999	0.5531	0.7749
Kendall's τ	0.6171	0.7183	0.4028	0.5768
Correlation Coefficients	0.7999	0.9053	0.5190	0.7737
Ratio of Outliers	N/A	N/A	0.7800	0.7306
RMSE	9.6656	6.8405	11.4707	8.5021

We ought to make some technical remarks on these last results. Images with no distortion or from the first database release were excluded from the test sequence using LIVE database. This explains lower prediction measures than the ones reported in [101]. Also, the choice of transformation from color to greyscale space could explain the discrepancy of the results between different IQA validation experiments. Remark as well that the SSIM measures computed in [61] omits the automatic downsampling of the images. Finally, the ratio of outliers could not be computed for the LIVE database since the standard deviations of the MOS were not available to the author.

A deeper look at the L^2-error reveals its intrinsic weaknesses for IQA. As explained in [113], four assumptions on image quality assessment are implicitly made when applying the L^2-error: 1) the image quality is independent of the spatial order of the pixels of the original image, 2) the image quality is independent of the sign of the sample errors, 3) the image quality is signal independent in the sense that it depends only on the error image no matter what is the original image and 4) all image samples are equally important for image quality. These assumptions were all proved to be erroneous.

1.5 Thesis Outline

As announced earlier, the goal of this thesis is to provide the mathematical tools needed to incorporate a perceptual criterion such as the SSIM index in image processing algorithms. This perceptual criterion is important for the improvement of image processing algorithms. Indeed, algorithms have to be optimized according to a perceptually relevant model in order to obtain better image quality. This will then eventually lead to concrete technological applications in domain such as digital imaging, remote sensing and medical imaging.

Besides numerous algorithms that are based on the L^p-norms, a few algorithms have been devised in literature attempting to achieve SSIM-optimal image processing. In particular, Channappayya et al. developed a SSIM-optimal affine transform [22] and a SSIM-optimal linear operator [21, 20] for image denoising. More recently, Rehman et al. [85] incorporated the SSIM index into the non-local means whereas Wang et al. [111] tried to do the same for fractal image compression.

The layout of this thesis is divided as follows. In Chapter 2, normalized metrics are introduced in full mathematical generality. A new normalized metric with stability constant is introduced and methods to combine several metrics into either a scalar-valued or vector-valued metric are developed. Some necessary or sufficient conditions are also given for a metric to be "normalizable". We then demonstrate how the SSIM index can be modified into a normalized metric and show that this approximation is valid in practice.

In the next chapter, we present more mathematical properties of the SSIM index. First, some symmetries and invariances of the SSIM index are discussed. We also prove some convexity and quasi-convexity properties of each of the components of the SSIM index as well as their combination. Then, methods to prove these properties for generalizations of the SSIM index are sketched. Finally, an analytical solution of the problem of projecting images onto convex sets according to the SSIM index is given.

In Chapter 4, we will describe the geodesic path between two images according to the SSIM index. The problem is to find series of images that are locally maximizing the SSIM index. A discussion on similar works by other authors will be provided as well as our own proofs.

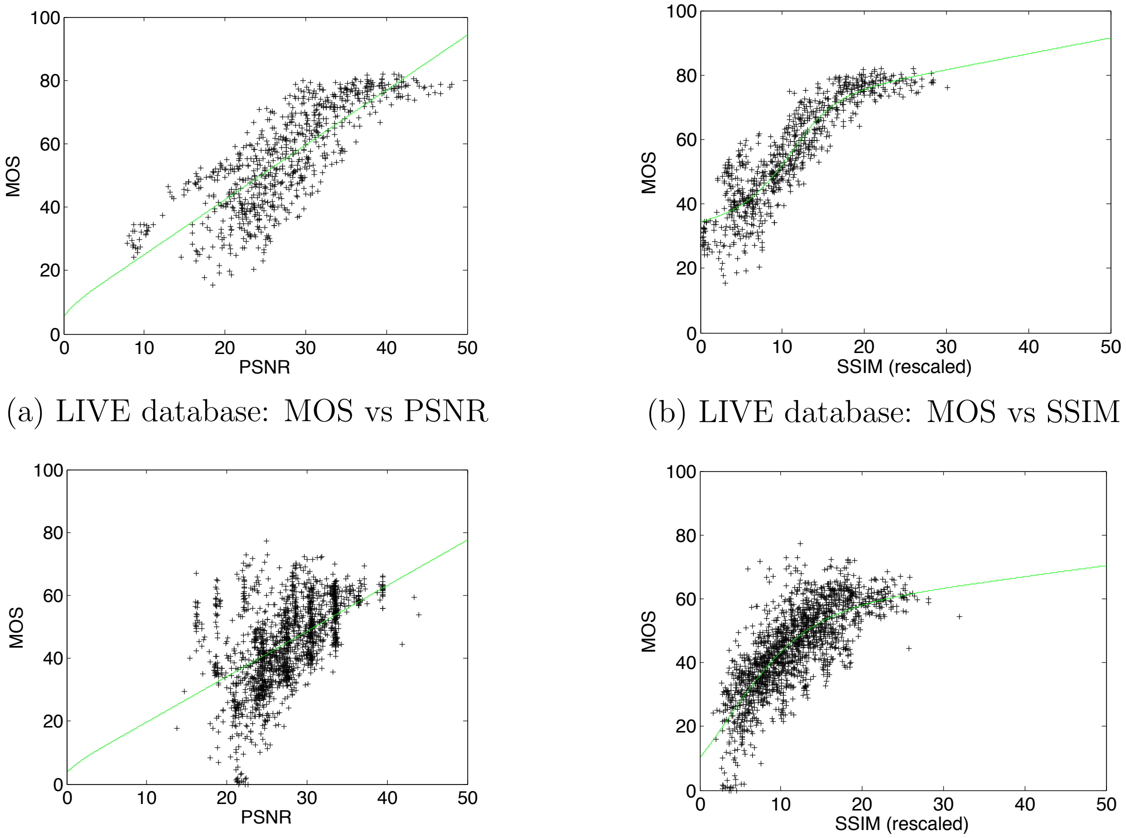
Immediately following this chapter, we analyze the similar problem of how to maximize the weighted sum of the SSIM index between several images. It turns out that the solution differs from the previous problem. The description of the solution as well as a comparison between the SSIM-mean and the geodesic path will also be provided in Chapter 5.

Then in Chapter 6, we will show how to optimize the SSIM index for orthogonal or redundant basis. That is, given a fixed basis, we will show how to choose the basis elements and their coefficients in order to maximize the SSIM index.

The self-similarity of images will be studied in Chapter 7. We will revisit the self-similarity of images in some experimental simulations. In particular, by computing the local block similarity of images according to the SSIM index, we will see that images are less self-similar than they might appear at first hand.

In Chapter 8, we will take advantage of the distortion model of images by using the residual image to assess and enhance image denoising algorithms. An estimator of the PSNR and the SSIM index between an original image and its attempted reconstruction that rely only on the noisy and the denoised images will be proposed. We will also use Fisher's significance tests to estimate locally the quality of the image denoising.

The novelty of this thesis lies in many fronts. First, the properties of the SSIM index are studied in depth to extend several of its known properties. Second, the optimization of the SSIM index is not limited to linear operators or to a particular algorithm. Instead, it will be studied in general for different kind of problems: geodesic path between images, decomposition into an optimal basis, best SSIM estimator, etc. In several instances the optimization takes into account the stability constants, thus allowing to treat every kind of signals. Finally, a general framework for SSIM-optimal image processing, and image denoising in particular, is laid down, leading the way to many applications.



(a) LIVE database: MOS vs PSNR

(b) LIVE database: MOS vs SSIM

(c) TID database: MOS vs PSNR

(d) TID database: MOS vs SSIM

Figure 1.4: Results of psychovisual testing of the Peak-to-Noise Signal Ratio (PSNR) and of the Structural Similarity (SSIM) index for LIVE database, second release, and for TID-2008 database. The green line represents the least square regression of a five parameters generalized logistic curve.

Chapter 2

On Normalized Metrics

In an experiment conducted by Weber in 1834, he observed that the change of perception of a stimulus depends not simply on the difference between stimulus intensities, but rather on their ratio (see e.g. [110]). Thus, he established that it is the relative error, not the absolute one, that needs to be considered in image quality assessment. In the metric space framework, this observation leads to the creation of normalized distances.

In the context of image processing, the metric space structure is useful for bounding the error between images or in order to prove the convergence of certain image processing algorithms. For example, for the image query retrieval problem in computer vision a search in the metric space is performed.

Moreover, Li [64] argued that the right model for images necessitates only a metric space structure and not a full Hilbert space structure. That is, instead of representing an image as a linear combination of some basis (i.e. discrete cosine or wavelet), he suggests representing it as the fixed point of some non-local self-similar and non-expansive operator (e.g. non-local means [16] or Block Matching 3-D (BM3D) [31]).

In this chapter, we will study normalized metrics and illustrates how a modification of the Structural Similarity (SSIM) index can enter into this class of metrics. After some preliminary discussions on Ptolemaic metric spaces, we will survey the few contributions on normalized metrics that were made in the literature. Our own contributions will also be added in the research of a complete characterization of these metrics. We will pursue this chapter with a brief introduction to vector-valued metrics and their scalarization to real-valued metrics. Finally, it will be demonstrated with a psychovisual experiment that one of the normalized metric predicts the human perception of image quality at least as well as the SSIM index.

2.1 Ptolemaic Metric Spaces

This section presents the characterization of inner-product spaces found in 1952 by I.J. Schoenberg [99] who linked them with Ptolemaic spaces. This characterization has important consequences for the results of the subsequent sections, since Ptolemy's inequality will be invoked in numerous occasions.

For the sake of completeness, let us begin with the definition of metric spaces.

Definition 2.1.1. *Let S be a non-empty set. A* metric *is a function*

$$d : S \times S \to \mathbf{R}_+,$$

with the following properties:

Positive definiteness :

$$d(x, y) = 0 \iff x = y, \quad \forall x, y \in S;$$

Symmetry :

$$d(x, y) = d(y, x), \quad \forall x, y \in S;$$

Triangle inequality :

$$d(x, z) \leq d(x, y) + d(y, z), \quad \forall x, y, z \in S.$$

We call a metric space (S, d) *the set S embedded with the metric d.*

An important identity for the points in $\mathbf{R}^N$ is Ptolemy's inequality. This is a generalization of the well-known geometric identity due to Ptolemy which states that if a quadrilateral is inscribed in a circle, then the sum of the products of the lengths of its opposed edges equals the product of the lengths of its diagonals. We can further generalize this property for other metric spaces.

Definition 2.1.2. *A metric space (S, d) is said to be* Ptolemaic *if*

$$d(x, y)d(z, w) + d(y, z)d(x, w) \geq d(y, w)d(x, z) \tag{2.1}$$

for all $x, y, z, w \in S$.

Theorem 2.1.3 (Ptolemy's inequality in real inner-product space)**.** *Let x, y, z, w be four points of X, a real inner-product space. Then*

$$\|x-y\|\|z-w\| + \|y-z\|\|x-w\| \geq \|y-w\|\|x-z\|. \tag{2.2}$$

This theorem was first found by Schoenberg in [99], where he mentions that one can generalize the proof of the Ptolemy's inequality in the plane given by Johnson [57]. We provide the details here.

Proof. Without loss of generality, we assume that $w = 0$. (We can always translate all the points by subtracting them w.) Let $B(0,r)$ be a ball of radius r. Let x', y', z' be the inversion of x, y, z, respectively, with respect to the ball $B(0,r)$. It means that $\langle x, x' \rangle = \|x\|\|x'\| = r^2$. Then by the triangle inequality, we have

$$\|x'-y'\| + \|y'-z'\| \quad \geq \quad \|x'-z'\|. \tag{2.3}$$

The Cauchy-Schwarz inequality $\langle x, x' \rangle \leq \|x\|\|x'\|$ is an equality if and only if $x' = ax$ for $a \in \mathbf{R}$. Hence,

$$x' = ax = \frac{r^2}{\|x\|^2}x.$$

Similar equalities also hold for y' and z'. After substituting x', y' and z' in the squared norms, i.e.

$$\|x'-y'\|^2 \quad = \quad \frac{r^4}{\|x\|^2\|y\|^2}\|x-y\|^2,$$

the inequality (2.3) simplifies to

$$\|z\|\|x-y\| + \|x\|\|y-z\| \quad \geq \quad \|y\|\|x-z\|.$$

Translating all the points by w, we get the desired result. □

We are now ready to state the theorem completing the characterization of inner-product spaces.

Theorem 2.1.4 (Characterization of inner-product spaces [99])**.** *Let X be a real semi-normed space (norms without the triangle inequality property). If Ptolemy's inequality holds for all points of this space, then it is an inner-product space.*

2.2 Characterization of Normalized Metrics

Since we are working with relative distances, we need a "zero" element in our metric space to which we will compare the magnitude of the other elements. It brings us to define the concept of norm, which will be useful in particular for the simplification of the notation, although not essential for the rest of the discussion. Moreover, for the manipulation of images, we want to be able to add them together (change of luminance) and to multiply them by a scalar (change of contrast). Thus the vector space structure is required. The combination of these requirements yields to the study of real normed vector space.

We call a real *normed space* the vector space X equipped with the norm $\|\cdot\|$. The *metric* induced by $\|\cdot\|$ is defined by

$$d(x, y) := \|x - y\|, \quad \forall x, y \in X.$$

Normalized metrics are a special kind of metrics that measure relative distance between two elements.

Definition 2.2.1. *Given a metric space (X, d) with $a \in X$ fixed and a function $M : \mathbf{R}_+ \times \mathbf{R}_+ \to \mathbf{R}_+$, a* normalized metric *(or M-*relative distance *[50]) is a metric of the form:*

$$\rho_M(x, y) := \frac{d(x, y)}{M(d(x, a), d(y, a))}. \tag{2.4}$$

For well-defineteness, we exclude from the domain X all the points for which $M = 0$ and $d > 0$. If $M = 0$ and $d = 0$, then we complete the metric as $\rho_M := 0$.

Note 2.2.2. *If (X, d) is a metric induced by a norm, then we can always take $a = 0$ and the normalized metric can be written more simply as*

$$\rho_M(x, y) := \frac{\|x - y\|}{M(\|x\|, \|y\|)}. \tag{2.5}$$

This is the notation we will adopt for the remainder of this work.

Contrary to the relative error $\frac{|x-y|}{|x|}$, which is not a metric, the function M is selected in a fashion that yields to a metric. The question is, for which function M and metric space (X, d) is $\rho_M(x, y)$ a metric? Our search for sufficient and necessary conditions will be the theme of the following sections.

2.2.1 Literature Review

The first well-known example of normalized metric is the chordal metric

$$d(x,y) = \frac{\|x-y\|_2}{\sqrt{1+\|x\|_2^2}\sqrt{1+\|y\|_2^2}}, \quad x,y \in \mathbf{R}^N.$$

In 1976, Schattschneider [98] discovered that

$$d(x,y) := \frac{|x-y|}{|x|+|y|}$$

is a metric on $\mathbf{R}$. She called this a *multiplicative metric*. Later, Yianilos [124] generalized this result on $\mathbf{R}^N$ for

$$d(x,y) = \frac{\|x-y\|_2}{\alpha + \beta(\|x\|_2 + \|y\|_2)} \tag{2.6}$$

with $\alpha \geq 0$ and $\beta > 0$.

Klamkin and Meir [58] found that the generalized multiplicative metric

$$d(x,y) = \frac{\|x-y\|_2}{(\|x\|_2^p + \|y\|_2^p)^{1/p}} \tag{2.7}$$

is a metric for all $p \geq 1$. They also showed that

$$d(x,y) = \frac{\|x-y\|_2}{\|x\|_2\|y\|_2} \tag{2.8}$$

is a metric. In fact, it follows directly from Ptolemy's inequality (2.2). Furthermore, they generalized the chordal metric by proving that

$$d(x,y) = \frac{\|x-y\|_2}{(\alpha + \beta\|x\|_2^p)^{1/p}(\alpha + \beta\|y\|_2^p)^{1/p}} \tag{2.9}$$

is a metric for $\alpha > 0$, $\beta \geq 0$ and $p \geq 1$.

Finally, in [50], Hästö found necessary and sufficient conditions for two classes of normalized metrics. The first class consists of metrics normalized by a generalization of the mean.

Definition 2.2.3. *The* generalized p-mean *of x and y in $\mathbf{R}_+$ is*

$$A_p(x,y) = \begin{cases} \left(\frac{x^p+y^p}{2}\right)^{1/p}, & \textit{if } p > 0, \\ \max(x,y) & \textit{if } p = \infty, \\ \min(x,y) & \textit{if } p = -\infty, \\ \sqrt{xy} & \textit{if } p = 0. \end{cases} \tag{2.10}$$

He also proved the following theorem.

Theorem 2.2.4 ([50])**.**

$$d(x,y) = \frac{\|x-y\|}{A_p^q(\|x\|,\|y\|)} \tag{2.11}$$

is a metric if and only if $0 < q \leq 1$ *and* $p \geq \max(1-q,(2-q)/3)$.

For $q = 1$, it extends the range of p for which the generalized multiplicative distance (2.7) is a metric to $p \geq 1/3$ while showing that it is a tight bound.

The second result that Hästö derived is for metrics normalized by a product of functions. Before stating this result, we need to first define moderately increasing functions.

Definition 2.2.5. *A function $f : \mathbf{R}_+ \to \mathbf{R}_+$ is* moderately increasing *if*

1. *$f(x)$ is increasing, i.e., $f(x_2) \geq f(x_1)$ for $x_2 \geq x_1$ and*
2. *$f(x)/x$ is decreasing for $x > 0$.*

Theorem 2.2.6 ([50])**.** *If $M(\|x\|,\|y\|) = f(\|x\|)f(\|y\|)$, then $\rho_M(x,y)$ is a metric if and only if f is moderately increasing and convex.*

In particular, this result established that the sufficient condition $p \geq 1$ for the generalized chordal metric of Klamkin and Meir is also necessary.

Hästö further proved these two results with the help of the subsequent lemma.

Lemma 2.2.7 (Hästö's Lemma [50])**.** *Let $(X, \|\cdot\|)$ be a Ptolemaic normed space and let $M(\|x\|,\|y\|)$ be a moderately increasing function with respect to $\|x\|$. Then*

$$\rho_M(x,y) \quad = \quad \frac{\|x-y\|}{M(\|x\|,\|y\|)} \tag{2.12}$$

is a metric on X if and only if it is a metric on $\mathbf{R}_+$.

We present and show the details of Hästö's proof.

Proof. If ρ_M is a metric on X, then it is in particular a metric on any one-dimensional subspace of X which is isometric to $\mathbf{R}_+$. It is clear that the symmetry and positive definiteness properties are preserved regardless of the underlying space. Thus, the real challenge is the triangle inequality. Without loss of generalities, let $x, y, z \in X$ with $|x| \leq |z|$. We want to show that

$$\begin{aligned} \rho_M(x,y) + \rho_M(y,z) &\geq \rho_M(x,z), \text{ i.e.} && (2.13)\\ \frac{M(|x|,|z|)}{M(|x|,|y|)}|x-y| + \frac{M(|x|,|z|)}{M(|z|,|y|)}|z-y| &\geq |x-z|. && (2.14) \end{aligned}$$

From the triangle inequality for the norm $\|\cdot\|$ and Ptolemy's inequality (2.1), we have two sufficient conditions:

$$\begin{aligned} \frac{M(|x|,|z|)}{M(|x|,|y|)}|x-y| + \frac{M(|x|,|z|)}{M(|z|,|y|)}|z-y| &\geq |x-y| + |y-z| \text{ and} && (2.15)\\ \frac{M(|x|,|z|)}{M(|x|,|y|)}|x-y| + \frac{M(|x|,|z|)}{M(|z|,|y|)}|z-y| &\geq \frac{|z|}{|y|}|x-y| + \frac{|x|}{|y|}|y-z|. && (2.16) \end{aligned}$$

Since M is increasing, the first condition holds if $|y| \leq |x|$. When $|y| \geq |z|$, the second condition is true since M is moderately increasing. In the case $|x| < |y| < |z|$, we want to combine both conditions. We have

$$\begin{aligned} \left(\frac{M(|x|,|z|)}{M(|x|,|y|)} - 1\right)|x-y| &\geq |y-z|\left(1 - \frac{M(|x|,|z|)}{M(|z|,|y|)}\right) \text{ and} && (2.17)\\ \left(\frac{|z|}{|y|} - \frac{M(|x|,|z|)}{M(|x|,|y|)}\right)|x-y| &\leq \left(\frac{M(|x|,|z|)}{M(|z|,|y|)} - \frac{|x|}{|y|}\right)|y-z|. && (2.18) \end{aligned}$$

From the moderately increasing property of M, all sides of both equalities are positive. Thus, the two sufficient conditions (2.15) and (2.16) can be rearranged into a single one:

$$\frac{\left(1 - \frac{M(|x|,|z|)}{M(|z|,|y|)}\right)}{\left(\frac{M(|x|,|z|)}{M(|x|,|y|)} - 1\right)} \leq \frac{|x-y|}{|y-z|} \leq \frac{\left(\frac{M(|x|,|z|)}{M(|z|,|y|)} - \frac{|x|}{|y|}\right)}{\left(\frac{|z|}{|y|} - \frac{M(|x|,|z|)}{M(|x|,|y|)}\right)}. \qquad (2.19)$$

It further simplifies to

$$\begin{aligned}\frac{|z|}{|y|} - \frac{M(|x|,|z|)}{M(|x|,|y|)} - \frac{|z|M(|x|,|z|)}{|y|M(|z|,|y|)} &\leq \frac{|x|}{|y|} - \frac{M(|x|,|z|)}{M(|z|,|y|)} - \frac{|x|M(|x|,|z|)}{|y|M(|x|,|y|)}, \\ \frac{|z|-|x|}{M(|x|,|z|)} &\leq \frac{|z|-|y|}{M(|z|,|y|)} + \frac{|y|-|x|}{M(|x|,|y|)}. \qquad (2.20)\end{aligned}$$

So if the triangle holds in $\mathbf{R}_+$, the sufficient condition (2.19) will be met and ρ_M will be a metric on X. □

2.2.2 Necessary and Sufficient Conditions

It is difficult to find necessary and sufficient conditions for a normalized distance $\rho_M(x, y)$ to be a metric. Here, we will give a partial answer by extending the results of Hästö [50] and our previous results [15]. The importance of these results will be manifested in particular when we will apply them to define a metric from each component of the SSIM index.

Theorem 2.2.8 (Sufficient Condition)**.** *Let $(X, \|\cdot\|)$ be a Ptolemaic normed space. Let $M(x, y) : \mathbf{R}_+^2 \to \mathbf{R}_+$ be a symmetric and moderately increasing function. Let*

$$f(y, g(y, x, z)) \;:=\; M(y, x)M(y, z) \qquad (2.21)$$

with f convex in the first argument and $g(x, y, z)$ pair-wise symmetric. Then ρ_M is a metric.

Proof. By Hästö's Lemma 2.2.7, we only need to check that ρ_M is a metric on $\mathbf{R}_+$. Without loss of generality, consider $x \leq z$. In fact, from the sufficient condition (2.15) of the proof of the lemma, ρ_M is a metric if $0 \leq y \leq x$, since M is increasing. From the second sufficient condition (2.16), ρ_M is a metric if $y \geq z$, since M is moderately increasing. It remains to treat the case $0 \leq x < y < z$. By convexity of f and symmetry of g, we have for all $\lambda \in (0, 1)$, with $y = \lambda x + (1 - \lambda)z$,

$$f(y, g(y, x, z)) \;\leq\; \lambda f(x, g(x, y, z)) + (1 - \lambda)f(z, g(z, x, y)). \qquad (2.22)$$

It is equivalent to

$$M(y, x)M(y, z) \;\leq\; \lambda M(x, y)M(x, z) + (1 - \lambda)M(z, x)M(z, y). \qquad (2.23)$$

Rearranging the terms and using the symmetry of M lead to

$$\frac{1}{M(x,z)} \leq \frac{\lambda}{M(y,z)} + \frac{1-\lambda}{M(x,y)}. \tag{2.24}$$

Finally, by taking

$$\lambda = \frac{z-y}{z-x},$$

we obtain

$$\frac{z-x}{M(x,z)} \leq \frac{z-y}{M(y,z)} + \frac{y-x}{M(x,y)}. \tag{2.25}$$

□

Note that in particular, it implies that ρ_M is a metric for

$$M(|x|,|y|) = f(|x|)f(|y|),$$

thus generalizing the sufficient condition of Hästö. We can also apply the sufficient condition just found to powers of the generalized p-mean between x, y and a constant c.

Definition 2.2.9. *The* generalized p-mean *between x, y and a constant c in $\mathbf{R}_+$ is*

$$A_{p,c}(x,y) = \begin{cases} \left(\frac{x^p+y^p+c^p}{3}\right)^{1/p}, & \text{if } p > 0, \\ \max(x,y,c), & \text{if } p = \infty, \\ \min(x,y,c), & \text{if } p = -\infty, \\ (xyc)^{1/3}, & \text{if } p = 0. \end{cases} \tag{2.26}$$

We will generalize some of these results for the family of normalized metrics taking the form

$$\rho_{A^q_{p,c}}(x,y) := \frac{\|x-y\|}{A^q_{p,c}(\|x\|,\|y\|)}. \tag{2.27}$$

Corollary 2.2.10. *$\rho_{A^q_{p,c}}$ is a metric for $c \geq 0$, $1/2 \leq q \leq 1$ and $p \geq 1$.*

Proof. It is easy to verify that $A^q_{p,c}$ is positive, symmetric and moderately increasing for

$0 < q \leq 1$. Posing

$$\begin{aligned} g(x,y,z) &:= y^p z^p + y^p x^p + x^p z^p + c^p(x^p + y^p + z^p) \text{ and} \quad (2.28) \\ f(y, g(x,y,z)) &:= (y^{2p} + y^p c^p + c^{2p} + g(x,y,z))^{q/p}, \quad (2.29) \end{aligned}$$

we can see that

$$3^{2q/p} A^q_{p,c}(y,x) A^q_{p,c}(y,z) \;=\; f(y, g(x,y,z)).$$

It is clear that g is pair-wise symmetric. It remains, however to find the range of values of p and q for which f is convex in the first component.

Taking the second derivative of $f(y,a)$ with respect to y leads to

$$\begin{aligned} \frac{\partial^2 f}{\partial y^2} &= q y^{p-2}(y^{2p} + y^p + a)^{\frac{q}{p}-2}\big((4q-2)y^{3p} + (4q+p-3)c^p y^{2p} \quad (2.30) \\ &\quad + (q+4p-3)c^{2p}y^p + (p-1)c^{3p} + a((4p-2)y^p + (p-1)c^p)\big). \end{aligned}$$

For $\frac{\partial^2 f}{\partial y^2}$ to be positive, it is sufficient that $p \geq 1$ and $q \geq 1/2$. □

From (2.30), we can extend the range of p to $p \geq 1/2$, if $c = 0$. Remark that in this case the sufficient condition is not necessary based on Hästö's Theorem 2.2.4:

$$p \geq \max(1-q, (2-q)/3) \geq 1/3 \text{ for } 0 < q \leq 1.$$

It is worth noting that the normalized metric is continuous at $(0,0)$ if and only if $c > 0$.

We will now present two different necessary conditions for normalized metrics.

Theorem 2.2.11 (Necessary Condition 1)**.** *Let d be a Ptolemaic metric. If $\rho_M = \frac{d(x,y)}{M(\|x\|,\|y\|)}$ is a metric, then M is positive, symmetric and moderately increasing.*

Proof. Suppose that ρ_M is a metric on X. In particular, it is also a metric on a one dimensional sub-space isometric to $\mathbf{R}$.

- If M is not symmetric, then $M(|x|,|y|) > M(|y|,|x|)$ for some $|x| \neq |y|$. This implies that

$$\begin{aligned} \frac{d(x,x)}{M(|x|,|x|)} + \frac{d(x,y)}{M(|x|,|y|)} &= \frac{d(y,x)}{M(|x|,|y|)}, \qquad (2.31) \\ &< \frac{d(y,x)}{M(|y|,|x|)}. \end{aligned}$$

Hence, ρ_M is not a metric; a contradiction.

- If M is not positive, then $M(|x|,|y|) < 0$ for some x and y. Since ρ_M is a metric, M has to be symmetric. This implies that

  $$\begin{aligned} \frac{d(x,y)}{M(|x|,|y|)} + \frac{d(y,x)}{M(|y|,|x|)} &< 0, \qquad (2.32) \\ &= \frac{d(x,x)}{M(|x|,|x|)}. \end{aligned}$$

  Hence, ρ_M is again not a metric; another contradiction.

- We now show that if $M(|y|,|x|) > M(|z|,|x|)$ for some $|x| < |y| < |z|$, then ρ_M cannot be a metric. Note that we already established that M is symmetric. Consider the following two cases.

  1. If $M(|y|,|z|) \geq M(|x|,|z|)$, then for $0 < x < y < z$,

     $$\begin{aligned} d(x,y)\frac{M(|x|,|z|)}{M(|x|,|y|)} + d(y,z)\frac{M(|x|,|z|)}{M(|y|,|z|)} &< d(x,y) + d(y,z), \qquad (2.33) \\ &= d(x,z). \end{aligned}$$

  2. If $M(|y|,|z|) < M(|x|,|z|)$, then $M(|y|,|z|) < M(|x|,|y|)$ as well and for $y < 0 < x < z$, i.e. for $x = |x|$, $y = -|y|$ and $z = |z|$,

     $$\begin{aligned} d(x,y)\frac{M(|y|,|z|)}{M(|x|,|y|)} + d(x,z)\frac{M(|y|,|z|)}{M(|x|,|z|)} &< d(x,y) + d(x,z), \qquad (2.34) \\ &= d(y,z). \end{aligned}$$

  Thus, ρ_M cannot be a metric, which contradicts our previous assumption. Therefore, M has to be increasing.

- We perform a similar argument in the case that

  $$\frac{M(|x|,|z|)}{|x|} < \frac{M(|y|,|z|)}{|y|} \text{ for some } |x| < |y| < |z|.$$

  If $M(|x|,|z|)/|z| \leq M(|x|,|y|)/|y|$, then for $0 < x < y < z$,

  $$\begin{aligned} d(x,y)\frac{M(|x|,|z|)}{M(|x|,|y|)} + d(y,z)\frac{M(|x|,|z|)}{M(|y|,|z|)} &< \frac{|z|}{|y|}d(x,y) + \frac{|x|}{|y|}d(y,z), \\ &= d(x,z). \end{aligned} \tag{2.35}$$

  And, if $M(|x|,|z|)/|z| > M(|x|,|y|)/|y|$, then

  $$\frac{|z|}{|y|}M(|x|,|y|) < M(|x|,|z|) < \frac{|x|}{|y|}M(|y|,|z|)$$

  and for $y < 0 < x < z$, i.e. for $x = |x|$, $y = -|y|$ and $z = |z|$,

  $$\begin{aligned} d(x,z)\frac{M(|x|,|y|)}{M(|x|,|z|)} + d(y,z)\frac{M(|x|,|y|)}{M(|y|,|z|)} &< \frac{|y|}{|z|}d(x,z) + \frac{|x|}{|z|}d(y,z), \\ &= d(x,y). \end{aligned} \tag{2.36}$$

  Both cases imply that ρ_M is not a metric; more contradictions.

So if ρ_M is a metric, then M must be positive, symmetric and moderately increasing everywhere. □

Theorem 2.2.12 (Necessary Condition 2)**.** *Let M be a positive, symmetric and moderately increasing function. If $\rho_M(x,y) = \frac{d(x,y)}{M(\|x\|,\|y\|)}$ is a metric, then d is a metric for $\|y\| \geq \max(\|x\|,\|z\|)$ and d is Ptolemaic for $\|y\| \leq \min(\|x\|,\|z\|)$.*

Proof. Suppose that d is not a metric for $\|y\| \geq \max(\|x\|,\|z\|)$. Then for some x, y, z such that $\|y\| \geq \max(\|x\|,\|z\|)$,

$$d(x,y) + d(y,z) < d(x,z). \tag{2.37}$$

Now, since we assumed that M is increasing and symmetric, we have

$$\frac{M(\|x\|,\|z\|)}{M(\|x\|,\|y\|)}d(x,y) + \frac{M(\|x\|,\|z\|)}{M(\|y\|,\|z\|)}d(y,z) \leq d(x,y) + d(y,z). \tag{2.38}$$

Combining the last two equations, we see that ρ_M is not a metric.

Similarly, suppose that d is not Ptolemaic for $\|y\| \leq \min(\|x\|, \|z\|)$. Then for some x, y, z such that $\|y\| \leq \min(\|x\|, \|z\|)$,

$$\frac{\|z\|}{\|y\|} d(x, y) + \frac{\|x\|}{\|y\|} d(y, z) \quad < \quad d(x, z). \tag{2.39}$$

Again, since we assumed that M is moderately increasing and symmetric, we have

$$\frac{M(\|x\|, \|z\|)}{M(\|x\|, \|y\|)} d(x, y) + \frac{M(\|x\|, \|z\|)}{M(\|y\|, \|z\|)} d(y, z) \leq \frac{\|z\|}{\|y\|} d(x, y) + \frac{\|x\|}{\|y\|} d(y, z). \tag{2.40}$$

Combining once more the last two equations, we see that ρ_M is still not a metric. □

Corollary 2.2.13. *Let M be a positive, symmetric and moderately increasing function. Then $\rho_M(x, y) := \|x - y\|_p / M(\|x\|_p, \|y\|_p)$ is not a metric for $p \neq 2$.*

Proof. From Theorem 2.1.4, $\| \cdot \|_p$ is not Ptolemaic for $p \neq 2$. From Theorem 2.2.12, it implies automatically that ρ_M is not a metric. □

This looks like bad news. Of the L^p-norms, only L^2 is “normalizable” with positive, symmetric and moderately increasing functions. Remark however that it is possible to construct normalized metrics from non-Ptolemaic metrics (or even from non-metrics) but it would involve other types of functions M. For example, if we can show the following two Ptolemy-type properties for d

$$\frac{\|x\|^\alpha}{\|y\|^\alpha} d(x, y) + \frac{\|z\|^\alpha}{\|y\|^\alpha} d(z, y) \geq d(x, z) \text{ for } \|y\| \leq \min(\|x\|, \|z\|) \tag{2.41}$$

and

$$\frac{\|x\|^\beta}{\|y\|^\beta} d(x, y) + \frac{\|z\|^\beta}{\|y\|^\beta} d(z, y) \geq d(x, z) \text{ for } \|y\| \geq \max(\|x\|, \|z\|), \tag{2.42}$$

then we will need the following property of M for ρ_M to possibly be a metric:

$$\frac{\|x\|^\alpha}{\|y\|^\alpha} \leq \frac{M(\|x\|, \|z\|)}{M(\|y\|, \|z\|)} \leq \frac{\|x\|^\beta}{\|y\|^\beta}. \tag{2.43}$$

Note that this kind of condition can be generalized to any given pair of functions of $\|x\|/\|y\|$.

2.3 Vector-valued Metrics

In image quality assessment, we take into account several features of images. For example, in the SSIM index, the luminance, contrast and structure of images are all locally compared and usually combined into a single score. This is done via multiplication of the local distortion scores and the average of all the scores over the whole image. One may wonder why it is necessary to collapse all the scores into a single one. Indeed, it has already been demonstrated that the SSIM map (see [114]) provides valuable information on the location of distortions in an image. This leads us now to the idea of vector-valued metrics.

One attractive side of vector-valued metrics is that it accounts for the fact that certain objects are not comparable. We will see that contrary to metrics which are totally ordered, vector-valued are not. This means that for a certain pair of objects, it is not possible to tell whether they are closer or farther from each other than another pair. Concretely, it is interpreted as the impossibility to compare the quality of certain images.

2.3.1 On Cone Metric Spaces

This section follows the definitions given in a recent paper by Huang and Zhang [53]. It will be useful later when we study the possibility of a vector-valued SSIM index.

In the definition of a metric, d maps two objects in $S \times S$ to a real number that represents the distance between them. One can wonder why the specific use of $\mathbf{R}$ as a measure of distance was chosen. One reason suggested is that real numbers are totally ordered.

Definition 2.3.1. *A set S is* totally ordered *if for any element $a, b \in S$, either (i) $a \preceq b$ or (ii) $a \succeq b$, where "$\succeq$" is an order relationship.*

For real numbers, the order relationship is the usual inequality.

For applications in image processing and quality assessment, it may be desirable to have a metric mapping to a vector space E. In this case, we lose the total ordering property, but we can still define a partial ordering from cones.

Definition 2.3.2. *Let X be a vector space on $\mathbf{R}$. A* salient cone *$P \subset X$ obeys the following set of conditions:*

1. *P is closed, nonempty and $P \neq \{0\}$,*

2. *If $x, y \in P$, then $ax + by \in P$ for all $a, b \in \mathbf{R}_+$ (i.e. $a, b \geq 0$) and*

3. *$x \in P$ and $-x \in P$ implies that $x = 0$.*

For $x, y \in X$, we write $x \succeq y$ if and only if $x - y \in P$. This is also written symbolically as $y \in x + P$.

Example 2.3.3. *The positive orthant $\mathbf{R}^N_+ \subset \mathbf{R}^N$ is an example of a cone. The partial ordering associated with this cone is*

$$x \succeq y \iff x_k \geq y_k, \quad 1 \leq k \leq N, \tag{2.44}$$

where $x = (x_1, x_2, \dots, x_N)$.

Definition 2.3.4. *A* cone metric space *$(S, \mathbf{d})$ is a set S and an associated vector-valued function*

$$\mathbf{d}(x, y) : S \times S \quad \rightarrow \quad X, \tag{2.45}$$

with X a vector space, satisfying the following conditions for all $x, y, z \in S$:

1. *$\mathbf{d}(x, y) = \mathbf{0} \iff x = y$,*

2. *$\mathbf{d}(x, y) = \mathbf{d}(y, x)$ and*

3. *$\mathbf{d}(x, y) + \mathbf{d}(y, z) \succeq \mathbf{d}(x, z)$.*

Note that contrary to the definition of Huang and Zhang, we did not assume that X is a Banach space.

2.3.2 Scalarization of Vector-valued Metrics

Given several metrics $\{d_m\}_{m=1}^M$, one can consider them as components of a vector $\mathbf{d}(x, y)$ in a cone metric space. This will lead to a sometimes difficult multi-objective optimization. Here, we will study how to collapse them into a single metric. Simpler optimization schemes of image processing algorithms will be the principal benefit of this procedure.

First of all, it is useful to recall the following classical theorem.

Lemma 2.3.5 (Orthogonal Decomposition Theorem)**.** *Let* $\{X_m\}_{m=1}^M$ *be an orthogonal decomposition of* $\mathbf{R}^N$*, i.e.*

$$\bigcup_{m=1}^{M} X_m = \mathbf{R}^N$$

and

$$X_i \perp X_j \text{ for all } i \neq j.$$

Then each element x *of* $\mathbf{R}^N$ *can be written uniquely as*

$$x = \sum_{m=1}^{M} x_m, \tag{2.46}$$

where $x_m = P_m(x)$ *are the projections of* x *on* X_m*.*

Definition 2.3.6. *A norm in* $\mathbf{R}^M$ *is said to satisfy the* increasing property *if for any* $a \in \mathbf{R}_+^M$ *and* $b \in \mathbf{R}_+^M$*,*

$$\|a + b\| \geq \|a\|. \tag{2.47}$$

The increasing property in (2.47) holds for suitably weighted l^p norms, e.g.

$$\|x\|_{w,p} = \left[\sum_{m=1}^{M} w_m |x_m|^p\right]^{1/p} \tag{2.48}$$

where $w_{k,p} > 0$ and $1 \leq p < \infty$ for $1 \leq m \leq M$. It also holds for the L^∞-norm, e.g.

$$\|x\|_\infty = \max_{1\leq m\leq M} \{|x_m|\}. \tag{2.49}$$

We can show however that (2.47) does not hold for **all** norms.

The following theorem shows that, in general, if $\|\cdot\|$ possesses the increasing property then the norm of a vector of metrics is also a metric.

Theorem 2.3.7. *Let* $\{X_m\}_{m=1}^M$ *be a standard orthogonal decomposition of* $\mathbf{R}^N$*. Let* $P_m : \mathbf{R}^N \to X_m$ *for* $1 \leq m \leq M$ *be the associated projection of* $x \in \mathbf{R}^N$ *to* X_m *with* $x_m := P_m(x)$*. Let* $\{d_m\}_{m=1}^M$ *be* M *metrics on, respectively,* $\{X_m\}_{m=1}^M$*. Define*

$$\begin{aligned} \mathbf{d}(x,y) : \mathbf{R}^N \times \mathbf{R}^N &\to \mathbf{R}_+^M \\ (x,y) &\mapsto (d_1(x_1,y_1), \ldots, d_M(x_M,y_M)). \end{aligned} \tag{2.50}$$

Let $\|\cdot\|$ *be a norm in* $\mathbf{R}^M$ *that satisfies the increasing property (2.47) in* $\mathbf{R}^M_+$*. Then*

$$d(x,y) := \|\mathbf{d}\| \tag{2.51}$$

is a metric in $\mathbf{R}^N$*.*

Proof. It is clear that $x = y$ implies that $\mathbf{d} = \mathbf{0}$. Now, if $\mathbf{d}(x,y) = \mathbf{0}$, then $d_m(x_m, y_m) = 0$ for all $1 \leq m \leq M$, so $x_m = y_m$ for all $1 \leq m \leq M$ since d_m is a metric. From the Orthogonal Decomposition Theorem (Lemma 2.3.5), we can imply that $x = y$. So by the property of norms, we have

$$\|\mathbf{d}\| = \mathbf{0} \iff x = y.$$

The symmetry property is straightforward. It remains to prove that $\|\mathbf{d}\|$ satisfies the triangle inequality. Since $d_1, d_2, \ldots, d_M$ are metrics, every component of $\mathbf{d}$ satisfies the triangle inequality. From our assumption that the $\|\cdot\|$ norm satisfies the increasing property, we conclude:

$$\begin{aligned} \|\mathbf{d}(x,z)\| &\leq \|\mathbf{d}(x,y) + \mathbf{d}(y,z)\| \\ &\leq \|\mathbf{d}(x,y)\| + \|\mathbf{d}(y,z)\|. \end{aligned} \tag{2.52}$$

□

2.4 The SSIM Index as a Normalized Metric

It is rather straightforward to prove the following properties of the SSIM index (1.16) or (1.21).

Proposition 2.4.1. *The SSIM index is bounded between* -1 *and* 1*.*

Proof. Since $\mathbf{x}, \mathbf{y} \in \mathbf{R}^N_+$, both $\bar{x} \geq 0$ and $\bar{y} \geq 0$. It implies that $l(x,y) \geq 0$. Since $s_x \geq 0$ and $s_y \geq 0$, then $c(\mathbf{x}, \mathbf{y}) \geq 0$ as well. Since

$$\frac{2xy + c}{x^2 + y^2 + \epsilon} \leq 1 \iff x^2 - 2xy + y^2 \geq 0 \iff (x-y)^2 \geq 0,$$

we have $l(\mathbf{x}, \mathbf{y}) \leq 1$ and $c(\mathbf{x}, \mathbf{y}) \leq 1$. By the Cauchy-Schwarz inequality,

$$|s_{x,y}| \leq s_x s_y.$$

It implies automatically that $-1 \leq s(\mathbf{x},\mathbf{y}) \leq 1$. Combining the information on $l(\mathbf{x},\mathbf{y})$, $c(\mathbf{x},\mathbf{y})$ and $s(\mathbf{x},\mathbf{y})$, we arrive at the desired conclusion. □

Proposition 2.4.2. *SSIM*$(\mathbf{x},\mathbf{y}) = 1 \iff \mathbf{x} = \mathbf{y}$.

Proof. If $\mathbf{x} = \mathbf{y}$, then $\bar{x} = \bar{y}$ and $s_{x,y} = s_x^2 = s_y^2$. Therefore, it is clear that $\text{SSIM}(\mathbf{x},\mathbf{y}) = 1$. If $\text{SSIM}(\mathbf{x},\mathbf{y}) = 1$, then the only possibility is $l(\mathbf{x},\mathbf{y}) = 1$, $c(\mathbf{x},\mathbf{y}) = 1$ and $s(\mathbf{x},\mathbf{y}) = 1$ since $0 \leq l(\mathbf{x},\mathbf{y}) \leq 1$, $0 \leq c(\mathbf{x},\mathbf{y}) \leq 1$ and $-1 \leq s(\mathbf{x},\mathbf{y}) \leq 1$. By the Cauchy-Schwarz inequality,

$$s_{x,y} = s_x s_y \iff \mathbf{y} = a\mathbf{x} + b \quad \text{with } a > 0.$$

It means that $\bar{y} = a\bar{x} + b$ and $s_y = as_x$. But since

$$\frac{2xy + c}{x^2 + y^2 + c} = 1 \iff x^2 - 2xy + y^2 = 0 \iff (x-y)^2 = 0,$$

we find that $\bar{x} = \bar{y}$ and $s_x = s_y$. This implies that $a = 1$ and $b = 0$, i.e. that $\mathbf{y} = \mathbf{x}$. □

Proposition 2.4.3. *The SSIM index is symmetric.*

Proof. It is clear that $l(\mathbf{x},\mathbf{y})$, $c(\mathbf{x},\mathbf{y})$ and $s(\mathbf{x},\mathbf{y})$ are symmetric, so automatically it implies that $\text{SSIM}(\mathbf{x},\mathbf{y}) = \text{SSIM}(\mathbf{y},\mathbf{x})$. □

Obviously, the SSIM index is not a metric. To transform the SSIM index into a metric, it is tempting to try a function like $T(\mathbf{x},\mathbf{y}) = 1 - \text{SSIM}(\mathbf{x},\mathbf{y})$ or $T(\mathbf{x},\mathbf{y}) = \sqrt{1 - \text{SSIM}(\mathbf{x},\mathbf{y})}$. With this, the only thing that remains is to prove the triangle inequality. Still, it is not clear if this will lead to a metric, but we can show that the mean component S_1 and the zero-mean component S_2 of the SSIM index can be transformed into a normalized metric. Moreover, both components can be combined into a single metric either by taking a weighted L^p-norm or by considering them as components of a generalized cone metric.

Corollary 2.4.4. *Let S_1 and S_2 be a decomposition of the SSIM index between $\mathbf{x}$ and $\mathbf{y}$ on a block/image of size n:*

$$\textit{SSIM}(\mathbf{x},\mathbf{y}) \quad = \quad S_1(\mathbf{x},\mathbf{y})S_2(\mathbf{x},\mathbf{y}) \tag{2.53}$$

where

$$S_1(\mathbf{x},\mathbf{y}) = \frac{2\bar{x}\bar{y} + \epsilon_1}{\bar{x}^2 + \bar{y}^2 + \epsilon_1} \quad \textit{and} \quad S_2(\mathbf{x},\mathbf{y}) = \frac{2s_{x,y} + \epsilon_2}{s_x^2 + s_y^2 + \epsilon_2}. \tag{2.54}$$

Then

$$d_1(\mathbf{x},\mathbf{y}) := \sqrt{1 - S_1(\mathbf{x},\mathbf{y})} \quad \textit{and} \quad d_2(\mathbf{x},\mathbf{y}) := \sqrt{1 - S_2(\mathbf{x},\mathbf{y})} \tag{2.55}$$

are metrics.

Proof. Develop

$$\begin{aligned} d_1(\mathbf{x},\mathbf{y}) &:= \sqrt{1 - S_1(\mathbf{x},\mathbf{y})} \\ &= \sqrt{\frac{\bar{x}^2 + \bar{y}^2 - 2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2 + c_1}} \\ &= \frac{|\bar{x} - \bar{y}|}{\sqrt{\bar{x}^2 + \bar{y}^2 + \epsilon_1}}. \end{aligned} \tag{2.56}$$

Also,

$$\begin{aligned} d_2(\mathbf{x},\mathbf{y}) &:= \sqrt{1 - S_2(\mathbf{x},\mathbf{y})} \\ &= \sqrt{\frac{s_x^2 + {s_y}^2 - 2s_{x,y}}{s_x^2 + s_y^2 + \epsilon_2}} \\ &= \frac{\|(\mathbf{x} - \bar{\mathbf{x}}) - (\mathbf{y} - \bar{\mathbf{y}})\|_2}{\sqrt{\|\mathbf{x} - \bar{\mathbf{x}}\|_2^2 + \|\mathbf{y} - \bar{\mathbf{y}}\|_2^2 + (n-1)\epsilon_2}}. \end{aligned} \tag{2.57}$$

We recognize $d_1(\bar{x},\bar{y})/\sqrt{3}$ with $c = \sqrt{\epsilon_1}$ and $d_2(\mathbf{x} - \bar{\mathbf{x}}, \mathbf{y} - \bar{\mathbf{y}}))/\sqrt{3}$ with $c = \sqrt{(n-1)\epsilon_2}$ as metrics in $\mathbf{R}$ and $\mathbf{R}^{n-1}$, respectively, of the form of Corollary 2.2.10 with $p = 2$, $q = 1$. Since projections and scalar multiplications send elements of $\mathbf{R}^n$ into $\mathbf{R}^n$, it is clear that d_1 and d_2 are also metrics in $\mathbf{R}^n$. □

In practice, we only have proved that the components of the SSIM index can be transformed into metrics but we have not yet shown anything about the full SSIM index itself. So in order to obtain a metric, we need either to consider a vector-valued form or to combine the components of the SSIM index into one.

From the results of Section 2.3, we have that

$$\big(\mathbf{R}^n, \mathbf{d} = (d_1, d_2)\big) \tag{2.58}$$

is a cone metric space, where d_1 and d_2 are the components of the SSIM index given in (2.55). It has the partial order

$$\mathbf{d}(\mathbf{x},\mathbf{y}) \succeq \mathbf{d}(\mathbf{z},\mathbf{v}) \iff d_i(\mathbf{x},\mathbf{y}) \geq d_i(\mathbf{z},\mathbf{v}), \quad \text{for } i = \{1,2\}, \tag{2.59}$$

where $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{v} \in \mathbf{R}^n$.

A scalarization of this cone metric can be done via

$$\begin{aligned} \|\mathbf{d}\|_{(\mathbf{w},p)} &= (w_1 d_1^p + w_2 d_2^p)^{1/p} \\ &= (w_1(1-S_1)^{p/2} + w_2(1-S_2)^{p/2})^{1/p}. \end{aligned} \tag{2.60}$$

From (2.51), this is a metric for all $p > 0$ and all $(w_1, w_2) \in \mathbf{R}_+^2$. In particular, for $p = 2$ and $w = (1, 1)$,

$$\begin{aligned} D_2 &:= \|(d_1, d_2)\|_{((1,1),2)} = \sqrt{d_1^2 + d_2^2} \\ &= \sqrt{2 - S_1 - S_2} \end{aligned} \tag{2.61}$$

is a metric.

Writing $S_1 = 1 - d_1^2$ and $S_2 = 1 - d_2^2$, we compare

$$D_2 = \sqrt{d_1^2 + d_2^2} \tag{2.62}$$

with

$$\begin{aligned} \sqrt{1 - \text{SSIM}} &= \sqrt{1 - (1 - d_1^2)(1 - d_2^2)} \\ &= \sqrt{d_1^2 + d_2^2 - d_1^2 d_2^2}. \end{aligned} \tag{2.63}$$

That is, D_2 can be seen as a lower degree approximation and an upper bound of $\sqrt{1 - \text{SSIM}}$. When either $d_1 = 0$ or $d_2 = 0$, then the two expressions are equivalent. In the next section, we will study more concretely how D_2 approximates $\sqrt{1 - \text{SSIM}}$ for common image distortions.

To define a metric that approximates the full SSIM index, we need to combine the local scores. Given $\mathbf{x} \in \mathbf{R}^N$ an image, we denote as $\mathbf{x}_i \in \mathbf{R}^n$ the extraction of the i-th image block. Let $N - B$ be the numbers of over-lapping blocks in the image $\mathbf{x}$. A cone metric over the whole image can then be defined as

$$\mathbf{D}_p(\mathbf{x}, \mathbf{y}) = (D_p(\mathbf{x}_1, \mathbf{y}_1), D_p(\mathbf{x}_2, \mathbf{y}_2), \ldots, D_p(\mathbf{x}_{N-B}, \mathbf{y}_{N-B})), \tag{2.64}$$

where $D_p := \|(d_1, d_2)\|_p$. Note that the subset of $\{\mathbf{x}_i\}_{i=1}^{N-B}$ formed with non-overlapping blocks represents an orthogonal decomposition of $\mathbf{x}$. We can thus apply Theorem 2.51 to show that

$$D_{p,q} := \|\mathbf{D}_p\|_q \tag{2.65}$$

is a metric.

We consider two examples of metrics to be compared with the SSIM index: $D_{2,1}$ and $D_{2,2}$. We proposed the first case in [15] as an approximation to the SSIM index. In the first case,

$$\begin{aligned} D_{2,1}(\mathbf{x},\mathbf{y}) &= \sum_{i=1}^{N-B} D_2(\mathbf{x}_i,\mathbf{y}_i) \\ &= \sum_{i=1}^{N-B} \sqrt{d_1^2(\mathbf{x}_i,\mathbf{y}_i) + d_2^2(\mathbf{x}_i,\mathbf{y}_i)} \\ &= \sum_{i=1}^{N-B} \sqrt{2 - S_1(\mathbf{x}_i,\mathbf{y}_i) - S_2(\mathbf{x}_i,\mathbf{y}_i)}, \end{aligned} \tag{2.66}$$

whereas in the second case

$$\begin{aligned} D_{2,2}(\mathbf{x},\mathbf{y}) &= \sqrt{\sum_{i=1}^{N-B} D_2^2(\mathbf{x}_i,\mathbf{y}_i)} \\ &= \sqrt{\sum_{i=1}^{N-B} d_1^2(\mathbf{x}_i,\mathbf{y}_i) + d_2^2(\mathbf{x}_i,\mathbf{y}_i)} \\ &= \sqrt{2 - \sum_{i=1}^{N-B} (S_1(\mathbf{x}_i,\mathbf{y}_i) + S_2(\mathbf{x}_i,\mathbf{y}_i))}. \end{aligned} \tag{2.67}$$

This has to be compared with

$$\sqrt{1 - \mathrm{SSIM}(\mathbf{x},\mathbf{y})} = \sqrt{1 - \sum_{i=1}^{N-B} S_1(\mathbf{x}_i,\mathbf{y}_i) S_2(\mathbf{x}_i,\mathbf{y}_i)}. \tag{2.68}$$

2.4.1 Psycho-visual Validation of Normalized Metrics

For practical applications, the metric obtained from the modification of SSIM is essentially equivalent to the original SSIM index and even performs slightly better in the case of $D_{2,2}$. This assertion is shown via a perceptual experiment.

Most typical image distortions $Y = \mathscr{D}(X)$, e.g. additive noise, blur, image compression, roughly preserve the local mean of images, so that $\bar{x} \simeq \bar{y}$. This explains the observation by Rouse and Hemami [90] on the LIVE database [102] that the contrast and structure components of the SSIM index (1.16) are the most important to predict the human perception of image distortion and justifies the use of $D_{2,1}$ or $D_{2,2}$ as an approximation of SSIM. However, in the case that there exist significant changes in the mean image intensity, as in the TID2008 database [79, 80], such an approximation is no longer valid.

To illustrate this fact, in Fig. 2.1, the SSIM index is compared with the score given by the metric $D_{2,2}$ on the 1700 distorted images of the TID2008 database. As in [114], the quality maps were computed on 11×11 local patches with circular Gaussian weighting (standard deviation of 1.5 pixels) of a downsampled image and then pooled into a single index by averaging. The resulting Pearson's correlation coefficient between $D_{2,2}$ and $\sqrt{1 - SSIM}$ is 0.9998, showing a nearly perfect relationship. As expected, the same experiment with the LIVE database gives also a very high correlation of 0.9986.

The Mean Opinion Scores (MOS) of the image quality of both TID2008 and LIVE database, release 2, are compared with $D_{2,1}$, $D_{2,2}$ and SSIM in Table 2.1. Two kinds of rank correlation were computed: Spearman's correlation and Kendall's correlation (see Appendix A for details). The metric $D_{2,1}$ matches very well the SSIM index, particularly in the presence of distortions preserving the mean (e.g. noise distortions, denoising, blurring and compression), but does not correlate with the MOS as well as the SSIM index in general. On the other hand, the new metric $D_{2,2}$ compares advantageously to the SSIM index, with even a slight gain in rank correlation for the TID-2008 database.

This is an important result. Not only we showed how the SSIM index can be modified into a valid metric distance, but a psychovisual test revealed that this new metric performs at least as well as the SSIM image quality assessment measure. Let us call this new $D_{2,2}$ normalized metric the *SSIM metric*. Remark that as it can be seen by comparing (2.67) and (2.68), the main difference between the SSIM metric and the SSIM index is the fact that the former adds all the local components whereas the latter takes a sum of the product of the local components.

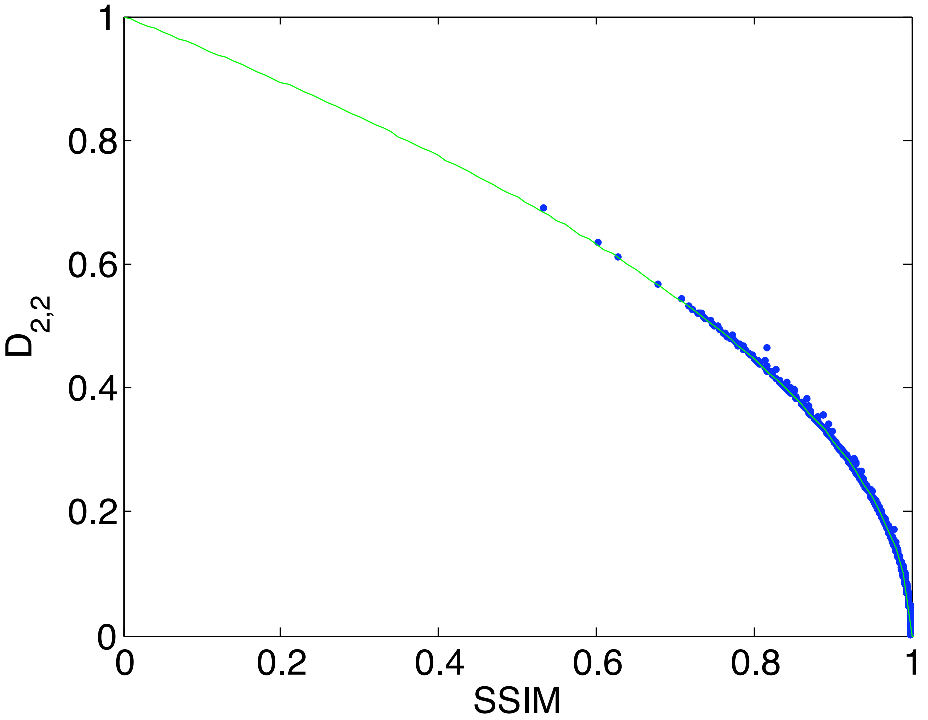


Figure 2.1: Comparison of the SSIM index and of the metric $D_{2,2}$ for the distorted images of the TID-2008 database. It is observed that the data follows extremely closely $\sqrt{1 - SSIM}$.

Table 2.1: Correlation between MOS and image quality assessment measures.

Correlation	Measure	LIVE	TID2008
Kendall's tau	SSIM	0.718	0.577
	$D_{2,1}$	0.712	0.501
	$D_{2,2}$	0.718	0.581
Spearman's rho	SSIM	0.900	0.775
	$D_{2,1}$	0.896	0.667
	$D_{2,2}$	0.900	0.780

Chapter 3

Properties of the SSIM Index

One reason for the popularity of the Mean Squared Error (MSE) in image processing applications is its mathematical convenience (see [113]). Indeed, the MSE is convex, differentiable and invariant under orthogonal transforms. Moreover, its square root, the so-called *Root Mean Squared Error (RMSE)* satisfies the properties of a metric. The problem with the MSE, however, is that it does not model the perception of image quality well.

In this chapter, we make a detailed study of the components of the Structural Similarity (SSIM) index. The goal is to show that they share many properties of the L^2-distance and thus are an attractive alternative to L^2. The invariance, symmetry, local convexity and quasi-convexity of the SSIM index will be studied. These properties are important for the optimization of the SSIM index. As an illustration, we will use some of these results to find an analytical solution of the projection on a convex set according to the SSIM index.

3.1 Invariance and Symmetry Properties

Invariance and symmetry are two fundamental properties for the simplification of the SSIM index and represent the cornerstone in many subsequent proofs. Essentially, symmetry allows to reduce the dimension of the signal to be studied whereas invariance permits to reduce the number of configurations to be studied. As it is well known, invariance under orthogonal transforms also play an important role when passing from spatial to frequency domain.

The following properties will hold for zero-mean signals $\mathbf{x}$ and $\mathbf{y}$ in $\mathbf{R}^N$ and for any quality measure of the form

$$q(\mathbf{x},\mathbf{y}) \;=\; f(\|\mathbf{x}\|_2, \|\mathbf{y}\|_2, \cos(\theta)), \tag{3.1}$$

where θ is the angle between $\mathbf{x}$ and $\mathbf{y}$. This includes normalized metrics:

$$\begin{aligned}\rho_M(\mathbf{x},\mathbf{y}) &:= \frac{\|\mathbf{x}-\mathbf{y}\|_2}{M(\|\mathbf{x}\|_2, \|\mathbf{y}\|_2)} && (3.2)\\ &= \frac{\sqrt{\|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2\|\mathbf{x}\|_2\|\mathbf{y}\|_2\cos(\theta)}}{M(\|\mathbf{x}\|_2, \|\mathbf{y}\|_2)} && (3.3)\end{aligned}$$

and the SSIM index of a zero-mean signal:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) \;:=\; \left(\frac{2s_xs_y+\epsilon_2}{s_x^2+s_y^2+\epsilon_2}\right)^{\beta}\left(\frac{s_{xy}+\epsilon_3}{s_xs_y+\epsilon_3}\right). \tag{3.4}$$

Note that when $\beta = 1$ and $\epsilon_3 = \epsilon_2/2$, the zero-mean SSIM index can be transformed in a particular case of a normalized metric via

$$\mathrm{NRMSE}_\epsilon(\mathbf{x},\mathbf{y}) := \sqrt{1-\mathrm{SSIM}_\epsilon(\mathbf{x},\mathbf{y})}. \tag{3.5}$$

The following property is analogous to Parseval's identity for L^2-norm.

Proposition 3.1.1. *The quality measure $q(\mathbf{x},\mathbf{y})$ is invariant under isometries.*

Proof. If U is an orthogonal matrix, then

$$\begin{aligned}q(U\mathbf{x},U\mathbf{y}) &= f\left(\|U\mathbf{x}\|_2, \|U\mathbf{y}\|_2, \frac{\langle U\mathbf{x}, U\mathbf{y}\rangle}{\|U\mathbf{x}\|_2\|U\mathbf{y}\|_2}\right)\\ &= f\left(\|\mathbf{x}\|_2, \|\mathbf{y}\|_2, \frac{\mathbf{x}^TU^TU\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2}\right)\\ &= q(\mathbf{x},\mathbf{y}). && (3.6)\end{aligned}$$

□

This explains the preserved SSIM measure in the discrete cosine transform (DCT) domain [23].

The next property of invariance is satisfied by only some particular normalized metrics.

Proposition 3.1.2. *If $M(\|\mathbf{x}\|_2, \|\mathbf{y}\|_2)$ is homogeneous, then $\rho_M(x, y)$ is invariant under scaling.*

Proof. If $\alpha > 0$, then

$$\begin{aligned} \rho_M(\alpha\mathbf{x}, \alpha\mathbf{y}) &= \frac{\|\alpha\mathbf{x} - \alpha\mathbf{y}\|_2}{M(\|\alpha\mathbf{x}\|_2, \|\alpha\mathbf{y}\|_2)} && (3.7) \\ &= \frac{\alpha\|\mathbf{x} - \mathbf{y}\|_2}{\alpha M(\|\mathbf{x}\|_2, \|\mathbf{y}\|_2)} \\ &= \rho_M(\mathbf{x}, \mathbf{y}). && (3.8) \end{aligned}$$

□

For the SSIM index, the scale invariance property is valid only when its stability constant ϵ equals 0. For $\epsilon > 0$,

$$\begin{aligned} \text{SSIM}_\epsilon(\alpha\mathbf{x}, \alpha\mathbf{y}) &= \frac{2\langle\alpha\mathbf{x}, \alpha\mathbf{y}\rangle + \epsilon}{\|\alpha\mathbf{x}\|_3^2 + \|\alpha\mathbf{y}\|_2^2 + \epsilon} \\ &= \frac{2\langle\mathbf{x}, \mathbf{y}\rangle + \epsilon/\alpha^2}{\|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_3^2 + \epsilon/\alpha^2} && (3.9) \\ &= \text{SSIM}_{\epsilon/\alpha^2}(\mathbf{x}, \mathbf{y}), \end{aligned}$$

which is very close to scaling invariance if ϵ and ϵ/α^2 are negligible in comparison to $\|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2$.

The next property will be very useful for analyzing the SSIM$(\mathbf{x}, \mathbf{y})$ when $\mathbf{y}$ is fixed, thus reducing the dimensions of interest to only three.

Proposition 3.1.3. *For $\mathbf{x}, \mathbf{y} \in \mathbf{R}^N$ $(N > 1)$ with $\mathbf{y}$ fixed, $g(\mathbf{x}) := q(\mathbf{x}, \mathbf{y})$ exhibits a rotational symmetry around the direction of $\mathbf{y}$.*

Proof. Consider any $\mathbf{x}_{(1)} = a\mathbf{y} + b\mathbf{u}$ and $\mathbf{x}_{(2)} = a\mathbf{y} + b\mathbf{v}$ where $\mathbf{u}, \mathbf{v} \in \mathbf{R}^N$ with $\langle\mathbf{y}, \mathbf{u}\rangle = \langle\mathbf{y}, \mathbf{v}\rangle = 0$ and $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$. In this case,

$$g(\mathbf{x}_{(1)}) = g(\mathbf{x}_{(2)}), \qquad (3.10)$$

since $\|\mathbf{x}_{(1)}\| = \|\mathbf{x}_{(2)}\|$ and $\langle\mathbf{y}, \mathbf{x}_{(1)}\rangle = \langle\mathbf{y}, \mathbf{x}_{(2)}\rangle$ for all $a, b \in \mathbf{R}$ and for all $\mathbf{y} \in \mathbf{R}^N$. □

The next and final preliminary property of $\rho_M(\mathbf{x}, \mathbf{y})$, which will also apply to the SSIM index, is useful in finding its region of convexity.

Proposition 3.1.4. *For $\mathbf{x}, \mathbf{y} \in \mathbf{R}^N$ with $N > 1$, $g(\mathbf{x}) := \rho_M(\mathbf{x}, \mathbf{y})$ is monotonically increasing perpendicularly to the direction of $\mathbf{y}$.*

Proof. Consider any $\mathbf{x}_{(1)} = a\mathbf{y} + b\mathbf{u}$ and $\mathbf{x}_{(2)} = a\mathbf{y} + b'\mathbf{u}$ where $\mathbf{u} \in \mathbf{R}^N$ with $\langle \mathbf{y}, \mathbf{u} \rangle = 0$ and $\|\mathbf{u}\| = 1$. In this case,

$$\begin{aligned} \rho_M(\mathbf{x}_{(1)}, \mathbf{y}) &= \frac{\sqrt{(a-1)^2\|\mathbf{y}\|^2 + b^2}}{M(\sqrt{a^2\|\mathbf{y}\|^2 + b^2}, \|\mathbf{y}\|)} \\ &\leq \frac{\sqrt{(a-1)^2\|\mathbf{y}\|^2 + b'^2}}{M(\sqrt{a^2\|\mathbf{y}\|^2 + b'^2}, \|\mathbf{y}\|)} \\ &= \rho_M(\mathbf{x}_{(2)}, \mathbf{y}), \qquad (3.11) \end{aligned}$$

for all $a \in \mathbf{R}, b \leq b' \in \mathbf{R}$ and for all $\mathbf{y} \in \mathbf{R}^N$. □

3.2 Level Sets of the SSIM Index

We are interested in understanding the shape of the SSIM index and of normalized metrics by studying their level sets. The rewards will be manyfold. Firstly, it could help to select the parameters α, β of the SSIM index (1.16) or p, q of normalized metrics (2.27) which correspond more faithfully to human perception. Secondly, the shape of the level sets enters in the quasi-convexity definition. Finally, it will be useful for finding the projection of the SSIM index onto convex sets.

Any signal $\mathbf{y} \in \mathbf{R}^N_+$ can be expressed as a sum of a mean term in the direction of $\mathbf{1} = (1, 1, \ldots, 1)$ and a zero-mean term, i.e.

$$\begin{aligned} \mathbf{y} &= \bar{y}\mathbf{1} + (\mathbf{y} - \bar{y}\mathbf{1}), \qquad (3.12) \\ &=: \bar{\mathbf{y}} + \mathbf{y}_0. \qquad (3.13) \end{aligned}$$

Taking the simplified form of the SSIM index (1.21), we observe that S_1 depends only on the mean of the signal $\bar{y}$ and that S_2 depends only on its zero-mean component $\mathbf{y}_0$. This decomposition allows us to study each component independently.

By Proposition 3.1.3, the SSIM index exhibits a rotational symmetry around the direction of $\mathbf{y}_0$. It means that we only need three directions to fully describe the SSIM index: $\frac{\mathbf{1}}{\sqrt{N}}, \frac{\mathbf{y}_0}{\|\mathbf{y}_0\|}$ and any direction perpendicular to $\bar{\mathbf{y}}$ and $\mathbf{y}_0$. In this section, we describe the level sets of the SSIM index and of normalized metrics in the two-dimensional zero-mean space and compare them with the level sets of the MSE.

A single *k-level set* of the SSIM index consists of all the points

$$\{\mathbf{x} \in \mathbf{R}^2 : \mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = k\}. \tag{3.14}$$

By rotational invariance of the SSIM index (Proposition 3.1.1), it can be first considered that $\mathbf{y} = (\|\mathbf{y}\|, 0)$ since all the level sets will be a rotation of this one. The signal $\mathbf{x}$ can then be written in polar coordinates as

$$\mathbf{x} \;=\; (r\cos(\theta), r\sin(\theta)). \tag{3.15}$$

The equation that needs to be solved is thus

$$\left(\frac{2r\|\mathbf{y}\| + \epsilon_2}{r^2 + \|\mathbf{y}\|^2 + \epsilon_2}\right)^{\beta} \left(\frac{r\|\mathbf{y}\|\cos(\theta) + \epsilon_3}{r\|\mathbf{y}\| + \epsilon_3}\right) \;=\; k, \tag{3.16}$$

or isolating $\cos(\theta)$,

$$\cos(\theta) \;=\; k\left(1 + \frac{\epsilon_3}{r\|y\|}\right)\left(\frac{r^2 + \|\mathbf{y}\|^2 + \epsilon_2}{2r\|\mathbf{y}\| + \epsilon_2}\right)^{\beta} - \frac{\epsilon_3}{r\|\mathbf{y}\|}. \tag{3.17}$$

Since we were able to isolate $\cos(\theta)$, it means that there will be a unique value of $\theta \geq 0$ for each r. On the other hand, the maximum value for the left hand side is $\cos(\theta) = 1$, leading to

$$r = \frac{\|\mathbf{y}\| \pm \sqrt{\|\mathbf{y}\|^2 - k^{2/\beta}(\|\mathbf{y}\|^2 + \epsilon_2) + \epsilon_2 k^{1/\beta}}}{k^{1/\beta}}. \tag{3.18}$$

So by varying r between the bounds given in (3.18), it is possible to find all values of $\cos(\theta)$ and to draw the level sets from the parametric curves for $\theta \geq 0$ and $\theta \leq 0$.

Figures 3.1 to 3.4 illustrate the level sets for MSE and for the zero-mean component of the SSIM index with zero stability constants (drawn in red). We remark that the level sets for the SSIM index with $\beta = 1$ are circles scaled and shifted out of the center. For $\beta < 1$, the level sets take an egg-shape and becomes more elongated as β decreases. In the limit, the level sets tend to half-lines following the direction of its center. On the other hand, the level sets for $\beta > 1$ have a more bulky shape and bend along the circle of constant magnitude. This flexibility in the shape of the level sets could be very interesting to model any similarity measure depending only on the magnitude and the angle between two points.

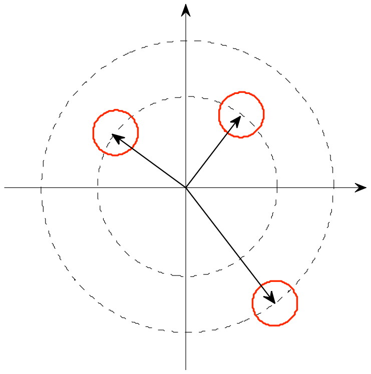


Figure 3.1: Level sets of MSE.

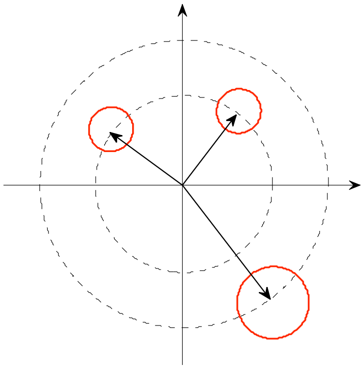


Figure 3.2: Level sets of SSIM with $\beta = 1$.

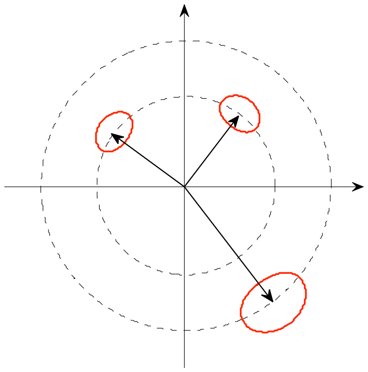


Figure 3.3: Level sets of SSIM with $\beta = 2$.

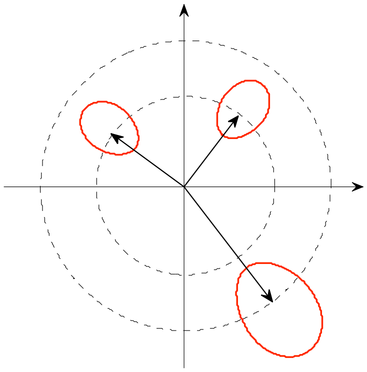


Figure 3.4: Level sets of SSIM with $\beta = 0.5$.

We compare the level sets of the SSIM index with the level sets of pq-normalized metrics of the form

$$\rho_{p,q,c}(\mathbf{x}, \mathbf{y}) \quad = \quad \frac{\|\mathbf{x} - \mathbf{y}\|_2}{(\|\mathbf{x}\|_2^p + \|\mathbf{y}\|_2^p + c^p)^{q/p}} \tag{3.19}$$

with $q \leq 1$. By rotational symmetry, the level sets also reduce to two dimensions. Again writing $\mathbf{y} = (\|\mathbf{y}\|, 0)$ and $\mathbf{x}$ in polar coordinates, the equation describing the level sets become

$$\frac{r^2 - 2r\|\mathbf{y}\| \cos(\theta) + \|\mathbf{y}\|^2}{(r^p + \|\mathbf{y}\|^p + c^p)^{2q/p}} \quad = \quad k^2. \tag{3.20}$$

Isolating $\cos(\theta)$ once again, it simplifies to

$$\cos(\theta) \;=\; \frac{r^2+\|\mathbf{y}\|^2}{2r\|\mathbf{y}\|} - k^2\frac{(r^p+\|\mathbf{y}\|^p+c^p)^{2q/p}}{2r\|\mathbf{y}\|}. \tag{3.21}$$

As expected, there will be a unique value of $\theta \geq 0$ for each r. For $\cos(\theta) = 1$, the equation simplifies to

$$f(r) \;:=\; r^p + \|\mathbf{y}\|^p + c^p - \left|\frac{r-\|\mathbf{y}\|}{k}\right|^{p/q}. \tag{3.22}$$

If $k \leq \|\mathbf{y}\|/(\|\mathbf{y}\|^p + c^p)^{q/p}$, then this equation has two and only two solutions: one with $r < \|\mathbf{y}\|$ and one with $r > \|\mathbf{y}\|$. Indeed, for existence notice that

$$f(0) = \|\mathbf{y}\|^p + c^p - (\|\mathbf{y}\|/k)^{p/q} \leq 0 \text{ if and only if } k \leq \|\mathbf{y}\|/(\|\mathbf{y}\|^p + c^p)^{q/p}$$

and $f(\|\mathbf{y}\|) = 2\|\mathbf{y}\|^p + c^p > 0$. For $r < \|\mathbf{y}\|$, $f(r)$ is increasing in r. So if a solution exists for $0 \leq r < \|\mathbf{y}\|$, it is unique. Also, notice that $\lim_{r\to\infty} f(r) < 0$ ensures the existence of the solution for $r > \|\mathbf{y}\|$. Since $q \leq 1$ and $k < 1$, $f(r)$ is decreasing for $r > \|\mathbf{y}\|$ and the solution is unique. Thus we can find numerically the minimum and the maximum value of r and draw the level sets from the parameterization of $\cos(\theta)$ in function of r. Two examples of level sets for $p = 1/4$ and $p = 16$ are shown in Fig. 3.5 and 3.6. For the case $p > 2$,

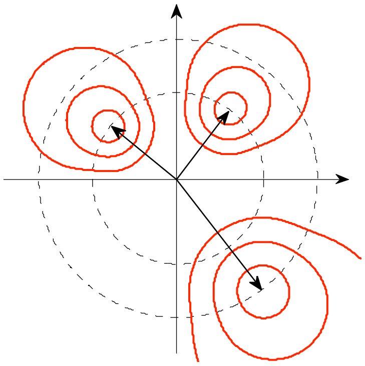


Figure 3.5: Level sets of pq-normalized metric with $p = 16$, $q = 1$ and $c = 0$.

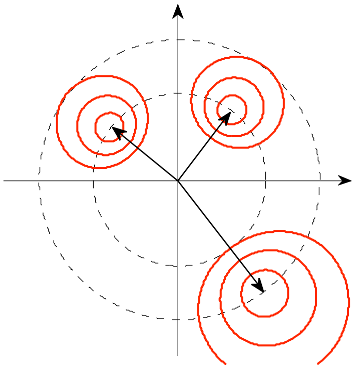


Figure 3.6: Level sets of pq-normalized metric with $p = 1/4$, $q = 1$ and $c = 0$.

3.3 Convexity Properties

The general problem of optimization can be stated as follows:

Given a function $f : X \to \mathbf{R}$, find $x \in X$ such that $f(x)$ is minimized.

Without knowing any particular properties of f, the only way to solve the problem is to check all possible values of $x \in X$. This is not a very attractive solution, particularly when X has a very large or even infinite dimension. This is why we need to specify some properties of f and X that would make the problem more tractable.

The most desirable property of f and X is convexity:

Definition 3.3.1. *A subset X of a vector space is* convex *if for all $x, y \in X$ and for all $\lambda \in (0, 1)$,*

$$\lambda x + (1 - \lambda)y \in X. \tag{3.23}$$

Definition 3.3.2. *A function $f : X \to \mathbf{R}$ is* convex *if for all $x, y \in X$ and for all $\lambda \in (0, 1)$,*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \tag{3.24}$$

The function f is said to be strictly convex *if the inequality is strict.*

With these properties, uniqueness of the minimum is guaranteed.

Theorem 3.3.3. *Let X be a convex vector space and let $f : X \to \mathbf{R}$ be a strictly convex function. If f has a minimum, then it is unique.*

In particular, this has useful applications for numerical optimization. For example, if the function f is differentiable, then the fixed point of the gradient descent algorithm (if it exists) is always the global minimum.

A weaker form of convexity is quasi-convexity.

Definition 3.3.4. *Given a convex set X, a function $f : X \to \mathbf{R}$ is said to be* quasi-convex *if its h-*sublevel set*, defined as*

$$X_h = \{x \in X | f(x) \leq h\}, \tag{3.25}$$

is a convex set for all $h \in Range(f)$.

Quasi-convexity is a useful property for non-linear optimization [6]:

Theorem 3.3.5. *Let X be a convex vector space and let $f : X \to \mathbf{R}$ be a quasi-convex function. If f has a minimum, then it is either unique or the function is constant in a neighbourhood of the minimum.*

These two properties, convexity and quasi-convexity, will be studied for the components of the SSIM index. The goal is to provide important tools for the numerical or analytical optimization of the SSIM index or of one of its modifications. Our belief is that, in virtue of these convexity properties, numerical optimization of the SSIM index is not much harder than optimizing L^p-distances. For the interested reader, some details on optimization techniques for quasi-convex functions can be found in [6].

3.3.1 Quasi-Convexity

We show the region of quasi-convexity of the components of the SSIM index and outline how to generalize this result for normalized metrics.

Definition 3.3.6. *For $\mathbf{x}, \mathbf{y} \in \mathbf{R}^N$, the* normalized mean squared error (NMSE) *with stability constant c is defined as*

$$NMSE_c(\mathbf{x}, \mathbf{y}) \quad := \quad \frac{\|\mathbf{x} - \mathbf{y}\|^2}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 + c}. \tag{3.26}$$

It is related to the mean component S_1 and zero-mean component S_2 of the SSIM index via $\mathrm{NMSE}_c(\mathbf{x}, \mathbf{y}) = 1 - S_i$ for $i = 1, 2$ and for $\mathbf{x}, \mathbf{y} \in \mathbf{R}_+$ and $\mathbf{R}^{N-1}$ respectively.

Theorem 3.3.7. *For $\mathbf{y}$ fixed, $NMSE_c(\mathbf{x}, \mathbf{y})$ is quasi-convex on*

$$H = \left\{ \mathbf{x} \in \mathbf{R}^N : \langle \mathbf{x}, \mathbf{y} \rangle \geq -\frac{c}{2} \right\}.$$

Proof. We easily see that

$$x \in H \iff \mathrm{NMSE}_c(\mathbf{x}, \mathbf{y}) \leq 1. \tag{3.27}$$

So it is sufficient to show that X_h is convex for all $h \leq 1$. To show that, we first examine the inequality

$$\mathrm{NMSE}_c(\mathbf{x}, \mathbf{y}) \quad = \quad \frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{\|\mathbf{x}\|_2^2 + \|y\|_2^2 + c} \leq h. \tag{3.28}$$

For $h < 1$, this inequality becomes

$$\|\mathbf{x}\|_2^2 - \frac{2}{1-h^2}\langle \mathbf{y}, \mathbf{x}\rangle \leq -\|\mathbf{y}\|_2^2 + c\frac{h^2}{1-h^2}. \tag{3.29}$$

Completing the square yields

$$\left\| \mathbf{x} - \frac{\mathbf{y}}{1-h^2} \right\|^2 \leq \|\mathbf{y}\|_2^2 \left(\frac{1}{(1-h^2)^2} - 1 \right) + c\frac{h^2}{1-h^2}. \tag{3.30}$$

The set of points $\mathbf{x}$ for which this inequality holds is a hyper-sphere, which is a convex set.

In the particular case when $h = 1$, we have the inequality

$$-2\langle \mathbf{y}, \mathbf{x}\rangle \leq c \tag{3.31}$$

which is satisfied for all $\mathbf{x} \in H$, as defined earlier. The set H is bounded by a hyper-plane and therefore is convex. □

The quasi-convexity region H_1 of $S_1(\bar{x}, \bar{y})$ is thus

$$H_1 \quad = \quad \left\{ \bar{x} : \bar{x}\bar{y} \geq -\frac{\epsilon_1}{2} \right\}, \tag{3.32}$$

and the quasi-convexity region H_2 of $S_2(\mathbf{x} - \bar{\mathbf{x}}, \mathbf{y} - \bar{\mathbf{y}})$ is

$$H_2 \quad = \quad \left\{ \mathbf{x} - \bar{\mathbf{x}} : \langle \mathbf{x} - \bar{\mathbf{x}}, \mathbf{y} - \bar{\mathbf{y}} \rangle \geq -\frac{\epsilon_2}{2} \right\}. \tag{3.33}$$

In practice, $\bar{x}$ and $\bar{y}$ are positive and thus S_1 is quasi-convex on its whole domain of definition.

In general, to verify if a compact and simply connected subset of $\mathbf{R}^2$ is convex, it is possible to study the sign of curvature of the parametrization of its boundary curve:

Theorem 3.3.8. *A simple closed curve Γ is convex if and only if*

$$\gamma'(t) \times \gamma''(t) \geq 0, \quad \textit{for all } t \in [0, 1], \tag{3.34}$$

where $\gamma(t)$, $t \in [0, 1]$, is a parametrization of Γ oriented counter-clockwise.

We already found a parameterization of normalized metrics and the SSIM index in function of the radius r in polar coordinates. Deriving twice and computing the cross-product, it is mandatory to check the sign of the curvature at every point and for every level set. If the minimum of the curvature is greater or equal to zero everywhere (going in an anti-clockwise orientation), then the curve is convex and the function is quasi-convex. It is possible to carry on the computations given a set of parameters ϵ_2, ϵ_3 and β for the second component of the SSIM index or given c, p, q for pq-normalized metrics, but in general it is hard to determine for which parameters we will have quasi-convexity.

For $\mathbf{y}$ fixed, it is observed empirically (see Figs. 3.5 and 3.6) that normalized metrics are locally quasi-convex around $\mathbf{y}$. Also, from the inspection of the level sets, it is conjectured that $\mathrm{SSIM}(\mathbf{x}, \mathbf{y})$ is quasi-convex for $\beta \leq 1$. For $\beta > 1$, the convexity of the level sets might eventually break down when r is minimal. For the pq-normalized metrics, it is observed that the quasi-convexity eventually breaks down when $r = \|\mathbf{y}\|$ for $p > 2$ and when r is minimal for $p < 2$.

3.3.2 Region of Convexity

For $\mathbf{y}$ fixed, we wish to find the set of all points where $g(\mathbf{x}) = \mathrm{NMSE}_c(\mathbf{x}, \mathbf{y})$ is convex. In particular, we shall show that $\mathrm{NMSE}_c(\mathbf{x}, \mathbf{y})$ is locally convex, i.e. there exists a sphere around the minimum $\mathbf{x} = \mathbf{y}$ for which $\mathrm{NMSE}_c(\mathbf{x}, \mathbf{y})$ is convex. For this, the following technical lemma is needed.

Lemma 3.3.9. *Let X be a two-dimensional convex subset of $\mathbf{R}^N$ and let $\mathbf{y} \in X$. If f is convex on X, has rotational symmetry around the direction of $\mathbf{y}$ and is monotonically increasing in a direction perpendicular to $\mathbf{y}$, then f is convex on the rotation of X around $\mathbf{y}$.*

Proof. Consider any two points $\mathbf{x}_{(1)}$ and $\mathbf{x}_{(2)}$ in the region consisting of the rotation of X around the direction of $\mathbf{y}$ and consider a third point

$$\mathbf{z} \;=\; \lambda \mathbf{x}_{(1)} + (1-\lambda)\mathbf{x}_{(2)}, \text{ where } 0 \leq \lambda \leq 1 \tag{3.35}$$

somewhere in the line segment joining these two points. For convexity of f, we need to show that

$$f(\mathbf{z}) \;\leq\; \lambda f(\mathbf{x}_{(1)}) + (1-\lambda) f(\mathbf{x}_{(2)}). \tag{3.36}$$

We decompose $\mathbf{x}_{(1)}$ and $\mathbf{x}_{(2)}$ as $\mathbf{x}_{(1)} = a_1\mathbf{y} + b_1\mathbf{u}$ and $\mathbf{x}_{(2)} = a_2\mathbf{y} + b_2\mathbf{v}$ where $\mathbf{u}$ and $\mathbf{v}$ are unit vectors perpendicular to $\mathbf{y}$, i.e., $\langle \mathbf{y}, \mathbf{u} \rangle = \langle \mathbf{y}, \mathbf{v} \rangle = 0$ and $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$. Now, consider two points $\mathbf{x}_{(3)} = a_1\mathbf{y} + b_1\mathbf{v}$ and $\mathbf{x}_{(4)} = a_2\mathbf{y} + b_2\mathbf{u}$ and construct two other points $\mathbf{x}_{(5)} = \lambda\mathbf{x}_{(1)} + (1-\lambda)\mathbf{x}_{(4)}$ and $\mathbf{x}_{(6)} = \lambda\mathbf{x}_{(2)} + (1-\lambda)\mathbf{x}_{(3)}$ (see Fig. 3.7).

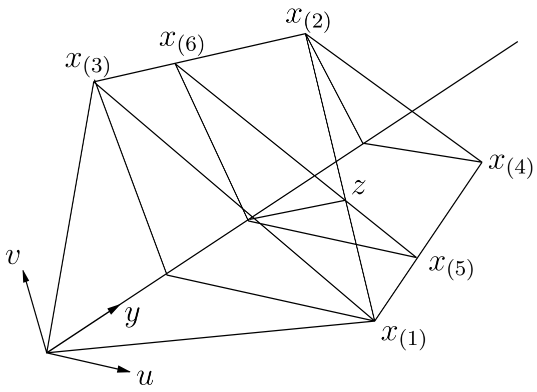


Figure 3.7: Points $x_{(1)}$, $x_{(2)}$, ..., $x_{(6)}$ and z in the proof of Lemma 3.3.9.

Considering

$$\mathbf{z}' \;=\; \mu \mathbf{x}_{(5)} + (1-\mu)\mathbf{x}_{(6)} \tag{3.37}$$

where

$$\mu \;=\; \frac{\lambda b_1}{\lambda b_1 + (1-\lambda) b_2}, \tag{3.38}$$

it can be checked that in fact $\mathbf{z} = \mathbf{z}'$.

By convexity of f in the two-dimensional region X and by rotational symmetry, we have

$$f(\mathbf{x}_{(5)}) \;\leq\; \lambda f(\mathbf{x}_{(1)}) + (1-\lambda) f(\mathbf{x}_{(4)}) \tag{3.39}$$

and

$$f(\mathbf{x}_{(6)}) \;\leq\; \lambda f(\mathbf{x}_{(2)}) + (1-\lambda) f(\mathbf{x}_{(3)}). \tag{3.40}$$

Now, we show that

$$f(\mathbf{z}) \;\leq\; \mu f(\mathbf{x}_{(5)}) + (1-\mu) f(\mathbf{x}_{(6)}). \tag{3.41}$$

Since $\mathbf{x}_{(5)}$ and $\mathbf{x}_{(6)}$ have the same distance to the line generated by $\mathbf{y}$, then by rotational symmetry $f(\mathbf{x}_{(5)}) = f(\mathbf{x}_{(6)})$.

It remains to show that $f(\mathbf{z}) \leq f(\mathbf{x}_{(5)})$. We write

$$\begin{aligned} \mathbf{z} &= (\lambda a_1 + (1-\lambda)a_2)\mathbf{y} + \lambda b_1 \mathbf{u} + (1-\lambda) b_2 \mathbf{v} && (3.42) \\ &= a\mathbf{y} + b\mathbf{w} && (3.43) \end{aligned}$$

with $\langle \mathbf{w}, \mathbf{y} \rangle = 0$ and $\|\mathbf{w}\| = 1$, and

$$\begin{aligned} \mathbf{x}_{(5)} &= (\lambda a_1 + (1-\lambda)a_2)\mathbf{y} + (\lambda b_1 + (1-\lambda) b_2)\mathbf{u} && (3.44) \\ &= a\mathbf{y} + b'\mathbf{u} && (3.45) \end{aligned}$$

with $\langle \mathbf{u}, \mathbf{y} \rangle = 0$ and $\|\mathbf{u}\| = 1$.

From the triangular inequality for norms, we know that $b \leq b'$. We then use the fact that f is monotonicity increasing in a direction perpendicular to $\mathbf{y}$ to conclude that $f(\mathbf{z}) \leq f(\mathbf{x}_{(5)})$. From rotational symmetry, we derive $f(\mathbf{x}_{(4)}) = f(\mathbf{x}_{(2)})$. From all of the above,

$$\begin{aligned} f(\mathbf{z}) \leq f(\mathbf{x}_{(5)}) &\leq \lambda f(\mathbf{x}_{(1)}) + (1-\lambda) f(\mathbf{x}_{(4)}) && (3.46) \\ &= \lambda f(\mathbf{x}_{(1)}) + (1-\lambda) f(\mathbf{x}_{(2)}). && (3.47) \end{aligned}$$

□

Theorem 3.3.10. *For $x, y \in \mathbf{R}$, $g(x) = NMSE_0(x, y)$ is convex on*

$$0 \leq x \leq \sqrt{3}|y|. \tag{3.48}$$

For $\mathbf{x}, \mathbf{y} \in \mathbf{R}^N$ with $N > 1$, $g(\mathbf{x}) = NMSE_0(\mathbf{x}, \mathbf{y})$ is convex on the region bounded by the rotation of

$$R(t) = \sqrt{\frac{-2t^2 - \|\mathbf{y}\|^2 + \sqrt{16t^2\|\mathbf{y}\|^2 + \|\mathbf{y}\|^4}}{2}}, \tag{3.49}$$

with $0 \leq t \leq \sqrt{3}\|\mathbf{y}\|$, around the direction of $\mathbf{y}$. I.e. g is convex on the region bounded by $t\mathbf{y}/\|\mathbf{y}\| + R(t)\mathbf{u}/\|\mathbf{u}\|$ for any $\mathbf{u}$ perpendicular to $\mathbf{y}$ (see Fig. 3.8).

Proof. The one dimensional case can easily be verified by finding the region where the second derivative of $g(x)$ is positive. For the general case, we proceed in three steps.

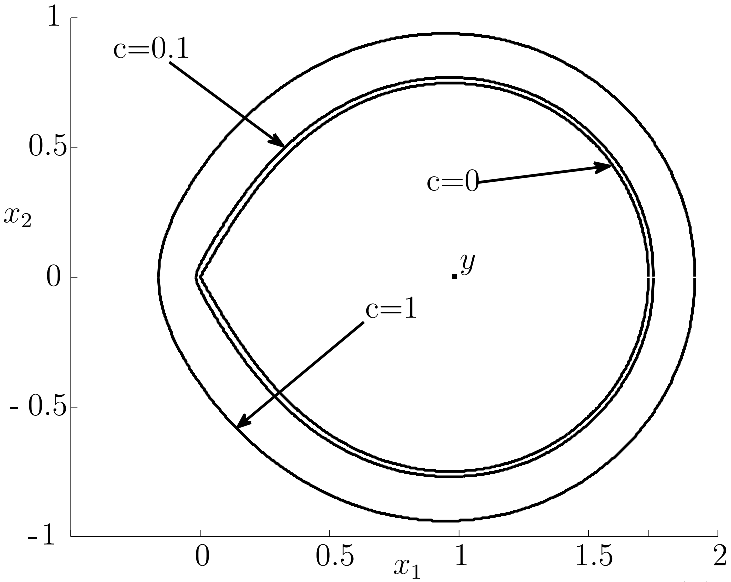


Figure 3.8: Boundary of the region where $g(x)$ = $\mathrm{NMSE}_c(x, y)$ is convex with $y = (1, 0)$ and $x = (x_1, x_2)$ for $c = 0$, $c = 0.1$ and $c = 1$.

1. Consider the two dimensional case and denote $\mathbf{x} = (x_1, x_2)$. For $\mathbf{y} = (1, 0)$, we have

$$g(\mathbf{x}) = \mathrm{NMSE}_0(\mathbf{x}, \mathbf{y}) \quad = \quad \frac{(x_1 - 1)^2 + x_2^2}{x_1^2 + x_2^2 + 1}. \tag{3.50}$$

To determine the region of convexity of g, it suffices to compute the eigenvalues of its Hessian matrix and determine when they are non-negative. It can be checked that this happens when

$$|x_2| \quad \leq \quad \sqrt{\frac{-2x_1^2 - 1 + \sqrt{16x_1^2 + 1}}{2}} =: R(x_1) \tag{3.51}$$

and

$$0 \leq \quad x_1 \quad \leq \sqrt{3}. \tag{3.52}$$

We shall call this region X.

2. For any $\mathbf{y} \in \mathbf{R}^2$, we can write $\mathbf{y} = (\|\mathbf{y}\| \cos(\theta), \|\mathbf{y}\| \sin(\theta))$ where θ is the angle

between $\mathbf{y}$ and $(1, 0)$. The vector $\mathbf{y}$ can be thought as a transformation of $e_1 = (1, 0)$:

$$\mathbf{y} \quad = \quad \|\mathbf{y}\|\mathrm{Rot}_\theta e_1. \tag{3.53}$$

By the invariance properties of NMSE (Prop. 3.1.1 and 3.1.2), we have

$$\mathrm{NMSE}_0(\mathbf{x}, \mathbf{y}) \quad = \quad \mathrm{NMSE}_0(\mathrm{Rot}_{-\theta}\mathbf{x}/\|\mathbf{y}\|, e_1). \tag{3.54}$$

So $\mathrm{NMSE}_0(\mathbf{x}, \mathbf{y})$ is convex on $\|\mathbf{y}\|\mathrm{Rot}_\theta X$, i.e. on a rotation of X by an angle θ and a scaling by a factor $\|y\|$.

3. Since $g(\mathbf{x}) = \mathrm{NMSE}_0(\mathbf{x}, \mathbf{y})$ is convex on the convex region X, by rotational symmetry and monotonicity of g (Prop. 3.1.3 and 3.1.4), it will also be convex in the region consisting of the rotation of X around the direction of $\mathbf{y}$ in virtue of Lemma 3.3.9.

□

Here we considered the case $c = 0$. The general case is similar but since we cannot use the scaling property, we need to consider $y = (\|\mathbf{y}\|, 0)$ in Step 1 which leads to a complicated equation for the region of convexity. The region with $y = (1, 0)$ and $c \neq 0$ will look very similar than the one for $c = 0$ but the region will be slightly larger. It adds approximately a "sausage" around the convexity region of diameter increasing with c (see Fig. 3.8). For practical purpose, c is usually very small ($c \leq 0.0001$) and the region of convexity of $\mathrm{NMSE}_c(\mathbf{x}, \mathbf{y})$ is almost identical to the region of convexity of $\mathrm{NMSE}_0(\mathbf{x}, \mathbf{y})$.

There is another region of convexity for some negative $\mathbf{x}$ but it is not of interest here since it is not around $\mathbf{y}$, the minimum of $\mathrm{NMSE}_c(\mathbf{x}, \mathbf{y})$. An easy estimate of the region of convexity is the hyper-sphere centered at $\mathbf{y}$ and of radius $(\sqrt{3}-1)\|\mathbf{y}\|$. Indeed, this sphere is inscribed in the region of convexity. From Theorem 3.3.10, we can directly deduce the convexity (or in fact, concavity) region for $S_1(\bar{x}, \bar{y})$ and for $S_2(\mathbf{x} - \bar{\mathbf{x}}, \mathbf{y} - \bar{\mathbf{y}})$.

3.3.3 Convexity of Cone Metrics

We generalize convexity for vector-valued metrics. In particular, it will apply to the metric $\mathbf{d}$ in (2.50).

Theorem 3.3.11. *If d_m is convex on R_m for $1 \leq m \leq M$, then the cone metric $\mathbf{d} = (d_1, d_2, \ldots, d_m)$ is convex on $R_1 \times \ldots \times R_M$ in the sense that, for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{R}^N$ and for* *all* $0 \leq \lambda \leq 1$*, the following ordering holds*

$$\mathbf{d}(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}, \mathbf{z}) \preceq \lambda\mathbf{d}(\mathbf{x}, \mathbf{z}) + (1-\lambda)\mathbf{d}(\mathbf{y}, \mathbf{z}), \tag{3.55}$$

where $\preceq$ *is the product order (2.44).*

Proof. Since the projection operator $P_m(\mathbf{x}) = \mathbf{x}_m$ is linear and d_m is convex on R_m for $1 \leq m \leq M$, we have

$$\begin{aligned} d_m(P_m(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}), P_m(\mathbf{z})) &= d_m(\lambda\mathbf{x}_m + (1-\lambda)\mathbf{y}_m, \mathbf{z}_m) \\ &\leq \lambda d_m(\mathbf{x}_m, \mathbf{z}_m) + (1-\lambda)d_m(\mathbf{y}_m, \mathbf{z}_m). \end{aligned} \tag{3.56}$$

Since the inequality holds for all $1 \leq m \leq M$, we conclude that $\mathbf{d}$ is convex on $R_1 \times \ldots \times R_M$ in the sense of (3.55). □ □

We can also show the convexity of $\|\mathbf{d}\|$ with the following theorem.

Theorem 3.3.12. *If* d_m *is convex on* R_m *for* $1 \leq m \leq M$ *and if* $\|\cdot\|$ *is a norm with the increasing property (see (2.47)), then* $\|\mathbf{d}\|$ *is convex on* $R_1 \times \ldots \times R_M$*.*

Proof. From Theorem 3.3.11, $\mathbf{d}$ is convex on $R_1 \times \ldots \times R_M$. Since $\|\cdot\|$ is a norm, it is convex as well. So by the increasing property, we have the following inequalities on $R_1 \times \ldots \times R_M$

$$\begin{aligned} \|\mathbf{d}(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}, \mathbf{z})\| &\leq \|\lambda\mathbf{d}(\mathbf{x}, \mathbf{z}) + (1-\lambda)\mathbf{d}(\mathbf{y}, \mathbf{z})\| \\ &\leq \lambda\|\mathbf{d}(\mathbf{x}, \mathbf{z})\| + (1-\lambda)\|\mathbf{d}(\mathbf{y}, \mathbf{z})\|. \end{aligned} \tag{3.57}$$

□

For our case of interest, we deduce that the region of convexity of $\mathbf{S} = (S_1(\mathbf{x}, \mathbf{y}), S_2(\mathbf{x}, \mathbf{y}))$ and of $\|\mathbf{S}\|_2$ is the (hyper)-cylinder with $0 \leq \bar{x} \leq \sqrt{3}\bar{y}$ and with $\mathbf{x} - \bar{\mathbf{x}}$ following a tear drop shape around $\mathbf{y} - \bar{\mathbf{y}}$ like it is described in Theorem 3.3.10.

Contrary to convex functions, the sum of quasi-convex functions is not necessarily quasi-convex. Nevertheless, we can find a similar theorem about the quasi-convexity of $\mathbf{d}$, but we cannot establish the same result for $\|\mathbf{d}\|$.

Theorem 3.3.13. *If* d_m *is quasi-convex on* H_m *for* $1 \leq m \leq M$*, then* $\mathbf{d}$ *is quasi-convex on* $H = H_1 \times \ldots \times H_M$*.*

Proof. Since d_m is quasi-convex on H_m for $1 \leq m \leq M$, we have that the cartesian product

$$\textstyle\prod_{m=1}^{M} \{P_m(\mathbf{x}) \in H_m | d_m(P_m(\mathbf{x}), P_m(\mathbf{y})) \leq h_m\} \tag{3.58}$$

is a convex set for all $h = (h_1, h_2, \ldots, h_M)$. By the Orthogonal Decomposition Theorem 2.3.5, (3.58) is equivalent to

$$\{\mathbf{x} = \textstyle\sum_{m=1}^{M} P_m(\mathbf{x}) \in H | \mathbf{d}(\mathbf{x}, \mathbf{y}) \preceq h\}, \tag{3.59}$$

which we know to be convex. Hence, $\mathbf{d}$ is quasi-convex. □

Therefore, $\mathbf{S} = (S_1(\mathbf{x}, \mathbf{y}), S_2(\mathbf{x}, \mathbf{y}))$ is quasi-convex in the region $H_1 \times H_2$, where H_1 and H_2 are the regions of quasi-convexity described in (3.32) and (3.33).

In summary, both of the components S_1 and S_2 of the SSIM index are locally convex and quasi-convex, but the SSIM index itself does not have these nice properties in general. The proposed SSIM metric has the advantage to be also locally convex but on the other hand it is not quasi-convex.

3.4 Projection onto a Convex Set

In some image processing problems, a first estimate $\mathbf{z}$ is further regularized according to the constraint on the data $\mathbf{y}$. The goal is then to find the closest signal to $\mathbf{z}$ that belongs to the constraint set, more often than not a convex set. We provide two examples.

First, in the JPEG compression post-processing algorithm of Li [64], the constraint set represents the range of possible values that can take the original signal given its quantized version. Indeed, the uniform quantization operation can be written as

$$\tilde{c}_k \quad = \quad \left[\frac{c_k}{\Delta}\right], \tag{3.60}$$

where c_k is the coefficient obtained after the Discrete Cosine Transform (DCT) of the original image, Δ is the quantization step and $[x]$ represents the nearest integer of x. So given a quantized coefficient $\tilde{c}_k$, the range of possible values for the original coefficient is

$$(\tilde{c}_k - 1/2)\Delta < c_k \leq (\tilde{c}_k + 1/2)\Delta. \tag{3.61}$$

This inequality is valid for all the coefficients, hence the constraint set is an hyper-cube of 64 dimensions (assuming a DCT over an 8 × 8 block).

Second, for image denoising problems, the additive white Gaussian noise model implies that the equi-probable levels sets are hyper-spheres centered around the noisy image $\mathbf{y}$. The constraint set can be defined as the set of points such that $\|\mathbf{x}-\mathbf{y}\|_2^2 \leq \sigma^2$, where σ^2 is the estimated variance of the noise.

In both case, the optimization problem can be formulated as

$$\text{Maximize SSIM}(\mathbf{z}, \mathbf{x}) \text{ subject to } \mathbf{x} \in A, \tag{3.62}$$

where A is convex. Nothing can be said of the solution in general for the full SSIM index or even for the modified SSIM metric, since they are not known to be either convex or quasi-convex. However, the local SSIM was proved to be quasi-convex on $\langle \mathbf{x}, \mathbf{z}\rangle \geq -\epsilon_2/2$ for zero-mean signals. We can thus exploit Theorem 3.3.5 to prove the existence and uniqueness of the solution.

In fact, an analytic solution can be found when A is a hyper-sphere. Indeed, let

$$A := \{\mathbf{x} \in \mathbf{R}^N : \|\mathbf{x}-\mathbf{y}\|_2^2 \quad \leq \quad \sigma^2\}, \tag{3.63}$$

be the constraint set and consider the level sets of the SSIM index (see (3.30)):

$$\{x \in \mathbf{R}^N : \|\mathbf{x}-\mu\mathbf{z}\|_2^2 = \|\mathbf{z}\|_2^2(\mu^2-1) + \epsilon_2(\mu-1)\}, \tag{3.64}$$

where $\epsilon_2 \geq 0$ is a stability constant and $\mu \geq 1$ is a constant to be determined. If $\mathbf{z} \in A$, then the optimal solution is trivially $\mathbf{x} = \mathbf{z}$. Otherwise, the two hyper-spheres will be tangent at

$$\mathbf{x}_\lambda \quad = \quad (1-\lambda)\mathbf{y} + \lambda\mu\mathbf{z}, \tag{3.65}$$

with $0 \leq \lambda \leq 1$. Plugging in $\mathbf{x}_\lambda$ in the constraint set (3.63) and solving for λ leads to

$$\lambda \quad = \quad \frac{\sigma}{\|\mathbf{y}-\mu\mathbf{z}\|_2}. \tag{3.66}$$

Substituting λ and $\mathbf{x}_\lambda$ in the SSIM level set formula (3.64) simplifies to

$$(\|\mathbf{y}-\mu\mathbf{z}\| - \sigma)^2 \quad = \quad \|\mathbf{z}\|^2(\mu^2-1) + \epsilon_2(\mu-1). \tag{3.67}$$

After squaring the left hand side and isolating the term $\|\mathbf{y} - \mu\mathbf{z}\|$, we obtain

$$\begin{aligned} \|\mathbf{y} - \mu\mathbf{z}\| &= \frac{\|\mathbf{y}\|_2^2 + \|\mathbf{z}\|_2^2 + \epsilon_2 + \sigma^2}{2\sigma} - \frac{2\langle \mathbf{y}, \mathbf{z}\rangle + \epsilon_2}{2\sigma}\mu && (3.68) \\ &=: a - b\mu. && (3.69) \end{aligned}$$

Squaring once again the left hand side simplifies to quadratic function of μ:

$$\mu^2(b^2 - \|\mathbf{z}\|_2^2) + 2\mu(\langle \mathbf{y}, \mathbf{z}\rangle - ab) + (a^2 - \|\mathbf{y}\|^2) = 0. \tag{3.70}$$

The solution for μ is thus

$$\mu_\pm = \frac{ab - \langle \mathbf{y}, \mathbf{z}\rangle \pm \sqrt{(\langle \mathbf{y}, \mathbf{z}\rangle - ab)^2 - (b^2 - \|\mathbf{z}\|_2^2)(a^2 - \|\mathbf{y}\|^2)(a^2 - \|\mathbf{y}\|^2)}}{b^2 - \|\mathbf{z}\|_2^2}. \tag{3.71}$$

The first level set to intersect the constraint is with a smaller μ, thus the negative branch corresponds to the global maximum of SSIM while the positive branch corresponds to the global minimum of SSIM. In this second case, the sign of λ has to be taken to be negative.

The maximal point is then

$$x_{\max} = \mathbf{y} + \sigma\frac{\mu_-\mathbf{z} - \mathbf{y}}{\|\mu_-\mathbf{z} - \mathbf{y}\|_2}, \tag{3.72}$$

that is, the point obtained by moving by σ unit away from $\mathbf{y}$ in the direction opposed to $\mathbf{y} - \mu_-\mathbf{z}$. The minimal point can be found as well according to

$$x_{\min} = \mathbf{y} - \sigma\frac{\mu_+\mathbf{z} - \mathbf{y}}{\|\mu_+\mathbf{z} - \mathbf{y}\|_2}. \tag{3.73}$$

If we seek to find the maximal SSIM on the boundary of a L^2-ball, then we proceed exactly in the same way, but we allow λ to be greater than one when $\mathbf{z}$ is inside the L^2-ball. In the particular case that $\mathbf{z} = \mathbf{y}$, the maximal SSIM will be at

$$\mathbf{x}_{\max} = \mathbf{y} + \sigma\frac{\mathbf{y}}{\|\mathbf{y}\|_2}, \tag{3.74}$$

whereas the minimal SSIM is in the opposed direction:

$$\mathbf{x}_{\min} = \mathbf{y} - \sigma\frac{\mathbf{y}}{\|\mathbf{y}\|_2}. \tag{3.75}$$

This result is equivalent to the matched-mean solution of the SSIM-ball problem described in [49].

Note that the minimal solution does not fit well with the solution dictated by the human perception, since $\mathbf{x}_{\min}$ and $\mathbf{y}$ have a perfect correlation. To alleviate this problem, a modified version of the SSIM index with a different exponent weight β between the correlation term and the contrast term could be studied. Alternatively, one could study pq-normalized metric with different exponents p and q.

Chapter 4

Geodesics of the SSIM Index

Geodesics are the "straight lines" in non-Euclidean geometries. There are several reasons why we are interested to describe the geodesics according to the Structural Similarity (SSIM) index. First, the arc length along the geodesics can lead naturally to a metric. This is the approach proposed by Richter in [88] were he posed the problem of finding the SSIM-geodesic. Second, geodesics can describe a perceptually smooth evolution between two images. Finally, the mid-point of the geodesic can be seen as an average between two images in the geometry described by the SSIM index.

In the first section of this chapter, we will define and review the basic properties of geodesics. In particular, we will state the necessary theorems from calculus of variation. Contrary to Richter [88], our approach will not need all the apparatus of differential geometry since we will be already working in $\mathbf{R}^N$. In the second section, we will study the geodesics of normalized metrics. After outlining a general method, we will explicitly solve for a few particular cases. In the last section, the geodesics of the SSIM index will be described with examples and some extensions will be considered.

4.1 Geodesics: Preliminaries

4.1.1 Definitions

A geodesic is the shortest path between two points according to a certain metric. Let $\Gamma_{x,y}$ be the set of all rectifiable curves, i.e. of finite length, between x and y. Let (X, d) be a metric space and let $\gamma : [0,1] \to X$ be the parameterization of a curve in $\Gamma_{x,y}$. Then the *length* of γ according to the metric d is defined as

$$L(\gamma) \quad := \quad \sup_{(t_0,t_1,\ldots,t_n)} \left\{ \sum_{i=1}^{n} d(\gamma(t_i), \gamma(t_{i-1})) : n \in \mathbf{N}, 0 = t_0 < t_1 < \ldots < t_n = 1 \right\}. \tag{4.1}$$

If γ is piecewise differentiable and d is induced by the norm $\| \cdot \|$, then the length of γ is

$$L(\gamma) \quad = \quad \int_0^1 \|\gamma'(t)\| dt. \tag{4.2}$$

The *geodesic path* is the curve of minimal length between two points:

$$\arg \inf_{\gamma \in \Gamma_{x,y}} L(\gamma). \tag{4.3}$$

From the geodesic, one can define a metric called *intrinsic metric.*

Theorem 4.1.1. *Let γ be a piecewise differentiable path between x and y. Then*

$$d(x,y) = \inf_{\gamma \in \Gamma_{x,y}} \int_0^1 \|\gamma'(t)\| dt \qquad \textit{is a metric.} \tag{4.4}$$

Proof. For positive definiteness, one can easily check that $d(x,x) = 0$. Now, if $d(x,y) = 0$, it implies that $\|\gamma'(t)\| = 0$ almost everywhere: hence $\gamma(t)$ is constant almost everywhere. Since γ is continuous, we have $\gamma(t) = \gamma(0) = \gamma(1)$ for all $t \in [0,1]$. That is, $x = y$.

The symmetry property is clear since the length of a curve does not depend on its orientation.

For the triangular inequality, consider three points x, y and z. Since the combination of the path between x and y with the path between y and z is also a path between x and z, then by the definition of an intrinsic metric $d(x,z) \leq d(x,y) + d(y,z)$. □

4.1.2 The Euler-Lagrange Equation

To find the geodesic between two points, one generally relies on the calculus of variation, particularly the Euler-Lagrange equation. The solution of this equation may be viewed as a generalization of finding critical points for functionals.

Given the problem

$$\arg\inf_{f(t)} \int F(t, f(t), f'(t))dt, \tag{4.5}$$

a necessary condition for $f(t)$ to minimize $\int F$ is

$$\frac{\partial}{\partial f}F - \frac{d}{dt}\left(\frac{\partial}{\partial f'}F\right) \quad = \quad 0. \tag{4.6}$$

This is the Euler-Lagrange equation. Keeping everything else constant, the partial derivative according to f or f' behaves exactly the same as if f and f' were two independent variables.

The computation of the Euler-Lagrange equation (4.6) leads to a second order ordinary differential equation (ODE). Note that from the context it will be often evident when the necessary condition is also sufficient. Note also that some authors call geodesics all the solutions of the Euler-Lagrange equation, which would be analogous to the set of critical points of a function. However we will stick with the definition of a geodesic as the minimal path between two points.

A special case of the Euler-Lagrange equation is when t is absent. In this case, the necessary condition (4.6) becomes

$$F - f'\left(\frac{\partial}{\partial f'}F\right) \quad = \quad c. \tag{4.7}$$

Here is a short proof taken from [46]. We denote $F_f = \frac{\partial}{\partial f}F$ and $F_{f'} = \frac{\partial}{\partial f'}F$. By the chain rule, the Euler-Lagrange (4.6) becomes

$$F_f - \frac{d}{dt}F_{f'} \quad = \quad F_f - f'F_{f'f} - f''F_{f'f'} = 0. \tag{4.8}$$

Multiply by f':

$$f'F_f - (f')^2F_{f'f} - f'f''F_{f'f'} \quad = \quad \frac{d}{dt}(F - f'F_{f'}) = 0. \tag{4.9}$$

Hence,

$$F - f'F_{f'} = c. \tag{4.10}$$

Example 4.1.2. *The arc length of a graph* $(t, f(t))$ *in a Euclidean space is*

$$\int_a^b \sqrt{1+(f'(t))^2}dt. \tag{4.11}$$

By the Euler-Lagrange equation (4.7),

$$\sqrt{1+(f'(t))^2} - \frac{(f')^2}{\sqrt{1+(f'(t))^2}} \quad = \quad c. \tag{4.12}$$

It simplifies to

$$f'(t) \quad = \quad \pm\sqrt{1/c^2-1} =: A. \tag{4.13}$$

Thus $f(t) = At + B$*. The constants* A *and* B *can be found from the initial conditions* $(a, f(a))$ *and* $(b, f(b))$*. The final solution is*

$$f(t) = f(b)\frac{t-a}{b-a} + f(a)\frac{b-t}{b-a}. \tag{4.14}$$

4.2 Geodesics of Normalized Metrics

In this section, we will describe the geodesic paths with constant speed associated to different normalized metrics in $\mathbf{R}^N$ and draw their geodesics. We will start from some general considerations before we study several particular cases.

4.2.1 General Case

We will outline how to solve, in general, the problem of finding the geodesic associated with a given normalized distance

$$\rho_M(x,y) = \frac{\|\mathbf{x}-\mathbf{y}\|_2}{M(\|\mathbf{x}\|_2, \|\mathbf{y}\|_2)}. \tag{4.15}$$

The length of the curve parameterized by $\gamma : [0,1] \to \mathbf{R}^N$ can be written as

$$L(\gamma) \quad = \quad \int_0^1 \frac{\|\gamma'(t)\|_2}{M(\|\gamma(t)\|_2, \|\gamma(t)\|_2)}dt. \tag{4.16}$$

We will use methods from calculus of variations, and in particular, the Euler-Lagrange equation to solve for the minimal path.

Note that for the application we have in mind, i.e. to find the perceptual average of images, not only do we need to find the geodesic path but it is also important to travel at constant speed. Indeed, the mid-point is the location such that $\gamma(1/2)$. If we do not have an arc length parameterization of the geodesic path, then there is no obvious way to retrieve this mid-point. Thus, we have to always make sure that

$$\frac{\|\gamma'(t)\|_2}{M(\|\gamma(t)\|_2, \|\gamma(t)\|_2)} = S, \tag{4.17}$$

where S is a constant. We call this the *constant speed condition.*

The first step to find the geodesic is to take advantage of symmetries and reduce the problem to only two dimensions. Writing the parametric curve $\gamma_N(t)$ in $\mathbf{R}^N$ in hyperspherical coordinates, where $\rho(t) \geq 0$, $0 \leq \phi_i(t) \leq \pi$ for $1 \leq i \leq N-2$ and $0 \leq \theta(t) \leq 2\pi$ are the radius, polar angles and azimuthal angle, respectively, the arc length becomes

$$L(\gamma_N) \;=\; \int_0^1 \frac{\sqrt{(\rho'(t))^2 + (\rho(t))^2 \sum_{i=1}^{N-2} (\phi_i'(t))^2 + (\rho(t))^2(\theta'(t))^2}}{M(\rho(t), \rho(t))} dt. \tag{4.18}$$

By rotational invariance of normalized distances (Proposition 3.1.1), we can assume without loss of generality that the endpoints $\mathbf{x}, \mathbf{y}$ are in the azimuthal plane, i.e. that $\phi_i(0) = \phi_i(1) = \pi/2$ for $1 \leq i \leq N-2$.

It is clear that for the arc length (4.18) to be minimized, we need to have $\phi_i'(t) \equiv 0$ for $1 \leq i \leq N-2$, i.e. $\phi_i(t) \equiv \pi/2$ for $1 \leq i \leq N-2$. Thus the problem reduces to two dimensions and the solution can be expressed in polar coordinates:

$$\gamma_2(t) \;:=\; \big(\rho(t)\cos(\theta(t)), \rho(t)\sin(\theta(t))\big), \tag{4.19}$$

where $\gamma_2(t)$ is on the plane generated by $\mathbf{0}$, $\mathbf{x}$ and $\mathbf{y}$ and where $\rho(t) = \|\gamma_2(t)\|$ and $\theta(t)$ are the angles between $\gamma_2(t)$ and $\mathbf{x}$.

The arc length is then

$$L(\gamma_2) \;=\; \int_0^1 \frac{\sqrt{(\rho'(t))^2 + (\rho(t))^2(\theta'(t))^2}}{M(\rho(t), \rho(t))} dt \tag{4.20}$$

and the initial conditions are

$$\rho(0) = \|\mathbf{x}\|_2, \quad \rho(1) = \|\mathbf{y}\|_2, \quad \theta(0) = \alpha, \quad \text{and} \quad \theta(1) = \beta. \tag{4.21}$$

If $\alpha = \beta$, then the geodesic path will lie on the straight line relying $\mathbf{x}$ and $\mathbf{y}$. Indeed, since $(\theta'(t))^2 \geq 0$ and $\theta(0) = \theta(1)$, it is clear that the parameterization of the path minimizing (4.20) will satisfy $\theta'(t) \equiv 0$. The formula for the arc length then simplifies to

$$L(\gamma_1) = \int_0^1 \frac{\|\rho'(t)\|_2}{M(\rho(t), \rho(t))} dt, \tag{4.22}$$

where $\gamma_1(t) := \rho(t)$.

To enforce the constant speed condition (4.17), we need

$$\frac{|\rho'(t)|}{M(\rho(t), \rho(t))} = S. \tag{4.23}$$

Since we are in a one-dimensional subspace, it is clear that $\rho(t)$ is monotonic. We can thus drop the absolute value, the sign of $\rho'(t)$ being absorbed in the constant. The solution to the ODE (4.23) is

$$f(\rho(t)) := \int \frac{1}{M(\rho, \rho)} d\rho = St + R. \tag{4.24}$$

From the initial conditions (4.21), we have

$$\theta(t) = \alpha = \beta, \text{ and } f(\rho(t)) = f(\|\mathbf{x}\|_2)(1-t) + f(\|\mathbf{y}\|_2)t. \tag{4.25}$$

If $\mathbf{0}$, $\mathbf{x}$ and $\mathbf{y}$ are co-linear and are opposite to each other, then there are two geodesics in $\mathbf{R}^2$ and an infinite number of geodesics in $\mathbf{R}^N$ with $N > 2$, i.e. there are two geodesics for each plane passing by $\mathbf{0}$, $\mathbf{x}$ and $\mathbf{y}$. By continuity of the arc length formula, for every fixed half-plane, we can approximate the geodesic path with another path such that $|\alpha - \beta| < \pi$.

It remains to solve the geodesic problem for $0 < |\beta - \alpha| < \pi$. Since $(\theta'(t))^2 \geq 0$, it is clear that the angle of the geodesic path is monotonic. Without loss of generality, we assume that $\theta(t)$ is strictly monotonic (either increasing or decreasing). If it is not the case, and since $L(\gamma)$ is continuous, then there will be another path with a strictly monotonic angle that will have approximately the same length.

For strictly monotonic $\theta(t)$, we can express the radius as a function of the polar angle. After the change of variables $\theta = \theta(t)$ and $r(\theta) = \rho(t)$, we express the arc length in terms of a single variable,

$$L(\gamma_2) \;=\; \int_\alpha^\beta \frac{\sqrt{(r'(\theta))^2+(r(\theta))^2}}{M(r(\theta),r(\theta))}d\theta, \tag{4.26}$$

with $r(\alpha)=\rho(0)$ and $r(\beta)=\rho(1)$.

We denote $r'(t)$ by $\dot{r}$. By the special case of the Euler-Lagrange equation (4.7), the stationary points satisfy

$$F-\dot{r}F_{\dot{r}} \;=\; c_1, \tag{4.27}$$

where

$$F \;=\; \frac{\sqrt{\dot{r}^2+r^2}}{M(r,r)}. \tag{4.28}$$

That is,

$$\frac{\sqrt{\dot{r}^2+r^2}}{M(r,r)}-\frac{\dot{r}^2}{\sqrt{\dot{r}^2+r^2}}\frac{1}{M(r,r)} \;=\; c_1. \tag{4.29}$$

Necessarily $c_1\neq 0$, since otherwise $r\equiv 0$. Rearranging (4.29) leads to

$$|\dot{r}| = r\sqrt{\frac{r^2}{c_1^2M^2(r,r)}-1}. \tag{4.30}$$

We will solve this ODE for $M(r,r)=r$, for $M(r,r)=r^q$ with $0<q<1$ and for $M(r,r)=\sqrt{r^2+\epsilon^2}$.

4.2.2 Case $M(r,r)=r$

This case was first solved by Martin and Osgood [70] in 1986. Here, we report our own derivation.

Several normalized metrics satisfy $M(r,r) = r$:

$$M(|x|,|y|) = \left(\frac{|x|^p + |y|^p}{2}\right)^{1/p}; \tag{4.31}$$

$$M(|x|,|y|) = \max(|x|,|y|). \tag{4.32}$$

In particular, the normalized distance

$$d(\mathbf{x},\mathbf{y}) = \frac{\|\mathbf{x}-\mathbf{y}\|_2}{\sqrt{\frac{\|\mathbf{x}\|_2^2+\|\mathbf{y}\|_2^2}{2}}} \tag{4.33}$$

is included in this category. This is equivalent to $\sqrt{\frac{1-SSIM}{2}}$ with zero-mean signals and zero stability constant.

If $\alpha = \beta$, then from (4.24) we integrate

$$f(r) := \int \frac{1}{r} dr = \log(r) \tag{4.34}$$

to find the solution with constant speed. Thus, from (4.25) the solution is

$$\log(r) = \log(\|\mathbf{x}\|)(1-t) + \log(\|\mathbf{y}\|)t \tag{4.35}$$

or

$$r(t) = \|\mathbf{x}\|^{1-t}\|\mathbf{y}\|^t. \tag{4.36}$$

If $\alpha \neq \beta$, we need to find the function $r(\theta)$ that minimizes

$$\int_\alpha^\beta \sqrt{\left(\frac{\dot{r}}{r}\right)^2 + 1}\ d\theta. \tag{4.37}$$

From the Euler-Lagrange equation (4.30), the parameterization of the geodesic satisfies

$$\pm\log(r) = \sqrt{1/c_1^2 - 1\theta} + c_2 =: A\theta + B, \tag{4.38}$$

where c_2, A and B are some constants to be determined. Notice that $|\dot{r}| > 0$ always, so the sign of $\log(r)$ does not change and it is absorbed in the constants A and B.

From the initial conditions (4.21), we find the values of A and B:

$$\begin{aligned} \log(\|\mathbf{x}\|) &= A\alpha + B; && (4.39)\\ \log(\|\mathbf{y}\|) &= A\beta + B. && (4.40) \end{aligned}$$

So

$$\begin{aligned} A &= \frac{\log(\|\mathbf{y}\|) - \log(\|\mathbf{x}\|)}{\beta - \alpha}, && (4.41)\\ B &= \log(\|\mathbf{x}\|) - \frac{\log(\|\mathbf{y}\|) - \log(\|\mathbf{x}\|)}{\beta - \alpha}\alpha && (4.42) \end{aligned}$$

and

$$r(\theta) = \|\mathbf{x}\|^{\frac{\beta-\theta}{\beta-\alpha}} \|\mathbf{y}\|^{\frac{\theta-\alpha}{\beta-\alpha}}. \qquad (4.43)$$

The minimal path is thus a logarithmic spiral passing by $\mathbf{x}$ and $\mathbf{y}$ in the plane generated by $\mathbf{x}$ and $\mathbf{y}$.

For constant speed, we choose $\theta = \theta(t)$ such that

$$\sqrt{\left(\frac{r'(\theta)}{r(\theta)}\right)^2 + (\theta')^2} = S. \qquad (4.44)$$

Taking the derivative of r with respect to t in (4.43) yields to

$$\frac{d}{dt} r(\theta(t)) = \frac{\theta'}{\beta - \alpha} \log(\|\mathbf{y}\|/\|\mathbf{x}\|) \|\mathbf{x}\|^{\frac{\beta-\theta}{\beta-\alpha}} \|\mathbf{y}\|^{\frac{\theta-\alpha}{\beta-\alpha}}. \qquad (4.45)$$

Substituting (4.45) in (4.44) leads to

$$\theta' \sqrt{\left(\frac{\log(\|\mathbf{y}\|/\|\mathbf{x}\|)}{\beta - \alpha}\right)^2 + 1} = S. \qquad (4.46)$$

This means that $\theta(t)$ is linear. After plugging the initial conditions (4.21), we finally obtain the geodesic:

$$\gamma(t) = \|\mathbf{x}\|^{1-t} \|\mathbf{y}\|^{t} \big(\cos(\alpha + t(\beta - \alpha)), \sin(\alpha + t(\beta - \alpha))\big). \qquad (4.47)$$

Alternatively, we can write the minimal path as[1]

$$\gamma(t) \;=\; \|\mathbf{x}\|^{1-t}\|\mathbf{y}\|^{t}\big(\mathbf{x}/\|\mathbf{x}\|\cos(\omega t) + \mathbf{u}\sin(\omega t)\big), \tag{4.48}$$

where ω is the angle between $\mathbf{x}$ and $\mathbf{y}$ and where $\mathbf{u}$ is a unit vector in the plane generated by $\mathbf{x}$ and $\mathbf{y}$ that is perpendicular to $\mathbf{x}$ and in the same half-plane than $\mathbf{y}$, i.e. $\omega = \arccos\frac{\langle \mathbf{x},\mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|}$, $\langle \mathbf{x}, \mathbf{u}\rangle = 0$, $\|\mathbf{u}\| = 1$ and $\langle \mathbf{u}, \mathbf{y}\rangle \geq 0$. This last vector can be found by Gram-Schmidt orthonormalization procedure:

$$\begin{aligned}\mathbf{u} \;&=\; \frac{\mathbf{y} - \langle \mathbf{y}, \mathbf{x}/\|\mathbf{x}\|\rangle \mathbf{x}/\|\mathbf{x}\|}{\big\|\mathbf{y} - \langle \mathbf{y}, \mathbf{x}/\|\mathbf{x}\|\rangle \mathbf{x}/\|\mathbf{x}\|\big\|} && (4.49)\\ &=\; \frac{1}{\sin(\omega)}\frac{\mathbf{y}}{\|\mathbf{y}\|} - \frac{\cos(\omega)}{\sin(\omega)}\frac{\mathbf{x}}{\|\mathbf{x}\|}. && (4.50)\end{aligned}$$

So

$$\gamma(t) \;=\; \|\mathbf{x}\|^{1-t}\|\mathbf{y}\|^{t}\left(\frac{\sin(\omega(1-t))}{\sin(\omega)}\frac{\mathbf{x}}{\|\mathbf{x}\|} + \frac{\sin(\omega t)}{\sin(\omega)}\frac{\mathbf{y}}{\|\mathbf{y}\|}\right). \tag{4.51}$$

Computing the integral leads to the arc length formula:

$$L(\gamma) = \sqrt{(\log\|\mathbf{y}\| - \log\|\mathbf{x}\|)^2 + (\beta - \alpha)^2}, \tag{4.52}$$

which is called the quasi-hyperbolic metric [107].

Example 4.2.1. *We compare the last result to the length of the straight line path γ_l according to the normalized metric, which is*

$$L(\gamma_l) = \begin{cases} \left|\log\left(\frac{\|\mathbf{y}\|^2 - \langle \mathbf{x},\mathbf{y}\rangle + \|\mathbf{y}\|_2\|\mathbf{y}-\mathbf{x}\|_2}{\langle \mathbf{x},\mathbf{y}\rangle - \|\mathbf{x}\|_2^2 + \|\mathbf{x}\|_2\|\mathbf{y}-\mathbf{x}\|_2}\right)\right|, & \textit{if } \mathbf{y} \neq \lambda\mathbf{x},\\ \big|\log\|\mathbf{y}\|_2 - \log\|\mathbf{x}\|_2\big|, & \textit{if } \mathbf{y} = \lambda\mathbf{x} \textit{ with } \lambda > 0,\\ \infty & \textit{for } \mathbf{y} = -\lambda\mathbf{x} \textit{ with } \lambda \geq 0. \end{cases} \tag{4.53}$$

[1] This idea was suggested by József Vass in [10].

The results are summarized in the following table.

$\mathbf{x}$	$\mathbf{y}$	$\gamma(t)$	$\gamma_l(t)$	$L(\gamma)$	$L(\gamma_l)$
$(1,0)$	$(4,0)$	$(4^t,0)$	$(1+3t,0)$	$\log(4)$	$\log(4)$
$(1,0)$	$(0,1)$	$(\cos(\pi t/2),\sin(\pi t/2))$	$(1-t,t)$	$\pi/2$	$\log\left(\frac{1+\sqrt{2}}{-1+\sqrt{2}}\right)$
$(1,0)$	$(0,4)$	$(4^t\cos(\pi t/2),4^t\sin(\pi t/2))$	$(1-t,4t)$	$\sqrt{\log^2(4)+(\pi/2)^2}$	$\log\left(\frac{16+4\sqrt{17}}{-1+\sqrt{17}}\right)$
$(3,0)$	$(-1,0)$	$(3^{1-t}\cos(\pi t),\pm 3^{1-t}\sin(\pi t))$	$(2-3t,0)$	$\sqrt{\log^2(2)+\pi^2}$	∞

We can verify that, as expected, $L(\gamma)\leq L(\gamma_l)$ for each of the four cases.

4.2.3 Case $M(r,r)=r^q$ with $0<q<1$

The case $M(r,r)=r^q$ with $0<q<1$ occurs when a metric is normalized by a power of the generalized mean:

$$M(|x|,|y|) \;=\; \left(\frac{|x|^p+|y|^p}{2}\right)^{q/p} \text{ with } 0<q<1. \tag{4.54}$$

The geodesic path for this normalized metric was solved by Hästö [51] in 2005. Again, we present our own derivation for the geodesic with constant speed.

If $\alpha=\beta$, then from (4.24)

$$f(r) := \int \frac{1}{r^q}dr = \frac{r^{1-q}}{1-q} \tag{4.55}$$

and from (4.25) the constant speed path is

$$r(t) \;=\; (t\|\mathbf{y}\|^{1-q}+(1-t)\|\mathbf{x}\|^{1-q})^{\frac{1}{1-q}}. \tag{4.56}$$

In the limiting cases $q\to 0$ and $q\to 1$, we find

$$r(t)=t\|\mathbf{y}\|+(1-t)\|\mathbf{x}\| \text{ and } r(t)=\|\mathbf{y}\|^t\|\mathbf{x}\|^{1-t}, \text{ respectively.}$$

If $\alpha \neq \beta$, then from (4.30) the geodesic satisfies

$$|\dot{r}| = r\sqrt{\frac{r^{2(1-q)}}{c_1^2} - 1}. \tag{4.57}$$

If $\dot{r}(\theta) = 0$ for some θ, then $r(\theta) = c_1^{\frac{1}{1-q}}$. Hence the function $r(\theta)$ has at most one critical point. It cannot be local maximum as it is clear that the shortest path must satisfy $r(\theta) \leq \max(\|\mathbf{x}\|, \|\mathbf{y}\|)$. Otherwise by taking $\tilde{r}(\theta) = \max(\|\mathbf{x}\|, \|\mathbf{y}\|)$ for that particular portion of the path where $r(\theta) > \max(\|\mathbf{x}\|, \|\mathbf{y}\|)$ the length of the path will be automatically reduced.

If $\dot{r} \neq 0$, by separation of variables,

$$\pm \int \frac{c_1}{r^{2-q}\sqrt{1 - c_1^2 r^{-2(1-q)}}} dr \quad = \quad \int d\theta. \tag{4.58}$$

Performing the change of variables $u = \frac{c_1}{r^{1-q}}$ leads to

$$\pm \int \frac{du}{(1-q)\sqrt{1-u^2}} \quad = \quad \int d\theta \tag{4.59}$$

$$f(r) := \frac{\arcsin(c_1/r^{1-q})}{1-q} \quad = \quad \pm(\theta + c_{\pm}), \tag{4.60}$$

where c_+ and c_- are the constants for each branch of $f(r)$. Isolating r, we obtain

$$r(\theta) \quad = \quad \left(\frac{\pm c_1}{\sin((1-q)(\theta + c_{\pm}))}\right)^{\frac{1}{1-q}}. \tag{4.61}$$

Let (r^*, θ^*) be the polar coordinate at the minimal value of $r(\theta)$. To find a relationship between c_+ and c_-, we study $r(\theta)$ at its minimum:

$$r^* = c_1^{\frac{1}{1-q}} \quad = \quad \left(\frac{\pm c_1}{\sin((1-q)(\theta^* + c_{\pm}))}\right)^{\frac{1}{1-q}}, \tag{4.62}$$

hence,

$$\sin((1-q)(\theta^* + c_{\pm})) \quad = \quad \pm 1. \tag{4.63}$$

The periodic solution is thus

$$\begin{aligned}(1-q)(\theta^*+c_\pm) &= \pm\frac{\pi}{2}+k_\pm\pi, \quad k_\pm \in \mathbf{Z}, \text{ or} && (4.64)\\ c_2 := c_+ &= c_- + \frac{k\pi}{1-q}, \quad k_\pm \in \mathbf{Z}. && (4.65)\end{aligned}$$

Since a shift of $k\pi$ in the sinus function only changes its sign, we have

$$r(\theta) = \left(\frac{c_1}{\sin((1-q)(\theta+c_2))}\right)^{\frac{1}{1-q}}. \qquad (4.66)$$

Solving for c_1 and c_2 using the initial conditions:

$$\begin{aligned}\|\mathbf{x}\| &= \left(\frac{c_1}{\sin((1-q)(\alpha+c_2))}\right)^{\frac{1}{1-q}} \text{ and} && (4.67)\\ \|\mathbf{y}\| &= \left(\frac{c_1}{\sin((1-q)(\beta+c_2))}\right)^{\frac{1}{1-q}}, && (4.68)\end{aligned}$$

we find

$$\left(\frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}\right)^{1-q} = \frac{\sin((1-q)\alpha)\cos((1-q)c_2)+\cos((1-q)\alpha)\sin((1-q)c_2)}{\sin((1-q)\beta)\cos((1-q)c_2)+\cos((1-q)\beta)\sin((1-q)c_2)}. \qquad (4.69)$$

It thus leads to

$$\begin{aligned}c_1 &= \|\mathbf{x}\|^{1-q}\sin((1-q)(\alpha+c_2)) \text{ and} && (4.70)\\ c_2 &= \frac{1}{1-q}\arctan\left(\frac{\sin((1-q)\beta)\left(\frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}\right)^{1-q} - \sin((1-q)\alpha)}{\cos((1-q)\alpha) - \cos((1-q)\beta)\left(\frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}\right)^{1-q}}\right). && (4.71)\end{aligned}$$

To enforce constant speed, we need

$$\frac{\sqrt{\dot{r}^2+r^2\dot{\theta}^2}}{r^q} = S. \qquad (4.72)$$

Taking the derivative of r with respect to t,

$$\frac{d}{dt}r(\theta(t)) = (r(\theta))^{2-q}\frac{\cos((1-q)(\theta+c_\pm))}{\pm c_1}\dot{\theta}. \tag{4.73}$$

Substituting $r(\theta)$ and $r'(\theta)$ in (4.72), we obtain after simplifications

$$\dot{\theta}\frac{\pm c_1}{\sin^2((1-q)(\theta+c_\pm))} = S. \tag{4.74}$$

We solve this ODE by integrating both sides:

$$\int \frac{\pm c_1}{\sin^2((1-q)(\theta+c_\pm))}d\theta = S\int dt \tag{4.75}$$

$$\frac{\cot((1-q)(\theta+c_\pm))}{1-q} = \pm(St+R). \tag{4.76}$$

Isolating θ and plugging in the initial conditions (4.21), we find the parameterization of the angle for constant speed:

$$\theta(t) = \pm\frac{\operatorname{arccot}(t\cot((1-q)(\beta+c_2)) + (1-t)\cot((1-q)(\alpha+c_2)))}{1-q} - c_2 - \frac{k\pi}{1-q}, \tag{4.77}$$

where $k \in \mathbf{Z}$. We choose k in order to ensure that

$$\min(\alpha,\beta) \leq \theta(t) \leq \max(\alpha,\beta) \tag{4.78}$$

and take the positive or negative sign of arccot depending on whether $\alpha < \beta$ or $\alpha > \beta$.

Combining this last formula with the equation (4.66) for $r(\theta)$ and with the constants c_1 and c_2 given in (4.70) and (4.71), it leads to the complete parameterization of the geodesic.

In Fig. 4.1, we draw the geodesics and find the midpoints between $\mathbf{x} = (1,0)$ and $\mathbf{y} = (0,4)$ for different values of the parameter q.

4.2.4 Case $M(r,r) = \sqrt{r^2+\epsilon^2}$

This is the most interesting case since it is related to the most common form of the SSIM index. Indeed, this normalized metric is equivalent to $\sqrt{\frac{1-\text{SSIM}}{2}}$ with zero-mean signals

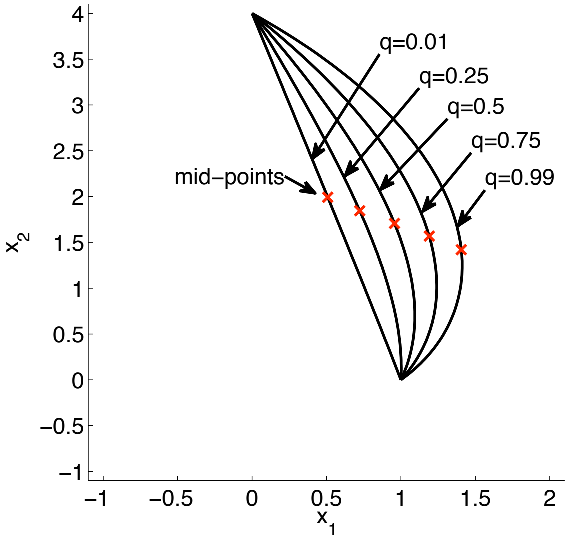


Figure 4.1: Geodesics between $(1, 0)$ and $(0, 4)$ for a metric normalized by $M(r, r) = r^q$ with $q = \frac{1}{100}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and $\frac{99}{100}$.

and arbitrary stability constant. Parts of the results describing this new case were published in Applied Mathematics Letters [10].

If $\alpha = \beta$, then proceeding as before we obtain

$$f(r) = \int \frac{1}{\sqrt{r^2 + \epsilon^2}} dr = \log(2\sqrt{r^2 + \epsilon^2} + 2r) \tag{4.79}$$

and

$$\begin{aligned} \sqrt{r^2 + \epsilon^2} + r &= (\sqrt{\|\mathbf{y}\|^2 + \epsilon^2} + \|\mathbf{y}\|)^t (\sqrt{\|\mathbf{x}\|^2 + \epsilon^2} + \|\mathbf{x}\|)^{1-t} & (4.80) \\ &=: (g(\|\mathbf{y}\|))^t (g(\|\mathbf{x}\|))^{1-t}. & (4.81) \end{aligned}$$

Isolating r, we find

$$r(t) = \frac{(g(\|\mathbf{y}\|))^t (g(\|\mathbf{x}\|))^{1-t}}{2} - \frac{\epsilon^2}{2(g(\|\mathbf{y}\|))^t (g(\|\mathbf{x}\|))^{1-t}}. \tag{4.82}$$

When $\epsilon \to 0$, we are back to the geometric mean.

Let us now solve the case $\alpha \neq \beta$. We denote

$$m(r) \quad := \quad \frac{r}{M(r,r)} = \frac{r}{\sqrt{r^2+\epsilon^2}}. \tag{4.83}$$

Note that for $r > 0$ and $\epsilon > 0$, we have $0 < m(r) < 1$. If $\dot{r} = 0$, we find from (4.30) that $m(r) = c_1$, hence $r = \frac{c_1 \epsilon}{\sqrt{1-c_1^2}}$. Again, from the same argument that for the case $M(r,r) = r^q$ it is clear that the polar parameterization of the geodesic has at most one critical point and that it cannot be a maximum.

If $\dot{r} \neq 0$, then $m(r) > c_1$. We need to integrate

$$f(r,c_1) := \int \frac{c_1}{r\sqrt{m^2(r) - c_1^2}} dr. \tag{4.84}$$

Performing the change of variables $u = \frac{c_1\sqrt{r^2+\epsilon^2}}{r}$, $du = \frac{-c_1\epsilon^2}{r^2\sqrt{r^2+\epsilon^2}} dr$,

$$\begin{aligned} f(r,c_1) \quad &= \quad \int \frac{-(r^2+\epsilon^2)}{\epsilon^2\sqrt{1-u^2}} du && (4.85) \\ &= \quad \int \frac{-c_1^2}{(u^2-c_1^2)\sqrt{1-u^2}} du - \int \frac{1}{\sqrt{1-u^2}} du. && (4.86) \end{aligned}$$

Now, setting $v = \frac{\sqrt{1-c_1^2}}{c_1} \frac{u}{\sqrt{1-u^2}}$ and $dv = \frac{\sqrt{1-c_1^2}}{c_1} \frac{1}{(1-u^2)^{3/2}} du$ in the first integral, the change of variable leads to

$$f(r,c_1) = \frac{c_1}{\sqrt{1-c_1^2}} \int \frac{1}{1-v^2} dv - \int \frac{1}{\sqrt{1-u^2}} du. \tag{4.87}$$

Integrating, we obtain

$$f(r,c_1) = \frac{c_1}{\sqrt{1-c_1^2}} \operatorname{artanh}(v) - \arcsin(u), \quad 0 \leq c_1 < m(r), \tag{4.88}$$

where

$$u = \frac{c_1}{m(r)} \qquad \text{and} \qquad v = \frac{\sqrt{1-c_1^2}}{\sqrt{(m(r))^2 - c_1^2}}. \tag{4.89}$$

The parameterization of the minimal path will thus satisfy

$$f(r,c_1) = \pm(\theta + c_\pm), \tag{4.90}$$

where $c_\pm$ are the integration constants for each branch of r.

Note that since $c_1 < m(r) < 1$, hence $v(r) > 1$ and $f(r,c_1)$ has $\frac{c_1}{\sqrt{1-c_1^2}} i\pi/2$ as an imaginary part. Also, for each r fixed,

$$\begin{aligned} f(r,m(r)) := \lim_{c_1 \to m(r)} f(r,c_1) &= -\pi/2 + \frac{m(r)}{\sqrt{1-(m(r))^2}} i\pi/2, & (4.91) \\ f(r,0) := \lim_{c_1 \to 0} f(r,c_1) &= 0. & (4.92) \end{aligned}$$

Since $m(r)$ is increasing in r, while $u(r)$ and $v(r)$ are decreasing in r, we see that the real part of $f(r,c_1)$ is increasing in r. It implies that $r(\theta)$ will be increasing whenever the RHS is increasing and it will be decreasing whenever the RHS is decreasing. Since the right hand side of (4.90) has two branches of opposite sign with one increasing in θ and another one decreasing in θ, $r(\theta)$ will have a unique minimum. Let (r^*, θ^*) be the coordinate for which the radius of the geodesic is minimal.

From the initial conditions, we have

$$\begin{aligned} f(\|\mathbf{x}\|, c1) &= \pm(\alpha + c_+), & (4.93) \\ f(r^*, c1) &= \pm(\theta^* + c_+), & (4.94) \\ f(r^*, c1) &= \mp(\theta^* + c_-), & (4.95) \\ f(\|\mathbf{y}\|, c1) &= \mp(\beta + c_-). & (4.96) \end{aligned}$$

Adding the second and third equation, we obtain

$$2f(r^*, c1) = \pm(c_+ - c_-). \tag{4.97}$$

Isolating c_+ and c_- in the first and fourth equation and substituting them in (4.97) leads to the single non-linear equation

$$|2f(r^*, c1) - f(\|\mathbf{x}\|, c1) - f(\|\mathbf{y}\|, c1)| = |\beta - \alpha|. \tag{4.98}$$

Without loss of generality, assume that $\|\mathbf{x}\| \leq \|\mathbf{y}\|$. From (4.30), either (i) $r^*(\alpha) = \|\mathbf{x}\|$,

(ii) $\dot{r}(\theta^*) = 0$ or (iii) both. Note that (ii) implies that $m(r^*) = c_1$. In case (i), we have

$$|f(\|\mathbf{y}\|, c_1) - f(\|\mathbf{x}\|, c_1)| \quad = \quad |\beta - \alpha| \tag{4.99}$$

whereas in case (ii), we have

$$|f(\|\mathbf{y}\|, m(r^*)| + f(\|\mathbf{x}\|, m(r^*)) - 2f(r^*, m(r^*))| \quad = \quad |\beta - \alpha|. \tag{4.100}$$

Depending on the sign of

$$|f(\|\mathbf{y}\|, m(\|\mathbf{x}\|)) - f(\|\mathbf{x}\|, m(\|\mathbf{x}\|))| - |\beta - \alpha|, \tag{4.101}$$

we are either in case (i), case (ii) or case (iii).

In case (i),

$$0 < f(\|\mathbf{y}\|, c_1) - f(\|\mathbf{x}\|, c_1) = |\beta - \alpha| < |f(\|\mathbf{y}\|, m(\|\mathbf{x}\|)) - f(\|\mathbf{x}\|, m(\|\mathbf{x}\|))|, \tag{4.102}$$

since from (4.92)

$$\lim_{c_1 \to 0} f(\|\mathbf{y}\|, c_1) - f(\|\mathbf{x}\|, c_1) = 0. \tag{4.103}$$

By the intermediate value theorem, there exists a c_1 such that $0 < c_1 < m(\|\mathbf{x}\|)$ for which (4.99) is satisfied.

In case (ii),

$$\pi > |f(\|\mathbf{y}\|, m(r^*)) + f(\|\mathbf{x}\|, m(r^*)) - 2f(r^*, m(r^*))| \quad = \quad |\beta - \alpha| \tag{4.104}$$

and

$$|\beta - \alpha| \quad > \quad |f(\|\mathbf{y}\|, m(\|\mathbf{x}\|)) - f(\|\mathbf{x}\|, m(\|\mathbf{x}\|))|, \tag{4.105}$$

since from (4.91) and (4.92)

$$\lim_{r^* \to 0} f(\|\mathbf{y}\|, m(r^*)) + f(\|\mathbf{x}\|, m(r^*)) = 0 \tag{4.106}$$

and

$$\lim_{r^* \to 0} -2f(r^*, m(r^*)) = \pi. \tag{4.107}$$

By the intermediate value theorem, there exists an r^* such that $0 < r^* < \|\mathbf{x}\|$ for which (4.100) is satisfied.

For case (iii), we simply have

$$|\beta - \alpha| = |f(\|\mathbf{y}\|, m(\|\mathbf{x}\|)) - f(\|\mathbf{x}\|, m(\|\mathbf{x}\|))|. \tag{4.108}$$

In this last case, both (4.99) and (4.100) have a solution for $c_1 = m(\|\mathbf{x}\|)$ and $r^* = \|\mathbf{x}\|$.

To ensure constant speed, the following equation needs to be satisfied:

$$\frac{\sqrt{\dot{r}^2 + r^2\dot{\theta}^2}}{\sqrt{r^2+\epsilon^2}} \quad = \quad S. \tag{4.109}$$

Since $\theta(r) = \pm(f(r, c_1) + c_\pm)$, we have

$$\frac{d}{dt}\theta(r(t)) = \pm r'(t) \frac{c_1}{r(t)\sqrt{\frac{r^2(t)}{r^2(t)+\epsilon^2} - c_1^2}}. \tag{4.110}$$

Substituting $\frac{d}{dt}\theta(r(t))$ in the constant speed condition (4.109), we obtain the following ODE:

$$|r'|\sqrt{\frac{1}{r^2+\epsilon^2} + \frac{c_1^2}{r^2 - c_1^2(r^2+\epsilon^2)}} \quad = \quad S. \tag{4.111}$$

The solution of this separable first order ODE can be then found by integration:

$$g(r) := \int \sqrt{\frac{1}{r^2+\epsilon^2} + \frac{c_1^2}{r^2 - c_1^2(r^2+\epsilon^2)}} dr = \pm S \int dt. \tag{4.112}$$

Let t^* be the time t for which $r(t^*) = r^*$. Let R and T be the constants of integration for each branch of $g(r)$. Then the initial conditions lead to these four equations:

$$g(\|\mathbf{x}\|) \quad = \quad \pm R, \tag{4.113}$$
$$g(r^*) \quad = \quad \pm(St^* + R), \tag{4.114}$$
$$g(r^*) \quad = \quad \mp(St^* + T), \tag{4.115}$$
$$g(\|\mathbf{y}\|) \quad = \quad \mp(S + T). \tag{4.116}$$

After subtracting (4.113) from (4.114) and also subtracting (4.115) from (4.116), the system simplifies to

$$\begin{aligned} g(r^*) - g(\|\mathbf{x}\|) &= \pm S t^*, & (4.117)\\ g(\|\mathbf{y}\|) - g(r^*) &= \mp S(1-t^*). & (4.118)\end{aligned}$$

Subtracting the first and second equation gives

$$\pm S = 2g(r^*) - g(\|\mathbf{x}\|) - g(\|\mathbf{y}\|), \tag{4.119}$$

whereas solving for t^* leads to

$$t^* = \frac{g(r^*) - g(\|\mathbf{x}\|)}{2g(r^*) - g(\|\mathbf{x}\|) - g(\|\mathbf{y}\|)}. \tag{4.120}$$

Thus

$$h(t) := g(r(t)) = \begin{cases} g(\|\mathbf{x}\|) + S't, & \text{if } t \leq t^*, \\ g(\|\mathbf{y}\|) + S'(1-t), & \text{if } t \geq t^*, \end{cases} \tag{4.121}$$

where $S' = 2g(r^*) - g(\|\mathbf{x}\|) - g(\|\mathbf{y}\|)$.

We still need to calculate the integral $g(r)$. In order to simplify the integral, we perform the change of variable $w = r^2 + \epsilon^2$, $dw = 2rdr$ and $a = \sqrt{1-c_1^2}$. The integral then becomes

$$\begin{aligned} z(w) := g(\sqrt{w - \epsilon^2}) &= \int \sqrt{\frac{1}{w} + \frac{1-a^2}{a^2 w - \epsilon^2}} \frac{1}{2\sqrt{w-\epsilon^2}} dw & (4.122)\\ &= \frac{1}{2} \int \sqrt{\frac{1}{w(a^2 w - \epsilon^2)}} dw & (4.123)\\ &= \frac{1}{a} \log(2a(\sqrt{a^2 w - \epsilon^2} + a\sqrt{w})). & (4.124)\end{aligned}$$

Thus

$$g(r) = \frac{\log\left(2\sqrt{1-c_1^2}(\sqrt{(1-c_1^2)(r^2+\epsilon^2) - \epsilon^2} + \sqrt{(1-c_1^2)(r^2+\epsilon^2)})\right)}{\sqrt{1-c_1^2}}, \tag{4.125}$$

with c_1 determined numerically by either (4.99) or (4.100).

Computing the inverse of $z(w)$ and substituting it back with $w = r^2 + \epsilon^2$ successfully leads to

$$w = \left(\frac{1}{4a^2}\exp(az(w)) + \epsilon^2\exp(-az(w))\right)^2 \text{ and} \tag{4.126}$$

$$r = \sqrt{\left(\frac{1}{4a^2}\exp(ag(r)) + \epsilon^2\exp(-ag(r))\right)^2 - \epsilon^2}. \tag{4.127}$$

We thus have

$$r(t) = \sqrt{\left(\frac{1}{4(1-c_1^2)}\exp(\sqrt{1-c_1^2}h(t)) + \epsilon^2\exp(-\sqrt{1-c_1^2}h(t))\right)^2 - \epsilon^2}. \tag{4.128}$$

Gathering everything together, we are now able to trace the constant speed path between $\mathbf{x}$ and $\mathbf{y}$.

In Fig. 4.2, we draw the geodesics and find the midpoints between $\mathbf{x} = (1/2, 0)$ and $\mathbf{y} = (1/2, 1)$ for different values of the parameter ϵ. Figure 4.3 offers a comparison of the geodesic path according to a metric normalized with r^q and with $\sqrt{r^2 + \epsilon^2}$ for different values of q and ϵ. In Appendix B, a Matlab implementation of the constant speed geodesics of normalized metrics can be found under the name `normalizedgeodesic`.

4.3 Geodesics of the SSIM Index

Recall from (3.12) that any element $\mathbf{x}$ in $\mathbf{R}^N$ can be decomposed into two orthogonal components containing the mean of $\mathbf{x}$ and a zero-mean vector obtained by orthogonal projection:

$$\mathbf{x} = \bar{\mathbf{x}} \oplus (\mathbf{x} - \bar{\mathbf{x}}). \tag{4.129}$$

One can then solve for $\bar{\mathbf{x}}$ and $\mathbf{x} - \bar{\mathbf{x}}$ independently to find the minimal path according to the SSIM index. The minimal path between $\mathbf{x}$ and $\mathbf{y}$ thus will be the minimal path between $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ plus the minimal path between $\mathbf{x} - \bar{\mathbf{x}}$ and $\mathbf{y} - \bar{\mathbf{y}}$.

Also, notice that the minimal path according to each component of the SSIM index is equivalent to the minimal path according to the normalized metric with $M(r) = \sqrt{r^2 + \epsilon^2}$. Indeed, these two paths are related by $f(x) = \sqrt{1 - x^2}$, which is one-to-one for $x \geq 0$. Thus to minimize x is equivalent to maximize $f(x)$.

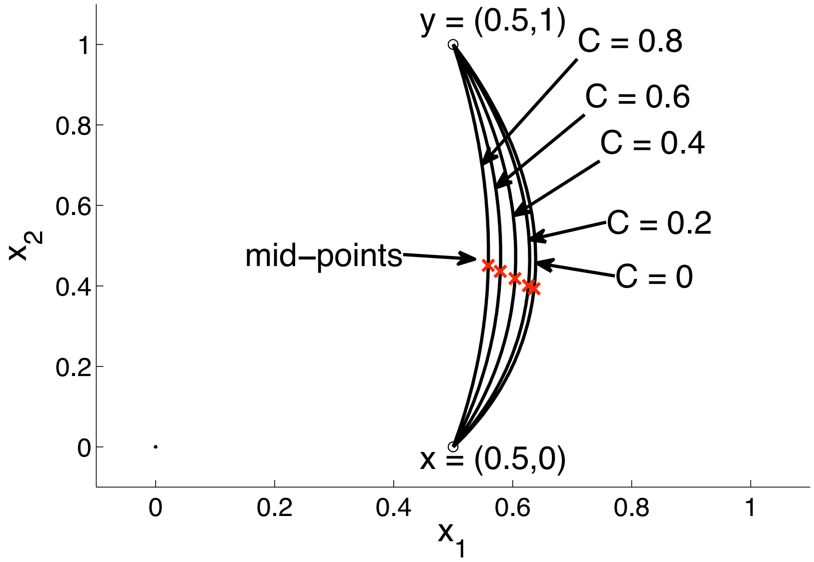


Figure 4.2: Geodesics and mid-points between $(1/2, 0)$ and $(1/2, 1)$ for $M(r, r) = \sqrt{r^2 + C^2}$ with $C = 0, 1/5, 2/5, 3/5$ and $4/5$.

Denote $\omega = \arccos(s_{x,y}/s_x s_y)$. For the case $\epsilon_1 = \epsilon_2 = 0$, we apply the solution of the minimal path for $\bar{x}$ and $\mathbf{x} - \bar{\mathbf{x}}$:

$$\gamma(t) = \bar{x}^{1-t}\bar{y}^t + s_x^{1-t}s_y^t\left(\frac{\sin(\omega(1-t))}{\sin(\omega)}\frac{\mathbf{x}-\bar{\mathbf{x}}}{s_x} + \frac{\sin(\omega t)}{\sin(\omega)}\frac{\mathbf{y}-\bar{\mathbf{y}}}{s_y}\right). \tag{4.130}$$

As a direct application of the SSIM geodesic, given two still images $\mathbf{x}$ and $\mathbf{y}$, it is possible to create a transition, or cross-fading, from $\mathbf{x}$ to $\mathbf{y}$ that preserves most structural information from both images. For example, it could be used for slide transition in a picture slideshow or for a scene transition in a movie. The goal is to make a transition that is perceptually linear. This means in particular that at time $t = 1/2$, the image obtained from the geodesic should be equally similar to both the images x and y. So instead of simply applying a linear interpolation $\mathbf{z}(t) = t\mathbf{x} + (1-t)\mathbf{y}$ that would lead to a non-linear perception of change, we apply the formula for the geodesic of SSIM.

Figure 4.4 shows an example of the SSIM-geodesic for $\epsilon_1 = \epsilon_2 = 0$. Some SSIM-interpolated images are compared with their linear interpolation counterparts. Note how the brightest image dominates the visual perception in the linear interpolation case while

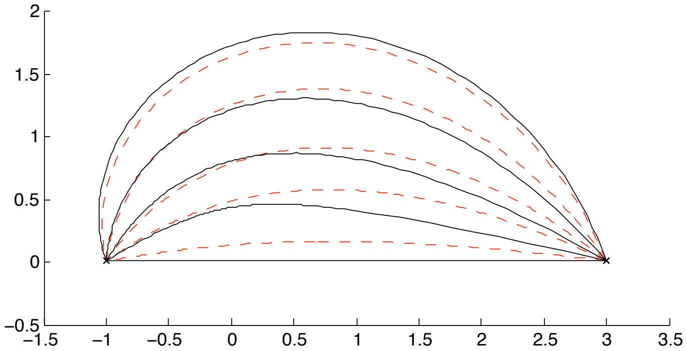


Figure 4.3: Geodesics between $(3, 0.01)$ and $(-1, 0.01)$ for $M(r, r) = r^q$ with $q = 0, 1/4, 1/2, 3/4, 1$ (full black) and for $M(r, r) = \sqrt{r^2 + \epsilon^2}$ with $\epsilon = \sqrt{0.1}, \sqrt{0.5}, \sqrt{0.9}, \sqrt{1.1}$ and $\sqrt{1.3}$ (dashed red).

both images are adequately weighted in the SSIM geodesic case. Note however that, as it is the case for linear interpolation, the SSIM geodesic fails to capture changes in the geometry of the images. A Matlab implementation of the image transition can be found in the Appendix B under the name `EXgeodesicSSIM`.

4.3.1 SSIM-geodesic Distance

One motivation for defining the geodesic according to SSIM was to derive a metric distance. As mentioned in the introducing paragraph of this chapter, this was also the motivation of Ritcher in [88]. But contrary to his work where he found, but not solved, a second order differential equation describing the SSIM-geodesic path, we analytically solved the problem up to the numerical determination of one constant. We now use this result to define a new distance between images.

From the orthogonal decomposition $\mathbf{x} = \bar{\mathbf{x}} \oplus (\mathbf{x} - \bar{\mathbf{x}})$, it is clear that the geodesic distance between $\mathbf{x}$ and $\mathbf{y}$ is

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(d_0(\bar{x}, \bar{y}))^2 + (d_1(\mathbf{x} - \bar{\mathbf{x}}, \mathbf{y} - \bar{\mathbf{y}}))^2}, \tag{4.131}$$

where d_0 and d_1 are the geodesic distances for each orthogonal components. Since we enforced the constant speed condition, the length of each geodesic path is equivalent to its speed for one unit of time. Thus

$$d(\mathbf{x},\mathbf{y}) = \sqrt{S_0^2 + S_1^2}, \tag{4.132}$$

where S_0 and S_1 are the (constant) speeds for each component of the geodesic path.

For the geodesic path of the mean component, we found in (4.24) that

$$\int \frac{1}{M(\rho,\rho)} d\rho \quad = \quad S_0 t + R_0, \tag{4.133}$$

where $\rho(t)$ here represents the evolution of the mean component and S_0 is its speed. The integral for the case $M(\rho,\rho) = \sqrt{\rho^2 + \epsilon^2}$ was solved in (4.79). Combining this result with the initial conditions $\rho(0) = \bar{x}$ and $\rho(1) = \bar{y}$, we obtain

$$S_0 \quad = \quad \log(2\sqrt{\bar{y}^2 + \epsilon_1^2} + 2\bar{y}) - \log(2\sqrt{\bar{x}^2 + \epsilon_1^2} + 2\bar{x}). \tag{4.134}$$

For the zero-mean component, in (4.119) the speed was found to be

$$S_1 \quad = \quad |2g(r^*) - g(s_x) - g(s_y)|, \tag{4.135}$$

where g is defined in (4.125) and r^* has to be solved for numerically as described previously.

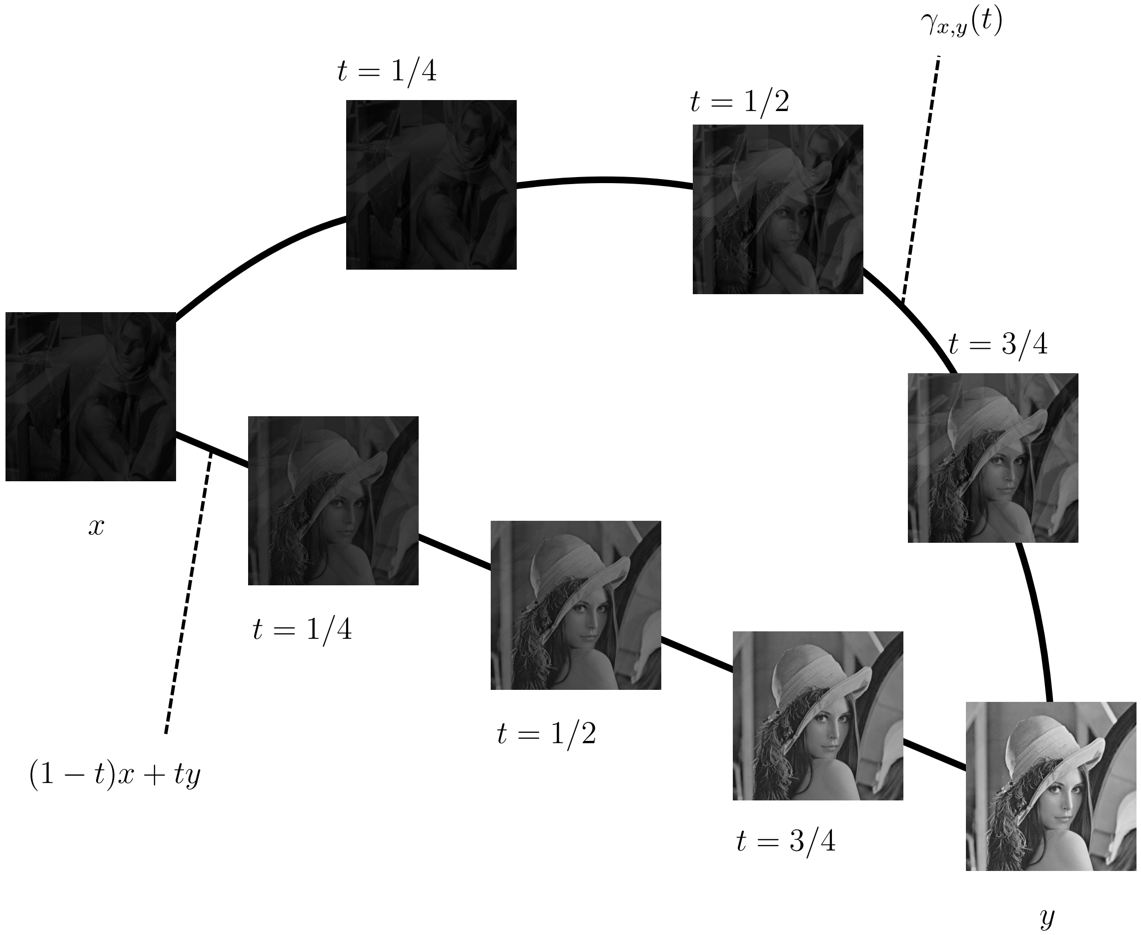


Figure 4.4: Image geodesics between Barbara (low luminance and contrast) and Lena according to the SSIM index and the Euclidean metric (linear path).

Chapter 5

Maximum SSIM Estimators

The problem of interest consists in approximating a weighted set of points on a real line or in a multi-dimensional space with a single number or vector. In the context of statistical inference, the weights represent probabilities and the result is called a point estimator. For image processing, the set of points are interpreted as image patches and the estimator aggregates them into an average image. It is desired for the average to be perceptually relevant, i.e. that the averaged image represents best the content of the set of patches from a visual perspective. This is why we are interested in maximum Structural Similarity (SSIM) estimators.

The implications of finding the SSIM-mean are important. First, it would allow to compute the Maximum A Posteriori estimator with $1 - \text{SSIM}$ has a loss function. Second, it could lead to the development of a K-means type algorithm for the classification of image patches. Third, it could be the basis on which an SSIM-based aggregation algorithm of overlapping patches would be devised.

The organization of the chapter is as follows: First, we give a review of some classical point estimators in $\mathbf{R}$ and $\mathbf{R}^N$. These solutions are easily derived and probably well known, but we present them starting from first principles. The problem of finding point estimators maximizing SSIM is then attacked. We start with the simplest one-dimensional case and find some analytical results before moving to the general multi-dimensional case. These estimates are compared with the arithmetic mean and the SSIM-geodesic mean for both low dimensional paths and for images.

5.1 Classical Point Estimators

In the discrete case, the problem can be formulated as

$$\mathbf{z} \;=\; \arg\min_{\mathbf{z}^*} \left(\sum_{k=1}^{K} p_k (d(\mathbf{x}_k, \mathbf{z}^*))^r \right)^{1/r}, \tag{5.1}$$

where each $\mathbf{x}_k = (x_{k,1}, x_{k,2}, \dots, x_{k,N})$ consists of a vector in $\mathbf{R}^N$ and $(p_1, p_2, \dots, p_K)$ represents the set of weights. In the continuous case we replace the weights by a probability density function (p.d.f.) and obtain:

$$\mathbf{z} \;=\; \arg\min_{\mathbf{z}^*} \left(\int p(\mathbf{x})(d(\mathbf{x}, \mathbf{z}^*))^r d\mathbf{x} \right)^{1/r}. \tag{5.2}$$

Depending on the choice of the metric $d(\cdot,\cdot)$ and of the exponent r we will find different estimators. We describe the most common point estimators for both the discrete and continuous case.

Mean. As it is well known, the point estimator that minimizes the expected mean squared error is the arithmetic mean. This error corresponds to an Euclidean norm raised to the power $r = 2$. The quadratic risk is then

$$\begin{aligned} R_2(\mathbf{z}) &= \int p(\mathbf{x}) \|\mathbf{x} - \mathbf{z}\|_2^2 d\mathbf{x} & (5.3) \\ &= \sum_{n=1}^{N} \int p(\mathbf{x}) |x_n - z_n|^2 d\mathbf{x}. & (5.4) \end{aligned}$$

To show that the mean estimator minimizes the quadratic risk, simply take the partial derivative of (5.3) with respect to each of the N components of $\mathbf{z}$ and set the result to zero to find the critical points. By convexity of (5.3), it is clear that the unique critical point is the global minimum. We thus have

$$\mathbf{z} = \frac{\sum_{k=1}^{K} p_k \mathbf{x}_k}{\sum_{k=1}^{K} p_k} \tag{5.5}$$

in the discrete case and

$$\mathbf{z} = \int \mathbf{x} p(\mathbf{x}) d\mathbf{x} \tag{5.6}$$

in the continuous case.

Median. The absolute risk on the real line is

$$R_1(z) = \int p(x)|x - z| dx. \tag{5.7}$$

The point estimator that minimizes this risk is the median of the p.d.f. $p(x)$.

There is at least two ways to generalize this result to $\mathbf{R}^N$ by either replacing the absolute value by a L^1-norm or a L^2-norm. The first case simply leads to a component-wise median and is sometimes called the *marginal median.* Remark that the fact that this estimator is not rotationally invariant could be an issue.

The second king of multivariate median is known as L^1-median, geometric median or spatial median. Finding the solution of

$$\arg\min_{\mathbf{z}} \sum_{k=1}^{K} p_k \|\mathbf{x}_k - \mathbf{z}\|_2 \tag{5.8}$$

is more computationally challenging (see e.g. [104]).

Mid-Range. The mid-range of $(x_1, x_2, \ldots, x_K)$ on $\mathbf{R}$ is defined as

$$z = \frac{x_{(1)} + x_{(K)}}{2} \tag{5.9}$$

where $x_{(k)}$ is the k-th rank order statistic. This statistic is the point estimator that minimizes the maximum distance between z and any other point. For the general multi-dimensional continuous case, we search the point $\mathbf{z}$ such that

$$R_\infty(\mathbf{z}) = \max_{1 \leq k \leq K} (p_k \|\mathbf{z} - \mathbf{x}_k\|_2) \tag{5.10}$$

is minimized.

Given a pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$, the $\mathbf{z}$ which minimizes the maximum weighted distance between both of these points must satisfy

$$p_i \|\mathbf{x}_i - \mathbf{z}\|_2 = p_j \|\mathbf{x}_j - \mathbf{z}\|_2. \tag{5.11}$$

It implies that

$$\mathbf{z} = \frac{p_i \mathbf{x}_i + p_j \mathbf{x}_j}{p_i + p_j}. \tag{5.12}$$

It remains to find $\mathbf{x}_i$ and $\mathbf{x}_j$. Substituting the z found in (5.12) into (5.11), we find

$$\begin{aligned} p_i \|\mathbf{x}_i - \mathbf{z}\|_2 &= p_i \left\| \mathbf{x}_i - \frac{p_i \mathbf{x}_i + p_j \mathbf{x}_j}{p_i + p_j} \right\|_2 & (5.13) \\ &= \frac{p_i p_j}{p_i + p_j} \|\mathbf{x}_i - \mathbf{x}_j\|_2. & (5.14) \end{aligned}$$

Choosing the pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ that maximizes this last distance leads to the mid-range point estimator.

5.2 SSIM-Mean

We want to solve the following problem: Given $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K\}$ a set of K images and $(p_1, p_2, \ldots, p_K)$ a set of associated weights across patches, find $\mathbf{z}$ maximizing

$$\sum_{k=1}^{K} p_k \, \mathrm{SSIM}(\mathbf{x}_k, \mathbf{z}). \tag{5.15}$$

We will study three cases in order: the one dimensional case, the zero-mean case and the general case.

5.2.1 One Dimensional Case

Consider two images with constant luminance. Their variance and covariance will vanish and the one dimensional problem thus reduces to

$$z^* = \arg\max_z \sum_{k=1}^{K} p_k \frac{2 z x_k + \epsilon_1}{z^2 + x_k^2 + \epsilon_1}. \tag{5.16}$$

If all the points are close enough, we can prove existence and uniqueness of the minimizer.

Proposition 5.2.1. *Let $\{x_1, x_2, \ldots, x_K\}$ be K points in $\mathbf{R}_+$. If $\frac{\max_k(x_k)}{\min_k(x_k)} \leq \sqrt{3}$, then there exists a unique minimizer to the problem (5.16).*

Proof. Denote each term of (5.16) by $g_k(z)$ and the total sum by $g(z)$. First observe that the local convexity property (Theorem 3.3.10) implies that $g_k(z)$ is convex on $0 \leq z \leq \sqrt{3}x_k$. Hence, $g(z)$ is convex on $0 \leq z \leq \min_k \sqrt{3}x_k$. Now, if $\frac{\max_k(x_k)}{\min_k(x_k)} \leq \sqrt{3}$, then all the points $\{x_1, x_2, \ldots, x_K\}$ lie in the region of convexity of $g(z)$. By quasi-concavity of SSIM on $\mathbf{R}_+$, the maximizer z^* must be between $\min_k x_k$ and $\max_k x_k$ since $g(z)$ is decreasing for $z \leq \min_k x_k$ and is increasing for $z \geq \max_k x_k$. These two last facts put together with Theorem 3.3.3 implies the uniqueness of the maximizer. Its existence follows easily from the fact that $g(z)$ is a continuous function on the interval $\min_k x_k \leq z \leq \max_k x_k$. □

In general we could have more than one local maximum. For example, for $K = 2$ with $\epsilon_1 = 0$ and with $p_1 = p_2 = 1$ we can derive a complete analytical solution.

Proposition 5.2.2. *The function*

$$f(z) = \frac{2zx_1}{z^2 + x_1^2} + \frac{2zx_2}{z^2 + x_2^2}, \quad 0 < z < \infty, \tag{5.17}$$

has a unique maximum at $z = \sqrt{x_1 x_2}$ if $3 - 2\sqrt{2} \leq x_1/x_2 \leq 3 + 2\sqrt{2}$ and has two global maxima at

$$z_\pm = \sqrt{\frac{x_1^2 + x_2^2 - 4x_1x_2 \pm \sqrt{(x_1^2 + x_2^2 - 4x_1x_2)^2 - 4x_1^2x_2^2}}{2}} \tag{5.18}$$

and a local minimum at $\sqrt{x_1x_2}$ otherwise.

Proof. To find the critical points, we take the derivative of f with respect to z and set it to zero:

$$f'(z) = \frac{2x_1(x_1^2 - z^2)}{(z^2 + x_1^2)^2} + \frac{2x_2(x_2^2 - z^2)}{(z^2 + x_2^2)^2} = 0. \tag{5.19}$$

After simplifications, it leads to the following equation of the 6-th degree:

$$z^6(x_1+x_2) - z^4(x_1^3 - 2x_1^2x_2 - 2x_1x_2^2 + x_2^3) + z^2x_1x_2(x_1^3 - 2x_1^2x_2 - 2x_1x_2^2 + x_2^3) - x_1^4x_2^3 - x_1^3x_2^4 = 0. \tag{5.20}$$

Dividing out the factor $(x_1 + x_2)$ and substituting $Z = z^2$ we find

$$Z^3 - Z^2(x_1^2 - 3x_1x_2 + x_2^2) + Zx_1x_2(x_1^2 - 3x_1x_2 + x_2^2) - x_1^3x_2^3 = 0. \tag{5.21}$$

It is clear that $Z = x_1x_2$ is one of the solution. After polynomial division we obtain

$$Z^2 - Z(x_1^2 - 4x_1x_2 + x_2^2) + x_1^2x_2^2 = 0. \tag{5.22}$$

The two solutions of this quadratic equation are

$$Z = \frac{(x_1^2 - 4x_1x_2 + x_2^2) \pm \sqrt{(x_1^2 - 4x_1x_2 + x_2^2)^2 - 4x_1^2x_2^2}}{2}. \tag{5.23}$$

Note that these solutions are real if and only if $(x_1^2 - 4x_1x_2 + x_2^2)^2 - 4x_1^2x_2^2 \geq 0$. Thus if

$$3 - 2\sqrt{2} \leq \frac{x_1}{x_2} \leq 3 + 2\sqrt{2}, \tag{5.24}$$

then f will have a unique (positive) critical point at $z = \sqrt{x_1x_2}$. Since $f(0) = 0$, $f(\infty) = 0$ and $f(z) \geq 0$ for $0 < z < \infty$, f will thus be maximal at $z = \sqrt{x_1x_2}$. On the other hand, if

$$3 - 2\sqrt{2} \geq \frac{x_1}{x_2} \text{ or } \frac{x_1}{x_2} \geq 3 + 2\sqrt{2}, \tag{5.25}$$

then f will have three positive critical points. By studying the sign of the second derivative of f at $\sqrt{x_1x_2}$ we can determine that it is a local minimum. Indeed,

$$f''(z) = \frac{4x_1z(z^2 - 3x_1^2)}{(z^2 + x_1^2)^3} + \frac{4x_2z(z^2 - 3x_2^2)}{(z^2 + x_2^2)^3} \tag{5.26}$$

and

$$f''(\sqrt{x_1x_2}) = \frac{4\sqrt{x_1x_2}}{(x_1 + x_2)^3}\left(\frac{x_1}{x_2} + \frac{x_2}{x_1} - 6\right). \tag{5.27}$$

We then have $f''(z) \leq 0$ if and only if

$$3 - 2\sqrt{2} \leq \frac{x_1}{x_2} \leq 3 + 2\sqrt{2}. \tag{5.28}$$

Note that f has multiplicative symmetry in the sense that $f(y_1) = f(y_2)$ if $y_1y_2 = x_1x_2$. It implies that the two other critical points of f are necessary local maxima with equal values. □

Figure 5.1 shows an example of the plotting of the function $f(z)$ described in (5.17) for two choices of x_1 and x_2. For the first combination (shown in blue), $x_1 = 0.1$ and $x = 0.4$. In this case, the absolute maximum of the sum of the one dimensional structural similarity is their geometric mean, i.e. at $z = 0.2$. For the second combination (shown in magenta), $x_1 = 0.05$ and $x_2 = 0.8$. In this other case, the geometric mean of these points is the same, but it is a local minimum instead of a global maximum. The two global maxima on each side of the geometric mean are shown with red Xs.

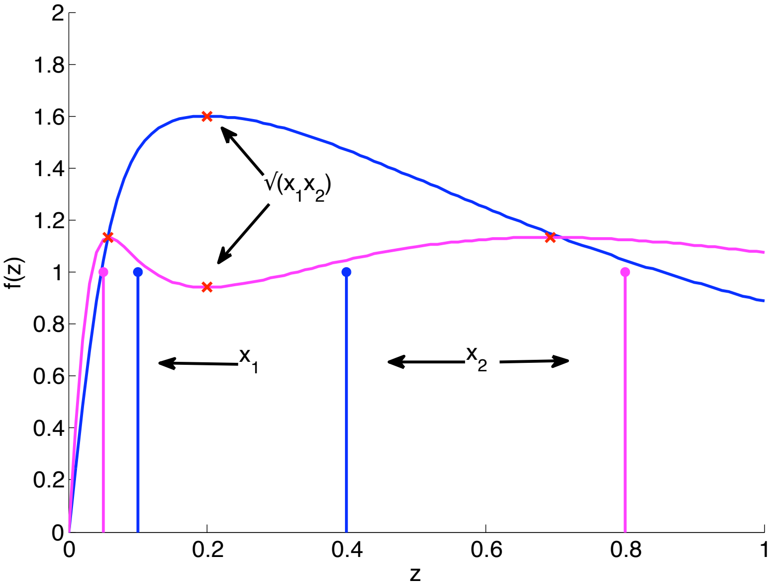


Figure 5.1: Plot of the one dimensional sum of the structural similarity between two points x_1 and x_2 for stability constant $\epsilon_1 = 0$. Blue: $x_1 = 0.1$ and $x_2 = 0.4$. Magenta: $x_1 = 0.05$ and $x_2 = 0.8$. The critical points of each function are shown with red Xs. Note how the geometric mean $\sqrt{x_1 x_2}$ is either a global maximum or a local minimum depending on how far x_1 and x_2 are from each other.

Note that we can find an approximation of the analytical solution for the case $\epsilon_1 > 0$ by writing the solution as a Taylor series in ϵ. The polynomial equation to solve is of the form

$$f_0(z) + \epsilon_1 f_1(z) + \epsilon_1^2 f_2(z) + \epsilon_1^3 f_3(z) = 0 \tag{5.29}$$

where $f_i(z)$ for $0 \leq i \leq 3$ are polynomials in z. The linear approximation of the solution will be

$$z = z_0 + \epsilon_1 z_1 \tag{5.30}$$

where z_0 is the solution for $\epsilon_1 = 0$ and z_1 is the first order approximation coefficient. Substituting z in the polynomial equation and keeping only the first order terms, we obtain

$$f_0(z_0) + \epsilon_1(Az_1 + B) = 0. \tag{5.31}$$

Since $f_0(z_0) = 0$, the linear approximation coefficient will be $z_1 = -B/A$ where A and B are in function of x_1 and x_2 only. Following the method laid down in the previous proof combined with the technique to find the linear perturbation, one can find

$$\begin{aligned} A &= 6z_0^5(x_1 + x_2) + 4z_0^3(2x_1x_2^2 + 2x_1^2x_2 - x_1^3 - x_2^3) \\ &\quad +2z_0(x_1x_2^4 + x_1^4x_2 - 2x_1^2x_2^3 - 2x_1^3x_2^2), \qquad (5.32) \\ B &= 2z_0^5 + z_0^4(x_1 + x_2) + 2z_0^3(x_1^3 + x_2^3) - 2z_0^2(x_1^3 + x_2^3) + z_0(x_1^4 + x_2^4) \\ &\quad -(x_1x_2^4 + x_1^4x_2 + 2x_1^2x_2^3 + 2x_1^3x_2^2). \qquad (5.33) \end{aligned}$$

In Figs. 5.2 and 5.3 a visual comparison of the analytical solution for the critical points of $f(z) = \frac{2x_1z+\epsilon_1}{x_1^2+z^2+\epsilon_1} + \frac{2x_2z+\epsilon_1}{x_2^2+z^2+\epsilon_1}$ when $\epsilon_1 = 0$ with the linear approximation of the analytical solution for $\epsilon_1 > 0$ and of the numerical approximation of the solution for $\epsilon_1 > 0$ is presented for two cases: i) $x_1 = 0.1$ and $x_2 = 0.4$ and ii) $x_1 = 0.05$ and $x_2 = 0.8$. Only a global maximum is found in the first case, whereas in the second case we observe two local maxima and one local minimum.

To find the numerical approximations, we employed a standard one dimensional search over an interval (Brent's method [7]). Starting from the analytical solution, we determined in a first step how many extrema to search for. This number will not change for ϵ_1 small enough. We then constructed the search intervals from the position and number of extrema in the case $\epsilon_1 = 0$.

Observe that the numerical approximation is generally more precise that the analytical approximation. Moreover, if ϵ_1 grows too much away from 0, the analytical linear approximation will not be valid anymore and the analytical solutions for the case $\epsilon_1 = 0$ cannot be used as bounds for the search algorithm. Finally, note that when ϵ_1 is not negligible, the number of extrema could pass from 3 to 1. The best option in this case seems to first sample the function to be optimized at regular intervals (possibly with a logarithmic scale) and then to use numerical methods to determine more precisely the location of local extrema once they are detected within an interval. That is, if there exists $z_{k-1} < z_k < z_{k+1}$ such that $f(z_{k-1}) < f(z_k)$ and $f(z_k) > f(z_{k+1})$, then we search for a local maximum within the interval $]z_{k-1}, z_{k+1}[$.

The one-dimensional case for two points with zero stability constant and non-uniform weights can also be solved analytically. This will allow us to compare this solution with the SSIM-geodesic path and the linear path between two points. By normalizing the weights with $\lambda = \frac{w_1}{w_1+w_2}$ and $1-\lambda = \frac{w_2}{w_1+w_2}$, the problem to solve becomes

$$\arg\max_z f_\lambda(z) = \arg\max_z \left((1-\lambda)\frac{2zx_1}{z^2+x_1^2} + \lambda \frac{2zx_2}{z^2+x_2^2} \right), \quad 0 < z < \infty. \tag{5.34}$$

Observe that when $z = \sqrt{x_1 x_2}$, then $f_\lambda(z) = \frac{4\sqrt{x_1x_2}}{x_1+x_2}$ for all λ. Also, it is clear that for every z fixed, $f_\lambda(z)$ is decreasing with respect to λ on $0 \leq z \leq \sqrt{x_1x_2}$ and increasing with respect to λ on $\sqrt{x_1x_2} \leq z \leq 1$. Since we know from Proposition 5.2.2 that $f_{1/2}(z)$ has either two maxima with one on each side of $\sqrt{x_1x_2}$ or a unique maximum at $\sqrt{x_1x_2}$, thus the maximum of $f_\lambda(z)$ is in the interval $[0, \sqrt{x_1x_2}]$ for $0 \leq \lambda \leq 1/2$ and in the interval $[\sqrt{x_1x_2}, 1]$ for $1/2 \leq \lambda \leq 1$.

Since we now know how to compute the SSIM weighted mean between two points, we can as well draw a path between two points by varying the weights. Figures 5.4 and 5.5 illustrate these paths with an example for each of the two cases of Proposition 5.2.2. The position of the SSIM weighted mean is plotted as a function of the weight ratio λ. A very interesting observation is the presence of a discontinuity in the second case. This corresponds to a jump in the path describing the linearly weighted average between these two points. Notice that this jump represents the change of mode from one local maximum to the next. Thus, if a smooth transition between two images is desired, the SSIM-geodesic is the method of choice since it will not lead to abrupt changes in the series of images. On the other hand, to take the SSIM-mean is the way to go if we want to estimate the perceptual average between several images.

5.2.2 N-Dimensional Case

For zero-mean signal, i.e. for the second component S_2 of the SSIM index, the cost function to maximize is

$$G(z) = \sum_{k=1}^{K} p_k S_2(\mathbf{x}_k, \mathbf{z}) \tag{5.35}$$

$$= \sum_{k=1}^{K} p_k \frac{2\langle \mathbf{x}_k, \mathbf{z}\rangle + \epsilon_2}{\|\mathbf{x}_k\|_2^2 + \|\mathbf{z}\|_2^2 + \epsilon_2}. \tag{5.36}$$

We solve this problem for a general quality metric.

Theorem 5.2.3. *The mean of* $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K$ *according to the quality metric*

$$Q(\mathbf{x}, \mathbf{y}) = f(\|\mathbf{x}\|, \|\mathbf{y}\|)\langle \mathbf{x}, \mathbf{y}\rangle + g(\|\mathbf{x}\|, \|\mathbf{y}\|), \tag{5.37}$$

i.e.

$$\mathbf{m} = \arg\max_{\mathbf{z}} \sum_{k=1}^{K} p_k f(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\langle \mathbf{x}_k, \mathbf{z}\rangle + g(\|\mathbf{x}_k\|, \|\mathbf{z}\|) \tag{5.38}$$

is in the direction of

$$\mathbf{u} = \sum_{k=1}^{K} p_k f(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\mathbf{x}_k. \tag{5.39}$$

Proof. Let $\mathbf{u} = \mathbf{z}/\|\mathbf{z}\|$. Then, adding this constraint with a lagragian λ,

$$\mathbf{m} = \sum_{k=1}^{K} p_k \big(f(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\langle \mathbf{x}_k, \mathbf{u}\|\mathbf{z}\|\rangle + g(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\big) + \lambda(1 - \|\mathbf{u}\|). \tag{5.40}$$

Taking the derivative with respect to each component of u, we obtain

$$\frac{d}{d\mathbf{u}}\mathbf{m} = \|\mathbf{z}\| \sum_{k=1}^{K} p_k f(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\mathbf{x}_k - \lambda \frac{\mathbf{u}}{\|\mathbf{u}\|}. \tag{5.41}$$

Since $\|\mathbf{u}\| = 1$, setting the derivative to zero, it simplifies to

$$\mathbf{u} = \frac{\|\mathbf{z}\|}{\lambda} \sum_{k=1}^{K} p_k f(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\mathbf{x}_k. \tag{5.42}$$

It remains to determine λ by substituting $\mathbf{u}$ into the constraint $\|\mathbf{u}\| = 1$:

$$\|\mathbf{u}\| = \frac{\|\mathbf{z}\|}{\lambda} \left(\sum_{j=1}^{K} \sum_{k=1}^{K} p_j p_k f(\|\mathbf{x}_j\|, \|\mathbf{z}\|) f(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\langle \mathbf{x}_j, \mathbf{x}_k\rangle \right)^{1/2} = 1. \tag{5.43}$$

Isolating λ and substituting back into the expression for $\mathbf{u}$ we finally conclude that

$$\mathbf{u}(\|\mathbf{z}\|) = \frac{\sum_{k=1}^{K} p_k f(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\mathbf{x}_k}{\left(\sum_{j=1}^{K} \sum_{k=1}^{K} p_j p_k f(\|\mathbf{x}_j\|, \|\mathbf{z}\|) f(\|\mathbf{x}_k\|, \|\mathbf{z}\|)\langle \mathbf{x}_j, \mathbf{x}_k\rangle\right)^{1/2}}. \tag{5.44}$$

Notice that the denominator of $\mathbf{u}$ is the norm of the numerator. □

Thus we can find analytically the maximal point on any given hyper-sphere of fixed radius. By varying the radius between 0 and ∞ we can find numerically $\|\mathbf{z}\|$. In practice the magnitude of $\mathbf{z}$ is bounded between 0 and 1. The problem to solve is

$$\begin{aligned}\|\mathbf{z}\| &= \arg\max_r \left[\sum_{k=1}^{K} p_k\big(f(\|\mathbf{x}_k\|, r)\langle \mathbf{x}_k, r\mathbf{u}(r)\rangle + g(\|\mathbf{x}_k\|, r)\big)\right] \qquad (5.45)\\ &= \arg\max_r \left[r \left(\sum_{j=1}^{K}\sum_{k=1}^{K} p_j p_k f(\|\mathbf{x}_k\|, r) f(\|\mathbf{x}_j\|, r)\langle \mathbf{x}_k, \mathbf{x}_j\rangle \right)^{1/2} + \sum_{k=1}^{K} p_k g(\|\mathbf{x}_k\|, r)\right].\end{aligned}$$

We then write $\mathbf{z} = \|\mathbf{z}\|\mathbf{u}$, where $\mathbf{u}(\|\mathbf{z}\|)$ is the maximal point on a given hyper-sphere.

For zero-mean signals, the magnitude of the mean SSIM is then

$$\begin{aligned}\|\mathbf{z}\| &= \arg\max_r \left[2r \left(\sum_{j=1}^{K}\sum_{k=1}^{K} \frac{p_j p_k \langle \mathbf{x}_j, \mathbf{x}_k\rangle}{(\|\mathbf{x}_j\|^2 + r^2 + \epsilon_2)(\|\mathbf{x}_k\|^2 + r^2 + \epsilon_2)} \right)^{1/2} \right.\\ &\qquad \left. + \sum_{k=1}^{K} \frac{p_k \epsilon_2}{\|\mathbf{x}_k\|^2 + r^2 + \epsilon_2} \right]. \qquad (5.46)\end{aligned}$$

For non-zero mean signals, we simply replace p_k by $p_k S_1(\bar{\mathbf{x}}_k, \bar{\mathbf{z}})$ and $\mathbf{x}_k$ by $\mathbf{x}_k - \bar{\mathbf{x}}_k$ as well as $\|\mathbf{z}\|$ by $\mathbf{z} - \bar{\mathbf{z}}$ in the formula (5.46) for the numerical optimization of $\|\mathbf{z}\|$. We obtain

$$\begin{aligned}(\bar{\mathbf{z}}, s_{\mathbf{z}}) &= \arg\max_{(b,r)} \Bigg[2r \left(\sum_{j=1}^{K}\sum_{k=1}^{K} p_k p_j \left(\frac{S_1(\bar{\mathbf{x}}_j, b)}{s_{\mathbf{x}_j}^2 + r^2 + \epsilon_2} \right) \left(\frac{S_1(\bar{\mathbf{x}}_k, b)}{s_{\mathbf{x}_k}^2 + r^2 + \epsilon_2} \right) s_{\mathbf{x}_j, \mathbf{x}_k} \right)^{1/2} \\ &\qquad + \sum_{k=1}^{K} \frac{p_k S_1(\bar{\mathbf{x}}_k, b)\epsilon_2}{s_{\mathbf{x}_k}^2 + r^2 + \epsilon_2} \Bigg]. \qquad (5.47)\end{aligned}$$

Here, we have to optimize in function of both $\bar{\mathbf{z}}$ and $s_{\mathbf{z}}$. The complete solution is then $\mathbf{z} = s_{\mathbf{z}}\mathbf{u} + \bar{\mathbf{z}}$.

We thus have in hands two non-linear optimization problems, one with a single variable and the other one with two variables. Note that in both of these cases the global maximum is not necessary unique, but that in practice it will generally be so.

For the one dimensional optimization problem, we employ the same technique that was outlined in the previous section. We first sample the domain $[0, 1]$ on a logarithmic scale. This choice of sampling is due to the multiplicative nature of the structural similarity metric. Let $(t_1, t_2, \ldots, t_n)$ be n uniformly sampled points between 0 and 1. The transformation $t_i \mapsto 10^{-bt_i}$ with b a fixed integer sends the uniform interval $[0, 1]$ to a logarithmic scale between 10^{-b} and 1. The location of local maxima can be determined from this sampling by comparing the magnitude of three successive points. The standard black-box method (Matlab's `fminbnd`) alternates between a golden section search and a parabolic interpolation (see [7]). The global maximum is finally determined by taking the greatest local maximum.

An example of the S_2-mean of five random points in the plane $\mathbf{R}^2$ is shown in Fig. 5.6. This also corresponds to the SSIM-mean of five points lying on the zero-mean plane, a $\mathbf{R}^2$ subspace of the $\mathbf{R}^3$ space. Figure 5.7 illustrates the one dimensional optimization to find the norm of $\mathbf{z}$. Notice how the SSIM-mean differs with the arithmetic mean: the SSIM-mean roughly takes the average direction (or angle) of the five points whereas the arithmetic mean takes, as we already know, the average position. This particular example was chosen to showcase the difference between the two estimators, but it is observed that the S_2-mean and the arithmetic mean are often comparable.

For the two-dimensional optimization problem, a similar sampling is performed on a two-dimensional grid in $[0, 1] \times [0, 1]$. The points that are detected as locally maximal in both directions constitute the initial guesses of the numerical scheme. The standard medium-scale black-box method (Matlab's `fminunc`) uses Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method (see [43]).

The main function is implemented in Matlab in Appendix B under the name `ssimmean`. The two functions to be optimized for zero-mean and non zero-mean are constructed in, respectively, `funopt1` and `funopt2`. The Matlab function `globalmin` proceeds with the global optimization for one or two dimensions.

We provide two examples of the SSIM-mean on a two-dimensional plane. In the first case, we took all the points on the same side of the mean direction $y = x$ and in the same quadrant. Thus, after the decomposition of the points into a mean component in the direction of $y = x$ and a zero-mean component in the direction of $y = -x$, all the zero-mean components of the points were trivially positively correlated. A comparison between the SSIM-mean and the arithmetic mean is then shown in Fig. 5.8. The result of the optimization over the mean component $\bar{\mathbf{z}}$ and the norm of the zero-mean component $s_{\mathbf{z}}$ can be seen in Fig. 5.9. It is observed that the points with small zero-mean components drag the SSIM-mean toward the axis $y = x$.

As it is illustrated in Fig. 5.10, the sum of the SSIM index exhibits some strange behavior for points with negatively correlated zero-mean component. In this case, the SSIM-mean will be on the side of the mean direction $y = x$ for which a majority of points lie. Because of the influence of points that are anti-correlated, the SSIM-mean might be outside the convex hull of the data points. This undesirable result indicates the poor performance of the SSIM index for signals that are negatively correlated. Also, observe in Fig. 5.11 how several local maxima are found in the optimization procedure for this particular example.

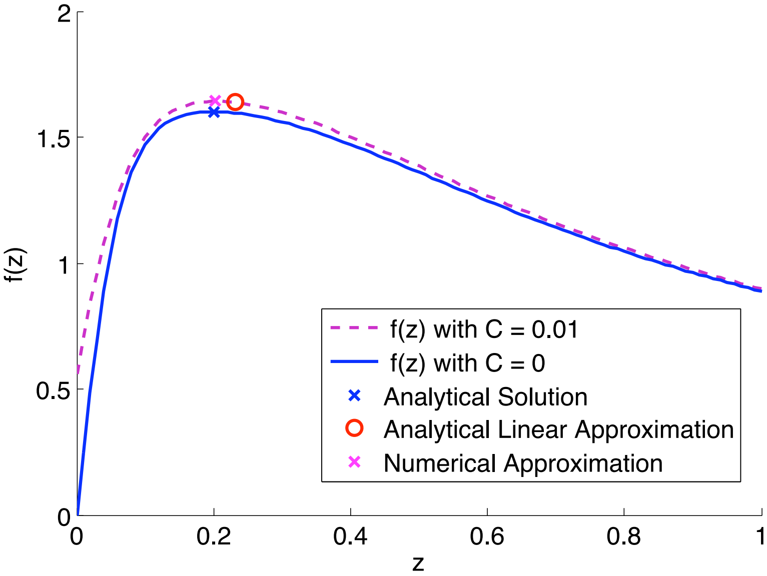


Figure 5.2: The function $f(z) = \frac{2x_1z+C}{x_1^2+z^2+C} + \frac{2x_2z+C}{x_2^2+z^2+C}$ with $x_1 = 0.1$ and $x_2 = 0.4$ and its extrema.

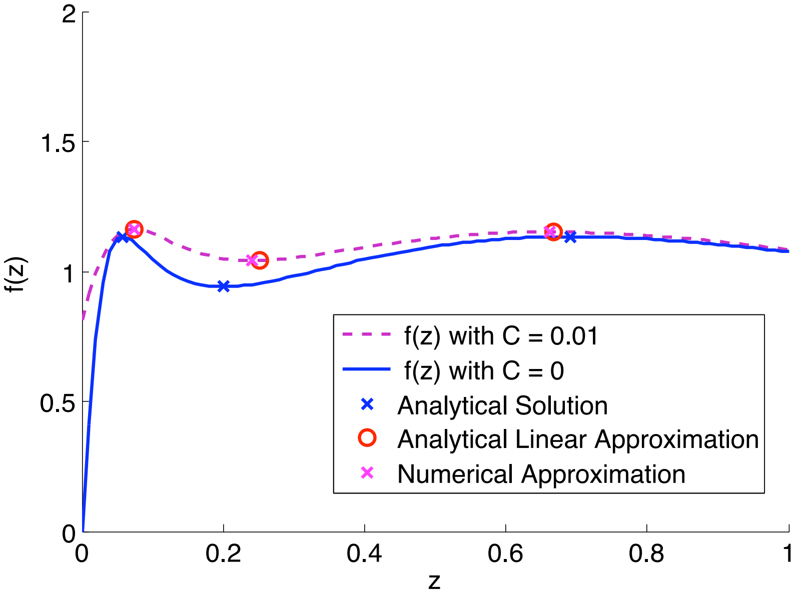


Figure 5.3: The function $f(z) = \frac{2x_1z+C}{x_1^2+z^2+C} + \frac{2x_2z+C}{x_2^2+z^2+C}$ with $x_1 = 0.05$ and $x_2 = 0.8$ and its extrema.

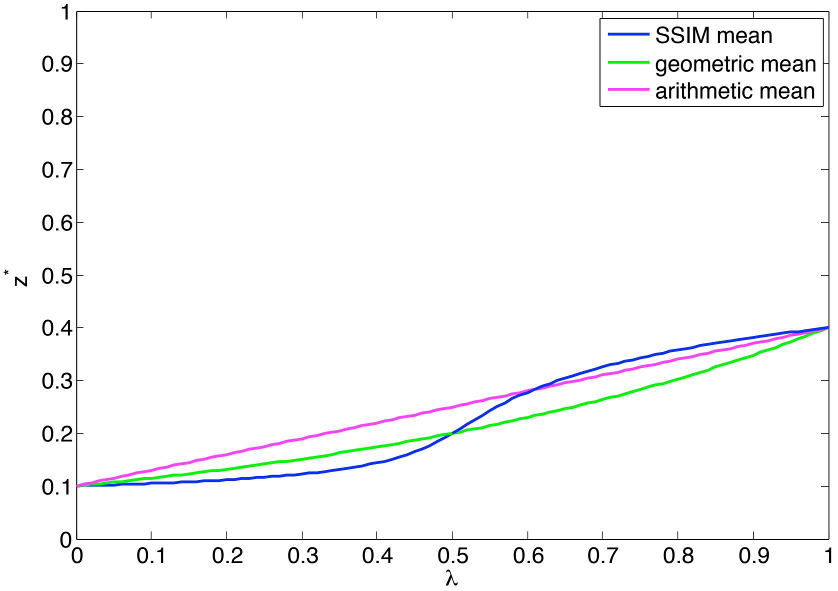


Figure 5.4: Optimal path between $x_1 = 0.1$ and $x_2 = 0.4$ according to different metrics.

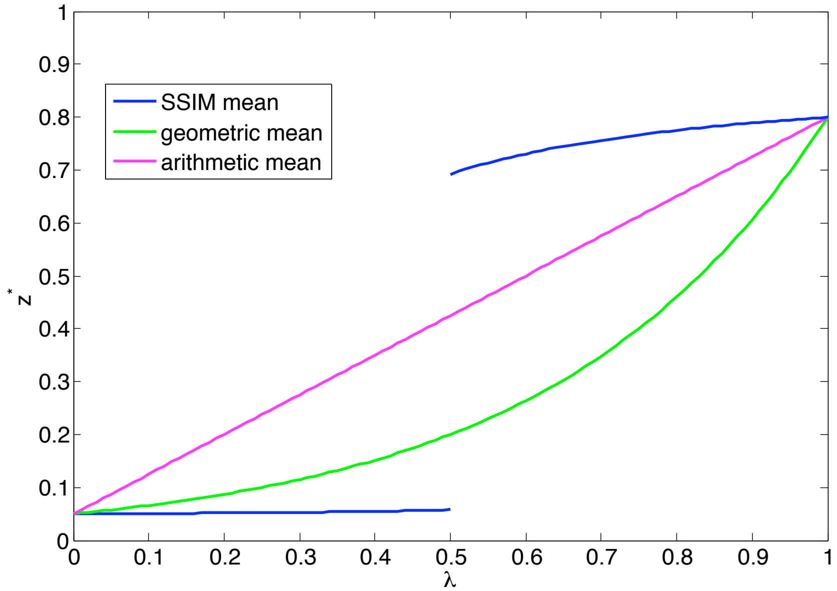


Figure 5.5: Optimal path between $x_1 = 0.05$ and $x_2 = 0.8$ according to different metrics.

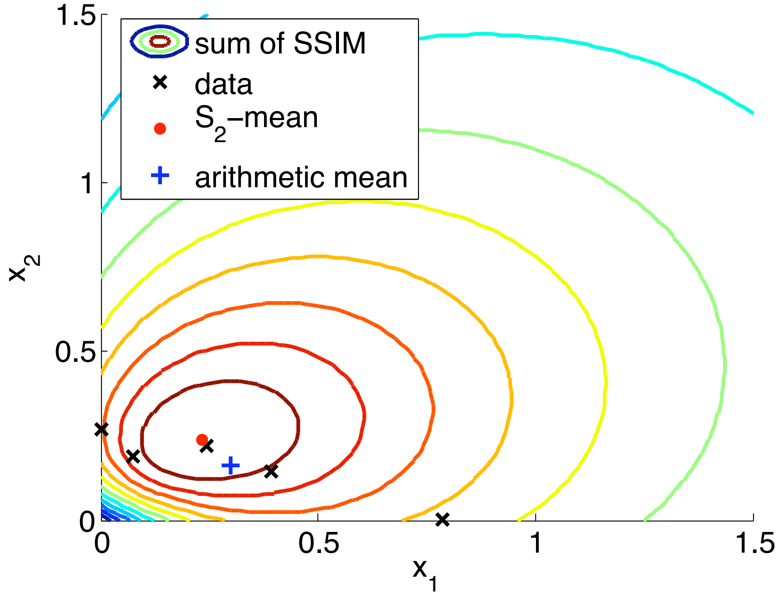


Figure 5.6: Comparison between the arithmetic mean and the S_2-mean (SSIM without the mean distortion component) of points in $\mathbf{R}^2$.

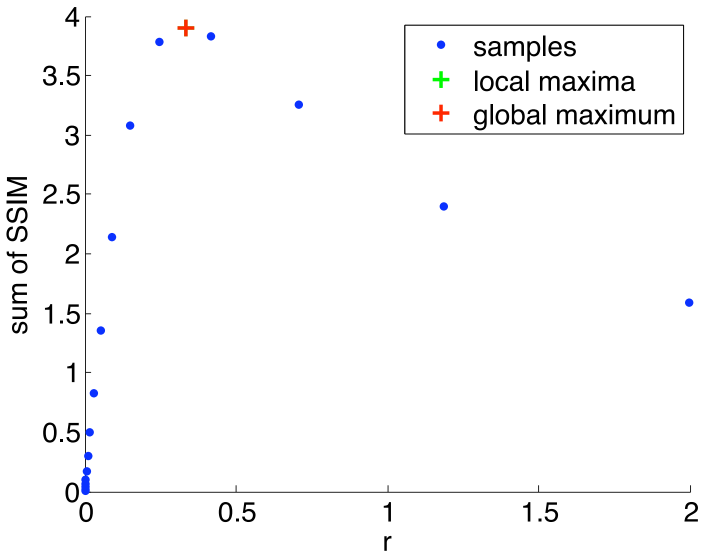


Figure 5.7: Global optimization of the norm r in the computation of the S_2-mean.

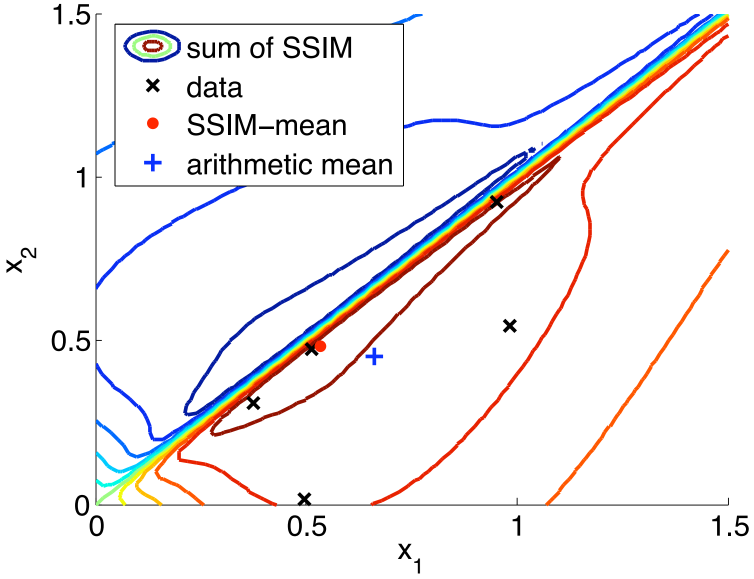


Figure 5.8: Comparison between the arithmetic mean and the SSIM-mean of positively correlated points in $\mathbf{R}^2$.

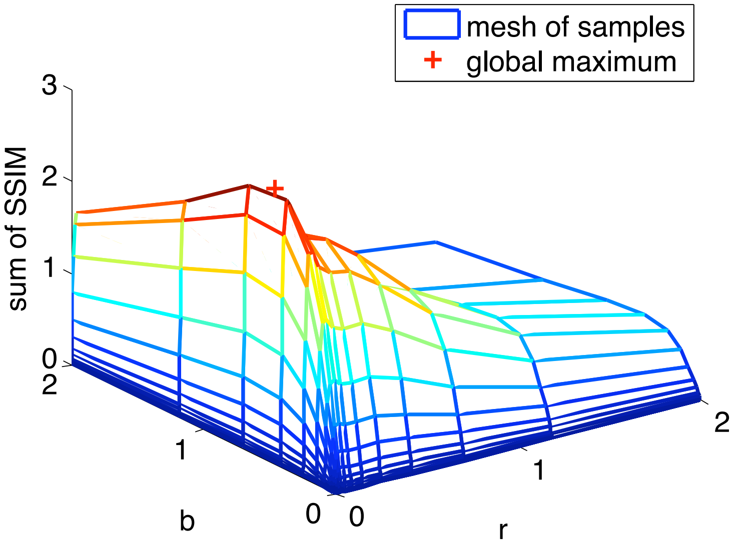


Figure 5.9: Global optimization of the norm r and the mean component b in the computation of the SSIM-mean for the points in Fig. 5.8.

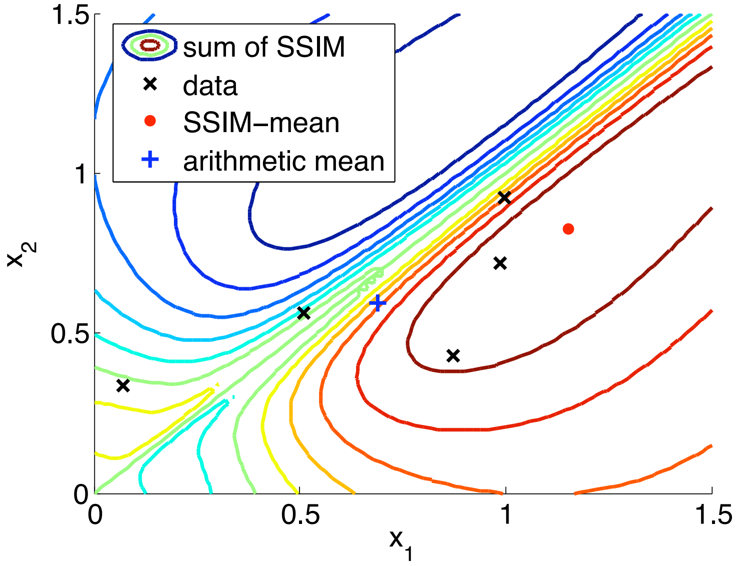


Figure 5.10: Comparison between the arithmetic mean and the SSIM-mean of points in $\mathbf{R}^2$ that do not have positively correlated zero-mean components.

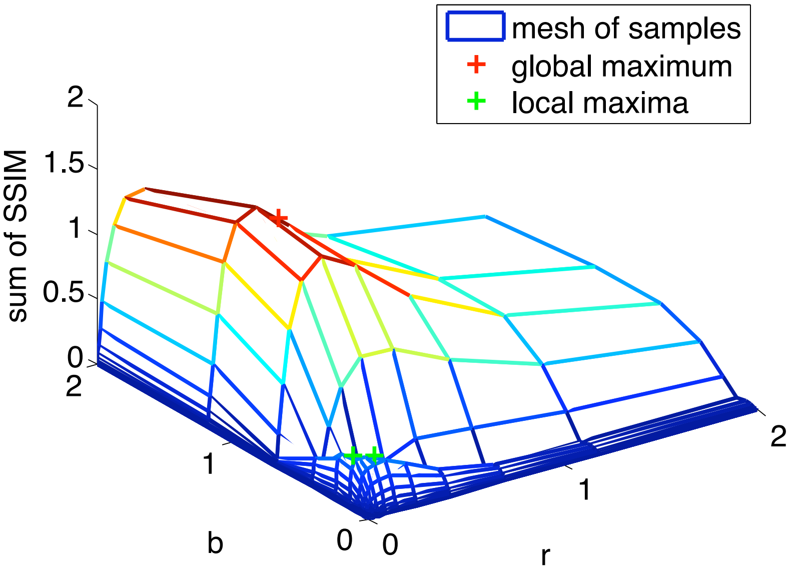


Figure 5.11: Global optimization of the norm r and the mean component b in the computation of the SSIM-mean for the points in Fig. 5.10.

Chapter 6

SSIM-based Approximation using Bases

One practical way of representing images is by decomposing them in terms of a basis. The basis is most commonly orthogonal, e.g. Discrete Cosine Transform (DCT) or Daubechies wavelets [33], but it can also be biorthogonal (e.g. Cohen-Daubechies-Fauvreau wavelets [29]) or redundant (e.g. Gabor filter).

The approximation problem consists of finding the basis and coefficients that best approximate an image economically. That is, we want an approximated image to look the closest to the original image and to be represented with the fewest number of basis blocks. Of course, the choice of basis blocks and coefficients depends on what we mean by "closest." The most popular choice is to employ the L^2-norm. Here, we restudy the problem using the Structural Similarity (SSIM) index.

After reviewing the classical L^2-based approximation problem, we will solve the local SSIM-based approximation problem for orthogonal, biorthogonal and redundant bases. The simplest case with zero-stability constants will be considered first and then the more general case that includes the stability constants will be treated last. As an application, we will outline how to use this sparse representation of images for compression, restoration and super-resolution of images.

6.1 L^2-based Approximation

It is helpful to review this standard result. We consider $\mathbf{x}$ to be a given signal and $\mathbf{y}$ to be an approximation to $\mathbf{x}$. We generally also consider $\mathbf{y}$ to be an element of a particular subset $A \subset \mathbf{R}^N$ (details to be given below) and look for solutions to the problem

$$\mathbf{y}_A = \arg \max_{\mathbf{y} \in A} \|\mathbf{x} - \mathbf{y}\|_2. \tag{6.1}$$

In the case $A = \mathbf{R}^N$, $\mathbf{y} = \mathbf{x}$ and $\|\mathbf{x} - \mathbf{y}\|_2 = \|\mathbf{x} - \mathbf{x}\|_2 = 0$.

6.1.1 Orthogonal Basis

We start with a set of (complete) orthogonal basis functions in $\mathbf{R}^N$ to be denoted as

$$\Psi \quad = \quad \{\psi_1, \psi_2, \cdots, \psi_N\}. \tag{6.2}$$

The L^2-based expansion of $\mathbf{x}$ in this basis is, of course,

$$\mathbf{x} = \sum_{k=1}^{N} f_k \psi_k, \qquad f_k = \langle \mathbf{x}, \psi_k \rangle, \quad 1 \leq k \leq N. \tag{6.3}$$

The expansions of the approximation $\mathbf{y}$ will be denoted as follows,

$$\mathbf{y} = \mathbf{y}(\mathbf{a}) = \sum_{k=1}^{N} a_k \psi_k, \tag{6.4}$$

where the notation $\mathbf{y}(\mathbf{a})$ acknowledges the dependence of the approximation on the coefficients a_k.

In the orthogonal case, the approximation spaces A in (6.1) will be the span of subsets of the set of basis functions $\{\psi_k\}_{k=1}^N$. At this point, we do not exactly specify which other ψ_k basis functions will be used but consider all possible subsets of $M < N$ basis functions:

$$A = \text{span}\{\psi_{\gamma(1)}, \psi_{\gamma(2)}, \cdots, \psi_{\gamma(M)}\}, \tag{6.5}$$

where $\gamma(i) \in \{1, 2, \cdots, N\}$ and $a_{\gamma(M+1)} = \cdots = a_{\gamma(N)} = 0$. Of course, we are interested in finding the *optimal* M-dimensional subset in the L^2 sense.

The well-known L^2-based optimal approximation is summarized in the following theorem:

Theorem 6.1.1. *For a given $\mathbf{x} \in \mathbf{R}^N$, the M coefficients a_k of the optimal L^2-based approximation $\mathbf{y} \in A$ to $\mathbf{x}$ are given by the M Fourier coefficients $f_k = \langle \mathbf{x}, \psi_k \rangle$ of greatest magnitude, i.e.*

$$a_{\gamma(k)} = \begin{cases} f_{\gamma(k)} = \langle \mathbf{x}, \psi_{\gamma(k)} \rangle, & 1 \leq k \leq M, \\ 0, & M+1 \leq k \leq N. \end{cases} \tag{6.6}$$

where $|f_{\gamma(1)}| \geq |f_{\gamma(2)}| \geq \ldots \geq |f_{\gamma(M)}| \geq |f_l|$ with $l \in \{1, 2, \cdots, N\} \setminus \{\gamma(1), \cdots, \gamma(M)\}$.

6.1.2 Linear Redundant Basis

A redundant or over-complete basis of $\mathbf{R}^N$ consists of $P \geq N$ column vectors $\Psi = \{\psi_k\}_{k=1}^P$ such that N of them are linearly independent. Given $\mathbf{x}$ in $\mathbf{R}^N$, we search for an approximation $\mathbf{y}$ of $\mathbf{x}$ with the help of the first $M < N$ linearly independent vectors of a redundant basis:

$$\mathbf{y} = \sum_{k=1}^{M} a_k \psi_k. \tag{6.7}$$

We seek for the coefficients $\mathbf{a} = [a_1, a_2, \ldots, a_M]$ that will minimize the L^2-error:

$$\begin{aligned} \|\mathbf{x} - \mathbf{y}\|_2^2 &= \langle \mathbf{x} - \sum_{j=1}^{M} a_j \psi_j, \mathbf{x} - \sum_{k=1}^{M} a_k \psi_k \rangle & (6.8) \\ &= \|\mathbf{x}\|_2^2 - 2 \sum_{k=1}^{M} a_k \langle \mathbf{x}, \psi_k \rangle + \sum_{j=1}^{M} \sum_{k=1}^{M} a_j a_k \langle \psi_j, \psi_k \rangle. & (6.9) \end{aligned}$$

Taking the partial derivative of the L^2-norm with respect to a_k and setting it to zero, we find

$$\frac{\partial}{\partial a_k} \|\mathbf{x} - \mathbf{y}\|_2^2 = -2\langle \mathbf{x}, \psi_k \rangle + 2 \sum_{j=1}^{M} a_j \langle \psi_j, \psi_k \rangle = 0, \quad 1 \leq k \leq M. \tag{6.10}$$

This simplifies to

$$\sum_{j=1}^{M} a_j \langle \psi_j, \psi_k \rangle \;=\; \langle \mathbf{x}, \psi_k \rangle, \quad 1 \leq k \leq M. \tag{6.11}$$

So to find the optimal coefficients $\mathbf{a}$, we need to solve a $M \times M$ linear system of equations $\Phi\mathbf{a} = \mathbf{f}$ with $\phi_{j,k} := \langle \psi_j, \psi_k \rangle$ and $\mathbf{f} = [f_1, f_2, \ldots, f_M]$ where $f_k = \langle \mathbf{x}, \psi_k \rangle$. In practice, $\mathbf{a}$ is found by multiplying the pseudo-inverse Ψ^+ of the dictionary matrix Ψ with $\mathbf{x}$:

$$\mathbf{a} \;=\; (\Psi^T \Psi)^{-1} \Psi^T \mathbf{x} \tag{6.12}$$

$$\;=\; \Psi^+ \mathbf{x}. \tag{6.13}$$

This pseudo-inverse can be computed from the singular value decomposition $U\Sigma V^T$ of Ψ:

$$\Psi^+ \;=\; V \Sigma^+ U^T, \tag{6.14}$$

where Σ^+ is the diagonal matrix whose positive elements are the reciprocal of the non-zero elements of Σ.

Biorthogonal Basis

A biorthogonal wavelet basis $\{\psi_k\}_{k=1}^N$ and its dual basis $\{\tilde{\psi}_k\}_{k=1}^N$ are defined such that

$$\langle \psi_j, \tilde{\psi}_k \rangle = \delta_{j,k}, \tag{6.15}$$

where $\delta_{j,k}$ is Kronecker's delta. We decompose $\mathbf{x}$ as

$$\mathbf{x} \;=\; \sum_{j=1}^{N} \langle \mathbf{x}, \tilde{\psi}_j \rangle \psi_j = \sum_{j=1}^{N} \langle \mathbf{x}, \psi_j \rangle \tilde{\psi}_j. \tag{6.16}$$

Notice that

$$\langle \mathbf{x}, \psi_k \rangle \;=\; \sum_{j=1}^{N} \langle \mathbf{x}, \tilde{\psi}_j \rangle \langle \psi_j, \psi_k \rangle \quad \text{for } 1 \leq k \leq N. \tag{6.17}$$

Comparing this last equation with (6.11), we find

$$a_k \;=\; \langle \mathbf{x}, \tilde{\psi}_k \rangle, \quad \text{for } 1 \leq k \leq M. \tag{6.18}$$

Thus, for biorthogonal basis it is not necessary to solve a linear system to find the optimal coefficients.

6.1.3 Non-Linear Approximation

For a non-linear approximation, we pick the M best vectors from the $P \geq N$ vectors $\Psi = \{\psi_k\}_{k=1}^{P}$. The problem can be written as

$$\arg\min \|\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{subject to } \|\mathbf{a}\|_0 = M, \tag{6.19}$$

where $\mathbf{x}$ is the signal to be approximated,

$$\mathbf{y} := \Psi\mathbf{a} \quad = \quad \sum_{k=1}^{P} a_k \psi_k \tag{6.20}$$

is the approximation and

$$\|\mathbf{a}\|_0 := \sum_{k=1}^{P} a_k^0 = \{\#k : |a_k| > 0\} \tag{6.21}$$

is the 0-pseudonorm. We follow the convention that $0^0 = 0$.

To solve the problem given in (6.19), we choose M basis from the dictionary of size P. And, we solve a linear system in a similar fashion as that in the linear case (see (6.11)) to determine the coefficients $\mathbf{a}$. We have $\frac{P!}{(P-M)!M!}$ possibilities, which grow exponentially with P. In fact, it has been shown that finding the approximation that minimizes $\|\mathbf{x}-\mathbf{y}\|_2^2$ is a NP-hard problem [34].

Since the optimal solution is not computationally tractable, we instead search for a sub-optimal solution with a greedy approach: the Matching Pursuit (MP) algorithm. Note that an alternative approach would be the Basis Pursuit [27]. This replaces the L^0-“norm” by L^1-norm in the optimization problem, thus transforming it into a linear programming problem. These two competing approaches have gained renewed interest recently with the advent of compressive sensing [19, 36].

Matching Pursuit

The MP algorithm from Mallat and Zhang [69] adds vectors one at a time until a M vectors approximation is found. For the first vector, we want to minimize

$$\|\mathbf{x} - a_{\gamma(1)}\psi_{\gamma(1)}\|_2^2 = \|\mathbf{x}\|_2^2 - 2a_{\gamma(1)}f_{\gamma(1)} + f_{\gamma(1)}^2. \tag{6.22}$$

By taking partial derivatives and setting them to zero, we find that the solution is exactly the same as the orthogonal case: We pick the index that maximizes $|f_k| = |\langle \mathbf{x}, \psi_k \rangle|$.

For the K-th vector, the L^2-error is

$$\begin{aligned} \|\mathbf{x} - \sum_{k=1}^{K} a_{\gamma(k)}\psi_{\gamma(k)}\|_2^2 &= \|\mathbf{x}\|_2^2 + \sum_{k=1}^{K} a_{\gamma(k)}^2 - 2\sum_{k=1}^{K} a_{\gamma(k)}f_{\gamma(k)} \\ &\quad +2\sum_{j=1}^{K}\sum_{k=1\,(j\neq k)}^{K} a_{\gamma(j)}a_{\gamma(k)}\phi_{\gamma(j),\gamma(k)}, \end{aligned} \tag{6.23}$$

where $f_k = \langle \mathbf{x}, \psi_k \rangle$ and $\phi_{j,k} = \langle \psi_j, \psi_k \rangle$. Taking the partial derivative of the L^2-error with respect to $a_{\gamma(K)}$ and setting it to zero, we find

$$a_{\gamma(K)} = f_{\gamma(K)} - \sum_{k=1}^{K-1} a_{\gamma(k)}\phi_{\gamma(k),\gamma(K)} \tag{6.24}$$

$$= \langle \mathbf{x}, \psi_{\gamma(K)} \rangle - \sum_{k=1}^{K-1} a_{\gamma(k)}\langle \psi_{\gamma(k)}, \psi_{\gamma(K)} \rangle. \tag{6.25}$$

Substituting $a_{\gamma(K)}$ back in the equation of the L^2-error, we conclude that the error will be minimized when

$$|\langle \mathbf{x} - \sum_{j=1}^{K-1} a_{\gamma(j)}\psi_{\gamma(j)}, \psi_k \rangle| \tag{6.26}$$

is maximized. In the case of an orthogonal basis, the matching pursuit algorithm coincides with the optimal algorithm who picks the M biggest basis coefficients.

Orthogonal Matching Pursuit

The Orthogonal Matching Pursuit (OMP) [76] combines the MP with a Gram-Schmidt procedure to obtain an orthogonal basis.

Given a linearly independent basis $\{\psi_1, \psi_2, \dots, \psi_N\}$ of $\mathbf{R}^N$, the Gram-Schmidt procedure successively projects the basis to an orthogonal subspace

$$\begin{aligned} G_1 &= \psi_1 && (6.27)\\ G_2 &= \psi_2 - \frac{\langle G_1, \psi_2\rangle}{\langle G_1, G_1\rangle} G_1 && (6.28)\\ &\vdots \\ G_N &= \psi_N - \sum_{j=1}^{N-1} \frac{\langle G_j, \psi_N\rangle}{\langle G_j, G_j\rangle} G_j. && (6.29) \end{aligned}$$

The basis is then normalized with $g_k = G_k / \|G_k\|_2$.

The OMP algorithm thus alternates between finding the best matching vector and the orthonormalization process. The trade-off is a convergence with a finite number of iterations against an extra computational cost for the orthonormalization.

For the numerical implementation, the naive approach to compute an orthonormal basis is well known to be unstable. A correct way to perform the OMP is with the following algorithm [92]:

```
1. Initialize: $I = \varnothing, \mathbf{r} := \mathbf{x}$ and $\mathbf{a} := \mathbf{0}$.
2. While $\|\mathbf{r}\|^2 > T$, do
3.     $k^* := \arg\max_k |\langle \psi_k, \mathbf{r}\rangle|$;
4.     Add $k^*$ to the set of indices $I$;
5.     $\mathbf{a}_I := \Psi_I^+ \mathbf{x}$;
6.     $\mathbf{r} := \mathbf{x} - \Psi_I \mathbf{a}_I$;
7. end while.
```

Here, $\mathbf{a}_I$ and Ψ_I represent the restriction of, respectively, $\mathbf{a}$ and Ψ to the elements or columns of indices I. Note that since the pseudo-inverse of Ψ has to be computed for incrementally larger matrices, there are ways to make computations more efficient with the help of Cholesky factorization (see [92]).

It is not immediately clear that this algorithm really performs the OMP. To see this, note that since $\mathbf{g} = [g_1, g_2, \ldots, g_K]$ is an orthonormal basis, the change of basis

$$\sum_{j=1}^{K} a_j \psi_j \quad = \quad \sum_{j=1}^{K} \langle \mathbf{x}, g_j \rangle g_j \tag{6.30}$$

$$\tag{6.31}$$

is performed via

$$\begin{aligned} \mathbf{a} &= (\Psi^T \Psi)^{-1} \Psi^T \mathbf{g}\mathbf{g}^T \mathbf{x} && (6.32) \\ &= \Psi^{+} \mathbf{x}. && (6.33) \end{aligned}$$

Thus, the residual $\mathbf{r}$ can be computed from the original basis Ψ and the orthonormalization is hidden in the computation of the coefficients $\mathbf{a}$.

6.2 SSIM-based Approximation

The problem now can be stated as follows:

Given $\mathbf{x}$ and Ψ, find the $\mathbf{y} = \Psi\mathbf{c}$ that maximizes $S := \mathrm{SSIM}(\mathbf{x}, \mathbf{y})$.

This problem was first solved for the orthogonal case in [12] before being generalized to the redundant case in [86] and [84].

6.2.1 Linear Approximation

In a linear approximation, the choice of dictionary vectors $\Psi = (\psi_1, \psi_2, \ldots, \psi_M)$ is already fixed and we only need to find the coefficients $\mathbf{c} = (c_1, c_2, \ldots, c_M)$ that maximizes the SSIM. To do that, we search for the stationary points of the partial derivatives of SSIM with respect to c_k. First, we write the mean, the variance and the covariance of $\mathbf{y}$ in terms of $\mathbf{c}$:

$$\begin{aligned}
\bar{\mathbf{y}} &= \langle \sum_{k=1}^{M} c_k \psi_k \rangle = \sum_{k=1}^{M} c_k \langle \psi_k \rangle, && (6.34)\\
(N-1)s_{\mathbf{y}}^2 &= \langle \mathbf{y}, \mathbf{y} \rangle - N \langle \mathbf{y} \rangle^2 \\
&= \langle \sum_{j=1}^{M} c_j \psi_j, \sum_{k=1}^{M} c_k \psi_k \rangle - N \bar{\mathbf{y}}^2 && (6.35)\\
&= \sum_{j=1}^{M} \sum_{k=1}^{M} c_j c_k \langle \psi_j, \psi_k \rangle - N \bar{\mathbf{y}}^2, \\
(N-1)s_{\mathbf{xy}} &= \langle \mathbf{x}, \mathbf{y} \rangle - N \langle \mathbf{x} \rangle \langle \mathbf{y} \rangle \\
&= \langle \mathbf{x}, \sum_{k=1}^{M} c_k \psi_k \rangle - N \bar{\mathbf{x}} \bar{\mathbf{y}} \\
&= \sum_{k=1}^{M} c_k \langle \mathbf{x}, \psi_k \rangle - N \bar{\mathbf{x}} \bar{\mathbf{y}}. && (6.36)
\end{aligned}$$

Next, we find the partial derivatives:

$$\begin{aligned}
\frac{\partial \bar{\mathbf{y}}}{\partial c_k} &= \langle \psi_k \rangle; && (6.37)\\
(N-1)\frac{\partial s_{\mathbf{y}}^2}{\partial c_k} &= 2 \sum_{j=1}^{M} c_j \langle \psi_j, \psi_k \rangle - 2N \bar{\mathbf{y}} \langle \psi_k \rangle; && (6.38)\\
(N-1)\frac{\partial s_{\mathbf{xy}}}{\partial c_k} &= \langle \mathbf{x}, \psi_k \rangle - N \bar{\mathbf{x}} \langle \psi_k \rangle. && (6.39)
\end{aligned}$$

The SSIM can be written as

$$\log S = \log(2\bar{\mathbf{x}}\bar{\mathbf{y}} + \epsilon_1) - \log(\bar{\mathbf{x}}^2 + \bar{\mathbf{y}}^2 + \epsilon_1) + \log(2s_{\mathbf{xy}} + \epsilon_2) - \log(s_{\mathbf{x}}^2 + s_{\mathbf{y}}^2 + \epsilon_2). \tag{6.40}$$

So for all $1 \leq k \leq M$,

$$\frac{\partial S}{\partial c_k} = S \left[\frac{2\bar{\mathbf{x}}\langle \psi_k \rangle}{2\bar{\mathbf{x}}\bar{\mathbf{y}} + \epsilon_1} - \frac{2\bar{\mathbf{y}}\langle \psi_k \rangle}{\bar{\mathbf{x}}^2 + \bar{\mathbf{y}}^2 + \epsilon_1} + \frac{2\langle \mathbf{x}, \psi_k \rangle - 2N\bar{x}\langle \psi_k \rangle}{(N-1)(2s_{\mathbf{xy}} + \epsilon_2)} - \frac{2\sum_{j=1}^{M} c_j \langle \psi_j, \psi_k \rangle - 2N\bar{\mathbf{y}}\langle \psi_k \rangle}{(N-1)(s_{\mathbf{x}}^2 + s_{\mathbf{y}}^2 + \epsilon_2)} \right]. \tag{6.41}$$

Solution for Oscillatory Basis

In a particular case where the basis is made of normalized oscillatory functions, we have

$$\langle\psi_k\rangle = 0 \text{ and } \langle\psi_k, \psi_k\rangle = 1 \quad \text{for } 1 \le k \le M. \tag{6.42}$$

It leads to

$$\begin{aligned} \bar{\mathbf{y}} &= 0, && (6.43)\\ (N-1)s_{\mathbf{y}}^2 &= \sum_{j=1}^{M}\sum_{k=1}^{M} c_j c_k \langle\psi_j, \psi_k\rangle && (6.44)\\ (N-1)s_{\mathbf{xy}} &= \sum_{k=1}^{M} c_k \langle\mathbf{x}, \psi_k\rangle. && (6.45) \end{aligned}$$

So the partial derivative in (6.41) reduces to

$$\frac{\partial S}{\partial c_k} = S\left[\frac{2\langle\mathbf{x}, \psi_k\rangle}{(N-1)(2s_{\mathbf{xy}} + \epsilon_2)} - \frac{2\sum_{j=1}^{M} c_j \langle\psi_j, \psi_k\rangle}{(N-1)(s_{\mathbf{x}}^2 + s_{\mathbf{y}}^2 + \epsilon_2)}\right]. \tag{6.46}$$

We now search for stationary points:

$$\frac{\partial S}{\partial c_k} = 0 \;\Rightarrow\; \frac{\langle\mathbf{x}, \psi_k\rangle}{\sum_{j=1}^{K} c_j \langle\psi_j, \psi_k\rangle} = \frac{2s_{\mathbf{xy}} + \epsilon_2}{s_{\mathbf{x}}^2 + s_{\mathbf{y}}^2 + \epsilon_2} =: \frac{1}{\alpha}, \quad \text{for } 1 \le k \le M. \tag{6.47}$$

We can rewrite this equation as

$$\sum_{j=1}^{M} c_j \langle\psi_j, \psi_k\rangle = \alpha\langle\mathbf{x}, \psi_k\rangle, \quad \text{for } 1 \le k \le M. \tag{6.48}$$

This equation is very similar to the one of optimal coefficients for the L^2-based approximation (6.11). In fact, since the equations (6.11) and (6.48) are identical up to a scaling factor and since the solution of the linear system is unique, we have

$$c_k = \alpha a_k. \tag{6.49}$$

Now, we seek to find an expression for α:

$$\begin{aligned}\alpha &= \frac{s_{\mathbf{x}}^2 + s_{\mathbf{y}}^2 + \epsilon_2}{2s_{\mathbf{xy}} + \epsilon_2} \\ &= \frac{s_{\mathbf{x}}^2 + \frac{1}{N-1}\sum_{j=1}^{M}\sum_{k=1}^{M} c_j c_k \langle \psi_j, \psi_k \rangle + \epsilon_2}{\frac{2}{N-1}\sum_{k=1}^{M} c_k \langle \mathbf{x}, \psi_k \rangle + \epsilon_2}. \qquad (6.50)\end{aligned}$$

Substituting the equation for c_k (6.49) into the equation for α (6.50), we obtain

$$\alpha = \frac{\alpha^2 A + B}{\alpha C + D}, \qquad (6.51)$$

where

$$\begin{aligned} A &= \frac{1}{N-1}\sum_{j=1}^{M}\sum_{k=1}^{M} a_j a_k \langle \psi_j, \psi_k \rangle, & (6.52) \\ B &= s_{\mathbf{x}}^2 + \epsilon_2, & (6.53) \\ C &= \frac{2}{N-1}\sum_{k=1}^{M} a_k \langle \mathbf{x}, \psi_k \rangle, & (6.54) \\ D &= \epsilon_2. & (6.55)\end{aligned}$$

Isolating α gives a quadratic equation

$$\alpha^2(C - A) + \alpha D - B = 0. \qquad (6.56)$$

Solving for α, we find

$$\begin{aligned}\alpha &= \frac{-D \pm \sqrt{D^2 + 4(C-A)B}}{2(C-A)} & (6.57) \\ &= \frac{-\epsilon_2 \pm \sqrt{\epsilon_2^2 + 2C(s_{\mathbf{x}}^2 + \epsilon_2)}}{C}. & (6.58)\end{aligned}$$

Note that $C - A = C/2 = A$, since the a_k's are found by solving the linear system (6.11).

Flat Approximation Case

A flat approximation $\psi_0 \equiv 1$ is added to the oscillatory basis. In this case,

$$\bar{\mathbf{y}} \;=\; c_0. \tag{6.59}$$

The coefficient c_0 is the stationary point of (6.41):

$$\begin{aligned}\frac{\partial S}{\partial c_0} &= S\left[\frac{2\bar{x}\langle\psi_0\rangle}{2\bar{x}c_0+\epsilon_1} - \frac{2c_0\langle\psi_0\rangle}{\bar{x}^2+c_0^2+\epsilon_1} + \frac{2\langle\mathbf{x},\psi_0\rangle - 2N\bar{x}\langle\psi_0\rangle}{(N-1)(2s_{x,y}+\epsilon_2)} - \frac{2\sum_{j=0}^{M} c_j\langle\psi_j,\psi_0\rangle - 2Nc_0\langle\psi_0\rangle}{(N-1)(s_x^2+s_y^2+\epsilon_2)}\right] \\ &= S\left[\frac{2\bar{x}}{2\bar{x}c_0+\epsilon_1} - \frac{2c_0}{\bar{x}^2+c_0^2+\epsilon_1}\right]. \end{aligned} \tag{6.60}$$

Solving for the stationary point leads to the following quadratic equation in c_0:

$$c_0^2\bar{x} + \epsilon_1 c_0 - \bar{x}(\bar{x}^2+\epsilon_1) = 0. \tag{6.61}$$

Its solution is

$$c_0 = \frac{-\epsilon_1 \pm \sqrt{\epsilon_1^2 + 4\bar{x}^2(\bar{x}^2+\epsilon_1)}}{2\bar{x}}. \tag{6.62}$$

We take the positive branch to maximize the SSIM index, which is simply $c_0 = \bar{x}$, as expected. The other coefficients are found as in the oscillatory basis case.

Orthogonal Basis

In the orthogonal case, the constants in the equation for α (6.58) simplify to

$$\alpha \;=\; \frac{-\epsilon_2 \pm \sqrt{\epsilon_2^2 + (\frac{4}{N-1}\sum_{k=1}^{M}\langle\mathbf{x},\psi_k\rangle^2)(s_{\mathbf{x}}^2+\epsilon_2)}}{\frac{2}{N-1}\sum_{k=1}^{M}\langle\mathbf{x},\psi_k\rangle^2}. \tag{6.63}$$

The SSIM index is maximized with the positive branch. If $\epsilon_2 = 0$, then

$$\alpha \;=\; \frac{s_{\mathbf{x}}}{\sqrt{\frac{1}{N-1}\sum_{k=1}^{M}\langle\mathbf{x},\psi_k\rangle^2}}, \tag{6.64}$$

thus the coefficients are adjusted in order to preserve the variance of the original signal.

Biorthogonal basis

The SSIM-optimal decomposition with a bi-orthogonal basis is a particular case of the general solution with an oscillatory basis. From (6.18), (6.49) and (6.58), we have

$$c_k \quad = \quad \alpha \langle \mathbf{x}, \tilde{\psi}_k \rangle, \tag{6.65}$$

with

$$\alpha \quad = \quad \frac{-\epsilon_2 + \sqrt{\epsilon_2^2 + \frac{4}{N-1} \sum_{k=1}^{M} \langle \mathbf{x}, \tilde{\psi}_k \rangle \langle \mathbf{x}, \psi_k \rangle (s_{\mathbf{x}}^2 + \epsilon_2)}}{\frac{2}{N-1} \sum_{k=1}^{M} \langle \mathbf{x}, \tilde{\psi}_k \rangle \langle \mathbf{x}, \psi_k \rangle}. \tag{6.66}$$

6.2.2 Non-Linear Approximation

Similar to the L^2-case, the problem is to find, given a dictionary $\Psi \in \mathbf{R}^{P \times N}$ and a signal $\mathbf{x} \in \mathbf{R}^N$, the coefficients $\mathbf{c} \in \mathbf{R}^P$ with $\|\mathbf{c}\|_0 = M < N$ such that

$$\mathrm{SSIM}(\mathbf{x}, \Psi \mathbf{c}) \tag{6.67}$$

is maximized.

SSIM-based matching pursuit

We already know how to pick the best coefficients given a set of vectors (see Section 6.2.1). We now pick the vectors in a greedy way.

First step. We want to find ψ_{γ_0} and c_{γ_0} that will maximize $S(\mathbf{x}, c_{\gamma_0}\psi_{\gamma_0})$. The SSIM is

$$S(\mathbf{x}, c_{\gamma_0}\psi_{\gamma_0}) = \frac{2c_{\gamma_0}\langle \mathbf{x}, \psi_{\gamma_0} \rangle + \epsilon_2(N-1)}{\|x\|^2 + c_{\gamma_0}^2 + \epsilon_2(N-1)}. \tag{6.68}$$

For any fixed c_k, the absolute value of SSIM will be maximized when $|\langle \mathbf{x}, \psi_k \rangle|$ is maximized as well. We thus pick

$$\gamma_0 \quad = \quad \arg \max_{1 \le k \le P} |\langle \mathbf{x}, \psi_k \rangle| \tag{6.69}$$

and

$$c_{\gamma_0} \quad = \quad \alpha \langle \mathbf{x}, \psi_{\gamma_0} \rangle. \tag{6.70}$$

In general, we want to find ψ_{γ_K} and c_{γ_K} that will maximize

$$S(\mathbf{x}, \sum_{k=0}^{K-1} c_{\gamma_k}\psi_{\gamma_k} + c_{\gamma_K}\psi_{\gamma_K}). \tag{6.71}$$

For every choice of ψ_{γ_K}, we would need to find $\{a_{\gamma_k}\}_{0\leq k\leq K}$, i.e. we have to solve a $K \times K$ linear system of equations and compute the SSIM with $c_{\gamma_k} = \alpha a_{\gamma_k}$, then pick the basis ψ_{γ_K} that gives the maximum value. Although this procedure makes sense in theory, in practice it is intractable given that a potentially large linear system has to be solved for every possible basis of the dictionary and at every iteration of the greedy algorithm.

SSIM-based Orthogonal Matching Pursuit

According to the MP algorithm, the choice of the first basis that maximizes the SSIM index is the same as that of the optimal L^2-basis. Indeed, (6.68) is maximized when $|\langle \mathbf{x}, \psi_k\rangle|$ is maximized.

For the choice of the K-th basis, we seek to maximize

$$\begin{aligned} S(\mathbf{x}, \sum_{j=1}^{K-1} a_{\gamma_j}\psi_{\gamma_j} + a_{\gamma_K}\psi_{\gamma_K}) &= S(\mathbf{x}, \sum_{j=1}^{M-1} c_{\gamma_j} g_{\gamma_j} + c_{\gamma_M} g_{\gamma_M}) & (6.72)\\ &= \frac{2\sum_{j=1}^{K-1} c_{\gamma_j}\langle \mathbf{x}, g_{\gamma_j}\rangle + c_{\gamma_K}\langle \mathbf{x}, g_{\gamma_K}\rangle + \epsilon_2}{\|\mathbf{x}\|^2 + \sum_{j=1}^{K} c_{\gamma_j}^2 + \epsilon_2}. & (6.73)\end{aligned}$$

The choice of basis that will maximize the SSIM index is

$$\gamma_K = \arg\max_{1\leq k\leq P} |\langle \mathbf{x}, g_k\rangle|. \tag{6.74}$$

Note that

$$\begin{aligned} \langle \mathbf{x}, g_k\rangle &= \langle \mathbf{x}, \psi_k - \sum_{j=1}^{K-1}\langle g_j, \psi_k\rangle g_j\rangle & (6.75)\\ &= \langle \mathbf{x} - \sum_{j=1}^{K-1}\langle \mathbf{x}, g_j\rangle g_j, \psi_k\rangle & (6.76)\\ &= \langle \mathbf{r}, \psi_k\rangle. & (6.77)\end{aligned}$$

Thus, the optimal basis for the SSIM-based and the L^2-based algorithms are exactly the same. Indeed, the SSIM-based coefficients will be simply a scaling of the L^2-based coefficients. The difference will be in the stopping criterion: the SSIM-OMP stopping criterion will obviously depend on the SSIM index instead of the L^2-error.

6.3 Applications

6.3.1 Image and Video Coding

Both JPEG and JPEG2000 image compression standards are based on the decomposition of images into a basis. For JPEG compression, an orthogonal cosine basis is taken whereas JPEG2000 standard recommends a biorthogonal wavelet transform. Moreover, the H.264 standard of video coding also incorporates the representation of the residual video frames after motion compensation with a cosine basis.

A general introduction to data compression can be found in [97]. The reader is also referred to the author Master's Thesis [8]. The general framework is as follows. In a first step, the image is decomposed into a basis. The basis coefficients are then quantized to a finite number of possible values and it is these quantized coefficients that are losslessly encoded with an entropy coding scheme. The image decompression consists in the decoding, the dequantization and the image reconstruction with an inverse transform. Note that ignoring the round-off errors, the loss of information only occurs in the quantization step. For video compression, an extra step consists in computing and encoding motion vector between frames. A JPEG-like compression is then performed on the first frame of a group of frames as well as on the residual frames.

From the invariance property under orthogonal transforms (3.1.1), it is possible to optimize the SSIM index in the frequency domain for perceptual coding. However, one constraint is that the local SSIM coefficients can only be measured for non-overlapping blocks. Thus the SSIM-optimal coding will not prevent the "blocking effect", i.e. the apparition of image artifacts at the interface of two independently coded blocks. To remedy this problem one should combine locally optimal blocks into a globally approximately optimal image with an aggregation process.

For biorthogonal wavelet-based compression, since the transform is not orthogonal, we cannot optimize directly in the frequency domain. Nevertheless, it is generally assumed that the biorthogonal wavelets possess Riesz stability

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \sum_k |\langle \mathbf{x}, \psi_k \rangle|^2 \leq (1+\delta)\|\mathbf{x}\|_2^2, \tag{6.78}$$

where $\delta > 0$. We thus have an approximate preservation of energy. Moreover, under the simplifying assumption that the signal is independent and identically distributed, one can show that

$$\begin{aligned} \mathbf{E}[|\langle \mathbf{x}, \psi_k \rangle|^2] &= \mathbf{E}[\sum_j \sum_l \psi_k(j)\psi_k(l)\mathbf{x}(j)\mathbf{x}(l)] & (6.79) \\ &= \sum_j \psi_k^2(j) E[\mathbf{x}^2(j)]] & (6.80) \\ &= \langle \psi_k, \psi_k \rangle \mathbf{E}[\|\mathbf{x}\|_2^2]. & (6.81) \end{aligned}$$

In [89], Richter and Kim proposed an optimal Multi-Scale SSIM JPEG2000 encoder on non-overlapping blocks. In fact, a rate allocation scheme is performed on the linear approximation of the SSIM index. Wang et al. [116] suggested another bit allocation scheme based on an iterative refinement of bitplanes trimming functions. In this case, the image is transformed back to the image domain and the full SSIM index is computed at each iteration. The strategy taken is then to allocate bits to the region with the lowest predicted quality. In comparison to the two previous propositions, the optimal decomposition found in Section 6.2 would allow to devise a direct rate allocation scheme according to the exact SSIM index (over non-overlapping blocks).

An important problem in image and video coding is to find the rate-distortion curve. This curve describes the maximum rate that an algorithm can compress a signal for a given distortion level. Channappayya et al. [23] derived bounds for the rate-distortion curve of the SSIM index for quantized images. Several authors (see [112] and references therein) also have developed algorithms to estimate rate-distortion curve of the SSIM index for video coding. Many simplifying assumptions are generally made and a numerical solution rather than an analytical solution is sought. From the tools developed in this chapter, we are still a far cry from a complete rate-distortion theory. Nevertheless, an analytical solution can be computed for some simple approximation problems.

6.3.2 Image Denoising by Sparse Representation

Best K-terms approximation can be included in an image denoising algorithm. We present the work of Elad and Aharon [40] and present the extension done in collaboration with A. Rehman and M. Rostami [86, 84] to accomodate for the SSIM index.

A major idea in image representation is to decompose image into a sparse basis. That is, images are modeled as being well approximated by the sum of only few atom elements. Traditionally, several bases have been studied: Fourier, Gabor, separable wavelets, curvelets and all the family of X-lets. Some success have been obtained with hard or soft thresholding techniques [35] or coring methods [103], but it is possible to push further the sparse representation by learning the dictionary of bases from images themselves. There are two possibilities: either one trains the dictionary from a set of clean natural images or one learns adaptively the basis from the noisy image. We will adopt the second option since this is the method that generated best results in [40].

We represent blocks in a vectorial form with the lexicographical order. Consider blocks of length n and assume that the dictionary $\boldsymbol{\Psi} \in \mathbf{R}^{n\times K}$ with $K > n$ is given. Once the dictionary is fixed, the L^2 sparse approximation problem for an image block $\mathbf{x}_i$ is

$$\hat{\mathbf{c}}_i = \arg\min_{\mathbf{c}_i} \|\mathbf{c}_i\|_0 \text{ subject to } \|\Psi\mathbf{c}_i - \mathbf{x}_i\|_2^2 \leq T, \tag{6.82}$$

where $\mathbf{c}_i$ is the coefficient vector of length m associated with the i-th image block, $\|\cdot\|_0$ is the counting pseudo-norm and T is a stopping criterion. The problem of sparse approximation can be approximatively solved either by matching or basis pursuit as described in Section 6.1.3. The optimal coefficients are found for every sliding block of an image. The global solution is then given by the minimizer of

$$\sum_{i=1}^{M} \|\boldsymbol{\Psi}\hat{\mathbf{c}}_i - \mathbf{x}_i\|_2^2 + \lambda\|\mathbf{x} - \mathbf{y}\|_2^2, \tag{6.83}$$

where λ controls the weight between fitting to the noisy image and respecting the sparsity constraint. An analytical solution of this problem is given by

$$\mathbf{x} \;=\; \left(\sum_i \mathbf{R}_i^T\mathbf{R}_i + \lambda I\right)^{-1}\left(\sum_i \mathbf{R}_i^T\boldsymbol{\Psi}\mathbf{c}_i + \lambda\mathbf{y}\right), \tag{6.84}$$

where $\mathbf{R}_i$ is the $n \times N$ matrix extracting the i-th block from $\mathbf{x}$.

The dictionary learning procedure described by Aharon and Elad [1] is called K-SVD, since it is seen as a generalization of the K-means clustering algorithm [67]. The goal of the algorithm is to find the dictionary Ψ and the coefficients $\mathbf{c}$ that minimize the Frobenius norm of the error between a training set $\mathbf{y}$ and its sparse representation $\Psi\mathbf{c}$:

$$\min_{\Psi,\mathbf{c}} \|\mathbf{y} - \boldsymbol{\Psi}\mathbf{c}\|_F^2 \text{ subject to } \|\mathbf{c}_i\|_0 \leq T \text{ for } 1 \leq i \leq K. \tag{6.85}$$

The algorithm starts with M examples $\mathbf{y} = \{\mathbf{y}_i\}_{i=1}^K$ forming the training set and an initial dictionary $\boldsymbol{\Psi}^{(0)} \in \mathbf{R}^{n \times K}$. In a first stage, a representation vector is found according to (6.82) for every $1 \leq i \leq K$. Then, each column of the dictionary and its corresponding coefficients are updated one at the time in order to minimize the Frobenius norm of the error of the sparse representation via a singular value decomposition. This process is iterated until convergence.

The SSIM-based sparse representation with a learned dictionary replaces the L^2-norm with the SSIM index in the local sparse approximation. In [84], the L^2-error is also replaced by the SSIM index in the global reconstruction problem (6.83). The SSIM-inspired algorithm is thus separated in three steps[1]:

1. Find

$$\hat{\mathbf{c}}_i = \arg\min_{\mathbf{c}_i} \|\mathbf{c}_i\|_0 \text{ subject to } SSIM(\boldsymbol{\Psi}\mathbf{c}_i, \mathbf{x}_i) \geq T \tag{6.86}$$

   with orthogonal matching pursuit.

2. Aggregate the local patches by averaging:

$$\mathbf{w} = \left(\sum_i \mathbf{R}_i^T \mathbf{R}_i\right)^{-1} \left(\sum_i \mathbf{R}_i^T \boldsymbol{\Psi}\hat{\mathbf{c}}_i\right). \tag{6.87}$$

3. Numerically optimize for the global solution

$$\arg\max_{\mathbf{x}} \{SSIM(\mathbf{w}, \mathbf{x}) + \lambda SSIM(\mathbf{x}, \mathbf{y})\} \tag{6.88}$$

   by gradient descent.

In [84], the SSIM-sparse representation algorithm was modified following [122] to allow

[1] This idea as well as the implementation of the algorithm was first developed by A. Rehman.

for super-resolution[2]. The global solution (6.88) is replaced with

$$\arg\max_{\mathbf{x}} \left\{ SSIM(\mathbf{w},\mathbf{x}) + \lambda SSIM(\mathbf{HDx},\mathbf{y}) \right\}, \tag{6.89}$$

where $\mathbf{H}$ is a blurring operator and $\mathbf{D}$ is a downsampling operator. In this case, two dictionaries need to be learned: $\boldsymbol{\Psi}_h$, the dictionary on high-resolution patches and $\boldsymbol{\Psi}_l$, the dictionary on low-resolution patches. Contrary to the image denoising problem, the dictionaries cannot be trained adaptively, but have to be trained from a set of examples. The SSIM-inspired super-resolution algorithm is then described as follows:

1. Train the dictionaries $\boldsymbol{\Psi}_h$ and $\boldsymbol{\Psi}_l$ as in [122].
2. Find the vector of coefficients $\mathbf{c}_i$ for all patches of the low-resolution image $\mathbf{y}$.
3. Reconstruct high-resolution patches with $\mathbf{x}_i = \boldsymbol{\Psi}_h\mathbf{c}_i$.
4. Aggregate the high-resolution patches by averaging to obtain $\mathbf{w}$.
5. Regularize the image by maximizing (6.89).

Note that the stopping criterion proposed in [84] is in fact a hidden L^2 constraint. Indeed,

$$SSIM(\mathbf{y},\boldsymbol{\Psi}\mathbf{c}) \geq T_{SSIM} \quad = \quad 1 - \frac{T_{MSE}}{\|\mathbf{y}\|_2^2 + \|\boldsymbol{\Psi}\mathbf{c}\|_2^2 + \epsilon_2}, \tag{6.90}$$

if and only if

$$\|\mathbf{y} - \boldsymbol{\Psi}\mathbf{c}\|_2^2 \quad \leq \quad T_{MSE}. \tag{6.91}$$

The constant T_{MSE} is set to $(1.15\sigma)^2$, where σ^2 is the variance of the noise. In fact, this constraint makes sense, since the goal is to approximate the original image, which is known to satisfy (6.91) with high probability, and not the noisy image. A fixed bound for the SSIM of the approximation would instead imply that the algorithm optimizes the structural similarity between the basis approximation and the noisy image. Besides taking the L^2 bound, other options could be to pre-process the noisy image to obtain a first estimate of the clean image. It is also suggested, at the cost of extra computation time, to apply the algorithm iteratively. A third possibility is to estimate the SSIM index from the noisy and the residual image as it will be discussed in Chapter 8.

[2] These ideas were developed by M. Rostami

Chapter 7

A Study of Self-Similarity of Images

The effectiveness of a good number of nonlocal image processing methods, including nonlocal means denoising [16], restoration [31, 125], compression [42], super-resolution [38, 41, 44], fractal image coding [71, 65, 47] and adaptive dictionary learning [40], is due to how well pixel-blocks of an image can, in some way, be approximated by other blocks. This property of natural images may be viewed as a form of self-similarity.

In this chapter, we propose to revisit the L^2-based affine model of self-similarity of [2] by appropriately penalizing the image blocks with low variance. This is done naturally via the Structural Similarity (SSIM) index.

7.1 L^2-based Affine Model of Self-Similarity

In [2], a simple model of affine self-similarity which includes a number of nonlocal image processing methods as special cases was introduced. (It was analyzed further in [60].) An image I will be represented by an image function $u : X \to R_g$, where $R_g \subset \mathbf{R}$ denotes the greyscale range. Unless otherwise specified, we work with normalized images, i.e. $R_g = [0, 1]$. The support X of an image function u is assumed to be an $L_1 \times L_2$-pixel array. Let $\mathscr{R}$ be a set of $n \times n$-pixel subblocks R_i, $1 \leq i \leq N_R$ such that $X = \cup_i R_i$, i.e. $\mathscr{R}$ forms a covering of X. We let $u(R_i)$ denote the portion of u that is supported on R_i.

We examine how well an image block $u(R_i)$ is approximated by other image blocks $u(R_j)$, $j \neq i$. Let us consider a block $u(R_i)$ being approximated as the range block and a block $u(R_j)$, $j \neq i$, approximating it as the domain block. In order to distinguish the roles of these blocks, we shall denote the domain blocks as $u(D_j)$ with the understanding that $D_j = R_j$. For two pixel blocks R_i and D_j, the approximation of an image range block $u(R_i)$ by a domain block $u(D_j)$ may be written in the following general form,

$$u(R_i) \approx \alpha_{ij} u(D_j) + \beta_{ij}, \quad i \neq j. \tag{7.1}$$

The error associated with the approximation in (7.1) is defined as

$$\Delta_{ij} = \min_{\alpha,\beta\in\Pi} \|u(R_i) - \alpha u(D_j) - \beta\|, \qquad i \neq j, \tag{7.2}$$

where $\|\cdot\|$ denotes the $L^2(X)$ norm (or RMSE) and where $\Pi \subset \mathbf{R}^2$ denotes the (α, β) parameter space appropriate for each case.

The affine self-similarity model is comprised of four cases. The optimal parameters and associated errors for each case will be given. In what follows, we denote $\mathbf{x} = u(R_i)$, $\mathbf{y} = u(D_j)$ and $N = n^2$.

Case 1: Purely translational. This is the strictest view of similarity: Two image subblocks $u(R_i)$ and $u(D_j)$ are considered to be "close," $u(R_i) \approx u(D_j)$, if the L^2 distance $\|u(R_i) - u(D_j)\|$ is small. This is the basis of non-local means denoising. There is no optimization here: $\alpha_{ij} = 1$, $\beta_{ij} = 0$ and the approximation error is simply

$$\Delta_{ij}^{(Case\ 1)} = \|\mathbf{x} - \mathbf{y}\| = N^{-1/2}\sqrt{(N-1)[s_{\mathbf{x}}^2 + s_{\mathbf{y}}^2 - 2s_{\mathbf{xy}}] + [\bar{\mathbf{x}} - \bar{\mathbf{y}}]^2}. \tag{7.3}$$

Case 2: Translational + greyscale shift. This is a slightly relaxed definition of similarity. Two image subblocks are considered similar if they are close up to a greyscale shift, i.e. $u(R_i) \approx u(D_j) + \beta$. This simple adjustment can improve the non-local means denoising method since more blocks are available in the averaging process. In this case, $\alpha_{ij} = 1$ and we optimize over β_{ij}:

$$\beta_{ij} = \bar{\mathbf{x}} - \bar{\mathbf{y}}, \qquad \Delta_{ij}^{(Case\ 2)} = \left[\frac{N-1}{N}\right]^{1/2} [s_{\mathbf{x}}^2 + s_{\mathbf{y}}^2 - 2s_{\mathbf{xy}}]^{1/2}. \tag{7.4}$$

Case 3: Affine transformation. A further relaxation is afforded by allowing affine greyscale transformations, i.e. $u(R_i) \approx \alpha u(D_j) + \beta$. This method has been employed in vector quantization [42]. We optimize over α and β.

$$\alpha_{ij} = \frac{s_{\mathbf{xy}}}{s_{\mathbf{y}}^2}, \quad \beta_{ij} = \bar{\mathbf{x}} - \alpha_{ij}\bar{\mathbf{y}}, \qquad \Delta_{ij}^{(Case\ 3)} = \left[\frac{N-1}{N}\right]^{1/2} \left[s_{\mathbf{x}}^2 - \frac{s_{\mathbf{xy}}^2}{s_{\mathbf{y}}^2}\right]^{1/2}. \tag{7.5}$$

Case 4: Cross-scale affine transformation. $u(R_i) \approx \alpha u(w(D_j))+\beta$, where D_j is larger than R_i and where w is a contractive spatial transformation. This is the basis of fractal image coding. The optimization process and the error distribution for Case 4 are almost identical to those of Case 3. For this reason, this case will not be discussed further.

Note that in both Cases 2 and 3, the means of the range block and optimally transformed range block are equal, i.e. $\bar{\mathbf{x}} = \alpha\bar{\mathbf{y}} + \beta$.

Of particular interest in [2] were the distributions of L^2 errors denoted as $\Delta_{ij}^{(Case\ k)}$, in approximating range blocks $u(R_i)$ by all other domain blocks $u(D_j)$, $j \neq i$, for the cases $1 \leq k \leq 3$. In order to reduce the computational cost, we employ nonoverlapping subblocks. Normally, one could consider eight affine spatial transformations that map a square spatial domain block D_j to a square range block R_i. In our computations, however, unless otherwise specified, we shall consider only the identity transformation, i.e. zero rotation.

Figure 7.1 shows Case 1 to 3 Δ-error distributions for all possible matches for the *Lena* and *Mandrill* images using 8×8-pixel blocks.

As we move from Case 1 to Case 3 above, the error in approximating a given range block $u(R_i)$ by a given domain block $u(D_j)$ will generally decrease since more parameters are involved in the fitting. It was observed that the Case 3 Δ-error distributions for images demonstrate a significant peaking near zero error, indicating that blocks of these images are generally very well approximated by other blocks under the action of an affine greyscale transformation.

For a given Case k, the Δ-error distributions of some images were observed to be more concentrated near zero approximation error than others. The former images were viewed as possessing greater degrees of self-similarity than the latter. A quantitative characterization of relative degrees of self-similarity was also considered in terms of the means and variances of the error distributions. To illustrate, in the seven well-known test images employed in the study, the degree of Case 3 self-similarity could be ordered as follows:

$$\begin{gathered} \textit{Lena} \approx \textit{San Francisco} > \textit{Peppers} > \textit{Goldhill} > \\ \textit{Boat} > \textit{Barbara} > \textit{Mandrill}. \end{gathered}$$

As shown in [2], the histogram distributions of the standard deviations $s_{u(R_i)}$ of the 8×8-pixel range blocks of both images are virtually identical to the Case 3 Δ-error distributions in Fig. 7.1. This is to be expected since the standard deviation of the image subblock $u(R_i)$

is the RMSE associated with the approximation by its mean: $u(R_i) \approx \bar{u}(R_i)$. This is, in turn, a suboptimal form of the Case 3 approximation obtained by fixing the greyscale parameter $\alpha = 0$. The distribution of α greyscale parameters is, however, found to be highly concentrated at zero [2], implying that in most cases the standard deviation is a very good estimate of the Case 3 Δ-error.

7.2 SSIM-based Affine Self-Similarity Model

We re-examine the above self-similarity model in terms of the Structural Similarity (SSIM) index (1.21). The first step is to determine the formulas for optimal SSIM-based approximations of image range blocks by domain blocks which correspond to Cases 1 to 3 above. We then present the distributions of SSIM measures between domain and range blocks for the *Lena* and *Mandrill* test images which, from the order above, lie on opposite ends of the L^2-based self-similarity spectrum.

7.2.1 Optimal SSIM-based Affine Approximation

We now consider the approximation of an image range block $u(R_i)$ by a domain block $u(D_j)$ as written in (7.1) according to the SSIM index. The SSIM measure associated with the approximation in (7.1) is defined as

$$S_{ij} = \max_{\alpha,\beta\in\Pi} S(u(R_i), \alpha u(D_j) + \beta), \quad i \neq j. \tag{7.6}$$

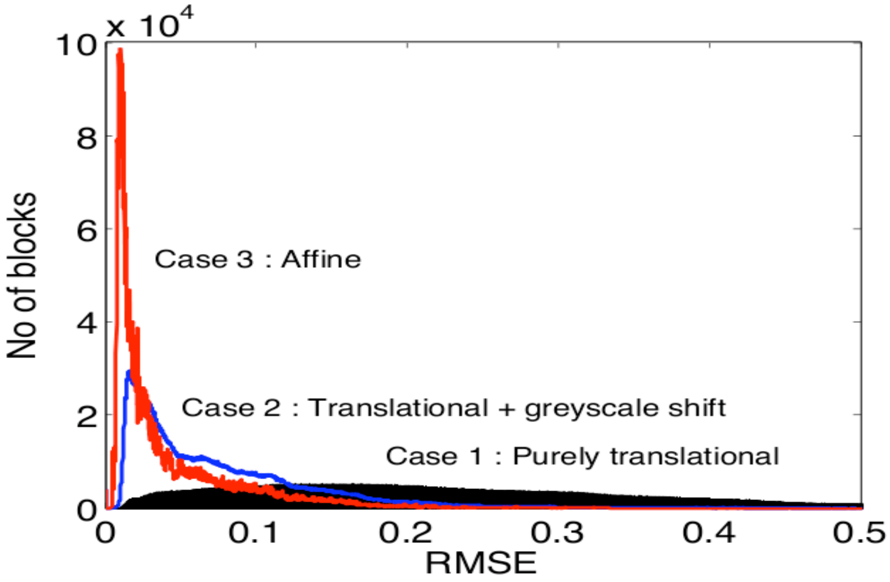
The optimal parameters and associated SSIM measures are given below for positive stability parameter, i.e. $\epsilon_1 = \epsilon_2 \geq 0$, thus extending our previous study [14]. In what follows, we once again denote $\mathbf{x} = u(R_i)$, $\mathbf{y} = u(D_j)$ and $N = n^2$.

Case 1: Purely translational. There is no optimization in this case: $\alpha_{ij} = 1$, $\beta_{ij} = 0$ and the SSIM measure is simply

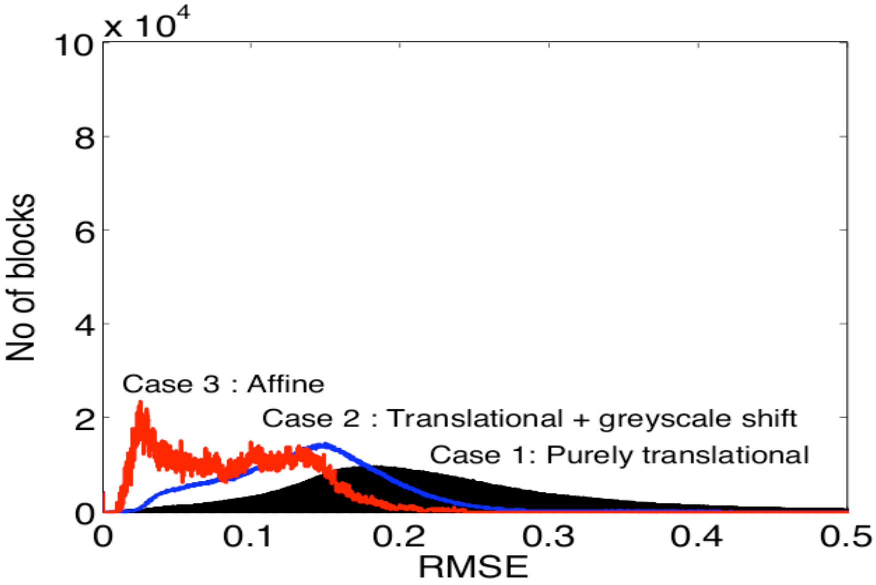
$$S_{ij}^{(\text{Case 1})} = S(\mathbf{x}, \mathbf{y}). \tag{7.7}$$

Case 2: Translational + greyscale shift. Here, $\alpha_{ij} = 1$ and we optimize over β.

$$\beta_{ij} = \bar{\mathbf{x}} - \bar{\mathbf{y}}, \qquad S_{ij}^{(\text{Case 2})} = S_2(\mathbf{x}, \mathbf{y}) = \frac{2s_{\mathbf{x},\mathbf{y}} + \epsilon_2}{s_{\mathbf{x}}^2 + s_{\mathbf{y}}^2 + \epsilon_2}. \tag{7.8}$$



(a) Cases 1, 2 and 3: *Lena*



(b) Cases 1, 2 and 3: *Mandrill*

Figure 7.1: Case 1 to 3 RMS Δ-error distributions for normalized *Lena* and *Mandrill* images over the interval $[0, 0.5]$.

Note that the SSIM-optimal β parameter is identical to its L^2 counterpart.

Case 3: Affine greyscale transformation. We optimize over α and β.

$$\alpha_{ij} = \frac{-\epsilon_2 + \sqrt{\epsilon_2^2 + 4s^2_{\mathbf{x},\mathbf{y}}(s^2_{\mathbf{x}} + \epsilon_2)/s^2_{\mathbf{y}}}}{2s_{\mathbf{xy}}}, \quad \beta_{ij} = \bar{\mathbf{x}} - \alpha_{ij}\bar{\mathbf{y}}, \tag{7.9}$$

$$S_{ij}^{(\text{Case } 3)} = \frac{2s^2_{\mathbf{xy}}}{\sqrt{\epsilon_2^2 s^4_{\mathbf{y}} + 4s^2_{\mathbf{xy}}s^2_{\mathbf{y}}(s^2_{\mathbf{x}} + \epsilon_2)} - \epsilon_2 s^2_{\mathbf{y}}}. \tag{7.10}$$

If $\epsilon_2 = 0$, then

$$\alpha_{ij} = \text{sign}(s_{\mathbf{xy}})\,\frac{s_{\mathbf{x}}}{s_{\mathbf{y}}}, \quad \beta_{ij} = \bar{\mathbf{x}} - \alpha_{ij}\bar{\mathbf{y}}, \qquad S_{ij}^{(\text{Case } 3)} = \frac{|s_{\mathbf{xy}}|}{s_{\mathbf{x}}s_{\mathbf{y}}}, \tag{7.11}$$

where sign(t) $= 1$ if $t > 0$, 0 if $t = 0$, and -1 if $t < 0$. In this case, the SSIM measure S_{ij} is the magnitude of the correlation between $\mathbf{x}$ and $\mathbf{y}$.

In Cases 2 and 3, the means of the range block and optimally transformed range block are equal, i.e. $\bar{\mathbf{x}} = \alpha\bar{\mathbf{y}} + \beta$, as was the case for L^2-fitting.

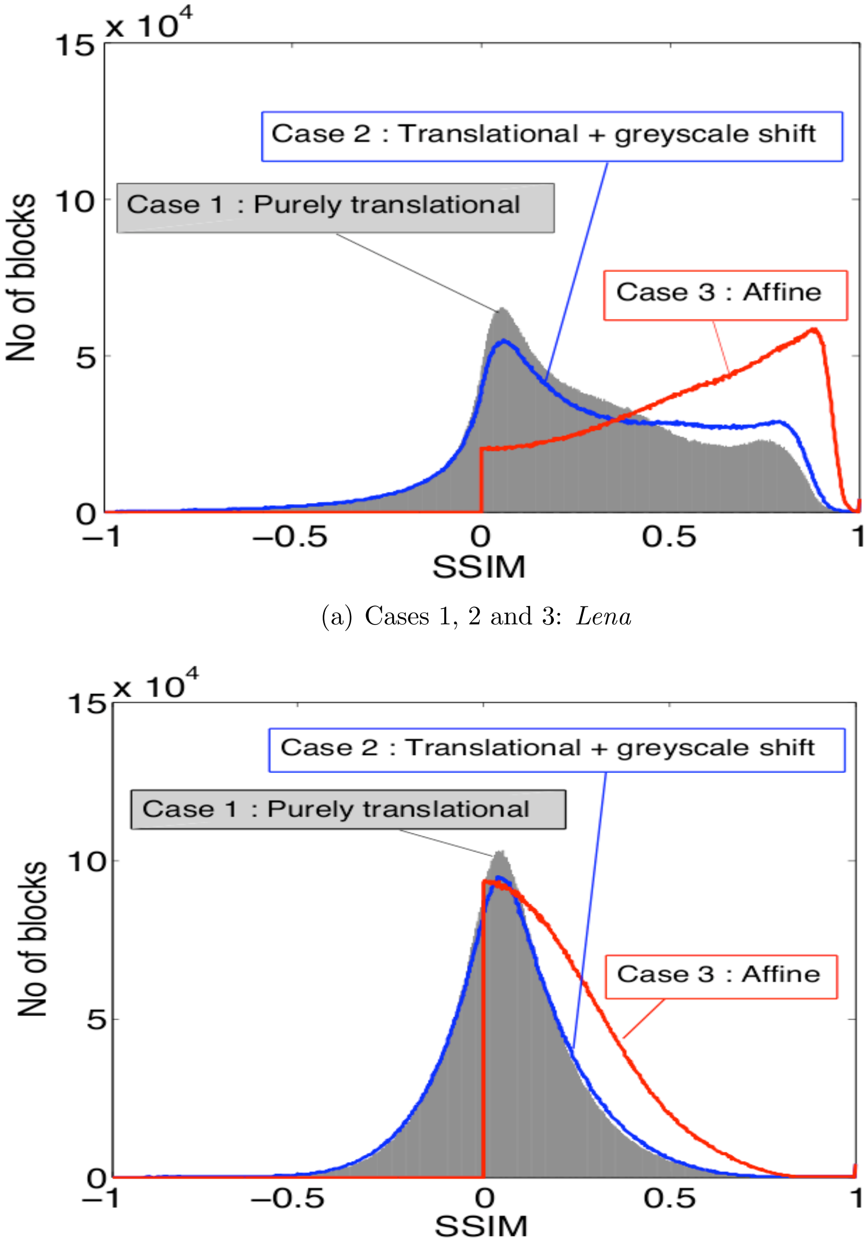
Note that the affine transform in Case 3 can also be seen as an approximation with the basis $\psi_0 = \mathbf{1}$ and $\psi_1 = \frac{1}{\sqrt{N-1}}\frac{\mathbf{y}-\bar{\mathbf{y}}}{s_y}$. In this case, the flat basis approximation (see Section 6.2.1) will lead to exactly the same result.

Since more parameters are involved as we move from Case 1 to 3, the associated SSIM measures behave as follows,

$$S_{ij}^{(\text{Case } 1)} \leq S_{ij}^{(\text{Case } 2)} \leq S_{ij}^{(\text{Case } 3)}. \tag{7.12}$$

Figure 7.2 shows Case 1 to 3 SSIM measure distributions over the interval $[-1, 1]$ of the *Lena* and *Mandrill* images using 8×8-pixel blocks once again.

Before commenting on these plots, we briefly discuss the issue of the stability parameters ϵ_1 and ϵ_2. As proposed in [114], the stability parameters employed in all computations reported below were $\epsilon_1 = 0.01^2$ and $\epsilon_2 = 0.03^2$. In the case of $\epsilon_1 = \epsilon_2 = 0$, the Case 1 SSIM measure distributions of the *Lena* and *Mandrill* images are almost identical. The slightly nonzero values of the stability parameters will increase the SSIM values associated with domain-range pairs with low variance. Since the *Lena* image contains a higher proportion of such blocks, there is a slight increase of the distribution for $S > 0$.



(a) Cases 1, 2 and 3: *Lena*

(b) Cases 1, 2 and 3: *Mandrill*

Figure 7.2: Case 1 to 3 SSIM measure distributions for normalized *Lena* and *Mandrill* images over $[-1, 1]$.

The difference between the two distributions is more pronounced in Case 2. For the *Lena* image, the better domain-range block approximations yielded by the greyscale shift cause its SSIM measure distribution to increase over the region $S \subset [0.5, 0.8]$.

But the situation is most interesting in Case 3, i.e. affine greyscale approximation. For both images, there are no negative SSIM values. This follows from the positivity of S_{ij} in (7.9) which is made possible by the inclusion of the α scaling factor. When the domain and range blocks are correlated, as opposed to anticorrelated, i.e. $s_{\mathbf{xy}} > 0$ then the optimal α coefficient is positive, implying that S will be positive. When $\alpha < 0$, the domain and range blocks are anticorrelated. So multiplying the domain block by a negative α value will "undo" this anticorrelation to produce a roughly correlated block.

The SSIM distribution for the *Lena* image has a much stronger component near $S = 1$. This indicates that many more blocks are well approximated in terms of the SSIM measure. Conversely, the SSIM measure for the *Mandrill* image is quite strongly peaked at $S = 0$. In summary, the SSIM measure corroborates the fact that the *Lena* image is more self-similar than the *Mandrill*'s. That being said, despite the dramatic peaking of the RMS Δ-error distribution of the *Lena* image at zero error, which is primarily due to a high proportion of low-variance blocks, its SSIM measure distribution does not demonstrate such peaking near $S = 1$.

As mentioned earlier, lower-variance blocks are more easily approximated in the L^2 sense than higher-variance blocks. Consequently, the Case 3 Δ-error distributions of images with a higher proportion of "flatter" (i.e. low variance) blocks will exhibit a greater degree of peaking near zero, particularly for Case 3. It turns out that our SSIM-based results allow us to address the question, raised in [2]: whether the self-similarity of an image is actually due to the "approximability" of its blocks which, in turn, is determined by their "flatness." If range blocks of low variance are easier to approximate, then perhaps a truer measure of self-similarity (or lack thereof) may be obtained if their corresponding Δ approximation errors are magnified appropriately to adjust for this "unfair advantage". The SSIM measure takes this "unfair advantage" into account, resulting in much less peaking near zero error compared to the RMS Δ-error distribution.

7.2.2 Relation between Optimal L^2- and SSIM-based Coefficients

At this point it is instructive to compare the affine greyscale transformations of the L^2- and SSIM-based approximations. Obviously, in Case 1, no comparison is necessary since no greyscale transformations are employed. In Case 2, the greyscale shift $\beta = \bar{u}(R_i) - \bar{u}(D_j)$ is the same in both approximations. In Case 3, it is sufficient to compare the α greyscale coefficients. Recall that for a given domain block $\mathbf{x} = u(D_j)$ and range block $\mathbf{y} = u(R_i)$,

$$\alpha_{L^2} = \frac{s_{\mathbf{xy}}}{s_{\mathbf{y}}^2}, \qquad \alpha_{SSIM} = \alpha_{ij} = \frac{-\epsilon_2 + \sqrt{\epsilon_2^2 + 4s_{\mathbf{xy}}^2(s_{\mathbf{x}}^2 + \epsilon_2)/s_{\mathbf{y}}^2}}{2s_{\mathbf{xy}}}. \tag{7.13}$$

It follows that

$$\frac{\alpha_{SSIM}}{\alpha_{L^2}} = \frac{-\epsilon_2 s_{\mathbf{y}}^2 + \sqrt{\epsilon_2^2 s_{\mathbf{y}}^4 + 4s_{\mathbf{xy}}^2(s_{\mathbf{x}}^2 + \epsilon_2)s_{\mathbf{y}}^2}}{2s_{\mathbf{xy}}^2} \geq 1. \tag{7.14}$$

This result implies that the SSIM-based affine approximation $\alpha u(D_j) + \beta$ will have a higher variance than its L^2-based counterpart. Such a “contrast enhancement” was also derived for SSIM-based approximations using orthogonal bases [12].

Finally, as it was observed in [111], note that given a fixed range block, the choice of best domain block for the L^2- and SSIM-based affine approximation is exactly the same. Indeed, for a fixed range block $\mathbf{x}$, the best L^2-based affine approximation will minimize

$$\Delta_{ij}^{(Case\ 3)} = \left[\frac{N-1}{N}\right]^{1/2} \left[s_{\mathbf{x}}^2 - \frac{s_{\mathbf{xy}}^2}{s_{\mathbf{y}}^2}\right]^{1/2}. \tag{7.15}$$

In a similar way, the best SSIM-based affine approximation will maximize

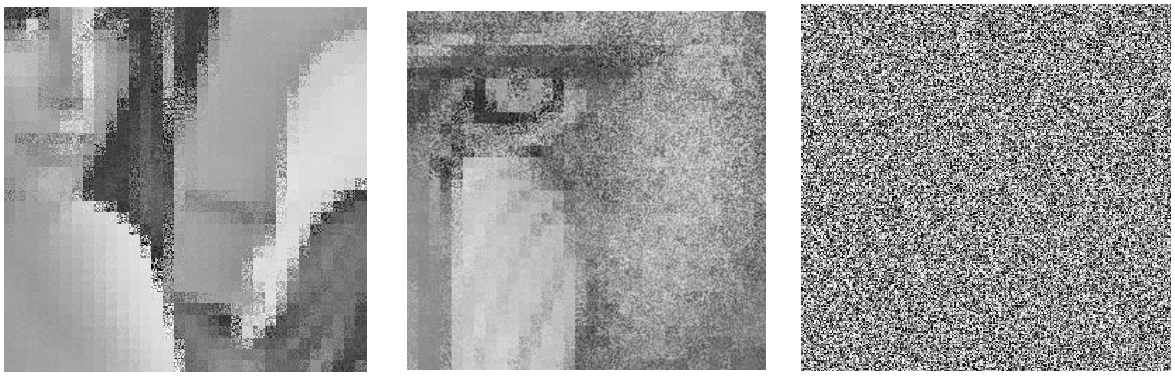
$$S_{ij}^{(\text{Case 3})} = \frac{2s_{\mathbf{xy}}^2/s_{\mathbf{y}}^2}{\sqrt{\epsilon_2^2 + 4(s_{\mathbf{x}}^2 + \epsilon_2)s_{\mathbf{xy}}^2/s_{\mathbf{y}}^2} - \epsilon_2}. \tag{7.16}$$

In both cases, the choice of block $\mathbf{y}$ that will maximize $s_{\mathbf{xy}}/s_{\mathbf{y}}$ will be optimal. Thus, for the block search itself, there is absolutely no difference between L^2-based and SSIM-based affine approximations. This can be explained by the fact that affine approximations do not change the structure of the blocks. Therefore they do not take advantage of the full potential of the SSIM index.

7.3 Self-Similarity of Natural Images vs. Pure Noise Images

The presence of noise in an image will decrease the ability of its subblocks to be approximated by other subblocks. In [2] it was observed that as independent and identically distributed (i.i.d.) Gaussian noise of increasing variance σ^2 is added to an image, any near-zero peaking of its Δ-error distribution becomes diminished. Moreover, a χ error distribution associated with the noise which peaks at σ eventually dominates the Δ-error distribution. This peaking at σ is actually the basis of the block-variance method of estimating additive noise.

Naturally, the SSIM measure distributions will also be affected by the presence of noise. But instead of simply adding noise to natural images, we wish to study pure noise images. Synthesizing such kinds of images allows us to compare the Δ-error distributions of natural images with a benchmark image that possesses no self-similarity. Indeed, for independent pure noise images there is no self-similarity between two blocks in the sense that the expectation of the covariance between them is zero. The only parameters affecting the self-similarity (in RMSE or SSIM-sense) are the local mean and variance of the image. This leads to the following idea: Generate an image from a uniform distribution with the local mean and variance matched to the statistics of a natural image. In our experiments, we chose a uniform distribution. But the histograms would have been similar for Gaussian or Poisson probability distribution. Figure7.3 shows two examples of pure noise images for which the local statistics are matched to a natural image. Also shown is a pure noise image following i.i.d. uniform distribution on $[0, 1]$. To be consistent throughout this chapter, disjoint blocks were used to compute the local statistics.



(a) *Lena*-like noise

(b) *Mandrill*-like noise

(c) Uniform noise

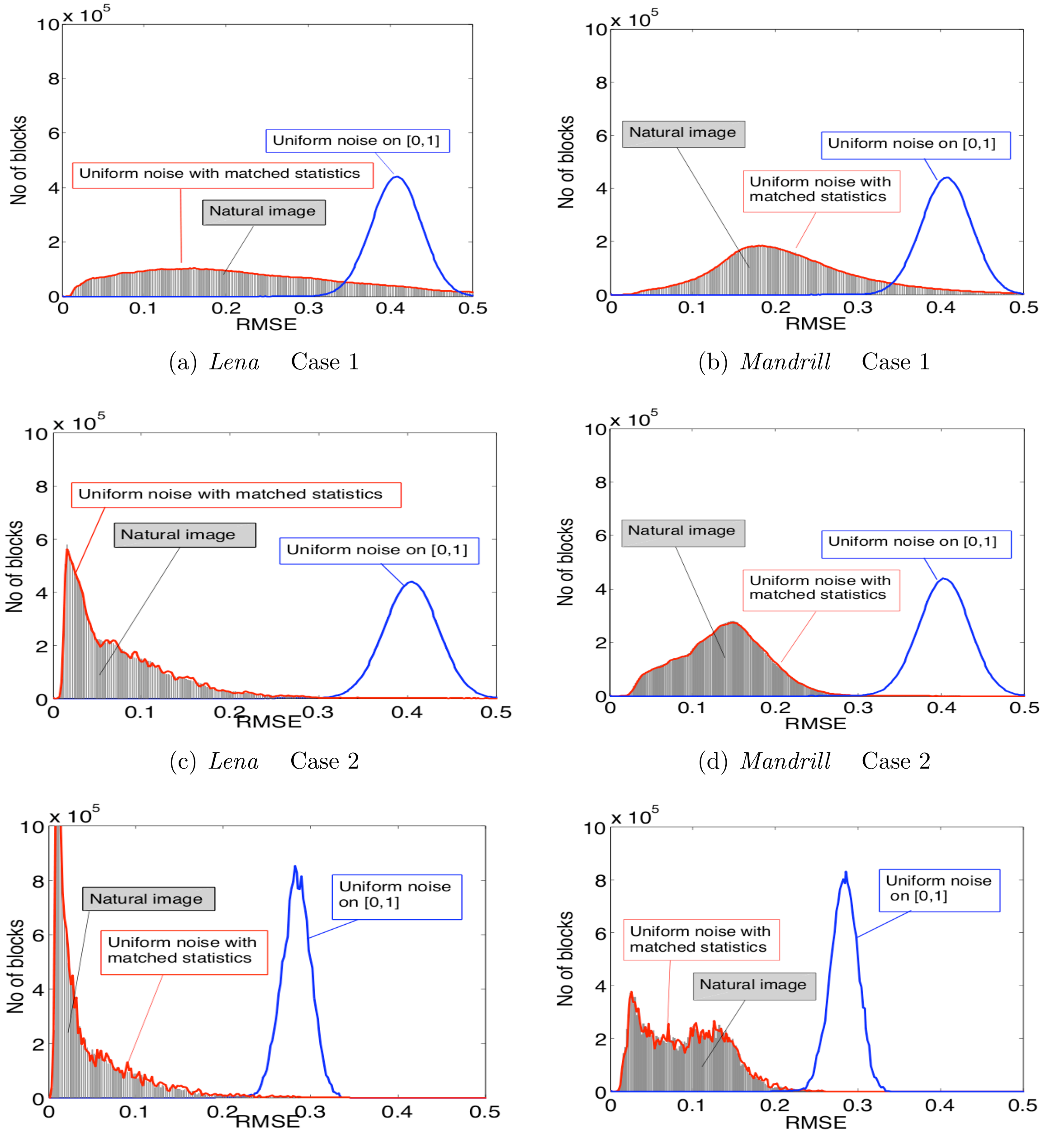
Figure 7.3: Images made of uniform noise with statistics matching the local mean and variance of natural images: (a) *Lena* (b) *Mandrill* (c) Noise image with each pixel value taken from an uniform distribution on [0,1].

In Fig. 7.4, we compare the RMS Δ-error distribution of natural images with the RMS Δ-error distribution of a pure noise image with the local statistics matched and of a pure noise image following a uniform distribution on $[0, 1]$. We observe that there is as much self-similarity for natural images than for pure noise images with matched statistics. Notice that all possible blocks were compared, whereas in non-local image processing only a limited number of (best) blocks are usually needed. So even if the best matches are generally more self-similar, on average, natural images are not more self-similar than pure noise images with matched statistics. Therefore, we conclude that low variance is the principal factor for self-similarity according to RMSE.

In order to correct this low variance bias, the same experiment was performed with the SSIM index for Case 1 to 3. The results are shown in Fig. 7.5. Now we can see a difference between the SSIM measure distributions of natural images and pure noise images. We quantify the self-similarity of images by computing the first moment (i.e. the mean) of the SSIM measure distributions. The results are shown in Table 7.1. Again, the local variance has a major influence on the self-similarity of images but now we can see, as expected, that natural images are more self-similar than pure noise images in the SSIM-sense. To theoretically determine the distribution of the SSIM between two blocks that are generated by a known probability distribution remains an open question. The difficulty here is the fact that rational functions are involved in the definition of the SSIM measure.

Table 7.1: Mean of the SSIM measure distributions of natural images (NI), pure noise images with matched statistics (MN) and uniform pure noise images (UN) for self-similarity models Case 1 to 3 of *Lena* and *Mandrill* images.

	Case 1	Case 1	Case 2	Case 2	Case 3	Case 3
	Lena	*Mandrill*	*Lena*	*Mandrill*	*Lena*	*Mandrill*
NI	0.2719	0.0682	0.3091	0.0731	0.5578	0.2246
MN	0.2698	0.0684	0.3074	0.0735	0.5206	0.1896
UN	0.0057	0.0057	0.0057	0.0057	0.1003	0.1004



(a) *Lena* Case 1

(b) *Mandrill* Case 1

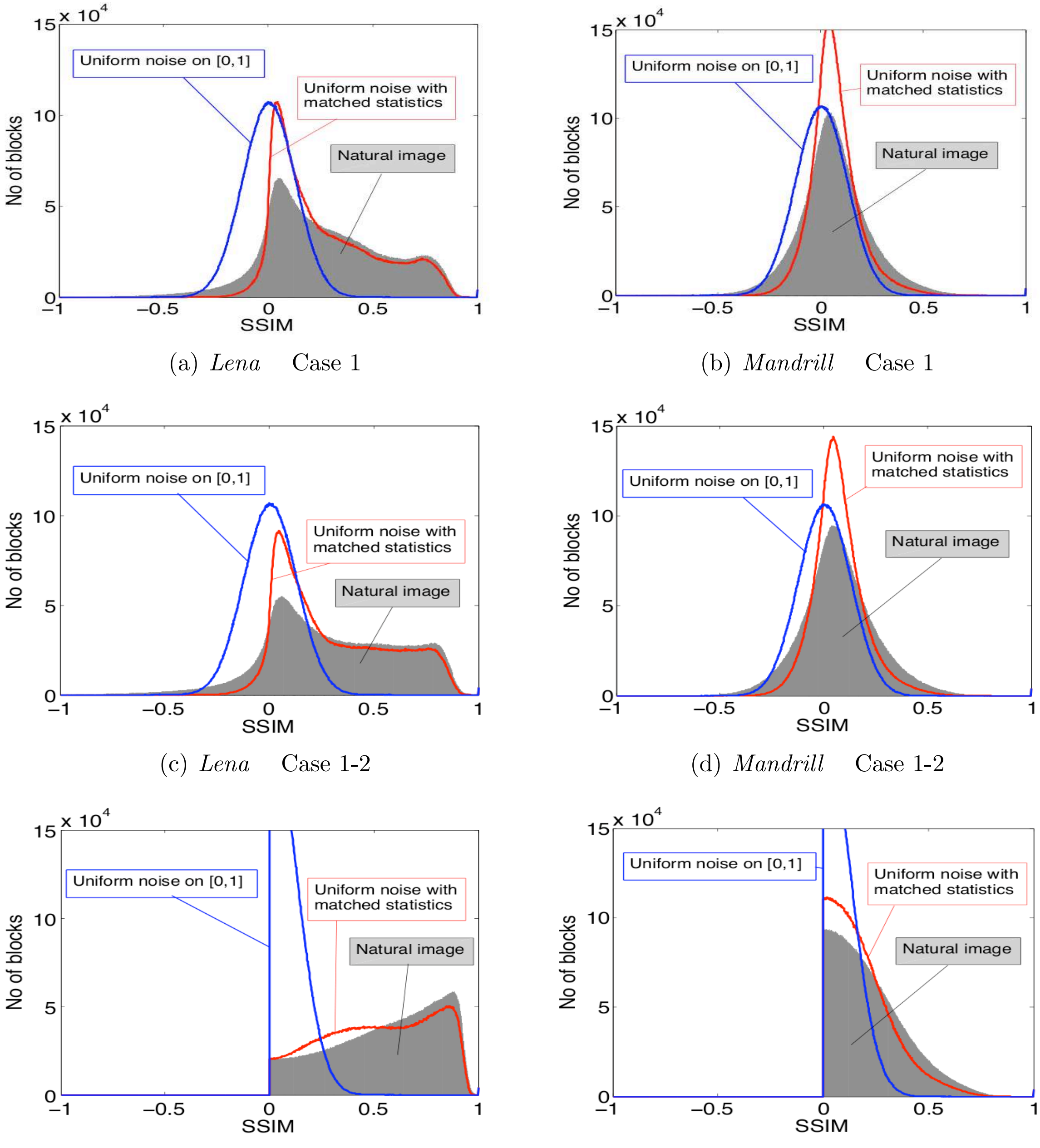
(c) *Lena* Case 2

(d) *Mandrill* Case 2

(e) *Lena* Case 3

(f) *Mandrill* Case 3

Figure 7.4: Comparison of RMS Δ-error distribution of *Lena* and *Mandrill* for Case 1 to 3 (grey histogram) with the RMS Δ-error distribution of pure noise images for which the local mean and local variance are matched (red) and with the RMS Δ-error distribution of an i.i.d. uniform pure noise image on [0, 1] (blue).



(a) *Lena* Case 1 (b) *Mandrill* Case 1

(c) *Lena* Case 1-2 (d) *Mandrill* Case 1-2

(e) *Lena* Case 3 (f) *Mandrill* Case 3

Figure 7.5: Comparison of SSIM measure distribution of *Lena* and *Mandrill* for Case 1 to 3 (grey histogram) with the SSIM measure distribution of pure noise images for which the local mean and local variance are matched (red) and the SSIM measure distribution of an i.i.d. uniform pure noise image on $[0, 1]$ (blue).

Chapter 8

The Use of Residuals in Image Denoising

As described in the introduction, the problem of image denoising may be conveniently summarized as follows: Given a noisy version $\mathbf{y}$ of an image $\mathbf{x}$, construct a denoising operator $\mathscr{D}$ so that $\mathscr{D}(\mathbf{y})$ is "close" to $\mathbf{x}$. In previous chapters, we have studied how to model and optimize closeness between images. This chapter addresses the model of image corruption.

We have mentioned that in most, if not all, image denoising studies, the noiseless image $\mathbf{x}$ is available for testing, thus enabling a direct comparison of $\mathbf{d}$ and $\mathbf{x}$ for the evaluation of denoising operators. But what if $\mathbf{x}$ is not available, as is the case in practical situations? Can one estimate the efficacy of the denoising operator from knowledge of $\mathbf{d}$ and $\mathbf{y}$ alone? This is the essence of *no-reference* quality assessment for image denoising. Note that contrary to traditional no-reference image quality assessment where the quality of a distorted image is sought after, here we want to estimate the quality of the restored image.

In general, image denoising is an ill-posed problem. In order to provide a meaningful solution, additional knowledge about the target image $\mathbf{x}$ must be imposed into the solution. In recent years, statistical models of natural images have attracted great interest. These have provided useful statistical prior knowledge about the clean natural image and thus have helped the denoiser to distinguish the image signal from noise. However, one intuitively useful aspect that is generally missing from state-of-the-art image denoising algorithms is the unexplored properties of the residual image $\mathbf{r} := \mathbf{y} - \mathbf{d}$. More precisely, these algorithms attempt to approximate $\mathbf{x}$ with $\mathbf{d}$ (by using the prior knowledge about

x) but generally ignore the desirable property that **r** should also follow the assumed distribution of the noise. This obviously motivates us to study the use of the residual in image denoising.

In literature, most examinations of the residual image have involved only a visual inspection of **r** [16]. In fact, several authors have employed **r** in the design of image denoising algorithms [56, 28, 78, 73, 68, 75, 100]. Nevertheless, none have deeply investigated the statistical properties of **r**, i.e. whether or not **r** is a valid sample from a particular noise distribution, and how such statistics can be used for no-reference image quality assessment and for improving image denoising results.

The image residual could be used in two different fashions: it can either be used to evaluate the performance of a particular instance of image denoising or to assess the general performance of an image denoising algorithm. We will mostly be interested in the first case, but we will keep in mind the possibility of the second case.

The chapter is divided as follows: In the first section, we describe ways to estimate the Mean Square-Error (MSE), the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity (SSIM) index between a given denoised image and the original image from only the knowledge of the noisy image and of the denoised image. Then, in the second section, we describe significance tests for determining locally the performance of image denoising algorithms. Finally, in the last section, we take advantage of these estimates and tests to propose boosting algorithms for image denoising.

8.1 No-Reference Estimation of MSE, PSNR and SSIM

For a quality assessment of image denoising algorithms, one relies generally on a full-reference image quality assessment (IQA) measure like the MSE or the SSIM index. The problem is that in real life situations, the original image is not provided for comparison. Our goal is thus to estimate IQA measures from only the noisy image $\mathbf{y} := \mathbf{x} + \mathbf{n}$ and the denoised image $\mathbf{d}$.

As before, we denote the sample mean of $\mathbf{x}$ by $\bar{x}$, the unbiased sample variance of $\mathbf{x}$ by s_x^2 and the unbiased covariance between $\mathbf{x}$ and $\mathbf{d}$ by $s_{x,d}$. These statistics are related to the L^2-norm and inner product via

$$\begin{aligned} \|\mathbf{x}\|^2 &= (N-1)s_x^2 + N\bar{x}^2 && (8.1) \\ \mathbf{x} \cdot \mathbf{d} &= (N-1)s_{x,d} + N\bar{x}\bar{d}. && (8.2) \end{aligned}$$

These are all the statistics that need to be estimated in the computation of the MSE, PSNR and SSIM index.

8.1.1 Estimating the Mean and Variance of the Signal and Noise

Let $\mathscr{N}$ be the random vector generating the image noise $\mathbf{n}$. Let $\mathscr{X}$ be the random vector of the full population of images from which $\mathbf{x}$ is sampled. We assume that $\mathscr{N}$ is additive white Gaussian noise. Specifically, here are the assumptions we make about $\mathscr{N}$:

1. $E[\mathscr{X}(i)\mathscr{N}(i)] = 0$ for all indices $i \in I$;
2. $E[\mathscr{N}(i)\mathscr{N}(j)] = 0$ for all indices $i \neq j \in I$;
3. $E[\mathscr{N}(i)] = 0$ for all indices $i \in I$;
4. $E[\mathscr{N}^2(i)] = \sigma^2$ for all indices $i \in I$.

Here the variance of the noise is assumed to be known, but it should be estimated in practice.

From these assumptions, we can easily deduce well known unbiased estimates and errors for the sample mean and sample variance of $\mathbf{n}$ as well as for the sample covariance between $\mathbf{x}$ and $\mathbf{n}$. Denoting $\hat{x}$ as the estimate of the sample x drawn from a probability distribution X, we obtain the following estimates and errors:

$$\hat{\bar{n}} = \mathbf{E}[\bar{\mathscr{N}}] = 0, \qquad \mathrm{Var}(\bar{\mathscr{N}}) = \frac{\sigma^2}{N}, \tag{8.3}$$

$$\hat{s}_n^2 = \mathbf{E}[s_{\mathscr{N}}^2] = \sigma^2, \qquad \mathrm{Var}(s_{\mathscr{N}}^2) = \frac{2\sigma^4}{N-1}. \tag{8.4}$$

$$\hat{s}_{x,n} = \mathbf{E}[s_{\mathscr{X},\mathscr{N}}] = 0. \tag{8.5}$$

The variance of $s_{\mathscr{X},\mathscr{N}}$ is empirically estimated from a training set. As it can be seen in Fig. 8.1(a), it is negligible. For simplification, we consider that $s_{x,n} = 0$ in what follows.

The estimates of the sample mean and sample variance of $\mathbf{x}$ easily follow. Indeed, since $\mathbf{x} = \mathbf{y} - \mathbf{n}$, we have

$$\hat{\bar{x}} = \bar{y} - \hat{\bar{n}} = \bar{y} \quad \text{and} \tag{8.6}$$

$$\hat{s}_x^2 = s_y^2 - 2\hat{s}_{x,n} - \hat{s}_n^2 = \max(0, s_y^2 - \sigma^2). \tag{8.7}$$

The maximum is taken to ensure the positivity of the variance of $\mathbf{x}$. The variance of these estimates is negligible when they are computed over the whole image. Indeed, since the number of pixels of an image is several orders of magnitude greater than the variance of the noise, we can see from (8.3) and (8.4) that the variance of $\hat{\bar{n}}$ and $\hat{s_n^2}$ tend to zero. It thus implies that estimates of functions of $\bar{x}$ and s_x^2 can be directly obtained by substitution. For example,

$$\begin{aligned} \hat{s}_x &= \sqrt{\max(0, s_y^2 - \sigma^2)} \text{ and} \qquad (8.8) \\ \hat{\bar{x}}^2 &= \bar{y}^2. \qquad (8.9) \end{aligned}$$

In particular, this observation will allow us to directly obtain an estimate of the PSNR from the estimate of the MSE.

8.1.2 Estimating the Covariance of the Noise and the Residual

We now estimate the covariance between the residual image $\mathbf{r} = \mathbf{y} - \mathbf{d}$ and a sampled noise $\mathbf{n} = \mathbf{y} - \mathbf{x}$. This is the key step for the estimation of both the PSNR and SSIM index.

We model a denoised image as

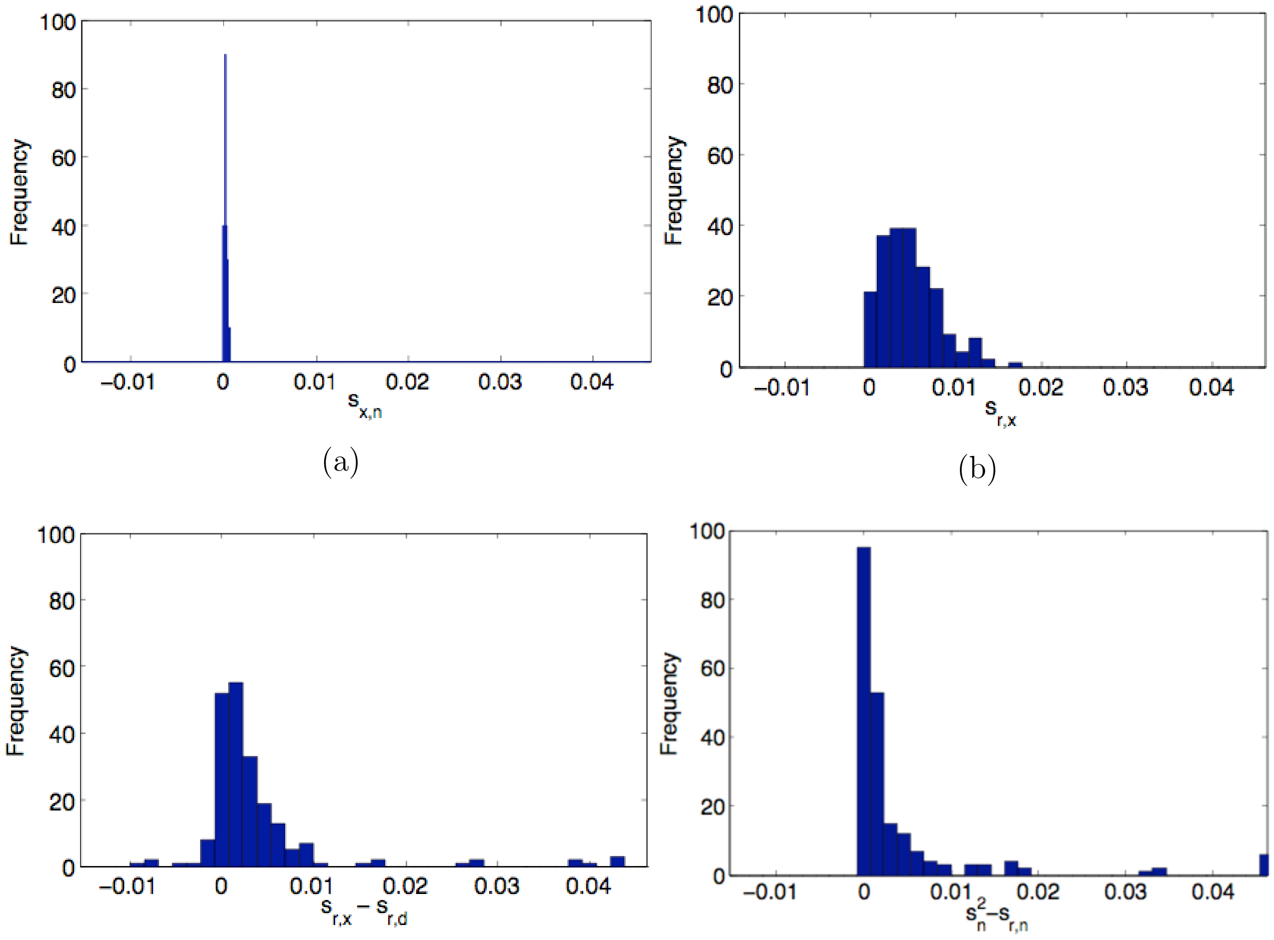
$$\mathbf{d}_i = a_i \mathbf{x}_i + b_i \mathbf{n}_i + \mathbf{g}_i, \quad \text{for all } i \in I, \qquad (8.10)$$

where $0 \leq a_i \leq 1$ and $0 \leq b_i \leq 1$ are linear attenuation factors of the original image and the noise, respectively, and $\mathbf{g}_i$ represents some image artifacts independent of both $\mathbf{x}_i$ and $\mathbf{n}_i$. The intuition behind this model is that most denoising algorithms, for example Gaussian filtering, reduce the amplitude of the signal and might add some spurious artifacts.

This model automatically implies the first and the third heuristic proposed in [11]:

$$\begin{aligned} s_{r,x} &\geq 0, \qquad (8.11) \\ s_{r,x} &\simeq s_{r,d} \text{ and} \qquad (8.12) \\ s_{r,n} &\leq s_n^2. \qquad (8.13) \end{aligned}$$

The second heuristic cannot be directly explained with the model (8.10). Although it might seem to be the most intuitive since it is assumed that the denoised image should be close to the original image, the heuristic reveals itself to be probably the weakest empirically. In Fig. 8.1, we test these heuristics from the computation of histograms obtained from various images, denoising algorithms and noise levels.



(a)

(b)

(c)

(d)

Figure 8.1: Histograms of global covariance for the images in the training set. (a) $s_{x,n}$ (b) $s_{r,x}$ (c) $s_{r,x} - s_{r,d}$ (d) $s_n^2 - s_{r,n}$.

From the heuristics (8.11) and (8.12) we deduce that

$$\hat{s}_{r,x} = \begin{cases} 0 & \text{if } s_{r,d} \leq 0, \\ s_{r,d} & \text{if } s_{r,d} \geq 0. \end{cases} \tag{8.14}$$

It follows that the estimate of $s_{\mathbf{r},\mathbf{n}}$ is

$$\hat{s}_{r,n} = \begin{cases} s_{r,y} & \text{if } s_{r,d} \leq 0, \\ s_r^2 & \text{if } s_{r,d} \geq 0. \end{cases} \tag{8.15}$$

Since $s_{y,r} = s_{r,d} + s_r^2$ and from the third heuristic (8.13), we finally obtain

$$\hat{s}_{r,n} \quad = \quad \min(s_r^2, s_{r,y}, s_n^2). \tag{8.16}$$

Similarly, we can also obtain an estimate of $\mathbf{r} \cdot \mathbf{n}$:

$$\widehat{\mathbf{r} \cdot \mathbf{n}} \quad = \quad \min(\|\mathbf{r}\|^2, \mathbf{r} \cdot \mathbf{y}, \|\mathbf{n}\|^2). \tag{8.17}$$

In addition, estimates of $s_{d,n}$ and $s_{x,d}$ may also be obtained as well, since

$$s_{d,n} + s_{r,n} \quad = \quad s_{x,n} + s_n^2 \simeq \sigma^2 \text{ and} \tag{8.18}$$
$$s_{y,d} \quad = \quad s_{n,d} + s_{x,d}. \tag{8.19}$$

8.1.3 Partial-Reference IQA Estimates

From the statistics of the noise obtained in the previous sections, we can now easily compute estimates of two common IQA measures: the MSE and its equivalent, the PSNR, as well as the SSIM index. We call the estimates partial-reference since no knowledge of the original image $\mathbf{x}$ is available, but only its distorted version $\mathbf{y} = \mathbf{x}+\mathbf{n}$ as well as the restored image $\mathbf{d}$.

The mean-square error (MSE) between the original image $\mathbf{x}$ and the denoised image $\mathbf{d}$ is

$$\text{MSE}(\mathbf{x}, \mathbf{d}) \quad = \quad \frac{1}{N}\|\mathbf{x} - \mathbf{d}\|_2^2, \tag{8.20}$$

where L is the size of the signals. The peak signal-to-noise ratio (PSNR) is computed from the MSE by the following formula:

$$\mathrm{PNSR}(\mathbf{x},\mathbf{d}) = 10\log_{10}\left(\frac{1}{\mathrm{MSE}(\mathbf{x},\mathbf{d})}\right), \tag{8.21}$$

where $\mathbf{x}$ and $\mathbf{d}$ are normalized into the range $[0,1]$.

Observe that

$$\begin{aligned} \mathrm{MSE}(\mathbf{x},\mathbf{d}) &= \mathrm{MSE}(\mathbf{r},\mathbf{n}) \qquad (8.22)\\ &= \frac{\|\mathbf{r}\|^2}{N} - \frac{2}{N}\mathbf{r}\cdot\mathbf{n} + \frac{\|\mathbf{n}\|^2}{N}. \qquad (8.23)\end{aligned}$$

A no-reference estimate is derived by substituting (8.16) in (8.22) and combining this with the estimate of $\|\mathbf{n}\|^2$:

$$\begin{aligned} \widehat{\mathrm{MSE}}(\mathbf{x},\mathbf{d}) &= \frac{\|\mathbf{r}\|^2}{N} - \frac{2}{N}\min(\|\mathbf{r}\|^2, \mathbf{r}\cdot\mathbf{y}, N\sigma^2) + \sigma^2 \qquad (8.24)\\ &= \max\left(\frac{\|\mathbf{r}\|^2}{N} - \frac{2}{N}\mathbf{r}\cdot\mathbf{y} + \sigma^2, \left|\frac{\|\mathbf{r}\|^2}{N} - \sigma^2\right|\right). \qquad (8.25)\end{aligned}$$

Since the IQA is taken over the whole image, the variance of the estimated variance of the noise is negligible and the "plug-in" estimate of the PSNR follows directly from the formula (8.21) applied to our estimate of the MSE (8.24).

The results of a number of experiments comparing the PSNR to its no-reference estimate are shown in Fig. 8.2. We used three images (*Lena*, *Boat* and *Barbara*) while artificially adding white Gaussian noise with seven different standard deviation values ($\sigma = 10, 15, 25, 35, 50, 70$ and 100). Ten denoising methods were applied to the resulting noisy images: Gaussian filtering, wavelet soft-thresholding [35], stationary wavelet hard-thresholding [30], anisotropic filter [77], total variation minimization [94], Wiener adaptive filter [63], curvelet denoising [18], scaled mixtures of Gaussian [81], non-local means [16] and sparse 3D transform-domain collaborative filtering [31]. A last "denoising" procedure applied was a pathological one, where the denoised image is the noisy image added with a Gaussian white noise image of variance σ^2. This can be expressed with

$$\mathbf{d} = \mathbf{y} + \mathbf{m} \text{ where } \sigma^2_{\mathscr{M}} = \sigma^2_{\mathscr{N}} \text{ and } \sigma_{\mathscr{M},\mathscr{N}} = \sigma_{\mathscr{X},\mathscr{M}} = 0. \tag{8.26}$$

The no-reference estimate of the PSNR provides a fairly good approximation of the true PSNR. Indeed, their Pearson's correlation coefficient is 0.87.

We employ the same "plug-in" method for the estimation of the SSIM index. Gathering the estimates of $\bar{x}$ (8.6), $\bar{x}^2$ (8.9), s_x^2 (8.7) and $s_{x,d}$ (8.19) combined with the estimates of $s_{n,d}$ (8.18) and $s_{n,r}$ (8.16) in the equation of the SSIM index (1.21), we obtain the no-reference SSIM map

$$\widehat{\mathrm{SSIM}}(\mathbf{x},\mathbf{d}) = \left(\frac{2\hat{\bar{x}}\overline{d}+\epsilon_1}{\hat{\bar{x}}^2+\overline{d}^2+\epsilon_1}\right)\left(\frac{2(s_{y,d}-\hat{s}_n^2+\hat{s}_{n,r})+\epsilon_2}{\hat{s}_x^2+s_d^2+\epsilon_2}\right) \tag{8.27}$$

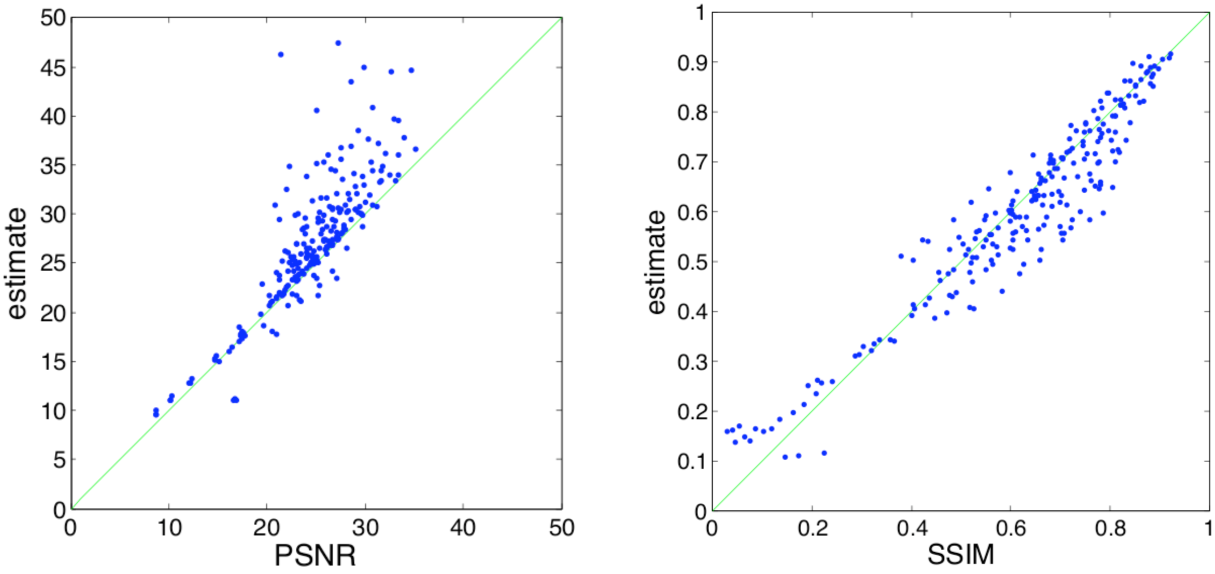
$$= \left(\frac{2\overline{y}\overline{d}+\epsilon_1}{\overline{y}^2+\overline{d}^2+\epsilon_1}\right)\left(\frac{2(s_{y,d}-\sigma^2+\hat{s}_{n,r})+\epsilon_2}{\max(0,s_y^2-\sigma^2)+s_d^2+\epsilon_2}\right). \tag{8.28}$$

Proceeding as in the experiment for the no-reference PNSR estimate, we present in Fig. 8.2(b) a scatter plot of quality indices against their estimates for different images, noise levels and denoising algorithms. We observe an excellent estimate of the SSIM index when the noise level is reasonable, but for high levels of noise ($\sigma \geq 50$), the estimate is slightly less accurate. A linear correlation coefficient of 0.96 between the SSIM index and its estimate represents a very convincing practical argument of the validity of the method. One can explain the better estimate of the SSIM index compared to the estimate of the PSNR by the fact that the former is obtained by averaging scores obtained on a local window whereas the later computes the average of errors on a single pixel. Thus, the greater sample size will allow a smaller error of estimation of the local score.

Contrary to the global SSIM index estimate, the local estimate of the SSIM map is rather poor. When comparing the full-reference SSIM map with its no-reference estimate (8.27), we found that the rough estimate $\hat{s}_x$ slightly overestimates s_x in flat regions of $\mathbf{x}$ while sometimes underestimating it around edges and textures, creating instabilities in the correlation term. An example of SSIM map and its estimate for an artificially corrupted image *Lena* denoised by Gaussian filtering is illustrated in Fig. 8.3. The noise was simulated to be additive, white and Gaussian with standard deviation of 25/255, whereas the Gaussian filter had a standard deviation of 2 pixels and was truncated on a 13×13 pixel window.

8.1.4 Analysis of the Estimate of s_x

When the estimates are computed locally as is the case for the SSIM index or for a weighed MSE, the variance of the estimates $\bar{x}$ and s_x^2 is non-negligible. Thus, a direct substitution may lead to a bias in the estimation of IQA measures. Ideally, if the distribution of the estimation error can be calculated analytically, it is the way to go. For example, if $\mathscr{N}$ is a



(a) (b)

Figure 8.2: (a) Scatter plot of the PSNR and its no-reference estimates. (b) Scatter plot of the SSIM index and its no-reference estimates.

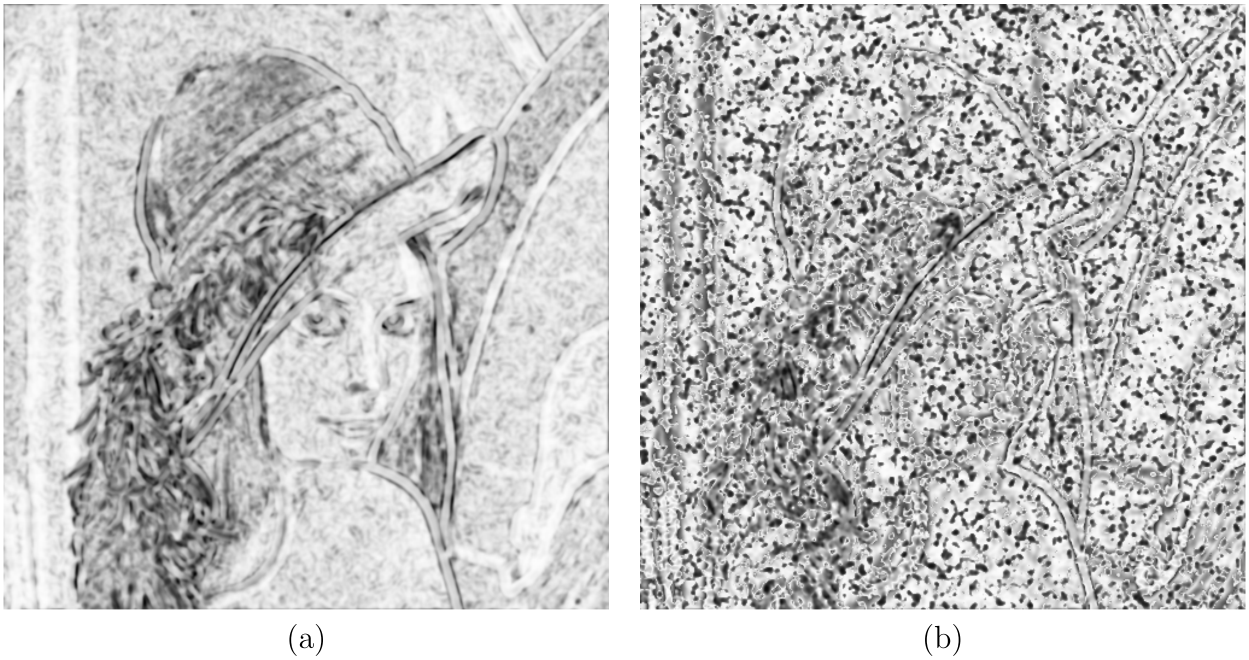


Figure 8.3: (a) SSIM map between the original image *Lena* and its denoised version. (b) Partial-Reference Estimate of the SSIM map.

Gaussian distribution of zero mean and variance σ^2, then the distribution of $\bar{\mathscr{N}}$ is simply a scaling of $\mathscr{N}$ by a factor $\frac{1}{N}$, i.e. a Gaussian distribution of zero mean and variance $\frac{\sigma^2}{N}$ as given by (8.3). Also, $s^2_{\mathscr{N}}$ is chi-squared distribution with $N-1$ degrees of freedom scaled by a factor $\frac{\sigma^2}{N-1}$.

However, in most of cases it is not possible to find the exact distribution. We then have to rely on random sampling. Given a sample from a known distribution, we apply the transformation on each sampled element to find the new distribution. For example, to estimate the local standard deviation of a clean image from a noisy one, we proceed as follows. First, we compute the local variance of the noisy image over with a sliding window. Then, we estimate the distribution of the variance of the noise image by random sampling. To do so, we generate a huge number (10000 in our experiment) of random copies of a block of the noise image and filter each copy by the local window. We then obtain a distribution of possible estimates by applying (8.8) for each location of the image with the true noise variance σ substituted by the calculated distribution of noise variances. Our scalar estimate will finally be the mean of the distribution of possible estimates. A confidence interval can also be easily obtained for each estimate.

A comparison between the rough estimate and the estimate obtained from random sampling is presented in Figs. 8.4 and 8.5. Notice that the two estimates only differ when the local variance of the noisy image is small. The estimate obtained from random sampling performs slightly better: its mean absolute error (MAE) is 0.02 whereas the MAE of the rough estimate is 0.0213. Both estimates perform relatively poorly in flat regions, by either overestimating or underestimating the local variance of the original image.

One limitation is the fact that the estimates are scalar, only taking into account s_y^2 in order to estimate s_x^2. A more powerful estimate would involve information on the local variance of neighbourhood pixel since it is observed to be strongly correlated.

8.2 Statistical Tests on the Image Residual

Here, the idea is to apply a statistical test locally on the residual image to determine whether or not it behaves like the model of pure noise. Indeed, some assumptions on the nature of the noise were made, and we can test if they are observed in the residual image. We employ two kinds of hypothesis tests about the residual image: (1) goodness-of-fit and (2) independence.

The first test will hopefully show that if $\mathbf{r}$ follows a normal distribution and if the intensity values of $\mathbf{r}$ are identically distributed with zero-mean and variance σ^2. The

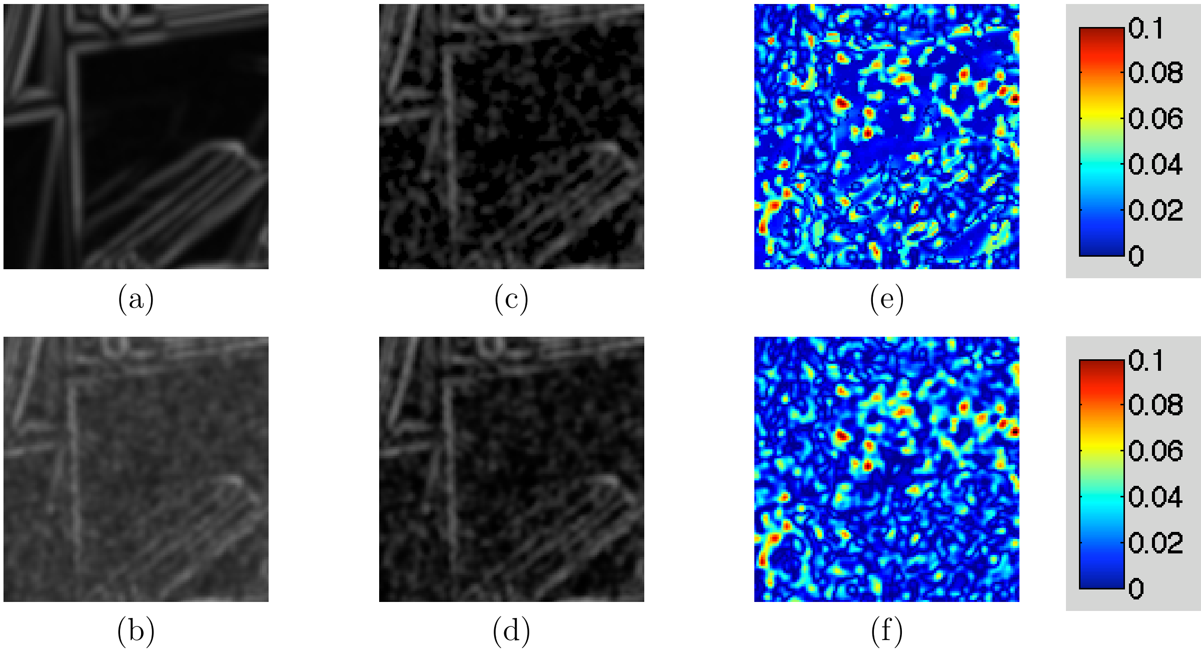


Figure 8.4: (a) Local variance of upper left corner of the original *Barbara* image. (b) Local variance of noisy *Barbara* image. (c) Rough estimate of the local variance of the original *Barbara* image. (d) Estimation of the local variance of the original *Barbara* image from random sampling. (e) Absolute error map of the rough estimate. (f) Absolute error map of the random sampling estimate.

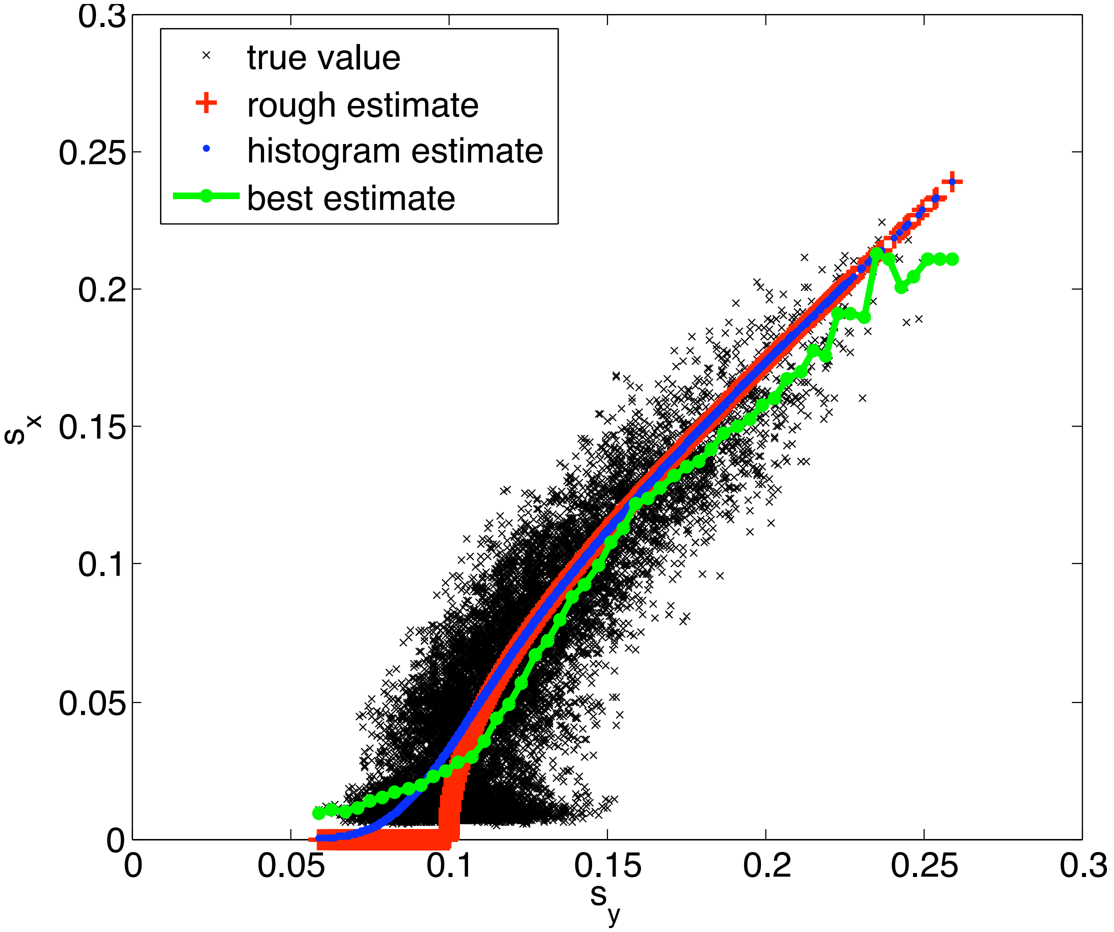


Figure 8.5: Scalar estimate of the local standard deviation of the original *Barbara* image from the local standard deviation of the noisy *Barbara* image. Black x: true value of the local standard deviation. Red cross: rough estimate $\hat{s}_x = \sqrt{\max(0, s_y^2 - \sigma^2)}$. Blue dot: estimate obtained by random sampling of noise images. Green line: average of the local standard deviation of the original image s_x given the standard deviation of the noisy image s_y.

second test will help determine whether or not **d** and **r** are independent and if there is any autocorrelation in **r**.

Note that there are several potential issues with this technique. First, the strength of the test will depend on how accurate the model of the noise is. The model studied in this section follows the standard model in image processing literature, but it is agreed that this might represent an oversimplification of reality. In the case of rejection of a hypothesis, it might be hard to distinguish between a misspecification of the model and a true problem with the denoising algorithm. Second, hypothesis testing only offers a dichotomic choice on rejecting or not a hypothesis, but does not give in general any information on how to correct the image denoising algorithm. Third, there is the potential difficulty often encountered in multiple testing. Repeating several times a hypothesis test will eventually lead to the detection of significance in the data no matter what is its true significance.

The first issue will be of importance in analyzing denoising procedures in the real-world, but we will concentrate here only in simulation studies where the noise is simulated from a given model. In order to answer the second issue, we will explore the possibility of denoising the residual image. Finally, we will attempt to address the third problem by adjusting appropriately the significance level of each individual test. Moreover, one can determine the overall quality of an image denoising procedure by comparing the significance level with the ratio of rejected image blocks.

8.2.1 Goodness-of-Fit Tests

Several goodness-of-fit tests are available to test the normality of a given sample. For example, one can use the kurtosis and the skewness of the distribution to compute the Jarque-Bera test [55]. Here we consider the Kolmogorov-Smirnov (K-S) test, applicable to any continuous distribution with the requirement, however, that all parameters are known. The K-S statistic is defined by

$$k = \sup_{x \in [0,1]} |E(x) - F(x)|, \tag{8.29}$$

where F is the theoretical cumulative distribution function of the distribution being tested and where E is the empirical cumulative distribution function of the sample. The hypothesis is rejected if the test statistic, k, is greater than a computed critical value.

The details of an implementation in Matlab are provided in the Appendix under the name `vwkstest`. The statistic k is computed in parallel using Matlab's matrix represen-

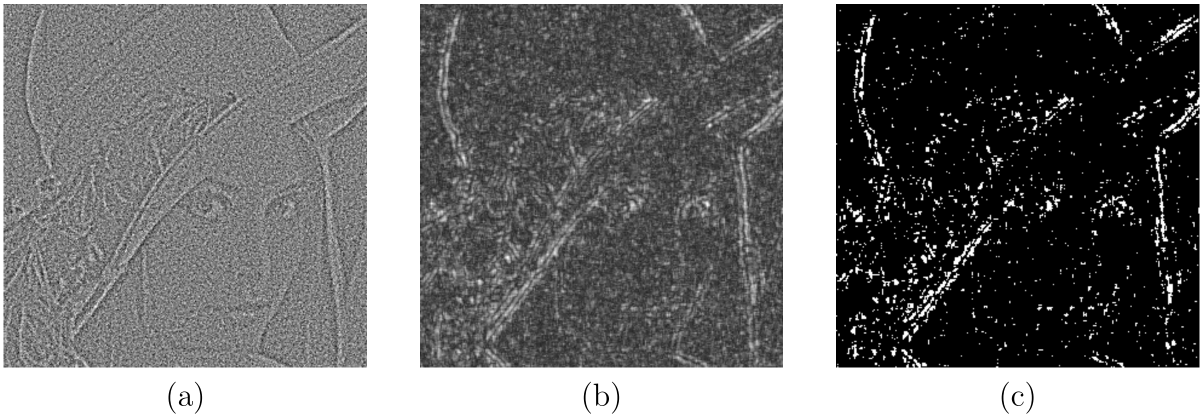


Figure 8.6: (a): Residual image. (b): The local Kolmogorov-Smirnov statistics of **r** scaled between 0 (black) and 0.5 (white). (c): The result of the K-S test (white: reject normality hypothesis, black: do not reject). All images were cropped between $(101, 101)$ and $(400, 400)$.

tation. The Kolmogorov-Smirnov cumulative distribution function is estimated by the truncation of some series (see [82]).

Using again the residual image $\mathbf{r}$ as described in Fig. 1.1, the K-S test was performed locally, using a 5×5 sliding window. The test can reveal at the same time if there is a departure from normality and if the mean and variance are, respectively, 0 and σ^2 over the entire residual image. Figure 8.6(a) shows again the residual to facilitate comparison while the resulting local K-S statistics can be seen in Fig. 8.6(b). In Fig. 8.6(c) are presented the result of the local hypothesis test for $\alpha = 0.05$. Observe that in this particular example departure from normality is detected at the vicinity of edges. This can be explained by the nature of the denoising procedure that smoothes out edges and textures.

8.2.2 Independence Tests

Correlation Coefficient Tests

Given n data points $(\mathbf{x}, \mathbf{y})$, the Pearson's correlation coefficient is defined by

$$\rho = \frac{s_{\mathbf{x},\mathbf{y}}}{s_{\mathbf{x}} s_{\mathbf{y}}}, \qquad (8.30)$$

where $s_{\mathbf{x}}$ and $s_{\mathbf{y}}$ are the sample standard deviations of, respectively, $\mathbf{x}$ and $\mathbf{y}$ and $s_{\mathbf{x},\mathbf{y}}$ is the sample covariance of $\mathbf{x}$ and $\mathbf{y}$. We use a criterion based on the value

$$t \;=\; \rho\sqrt{\frac{n-2}{1-\rho^2}}. \tag{8.31}$$

The criterion t follows a Student-t distribution with $n-2$ degrees of freedom.

For a localized solution, we computed the Pearson's linear correlation on a weighted sliding window. In this case, the sample standard deviation $s_{\mathbf{x}}$ and $s_{\mathbf{y}}$ and the sample covariance $s_{\mathbf{x},\mathbf{y}}$ are computed as in the SSIM index. Since it is not clear exactly which distribution the statistic t will follow, we assumed the same distribution as in the uniform window case, but with an unknown number of degrees of freedom. Instead of tuning the number of degrees of freedom for the desired significance level α, we adjusted the parameters of the weighted window given a fixed amount of degrees of freedom. Thus, specifying the distribution of the statistic t with $n-2$ degrees of freedom, we chose the parameters of a Gaussian window in order to reject approximately $100\alpha\%$ of the blocks (for a significance level of α) for two independent randomly generated noise images.

The linear correlation test on uniform 7 × 7 window is presented in Fig. 8.7(a) and (d). These results are compared with the weighted linear correlation test on a truncated circular Gaussian window of size 9 × 9 and of standard deviation 2 in Fig. 8.7(b) and (e). It can be seen that the former test presents more natural results with the significance test detecting dependence on clearly defined edges.

The Pearson's linear correlation test is only valid if the data is randomly sampled from a normal distribution. Since we already tested for normality for blocks $\mathbf{r}_i$, we can test for their correlation. On the other hand, since $\mathbf{d}$ is not expected to follow a normal distribution this test should not be valid for testing independence between $\mathbf{d}$ and $\mathbf{r}$. Nevertheless, the linear correlation between the denoised image and its residual appears to be the most accurate statistic in practice.

A simple alternative to this test is to replace the linear correlation ρ by the Spearman rank-order correlation

$$\rho_s = \frac{s_{\mathbf{s},\mathbf{t}}}{s_{\mathbf{s}}s_{\mathbf{t}}} \tag{8.32}$$

with $\mathbf{s}$ and $\mathbf{t}$ the ranks of $\mathbf{x}$ and $\mathbf{y}$, respectively. That is, if $\mathbf{x}_i$ is the k-th biggest element of $\mathbf{x}$, then $\mathbf{s}_i = k$. If there is a tie between two or more elements of $\mathbf{x}$, then an appropriate mid-rank is assigned for these elements. The advantage of this test over the linear correlation test is that we do not need anymore to make any assumption on the distribution of the sample. The cost to be paid is the loss of precision in the computation of the correlation coefficients as it can be seen in Fig. 8.7(c) and (f). The weighted version of the Spearman's correlation test did not provide any interesting results and is not included in the figure.

The independence between blocks of the residual is tested with the Pearson's linear correlation test on a weighted window. Since only the linear dependence is accounted for in correlation coefficients, we can adventure to test even for overlapping blocks. Indeed, the test is more likely to detect dependence for small shift of the residual blocks. The computation of the correlation coefficients was limited to a horizontal and a vertical shift of a single pixel to assure the independence between the correlation coefficients. The local independence test with significance level $\beta = 1 - (1-\alpha)^{1/2}$ was then performed for each shift. The combination of the locations where dependence was detected leads to an independence test with a significance level $\alpha = 0.05$. The results are shown in Fig. 8.8.

The vectorized Matlab's implementation of the weighted Pearson's linear correlation test and of the Spearman's rank correlation test are included in the Appendix under the respective names `vwpctest` and `vwsctest`. The handling of tie ranks for the Spearman's rank correlation was not implemented in Matlab since no simple solution (i.e. memory efficient) was found for a parallel processing. A C++ implementation of this algorithm is included in Numerical Recipes, third edition[82], under the name `crank`.

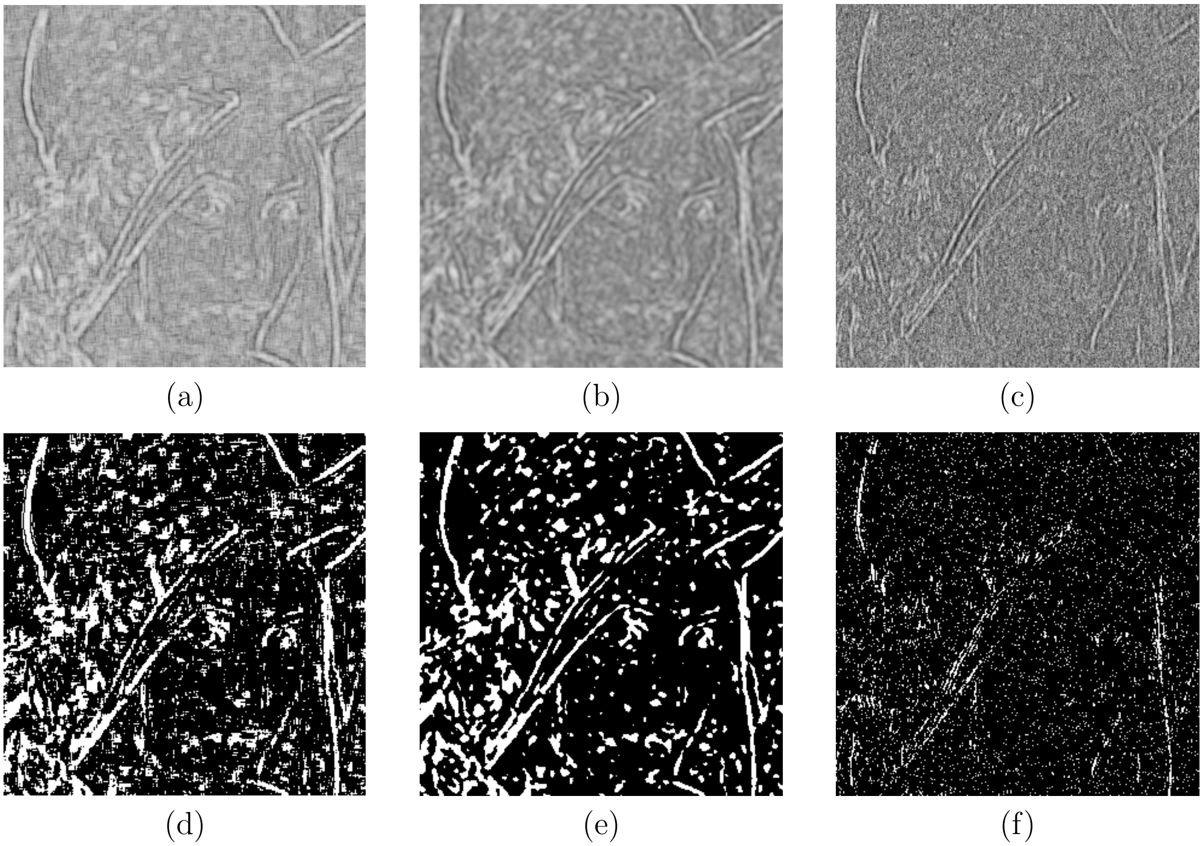
One attractive feature of Pearson's and Spearman's correlation coefficient test is their simplicity. The computational burden is light and a modest sample size is needed to estimate the correlation coefficient. That being said, the correlation coefficient describes only linear dependency and Pearson's coefficient lack of robustness to outliers.

G-test

Given n data points (x, y) and a $I \times J$ binning for these points, the G-test is based on the mutual information

$$MI(x;y) \;=\; \sum_{i=1}^{I}\sum_{j=1}^{J} p_{i,j} \ln\left(\frac{p_{i,j}}{p_{i,\cdot}p_{\cdot,j}}\right), \tag{8.33}$$

where $p_{i,j}$ is the sample joint probability that x is in the i-th bin and y is in the j-th bin and $p_{i,\cdot}$ and $p_{\cdot,j}$ are the sample marginal probabilities that, x is in the i-th bin and y is in the j-th bin, respectively. The criterion $G = 2nMI(x;y)$ follows a chi-square distribution with $r = (I-1)(J-1)$ degrees of freedom. The advantage of this method over Pearson's correlation coefficient is that it takes account of all types of dependencies. The drawback, however, is that a large number of samples is required to estimate the joint distribution.



(a)

(b)

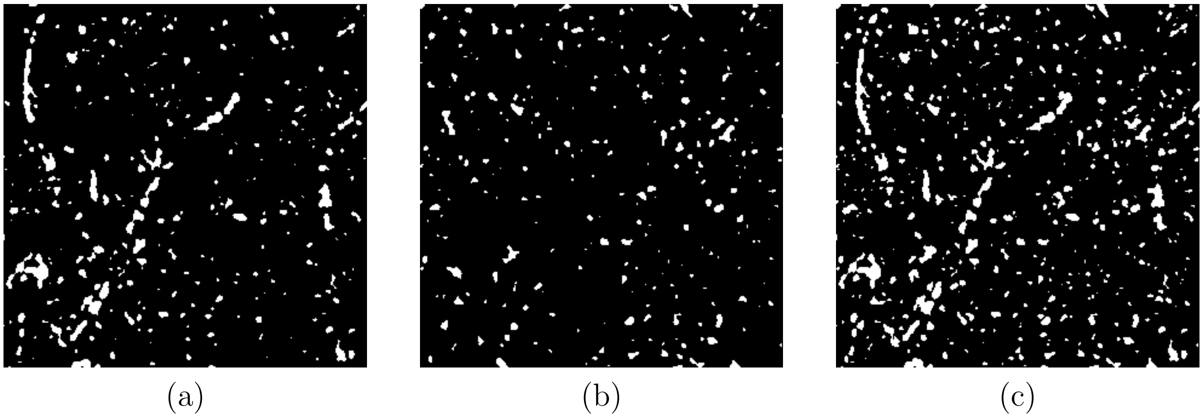
(c)

(d)

(e)

(f)

Figure 8.7: First row: Local statistics for the computation of local independence tests between **d** and **r**. (Black for negative correlation and white for positive correlation). (a): Pearson's correlation coefficients on a 7×7 sliding window. (b): Pearson's correlation coefficients on a 9×9 sliding circular Gaussian window with standard deviation of 2 pixels. (c): Spearman's rank correlation coefficients on a 7×7 sliding window. Second row: Result of local independence tests between **d** and **r** at a significance level $\alpha = 0.05$. (White: reject independence hypothesis, black: do not reject) (d): Pearson's linear correlation tests on a 7×7 sliding window. (e): Pearson's linear correlation tests with 47 degrees of freedom on a 9×9 sliding circular Gaussian weighted window with standard deviation of 2 pixels. (f): Spearman's rank correlation tests on a 7×7 sliding window. All images were cropped between $(101, 101)$ and $(400, 400)$.



(a) (b) (c)

Figure 8.8: Result of local autocorrelation test between **r** and its horizontal and vertical shifts. (White: reject independence hypothesis, black: do not reject) (a): Horizontal shift. (b): Vertical shift (c): Combination of horizontal and vertical shifts. All images were cropped between $(101, 101)$ and $(400, 400)$.

Even though it is desired to compute local statistics, a sufficient amount of data is required to obtain good estimates of the joint distribution. We employ the naive histogram estimation technique with uniform binning. The number of bins was arbitrary set to $b = 6$ to ensure a sufficient number of samples in each bin. We did not adapt locally the range of the bins in order to provide a uniform estimation of the joint distribution and of the mutual information. Using once again the images in Fig. 1.1, we present in Fig. 8.9 an example of the local mutual information and its associated G-test computed between a denoised image **d** and its residual **r**. The statistic was computed on a 15×15 truncated circular Gaussian window with standard deviation of 4.25 pixels. This choice of window leads to a rejection of $100\alpha\%$ of the samples (at a significance level α) when the G-test is performed between two independent random noise images. In general, the test does not lead to the expected rate of rejection given a significance level. This could be explained by the fact that the joint distribution is poorly approximated.

A vectorized version G-test on a weighted window is implemented in Matlab under the name `vwgtest`. The computation of the mutual information as well as of the joint entropy and of the entropy of each distribution is gathered in the function `vwMI` whereas the histogram estimation of the joint distribution is performed by the function `vwhist`.

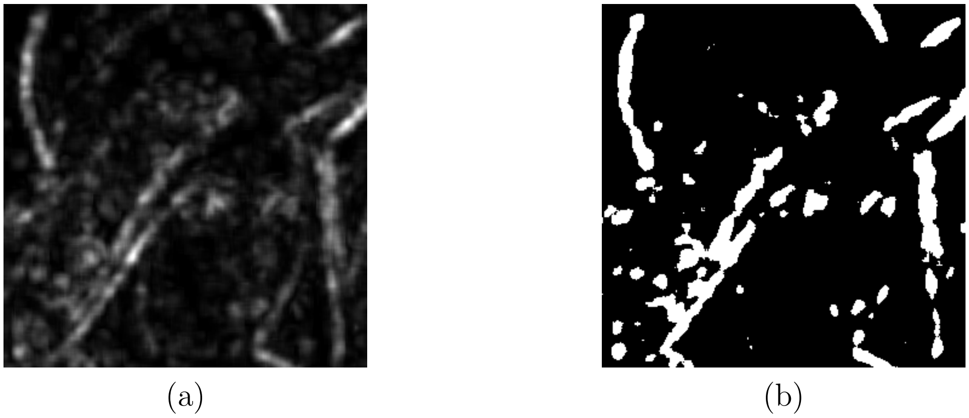


Figure 8.9: (a) Local mutual information between **r** and **d** computed on a 15×15 truncated circular Gaussian window with standard deviation of 4.25 pixels. Image scaled between 0 (black) and 1/3 (white). (b) Result of the G-test. (White: reject independence hypothesis, black: do not reject) (All images were cropped between $(101, 101)$ and $(400, 400)$.

8.3 Image Denoising Boosting

The statistics and tests developed in the previous section can not only help to determine the performance of a given denoising algorithm, but they can also guide the image denoising algorithm by giving feedback on its performance.

In this section we present a result from some preliminary experiments (first reported in [11]) which show that denoising a residual image and adding it to a denoised image can potentially enhance the quality of the latter. We then show how one could implement this scheme iteratively. Finally, we shall take advantage of the local statistical tests designed in the previous section to suggest a better way to improve the image quality.

8.3.1 Residual Filtering

In addition to denoising the noisy image, one may also try to denoise a residual by applying a different denoising algorithm and then adding it back. Mathematically, this may be expressed as follows,

$$\mathscr{E}\left(\mathbf{d}\right) = \mathbf{d} + \mathscr{D}_2(\mathbf{r}) = \mathscr{D}_1(\mathbf{y}) + \mathscr{D}_2(\mathbf{y} - \mathscr{D}_1(\mathbf{y})), \tag{8.34}$$

where $\mathscr{D}_1$ and $\mathscr{D}_2$ are two denoising algorithms.

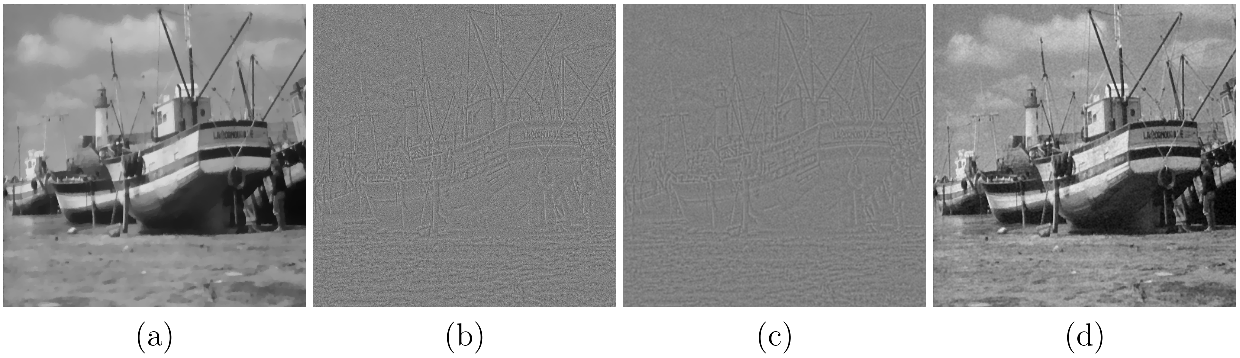


Figure 8.10: (a): *Boat* denoised by Total Variation Minimization (PSNR = 28.6, SSIM = 0.75). (b): Residual image. (c): Residual “denoised” by Adaptive Wiener Filter. (d): Enhanced denoised image (PSNR = 29.8, SSIM = 0.77). The images were cropped between $(100, 100)$ and $(400, 400)$.

The residual filtering step can be interpreted as a regularization step. Also, the modification of the non-local means to adjust the variance of the residual in Theorem 5.7 of [16] can be seen as applying Lee's filter on the residual image:

$$\mathscr{D}_2(\mathbf{r}) \quad = \quad \frac{\max(0, s_r^2 - \sigma^2)}{s_r^2}\mathbf{r} \tag{8.35}$$

and adding the result to the denoised image $\mathbf{d}$.

We consider image enhancement to be achieved if the image quality obtained by adding the denoised residual is greater than the image quality obtained by simply denoising the image with either of the two methods. Here we show an example where (8.34) is used successively. Gaussian white noise of standard deviation $\sigma = 15$ was first added to create a noisy image. This image was then denoised with the Total Variation Minimization [94] algorithm ($\mathscr{D}_1$). The residual image was then denoised with the Adaptive Wiener Filter [63] (Matlab's `wiener2`) ($\mathscr{D}_2$). The results are shown in Fig. 8.10. The PSNR and the SSIM indices for $\mathscr{E}(\mathbf{d})$ are greater than the quality measurements obtained by processing the image with either $\mathscr{D}_1(\mathbf{y})$ or $\mathscr{D}_2(\mathbf{y})$ alone.

In general, simply taking a traditional denoising algorithm to denoise the residual and then adding back the result to the denoised image does not guarantee an improvement in the quality of the denoised image. The point, however, is that with a good no-reference or partial-reference quality measure (cf. Section 8.1), we can at least perform this step and then verify whether or not an improved image is obtained.

8.3.2 Iterative Schemes and Statistical Tests

We now present an iterative scheme of the above denoising procedure. Let $\mathscr{Q}$ denote a no-reference quality measure. Then the larger that $\mathscr{Q}(\mathbf{d})$ is, the higher the quality of $\mathbf{d}$. We iterate the algorithm J times and then choose the image with the best quality. As such, the algorithm will always produce an image of at least the same quality as that of the denoised image.

1. Set $j \leftarrow 1$, $\tilde{\mathbf{d}}_{(0)} \leftarrow \mathbf{y}$ and $\mathbf{d}_{(0)} \leftarrow 0$;
2. Denoise the image: $\mathbf{d}_{(j)} \leftarrow \mathscr{D}_1(\tilde{\mathbf{d}}_{(j)})$;
3. Compute the residual: $\mathbf{r}_{(j)} \leftarrow \mathbf{y} - \mathbf{d}_{(j)}$;
4. Denoise the residual: $\tilde{\mathbf{r}}_{(j)} \leftarrow \mathscr{D}_2(\mathbf{r}_{(j)})$;
5. Add it back to the denoised image: $\tilde{\mathbf{d}}_{(j)} \leftarrow \mathbf{d}_{(j)} + \tilde{\mathbf{r}}_{(j)}$;
6. While $j < J$, increment $j \leftarrow j + 1$ and go to step 2.
7. Find j that maximizes $\mathscr{Q}(\mathbf{d}_{(j)})$ or $\mathscr{Q}(\tilde{\mathbf{d}}_{(j)})$;
8. If $\mathscr{Q}(\mathbf{d}_{(j)}) > \mathscr{Q}(\tilde{\mathbf{d}}_{(j)})$ return $\mathbf{d}_{(j)}$, else return $\tilde{\mathbf{d}}_{(j)}$.

Denoising could be further improved by using a statistical test to determine which parts of the residual need to be added back to the denoised image. If one of the hypothesis tests is rejected, it is likely that the residual is not Gaussian white noise, hence we may consider that the denoising was insufficient. In this case we perform locally an enhancement such as (8.34).

The estimate of the noise variance can also be performed on the set of windows that passes the statistical test. This can be seen as a generalization of the simple, yet efficient, technique that estimates the noise variance from a flat region of the image determined by the user. This noise variance estimate can then be used for the subsequent image denoising in iterative schemes.

Chapter 9

Conclusion and Future Work

The goal of this thesis was to provide the mathematical tools for perceptually optimal image processing. In a first section we will summarize the contributions to the scientific community that were brought in this thesis. We will also refer to all our related publications. In a second section we will discuss different avenues for future research.

9.1 Summary of Contributions

CHAPTER 2 Normalized, relative or multiplicative metrics are different names found in the literature for metrics formed by a ratio of a metric over a normalizing function. We extended previous work on the subject by finding sufficient or necessary conditions for a metric to be "normalizable" and then shown how the SSIM index can be transformed into this kind of metric. Part of this work has been first presented in [13] and then has been published in [15].

Specifically, we found in Theorem 2.2.8 a sufficient condition for a metric $d(x, y)$ to be "normalizable" by a function $M(|x|, |y|)$. This condition generalizes the case $M(|x|, |y|) = f(|x|)f(|y|)$ and partially generalizes the case $M(|x|, |y|) = A_p^q(x, y)$. Indeed, from Corollary 2.2.10 the q-th power of generalized p-mean with a constant c is a metric for a certain range of values of p, q, c.

We also discovered two different necessary conditions for normalized metrics. Theorem 2.2.11 describes necessary conditions on $M(|x|, |y|)$ for a Ptolemaic metric to be "normalizable", whereas Theorem 2.2.12 imposes necessary conditions on d to be "normalizable" by a symmetric, positive and moderately increasing function. The automatic consequence of this theorem is that from all L^p-distance, only L^2-distance can be normalized by symmetric, positive and moderately increasing functions.

In Theorem 2.3.7 we then studied how to combine a vector of metrics acting on an orthogonal decomposition of $\mathbf{R}^N$ into a single metric. The trick was to apply a norm possessing the increasing property. For example, L^p-norms have such property.

This theoretical development was then applied to the SSIM index. Indeed, it was shown how each component of SSIM can be transformed into a metric via $\sqrt{1-S_i}$ for $i = 1, 2$. It was then hinted how these metrics can be combined into a vector-valued metric in a cone metric space or "scalarized" into a single SSIM metric via

$$\mathrm{SSIM}^*(\mathbf{x}, \mathbf{y}) = \sqrt{2 - \sum_i (S_1(\mathbf{x}_i, \mathbf{y}_i) + S_2(\mathbf{x}_i, \mathbf{y}_i))}. \tag{9.1}$$

We then demonstrated experimentally the striking fact that the derived SSIM metric behaves in a very similar fashion than the original SSIM index and even performs slightly better in psycho-physical tests.

As an application, the contractivity of the affine fractal transform was studied according to this new SSIM metric. This application was orally presented before in [9], but this work has not been published otherwise.

CHAPTER 3 In this chapter, we proved several useful properties for general quality measures of the form $q(\|\mathbf{x}\|, \|\mathbf{y}\|, \cos(\theta))$, which include components of the SSIM index and other normalized metrics. Most of these results were included in [15] for the specific case of the SSIM index.

First, we observed invariance under isometries (Property 3.1.1) and invariance under scaling for homogeneous normalized metrics (Property 3.1.2). Some symmetry properties (Property 3.1.3) allowed to reduce the dimensions to only two for normalized metrics and three for the SSIM index. We thus gave an analytical description and traced the level sets of pq-normalized metrics and of the SSIM index on zero-mean signals.

The next endeavour was to describe the convexity and quasi-convexity region of the components of the SSIM index and of the SSIM metric. The quasi-convexity region for the SSIM index was straightforwardly described from the level sets. We also provided the analytical formula of the region of convexity of SSIM without stability constant and laid down a procedure to find the region of convexity of SSIM for any given stability constant.

Table 9.1: Region of validity for the properties of the components of SSIM index and of the SSIM metric.

Properties	S_1	S_2	$\|(S_1,S_2)\|$	(S_1,S_2)
Distance Preserving	$\mathbf{R}_+$	$\mathbf{R}^{N-1}$	$\mathbf{R}^N$	$\mathbf{R}\times\mathbf{R}^{N-1}$
Quasiconcavity	$\mathbf{R}_+$	H_2	-	$\mathbf{R}_+\times H_2$
Concavity	R_1	R_2	$R_1\times R_2$	$R_1\times R_2$

where

$$\begin{aligned} H_2 &= \left\{\mathbf{x}-\bar{\mathbf{x}}\in\mathbf{R}^{N-1} : \langle \mathbf{x}-\bar{\mathbf{x}}, \mathbf{y}-\bar{\mathbf{y}}\rangle \geq \frac{-\epsilon_2}{2}\right\}, \\ R_1 &= [0,\sqrt{3}\bar{y}], \end{aligned}$$

R_2 : A tear drop shape pointed at $\mathbf{0}$ containing the ball centered at $y-\bar{y}$ and of radius $(\sqrt{3}-1)\|\mathbf{y}-\bar{\mathbf{y}}\|$ (see Theorem 3.3.10).

That last part involved more work. We needed to prove Lemma 3.3.9 to ensure that the convexity of a function on a region in R^2 will extend to its rotation in $\mathbf{R}^N$ by symmetry and monotonicity. In Theorem 3.3.11 and Theorem 3.3.12 we deduced the generalized convexity and quasi-convexity regions for vector-valued metrics as well as the convexity region for scalarized metrics. All these results are summarized in Table 9.1.

Finally, the quasi-convexity property was used to prove existence and uniqueness of the maximizer of the SSIM index of zero-mean signals on convex regions. In fact, an analytical solution was provided in the case of a projection of a point on a L^2-ball according to SSIM.

CHAPTER 4 We showed in the second chapter how the SSIM index can be transformed into a metric. Richter [88] proposed another way through the geodesic distance, but he did not solve the second order differential equation describing the path.

Our contribution was to solve for this geodesic path by taking advantage of the symmetries of SSIM to reduce the problem to two dimensions. In fact, we remarked that in this problem the Euler-Lagrange equation reduces to a first order differential equation and can thus be solved by separation of variables. The integration was carried analytically, but a one-dimensional non-linear equation needed to be solved to find some unknown constant. A rigorous proof of the existence and uniqueness of the solution was provided. These results are compiled in [10].

In practice, the parameterization of the geodesic has to be according to its arc length in order to describe meaningful paths and distances. We solved this constant speed condition (a first order differential equation) not only for the SSIM geodesic, but also for the quasi-hyperbolic and the α-quasi-hyperbolic metrics.

The SSIM-geodesic allows to describe an evolution between images such that the SSIM index between two successive images is maximized, thus modeling a perceptually linear change between images. An implementation of these geodesic paths as well as an example of application for the morphing between two images with different luminance and contrast was provided.

CHAPTER 5 In the fifth chapter, we described a solution of the SSIM-mean estimation problem. We found the point in $\mathbf{R}^N$ that maximizes the weighted sum of the SSIM index. We derived an analytical solution for the simplest case of two real points with no stability constant and estimated the perturbation induced by the addition of the stability constant.

This simple case was used to illustrate some striking facts about the solution of this problem. If the points are close enough, the problem becomes convex and the solution is the geometric mean between the points. But when the points are far apart, the objective function has several local maxima and more than one solution can be found. The direct consequence of this fact is that the evolution of the SSIM-mean computed via

$$f(\lambda) \quad = \quad \max_z \left(SSIM(\lambda x_1, z) + SSIM((1-\lambda) x_2, z) \right) \tag{9.2}$$

can have a discontinuity at $\lambda = 1/2$. Thus, contrary to the SSIM-geodesic, the evolution of the SSIM-mean is not perceptually smooth.

For the general case with zero-mean, an analytical solution can be found on every hyper-sphere of fixed radius. The problem thus reduces to one or two dimensions whether or not we are looking for zero-means signals. The numerical optimization of the reduced problem can lead to several local maxima depending on the distribution of the points x_is. On the other hand, we can expect that these local maxima will be sufficiently spaced from the local convexity property of SSIM. The simple idea was thus to provide uniformly sampled points of the domain as initial guesses of a local optimization scheme. Since the dimension of the problem is very small, the cost for multiple tries is reasonable. Also, from the multiplicative nature of the SSIM index, the samples should in fact be distributed on a logarithmic scale.

The SSIM-mean estimator is very important in a variety of applications. A simple way to see that, just replace mean by SSIM-mean in the most popular algorithms: we obtain a K-SSIM-means clustering and a non-local SSIM-means. This is only a small sample of the possibilities. In fact, in every imaging problem written in a stochastic form, the Minimal Mean Squared Error estimator could be replaced with the Maximal Structural Similarity estimator.

CHAPTER 6 The principal contribution in this chapter consisted in providing a solution for the SSIM-optimal approximation according to an orthogonal or a redundant basis. This solution ended out to be surprisingly simple. The SSIM-optimal coefficients are simply scaling up the L^2-optimal coefficients and the choice of basis is the same for both the orthogonal approximation and the approximate solution of the optimal redundant basis by the orthogonal matching pursuit algorithm.

The scaling acts as a contrast enhancement of the approximated image. In fact, for zero stability constant the variance of the approximated image is exactly matched with the variance of the original image. The difference between the SSIM-optimal and L^2-optimal algorithms will be manifested when a budget for the total number of coefficients or a threshold on the error of approximation is set in a local approximation scheme. In this case, even though the bases are chosen in the same order for each local approximation, the amount of bases for each local approximation might differ.

Three examples of application were briefly described in the chapter as well as in publications: image compression [14], image denoising by sparse representation [86, 84] and image super-resolution by sparse representation [84].

CHAPTER 7 The study of the affine self-similarity of images was extended by defining a penalty term to remove the “flat block bias”. This variance weighted L^2 is in fact exactly the SSIM index. The histogram of the self-similarity measures according to the SSIM index still indicates the presence of affine self-similarity in images, but to a lesser extend to what the RMSE indicates.

In fact, by comparing the histogram of the RMS of natural images with the histogram of the RMS of random noise with local mean and variance matching the natural images, we confirmed the hypothesis that the principal source of self-similarity in images comes from the low variance of the image blocks.

This work was first published in [12].

CHAPTER 8 The inspiration to study the image residual in image denoising came from the non-local means paper [16] where it is proposed to inspect the residual image to determine if it “looks” like noise. This way of doing did not seem rigorous, but from there came the idea to apply statistical test to automatically verify if the residual image behaves like noise.

A heuristic estimate of the MSE, PSNR and SSIM index between an original image and a denoised image was also proposed from the sole knowledge of the noisy image and its denoised version. The idea was to combine various empirical estimates of the corvariance between the noise and the residual image. It was then verified that the estimated measures predict well the full measures on average.

Finally, it was outlined how any image denoising algorithm could be potentially enhanced by applying statistical tests on the residual image and by filtering the residual image. Since its publication in 2009 [11], this idea inspired some others to use similar techniques [83, 121, 25].

9.2 Future Work

This thesis is the foundation of a far reaching research program aiming to incorporate better image, distortion and perception models in image processing applications. In future years, it could be extended to other types of image distortions and to more refined models of images.

9.2.1 Fundamental Research

Characterization and properties of normalized metrics

Our work done on normalized metrics constitutes the first synthesis on the subject, but the full characterization of normalized metric spaces is still lacking. Intuitively, it seems that a certain type of convexity is needed for the normalizing function as it is the case for $M(|x|,|y|) = f(|x|)f(|y|)$ or in our own sufficient condition (Theorem 2.2.8), but what is the exact condition is still an open problem. Even for the particular case

$$M(|x|,|y|) \quad = \quad (|x|^p + |y|^p + c^p)^{q/p} \tag{9.3}$$

a necessary and sufficient condition is still to be found. The obvious angle of attack would be to study more deeply the proof of Hästö [50] for the case $c = 0$.

One particular aspect that has not been fully studied is if the mathematical properties of the classical Structural Similarity index (1.21) can be extended to more general forms of SSIM or to other normalized metrics. In particular, we only touched the surface of the properties of the SSIM index with general exponents α and β. It is not known yet if we can also prove that they can be transformed into metrics and what are their region of convexity and quasi-convexity. In fact, we have not even answered the question if $\sqrt{1 - SSIM}$ itself is a metric. One can argue that it is not so much a problem of importance since in many applications we work with zero-mean blocks and that anyway our proposed approximation works at least as well as the original SSIM index. Nevertheless, numerical simulation seems to indicate that $\sqrt{1 - SSIM}$ may be indeed a metric. However, to prove this fact seems arduous since the developed tools on normalized metrics do not apply directly here.

Similar questions can be raised for the various generalization of the SSIM index (MS-SSIM [119], CW-SSIM [95] and IW-SSIM [115]). A preliminary result indicates that the Complex-Wavelet SSIM should be slightly modified if we want even the simplest property of identity for metrics.

Optimal CW-SSIM image processing

The CW-SSIM allows to assess the quality of images that are slightly shifted, rotated or zoomed. However, it does not quantify the perceptual loss for shifts and instead assumes perfect quality for coefficients of the same magnitude and of a difference phase.

In order to correct this problem, let us define a function $f(\theta)$ for $-\pi \leq \theta \leq \pi$ such that $f(\theta) \leq 1$ with $f(\theta) = 1$ if and only if $\theta = 0$. For example, $f(\theta) = \cos(\theta)$ satisfy this condition. Let c_x and c_y be the complex coefficients of a linear transform of, respectively, the images $\mathbf{x}$ and $\mathbf{y}$. Then we write θ_{c_x,c_y} as the angle between c_x and c_y. A variation of the CW-SSIM is then defined as

$$\text{CW-SSIM}^*(\mathbf{x}, \mathbf{y}) = \frac{2|\langle c_x, c_y \rangle| f(\theta_{c_x,c_y}) + K}{\|c_x\|^2 + \|c_y\|^2 + K}, \tag{9.4}$$

where $K \geq 0$ is a stability constant.

Theorem 9.2.1. *Let c_x and c_y be the coefficients obtained after a complex-valued transform of the image $\mathbf{x}$ and $\mathbf{y}$, respectively. Then*

$$\textit{CW-SSIM}^*(\mathbf{x}, \mathbf{y}) = 1 \iff c_x = c_y. \tag{9.5}$$

Moreover, if the transform is invertible, then

$$\textit{CW-SSIM}^*(\mathbf{x}, \mathbf{y}) = 1 \iff \mathbf{x} = \mathbf{y}. \tag{9.6}$$

Proof. If $\mathbf{x} = \mathbf{y}$, then necessary $c_x = c_y$. It implies that $|\langle c_x, c_y \rangle| = \|c_x\|^2 = \|c_y\|^2$ and $\theta_{x,y} := \text{ang}(\langle c_x, c_y \rangle) = 0$. It follows automatically that CW-SSIM$^*(x, y) = 1$.

If CW-SSIM$^*(x, y) = 1$, then its numerator equals its denominator:

$$2|\langle c_x, c_y \rangle| f(\theta_{x,y}) + K = \|c_x\|^2 + \|c_y\|^2 + K. \tag{9.7}$$

But from the Cauchy-Swartz inequality

$$\begin{aligned} \|c_x\|^2 + \|c_y\|^2 &\geq 2\|c_x\|\|c_y\| && (9.8) \\ &\geq 2|\langle c_x, c_y \rangle|. && (9.9) \end{aligned}$$

Comparing with the previous equation, it implies that $f(\theta_{c_x,c_y}) \geq 1$, hence $\text{ang}(\langle c_x, c_y \rangle) = 0$. We thus have $c_y = ac_x$ with $a > 0$. Substituting this expression in the equation comparing the numerator and the denominator of the modified CW-SSIM (9.7), we conclude that $a = 1$. When the transform is invertible, we have $\mathbf{x} = \mathbf{y}$ as well. □

9.2.2 Applications

Search in Metric Spaces

An important application of a perceptual metric is to perform nearest neighborhood search. This enters in the computer vision problem of entry retrieval in an image database. The problem goes as follows: Given a database $\mathscr{D}$ of size n and a query q, find the element in $u \in \mathscr{D}$ such that $d(q, u)$ is minimized, or at least very close to the minimum. A related problem is to find the k nearest neighbors of a given query. Both of these problems can be solved by finding all the points such that $d(q, u) < r$ with r a range value adjusted iteratively so that the desired number of neighbors are retained. This is this formulation that we will discuss in this section.

In this problem, the size of n is assumed to be very large (i.e. $n > 10^4$) and the cost of computing a distance is generally considered to dominate the algorithmic complexity [26]. Thus, a search strategy is needed to solve this problem efficiently. There is a vast literature on the subject (see [26] for a review). The vast majority of these techniques can be summarized in two categories: pivot-based or partition-based.

In the first class of techniques, the triangular inequality allows to bound the distance between elements not yet compared. Indeed, the distance between the query q, a pivot p

and any other point $u \in \mathscr{D}$ such that $d(q,u) < r$ can be related via

$$\begin{aligned} d(p,q) &\leq d(q,u) + d(p,u) < d(p,u) + r \text{ and} && (9.10) \\ d(p,u) &\leq d(q,u) + d(p,q) < d(p,q) + r. && (9.11) \end{aligned}$$

Thus any point u that does not satisfy

$$d(p,q) - r \leq d(p,u) \leq d(p,q) + r \qquad (9.12)$$

should be eliminated as a candidate for the nearest neighbor of q.

Alternatively, given a set of centers $\{c_i\}_{i=1}^m$, the dictionary is partitioned in m equivalence classes (i.e. clusters). For example, one could divide the dictionary in Voronoi-like regions according to the rule that u belongs to the i-th class if

$$d(u,c_i) \leq d(u,c_j) \text{ for all } i \neq j. \qquad (9.13)$$

From the triangular inequality, he have the following inequalities for any element u satisfying $d(q,u) < r$:

$$\begin{aligned} d(u,c_i) &\leq d(q,c_i) + d(q,u) < d(q,c_i) + r \text{ and} && (9.14) \\ d(u,c_j) &\geq d(q,c_i) - d(q,u) > d(q,c_j) - r. && (9.15) \end{aligned}$$

It implies that u does not belong the class represented by c_j if

$$d(q,c_i) + r \leq d(q,c_j) - r. \qquad (9.16)$$

It remains to describe a search strategy taking advantage of these criteria. The procedure is generally divided into a pre-processing (or indexing) step and the actual search. In the pre-processing step, the data is organized according to the pre-computed distance between elements. It is then that the centers or pivots are chosen. In the search step, candidates are pruned taking advantage of the triangular inequality until only a few candidates remain. This small set has to be searched exhaustively in order to uncover the nearest neighbor.

For example, if the data is divided into m classes of n/m elements, then the search will consist in comparing the query with each of the centers of these classes. The exhaustive search within the selected classes will require an extra n/m comparisons per class. A hierarchical organization of the dictionary could lead in theory to close to $\log_2(n)$ query comparisons. In practice, it ends up that the process is not so clean as the dimensionality of the data in the dictionary increases. This is the so-called curse of the dimensions. In this case, only approximate solutions guarantee a faster search than the exhaustive linear search [120].

Since the SSIM index is designed to assess the quality of distorted images and not to compare in general the conceptual similarity between two different images, it can not be directly used for this problem. On the other hand, it could be combined with a particular metric that performs well in image retrieval: the Earth's Mover Distance (EMD)[93]. This metric measures differences in histogram (or signature) of features as a transportation problem. The features represent local characteristics of images (e.g. color, texture or corners) and can be compared via a local similarity measure called the ground metric. In [93] it is shown that the EMD is truly a metric when the local similarity measure is also a metric. It leads us to propose the SSIM metric as the ground metric of the EMD.

Here is the outline of the procedure. First, partition the images $\mathbf{x}$ and $\mathbf{y}$ to be compared into clusters of similar image patches. Following the theory of Chapter 5, one can easily show how to perform a K-means algorithm according to the SSIM metric. To each cluster, assign a weight representing the relative size of the cluster. We obtain a set of couples $\{(\mathbf{p}_i, w_i)\}_{i=1}^n$ and $\{(\mathbf{q}_j, v_j)\}_{j=1}^m$ for each image, where $\mathbf{p}_i$ and $\mathbf{q}_j$ represent the image patches centering, respectively, the i-th cluster of $\mathbf{x}$ and the j-th cluster of $\mathbf{y}$ and w_i and v_j represent the proportion of patches belonging to, respectively, the i-th cluster of the $\mathbf{x}$ and the j-th cluster of the $\mathbf{y}$. The next step is to compare the distance between all possible pairs of clusters. Denote the SSIM distance between $\mathbf{p}_i$ and $\mathbf{q_j}$ by d_{ij}. The EMD between $\mathbf{x}$ and $\mathbf{y}$ is then found by solving the following linear problem:

$$\text{Minimize } \sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij} f_{ij}, \tag{9.17}$$

with the constraints

$$\begin{aligned} f_{ij} &\geq 0 \text{ for } 1 \leq i \leq n \text{ and } 1 \leq j \leq m, & (9.18)\\ \sum_{j=1}^{m} f_{ij} &\leq w_i \text{ for } 1 \leq i \leq n, & (9.19)\\ \sum_{i=1}^{n} f_{ij} &\leq v_j \text{ for } 1 \leq j \leq m \text{ and} & (9.20)\\ \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} &= 1. & (9.21) \end{aligned}$$

The cost function represents the total work done by the flow f. The first constraint ensures that the flow is positive while the last constraint accounts for the preservation of the mass. The second and third constraint informs us about the capacity of respectively $\mathbf{p}_i$ and $\mathbf{q}_j$.

Non-Rigid Image Deformation

We showed in Chapter 4 that a direct application of the SSIM-geodesic was for the creation of a perceptual linear change between two images. Other similar applications would be for the temporal interpolation or resampling of a sequence of images or of slices of volumetric data [3]. A major limitation for all these applications is that the SSIM index assumes that all images are well registered. So in general, the morphing between two objects according to the SSIM-geodesic will not lead to much better results than if the morphing was done according to a linear interpolation.

To correct this deficiency, a registration step will be needed before the computation of the SSIM-geodesic. For example, diffeomorphic mappings (i.e. differentiable bijections) between coordinate systems can be found to allow for non-rigid registration [4]. In fact, they are defined according to some geometric geodesics. The SSIM-geodesic thus complete the transformation process between two images by defining a so-called photometrical change.

The exact formulation of the problem leading to a diffeomorphic map ϕ_1 is as follows (see [4, 3]). Given a domain $\Omega \subset \mathbf{R}^2$ and two images $\mathbf{x} : \Omega \to \mathbf{R}$ and $\mathbf{y} : \Omega \to \mathbf{R}$, we seek for a diffeomorphic map $\phi_1 : \Omega \to \Omega$ that will minimizes

$$\|\mathbf{y} - \mathbf{x} \circ \phi_1^{-1}\|_2^2. \tag{9.22}$$

To do so, we introduce an evolution of diffeomorphic maps ϕ_t with $0 \leq t \leq 1$ and $\phi_0 = \mathrm{Id}$ the identity map, i.e. $Id(x) = x$ for all $x \in \Omega$. Associated with this family of diffeomorphic mappings is the family of vector fields $\mathbf{v}_t = \nabla\phi_t$ for $0 \leq t \leq 1$. We equip the vector field with some norm $\| \cdot \|_V$ measuring its smoothness, thus ensuring the obtention of a diffeomorphic map. The variational problem

$$\hat{\mathbf{v}} \quad = \quad \arg\min \int_0^1 \|\mathbf{v}_t\|_V^2 dt + \alpha\|\mathbf{x} \circ \phi_1^{-1} - \mathbf{y}\|_2^2, \tag{9.23}$$

subject to

$$\nabla\phi_t \quad = \quad \mathbf{v}_t(\phi_t) \tag{9.24}$$

will then lead to a smooth evolution of vector fields describing the local change of coordinates minimizing the distance between $\mathbf{x}$ and $\mathbf{y}$. From this evolution of vector fields one can then find the geodesic evolution of diffeomorphisms ϕ_t starting from $\phi_0 = \mathrm{Id}$.

The L^2-norm in (9.23) can sometimes be justified if one compare an image template with a noisy measurement. But if the goal is to register two clean images, a perceptual distance such as the SSIM index would be more appropriate. In fact, the L^2-distance have already been replaced by various similarity measure in computer vision applications [3]. But note that these formulations do not account for intensity changes in images.

A geodesic distance combining both geometric and photometric (i.e. intensity) distortions was proposed in [72]. The SSIM-geodesic could not be directly inserted in their framework since it is not defined from a norm, but one could combine the SSIM-geodesic with the geometric geodesic that is found from (9.23) and (9.24). The parameter α will then control the weight between the geometric and the photometric geodesics. Indeed, for $\alpha \to 0$, the vector field $\hat{\mathbf{v}} = \mathbf{0}$ minimizes the variational problem given in (9.23), thus only intensity changes will occur. On the other hand, when $\alpha \to \infty$, the diffeomorphic map will seek for the best possible match between $\mathbf{x} \circ \phi_1^{-1}$ and $\mathbf{y}$, thus reducing the need to correct the intensities.

Perceptual image denoising: A general framework

Although the two previous applications remain exciting, the objective in mind while conducing this doctoral research was to develop state-of-the-art image denoising algorithms by optimizing a perceptual criterion. Here is a general framework inspired in part by K-SVD denoising [40], Block Matching 3-D (BM3D) [31] and collective sensing [64].

1. Initialize $\mathbf{d} = \mathbf{y}$, $\sigma := \sigma_y$ and $k = 0$.
2. Construct a self-similar representation of $\mathbf{d}^{(k)}$ by matching similar blocks according to the local full-reference quality metric $q(\cdot, \cdot)$.
3. Locally denoise the image blocks $\mathbf{d}_i^{(k)} := \mathscr{D}_1(\mathbf{y}_i)$ with the help of the non-local representation.
4. Compute the residual blocks $\mathbf{r}_i^{(k)} := \mathbf{y}_i - \mathbf{d}_i^{(k)}$.
5. Test if the residual blocks follow the model of the noise.
6. Update the estimate of the variance of the noise.

7. Filter the rejected residual blocks $\tilde{\mathbf{r}}_i^{(k)} := \mathscr{D}_2(\mathbf{r}_i^{(k)})$.

8. Test again if the residual blocks follows the model of the noise.

9. Update again the estimate of the variance of the noise.

10. Regularize the denoised blocks corresponding to the rejected residual blocks $\mathbf{d}_i^{(k+1)} := \mathbf{d}_i^{(k)} + \tilde{\mathbf{r}}_i^{(k)}$.

11. Aggregate the local estimates into a single image $\mathbf{d}^{(k+1)}$.

12. Repeat from Step 2 with $k := k + 1$ until convergence.

We give more details on the meaning of each step. The second step corresponds to the block matching step in the BM3D algorithm or the computation of distance weights $w_{i,j}$ in the non-local means (NLM) algorithm. For every block, we search for up to P blocks that are similar, where P is a moderately large number to be determined. By similar, we mean that $q(\mathbf{d}_i, \mathbf{d}_j) \leq \delta$ for some δ. For example, the SSIM index was used by Rehman in his SSIM-NLM algorithm [85].

Remark that as it is done in Chapter 7 and contrary to the non-local means algorithm, the search for similar blocks does not have to be only with simple translation of blocks, but could as well include affine transforms, or even contractions and rigid transforms of the blocks as in fractal coding. The fact that most of the affine self-similarity in images is related to flat blocks does not mean that higher variance image blocks can not be matched. Indeed, it is intuitively clear that edges can be matched together. One could even argue that textures are not only statistically self-similar (in the sense that they share the same probability distribution), but that they are also self-similar in the appropriate resolution since they reduce to edges in high resolution and to flat regions in a low resolution.

The local denoising step can employ algorithms such as the NLM or BM3D. Alternatively, Steps 2 and 3 together can be seen as a mix of the construction of a dictionary by K-SVD (or simply K-means clustering) and of the matching pursuit. Note that a SSIM-based version of the K-means algorithm can be easily deduced from the SSIM-mean (see Chapter 5), but it is less clear how the K-SVD algorithm could be modified into a SSIM-optimal algorithm.

A more Bayesian approach would be to interpret the matched blocks as a prior probability distribution on the block to be denoised. The denoising could then be seen as a statistical estimation problem. For the NLM algorithm, the mean indeed corresponds to the Minimal Mean Square Estimator [91]. This could be replaced by a Maximal SSIM estimator following the work in Chapter 5.

The residual testing and the noise variance estimation were briefly described in Chapter 8. It remains to determine which statistical test is the most appropriate or even if they are appropriate at all. Indeed, to keep some statistics about the independence and the distribution of the residual blocks could be more informative than the dichotomic decision to reject or not to reject a certain block.

For the choice of the residual filter, Lee's filter seem a good first try. A SSIM-optimal alternative would be to project the approximation $\mathbf{d}_i$ on the L^2-ball of variance σ and centered at $\mathbf{y}_i$ as described at the end of Chapter 3. Other choice of filters would need to be able to detect structure in possibly highly noisy patches. One direction would be to study the algorithms employed in astronomical image denoising.

There are two classical ways to perform the aggregation. Either the center pixel is retained as in the non-local means algorithm or a weighted average of the blocks is computed as in the K-SVD denoising or the BM3D algorithm. The weights could depend on the confidence we have on the denoising of a certain block as indicated by a statistical test. A SSIM-based approach would seek for the global image that will maximize the sum of the local SSIM between each block. This should be found with a small modification of the SSIM-mean estimator of Chapter 5.

A convergence criterion is generally set according to the relative change between two successive iterations. There might be no guarantee of the convergence of the algorithm. One way to force convergence is to cool down the denoising parameters. Indeed, several denoising algorithms have one parameter (call it λ) controlling the strength of the denoising. For $\lambda = 0$, no denoising at all is performed, whereas for $\lambda \to \infty$ a completely flat image is obtained. So by gradually decreasing the value of λ in the iterative scheme, the algorithm will eventually freeze (i.e. converge) to some solution. This might not be necessary the optimal solution in the sense of our quality metric, but with a sufficiently slow cooling, the iterative scheme will have the chance to stop the denoising at the right moment.

A difficulty could be to control the computational cost, particularly in the non-local processing (Step 2), but we do not believe that it is an essential difficulty for several reasons. First, parallelization of the processing on local blocks is very well possible. Second, the organization of the blocks according to their metric distance can save the need to compare all pair of blocks as it is generally done in image retrieval. Notice that this fact is known by the fractal imaging community [96], but does not seem to have been noticed by non-local image processors. Finally, the self-similar representation of the previous iteration could be used as an initial guess for the next step, thus effectively directing the search.

Appendix A

Psychovisual Testing

In psycho-physical experiments, subjects are asked their opinion on the perceived quality of distorted images. These experiments not only serve to compare the subjective performance of image processing algorithms but also allow the comparison of image quality assessment (IQA) measures. With this last goal in mind, the procedure is as follows:

Gathering of the test images. A set of test images is chosen. Ideally, images should represent a wide range of content within the set of images relevant for the desired application. It is important that the test images are independent from the images used for the tuning of the image processing algorithms and the IQA measures. This condition is hard to verify in practice when the test images are publicly available and, essentially, depends on the honesty of the researchers.

Simulation of image distortion at different levels. Each test image is artificially altered with a different kind of distortion to create several distortion levels. The range of distortion levels should span from just noticeable difference to severe image degradation. The types of distortion chosen depend on the application in mind. But generally, the aim is to simulate the kind of image distortions that one would encounter in practice. That is, different kinds of artifacts, noise and blur are simulated.

Selection of the subjects. To be statistically reliable, psycho-physical experiments need a number of subjects that is big enough, i.e. more than twenty-five. Again, the choice of subjects should normally be a sample of the population of interest. In theory, it should be non-expert subjects tested for normal vision. But in practice, the subjects are often the researchers and their students because of convenience.

Preparation of the controlled testing environment. In order for the experiment to be reproducible, the testing environment needs to be accurately specified. It includes the type of display monitors used, the calibration parameters of these monitors, the set up of the room (in particular the distance from the subjects to the screen and the ambient lighting condition), and the procedure of the experiment itself.

The actual psycho-physical testing. Different procedures are available in literature for carrying on the actual psycho-physical testing. The standard method proposed by the Video Quality Expert Group (VQEG) [106] can easily be adapted for still images. It is a double stimuli experiment with a continuous grading scale. Adjectives for five different gradings are provided to help the scoring: excellent, good, fair, poor and bad.

Processing of the opinion scores. Before computing the Mean Opinion Scores (MOS), it might be necessary to discard the outliers, particularly if the size of the experiment is small. The raw scores are then normalized to reduce bias between subjects. The MOS are computed from the normalized scores. Finally, a renormalization of the MOS to the 0-1 scale is done. An additional information that can be extracted from the normalized raw scores is the MOS variance or equivalently, the confidence intervals for each score.

A.1 Methods for the comparison of IQA measures

Psycho-physical testing can be used to determine if objective IQA measures follow subjective rating. One aspect consists of gathering the MOS as previously described, whereas the other aspect is the computation of the objective measures.

Once the objective and subjective scores are obtained, their comparison can be carried on through different metrics. According to [106], three features need to be verified: accuracy, monotonicity and consistency. Accuracy measures the fitting of the model with the data. Monotonicity of a model is verified if the MOS increases whenever the prediction score increases. Finally, consistency counts how many times or by how much the model fails to predict the right MOS.

Pearson Linear Correlation Coefficient. The Pearson Linear Correlation Coefficient (CC) between the MOS and a fitted curve of the predicting model informs us of the accuracy of the model. The fitting is carried on through non-linear least square fitting of a logistic function with four parameters. This choice is somehow arbitrary but was observed to provide a good fitting generally. In the first version of their report [105], the VQEG recommends to take a generalzed logistic function with either four or five parameters. In the report for the second phase [106] however, it is suggested to take only three.

A logistic function with five parameters is defined as follows:

$$f(x;\beta) \quad = \quad \frac{\beta_1}{1+\exp(-\beta_2(x-\beta_3))}+\beta_4+\beta_5 x. \tag{A.1}$$

For a generalized logistic function with three and four parameters, one or both of the parameters β_4 and β_5 are dropped.

We remind that the Pearson Linear Correlation Coefficient (CC) is defined as

$$cc(x,y) \quad = \quad \frac{s_{x,y}}{s_x s_y}, \tag{A.2}$$

where $s_{x,y}, s_x$ and s_y are, respectively, the sample covariance between x and y and the standard deviation of x and of y.

Spearman Rank Correlation Coefficient. As a rank correlation, the Spearman's ρ measures the monotonicity between the predicted and the observed data. It is defined as the Pearson linear correlation coefficient of the ranking order of data. That is, the data $\{(x_i, y_i)\}_{i=1}^N$ is first transformed into $\{(k_i, l_i)\}_{i=1}^N$, where k_i is the rank of x_i and l_i is the rank of y_i. The Pearson linear correlation is then computed from these ranks. In the case of a tie between the rank of two or more data points, the average rank is taken for each one.

Kendall Rank Correlation Coefficient. Kendall's τ is another rank correlation that can also measure the monotonicity of the data. It is computed by counting the difference between the number of pairs where the monotonicity is respected and where it is not. If there is any tie in a pair, this pair is ignored. The measure is then normalized with the number of possible pairs. Mathematically, it is written as

$$\begin{aligned} C &= \#\{(x_i, y_i); (x_j, y_j) \big| (x_i > x_j) \text{ \& } (y_i > y_j) \text{ or } (x_i < x_j) \text{ \& } (y_i < y_j)\} && \text{(A.3)} \\ D &= \#\{(x_i, y_i); (x_j, y_j) \big| (x_i > x_j) \text{ \& } (y_i < y_j) \text{ or } (x_i < x_j) \text{ \& } (y_i > y_j)\} && \text{(A.4)} \end{aligned}$$

with

$$\tau(x, y) = \frac{C - D}{\frac{1}{2}N(N-1)}, \qquad \text{(A.5)}$$

where N is the size of the data.

Ratio of outliers. The outlier ratio (OR) is simply the proportion of outliers in the data. It is thus a measure of consistency. An outlier is defined as a predicted data that is away from the observed data more than twice its standard deviation:

$$OR = \#\{x_i \mid |x_i - y_i| > 2\text{std}(y_i)\}/N. \qquad \text{(A.6)}$$

Root Mean Square Error. The Root Mean Square Error (RMSE) also measures the consistency of the data in the sense that it weighs more heavily on the outliers. It can also be seen as a measure of accuracy. The RMSE is simply the squared root of the Mean

Square Error defined in the Introduction:

$$\mathrm{RMSE}(x,y) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2}. \tag{A.7}$$

A.2 IQA databases

Several image quality assessment databases are publicly available for the assessment of image quality measures. Associated with these databases are the opinion scores of the perceived image quality. We describe five of the most popular ones: the LIVE database [102] from the Laboratory of Image and Video Engineering at the University of Texas at Austin, the Tempere Image Database (TID-2008) [80] from the Tempere University of Technology, the IVC database [62] from the Image and Video Communication research group of the Institut de Recherche en Communications et Cybernétique of Nantes, the Categorical Subjective Image Quality (CSIQ) database [61] from the Computational Perception and Image Quality Lab of the Oklahoma State University and the Toyama database [52] from the Toyama University.

In Table A.1, we summarize the number of images, distortion types, distortion levels and subjects for each database. A second table (Table A.2) describes the distortion type included in each database. It is clear from these tables that LIVE database and TID-2008 database are the two most extensive one and this is the two databases we will adopt in our experiments. Note that the ratings from the TID-2008 database were not gathered in a controlled laboratory environment but instead from different subjects on their home computer. In one way it could be argued that the results are thus not reliable, but on the other way one could answer that this is the most natural setting for image quality assessment. Moreover, the analysis in [79] indicates that the sheer number advantage eventually allows for better quality of the results. Following this line, it has recently been suggested to perform this kind of psycho-visual task online through what it is called crowd-sourcing [87].

Table A.1: Number of test images, distortion type, distortion levels, distorted images, subjects and ratings in various psycho-physical databases.

Database	Test images	Distortion types	Distortion levels	Distorted images	Subjects	Ratings
CSIQ	30	6	3-5	866	3	2601
IVC	10	4	5-6	235	15	3525
LIVE	29	5	6-8	779	20-29	22457
TID-2008	25	17	4	1700	833	256428
Toyama	14	2	6	168	16	2688

Table A.2: Type of distortion in various psycho-physical databases.

Distortion type	CSIQ	IVC	LIVE	TID-2008	Toyama
JPEG	X	X	X	X	X
JPEG-2000	X	X	X	X	X
LAR-coding		X			
quantization				X	
packet loss				X	
transmission error				XXX	
block-wise distortion				X	
fast fading			X		
BM3D denoising				X	
Gaussian blur	X	X	X	X	
white noise	X		X	X	
impulse noise				X	
non-white noise	X			XXXX	
contrast change	X			X	
luminance shift				X	

Appendix B

Matlab Codes

```
function normalizedgeodesic(x,y,q,C,color)

% =========================================================
% Geodesic path between ’x’ and ’y’ in R^2 according to a normalized metric
% with power ’q’ and stability constant ’C’ (display result according to
% option specified by ’color’)
% Method: see Chapter 4.

%% transform in polar coordinates
xc = x(1)+i*x(2);
yc = y(1)+i*y(2);
rx = abs(xc);
ux = angle(xc);
ry = abs(yc);
uy = angle(yc);

%% define z
t = 0:0.01:1;
%%
if q == 1 && C == 0
    z = rx.^(1-t)’.*ry.^(t)’.*exp(i*(ux*(1-t)’+uy*t’));
elseif 0 < q && q < 1 && C == 0
    A = atan( (-sin((q-1)*ux)+sin((q-1)*uy)*(rx/ry)^(q-1))...
     /(cos((q-1)*ux)-cos((q-1)*uy)*(rx/ry)^(q-1)) )/(q-1);
    theta = ux*(1-t)’+uy*t’;
    r = rx*((sin((q-1)*(theta+A)))./(sin((q-1)*(ux+A)))).^(1/(q-1));
    z = r.*exp(i*(theta));
elseif q == 1 && C > 0
    K = @(r,C) (r.^2+C^2)./r.^2;
    f = @(B,r,C) B./sqrt(1-B.^2).*atanh(sqrt(1-B.^2)...
     ./sqrt(1./K(r,C)-B.^2)) - asin(B.*sqrt(K(r,C)));
    eqn1 = @(B) f(B,ry,C) - f(B,rx,C) - uy + ux;
    B0 = (max(rx,ry))^2/((max(rx,ry))^2 + C^2)/2;
    F1 = @(Br,Bi) real(eqn1(Br+i*Bi));
    F2 = @(Br,Bi) imag(eqn1(Br+i*Bi));
    FF = @(B) [F1(B(1),B(2));F2(B(1),B(2))];
    Bv = fsolve(FF,[B0;0]);
    B = Bv(1)+i*Bv(2);
    A = f(B,rx,C) - ux;
    rt = ry:(rx-ry)/99:rx;
    theta = real(f(B,rt,C) - A);
    z = rt'.*exp(i*(theta'));
else
    z = (1-t).*(x(1)+i*x(2)) + t.*(y(1)+i*y(2));
    disp('Case not covered, displaying: case q=0')
end


Mx = max(real(z(:)));
mx = min(real(z(:)));
My = max(imag(z(:)));
my = min(imag(z(:)));

hold on, plot(real(z(:)),imag(z(:)),color,x(1),x(2),'xk',y(1),y(2),'xk'),...
    axis('equal'), axis([mx-1/2 Mx+1/2 my-1/2 My+1/2]),...
    %title(['geodesic: q=',num2str(q),', C=',num2str(C)])

% ==================================================================
function [] = EXgeodesicSSIM()

% Example of the geodesic evolution between two images according
% to SSIM and L^2

%% define x and y
X = imread('images/barbara.png');
x = double(X)/255;
xv = x(:);
xv = (xv-mean(xv))*0.4+mean(xv)*0.4;
x = reshape(xv,512,512);

xv = x(:);
Y = imread('images/lena.png');
y = double(Y)/255;
yv = y(:);

%% transform in polar coordinates
```

```
ux = mean(xv);
uy = mean(yv);
uxv = ux(ones(numel(xv),1));
uyv = uy(ones(numel(yv),1));
rx = sqrt((xv-uxv)'*(xv-uxv));
xn = (xv-uxv)/rx;
ry = sqrt((yv-uyv)'*(yv-uyv));
yn = (yv-uyv)/ry;
omega = acos(xn'*yn);

%%
nm = @(x) sqrt(x'*x);
rx2 = nm(xv);
xn2 = xv/rx2;
ry2 = nm(yv);
yn2 = yv/ry2;
omega2 = acos(xn2'*yn2);
s
for t = 0:0.01:1
    sv = uxv.^(1-t)'.*uyv.^(t)' + rx.^(1-t)'.*ry.^(t)'...
     .*(sin(omega*(1-t)')/sin(omega).*xn' + sin(omega*(t)')/sin(omega).*yn');
    s = reshape(sv,512,512);
    m = x.*(1-t)+y.*t;
    figure(33);
    subplot(1,2,1); imshow(s,[0,1]),title(['geodesic-SSIM at t=',num2str(t)])
    subplot(1,2,2); imshow(m,[0,1]),title(['geodesic-L2 at t=',num2str(t)])
    pause(0.01)
 end

% ============================================================
function [z] = ssimmean(X,w,C)

% Compute the weighted mean of K vectors according to SSIM (see Chapter 5)
% 0 < x <= 1
% Input
% X: K x N are K vectors of length N
% w: K x 1 are K weights
% C: 1 x 1 or 2 x 1 constants for zero-mean or non zero-mean signals
% Temporaty Variable
% z: M x 1
% Output
% mz:  1 x 1
% Dependencies: funopt1, funopt2, globalmin
% Examples:
% z = ssimmean(rand(5,2),ones(5,1),0)
% z = ssimmean(rand(5,2),ones(5,1),[0 0])
% r = rand(5,2); z = ssimmean([max(r')' min(r')'],ones(5,1),[0 0])

cas = length(C);
[K,N] = size(X);

if cas == 1
    mf = @(z) -funopt1(X,w,C,z);
    mz = globalmin(mf,0,N);
    X0 = X;
else
    mf = @(r,b) -funopt2(X,w,C(1),C(2),r,b);
    mz = globalmin(mf,[0 0],[N N]);
    Xm = mean(X,2); % K x 1
    X0 = X - repmat(Xm,1,N); % K x 1
end

ip = X0*X0'; % K x K
nx = diag(ip); % K x 1

if cas == 2
    S1 = (2*mz(2)*Xm+C(2))./(mz(2)^2+Xm.^2+C(2));  % K x 1
else % dm == 1
    S1 = ones(K,1);
    mz(2) = 0;
end
ws = w.*S1; % K x 1
wp = ws*ws'; % K x K

%%
d = nx+mz(1)^2+C(1); % K x 1
```

```
den = d*d'; % K x K
sx = sqrt(sum(sum(wp.*ip./den))); % 1 x 1
rat = ws./d;

if sx>0
    u = (rat'*X0)/sx;
else
    u = zeros(2,1);
end

z = mz(1)*u + mz(2);

% ============================================================
function v = funopt1(X,w,C,z)

% function to optimize (zero-mean case)
M = length(z);
ip = X*X'; % K x K
wp = w*w'; % K x K
nx = diag(ip); % K x 1
K = length(nx);
b = repmat(nx,1,M)+repmat(z.^2,K,1)+C; % K x M
den = zeros(K,K,M);
for m=1:M
    den(1:K,1:K,m) = (b(1:K,m))*(b(1:K,m))'; % K x K x M
end
sx = sqrt(sum(sum(repmat(wp,[1 1 M]).*repmat(ip,[1 1 M])./den,1),2)); % M x 1
v = 2*z'.*squeeze(sx) + C*(sum(repmat(w,1,M)./b,1))'; % M x 1

% =============================================================
function v = funopt2(X,w,C1,C2,r,b)
% function to optimize (non zero-mean case)
% 0 < x <= 1
% Input
% X: K x N
% w: K x 1
% C1: 1 x 1
% C2: 1 x 1
% r: M x 1
% b: M x 1
% Output
% v:  1 x 1

r = r(:);
b = b(:);

M = length(r);
[K,N] = size(X);

Xm = mean(X,2); % K x 1
Xmm = repmat(Xm,1,M);
X = X - repmat(Xm,1,N); % K x 1
ip = X*X'; % K x K
nx = diag(ip); % K x 1

bb = repmat(b',K,1);
d = repmat(nx,1,M)+repmat(r'.^2,K,1)+C2; % K x M
S1 = (2*bb.*Xmm+C1)./(bb.^2+Xmm.^2+C1);  % K x M
ws = repmat(w,1,M).*S1; % K x M
wp = zeros(K,K,M);
den = zeros(K,K,M);
for m=1:M
   wp(1:K,1:K,m) = (ws(1:K,m))*(ws(1:K,m))'; % K x K x M
   den(1:K,1:K,m) = (d(1:K,m))*(d(1:K,m))'; % K x K x M
end
sx = sqrt(sum(sum(wp.*repmat(ip,[1 1 M])./den,1),2)); % M x 1
v = 2*r.*squeeze(sx) + C2*(sum(ws./d,1))'; % M x 1

% ====================================================================
function z = globalmin(fun,lb,ub)

%==================================
% Global minimizer in one or two dimensions
% Goal: To find a global minimum of a function
%       constrained on an hyper-cube
% Input: fun - function to minimize
%        lb - lower bound
%        ub - upper bound
% Output: z - minimizer
%===================================

dm = length(lb);

n = 10; % number of initial guesses
a = -4; % smallest point = 10^a

if dm == 1
    t = logspace(log10(max(10^a,lb)),log10(max(10^(a),ub)),n);
    tm = [10^a t(1:end-1)];
    tp = [t(2:end) ub];
    v = fun(t);
    vm = fun(tm);
    vp = fun(tp);
    ind = find((v-vm <= 0).*(vp-v >= 0));
    lind = length(ind);
    lm = zeros(1,lind);
    for ii=1:lind
        lm(ii) = fminbnd(fun,tm(ind(ii)),tp(ind(ii)));
        if v(ind(ii)) < fun(lm(ii)) % if local minimum found not better,
           lm(ii) = t(ind(ii));     % keep the middle point
        end
    end
    vlm = fun(lm);
    [M,mind] = min(vlm);
    z = lm(vlm == M);
    lmm = [lm(1:mind-1); lm(mind+1:end)];
    vlmm = [vlm(1:mind-1); vlm(mind+1:end)];

   figure(63), close(63), figure(63), hold on,
   h1 = plot(t,-v,'.b','MarkerSize',15);
   h2 = plot(lmm,-vlmm,'+g','linewidth',2,'MarkerSize',10);
   h3 = plot(z,-M,'+r','linewidth',2,'MarkerSize',10);
   axis1 = gca;
   set(axis1,'FontSize',16)
   xlabel('r','FontSize',16)
   ylabel('sum of SSIM','FontSize',16)
   legend([h1; h2; h3],'samples','local maxima','global maximum');
elseif dm == 2
     t1 = logspace(log10(max(10^a,lb(1))),log10(max(10^(a),ub(1))),n);
     tm1 = [10^a t1(1:end-1)];
     tp1 = [t1(2:end) 1];
     t2 = logspace(log10(max(10^a,lb(2))),log10(max(10^(a),ub(2))),n);
     tm2 = [10^a t2(1:end-1)];
     tp2 = [t2(2:end) 1];
     [tx,ty] = meshgrid(t1,t2);
     g = @(x) fun(x(1),x(2));
     v = fun(tx,ty);
     [twx,twy] = meshgrid(tm1,t1);
     [tex,tey] = meshgrid(tp1,t1);
     [tnx,tny] = meshgrid(t2,tm2);
     [tsx,tsy] = meshgrid(t2,tp2);
     vw = fun(twx,twy);
     ve = fun(tex,tey);
     vn = fun(tnx,tny);
     vs = fun(tsx,tsy);
     ind = find((v-vs <= 0).*(v-vn <= 0).*(v-ve <= 0).*(v-vw <= 0));
     lind = length(ind);
     lm = zeros(lind,2);
     for ii=1:lind
        opts = optimset('largescale','off');
        lm(ii,:) = fmincon(g,[tx(ind(ii)) ty(ind(ii))],[],[],[],[],...
            [twx(ind(ii));tny(ind(ii))],[tex(ind(ii));tsy(ind(ii))],[],opts);
        if v(ind(ii)) < g(lm(ii,:)) % if local minimum found not better,
            lm(ii,:) = [tx(ind(ii)) ty(ind(ii))];  % keep the middle point
        end
     end
     vlm = fun(lm(:,1),lm(:,2));
     [M,mind] = min(vlm);
     z = [lm(mind,1) lm(mind,2)];
     lmm = [lm(1:mind-1,1:2); lm(mind+1:end,1:2)];
     vlmm = [vlm(1:mind-1); vlm(mind+1:end)];
     v = reshape(v,n,n);
     figure(63), close(63), figure(63), hold on,
     h1 = mesh(tx,ty,-v,’linewidth’,2);
     h2 = plot3(z(1),z(2),-M,’+r’,’linewidth’,2,’MarkerSize’,10);
     h3 = plot3(lmm(:,1),lmm(:,2),-vlmm,’+g’,’linewidth’,2,’MarkerSize’,10);
     axis1 = gca;
     set(axis1,’FontSize’,16)
     xlabel(’r’,’FontSize’,16)
     ylabel(’b’,’FontSize’,16)
     zlabel(’sum of SSIM’,’FontSize’,16)
     legend([h1; h2; h3],’mesh of samples’,’global maximum’,’local maxima’);
end

%%  =================================================================
function [A]=OMPssim(D,X,T0)

%=============================================
% Sparse coding of a group of signals based on a given
% dictionary and specified number of atoms to use.
% input arguments: D - the dictionary
%                  X - the signals to represent
%                  errorGoal - the maximal allowed representation error for
%                  each siganl.
% output arguments: A - sparse coefficient matrix.
%=============================================
[n,P]=size(X);
[n,K]=size(D);
C = (0.03*255)^2*(n-1);
T2 = (n-1)*T0^2;
E2 = (1.15*T0)^2*n;
maxNumCoef = n/2;
A = sparse(K,P);
for k=1:1:P,
    indx = [];
    x = X(:,k);
    f = D’*x; % Fourier coefficients
    proj = f;
    nx = x’*x;
    nxhat2 = max(0,nx-T2);
%currSSIM = C./(nxhat2+C);
    currL2 = sum(x.^2);
    %TS = (C + sqrt(C^2+4*nx*(T2+nx+C)))/(2*(T2+nx+C));
    j = 0;
    %while currSSIMs<TS && j < maxNumCoef,
    while currL2>E2 && j < maxNumCoef,
j = j+1;
        pos=find(abs(proj)==max(abs(proj)));
        pos=pos(1);
        indx(j)=pos;
        a = pinv(D(:,indx(1:j)))*x; % L^2 coefficients
        z = D(:,indx(1:j))*a; % L^2-approximation
        F = a'*f(indx(1:j)); % approximation variance
        % solve quadratic equation for alpha
        cc = C+nxhat2;
        q = 1/2*(C+sqrt(C^2+4*F*cc));
        alpha = cc/q; % scaling factor
        %zs = alpha*z;
        %r = x-zs; % residual
        r = x - z;
        proj = D'*r;
%currSSIM = (2*x'*zs+C)./(nxhat2+zs'*zs+C);
        currL2 = sum(r.^2);
   end;
   if ~isempty(indx)
       A(indx,k)=a*alpha;
   end
end;
return;

%%  ===================================================================
function [H,d,p] = vwkstest(funk,data,weight,alpha)

%% ====================================================================
% Goal : Test the hypothesis that sets of data follow a specified
%        probability distribution at a certain significance level.
% Method : Implementation of a vectorized version of Kolmogorov-Smirnov
%          (K-S) test. The empirical cumulative probability function (cdf)
%          constructed from the data is compared with the true cdf of the
%          given probability distribution function (pdf). The maximum
%          difference between these two distributions is the
%          statistic to be compared between with the Kolmogorov-Smirnov
%          distribution (see Chapter 8).
% Input:
%   data : N x M (M series of length N)
%   funk : cdf function of the continuous pdf to be compared
%   weight: N x 1
%   alpha : significance level (generally 0<alpha<0.5)
% Output:
%   H : M x 1 vector of hypothesis test (0 for false, 1 for true)
%   p : M x 1 vector of p-values (0 <= p <= 1)
% Call example :
% ncdf = @(x) cdf('norm',x,0,1)
% [H,p] = vwkstest(ncdf,randn(100,1000),ones(100,1),0.05);
% Speed : For large matrices, this code is approximately 60 times faster
%          than performing a loop of Matlab's 'kstest' while retaining
%          double precision.
%% ====================================================================

N = size(data,1); % number of rows in data

%% Compute the cdf of the pdf to be compared at data points
data = sort(data);
ff = funk(data);

%% Compute the weighted empirical cdf
weight = weight/sum(weight); % normalize weights
f0n = [0; cumsum(weight)]; % empirical cumulative distribution function

%% Compute maximum distance
dt1 = max(abs(bsxfun(@minus,ff,f0n(1:N))),[],1); % distance at left
dt2 = max(abs(bsxfun(@minus,ff,f0n(2:N+1))),[],1); % distance at right
d = max(dt1,dt2); % maximum distance between ecdf and cdf

%% Estimate p-value (see Numerical Recipes, 3rd edition (2007), p.335)
```

```
en = sqrt(N);
z = (en+0.12+0.11/en)*d;
z0 = max(z,0.05); % 0.05 is the minimal value of z for a p-value < 1
c1 = pi^2/8; % constants in the K-S cdf series
c2 = 4/sqrt(pi);
y = exp(-c1./z0.^2);
py = 1 - c2*sqrt(-log(y)).*(y + y.^9 + y.^25 + y.^49); % case z < 1.18
x = exp(-2*z0.^2);
px = 2*(x - x.^4 + x.^9); % case z >= 1.18
p = (z<=0.05).*1+(z<1.18).*(z>0.05).*py+(z>=1.18).*px; % vectorized if-then

%% Test hypothesis
p = p’; % put into column format
H = (alpha >= p); % vectorized comparison

%% ==================================================================
function [H,r,p] = vwpctest(data1,data2,weight,nu,alpha)

%% ==================================================================
% Goal : Test the hypothesis that data1 and data2 are linearly independent
% Method : Implementation of a vectorized version of a linear correlation
% test  (see Chapter 8)
% Hypothesis : data1 and data2 are jointly normally distributed
% Input:
%   data1 : N x M (M series of length N)
%   data2 : N x M (M series of length N)
%   weight: N x 1
%   nu : 1 x 1 number of degrees of freedom
%   alpha : significance level (generally 0<alpha<0.5)
% Output:
%   H : M x 1 vector of hypothesis test (0 for false, 1 for true)
%   p : M x 1 vector of p-values (0 <= p <= 1)
% Call example :
% [H,p] = vwpctest(randn(100,1000),randn(100,1000),ones(100,1),0.05);
%% ==================================================================

%% Compute weighted Pearson’s correlation coefficient
weight = weight./sum(weight);  % normalize weight
```

```
mx = (weight'*data1);
my = (weight'*data2);
sxy  = weight'*(data1.*data2)-mx.*my; % covariance between data1 and data2
sx2  = weight'*(data1.*data1)-mx.*mx; % variance of data1
sy2  = weight'*(data2.*data2)-my.*my; % variance of data2
r = sxy./(sqrt(sx2.*sy2)+eps); % correlation

%% Compute statistic
t  = r.*sqrt(nu./((1-r+eps).*(1+r+eps)));

%% Estimate p-value (double sided test)
p = cdf('t',t,nu)'; % Student-t cdf

%% Test hypothesis
alpha = alpha/2;
H = (alpha >= p)|((1-alpha) <= p); % vectorized comparison

%%  =================================================================
function [H,r,p] = vwsctest(data1,data2,weight,alpha)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Goal : Test the hypothesis that data1 and data2 are linearly independent
% Method : Implementation of a vectorized version of a rank correlation
% test (see Chapter 8)
% Input:
%   data1 : N x M (M series of length N)
%   data2 : N x M (M series of length N)
%   weight: N x 1
%   alpha : significance level (generally 0<alpha<0.5)
% Output:
%   H : M x 1 vector of hypothesis test (0 for false, 1 for true)
%   p : M x 1 vector of p-values (0 <= p <= 1)
% Call example :
% [H,p] = vwsctest(randn(100,1000),randn(100,1000),ones(100,1),0.05);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[N,M] = size(data1);
nu = N-2;
```

```
%% sort and rank data
[ord1,ind1] = sort(data1);
[ord2,ind2] = sort(data2);

%% handle ties
rank1 = crank(ord1);
rank2 = crank(ord2);

%%
tind1 = N*repmat(0:M-1,N,1)+ind1;
tind2 = N*repmat(0:M-1,N,1)+ind2;

%% linear correlation test on the rank
[H,r,p] = vwpctest(rank1(tind1),rank2(tind2),weight,nu,alpha);

%%  =================================================================
function [H,p] = vwgtest(mi,N,nu,alpha)

%% G-test
G = 2*N*mi;
p = 1-chi2cdf(G,nu);
H = (alpha >= p);

%% === Vectorized and weighted information measures ======  %%
function [mi,Hx,Hy,Hxy] = vwMI(xv,yv,b,g)

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Goal : Find vectorized and weighted information measures (entropy, joint
% entropy and mutual information) of xv and yv)
% Method : Implementation of a vectorized version. Construct histogram
% with vwhist and then compute vectorized joint and marginal distribution
% in order to obtain the information measures (see Chapter 8).
% Input:
%   xv : N x M (M series of length N)
%   yv : N x M (M series of length N)
%   b: 2 x 1 number of bins for x and y respectively
%   g: N x 1 weigths
% Output:
%   mi : M x 1 vector of mutual information
%   Hx : M x 1 vector of entropy of xv
%   Hy : M x 1 vector of entropy of yv
%   Hxy : M x 1 vector of joint entropy of xv and yv
% Dependency: vwhist
% Call example :
% g = fspecial('gaussian',[10 10],1.5); g = g(:);
% x = floor(256*rand(100,1000));
% y = floor(256*rand(100,1000));
% whist = vwhist(x,y,[6 6],g);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[N M] = size(xv);

%% construct joint probability distribution and joint entropy of x and y
whist = vwhist(xv,yv,b,g);
sumhist = sum(sum(whist,1),2);
pij = bsxfun(@rdivide,whist,max(eps,sumhist));
Hxy = -sum(sum(pij.*log2(max(pij,eps))));

%% marginal distribution and entropy of x
pi = sum(pij,1);
Hx = -sum(pi.*log2(max(pi,eps)));

%% marginal distribution and entropy of y
pj = sum(pij,2);
Hy = -sum(pj.*log2(max(pj,eps)));

%% mutual information of x and y
pi = repmat(pi,[b 1 1]);
pj = repmat(pj,[1 b 1]);
mi = squeeze(sum(sum(pij.*log2(max(pij,eps)./max(pj.*pi,eps)),1),2));

%%  =================================================================
function whist = vwhist(x,y,b,g)
```

```
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Goal : Find the weighted joint histogram of x and y
% Method : Implementation of a vectorized version. First build a sparse
% matrix of ones indicating the values of (x,y)
% Input:
%   x : N x M (M series of length N)
%   y : N x M (M series of length N)
%   b: 2 x 1 number of bins for x and y respectively
%   g: N x 1 weigths
% Output:
%   whist : b(1) x b(2) x M (M histograms of dimension b(1) x b(2))
% Call example :
% g = fspecial(’gaussian’,[10 10],1.5); g = g(:);
% x = floor(256*rand(100,1000));
% y = floor(256*rand(100,1000));
% whist = vwhist(x,y,[6 6],g);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% check dimensions
[N M] = size(x);
ng = length(g);
if (size(y,1) ~= N) || (size(y,2) ~= M)
   disp(’error: dimension of x and y not matching’)
   return
end
if ng > N
    disp(’error: g longer than x’)
    return
end
if ng < N
    disp(’error: g shorter than x’)
    return
end

%% handle case of scalar b or L
if length(b) == 1
    b(2) = b(1);
end
```

```
%% binning
x = double(x);
mx = min(min(x));
Mx = max(max(x));
Lx = (Mx-mx)*(1+eps)+eps; % epsilon ensures that Lx > 0 and (y-mx)/Lx < 1
x = floor((x-mx)/Lx*b(1));

y = double(y);
my = min(min(y));
My = max(max(y));
Ly = (My-my)*(1+eps)+eps; % epsilon ensures that Lx > 0 and (y-mx)/Lx < 1
y = floor((y-my)/Ly*b(2));

%% vectorized values of (x,y)
B = b(1)*b(2);
x = B*repmat(0:M-1,N,1)+x*b(2)+y+1;
x = x(:);

%% indices of x and y
y = repmat((1:N)',1,M);
y = y(:);

%% counting vector
w = true(N,1);
w = repmat(w,1,M);

%% vectorized and weighted histogram
S = sparse(x(:),y(:),w(:),B*M,N);

%% clear some memory
clear w
clear x
clear y

%% weighted histogram
whist = S*g;
whist = reshape(whist,[b(1) b(2) M]);
```

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Processing*, 54(11):4311–4322, 2006. 125

[2] S. K. Alexander, E. R. Vrscay, and S. Tsurumi. A simple model for the affine self-similarity of images. In M. Kamel and A. Campilho, editors, *Proc. Int. Conf. on Image Analysis and Recognition*, volume 5112 of *LNCS*, pages 192–203. Springer, Heidelberg, 2008. 128, 130, 131, 135, 136

[3] B. B. Avants, C. L. Epstein, and J. C. Gee. Geodesic image interpolation: Parameterizing and interpolating spatiotemporal images. In *VLSM*, pages 247–258, 2005. 173, 174

[4] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younès. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vision*, 61(2):139–157, February 2005. 173

[5] C. Boncelet. Image noise models. In A. C. Bovik, editor, *Handbook of Image and Video Processing*. Academic Press, 2005. 4

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, 2004. 54

[7] R.P. Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973. 98, 102

[8] D. Brunet. Métriques perceptuelles pour la compression dimages (Étude et comparaison des algorithmes JPEG et JPEG2000). Master's thesis, Université Laval, Québec, Canada, 2007. 11, 16, 123

[9] D. Brunet, D. La Torre, E. R. Vrscay, and Z. Wang. A natural normalized metric for images and a class of associated contractive fractal transforms. In *International Conference on Applied Mathematics, Modeling and Computational Science (AMMCS-2011)*, Waterloo, ON, Canada, July 2011. 164

[10] D. Brunet, J. Vass, E. R. Vrscay, and Z. Wang. Geodesics of the structural similarity index. *Applied Mathematics Letters*, 2012. In press. Available online on 16 March 2012. 5 pages. doi: 10.1016/j.aml.2012.03.001. 75, 80, 165

[11] D. Brunet, E. R. Vrscay, and Z. Wang. The use of residuals in image denoising. In M. Kamel and A. Campilho, editors, *Proc. Int. Conf. on Image Analysis and Recognition*, volume 5627 of *LNCS*, pages 1–12. Springer, Heidelberg, 2009. 144, 160, 168

[12] D. Brunet, E. R. Vrscay, and Z. Wang. Structural similarity-based approximation of signals and images using orthogonal bases. In M. Kamel and A. Campilho, editors, *Proc. Int. Conf. on Image Analysis and Recognition*, volume 6111 of *LNCS*, pages 11–22. Springer, Heidelberg, 2010. 116, 136, 167

[13] D. Brunet, E. R. Vrscay, and Z. Wang. A class of image metrics based on the structural similarity quality index. In M. Kamel and A. Campilho, editors, *Proc. Int. Conf. on Image Analysis and Recognition*, volume 6754 of *LNCS*, pages 264–275. Springer, Heidelberg, 2011. 163

[14] D. Brunet, E. R. Vrscay, and Z. Wang. Structural similarity-based affine approximation and self-similarity of images revisited. In M. Kamel and A. Campilho, editors, *Proc. Int. Conf. on Image Analysis and Recognition*, volume 6753 of *LNCS*, pages 100–110. Springer, Heidelberg, 2011. 131, 167

[15] D. Brunet, E. R. Vrscay, and Z. Wang. On the mathematical properties of the structural similarity index. *IEEE Trans. Image Processing*, 21(4):1488–1499, 2012. 30, 43, 163, 164

[16] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005. 9, 23, 128, 142, 147, 161, 167

[17] A. Buades, B. Coll, and J.M. Morel. Image enhancement by non-local reverse heat equation. *Informatica*, 0(x), 2006. 9

[18] E. Candès, L. Demanet, and L. Ying. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Comm. Pure Appl. Math.*, 57:219–266, 2004. 147

[19] E.J. Candès, J.K. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. 113

[20] S. S. Channappayya, A. C. Bovik, C. Caramanis, and R. W. Heath. Design of linear equalizers optimized for the structural similarity index. *IEEE Trans. Image Processing*, 17(6):1624–1639, 2008. 20

[21] S. S. Channappayya, A. C. Bovik, C. Caramanis, and R. W. Heath. SSIM-optimal linear image restoration. In *Acoustics, Speech and Signal Processing, IEEE International Conference on (ICASSP 2008)*, pages 765–768, 2008. 20

[22] S. S. Channappayya, A. C. Bovik, and R. W. Heath. A linear estimator optimized for the structural similarity index and its application to image denoising. In *Image Processing, IEEE International Conference on (ICIP 2006)*, pages 2637–2640, 2006. 20

[23] S. S. Channappayya, A. C. Bovik, and R. W. Heath. Rate bounds on SSIM index of quantized images. *IEEE Trans. Image Processing*, 17(6):857–872, September 2008. 47, 124

[24] C. Charrier, K. Knoblauch, L.T. Maloney, and A.C. Bovik. Calibrating MS-SSIM for compression distortions using MLDS. In *Proc. IEEE Int. Conf. Image Proc.*, Brussels, Belgium, 2011. 14

[25] P. Chatterjee and P. Milanfar. Patch-based near-optimal image denoising. *IEEE Trans. Image Processing*, 21(4):1635–1649, 2012. 168

[26] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33:273–321, 1999. 170

[27] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43:129–159, 2001. 113

[28] K.-S. Chuang and H. K. Huang. Assessment of noise in a digital image using the join-count statistic and the moran test. *Phys. Med. Bid.*, 37(2):357–369, 1992. 142

[29] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45(5):485–560, 1992. 109

[30] R. R. Coifman and D. L. Donoho. *Wavelets and Statistics*, chapter Translation-Invariant Denoising. Springer-Verlag, New York, NY, 1995. 147

[31] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans. Image Processing*, 16(8):2080–2095, 2007. 23, 128, 147, 174

[32] S. Daly. The visible differences predictor: An algorithm for the assessment of image fidelity. In A. B. Watson, editor, *Digital Images and Human Vision*, pages 179–206. MIT Press, 1993. 11

[33] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure and Appl. Math.*, 41(7):909–996, 1988. 109

[34] G. Davis, S. Mallat, and M. Avellaneda. Greedy adaptive approximation. *Journal of Constructive Approximation*, 13:57–98, 1997. 113

[35] D. L. Donoho and I. M. Johnstone. Ideal spatial adaption via wavelet shrinkage. *Biometrika*, 81:425455, 1994. 9, 125, 147

[36] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 113

[37] R. Dosselmann and X. Yang. A comprehensive assessment of the structural similarity index. *Signal, Image and Video Processing*, 5:81–91, 2011. 16

[38] M. Ebrahimi and E. R. Vrscay. Solving the inverse problem of image zooming using self-examples. In M. Kamel and A. Campilho, editors, *Proc. Int. Conf. on Image Analysis and Recognition*, volume 4633 of *LNCS*, pages 117–130. Springer, Heidelberg, 2007. 128

[39] M. Ebrahimi and E.R. Vrscay. Regularization schemes involving self-similarity in imaging inverse problems. *Journal of Physics: Conference Series*, 124(1):012021, 2008. 9

[40] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing*, 15(12):3736–3745, 2006. 9, 125, 128, 174

[41] M. Elad and D. Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. *The Computer Journal*, 50:1–16, 2007. 128

[42] C. Etemoglu and V. Cuperman. Structured vector quantization using linear transforms. *IEEE Trans. Signal Processing*, 51:1625–1631, 2003. 128, 129

[43] R. Fletcher. Unconstrained optimization. In *Practical Methods of Optimization*, volume 1. John Wiley and Sons, 1980. 102

[44] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22:56–65, 2002. 128

[45] C.F. Gauss. *Méthode des moindres carrés: Mémoires sur la combinaison des observations*. Mallet-Bachelier, Paris, 1855. Translated by Joseph Bertrand. 16

[46] I.M. Gelfand and S.V. Fomin. *Calculus of Variations*. Prentice-Hall, 1963. 68

[47] M. Ghazel, G. Freeman, and E. R. Vrscay. Fractal image denoising. *IEEE Trans. Image Processing*, 12:1560–1578, 2003. 128

[48] B. Girod. What's wrong with mean squared error? In A. B. Watson, editor, *Digital Images and Human Vision*, pages 207–220. MIT Press, Cambridge, MA, 1993. 16

[49] D. Glew. Self-similarity of images and non-local image processing. Master's thesis, University of Waterloo, Canada, 2011. 65

[50] P.A. Hästö. A new weighted metric: the relative metric I. *Journal of Mathematical Analysis and Applications*, 274(1):38–58, 2002. 26, 27, 28, 30, 168

[51] P.A. Hästö. A new weighted metric: the relative metric II. *Journal of Mathematical Analysis and Applications*, 301(2):336 – 353, 2005. 76

[52] Y. Horita, K. Shibata, and Y. Kawayoke. MICT image quality evaluation database, 2010. http://mict.eng.u-toyama.ac.jp/mictdb.html. 181

[53] L.-G. Huang and X. Zhang. Cone metric spaces and fixed points theorems of contractive mappings. *J. Math. Anal. Appl.*, 332:1468–1476, 2007. 36

[54] A.K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989. 2

[55] C.M. Jarque and A.K. Bera. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3):255–259, 1980. 154

[56] F.-C. Jeng and J. W. Woods. Inhomogeneous gaussian image models for estimation and restoration. *IEEE Trans. on Acoust., Speech & Signal Proc.*, 36(8):1305, 1988. 142

[57] R.A. Johnson. *Modern Geometry.* Houghton Mifflin Harcourt, 1929. 25

[58] M.S. Klamkin and A. Meir. Ptolemy's inequality, chordal metric, multiplicative metric. *Pacific J. Math.*, 101(2):389–392, 1982. 27

[59] A. Kolaman and O. Yadid-Pecht. Quaternion structural similarity: a new quality index for color images. *IEEE Trans. Image Processing*, 21(4):1526–1536, 2012. 15

[60] D. La Torre, E. R. Vrscay, M. Ebrahimi, and M. F. Barnsley. Measure-valued images, associated fractal transforms and the self-similarity of images. *SIAM J. Imaging Sci.*, 2:470–507, 2009. 128

[61] E. C. Larson and D. M. Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006:1–21, Jan.-Mar. 2010. 18, 19, 181

[62] P. Le Callet and F. Autrusseau. Subjective quality assessment IRCCyN/IVC database, 2005. http://www.irccyn.ec-nantes.fr/ivcdb/. 181

[63] J.-S. Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:165–168, 1980. 147, 161

[64] X. Li. Collective sensing: a fixed-point approach in the metric space. In P. Frossard et al., editor, *Proc. SPIE Visual Comm. and Image Processing*, volume 7744, 2010. 23, 62, 174

[65] N. Lu. *Fractal Imaging.* Academic Press, 1997. 128

[66] J. Lubin. A visual discrimination model for imaging system design and evaluation. In E. Peli, editor, *Vision Models for Target Detection and Recognition*, pages 245–283. World Scientific Publishing, 1995. 11

[67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967. 126

[68] F. Malgouyres. A noise selection approach of image restoration. In *SPIE, International Conference on Wavelets IX*, volume 4478, pages 34–41, 2001. 142

[69] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993. 114

[70] G.J. Martin and B.G. Osgood. The quasihyperbolic metric and associated estimates on the hyperbolic metric. *Journal d'Analyse Mathématique*, 47(1):37–53, 1986. 72

[71] M.F. Michael F. Barnsley and L.P. Hurd. *Fractal Image Compression*. A.K. Peters, Ltd., 1993. 128

[72] M.I. Miller and L. Younès. Group actions, homeomorphisms, and matching: A general framework. *International Journal of Computer Vision*, 41:61–84, 2001. 174

[73] F. Murtagh, J.-L. Starck, and M. Louys. Very-high-quality image compression based on noise modeling. *Int. J. Imag. Syst. Tech.*, 9(11):38–45, 1998. 142

[74] K. Okarma. Colour image quality assessment using structural similarity index and singular value decomposition. In M. Kamel and A. Campilho, editors, *Proc. Int. Conf. on Image Analysis and Recognition*, volume 5337 of *LNCS*, pages 55–65. Springer, Heidelberg, 2009. 15

[75] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *SIAM Multiscale Model. and Simu.*, 4:460–489, 2005. 142

[76] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, pages 40–44, 1993. 115

[77] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990. 147

[78] R. A. Peters II. A new algorithm for image noise reduction using mathematical morphology. *IEEE Transactions on Image Processing*, 4(3):554–568, 1995. 142

[79] N. Ponomarenko, F. Battisti, K. Egiazarian, J. Astola, and V. Lukin. Metrics performance comparison for color image database. In *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, AZ, USA, 2009. 44, 181

[80] N. Ponomarenko and K. Egiazarian. Tampere Image Database 2008 TID2008. [Online]. Available: http://www.ponomarenko.info/tid2008.htm, 2008. 18, 44, 181

[81] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. on Image Proc.*, 12(11):1338–1351, 2003. 9, 147

[82] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. 155, 157

[83] Y.-I. Pyo, R.-H. Park, and S. Chang. Noise reduction in high-iso images using 3-d collaborative filtering and structure extraction from residual blocks. *Consumer Electronics, IEEE Transactions on*, 57(2):687 –695, 2011. 168

[84] A. Rehman, M. Rostami, Z. Wang, D. Brunet, and E. R. Vrscay. SSIM-inspired image restoration using sparse representation. *EURASIP Journal on Advances in Signal Processing, Special Issue on Image and Video Quality Improvement Techniques for Emerging Applications*, 16, 2012. Available online on 20 January 2012. 12 pages. doi:10.1186/1687-6180-2012-16. 116, 125, 126, 127, 167

[85] A. Rehman and Z. Wang. SSIM-based non-local means image denoising. In *Proc. IEEE Int. Conf. Image Proc.*, pages 217–220, Brussels, Belgium, 2011. 20, 175

[86] A. Rehman, Z. Wang, D. Brunet, and E. R. Vrscay. SSIM-inspired image denoising using sparse representations. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 1121–1124, Prague, Czech Republic, 2011. 116, 125, 167

[87] F.P. Ribeiro, D.A.F. Florêncio, and V.H. Nascimento. Crowdsourcing subjective image quality evaluation. In B. Macq and P. Schelkens, editors, *Proc. IEEE Int. Conf. Image Proc.*, pages 3097–3100, 2011. 181

[88] T. Richter. SSIM as global quality metric: A differential geometry view. In *Third International Workshop on Quality of Multimedia Experience (QoMEX 2011)*, pages 189–194, Mechelen, Belgium, September 7-9, 2011. IEEE. 66, 88, 165

[89] T. Richter and K. J. Kim. A MS-SSIM optimal JPEG 2000 encoder. In *Data Compression Conference*, pages 401–410, Snowbird, UT, USA, 2009. 124

[90] D.M. Rouse and S.S. Hemami. Understanding and simplifying the structural similarity metric. In *Proc. IEEE Int. Conf. Image Proc.*, pages 1188–1191, 2008. 44

[91] G.G. Roussas. Nonparametric regression estimation under mixing conditions. *Stochastic Processes and their Applications*, 36(1):107–116, 1990. 176

[92] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. Technical report, Department of Computer Science, Technion, Israel Institute of Technology, Haifa, Israel, 2008. 115, 116

[93] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth movers distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:2000, 2000. 172

[94] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992. 8, 147, 161

[95] M.P. Sampat, Z. Wang, S. Gupta, A.C. Bovik, and M.K. Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE Trans. Image Processing*, 18(11):2385–2401, 2009. 15, 169

[96] D. Saupe. Accelerating fractal image compression by multi-dimensional nearest neighbor search. In *Data Compression Conference*, pages 222–231. Society Press, 1995. 176

[97] K. Sayood. *Introduction to data compression (third ed.)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. 123

[98] D.J. Schattschneider. A multiplicative metric. *Mathematics Magazine*, 49(4):203–205, 1976. 27

[99] I. J. Schoenberg. A remark on M.M. Day's characterization of inner product spaces and a conjecture of L.M. Blumenthal. *Proc. Amer. Soc.*, 3:961–964, 1952. 24, 25

[100] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (SIGGRAPH)*, 27(3):1–10, 2008. 142

[101] H. R. Sheikh, M.F. Sabir, and A. C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms, 2006. 18, 19

[102] H. R. Sheikh, K. Seshadrinathan, A. K. Moorthy, Z. Wang, A. C. Bovik, and L. K. Cormack. Image and video quality assessment research at LIVE. [Online]. Available: http://live.ece.utexas.edu/research/quality/, 2010. 44, 181

[103] E.P. Simoncelli and E.H. Adelson. Noise removal via Bayesian wavelet coring. In *Proc. IEEE Int. Conf. Image Proc.*, volume I, pages 379–382, Lausanne, 1996. IEEE Sig Proc Society. 9, 125

[104] Y. Vardi and C.-H. Zhang. The multivariate l1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4):1423–1426, 2000. 93

[105] VQEG. Final report from the video quality experts group on the validation of objective models of video quality assessment. Technical report, http://www.vqeg.org/, 2000. 179

[106] VQEG. Final report from the video quality experts group on the validation of objective models of video quality assessment, Phase II. Technical report, http://www.vqeg.org/, 2003. 16, 178, 179

[107] M. Vuorinen. *Conformal Geometry and Quasiregular Mappings*, volume 1319 of *Lecture Notes in Mathematics*. Springer-Verlag, 1988. 75

[108] G. Wahba. *Spline Models for Observational Data*. Society of Industrial and Applied Mathematics, 1990. 8

[109] M. J. Wainwright, O. Schwartz, and E. P. Simoncelli. Natural image statistics and divisive normalization: Modeling nonlinearity and adaptation in cortical neurons. In R. Rao and et al., editors, *Probabilistic Models of the Brain: Perception and Neural Function*, pages 203–222. MIT Press, Cambridge, 2002. 13

[110] B.A. Wandell. *Foundations of Vision*. Sinauer Associates Inc., 1995. 12, 23

[111] J. Wang, Y. Liu, P. Wei, Z. Tian, Y. Li, and N. Zheng. Fractal image coding using SSIM. *Proc. IEEE Int. Conf. Image Proc.*, pages 245–248, 2011. 20, 136

[112] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao. SSIM-motivated rate-distortion optimization for video coding. *IEEE Trans. Circuits Syst. Video Techn.*, 22(4):516–529, 2012. 124

[113] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009. 12, 16, 19, 46

[114] Z. Wang, A. C. Bovik, H. R. Skeikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, 2004. 2, 11, 12, 13, 14, 16, 36, 44, 133

[115] Z. Wang and Q. Li. Information content weighting for perceptual image quality assessment. *IEEE Trans. Image Processing*, 20(5):1185–1198, 2011. 13, 169

[116] Z. Wang, Q. Li, and X. Shang. Perceptual image coding based on a maximum of minimal structural similarity criterion. In *Proc. IEEE Int. Conf. Image Proc.*, San Antonio, TX, September 2007. 124

[117] Z. Wang, L. Lu, and A.C. Bovik. Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication*, 19(2):121–132, 2004. 15

[118] Z. Wang and E. P. Simoncelli. Maximum differentiation (MAD) competition: A methodology for comparing computational models of perceptual quantities. *Journal of Vision*, 8(12):1–13, September 2008. 16

[119] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multi-scale structural similarity for image quality assessment. In *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, pages 1398–1402, Pacific Grove, CA, 2003. 14, 169

[120] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 172

[121] B. Xiong and Z. Yin. Structural similar patches for nonlocal-means with modified robust M-estimator and residual images. In *Mechatronics and Automation (ICMA), 2011 International Conference on*, pages 709 –714, 2011. 168

[122] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Processing*, 19(11):28612873, 2010. 126, 127

[123] A. Yarbus. *Eye movements and vision.* Plenum Press, New York, 1967. 13

[124] P.N. Yianilos. Normalized forms for two common metrics. In *NEC Research Institute, Report 91-082-9027-1, 1991, Revision 7/7/2002. http://www.pnylab.com/pny.* Cambridge University Press, 1991. 27

[125] D. Zhang and Z. Wang. Image information restoration based on long-range correlation. *IEEE Trans. Circuits and Systems for Video Tech.*, 12:331–341, 2002. 128