Image Compression by Microtexture Synthesis

Von der Fakultät für Elektrotechnik und Informationstechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von
Diplom-Ingenieur
Johannes Ballé
aus Dortmund

Berichter:
Univ.-Prof. Dr.-Ing. Jens-Rainer Ohm
Univ.-Prof. Dr.-Ing. Peter Vary

Tag der mündlichen Prüfung: 30.8.2012

Diese Dissertation ist auf den Internetseiten
der Hochschulbibliothek online verfügbar.

Aachen Series on Multimedia and Communications Engineering

Volume 11

Johannes Ballé

Image Compression by Microtexture Synthesis

Shaker Verlag
Aachen 2012

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: D 82 (Diss. RWTH Aachen University, 2012)

Printed in Germany.

ISBN 978-3-8440-1449-5
ISSN 1614-7782

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

I don't want to live on in my work,
I want to live on in my apartment.

Woody Allen

Vorwort

Das vorliegende Manuskript ist das Ergebnis meiner fünf Jahre dauernden Bemühungen als Doktorand am Institut für Nachrichtentechnik in Aachen. Es war das umfangreichste Projekt, dem ich mich bisher verschrieben habe; es war auch das erste Mal, dass mein Ziel nicht nur die Entwicklung eines funktionalen Prototyps war. Was mich besonders motivierte, war der Wunsch, ein Verständnis der fundamentalen Zusammenhänge zwischen menschlicher Wahrnehmung einerseits und der Statistik und mathematischen Modellierung von Bildern andererseits zu erlangen.

Die Wege und Irrwege dieses Unterfangens verschlangen einen großen Teil meiner Zeit, zum Leidwesen meines Lebensgefährten Thomas, der sich zudem immer in der Rolle des Trostspenders wiederfand, wenn ich von Selbstzweifeln geplagt wurde. Ihm ist dieses Werk gewidmet – wiewohl dies nur eine symbolische Geste ist, die ihn nicht im Geringsten entschädigen kann.

So sehr mich die wissenschaftliche Arbeit fesselte, so sehr sind mir die Mitarbeiter, Studenten und Freunde des Instituts als Wegbegleiter ans Herz gewachsen. Nicht nur der offenherzige und uneitle Umgang miteinander, die vielen inoffiziellen Aktivitäten, sondern insbesondere auch der legendäre IENT-Humor, den man Außenstehenden allenfalls als entartet beschreiben kann, werden mir immer in Erinnerung bleiben. Viele der Studenten, die ich bei ihren Arbeiten betreuen durfte, sind später zu Kollegen geworden. Ihnen und auch den anderen bin ich zu Dank verpflichtet, da ohne sie diese Arbeit nicht möglich gewesen wäre. Ich wünsche allen zukünftigen Doktoren am Institut gutes Gelingen und den übrigen Mitarbeitern (weiterhin) eine gute Zeit.

Mein besonderer Dank gilt Jens-Rainer Ohm dafür, dass er mir immer mit einem Ratschlag oder berechtigter Kritik zur Seite stand, und dass er mich mit Wohlwollen gewähren ließ – insbesondere in der Zeit, in der ich selbst noch nicht so genau wusste, wohin die Reise gehen würde. Diese Art der Betreuung habe ich mir gewünscht, obgleich sie mich vor so manche Probe gestellt hat. Ich danke auch Mathias Wien, der meines Erachtens der Ausgangspunkt für das hervorragende Arbeitsklima am Institut ist, und der mir seit meiner Diplomarbeit ein Mentor war. Die beispielhafte Integrität der Institutsleitung ist mir auf Dauer zu einer scheinbaren Selbstverständlichkeit geworden.

Köln, im Oktober 2012

Contents

1 Introduction **1**

2 Preliminaries **5**
2.1 Fields . . . 5
2.1.1 Random fields . . . 5
2.1.2 Convolution . . . 7
2.1.3 Bochner's theorem . . . 8
2.2 Higher order statistics . . . 9
2.2.1 Moment and cumulant functions . . . 9
2.2.2 Moment and cumulant spectra . . . 12
2.2.3 Moment functions and spectra of deterministic fields . . . 12
2.2.4 Generalized Wiener–Lee Theorem . . . 13
2.2.5 IID and white noise fields . . . 14
2.2.6 Coherency functions . . . 15
2.3 Gauss–Markov Random Fields . . . 16
2.3.1 Finite Gaussian fields . . . 16
2.3.2 Stationary GMRFs . . . 18
2.4 Estimation theory . . . 18

3 Applied Gauss–Markov Random Fields **21**
3.1 Inverse filtering . . . 21
3.2 Filter inversion and stability . . . 22
3.3 Estimation . . . 26
3.3.1 Estimator for σ . . . 28
3.3.2 Estimator for $\boldsymbol{b}$ . . . 28
3.3.3 Other estimators . . . 29
3.4 Conditional sampling . . . 30
3.5 Similarity metrics . . . 32
3.5.1 2D Itakura Distance . . . 33
3.5.2 Log-spectral distance . . . 34
3.5.3 STSIM . . . 34
3.5.4 SSTSIM . . . 35
3.5.5 Magnitude and power root mean square error . . . 36
3.6 Subjective similarity of Gauss–Markov Texture . . . 36
3.6.1 Experimental setup and analysis . . . 37
3.6.2 Prior distribution of texture model parameters . . . 37
3.6.3 Presentation of stimuli . . . 38
3.6.4 Analysis of subjective scores . . . 38
3.6.5 Results . . . 40

4 Image analysis **47**
4.1 Biological vision 47
4.2 Quadrature feature detection 48
4.3 Statistical interpretation 51
4.4 Sparsity, kurtosis, and Gaussian texture 56

5 Compression of Gaussian texture in natural images **59**
5.1 Structure–texture classification and decomposition 59
5.1.1 Filterbank 60
5.1.2 Space–frequency partitioning 63
5.1.3 Parameter selection 67
5.2 Reconstruction 75
5.3 Texture parameterization 80
5.4 Coding 83
5.4.1 Partitioning information 83
5.4.2 Spectral magnitude information 84
5.5 Experimental results 86

6 Summary and conclusion **97**

A Further results **99**

Bibliography **129**

Notation

$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$	set of natural, integral, real, and complex numbers, respectively
$(\cdot)^{\mathrm{T}}$	matrix/vector transpose
$(\cdot)^*$	complex conjugate
$\boldsymbol{a} = (a_0, \ldots, a_{n-1})^{\mathrm{T}}$	n-vector
$\boldsymbol{A} = \left[A_{mn}\right]$	$m \times n$ matrix
$\boldsymbol{x} = (x_0, x_1)^{\mathrm{T}} \in \mathbb{Z}^2$	pixel position 2-vector
$\boldsymbol{f} = (f_0, f_1)^{\mathrm{T}} \in \mathbb{R}^2$	Fourier domain 2-vector
$\boldsymbol{z} = (z_0, z_1)^{\mathrm{T}} \in \mathbb{C}^2$	z-domain 2-vector
$\jmath = \sqrt{-1}$	complex unit
$e^{(\cdot)}$, $\exp(\cdot)$	element-wise natural exponentiation, s.t. $e^{\boldsymbol{x}} = (e^{x_0}, e^{x_1})^{\mathrm{T}}$
$\ln(\cdot)$	element-wise natural logarithm
$\arg(\cdot)$	argument (phase) of a complex number or 2-vector
$s : \boldsymbol{x} \mapsto s(\boldsymbol{x})$	2D field $\mathbb{Z}^2 \to \mathbb{R}$
$s^- : \boldsymbol{x} \mapsto s(-\boldsymbol{x})$	2D field conjugate to s
$s * h$	2D convolution
$\delta(\boldsymbol{x}) = \delta(x_0)\delta(x_1)$	Kronecker delta
$s(\boldsymbol{x}) \overset{\mathrm{Z}}{\circ\!\!-\!\!\bullet} S(\boldsymbol{z})$	2D z-transform of s
$s(\boldsymbol{x}) \overset{\mathrm{F}}{\circ\!\!-\!\!\bullet} S(e^{\jmath 2\pi \boldsymbol{f}})$	2D Fourier transform of s
$\mathrm{P}(\cdot)$	probability
$\mathrm{p}(\cdot)$	probability density
$\mathrm{p}(\cdot \mid \cdot)$	conditional probability density
$\mathrm{E}\{\cdot\}$	expectation
$\mathrm{Re}\{\cdot\}$	real part
$\mathrm{Im}\{\cdot\}$	imaginary part
$\lvert\mathbf{a}\rvert = \sqrt{\mathbf{a}^{\mathrm{T}}\mathbf{a}^*}$	real/complex/vector magnitude
$\lvert M \rvert$	cardinality of the set M
$\frac{\partial \cdot}{\partial t}$	partial derivative with respect to t
$\int \cdot \, \mathrm{d}\boldsymbol{f}$	double integral with respect to elements of $\boldsymbol{f}$
$\square = (-1/2, 1/2]^2$	unit square

1 Introduction

The reader may be familiar with the established methods of image and video compression, such as JPEG, JPEG 2000 [J2K] for still images and the line of MPEG and ITU standards, such as H.264/AVC [AVC], for video.

What all these standards have in common is the treatment of the images as *deterministic* entities. That is, a (moving) image is comprehended as an array of pixels (picture elements). Of all possible reconstructions of an image, the one which resembles the original image most closely in a least-square sense is deemed the one with the best quality; and the methods strive to achieve this goal while reducing the amount of data that is required to store or convey the image.

However, it has long been known that this is not strictly necessary. The human visual system (HVS) extracts only a fraction of the information that is present in a given image. The HVS is governed by higher-level processes that involve attention [Ser+05]. Obviously, attention may be directed to various parts of the image; what part of an image will attract our attention depends on its content, and on our state of mind. While it is clear that any viewer may determine whether a given reconstruction is *identical* to the original, provided the two images are rendered to him for a sufficient time span, experience tells us that this task is not generally possible if the time span is limited, and that the usual observer will not notice many of the differences that may exist between two images, as his observations will be led by his inclination to extract useful (or "interesting") information from the image.

Therefore, at least when we assume that an image is observed only a single time by a single viewer, it is obvious that not all of the information in the image is relevant; and it is plausible that a part of the information could be dropped without sacrificing subjective quality of the reconstructed image. This suggests that, for a given image, there can be more than one possible reconstruction that not only leads to the same subjective quality, but that also carries the *same information* relevant to the observer.

What is the connection of this to determinism vs. statistics? The answer is easy: With the tools of statistics, we can model the "degrees of freedom" that are inherent in an image. However, none of the established methods follow this line of thought, as it is not trivial to determine what exactly comprises the irrelevant information in the image. While there is no doubt that the design of current methods is strongly influenced by the statistics of natural images, the image itself is treated as a deterministic entity, pretending that each pixel is equally relevant.

This is in complete contrast with the established methodology in speech and audio compression. Here, processes that determine relevance and irrelevance have long been identified; for example, the masking effect [PS00a]. In speech coding, the source–filter model [GG92; AKZ11] leads to a significant reduction of "degrees of freedom," i.e., we may comprehend the space of all plausible speech signals as a subspace of the space of all possible audio signals; in statistical terms, the source–filter model predicts a probability distribution of the signal that makes the vast majority of possible audio signals extremely unlikely to appear as a speech sig-

nal. Speech compression methods are designed to exploit these statistics, and they generally allow the reconstructed signal to vary, as long as certain statistics are guaranteed.

Recently, the success of texture synthesis methods has spawned a new line of image and video compression methods [NHW07; Byr+08; NBW09; BZD11; ZB11] that are designed to exploit the irrelevance that is inherent in texture. These methods are assembled of a number of common "building blocks":

- *Segmentation*. The underlying idea of segmentation algorithms is that an image is composed of a number of segments that are bounded by edges. The design of most algorithms is based on the assumption that each segment contains homogeneous texture. The first step in an image or video coding system of this type is to perform such a segmentation. [BZD11] lists quite a number of different methods that have been considered:

  > Texture segmentation is often a two step process in which features are first obtained, followed by a segmentation step which is a "grouping" operation of the homogeneous regions based on the feature properties and a grouping metric [16–18]. Feature extraction is used to measure local texture properties in an image. Typically, four approaches have been used to extract texture features: statistical-based methods, model-based methods, transform or spatial-frequency methods and structural methods [16]. In statistical-based methods, characteristics of homogeneous regions are chosen as the texture features such as the co-occurrence matrix or geometrical features such as edges [17]. Model-based methods assume that the texture is described by a stochastic model and uses model parameters to segment the texture regions. Examples of such methods are found in [18] where a multiresolution Gaussian autoregressive model is described and in [9] where an image model is formulated using a seasonal autoregressive time series. Subband decomposition, especially the use of wavelets, is often seen in spatial-frequency methods [19]. Structural methods are based on the notion that textures are composed of primitives that are well-defined and spatially repetitive [16, 20]. Boundary detection or segmentation is followed to group the features into regions with similar texture properties.

  The selection of method is usually justified by empirical arguments, or not at all.

- *Classification*. The segmentation step yields a number of sets of connected pixel locations that, together, comprise the image domain. The segments are further classified into homogeneous texture or non-texture (structure). Problems such as over- or under-segmentation arise which have to be dealt with [OB05].

- *Conventional Compression*. The methods are embedded into a so-called *host codec* which takes over the regions classified as non-texture.

- *Synthesis*. Reconstruction of the textured regions is performed by employing non-parametric texture synthesis such as Efros and Leung [EL99] and related/derived methods. These methods are quite generic in the sense that all that is assumed about the texture is that it is stationary and Markov (c.f. Section 2.1.1).

- *Quality Metric*. All recent image and video compression methods require a quality metric to perform rate–distortion optimization. Conventional codecs use the Peak Signal-to-Noise Ratio (PSNR), or, more recently, the Structural Similarity Index (SSIM) [Wan+04]. However, these cannot be used for codecs employing texture synthesis [BSO11; ZB11]. Therefore, Zhang and Bull [ZB11] use a self-designed quality metric that is claimed to capture the specific artifacts caused by their compression method.

While the methods serve well to provide an estimate of the compression efficiency that is possible, which is quite remarkable, there is a major conceptual weakness: The building blocks do not follow a common underlying physical model – like the source–filter model for speech signals. The complexity of natural image and video signals has continued to defeat image models that have been proposed. While many of the models work most of the time, the challenge to provide a model that provides a satisfactory explanation of all possible images has been continually tough. This is essentially the root of the common *computer vision* approach: heuristics are admissible, because algorithms are evaluated empirically, and only with respect to their application. However, the above methods all target general-purpose image and video compression, such that no specific assumptions can be made about the images. Thus, the class of images that should be used for evaluation is undefined, and we can never be certain that such a method indeed works for all possible images.

The benefit of a statistical model is that it allows us to address the following questions in a mathematically concise manner:

1. What is the class of texture (precisely) that a given classification algorithm is able to detect in natural images, and is it a subset of the class of texture that is handled by the analysis and synthesis algorithms?

2. Is this class of texture safe to subject to analysis–synthesis without possibly compromising semantic information, i.e., information that is potentially relevant to an observer?

3. What is the optimal representation of this class of texture for compression purposes?

4. Is the quality metric capable of detecting *all* artifacts that the compression method may cause?

5. How can a rate–distortion tradeoff be achieved, and does an equivalent parameter to the quantization step size in transform coders exist?

This thesis deals with a single class of texture – Gauss–Markov Random Fields (GMRFs). While all of the cited publications take the empirical *computer vision* approach, striving to use the "building blocks" that appear to achieve the best performance in their own right, we will deal with algorithms that are specialized to the texture model, i.e., take an approach that is rather influenced by signal theory than by computer vision.

The primary contribution of this thesis is to suggest a new approach to image compression using texture synthesis; this approach is based on

- a solid signal-theoretic foundation (c.f. Chapter 2) and

- current models of the feature detection mechanisms of the human visual system (c.f. Chapter 4).

The goal is to avoid heuristics – or models that are not backed by physical evidence – wherever possible. Where we cannot completely dispense with heuristics, we can provide reasonable estimates of error.

In Chapter 3, we establish the analogy of the Gauss–Markov random field model to the source–filter model of speech coding and examine generalizations of theoretic results from one-dimensional random signals to two-dimensional random fields. Furthermore, we establish methods for analysis and synthesis of GMRFs and present an empirical evaluation of texture quality metrics specifically targeted at that model.

In Chapter 4, we provide a statistical interpretation of feature detection in the primary visual cortex of the human brain and show that Gaussian random fields take a special role – not only with respect to information entropy, but also in the context of feature detection in biological vision. Clearly, coding systems that specialize on this class of texture, therefore, deserve a dedicated investigation.

In the last chapter, we describe and evaluate the characteristics of such a coding system. The system is designed according to the theoretic and empirical results of all the previous chapters. The properties of Gaussian random fields allow us to replace the segmentation–classification approach of the above methods with a conceptually simple and elegant statistical testing framework. This gives rise to a unique structure–texture decomposition which does not need to rely on an – often artificial – image segmentation, avoiding problems of over- or under-segmentation. The representation of texture parameters is selected according to the results of Chapter 3.

We provide a complete description of the coding and decoding system, including entropy coding, and evaluate the results on a number of established test images using objective metrics. Visual results are provided, as well.

2 Preliminaries

In the present chapter, we lay the mathematical groundwork for the rest of the thesis. As it is based on a congregation of theories from different disciplines – particularly the theory of Higher Order Statistics, originating in physics and engineering, and the theory of Gauss–Markov Random Fields, originating in mathematical statistics and the geo-sciences, it is of extraordinary importance to establish a common notation for some of the important results in these fields. Otherwise, the mathematical arguments following this chapter would be tedious; moreover, much of the insight that connects those theories, and which is necessary to appreciate the later chapters, would remain hidden to the reader.

Nevertheless, we will not go as far as providing comprehensive proofs of the presented results, unless the author is unaware of an appropriate reference. In all other cases, the reader is referred to a corresponding publication.

2.1 Fields

In order to deal with "texture" in a technical and precise way, we need to establish what we mean by it. In the image processing literature, there is no commonly agreed technical definition of this term. However, a mathematical concept that helps us go a long way into this direction is the concept of a *field*. The concept is identical in spirit to the notion of an *electromagnetic field*, in that we define a field to be a function

$$a : \mathbb{Z}^2 \to \mathbb{R}, \tag{2.1}$$

which maps a discrete-valued 2D vector $\boldsymbol{x} = (x_0, x_1)^{\mathrm{T}} \in \mathbb{Z}^2$ to a continuous scalar value $a(\boldsymbol{x})$. A digital grayscale image is an example of such a field, where the domain of a is limited to the image extents, indicating the pixel positions. However, for the purpose of dealing with texture, it is often useful to think of the domain of a field as infinite (or cyclic).

2.1.1 Random fields

While digital images are examples of deterministic fields, a *random field* is merely a statistical concept. A random field can only be characterized by its statistical properties – as opposed to an *instance* of a random field, the outcome of sampling from a random field, which is deterministic.

The statistics of a random field can be specified in different ways. A very generic way to do that is stating its joint probability distribution. We denote the joint probability density function of the entire random field a as

$$\mathrm{p}(a) = \mathrm{p}(a(\boldsymbol{x}); \boldsymbol{x} \in D), \tag{2.2}$$

where D is the domain of a. Note that it is a function $\mathbb{R}^n \to \mathbb{R}$, where n is the number of elements in the domain of a ($n = |D|$). The notation on the left-hand side does not make this

fact obvious, but it serves as a convenient shorthand. Another way of specifying the statistical properties of a random field is to give the ("full") conditional probability density function

$$\mathrm{p}(a(\boldsymbol{x}) \mid a(\boldsymbol{y}); \boldsymbol{y} \in D - \{\boldsymbol{x}\}). \tag{2.3}$$

This is, likewise, a function $\mathbb{R}^n \to \mathbb{R}$, returning the probability density of the random field element at $\boldsymbol{x}$ given all other elements of a. The joint and the conditional PDF of a random field are, of course, related; however, it can be a highly non-trivial task to obtain one of them given the other one.

Stationarity (strict-sense)

A random field whose statistical properties are invariant with respect to a shift of positions is called a *stationary* field, i.e.

$$\mathrm{p}(\boldsymbol{z}^{\boldsymbol{y}} a) = \mathrm{p}(a) \tag{2.4}$$

for any $\boldsymbol{y}$. Here, $\boldsymbol{z}$ denotes the two-dimensional shift operator, such that $(\boldsymbol{z}^{\boldsymbol{y}} a)(\boldsymbol{x}) = a(\boldsymbol{x} + \boldsymbol{y})$ for all $\boldsymbol{x}$ and $\boldsymbol{y}$. Stationarity can be defined on infinite grids (i.e., $\mathbb{Z}^2$) and also on tori; in other words, on any structure where a shift does not change the domain of the field. On a finite field, such a concept can therefore not exist.

Markov Property

A problem of the above probability density functions can be complexity. Knowledge about the entire random field is necessary to evaluate its statistics, whether its domain is finite or infinite. Luckily, this complexity is not essential for practical purposes. It has often been observed that field elements far away from any position $\boldsymbol{x}$ have negligible influence on the conditional PDF at $\boldsymbol{x}$:

$$\mathrm{p}(a(\boldsymbol{x}) \mid a(\boldsymbol{y}); \boldsymbol{y} \in D - \{\boldsymbol{x}\}) \approx \mathrm{p}(a(\boldsymbol{x}) \mid a(\boldsymbol{y}); \boldsymbol{y} \in N_{\boldsymbol{x}}), \tag{2.5}$$

where $N_{\boldsymbol{x}}$ denotes a *neighborhood* of field positions around $\boldsymbol{x}$ (excluding $\boldsymbol{x}$ itself). The idealization of (2.5) is the *Markov Property*. Commonly, it is expressed using the concept of *conditional independence*,

$$\begin{aligned} \forall \boldsymbol{x}: \quad & \mathrm{p}(a(\boldsymbol{x}), a(\boldsymbol{y}); \boldsymbol{y} \in R_{\boldsymbol{x}} \mid a(\boldsymbol{y}); \boldsymbol{y} \in N_{\boldsymbol{x}}) = \\ & \mathrm{p}(a(\boldsymbol{x}) \mid a(\boldsymbol{y}); \boldsymbol{y} \in N_{\boldsymbol{x}}) \cdot \mathrm{p}(a(\boldsymbol{y}); \boldsymbol{y} \in R_{\boldsymbol{x}} \mid a(\boldsymbol{y}); \boldsymbol{y} \in N_{\boldsymbol{x}}), \end{aligned} \tag{2.6}$$

where $R_{\boldsymbol{x}} = D - N_{\boldsymbol{x}} - \{\boldsymbol{x}\}$. Dividing both sides of the equation by $\mathrm{p}(a(\boldsymbol{y}); \boldsymbol{y} \in R_{\boldsymbol{x}} \mid a(\boldsymbol{y}); \boldsymbol{y} \in N_{\boldsymbol{x}})$ shows that this definition is equivalent to the exact version of Equation 2.5 (provided the divisor is non-zero).

Texture as stationary Markov Random Fields

The above two properties, stationarity and the Markov Property, can be combined. The resulting conditional PDF is a function $\mathbb{R}^{|N_{\boldsymbol{x}}|} \to \mathbb{R}$ which is invariant with respect to $\boldsymbol{x}$.[1]

[1] Naturally, this requires that the Markov Neighborhood $N_{\boldsymbol{x}}$ is also invariant with respect to a shift of $\boldsymbol{x}$.

It turns out that the properties are highly relevant for modeling visual texture. Probably the best example for this is a line of algorithms for non-parametric texture synthesis emerging from [EL99]. This algorithm, with its descendants [WL00; Her+01; Ash01; Kwa+03; Kwa+05], is very successful in reproducing texture from a given sample, because it assumes next to nothing about the given texture except Markovianity and stationarity. The algorithms essentially work on multi-dimensional relative frequencies as empirical approximations to the conditional PDF.

2.1.2 Convolution

We denote convolution by the symbol $*$. For two fields a and b, it is defined in the following way:

$$(a * b)(\boldsymbol{x}) = \sum_{\boldsymbol{y}} a(\boldsymbol{x} - \boldsymbol{y})\, b(\boldsymbol{y}). \tag{2.7}$$

Here and in the following sections, we omit the limits of sums for simplicity. It is understood that they run across the domain(s) of the involved fields, where not noted otherwise. More generally, if we have two functions $a(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ and $b(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ for $n \in \mathbb{N}$:

$$(a * b)(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{\boldsymbol{y}_1} \cdots \sum_{\boldsymbol{y}_n} a(\boldsymbol{x}_1 - \boldsymbol{y}_1, \ldots, \boldsymbol{x}_n - \boldsymbol{y}_n)\, b(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n). \tag{2.8}$$

Convolution can also be denoted using matrix multiplication. For this to work on fields, we need a bijective (i.e. one-to-one) mapping between grid positions and matrix row/column positions. The mapping can be in any order. We denote the mapping from matrix rows to grid positions $\omega(i)$ and its inverse $\omega^{-1}(\boldsymbol{x})$. Note that for infinitely extended fields, this implies that the domain and range of the mapping will also be infinite and, as a consequence, the order of the corresponding matrix or vector quantities will be, too. We retain this possibility for modeling considerations. Naturally, algorithms can only be implemented using finite quantities.

We define the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ as follows:

$$\boldsymbol{A} = \operatorname*{mat}_{\omega} a = \Big[\, a\big(\omega(k) - \omega(l)\big) \,\Big]_{k,l}, \tag{2.9}$$

$$\boldsymbol{b} = \operatorname*{vec}_{\omega} b = \Big[\, b\big(\omega(k)\big) \,\Big]_{k}. \tag{2.10}$$

This implies

$$\begin{aligned} \boldsymbol{A}\boldsymbol{b} &= \left[\sum_l A_{kl} b_l \right]_k = \left[\sum_l a\big(\omega(k) - \omega(l)\big)\, b\big(\omega(l)\big) \right]_k \\ &= \Big[(a * b)\big(\omega(k)\big) \Big]_k = \operatorname*{vec}_{\omega} (a * b). \end{aligned} \tag{2.11}$$

Therefore, if we arrange the elements of a and b in the matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$, the product $\boldsymbol{A}\boldsymbol{b}$ is equivalent to the convolution $a * b$ (although its explicit computation may be more complex). An additional notation that can be useful at times is a sliding vectorization of a field:

$$\boldsymbol{A}(\boldsymbol{x}) = \operatorname*{mat}_{\omega,\boldsymbol{x}} a = \Big[\, a\big(\boldsymbol{x} + \omega(k) - \omega(l)\big) \,\Big]_{k,l}, \tag{2.12}$$

$$\boldsymbol{b}(\boldsymbol{x}) = \operatorname*{vec}_{\omega,\boldsymbol{x}} b = \Big[\, b\big(\boldsymbol{x} + \omega(k)\big) \,\Big]_{k}. \tag{2.13}$$

2.1.3 Bochner's theorem

Consider a real, non-negative spectral density function $\Phi_{t,2}(e^{\jmath 2\pi f})$ which is symmetric, i.e., satisfies $\Phi_{t,2}(e^{-\jmath 2\pi f}) = \Phi_{t,2}(e^{\jmath 2\pi f})$. Its inverse Fourier transform is given by

$$\varphi_{t,2}(\boldsymbol{x}) = \int_{\square} \Phi_{t,2}(e^{\jmath 2\pi f})\, e^{\jmath 2\pi f \boldsymbol{x}}\, \mathrm{d}\boldsymbol{f}. \tag{2.14}$$

$\varphi_{t,2}(\boldsymbol{x})$ is also real and symmetric. Now consider the sum

$$\sum_{\boldsymbol{x}_1 \in D} \sum_{\boldsymbol{x}_2 \in D} \varphi_{t,2}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\, g(\boldsymbol{x}_1)\, g(\boldsymbol{x}_2), \tag{2.15}$$

where $g(\boldsymbol{x})$ is an arbitrary field and D is any finite set of field positions,

$$\begin{aligned} &= \int_{\square} \Phi_{t,2}(e^{\jmath 2\pi f}) \sum_{\boldsymbol{x}_1 \in D} \sum_{\boldsymbol{x}_2 \in D} e^{\jmath 2\pi f \boldsymbol{x}_1}\, e^{-\jmath 2\pi f \boldsymbol{x}_2}\, g(\boldsymbol{x}_1)\, g(\boldsymbol{x}_2)\, \mathrm{d}\boldsymbol{f} \\ &= \int_{\square} \Phi_{t,2}(e^{\jmath 2\pi f}) \underbrace{\left| \sum_{\boldsymbol{x} \in D} e^{-\jmath 2\pi f \boldsymbol{x}}\, g(\boldsymbol{x}) \right|^2}_{\left|G'(e^{\jmath 2\pi f})\right|^2} \mathrm{d}\boldsymbol{f} \geq 0, \end{aligned} \tag{2.16}$$

as is easily seen because both parts of the integrand are non-negative. $G'(e^{\jmath 2\pi f})$ is the Fourier transform of $g'(\boldsymbol{x})$, where $g'(\boldsymbol{x}) = g(\boldsymbol{x})$ for $\boldsymbol{x} \in D$ and $g'(\boldsymbol{x}) = 0$ otherwise. (2.15) can be written in matrix form,

$$\boldsymbol{g}^{\mathrm{T}} \boldsymbol{\phi} \boldsymbol{g} = (\underset{\omega}{\mathrm{vec}}\, g)^{\mathrm{T}} (\underset{\omega}{\mathrm{mat}}\, \varphi_{t,2}) (\underset{\omega}{\mathrm{vec}}\, g) \geq 0, \tag{2.17}$$

where ω is defined exactly on D. Now, since this inequality holds for *any* $g(\boldsymbol{x})$ and D, it also holds for any vector $\boldsymbol{g}$. The symmetry of $\boldsymbol{\phi}$, which follows from the symmetry of $\varphi_{t,2}$ and (2.9), and (2.17) together yield the definition of positive semi-definiteness. Therefore, any matrix $\boldsymbol{\phi}$ constructed from a non-negative spectral density function as given above is positive semi-definite.

Furthermore, if we require $\Phi_{t,2}(e^{\jmath 2\pi f})$ to be (strictly) positive, and $g'(\boldsymbol{x}) \neq 0$ for some $\boldsymbol{x}$, then it follows that $|G'(e^{\jmath 2\pi f})|^2$ must be positive for some $\boldsymbol{f}$, and further that the inequality in (2.16) becomes strict. Requiring $g'(\boldsymbol{x}) \neq 0$ for some $\boldsymbol{x}$ is equivalent to requiring $\boldsymbol{g} \neq \boldsymbol{0}$; thus, we have the definition of positive definiteness. Hence, for positive spectral density functions, we have $\boldsymbol{\phi}$ positive definite.

We refer to this result as *Bochner's theorem*. $\varphi_{t,2}(\boldsymbol{x})$ can be called a *positive definite* function. Curiously, Bochner's definition of a positive definite function is given for the positive *semi*-definite case, which is somewhat inconsistent. Bochner's proof [Boc33, page 406], a fundamental result of harmonic analysis, actually works in the opposite direction for continuous functions in the context of multivariate characteristic functions (the Fourier transform of a – non-negative – probability density). However, for simplicity, we choose to call the above result by the same name.

2.2 Higher order statistics

Instead of always specifying the full PDF of a random field t, we may employ a number of summary statistics. Among the possible statistics that are well known, we have the central and non-central moments and cumulants. From the signal processing literature, we are also familiar with the autocovariance and autocorrelation functions as examples of second-order statistics of stationary signals. In the following section, we summarize a useful framework [NP93] of higher-order (i.e. going beyond second-order) statistics. However, as [NP93] only deals with one-dimensional signals and systems, we use the opportunity to extend the definitions and theorems to two dimensions. Additionally, we introduce a notation that is more useful with regard to the later chapters.

2.2.1 Moment and cumulant functions

The kth order moment function of a stationary field t is, likewise, a higher-order generalization of the autocorrelation function and a higher-dimensional generalization of the kth raw moment:

$$\varphi_{t,k}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{k-1}) = \mathrm{E}\left\{\prod_{i=0}^{k-1} t(\boldsymbol{x}+\boldsymbol{x}_i)\right\}, \tag{2.18}$$

where $\boldsymbol{x}_0$ is defined to be $\mathbf{0}$. For the first few k, this gives:

$$\begin{aligned}
\varphi_{t,1} &= \mathrm{E}\{t(\boldsymbol{x})\},\\
\varphi_{t,2}(\boldsymbol{x}_1) &= \mathrm{E}\{t(\boldsymbol{x})\,t(\boldsymbol{x}+\boldsymbol{x}_1)\},\\
\varphi_{t,3}(\boldsymbol{x}_1,\boldsymbol{x}_2) &= \mathrm{E}\{t(\boldsymbol{x})\,t(\boldsymbol{x}+\boldsymbol{x}_1)\,t(\boldsymbol{x}+\boldsymbol{x}_2)\},\\
&\ldots
\end{aligned}$$

We can see above that $\varphi_{t,1}$ is the mean value of t and $\varphi_{t,2}$ corresponds to its two-dimensional autocorrelation function. The moment functions, as given here, are not functions of $\boldsymbol{x}$, which requires that the right-hand sides of the equations are invariant with respect to a variation of $\boldsymbol{x}$. This condition is called *weak* or *wide-sense* stationarity if it holds for $k \in \{1,2\}$, and *stationarity up to order n* if it holds for all $k \leq n$. The definitions of kth order moment and cumulant functions and spectra in this section are valid if the random fields are stationary up to order k. If the random field is strict-sense stationary as per (2.4), it is stationary up to all orders.

Note that the moment functions are invariant (symmetric) with respect to permutation of their arguments.[2] Thus,

$$\varphi_{t,k}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{k-1}) = \varphi_{t,k}(\boldsymbol{x}_{\nu_1},\ldots,\boldsymbol{x}_{\nu_{k-1}}) \tag{2.19}$$

for all $\boldsymbol{x}_i$ and any k, where $\boldsymbol{\nu}$ is any permutation of the integers $\{1,\ldots,k-1\}$. Furthermore, due to the stationarity of t, a property of shift symmetry arises. If we allow $\boldsymbol{x}_0 \neq \mathbf{0}$,

$$\mathrm{E}\left\{\prod_{i=0}^{k-1} t(\boldsymbol{x}+\boldsymbol{x}_i)\right\} = \mathrm{E}\left\{\prod_{i=0}^{k-1} t(\boldsymbol{x}+\boldsymbol{x}_i-\boldsymbol{x}_0)\right\} = \varphi_{t,k}(\boldsymbol{x}_1-\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{k-1}-\boldsymbol{x}_0),$$

[2] This only holds for real-valued fields; we assume all fields to be real-valued unless noted otherwise.

but, equivalently,

$$= \varphi_{t,k}(\boldsymbol{x}_0 - \boldsymbol{x}_j, \ldots, \boldsymbol{x}_{j-1} - \boldsymbol{x}_j, \boldsymbol{x}_{j+1} - \boldsymbol{x}_j, \ldots, \boldsymbol{x}_{k-1} - \boldsymbol{x}_j) \tag{2.20}$$

for all $\boldsymbol{x}_i$ and any j or k; i.e., if each argument of the function is shifted by $\boldsymbol{x}_0$, the role of the shift and any one of its arguments can be exchanged.

The kth order cumulant function is a generalization of the kth cumulant. Generally, to compute the cumulant function of some order, knowledge about lower order statistics of the field is needed. In contrast, this is not the case for moment functions. The kth order cumulant function is defined by:

$$\psi_{t,k}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}) = \sum_{M \in P(k)} (-1)^{|M|-1} (|M|-1)! \prod_{I \in M} \mathrm{E}\left\{ \prod_{i \in I} t(\boldsymbol{x} + \boldsymbol{x}_i) \right\}, \tag{2.21}$$

where $P(k)$ denotes the set of all possible partitionings of the set $\{0, 1, \ldots, k-1\}$ (i.e., a set of sets of sets of integers) [NP93, pages 9, 15]. For example,

$$\begin{aligned} P(4) = \Big\{ & \{\{0,1,2,3\}\}, \{\{0\},\{1\},\{2\},\{3\}\}, \\ & \{\{0,1\},\{2,3\}\}, \{\{0,2\},\{1,3\}\}, \{\{0,3\},\{1,2\}\}, \\ & \{\{0,1,2\},\{3\}\}, \{\{0,1,3\},\{2\}\}, \{\{0,2,3\},\{1\}\}, \{\{1,2,3\},\{0\}\}, \\ & \{\{0,1\},\{2\},\{3\}\}, \{\{0,2\},\{1\},\{3\}\}, \{\{0,3\},\{1\},\{2\}\}, \\ & \quad \{\{1,2\},\{0\},\{3\}\}, \{\{1,3\},\{0\},\{2\}\}, \{\{2,3\},\{0\},\{1\}\} \Big\}. \end{aligned}$$

Due to the expected value in (2.21) corresponding to (shifted) versions of (2.18), cumulant functions of order k can be expressed in terms of moment functions of orders 1 to k:

$$\psi_{t,k}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}) = \sum_{M \in P(k)} (-1)^{|M|-1} (|M|-1)! \prod_{I \in M} \varphi_{t,n}(\boldsymbol{x}_{i_1} - \boldsymbol{x}_{i_0}, \ldots, \boldsymbol{x}_{i_{n-1}} - \boldsymbol{x}_{i_0}), \tag{2.22}$$

where $n = |I|$ and $\boldsymbol{i}$ is any n-vector containing each element of I exactly once (exactly which ordering is used is not important due to the symmetry of the moment functions). For practical purposes, we only require cumulant functions of up to order 4. These are given by:

$$\psi_{t,1} = \varphi_{t,1}, \tag{2.23}$$

$$\psi_{t,2}(\boldsymbol{x}_1) = \varphi_{t,2}(\boldsymbol{x}_1) - \varphi_{t,1}^2, \tag{2.24}$$

$$\begin{aligned} \psi_{t,3}(\boldsymbol{x}_1, \boldsymbol{x}_2) = {} & \varphi_{t,3}(\boldsymbol{x}_1, \boldsymbol{x}_2) + 2\varphi_{t,1}^3 \\ & - \varphi_{t,1}\big[\varphi_{t,2}(\boldsymbol{x}_1) + \varphi_{t,2}(\boldsymbol{x}_2) + \varphi_{t,2}(\boldsymbol{x}_2 - \boldsymbol{x}_1)\big], \end{aligned} \tag{2.25}$$

$$\begin{aligned} \psi_{t,4}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = {} & \varphi_{t,4}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) - 6\varphi_{t,1}^4 \\ & - \varphi_{t,2}(\boldsymbol{x}_1)\varphi_{t,2}(\boldsymbol{x}_3 - \boldsymbol{x}_2) - \varphi_{t,2}(\boldsymbol{x}_2)\varphi_{t,2}(\boldsymbol{x}_3 - \boldsymbol{x}_1) \\ & - \varphi_{t,2}(\boldsymbol{x}_3)\varphi_{t,2}(\boldsymbol{x}_2 - \boldsymbol{x}_1) \\ & - \varphi_{t,1}\big[\varphi_{t,3}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \varphi_{t,3}(\boldsymbol{x}_1, \boldsymbol{x}_3) + \varphi_{t,3}(\boldsymbol{x}_2, \boldsymbol{x}_3) \\ & \quad + \varphi_{t,3}(\boldsymbol{x}_2 - \boldsymbol{x}_1, \boldsymbol{x}_3 - \boldsymbol{x}_1)\big] \\ & + 2\varphi_{t,1}^2\big[\varphi_{t,2}(\boldsymbol{x}_1) + \varphi_{t,2}(\boldsymbol{x}_3 - \boldsymbol{x}_2) \\ & \quad + \varphi_{t,2}(\boldsymbol{x}_2) + \varphi_{t,2}(\boldsymbol{x}_3 - \boldsymbol{x}_1) \\ & \quad + \varphi_{t,2}(\boldsymbol{x}_3) + \varphi_{t,2}(\boldsymbol{x}_2 - \boldsymbol{x}_1)\big]. \end{aligned} \tag{2.26}$$

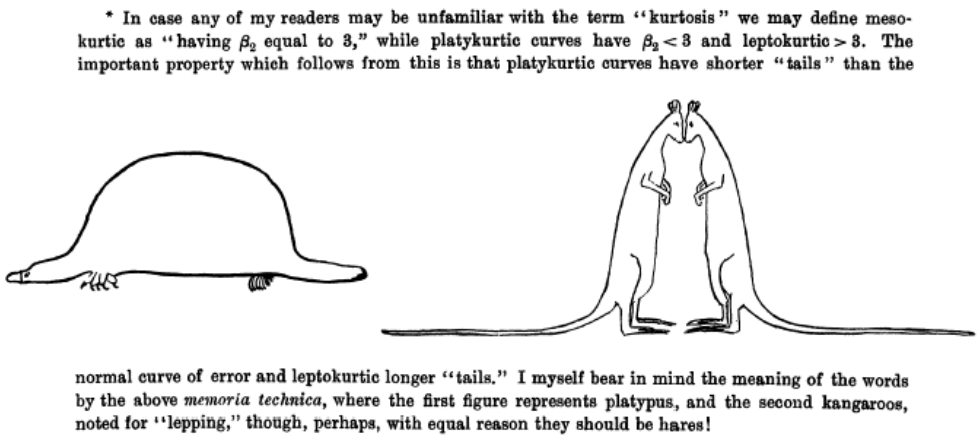


Figure 2.1 Student's mnemonic for the terms platykurtic and leptokurtic [Stu27].

Note that $\psi_{t,2}(\boldsymbol{x}_1)$ represents the well-known autocovariance function. If we assume t is zero-mean ($\psi_{t,1} = 0$), the cumulant functions simplify to:

$$\psi_{t,2}(\boldsymbol{x}_1) = \varphi_{t,2}(\boldsymbol{x}_1), \tag{2.27}$$

$$\psi_{t,3}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \varphi_{t,3}(\boldsymbol{x}_1, \boldsymbol{x}_2), \tag{2.28}$$

$$\begin{aligned}\psi_{t,4}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \varphi_{t,4}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) - \varphi_{t,2}(\boldsymbol{x}_1)\varphi_{t,2}(\boldsymbol{x}_3 - \boldsymbol{x}_2) \\ - \varphi_{t,2}(\boldsymbol{x}_2)\varphi_{t,2}(\boldsymbol{x}_3 - \boldsymbol{x}_1) - \varphi_{t,2}(\boldsymbol{x}_3)\varphi_{t,2}(\boldsymbol{x}_2 - \boldsymbol{x}_1).\end{aligned} \tag{2.29}$$

The moment functions, evaluated at the origin, result in the raw moments[3] $\mu_{t,k}$, while the cumulant functions result in the cumulants $\kappa_{t,k}$:

$$\mu_{t,k} = \varphi_{t,k}(\mathbf{0}, \ldots, \mathbf{0}), \tag{2.30}$$

$$\kappa_{t,k} = \psi_{t,k}(\mathbf{0}, \ldots, \mathbf{0}). \tag{2.31}$$

The most well-known higher order statistics – variance σ^2, skewness γ_1, and excess kurtosis γ_2 – can be defined in terms of cumulants, and, therefore, in terms of moments. For a zero-mean field t, we have:

$$\sigma^2 = \kappa_{t,2} = \mu_{t,2}, \tag{2.32}$$

$$\gamma_1 = \frac{\kappa_{t,3}}{\kappa_{t,2}^{1.5}} = \frac{\mu_{t,3}}{\sigma^3}, \tag{2.33}$$

$$\gamma_2 = \frac{\kappa_{t,4}}{\kappa_{t,2}^2} = \frac{\mu_{t,4} - 3\mu_{t,2}^2}{\mu_{t,2}^2} = \frac{\mu_{t,4}}{\sigma^4} - 3. \tag{2.34}$$

Thus, the definition of excess kurtosis as a "standardized 4th order cumulant" is an alternative justification for the offset ("excess") of 3 that makes the excess kurtosis of a Gaussian random variable equal zero. The terms platykurtic, mesokurtic, and leptokurtic refer to an excess kurtosis of below, equal to, and above zero, respectively. Student[4] [Stu27] provided a useful mnemonic which is reproduced in Figure 2.1.

[3] Note that, in the literature, raw moments are usually denoted μ'_k, while μ_k is reserved for the central moments. However, in this work, we have no need for the central moments, thus we drop the prime.

[4] *Student*, also known as William S. Gossett, is the eponym of *Students' t-distribution*.

2.2.2 Moment and cumulant spectra

The kth order moment and cumulant spectra are defined as the $(2k-2)$-dimensional z-transforms of the kth order moment and cumulant functions, respectively:

$$\Phi_{t,k}(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{k-1}) = \sum_{\boldsymbol{x}_1} \cdots \sum_{\boldsymbol{x}_{k-1}} \varphi_{t,k}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1})\, \boldsymbol{z}_1^{-\boldsymbol{x}_1} \cdots \boldsymbol{z}_{k-1}^{-\boldsymbol{x}_{k-1}}, \tag{2.35}$$

$$\Psi_{t,k}(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{k-1}) = \sum_{\boldsymbol{x}_1} \cdots \sum_{\boldsymbol{x}_{k-1}} \psi_{t,k}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1})\, \boldsymbol{z}_1^{-\boldsymbol{x}_1} \cdots \boldsymbol{z}_{k-1}^{-\boldsymbol{x}_{k-1}}. \tag{2.36}$$

The corresponding Fourier transforms result by formally substituting $\boldsymbol{z}_i = e^{\jmath 2\pi f_i}$. $\Phi_{t,2}(e^{\jmath 2\pi f})$ is well-known as the power spectral density of a random field.

2.2.3 Moment functions and spectra of deterministic fields

It is convenient to define a modified concept of moment functions and spectra for deterministic fields. This is analogous to the definition of correlation functions for energy signals, as opposed to random signals [OL10]. It is simply achieved by replacing the expectation operation with a sum. Otherwise, the definitions are equivalent to their statistical analogs. We denote them using an overdot:

$$\dot{\varphi}_{a,k}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}) = \sum_{\boldsymbol{x}} \prod_{i=0}^{k-1} a(\boldsymbol{x} + \boldsymbol{x}_i), \tag{2.37}$$

where, again, $\boldsymbol{x}_0 = \boldsymbol{0}$. Note that the symmetry properties (2.19) and (2.20) hold analogously for this definition. Similarly, we define deterministic moment spectra,

$$\dot{\Phi}_{a,k}(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{k-1}) = \sum_{\boldsymbol{x}_1} \cdots \sum_{\boldsymbol{x}_{k-1}} \dot{\varphi}_{a,k}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1})\, \boldsymbol{z}_1^{-\boldsymbol{x}_1} \cdots \boldsymbol{z}_{k-1}^{-\boldsymbol{x}_{k-1}}. \tag{2.38}$$

In addition, we can obtain the moment spectra from the z-transform of a as follows:

$$\begin{aligned}
&\dot{\Phi}_{a,k}(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{k-1}) \\
&\quad = \sum_{\boldsymbol{x}_1} \cdots \sum_{\boldsymbol{x}_{k-1}} \sum_{\boldsymbol{x}} \prod_{i=0}^{k-1} a(\boldsymbol{x} + \boldsymbol{x}_i)\, \boldsymbol{z}_1^{-\boldsymbol{x}_1} \cdots \boldsymbol{z}_{k-1}^{-\boldsymbol{x}_{k-1}},
\end{aligned}$$

substituting $\boldsymbol{y}_i = \boldsymbol{x} + \boldsymbol{x}_i$,

$$\begin{aligned}
&= \sum_{\boldsymbol{y}_1} \cdots \sum_{\boldsymbol{y}_{k-1}} \sum_{\boldsymbol{x}} a(\boldsymbol{x}) a(\boldsymbol{y}_1) \cdots a(\boldsymbol{y}_{k-1})\, \boldsymbol{z}_1^{-\boldsymbol{y}_1} \cdots \boldsymbol{z}_{k-1}^{-\boldsymbol{y}_{k-1}}\, \boldsymbol{z}_1^{\boldsymbol{x}} \cdots \boldsymbol{z}_{k-1}^{\boldsymbol{x}} \\
&= \left[\sum_{\boldsymbol{y}_1} a(\boldsymbol{y}_1)\, \boldsymbol{z}_1^{-\boldsymbol{y}_1} \right] \cdots \left[\sum_{\boldsymbol{y}_{k-1}} a(\boldsymbol{y}_{k-1})\, \boldsymbol{z}_{k-1}^{-\boldsymbol{y}_{k-1}} \right] \cdot \left[\sum_{\boldsymbol{x}} a(\boldsymbol{x})\, \boldsymbol{z}_1^{\boldsymbol{x}} \cdots \boldsymbol{z}_{k-1}^{\boldsymbol{x}} \right] \\
&= A(\boldsymbol{z}_1) \cdots A(\boldsymbol{z}_{k-1}) \cdot A\big((\boldsymbol{z}_1 \cdots \boldsymbol{z}_{k-1})^{-1}\big).
\end{aligned} \tag{2.39}$$

Note that for $k = 2$, this results in the well-known energy spectrum:

$$\dot{\Phi}_{a,2}(e^{\jmath 2\pi f}) = A(e^{\jmath 2\pi f}) \cdot A(e^{-\jmath 2\pi f}) = \left| A(e^{\jmath 2\pi f}) \right|^2. \tag{2.40}$$

Why do we only define moment functions for deterministic fields, and not cumulant functions? It turns out that the moment functions are significant for the Wiener–Lee Theorem introduced in the next section, while deterministic cumulant functions are not needed.

2.2.4 Generalized Wiener–Lee Theorem

As declared in Section 2.1.2, two fields can be convolved. If at least one of the fields is random, the result is a new random field. Otherwise, the result is deterministic. The Wiener–Lee Theorem, as understood in [OL10], deals with the convolution of a random signal and a deterministic impulse response (linear filtering). This can be generalized to fields, such that a stationary random field s is convolved with a deterministic field h (also called a two-dimensional linear filter) to obtain another stationary random field t. Furthermore, we can extend the theorem to higher orders.

Essentially, the generalized Wiener–Lee Theorem states that the moment and cumulant functions of t can be computed from the moment and cumulant functions of s, respectively, and the moment functions of h:

$$\varphi_{t,k} \equiv \varphi_{s,k} * \dot{\varphi}_{h,k}, \tag{2.41}$$

$$\psi_{t,k} \equiv \psi_{s,k} * \dot{\varphi}_{h,k}. \tag{2.42}$$

Note that, here, (2.8) instead of (2.7) applies as the functions are (k-1)-dimensional. Taking z-transforms on both sides, it follows that

$$\Phi_{t,k} \equiv \Phi_{s,k} \cdot \dot{\Phi}_{h,k}, \tag{2.43}$$

$$\Psi_{t,k} \equiv \Psi_{s,k} \cdot \dot{\Phi}_{h,k}. \tag{2.44}$$

The latter (2.44) for the one-dimensional case can be found in [NP93, page 37]. To prove[5] both variants of the theorem, we need product–sum expansion:

$$\prod_{i=0}^{m}\sum_{j=0}^{n} a_{ij} = \sum_{j_0=0}^{n}\cdots\sum_{j_m=0}^{n}\prod_{i=0}^{m} a_{ij_i}. \tag{2.45}$$

To prove (2.41), we plug (2.7) into (2.18), then use (2.45) and the linearity of the expectation operator,

$$\begin{aligned}
\varphi_{t,k}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{k-1}) &= \mathrm{E}\left\{\prod_{i=0}^{k-1}\sum_{\boldsymbol{y}} s(\boldsymbol{x}+\boldsymbol{x}_i-\boldsymbol{y})\,h(\boldsymbol{y})\right\} \\
&= \mathrm{E}\left\{\sum_{\boldsymbol{y}_0}\cdots\sum_{\boldsymbol{y}_{k-1}}\prod_{i=0}^{k-1} s(\boldsymbol{x}+\boldsymbol{x}_i-\boldsymbol{y}_i)\,h(\boldsymbol{y}_i)\right\} \\
&= \sum_{\boldsymbol{y}_0}\cdots\sum_{\boldsymbol{y}_{k-1}}\left[\prod_{i=0}^{k-1} h(\boldsymbol{y}_i)\right]\underbrace{\mathrm{E}\left\{\prod_{i=0}^{k-1} s(\boldsymbol{x}+\boldsymbol{x}_i-\boldsymbol{y}_i)\right\}}_{\varphi_{s,k}(\boldsymbol{x}_1+\boldsymbol{y}_0-\boldsymbol{y}_1,\ldots,\boldsymbol{x}_{k-1}+\boldsymbol{y}_0-\boldsymbol{y}_{k-1})},
\end{aligned}$$

[5] Note that the following treatment does not represent a rigorous mathematical proof. We omit all questions of well-definedness and existence of infinite sums. However, the answers to these questions are analogous to second-order results that are ubiquitous in the engineering literature.

and, finally, substituting $\boldsymbol{y}'_i = \boldsymbol{y}_i - \boldsymbol{y}_0$,

$$= \sum_{\boldsymbol{y}'_1} \cdots \sum_{\boldsymbol{y}'_{k-1}} \underbrace{\sum_{\boldsymbol{y}_0} \left[\prod_{i=0}^{k-1} h(\boldsymbol{y}_0 + \boldsymbol{y}'_i) \right]}_{\varphi_{h,k}(\boldsymbol{y}'_1, \ldots, \boldsymbol{y}'_{k-1})} \varphi_{s,k}(\boldsymbol{x}_1 - \boldsymbol{y}'_1, \ldots, \boldsymbol{x}_{k-1} - \boldsymbol{y}'_{k-1})$$

$$= (\varphi_{s,k} * \dot{\varphi}_{h,k})(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}). \tag{2.46}$$

Similarly, we can use (2.7) in (2.21) and then apply (2.45) twice to prove (2.42):

$$\begin{aligned}
&\psi_{t,k}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}) \\
&= \sum_{M \in P(k)} (-1)^{|M|-1} (|M|-1)! \prod_{I \in M} \mathrm{E}\left\{ \prod_{i \in I} \sum_{\boldsymbol{y}} s(\boldsymbol{x} + \boldsymbol{x}_i - \boldsymbol{y})\, h(\boldsymbol{y}) \right\} \\
&= \sum_{M \in P(k)} (-1)^{|M|-1} (|M|-1)! \prod_{I \in M} \mathrm{E}\left\{ \sum_{\boldsymbol{y}_i, i \in I} \prod_{i \in I} s(\boldsymbol{x} + \boldsymbol{x}_i - \boldsymbol{y}_i)\, h(\boldsymbol{y}_i) \right\} \\
&= \sum_{M \in P(k)} (-1)^{|M|-1} (|M|-1)! \prod_{I \in M} \sum_{\boldsymbol{y}_i, i \in I} \left[\prod_{i \in I} h(\boldsymbol{y}_i) \right] \mathrm{E}\left\{ \prod_{i \in I} s(\boldsymbol{x} + \boldsymbol{x}_i - \boldsymbol{y}_i) \right\} \\
&= \sum_{M \in P(k)} (-1)^{|M|-1} (|M|-1)! \sum_{\boldsymbol{y}_0} \cdots \sum_{\boldsymbol{y}_{k-1}} \prod_{I \in M} \left[\prod_{i \in I} h(\boldsymbol{y}_i) \right] \mathrm{E}\left\{ \prod_{i \in I} s(\boldsymbol{x} + \boldsymbol{x}_i - \boldsymbol{y}_i) \right\} \\
&= \sum_{\boldsymbol{y}_0} \cdots \sum_{\boldsymbol{y}_{k-1}} \left[\prod_{i=0}^{k-1} h(\boldsymbol{y}_i) \right] \underbrace{\sum_{M \in P(k)} (-1)^{|M|-1} (|M|-1)! \prod_{I \in M} \mathrm{E}\left\{ \prod_{i \in I} s(\boldsymbol{x} + \boldsymbol{x}_i - \boldsymbol{y}_i) \right\}}_{\psi_{s,k}(\boldsymbol{x}_1 + \boldsymbol{y}_0 - \boldsymbol{y}_1, \ldots, \boldsymbol{x}_{k-1} + \boldsymbol{y}_0 - \boldsymbol{y}_{k-1})},
\end{aligned}$$

and, by the same substitution as above,

$$= \sum_{\boldsymbol{y}'_1} \cdots \sum_{\boldsymbol{y}'_{k-1}} \underbrace{\sum_{\boldsymbol{y}_0} \left[\prod_{i=0}^{k-1} h(\boldsymbol{y}_0 + \boldsymbol{y}'_i) \right]}_{\dot{\varphi}_{h,k}(\boldsymbol{y}'_1, \ldots, \boldsymbol{y}'_{k-1})} \psi_{s,k}(\boldsymbol{x}_1 - \boldsymbol{y}'_1, \ldots, \boldsymbol{x}_{k-1} - \boldsymbol{y}'_{k-1})$$

$$= (\psi_{s,k} * \dot{\varphi}_{h,k})(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}). \tag{2.47}$$

2.2.5 IID and white noise fields

An *independent and identically distributed* (IID) random field w is a stationary field such that any two field elements $w(\boldsymbol{x}_1)$ and $w(\boldsymbol{x}_2)$ are statistically independent given that $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$. This implies that we can sample from such a field very efficiently, as we can compute the value of each element independently from all the others. The field is sufficiently specified by the probability density function $\mathrm{p}(w(\boldsymbol{x}))$ of any grid element $\boldsymbol{x}$. Equivalently, w is a stationary Markov Random Field with an empty Markov Neighborhood.

The independence implies

$$\psi_{w,k}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}) = \kappa_{w,k} \cdot \delta(\boldsymbol{x}_1) \cdots \delta(\boldsymbol{x}_{k-1}), \tag{2.48}$$

and thus

$$\Psi_{w,k}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{k-1}) = \kappa_{w,k}, \tag{2.49}$$

i.e., the kth order cumulant spectrum of the field is constant. This is due to a property of cumulants, guaranteeing that the kth order cumulant of a set of k random variables which can be partitioned in two statistically independent sets is equal to zero [NP93, page 13]. As all elements of w are mutually independent, this is the case for all k and $\boldsymbol{x}_i$ except when $\boldsymbol{x}_0 = \boldsymbol{x}_1 = \cdots = \boldsymbol{x}_{k-1} = \boldsymbol{0}$. This condition is called kth order *whiteness*; w is said to be a kth order white noise field. Thus, IID fields are white with respect to all orders, but a white field is not necessarily IID.

Note that the same property does not generally hold for moment spectra. For example,

$$\varphi_{w,4}(0,\boldsymbol{y},\boldsymbol{y}) = \mathrm{E}\Big\{w^2(\boldsymbol{x})\,w^2(\boldsymbol{x}+\boldsymbol{y})\Big\} = \Big[\mathrm{E}\Big\{w^2(\boldsymbol{x})\Big\}\Big]^2 \neq 0 \tag{2.50}$$

in general for any $\boldsymbol{y}$, even if the field is assumed to be zero-mean. Therefore, moment spectra of IID fields are not generally constant.

Stationary Gaussian fields

Stationary Gaussian random fields are stationary random fields such that any set of field elements is jointly Gaussian.

If we select any set of field elements of a stationary Gaussian random field w, the joint moments of order ≤ 2 will determine their statistics, all higher orders being redundant. Furthermore, any joint cumulant of order > 2 of the set is equal to zero [NP93, page 14]. This leads to the moment functions and spectra of order $k > 2$ being redundant, as well as

$$\psi_{w,k}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{k-1}) = 0 \qquad \text{for all } k > 2. \tag{2.51}$$

The cumulant functions and spectra of order $k > 2$ can therefore be used to measure "Gaussianity" of a given field if they are appropriately normalized.

2.2.6 Coherency functions

The kth order coherency function of a field y (with $k > 2$) is defined as

$$\Upsilon_{y,k}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{k-1}) = \frac{\Psi_{y,k}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{k-1})}{\sqrt{\Psi_{y,2}(\boldsymbol{z}_1)\cdots\Psi_{y,2}(\boldsymbol{z}_{k-1})\cdot\Psi_{y,2}(\boldsymbol{z}_1\cdots\boldsymbol{z}_{k-1})}}. \tag{2.52}$$

Coherency functions take a special form when y is *linear*. If $y = w * h$, where w is IID and h is deterministic,

$$\Upsilon_{y,k}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{k-1}) = \frac{\kappa_{w,k}}{(\kappa_{w,2})^{k/2}} \cdot \frac{\dot{\Phi}_{h,k}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{k-1})}{\sqrt{\dot{\Phi}_{h,2}(\boldsymbol{z}_1)\cdots\dot{\Phi}_{h,2}(\boldsymbol{z}_{k-1})\cdot\dot{\Phi}_{h,2}(\boldsymbol{z}_1\cdots\boldsymbol{z}_{k-1})}}. \tag{2.53}$$

On the unit hypersphere ($\boldsymbol{z}_i = e^{\jmath 2\pi f_i}$), the right-hand factor has unit magnitude, and thus, the following expression, the kth order coherence index, is constant:[6]

$$\left|\Upsilon_{y,k}(e^{\jmath 2\pi f_1},\ldots,e^{\jmath 2\pi f_{k-1}})\right| = \frac{|\kappa_{w,k}|}{(\kappa_{w,2})^{k/2}}. \tag{2.54}$$

[6] In [NP93, page 64], the right-hand side of this equation appears to be incorrect – there, the magnitude of $\kappa_{w,k}$ is not taken.

Moreover, if w is IID Gaussian, (2.54) is equal to zero. The coherence index is therefore a useful statistic to determine whether a field is linear, or Gaussian. This has been exploited for time series in [Hin90].

2.3 Gauss–Markov Random Fields

In this section, we review essential definitions and theorems from [RH05] and establish the notation.

2.3.1 Finite Gaussian fields

A finite random field t is Gaussian if the joint probability density function takes the form of a multivariate Gaussian,

$$\mathrm{p}(\boldsymbol{t}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left(-\tfrac{1}{2}\boldsymbol{t}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{t}\right), \tag{2.55}$$

where $\boldsymbol{t} = \mathrm{vec}_\omega\, t$ as given in (2.10) and ω is a bijective mapping between grid positions and vector indices defined on the domain of t, i.e., $\boldsymbol{t}$ is a vector composed of all elements of t. Here, the field is assumed to be zero-mean. $\boldsymbol{\Sigma} = \mathrm{E}\{\boldsymbol{t}\boldsymbol{t}^{\mathrm{T}}\}$ is called the covariance matrix of the field and must be symmetric positive definite.

Precision matrix

An equivalent way to express the joint PDF is by using $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$, the *precision matrix* of the field [RH05, page 22]:

$$\mathrm{p}(\boldsymbol{t}) = \sqrt{\left|(2\pi)^{-1}\boldsymbol{Q}\right|} \exp\left(-\tfrac{1}{2}\boldsymbol{t}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{t}\right). \tag{2.56}$$

The precision matrix reflects whether the Gaussian field is Markov (2.6). If it is, i.e., it constitutes a *Gauss–Markov Random Field*, the precision matrix is sparse [RH05, page 24].[7] Note that the Markov Property is not reflected in the sparsity pattern of the covariance matrix: To determine whether a field is Markov from its covariance matrix, it must be inverted.

Another property that is evident from $\boldsymbol{Q}$ is *homogeneity*. We define a homogeneous Gaussian random field as a field that satisfies

$$\boldsymbol{Q} = \operatorname*{mat}_{\omega} q \tag{2.57}$$

for any deterministic field q and bijective mapping ω defined on the domain of q. This means that the value of Q_{ij} for any two field elements i and j only depends on the relative position of these two elements. For fields defined on a finite rectangular grid, and a mapping that corresponds to a raster scan of the field elements, this implies that $\boldsymbol{Q}$ must be Toeplitz-block Toeplitz. Note that the concept of homogeneity is very similar to, but not equivalent to the concept of stationarity. It would be somewhat misleading to speak of a stationary field if the field is finite, as it cannot be shifted without changing its domain. However, if we imagine extending the domain of the field – without changing the underlying structure of its precision

[7] [RH05] distinguishes between the *local*, *global*, and *pairwise* Markov Property, and goes on to show that they are all closely related to the sparsity structure of a GMRF. Our definition (2.6) corresponds to the local Markov Property.

Algorithm 2.1 Conditional sampling of Gaussian random field

Input: $\boldsymbol{Q}, \boldsymbol{t}_B$
Output: $\hat{\boldsymbol{t}}_A$
1: obtain a matrix $\boldsymbol{L} = \boldsymbol{P}\boldsymbol{T}$ such that $\boldsymbol{Q}_{AA} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}$
2: sample $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1})$
3: solve $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{r} = \boldsymbol{w}$ for $\boldsymbol{r}$
4: solve $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\,\boldsymbol{d} = -\boldsymbol{Q}_{AB}\boldsymbol{t}_B$ for $\boldsymbol{d}$
5: **return** $\boldsymbol{r} + \boldsymbol{d}$

matrix given by (2.57) –, we can recognize stationarity (2.4) as the limiting case of homogeneity, as the domain extends to $\mathbb{Z}^2$.

Conditional probability density

If we partition the vector into a known part $\boldsymbol{t}_B$ and an unknown part $\boldsymbol{t}_A$, the conditional distribution given $\boldsymbol{t}_B$ is:

$$\mathrm{p}(\boldsymbol{t}_A \mid \boldsymbol{t}_B) = \sqrt{\left|(2\pi)^{-1}\boldsymbol{Q}_{AA}\right|} \exp\left(-\tfrac{1}{2}(\boldsymbol{t}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{Q}_{AA}(\boldsymbol{t}-\boldsymbol{\mu})\right), \tag{2.58}$$

where $\boldsymbol{\mu} = -\boldsymbol{Q}_{AA}^{-1}\boldsymbol{Q}_{AB}\boldsymbol{t}_B$ [RH05, page 26]. An example for a partitioning of $\boldsymbol{t}$ and $\boldsymbol{Q}$ is given by:

$$\boldsymbol{t} = \begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix}, \qquad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_{AA} & \boldsymbol{Q}_{AB} \\ \boldsymbol{Q}_{BA} & \boldsymbol{Q}_{BB} \end{bmatrix}. \tag{2.59}$$

However, the index sets A and B need not be consecutive. In the case when A is only a single element i, and B consists of all others ($\setminus i$), we obtain the full conditional PDF:

$$\mathrm{p}(t_i \mid \boldsymbol{t}_{\setminus i}) = \frac{1}{\sqrt{2\pi Q_{i,i}^{-1}}} \exp\left(-\frac{(t_i + Q_{i,i}^{-1}\boldsymbol{Q}_{i,\setminus i}\boldsymbol{t}_{\setminus i})^2}{2Q_{i,i}^{-1}}\right). \tag{2.60}$$

Note that this corresponds to a univariate Gaussian with a mean of $-Q_{i,i}^{-1}\boldsymbol{Q}_{i,\setminus i}\boldsymbol{t}_{\setminus i}$ and a variance of $Q_{i,i}^{-1}$.

Sampling

To sample from a zero-mean Gaussian random field, a matrix $\boldsymbol{L}$ must be found such that $\boldsymbol{Q} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}$ and $\boldsymbol{L} = \boldsymbol{P}\boldsymbol{T}$, where $\boldsymbol{T}$ is a triangular matrix and $\boldsymbol{P}$ is a permutation matrix. This could be, for example, the Cholesky decomposition. Then, to get a sample $\hat{\boldsymbol{t}}$, we can solve $\boldsymbol{L}^{\mathrm{T}}\hat{\boldsymbol{t}} = \boldsymbol{w}$, where $\boldsymbol{w}$ is an IID standard Gaussian vector, by forward or backward substitution [RH05, page 35]. This is proved simply by observing that $\mathrm{E}\{\hat{\boldsymbol{t}}\hat{\boldsymbol{t}}^{\mathrm{T}}\} = \mathrm{E}\{\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{L}^{-1}\} = \boldsymbol{L}^{-\mathrm{T}}\boldsymbol{1}\boldsymbol{L}^{-1} = \boldsymbol{\Sigma}$.

If a number of field elements is already fixed and we would like to do *conditional sampling*, i.e. with respect to texture images, "inpainting," or interpolation consistent with the statistics of the field, we can use (2.58) to get the conditional PDF (i.e., A is the region which needs to be interpolated, and B is the region already known). The mean of the conditional PDF corresponds to a "deterministic" component of $\boldsymbol{t}_A$, which is due to the knowledge about $\boldsymbol{t}_B$

and cannot be avoided if the statistics of the complete random field are to be obeyed. The interpolation is achieved by Algorithm 2.1, which is a variation of the conditional sampling algorithm and its discussion found in [RH05, pages 35, 36].

2.3.2 Stationary GMRFs

In the signal processing community, stationary random processes and field models have long been known and worked with. However, the various needs of researchers with respect to domain, dimensionality, Gaussianity, and stationarity of the field, as well as the type of Markov Neighborhood involved, led to closely related, but slightly different, and sometimes to equivalent, but differently named – or even different, but identically named – definitions of autoregressive processes and second-order random fields. An illuminating classification of the most popular models can be found in [DK89].

For this work, it will be sufficient to introduce the so-called *Conditional Autoregression* (CAR) model. This type of model is briefly discussed in [RH05, pages 72–75]. The discussion establishes the existence of such a process, and goes on to relate the full conditional PDF of a field t [RH05, page 73],[8]

$$\begin{aligned} \mathrm{p}(t(\boldsymbol{x}) \mid t(\boldsymbol{y}); \boldsymbol{y} \in \mathbb{Z}^2 - \{\boldsymbol{x}\}) &= \mathcal{N}\left(-\frac{1}{q(\mathbf{0})} \sum_{\boldsymbol{y} \in \mathbb{Z}^2 - \{\mathbf{0}\}} q(\boldsymbol{y})\, t(\boldsymbol{x}-\boldsymbol{y}), \frac{1}{q(\mathbf{0})} \right) \\ &= \frac{1}{\sqrt{2\pi q^{-1}(\mathbf{0})}} \exp\left(-\frac{\Big((t * q)(\boldsymbol{x})\Big)^2}{2\,q(\mathbf{0})} \right), \end{aligned} \tag{2.61}$$

where q is a deterministic field, symmetric with $q(\boldsymbol{x}) = q(-\boldsymbol{x})$ for all $\boldsymbol{x}$, representing the parameters of the distribution, to the power spectral density of t [RH05, page 72],

$$\Phi_{t,2}(e^{\jmath 2\pi f}) = \frac{1}{\sum_{\boldsymbol{y}} q(\boldsymbol{y}) e^{-\jmath 2\pi f \boldsymbol{y}}} = \frac{1}{Q(e^{\jmath 2\pi f})}. \tag{2.62}$$

The close connection of this kind of process to linear filtering is established in Chapter 3.

2.4 Estimation theory

For a good introduction to estimation theory, consider the book by Kay [Kay93]. Here, only the most basic definitions are provided.

An estimator is a function of a number of observations, yielding an estimate of an unobserved (hidden) variable. For a toy example, we may consider a number of elements of a white noise field $w(\boldsymbol{x}), \boldsymbol{x} \in D$ of which we desire to estimate the mean $\kappa_{w,1}$. A simple estimator would then be:

$$\hat{\kappa}_{w,1} = \frac{1}{|D|} \sum_{\boldsymbol{x} \in D} w(\boldsymbol{x}). \tag{2.63}$$

Throughout this thesis, estimators will be denoted as the respective quantity decorated with a tilde.

[8] The notation was slightly adapted.

Bias

A biased estimator is an estimator whose expected value is not equal to the true value of the hidden variable. Its opposite is an unbiased estimator, for which the equality holds. For example,

$$\mathrm{E}\left\{\hat{\kappa}_{w,1}\right\} = \frac{1}{|D|}\sum_{\boldsymbol{x}\in D}\mathrm{E}\{w(\boldsymbol{x})\} = \kappa_{w,1} \tag{2.64}$$

is an unbiased estimator due to the linearity of $\mathrm{E}\{\cdot\}$.

Consistency

A consistent estimator is an estimator which, as the number of observations increase, converges to the true value of the hidden variable. For example,

$$\lim_{D\to\mathbb{Z}^2}\hat{\kappa}_{w,1} = \lim_{D\to\mathbb{Z}^2}\frac{1}{|D|}\sum_{\boldsymbol{x}\in D}w(\boldsymbol{x}) = \kappa_{w,1} \tag{2.65}$$

is a consistent estimator if and only if w is first-order ergodic.

Robustness

A robust estimator is an estimator whose value is not arbitrarily distorted by a contamination of the observations. Contamination refers to a replacement of some of the observations by values from a different PDF, which is considered to be unknown and, possibly, of a much larger dispersion than the actual data. There are differing methods of how to measure robustness; a comprehensive treatment of the subject is provided by Maronna, Martin, and Yohai [MMY06]. The most commonly encountered example of a robust estimator is the median, which provides an alternative estimator for the above example,

$$\hat{\kappa}_{w,1} = \operatorname*{med}_{\boldsymbol{x}\in D} w(\boldsymbol{x}). \tag{2.66}$$

Provided that both PDFs (of the uncontaminated data and of the contamination) are symmetric, the influence of the contamination on the estimate is extremely low (with up to 50% of the observations contaminated). On the other hand, the median is often not as efficient as the mean when the data is uncontaminated, meaning that its variance is higher. It is possible to trade-off between the robustness of the median and the efficiency of the mean using *M-estimators* [MMY06].

3 Applied Gauss–Markov Random Fields

In this chapter, we review a number of methods related to Gauss–Markov Random Field and autoregressive models, both finite and stationary, and establish links between these very similar concepts. These links can be exploited when dealing with finite segments of stationary texture. Furthermore, we introduce a two-dimensional extension of the Itakura Distance, review some other similarity metrics applicable to GMRFs and evaluate them against subjective similarity of texture images.

3.1 Inverse filtering

Consider a random field generated by inverse filtering an IID standard Gaussian random field w using a deterministic field (linear filter) a:

$$\begin{aligned} t(\boldsymbol{x}) &= \sigma w(\boldsymbol{x}) - (a * t)(\boldsymbol{x}) \\ &= \sigma w(\boldsymbol{x}) - \sum_{\boldsymbol{y}} t(\boldsymbol{x}-\boldsymbol{y})\, a(\boldsymbol{y}). \end{aligned} \tag{3.1}$$

This is called a two-dimensional *Autoregressive Process*. Given an appropriate starting configuration, the process can be realized recursively. For this, a must have a restricted support, such that in every step, only elements of t are used that belong to the starting configuration, or have been previously computed. The most common configuration is *unilateral* support, as defined in [DK89], such that the support is restricted to one of the non-symmetric half planes, excluding the origin. Clearly, t is Gaussian and zero-mean, as each element of it is a linear combination of zero-mean Gaussians (of course, assuming that the starting configuration of the process was, likewise).

We can state the "directional" conditional PDF at $\boldsymbol{x}$ simply by observing that the sum in (3.1) solely depends on previously generated elements of t,

$$\mathrm{p}(t(\boldsymbol{x}) \mid t(\boldsymbol{y}); \boldsymbol{y} < \boldsymbol{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t(\boldsymbol{x}) + (a * t)(\boldsymbol{x}))^2}{2\sigma^2} \right), \tag{3.2}$$

where $\boldsymbol{y} < \boldsymbol{x}$ indicates "$\boldsymbol{y}$ is generated before $\boldsymbol{x}$." Note that this is not equivalent to the full conditional PDF, where also all "future" elements of t, with respect to $\boldsymbol{x}$, are conditioned upon. However, it is easy to confirm that this model fulfills a conditional independence property analogous to the Markov Property (2.6), i.e.,

$$\mathrm{p}(t(\boldsymbol{x}) \mid t(\boldsymbol{y}); \boldsymbol{y} < \boldsymbol{x}) = \mathrm{p}(t(\boldsymbol{x}) \mid t(\boldsymbol{y}); \boldsymbol{y} \in N_{\boldsymbol{x}}) \tag{3.3}$$

with an appropriately defined neighborhood $N_{\boldsymbol{x}}$ (see Figure 3.1 for an example).

Now let us consider the z-transform of the random field,

$$\begin{aligned} t(\boldsymbol{x}) &= \sigma w(\boldsymbol{x}) - (a * t)(\boldsymbol{x}) \\ &\quad \updownarrow \mathrm{Z} \\ T(\boldsymbol{z}) &= \sigma W(\boldsymbol{z}) - A(\boldsymbol{z}) \cdot T(\boldsymbol{z}) \\ &= \frac{\sigma}{1 + A(\boldsymbol{z})} \cdot W(\boldsymbol{z}). \end{aligned} \tag{3.4}$$

Let us now call $B(\boldsymbol{z}) = 1 + A(\boldsymbol{z})$ and $H(\boldsymbol{z}) = \frac{1}{B(\boldsymbol{z})}$. Another representation of t is then

$$\begin{aligned} T(\boldsymbol{z}) &= H(\boldsymbol{z}) \cdot \sigma W(\boldsymbol{z}) \\ &\quad \updownarrow \mathrm{Z} \\ t(\boldsymbol{x}) &= \sigma (w * h)(\boldsymbol{x}). \end{aligned} \tag{3.5}$$

It follows immediately due to the generalized Wiener–Lee Theorem, (2.41) and (2.43), that,

$$\varphi_{t,2}(\boldsymbol{x}) = \sigma^2 (\varphi_{w,2} * \dot{\varphi}_{h,2})(\boldsymbol{x}) = \sigma^2 \sum_{y} h(\boldsymbol{y}) h(\boldsymbol{y} + \boldsymbol{x}) \tag{3.6}$$

$$\text{and } \Phi_{t,2}(\boldsymbol{z}) = \sigma^2 \Phi_{w,2}(\boldsymbol{z}) \cdot \dot{\Phi}_{h,2}(\boldsymbol{z}) = \frac{\sigma^2}{B(\boldsymbol{z}) B(\boldsymbol{z}^{-1})}. \tag{3.7}$$

As t is Gaussian and zero-mean, we know that its cumulant spectra for all $k \in \mathbb{N} - \{2\}$ vanish, and that the distribution of t is fully determined by its second-order statistics, i.e., its power spectral density. As the Fourier phase of a is lost in (3.7), it must therefore be irrelevant for the statistics of the field. Moreover, we can now relate (3.7) to (2.62) to establish

$$Q(e^{\jmath 2\pi f}) = \frac{1}{\sigma^2} \left| 1 + A(e^{\jmath 2\pi f}) \right|^2. \tag{3.8}$$

As the CAR process is also Gaussian and zero-mean, we conclude that if (3.8) holds, the CAR process from Section 2.3.2 and the AR process are statistically equivalent. In the spatial domain, this relationship reads

$$q \equiv \frac{1}{\sigma^2} (\delta + a^-) * (\delta + a), \tag{3.9}$$

where a^- denotes the conjugate field to a, such that $a^-(\boldsymbol{x}) = a(-\boldsymbol{x})$ for any $\boldsymbol{x}$. The difference between the processes is only in the practical realization: While the CAR representation does not offer an obvious way of sampling from such a field, the AR representation allows sampling by a recursive procedure.

We can now easily state the full conditional PDF of an AR process (3.1) by first computing q according to (3.9) and plugging it into (2.61).

3.2 Filter inversion and stability

Given that the Fourier phase of the filter is irrelevant, we may ask what "degrees of freedom" we have with respect to the choice of a filter polynomial $B(\boldsymbol{z})$ when we would like to

achieve any given PSD of the form (3.7), and what representation of $B(\boldsymbol{z})$ is most efficient, i.e., eliminates these degrees of freedom. The latter is Question 3 from the introduction. These questions have *fundamentally* different answers depending on whether we consider one-dimensional time series or higher-dimensional fields.

Supposing the system described by (3.1) was a one-dimensional process, we would need to impose the following restrictions on $b(x)$:

- It must be causal and finite to be implemented.
- The frequency response must satisfy $|H(z)| < \infty$ for $|z| = 1$, i.e., the filter must be stable; all poles of $H(z)$ must lie strictly within the unit circle [OS75, page 69]. This is equivalent to requiring
$$B(z) \neq 0 \qquad \text{for} \qquad |z| \geq 1. \tag{3.10}$$

These restrictions can be complied with by finding the poles of $\Phi_{t,2}(z)$. Due to the Fundamental Theorem of Algebra, its denominator can be written as a product of first-order complex polynomials. The roots of the denominator polynomial must occur in complex conjugate pairs [OS75, page 390]. Therefore, it is always possible to assign the roots to two polynomial factors $B(z)$ and $B(z^{-1})$. This is called *spectral factorization*. To satisfy (3.10), we simply assign all roots within the unit circle to $B(z)$ (and, consequently, their corresponding roots outside the unit circle to $B(z^{-1})$). Then, the filter is guaranteed to be stable, and also *minimum-phase*. Therefore, if we are willing to accept a finite complex rational approximation, any 1D spectrum can be represented by a set of inverse filters $B(\boldsymbol{z})$ all having the same Fourier magnitude. The phase is irrelevant for the statistics of the process, but if we intend to implement the filter recursively, there is only one – the minimum-phase – solution.

These theoretical results are exploited in practice for efficient representations of linear filters. For example, the line-spectral pair representation [SJ84] decomposes $B(\boldsymbol{z})$ into two polynomials whose roots, due to symmetry, are all located on the unit circle and interleaved. This representation admits a particularly efficient numerical scheme to locate the zeros.

In two dimensions, the following restrictions on $b(\boldsymbol{x})$ are analogous to the ones above:

- It must be "causal" and finite, i.e., its support must be restricted in a way that it is recursively computable. Without loss of generality, we require the support to be unilateral, i.e. defined as a subset of the non-symmetric half plane as in Figure 3.1: $b(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{Z}^2 - U$, where
$$U = \{\boldsymbol{x} : x_0 > 0 \vee (x_0 = 0 \wedge x_1 > 0)\}. \tag{3.11}$$
- The frequency response must satisfy $|H(\boldsymbol{z})| < \infty$ for $|\boldsymbol{z}| = \mathbf{1}$. For unilateral support, this is equivalent to requiring the inverse filter to be "recursively stable" [Jai89, page 213]:
$$B(\boldsymbol{z}) \neq 0 \qquad \text{for} \qquad \big(|z_0| \geq 1 \wedge z_1 = \infty\big) \vee \big(|z_0| = 1 \wedge |z_1| \geq 1\big). \tag{3.12}$$

With polynomials of higher order, the Fundamental Theorem of Algebra is no longer applicable, i.e. for a given denominator polynomial of the two variables z_0 and z_1, a factorization into polynomials of lower orders may simply not exist [EW76]. Furthermore, we have no reason to assume that the singularities of $H(\boldsymbol{z})$ are isolated points in $\mathbb{C}^2$; and apparently, it

can be shown that this is *never* the case [Kra82, page 11]. Thus, it is conceptually not correct to speak of "poles" in the higher-dimensional case, and it may even be considered misleading. A visualization of the sets of points in $\mathbb{C}^2$ where a given complex rational function of two variables attains the value of zero, or becomes singular, is generally difficult due to the high number of dimensions. Projections of these manifolds can be visualized, for example, in [DM84, pages 184–185]. Clearly, as the roots of a 2D polynomial do not in general correspond to isolated points, there is no trivial extension of the line-spectral pair representation for two-dimensional signals.

A representation that can be extended from one to two dimensions is the *partial correlation coefficient* (PARCOR) representation. It can be shown that, in 1D, the stability of the inverse filter can be guaranteed by simply requiring the magnitude of all PARCOR coefficients to be less than 1. Moreover, it is guaranteed that the mappings between the class of 1D positive-definite autocorrelation functions (3.6), the class of 1D inverse filters (a, σ), and the class of 1D partial correlation coefficient arrays are mutually bijective [Mar80]. The equivalence between the former two is sometimes referred to as the *correlation matching property*. This property does not hold for two-dimensional fields [Mar78].

Interestingly, a similar PARCOR representation can be defined [Mar78; Mar80] for two dimensions. However, this representation requires the inverse filter representation to extend infinitely in one of the dimensions. A conversion from a finite-support inverse filter array to the PARCOR coefficient array almost always yields an array that is infinitely extended; in a practical application, it is therefore necessary to avoid this conversion to maintain precision. Marzetta points out that a conversion in the other direction is guaranteed to produce a finite-support inverse filter array from a finite-support PARCOR array, implying that the domain of PARCOR arrays is useful for filter design. However, the 2D extension of the most well-known algorithm to estimate the PARCOR array from a given covariance matrix, the Levinson recursion, is only optimal, like in 1D, if the recursion is performed infinite times in one of the dimensions. For certain spectra, like spectra with discontinuities, it can be shown to perform poorly if the support is restricted to a finite number of recursions [Mar78, pages 168–180]. Therefore, the PARCOR representation lends itself only to practical use if filter design is done in a supervised fashion, or if approximative solutions are considered adequate (though this may apply with severe restrictions, as the author is unaware of a method to quantify the error of such a solution if the true parameters are unknown). It should, however, be noted that – in principle – an alternative estimation algorithm for the PARCOR representation could be conceived that does not have these limitations.

While the root-finding approach to spectral factorization, and the line-spectral pair representation which builds on it, are thus infeasible in two dimensions, the approach followed in [Mar78] is destined to be inaccurate due to the requirement of infinite recursion, which in practice needs to be broken. A similar limitation arises in another spectral factorization approach due to Wiener and Doob when it is generalized to 2D [EW76].

The total ordering of field elements of [Mar78] implies only one definition of "2D minimum-phase" (and of the PARCOR representation). However, as there are 8 different ways a raster-scan ordering can be defined on $\mathbb{Z}^2$, each definition of causality implies a different definition of "minimum-phase." [EW76] follows a more general approach and proposes a number of *canonical* factorizations with 2, 4, or 8 factors, applying the min/max-phase terminology to each dimension, resulting in combinations of minimum-, mixed-, and maximum-phase terms, such as "*min-max* phase," for example.

		to				
		$\varphi_{t,2}(\boldsymbol{x})$	$q(\boldsymbol{x})$	$h(\boldsymbol{x}),\sigma$	$b(\boldsymbol{x}),\sigma$	$\rho(\boldsymbol{x})$
from	$\varphi_{t,2}(\boldsymbol{x})$		(2.62)	[EW76]		
	$q(\boldsymbol{x})$	(2.62)			[EW76]	
	$h(\boldsymbol{x}),\sigma$	$*$ (3.6)			[EW76]	
	$b(\boldsymbol{x}),\sigma$		$*$ (3.9)	[EW76]		[Mar80]
	$\rho(\boldsymbol{x})$		$*$		$*$ [Mar80]	

Table 3.1 Mapping between different representations of the stationary Gauss–Markov Random Field parameters. Mappings marked with an asterisk are guaranteed to preserve "finiteness" of the corresponding arrays.

The basic idea can be motivated as follows. Consider a power spectral density $\Phi_{t,2}(e^{\jmath 2\pi f})$, where $0 < \Phi_{t,2}(e^{\jmath 2\pi f}) < \infty$. The frequency response magnitude of the desired filter is determined to be

$$\left|H(e^{\jmath 2\pi f})\right| = \sqrt{\Phi_{t,2}(e^{\jmath 2\pi f})/\sigma^2}. \tag{3.13}$$

The problem is now to find $\arg H(e^{\jmath 2\pi f})$ such that the resulting filter is unilateral and stable. Another way to look at this is to consider the complex logarithm of $H(\boldsymbol{z})$:

$$\begin{aligned}\check{H}(\boldsymbol{z}) = \ln H(\boldsymbol{z}) &= \ln\Big[\,|H(\boldsymbol{z})|\cdot\exp\big(\jmath\arg H(\boldsymbol{z})\big)\Big] \\ &= \ln|H(\boldsymbol{z})| + \jmath\arg H(\boldsymbol{z}) \qquad (3.14)\end{aligned}$$

$$\check{h}(\boldsymbol{x}) = \check{h}_{\mathrm{s}}(\boldsymbol{x}) + \check{h}_{\mathrm{a}}(\boldsymbol{x}). \tag{3.15}$$

Here, $\check{h}_{\mathrm{s}}$ is the inverse Fourier transform of $\ln\left|H(e^{\jmath 2\pi f})\right|$, which is real; therefore,

$$\check{h}_{\mathrm{s}}(\boldsymbol{x}) = \check{h}_{\mathrm{s}}^*(-\boldsymbol{x}) \tag{3.16}$$

is symmetric. Likewise, as $\jmath\arg H(e^{\jmath 2\pi f})$ is imaginary,

$$\check{h}_{\mathrm{a}}(\boldsymbol{x}) = -\check{h}_{\mathrm{a}}^*(-\boldsymbol{x}) \tag{3.17}$$

is antisymmetric. $\check{h}$ is sometimes called the *complex cepstrum* of h.

If we choose $\check{h}_{\mathrm{a}}(\boldsymbol{x}) = \check{h}_{\mathrm{s}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in U$, $\check{h}(\boldsymbol{x})$ is unilateral. This implies that its transform (3.14) satisfies (3.12): all singularities of $\check{H}(\boldsymbol{z})$ are confined to the specified region. What is interesting here is that the logarithm maps both singularities and zeroes of $H(\boldsymbol{z})$ to singularities of $\check{H}(\boldsymbol{z})$. Therefore, all zeros and singularities of $H(\boldsymbol{z})$ are bound to the same region. This makes $H(\boldsymbol{z})$ "2D minimum-phase" according to Marzetta's definition, and, consequently, $h(\boldsymbol{x})$ is stable and unilateral.

The above development suggests a procedure to obtain $\arg H(\boldsymbol{z})$ using the DFT [EW76]. This is also the source of the limitation of this approach. In general, $\check{h}$ has support on $\mathbb{Z}^2$, and the DFT implies periodicity – an approximate solution is inevitable, even though the error may be made arbitrarily small.

To give a simplified summary of the above: Concepts such as spectral factorization, prediction error filters, and partial correlation coefficient representations exist for two-dimensional stationary fields – however, the lack of a Fundamental Theorem of Algebra for polynomials of higher orders than one has severe implications for their practical use. Only a small subset of algorithms and representations are feasible in practical (and therefore finite) systems. In Table 3.1, we list five representations of Gauss–Markov Random Field parameters along with the methods to convert from one to the other, with h and b denoting the minimum-phase filters. Only a few of the conversions guarantee a finite-support output array when a finite-support input array is given.

We see from the above review that a theoretical answer to Question 3 is, unfortunately, difficult to give. Analysis of functions with several complex parameters is still a subject of active mathematical research. New theoretic results in this field may have an impact on applications of two-dimensional random fields. Until then, it will remain difficult to design systems that do without (albeit, arbitrarily precise) approximations.

3.3 Estimation

Suppose the following situation arises. We are given a finite segment of a stationary random field, along with the information that it is Gaussian and zero-mean. Our task is to determine the statistical properties of this field. As we are aware from Section 3.1, this means that we need to estimate the parameters of the full conditional PDF[1] – or, equivalently, the power spectral density of the field. This problem is known as the *spectral estimation* problem. It is well known in the signal processing literature and several good introductions are available [Mar87; SM97; SM05].

Without loss of generality, we may assume that an AR process was used to generate the field, as the AR process can generally represent any given power spectrum. This can be seen from (2.62): In the CAR representation, the reciprocal of the power spectrum and q are a cosine transform pair,

$$\frac{1}{\Phi_{t,2}(e^{\jmath 2\pi f})} = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) e^{-\jmath 2\pi \boldsymbol{f}\boldsymbol{x}} = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \cos(2\pi \boldsymbol{f}\boldsymbol{x}). \tag{3.18}$$

The Fourier transform collapses to a cosine transform because of the symmetry of the PSD, which is obvious when considering the inverse transform,

$$q(\boldsymbol{x}) = \int_{\square} \frac{1}{\Phi_{t,2}(e^{\jmath 2\pi f})} \cos(2\pi \boldsymbol{f}\boldsymbol{x}) \, \mathrm{d}\boldsymbol{f}. \tag{3.19}$$

q must have a real, non-negative, and symmetric Fourier transform as the PSD is real, non-negative, and symmetric (as we are concerned with real-valued fields only). If the support of q is restricted, for example to a square region centered on the origin, the representation of the power spectrum is *band-limited* (in a non-conventional sense); i.e., the reciprocal of the PSD is assumed to be a smooth function. Due to the transform relationship, we can represent any reciprocal of a PSD, and consequently, any PSD, by arbitrarily extending the support of q. With respect to the AR representation, (3.8) simply constitutes a *spectral factorization* of $Q(\boldsymbol{z})$, which always exists (but is not necessarily finite [EW76]).

[1] The "joint PDF" of the field elements cannot be found as the field is infinitely extended.

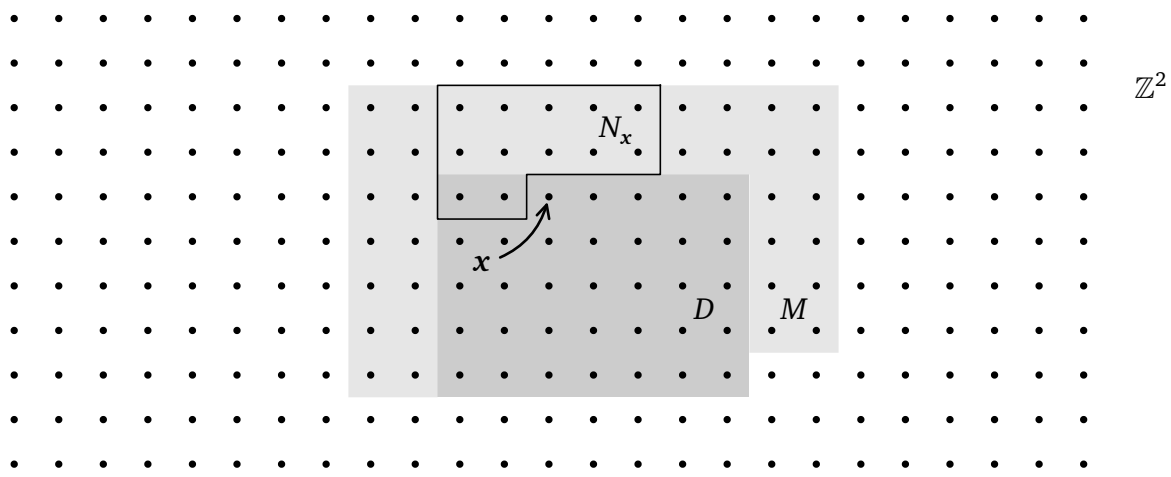


Figure 3.1 Finite segment D of a Gaussian random field on $\mathbb{Z}^2$.

The AR representation happens to encompass a (maximum likelihood, maximum entropy) spectral estimator which is not only one of the most efficient with respect to computational complexity, but has also been a traditional choice as a high-resolution spectral estimator. It is closely related to the linear prediction problem and can be derived as follows.

An illustration of the random field segment, which we will use for spectral estimation, is given in Figure 3.1. The set $D + M$ comprises all field elements that are necessary to evaluate the "directional" conditional PDF (3.2) at the elements in D, given a neighborhood $N_{\boldsymbol{x}}$. We assume that the field elements in D and M are observed. Now, by the "chain rule" of probability, and considering the conditional independence properties of the PDF,

$$\begin{aligned}\mathrm{p}(t(\boldsymbol{x}); \boldsymbol{x} \in D \mid t(\boldsymbol{y}); \boldsymbol{y} \in M) &= \prod_{\boldsymbol{x}\in D} \mathrm{p}(t(\boldsymbol{x}) \mid t(\boldsymbol{y}); \boldsymbol{y} \in N_{\boldsymbol{x}}) \\ &= (2\pi\sigma^2)^{-\frac{|D|}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{\boldsymbol{x}\in D} (t(\boldsymbol{x}) + (a * t)(\boldsymbol{x}))^2 \right). \qquad (3.20)\end{aligned}$$

Defining $b \equiv \delta + a$, $\boldsymbol{b} = \mathrm{vec}_{\omega}\ b$, and $\boldsymbol{t}(\boldsymbol{x}) = \mathrm{vec}_{\omega,\boldsymbol{x}}\ t$ (ω covering $N_{\mathbf{0}}$, and including the origin), this reduces to

$$\begin{aligned}\mathrm{p}(t(\boldsymbol{x}); \boldsymbol{x} \in D \mid t(\boldsymbol{y}); \boldsymbol{y} \in M) &= (2\pi\sigma^2)^{-\frac{|D|}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{\boldsymbol{x}\in D} (\boldsymbol{b}^{\mathrm{T}} \boldsymbol{t}(\boldsymbol{x}))^2 \right) \\ &= (2\pi\sigma^2)^{-\frac{|D|}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{\boldsymbol{x}\in D} \sum_{k=0}^{|N|} \sum_{l=0}^{|N|} b_k\, b_l\, t_k(\boldsymbol{x})\, t_l(\boldsymbol{x}) \right) \\ &= (2\pi\sigma^2)^{-\frac{|D|}{2}} \exp\left(-\frac{|D|}{2\sigma^2} \boldsymbol{b}^{\mathrm{T}} \hat{\boldsymbol{V}} \boldsymbol{b} \right), \qquad (3.21)\end{aligned}$$

where $|N|$ is the number of neighbors in $N_{\mathbf{0}}$ and $\hat{\boldsymbol{V}}$ represents a correlation matrix estimate given by

$$\hat{\boldsymbol{V}} = \frac{1}{|D|} \left[\sum_{\boldsymbol{x}\in D} t_k(\boldsymbol{x})\, t_l(\boldsymbol{x}) \right]_{k,l}, \qquad k, l \in \{0, \ldots, |N|\}. \qquad (3.22)$$

Assuming t is ergodic, this is a consistent estimate approaching

$$\boldsymbol{V} = \lim_{D\to\mathbb{Z}^2} \hat{\boldsymbol{V}} = \Big[\mathrm{E}\big\{t_k(\boldsymbol{x})\, t_l(\boldsymbol{x})\big\}\Big]_{k,l} = \operatorname*{mat}_{\omega} \varphi_{t,2}. \tag{3.23}$$

Note that the parameters of the PDF are now neatly encapsulated in σ and $\boldsymbol{b}$ (although $b_{\omega^{-1}(\mathbf{0})} = 1$ is redundant). The Maximum Likelihood estimates, $\hat{\boldsymbol{b}}$ and $\hat{\sigma}$, are derived by maximizing the log likelihood function,

$$L(\boldsymbol{b},\sigma) = \ln \mathrm{p}(t(\boldsymbol{x}); \boldsymbol{x}\in D \mid t(\boldsymbol{y}); \boldsymbol{y}\in M) = -\frac{|D|}{2}\left[\ln\big(2\pi\sigma^2\big) + \frac{\boldsymbol{b}^{\mathrm{T}}\hat{\boldsymbol{V}}\boldsymbol{b}}{\sigma^2}\right]. \tag{3.24}$$

It can be shown that $L(\boldsymbol{b},\sigma)$ is concave with respect to each of its arguments. This allows us to consider the maximization problem separately for each variable. Since there exists only one global maximum, we may also find its location simply by setting the first derivative to zero.

3.3.1 Estimator for σ

Assuming $\boldsymbol{b}$ is fixed, we derive the estimator $\hat{\sigma}$ by setting the derivative of $L(\boldsymbol{b},\sigma)$ with respect to σ to zero:

$$\begin{aligned} & \frac{\partial L(\boldsymbol{b},\hat{\sigma})}{\partial\hat{\sigma}} = -|D|\left[\frac{1}{\hat{\sigma}} - \frac{\boldsymbol{b}^{\mathrm{T}}\hat{\boldsymbol{V}}\boldsymbol{b}}{\hat{\sigma}^3}\right] \stackrel{!}{=} 0 \\ \Leftrightarrow \qquad & \hat{\sigma}^2(\boldsymbol{b},\hat{\boldsymbol{V}}) = \boldsymbol{b}^{\mathrm{T}}\hat{\boldsymbol{V}}\boldsymbol{b}. \end{aligned} \tag{3.25}$$

Here and in what follows, we emphasize the fact that the estimator is a function of $\boldsymbol{b}$ and $\hat{\boldsymbol{V}}$ using function notation. Comparing this equation to (3.21) through (3.20), we see that it can also be interpreted as a function of $\boldsymbol{b}$ and the field segment,

$$\hat{\sigma}^2(\boldsymbol{b},t,D) = \frac{1}{|D|}\sum_{x\in D}(t(\boldsymbol{x}) + (a*t)(\boldsymbol{x}))^2. \tag{3.26}$$

With (3.25) and (3.26), we thus have two alternative algorithms at hand we can use to estimate σ. While (3.25) can be used with an available estimate $\hat{\boldsymbol{V}}$, or, if it is known, the exact correlation matrix $\boldsymbol{V}$ in its place, (3.26) is an estimator in terms of the data itself. It can be interpreted in terms of linear prediction as the mean prediction error and corresponds to the empirical power of the driving noise signal $\sigma w(\boldsymbol{x})$ (this also results from rearranging (3.1)).

3.3.2 Estimator for $\boldsymbol{b}$

For all $\{n; \omega(n)\in N_{\mathbf{0}}\}$, we set

$$\begin{aligned} \frac{\partial L(\hat{\boldsymbol{b}},\sigma)}{\partial\hat{b}_n} &= -\frac{|D|}{2\sigma^2}\frac{\partial}{\partial\hat{b}_n}\sum_{k=0}^{|N|}\sum_{l=0}^{|N|}\hat{b}_k\,\hat{b}_l\,\hat{V}_{k,l} \\ &= -\frac{|D|}{2\sigma^2}\left[\frac{\partial}{\partial\hat{b}_n}\hat{b}_n^2\,\hat{V}_{n,n} + \sum_{\substack{k=0\\k\neq n}}^{|N|}\frac{\partial}{\partial\hat{b}_n}\hat{b}_k\,\hat{b}_n\,\hat{V}_{k,n}\right. \end{aligned}$$

$$+\sum_{\substack{l=0\\l\neq n}}^{|N|}\frac{\partial}{\partial\hat{b}_n}\hat{b}_n\,\hat{b}_l\,\hat{V}_{n,l}+\sum_{\substack{k=0\\k\neq n}}^{|N|}\sum_{\substack{l=0\\l\neq n}}^{|N|}\frac{\partial}{\partial\hat{b}_n}\hat{b}_k\,\hat{b}_l\,\hat{V}_{k,l}\Bigg]$$

$$=-\frac{|D|}{2\sigma^2}\left[2\,\hat{b}_n\,\hat{V}_{n,n}+\sum_{\substack{k=0\\k\neq n}}^{|N|}\hat{b}_k\,\hat{V}_{k,n}+\sum_{\substack{l=0\\l\neq n}}^{|N|}\hat{b}_l\,\hat{V}_{n,l}\right]$$

(due to the symmetry of $\hat{\boldsymbol{V}}$)

$$=-\frac{|D|}{\sigma^2}\sum_{l=0}^{|N|}\hat{b}_l\,\hat{V}_{n,l}\stackrel{!}{=}0 \tag{3.27}$$

$$\Leftrightarrow\qquad\sum_{l=0}^{|N|}\hat{b}_l\,\hat{V}_{n,l}=0. \tag{3.28}$$

The value of each $\hat{b}_n$ is determined by the solution set of these linear equations, and thus $\hat{\boldsymbol{b}}$ is a function of $\hat{\boldsymbol{V}}$. Plugging this result into (3.25) yields a simpler estimator for σ given that the estimate for $\boldsymbol{b}$ is known:

$$\begin{aligned}\hat{\sigma}^2(\hat{\boldsymbol{V}})=\hat{\sigma}^2(\hat{\boldsymbol{b}}(\hat{\boldsymbol{V}}),\hat{\boldsymbol{V}})&=\sum_{k=0}^{|N|}\sum_{l=0}^{|N|}\hat{b}_k\,\hat{b}_l\,\hat{V}_{k,l}\\&=\sum_{k=0}^{|N|}\hat{b}_k\underbrace{\sum_{l=0}^{|N|}\hat{b}_l\,\hat{V}_{k,l}}_{=0\text{ f. }\omega(k)\neq\mathbf{0}}\\&=\underbrace{\hat{b}_{\omega^{-1}(\mathbf{0})}}_{=1}\sum_{l=0}^{|N|}\hat{b}_l\,\hat{V}_{\omega^{-1}(\mathbf{0}),l}.\end{aligned} \tag{3.29}$$

Assuming $\omega(0)=\mathbf{0}$ without loss of generality, we can integrate (3.28) and (3.29) to the well-known equation system

$$\hat{\boldsymbol{V}}\hat{\boldsymbol{b}}=[\hat{\sigma}^2,0,\ldots,0]^{\mathrm{T}}. \tag{3.30}$$

This system of linear equations is known from linear prediction theory as the set of *augmented normal equations* [The92]. Note that a great variety of computational techniques to solve this set of equations has been proposed. The least involved is the Cholesky decomposition which exploits the symmetry and positive definiteness of $\hat{\boldsymbol{V}}$. However, since the matrix may in practice well become positive *semi*-definite, a modified Cholesky decomposition is useful for some applications. Such a decomposition, employing pivoting, exists and has been shown to be numerically stable [Hig90]. Only recently, efficient implementations of this algorithm have been developed for the mathematical standard library LAPACK [HHL09].

3.3.3 Other estimators

Some authors propose slightly different ways of estimating $\boldsymbol{V}$ that deviate from the unbiased, Maximum Likelihood approach. Most proposals allow a more computationally efficient matrix inversion by enforcing special structures of the correlation matrix. For 1D signals, it is

even possible to guarantee a recursively stable inverse filter by using a special, biased estimator (historically named the "autocorrelation method"), which guarantees that the matrix is Toeplitz. Although the biased estimator can be extended to fields, forcing $\hat{\boldsymbol{V}}$ to be Toeplitz-block Toeplitz and thus enabling more efficient matrix inversion algorithms, the stability result does not carry over to two dimensions [MSM84]. Thus, the only advantage remaining is computational efficiency at the cost of an inferior estimator.

Note that linear estimators do also exist for bilateral autoregressive processes such as the CAR [Jai89, page 209].[2] An estimator with comparable computational complexity to the one above could thus be used that directly estimates the elements of q. However, it is not suitable for some applications. It turns out that, as it directly matches the reciprocal of the PSD against the Fourier transform $Q(e^{\jmath 2\pi f})$, the resulting spectral estimate may be negative for some $\boldsymbol{f}$ (i.e., the function q is not restricted to be positive definite). This is an undesired property, as PSDs must be non-negative to be physically plausible, and some algorithms show undesired behavior when this prerequisite is not met. Using the AR estimator restricts the spectral estimate to be non-negative without requiring special precautions, as the spectral estimate is always of the form given in (3.7).

Rather than using the cosine transform of the reciprocal PSD (3.19), we could use the cosine transform of the PSD itself, which simply yields the coefficients of the autocorrelation function $\varphi_{t,2}$. Representing the field using a factorization of these coefficients, analogously to the AR representation, would amount to a *Moving Average* (MA) process, which is also capable of capturing all possible PSDs. However, it is well-known that such a process is much less efficient in representing spectral peaks; that is, a higher number of parameters is required for a similarly precise representation.

Obviously, still other ways of parameterizing the PSD are feasible. For example, we can model a PSD by keeping track of the power in each subband of a filterbank (ideally, the subband filters should be non-negative and orthogonal). The power in each subband is easily estimated by filtering with the subband filter and estimating variance of the output. The PSD is then modeled as a weighted sum of the subband filter frequency responses. A similar representation is used in Chapter 5.

3.4 Conditional sampling

Although the AR representation offers a straight-forward method to sample from a field by means of the recursive equation (3.1), the requirements of this method with respect to the stability of a pose a practical problem, particularly when the parameters of the field need to be estimated unsupervisedly, after the design of a system has been completed. Additionally, the recursive implementation does not allow to take border conditions on all sides of the field segment into account. For example, if we desire to interpolate the elements of a finite rectangular segment D of a field given all surrounding elements (Figure 3.2), the method is unsuitable.

We can utilize the CAR representation to solve this problem, assuming that the field is Markov (2.6), i.e. the deterministic field $q(\boldsymbol{x})$, representing the parameters of the model, has

[2] The estimator discussed above is equivalent to what the author calls a causal "Minimum Variance Representation" in this reference. Additionally, the author discusses non-causal MVRs equivalent to the CAR representation.

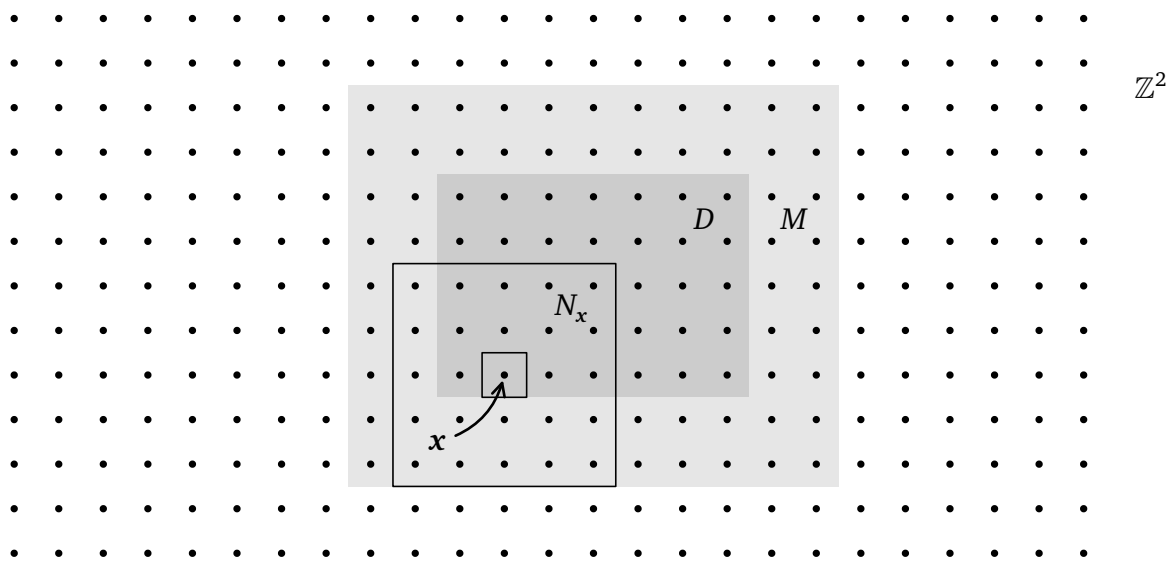


Figure 3.2 Finite segment D of a CAR on $\mathbb{Z}^2$ corresponding to a homogeneous finite GMRF. The outer segment $\mathbb{Z}^2 - D - M$ and D are conditionally independent given M.

restricted support on a finite region around the origin, corresponding to the Markov Neighborhood. If this is the case, we can designate a finite set of random field elements M around the segment D, shown in Figure 3.2, such that, given M, D and the set of all the remaining field elements are conditionally independent. Thus, we require only knowledge about the field elements in M to interpolate the elements in D.

This is done simply by matching the full conditional PDF of the CAR representation (2.61) to the full conditional PDFs of a finite GMRF on $D+M$ (2.60) such that for each field element in D, the PDF is identical. For a mapping ω defined on $D+M$, such that

$$\boldsymbol{t} = \underset{\omega}{\text{vec}}\, t = \begin{bmatrix} \boldsymbol{t}_D \\ \boldsymbol{t}_M \end{bmatrix}, \tag{3.31}$$

where $\boldsymbol{t}_D$ and $\boldsymbol{t}_M$ correspond to the elements in D and M, respectively, we require that

$$\begin{aligned} \mathrm{p}(t_i \mid \boldsymbol{t}_{\setminus i}) &= \frac{1}{\sqrt{2\pi Q_{i,i}^{-1}}} \exp\left(-\frac{([\boldsymbol{Q}\boldsymbol{t}]_i)^2}{2Q_{i,i}} \right) \\ &\overset{!}{=} \frac{1}{\sqrt{2\pi q^{-1}(\boldsymbol{0})}} \exp\left(-\frac{\Big((t * q)(\omega(i))\Big)^2}{2q(\boldsymbol{0})} \right) \end{aligned} \tag{3.32}$$

for all $i \in \{0, \ldots, |D| - 1\}$. If $\boldsymbol{Q}$ is partitioned analogously to (3.31) as

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_{DD} & \boldsymbol{Q}_{DM} \\ \cdot & \cdot \end{bmatrix}, \tag{3.33}$$

this requirement is satisfied if and only if $\boldsymbol{Q} = \operatorname{mat}_{\omega} q$ for the upper part of $\boldsymbol{Q}$ (the lower part indicated by dots is irrelevant).

This equivalence only holds if we choose M in such a way that it contains all field elements that are conditionally dependent from any of the elements in D; i.e., minimally, it is the union of all Markov Neighborhoods, but excluding D:

$$M = \sum_{\boldsymbol{x} \in D} N_{\boldsymbol{x}} - D. \tag{3.34}$$

An example of this is given in Figure 3.2. If we chose to extend ω by an additional field element not in $D + M$, the corresponding column in $[\, \mathrm{q}_{DD}\ \mathrm{q}_{DM} \,]$ would consist only of zeroes and would therefore be redundant. Now, we can apply (2.58) to obtain the joint PDF of $\boldsymbol{t}_D$ conditioned on $\boldsymbol{t}_M$,

$$\mathrm{p}(\boldsymbol{t}_D \mid \boldsymbol{t}_M) = \sqrt{\left|(2\pi)^{-1}\boldsymbol{Q}_{DD}\right|} \exp\left(-\tfrac{1}{2}(\boldsymbol{t}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{Q}_{DD}(\boldsymbol{t}-\boldsymbol{\mu})\right), \tag{3.35}$$

where $\boldsymbol{\mu} = \boldsymbol{Q}_{DD}^{-1}\boldsymbol{Q}_{DM}\boldsymbol{t}_M$. This allows us to use Algorithm 2.1 to interpolate the field elements in D.

Note that the finite GMRF on D is homogeneous (2.57) by definition, as it is a segment of a stationary random field. Therefore, the structure of Q is highly redundant (on top of being sparse, if the model order is sufficiently small compared to the size of D), which allows specialized algorithms to accelerate the computation of the Cholesky decomposition.

A universal correspondence between the parameter array of the CAR $q(\boldsymbol{x})$ and the precision matrix of a finite GMRF $\boldsymbol{Q}$ is evident from the above development. A similar relationship exists between the autocovariance function $\psi_{t,2}(\boldsymbol{x})$ and the covariance matrix $\boldsymbol{\Sigma}$. It is possible to obtain the elements of the covariance matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{t}_D$ by matching

$$\boldsymbol{\Sigma} = \Big[\mathrm{E}\{t_i t_j\} - \mathrm{E}\{t_i\}\,\mathrm{E}\{t_j\}\Big]_{ij} = \Big[\mathrm{E}\big\{t\big(\omega(i)\big)\,t\big(\omega(j)\big)\big\} - \mathrm{E}\big\{t\big(\omega(i)\big)\big\}\,\mathrm{E}\big\{t\big(\omega(j)\big)\big\}\Big]_{ij},$$

and, due to the stationarity of t,

$$= \Big[\psi_{t,2}\big(\omega(j) - \omega(i)\big)\Big]_{ij} = \operatorname*{mat}_{\omega} \psi_{t,2}. \tag{3.36}$$

However, it should be noted that the mapping from the matrix representations to the stationary fields is not necessarily unique, since elements are lost if the matrix is not sufficiently large.

3.5 Similarity metrics

Consider having estimated the parameter set of a zero-mean, stationary Gauss–Markov Random Field from a finite segment of it: We may be interested in how well we did compared to the true parameter set. A similar situation occurs when we need to store or process the true parameter set in a lossy way (for example, quantize it), and we would like to quantify the loss. In both cases, we have to deal with a perturbed, or inaccurate, set of parameters $(\tilde{\boldsymbol{b}}, \tilde{\sigma})$ and need to compare them to the true parameters $(\boldsymbol{b}, \sigma)$, or their best estimate.

We can measure the "similarity" between these two random field models using functions of the parameters that return a scalar. We refer to this type of function as a *similarity metric*, even though the function might not satisfy the formal requirements to be called a metric (e.g., symmetry or the triangle inequality). The following similarity "metric" is directly related to the AR estimator discussed in Section 3.3.

3.5.1 2D Itakura Distance

The *Itakura Distance* [Ita75] was introduced for audio signals by Fumitada Itakura in the context of speech modeling. It is distinguished by its close relationship to the linear prediction problem and the fact that its computation can be rather undemanding if an estimator such as discussed in Section 3.3 is already used. It turns out that the extension of the Itakura Distance to two-dimensions exists.

The 2D Itakura Distance is equal to the log ratio of the likelihood that the observed field segment corresponds to a given AR model $\tilde{\boldsymbol{b}}$ vs. the maximum likelihood that it corresponds to *any other* AR model, i.e. to the set of AR parameters $\boldsymbol{b}$ that best fit the data. The distance is invariant with respect to σ and thus invariant with respect to a scaling of the range of the field elements. Referring to (3.24), the distance is defined as

$$\begin{aligned} D_{\mathrm{I}}(\tilde{\boldsymbol{b}}, \hat{\boldsymbol{V}}) &= \frac{2}{|D|}\Big[L\big(\hat{\boldsymbol{b}}(\hat{\boldsymbol{V}}), \hat{\sigma}(\hat{\boldsymbol{V}})\big) - L\big(\tilde{\boldsymbol{b}}, \hat{\sigma}(\tilde{\boldsymbol{b}}, \hat{\boldsymbol{V}})\big)\Big] \\ &= -\ln\Big(2\pi\hat{\sigma}^2\big(\hat{\boldsymbol{V}}\big)\Big) - \frac{\hat{\boldsymbol{b}}^{\mathrm{T}}(\hat{\boldsymbol{V}})\,\hat{\boldsymbol{V}}\,\hat{\boldsymbol{b}}(\hat{\boldsymbol{V}})}{\hat{\sigma}^2(\hat{\boldsymbol{V}})} + \ln\big(2\pi\hat{\sigma}^2(\tilde{\boldsymbol{b}}, \hat{\boldsymbol{V}})\big) + \frac{\tilde{\boldsymbol{b}}^{\mathrm{T}}\hat{\boldsymbol{V}}\tilde{\boldsymbol{b}}}{\hat{\sigma}^2(\tilde{\boldsymbol{b}}, \hat{\boldsymbol{V}})} \\ &= -\ln\Big(2\pi\hat{\sigma}^2\big(\hat{\boldsymbol{V}}\big)\Big) - 1 + \ln\big(2\pi\hat{\sigma}^2(\tilde{\boldsymbol{b}}, \hat{\boldsymbol{V}})\big) + 1 \\ &= \ln\left(\frac{\hat{\sigma}^2(\tilde{\boldsymbol{b}}, \hat{\boldsymbol{V}})}{\hat{\sigma}^2\big(\hat{\boldsymbol{V}}\big)}\right). \end{aligned} \tag{3.37}$$

Since $\boldsymbol{b}$ is adapted to the data while $\tilde{\boldsymbol{b}}$ is predetermined, the denominator in this expression is always smaller than or equal to the numerator. Hence, the measure is always non-negative, a value of zero occurring only if $\tilde{\boldsymbol{b}} = \boldsymbol{b}$.

Note that the normalization $\frac{2}{|D|}$ ensures that the Itakura Distance of two field segments can be compared, even if they comprise a different number of field elements. The combined Itakura Distance of a union of field segments is the average of their individual Itakura Distances weighted by the number of elements they, respectively, consist of. This is a direct consequence of the properties of the logarithm.

We can find a frequency domain interpretation of the Itakura Distance by noting that due to (3.26),

$$D_{\mathrm{I}}(\tilde{\boldsymbol{b}}, \hat{\boldsymbol{V}}) = D_{\mathrm{I}}(\tilde{\boldsymbol{b}}, t, D) = \ln\left(\frac{\frac{1}{|D|}\sum_{x\in D}\Big[(\tilde{b} * t)(\boldsymbol{x})\Big]^2}{\hat{\sigma}^2\big(\hat{\boldsymbol{V}}\big)}\right),$$

and further, assuming t is ergodic and D is large enough, or, equivalently, $\boldsymbol{V}$ is known,

$$\begin{aligned} D_{\mathrm{I}}(\tilde{\boldsymbol{b}}, \boldsymbol{V}) = \lim_{D\to\mathbb{Z}^2} D_{\mathrm{I}}(\tilde{\boldsymbol{b}}, t, D) &= \ln\left(\frac{\mathrm{E}\Big\{\Big[(\tilde{b} * t)(\boldsymbol{x})\Big]^2\Big\}}{\sigma^2}\right) \\ &= \ln\left(\frac{\big(\varphi_{t,2} * \dot{\varphi}_{\tilde{b},2}\big)(\boldsymbol{0})}{\sigma^2}\right) \\ &= \ln\left(\frac{1}{\sigma^2}\int_{\square} \Phi_{t,2}(e^{\jmath 2\pi \boldsymbol{f}})\,\Big|1 + \tilde{A}(e^{\jmath 2\pi \boldsymbol{f}})\Big|^2\,\mathrm{d}\boldsymbol{f}\right) \end{aligned}$$

$$= \ln\left(\int_{\square} \frac{\left|1 + \tilde{A}(e^{\jmath 2\pi f})\right|^2}{\left|1 + A(e^{\jmath 2\pi f})\right|^2} \,\mathrm{d}\boldsymbol{f} \right) \tag{3.38}$$

$$= \ln\left(\int_{\square} \frac{\Phi_{t,2}(e^{\jmath 2\pi f})/\sigma^2}{\Phi_{\tilde{t},2}(e^{\jmath 2\pi f})/\tilde{\sigma}^2} \,\mathrm{d}\boldsymbol{f} \right). \tag{3.39}$$

Here, the quantities decorated with a tilde are defined in analogy to the development in Section 3.3; for instance, $\tilde{t}$ is the hypothetical random field resulting from the parameters $\tilde{\sigma}, \tilde{\boldsymbol{b}}$, and so forth.

From (3.38), it is obvious that D_{I} is not symmetric. Therefore, it cannot be called a metric in the strict sense.

3.5.2 Log-spectral distance

One of the oldest metrics, which is unrelated to the AR model, is the log-spectral distance (LSD):

$$D_{\mathrm{LSD}}(s,t) = 10\sqrt{\int_{\square} \left(\log_{10} \frac{\Phi_{s,2}(e^{\jmath 2\pi f})}{\Phi_{t,2}(e^{\jmath 2\pi f})} \right)^2 \mathrm{d}\boldsymbol{f}}. \tag{3.40}$$

It is in wide-spread use for many applications. However, as it has no interpretation in terms of AR modeling or linear prediction, its evaluation must take place after the spectra of the fields s and t have been estimated in a separate step. Many more metrics with different normalizations and properties have been conceived. An overview in the context of speech coding is given in [QBC88, pages 53 sqq.]. In this work, the LSD is considered the representative metric of this set.

3.5.3 STSIM

A different class of metrics is offered as the *structural similarity index* (SSIM) [Wan+04] and its derivates, such as the complex wavelet based CWSSIM [Sam+09]. The SSIM is defined as

$$D_{\mathrm{SSIM}}(s,t) = l^{\alpha}(s,t)\, c^{\beta}(s,t)\, s^{\gamma}(s,t), \tag{3.41}$$

where

$$l(s,t) = \frac{2\kappa_{s,1}\kappa_{t,1} + C_0}{\kappa_{s,1}^2 + \kappa_{t,1}^2 + C_0}, \tag{3.42}$$

$$c(s,t) = \frac{2\sqrt{\kappa_{s,2}\kappa_{t,2}} + C_1}{\kappa_{s,2} + \kappa_{t,2} + C_1}, \tag{3.43}$$

$$s(s,t) = \frac{\kappa_{st} + C_2}{\sqrt{\kappa_{s,2}\kappa_{t,2}} + C_2}, \tag{3.44}$$

and $\kappa_{st} = \mathrm{E}\Big\{ \big(s(\boldsymbol{x}) - \kappa_{s,1}\big)\big(t(\boldsymbol{x}) - \kappa_{t,1}\big) \Big\}$ is the cross-covariance between s and t. Essentially, these terms provide a comparison of the average luminance (l), a comparison of the average contrast (c), and a structural comparison (s). Typically, the statistics are estimated on rectangular windows of the input images. (3.41) is evaluated for each window, and the resulting

index for all windows are averaged. The CWSSIM is defined in the complex wavelet domain; i.e., (3.41) is evaluated for each window *and* for each subband, and then the values for all windows and subbands are averaged. This type of wavelet representation has shown to be valuable for texture segmentation [Bov91].

Obviously, the luminance (3.42) and contrast (3.43) terms are functions of the statistics of two texture images, while the structure term (3.44) compares the images (or subbands) themselves using the cross-covariance – it cannot be written as a function of a number of statistics obtained from each image separately. The cross-covariance isn't invariant with respect to a shift of one of the images; therefore, this term is incompatible with the stationarity assumption of texture. For this reason, Zhao et al. [Zha+08] proposed to remove this term for texture comparison. The STSIM ("structural texture similarity index") is defined analogously to the CWSSIM as a mean over the subbands (and their respective filters h) of a complex wavelet filterbank:

$$D_{\mathrm{STSIM}}(s,t) = \sum_h \frac{1}{H}\, l^{1/4}(h*s, h*t)\, c^{1/4}(h*s, h*t)\, c_{\mathrm{h}}^{1/4}(h*s, h*t)\, c_{\mathrm{v}}^{1/4}(h*s, h*t), \tag{3.45}$$

where l and c are defined as above and c_{h} and c_{v}, defined as below, replace the structure term:

$$c_{\mathrm{h}}(s,t) = 1 - \frac{1}{2}\left|\frac{\psi_{s,2}\left(\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right)}{\kappa_{s,2}} - \frac{\psi_{t,2}\left(\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right)}{\kappa_{t,2}}\right|, \tag{3.46}$$

$$c_{\mathrm{v}}(s,t) = 1 - \frac{1}{2}\left|\frac{\psi_{s,2}\left(\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right)}{\kappa_{s,2}} - \frac{\psi_{t,2}\left(\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right)}{\kappa_{t,2}}\right|. \tag{3.47}$$

The summary statistics used in (3.42), (3.43), (3.46), and (3.47) can be written in terms of the texture PSDs due to the Wiener–Lee Theorem,

$$\kappa_{h*t,1} = \dot{\mu}_{h,1}\, \kappa_{t,1}, \tag{3.48}$$

$$\kappa_{h*t,2} = \int_{\square} \dot{\Phi}_{h,2}(e^{\jmath 2\pi f})\, \Psi_{t,2}(e^{\jmath 2\pi f})\, \mathrm{d}\boldsymbol{f}, \tag{3.49}$$

$$\psi_{h*t,2}\left(\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right) = \int_{\square} \dot{\Phi}_{h,2}(e^{\jmath 2\pi f})\, \Psi_{t,2}(e^{\jmath 2\pi f})\, e^{\jmath 2\pi f_0}\, \mathrm{d}\boldsymbol{f}, \tag{3.50}$$

$$\psi_{h*t,2}\left(\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right) = \int_{\square} \dot{\Phi}_{h,2}(e^{\jmath 2\pi f})\, \Psi_{t,2}(e^{\jmath 2\pi f})\, e^{\jmath 2\pi f_1}\, \mathrm{d}\boldsymbol{f}, \tag{3.51}$$

and, therefore, we may interpret the STSIM (at least, when the statistics are estimated on large windows) as a spectral distance.

3.5.4 SSTSIM

We define the SSTSIM, a simplification of the STSIM, by dropping the directional components:

$$D_{\mathrm{SSTSIM}}(s,t) = \sum_h \frac{1}{H}\, l^{1/4}(h*s, h*t)\, c^{1/4}(h*s, h*t). \tag{3.52}$$

Note that, when comparing zero-mean fields, or when neglecting the lowpass subband, the luminance component can also be dropped as its value is bound to be 1.

3.5.5 Magnitude and power root mean square error

c, as a function of subband power, is convex and it is therefore generally possible to find its unique optimal point using iterative algorithms. Consider a quadratic function of subband power: This allows finding its unique optimal point with even less computationally complex methods. One such function is:

$$D_{\mathrm{PRMSE}}(s,t) = \left(\sum_h \frac{1}{H\dot{\mu}_{h,2}^2} \left(\mu_{s*h,2} - \mu_{t*h,2} \right)^2 \right)^{1/2}. \tag{3.53}$$

This function is not precisely a quadratic function of subband power due to the square root operation. However, since the square root is a monotonic function (and the value of the expression inside cannot be negative), its optimal point can be found in the same way by minimizing the inner expression, which is quadratic. A similar argument holds for this metric, which measures spectral magnitude as opposed to power:

$$D_{\mathrm{MRMSE}}(s,t) = \left(\sum_h \frac{1}{H\dot{\mu}_{h,2}} \left(\sqrt{\mu_{s*h,2}} - \sqrt{\mu_{t*h,2}} \right)^2 \right)^{1/2}. \tag{3.54}$$

3.6 Subjective similarity of Gauss–Markov Texture

With respect to validity, the objective similarity metrics introduced in Section 3.5 are all the same. Except for the fact that some of them may be less complex to evaluate than others, none of them is to be preferred over another, unless an application provides a reason to do so. As this work deals with compression of Gauss–Markov Texture, an important application-specific criterion is "perceived" similarity. Thus, it is useful to evaluate these metrics against subjective similarity. A study examining exactly this question was conducted with 25 test subjects. The results were published in [Bal12].

The studied similarity metrics comprised the Itakura Distance (3.38), the log-spectral distance (3.40), the STSIM, the simplified STSIM, and the magnitude and power root mean square error metrics. They were evaluated for pairs of AR models s and t, where $\sigma_s = \sigma_t = 1$. The two spectral distances were further diversified using different weightings, yielding the effective distances

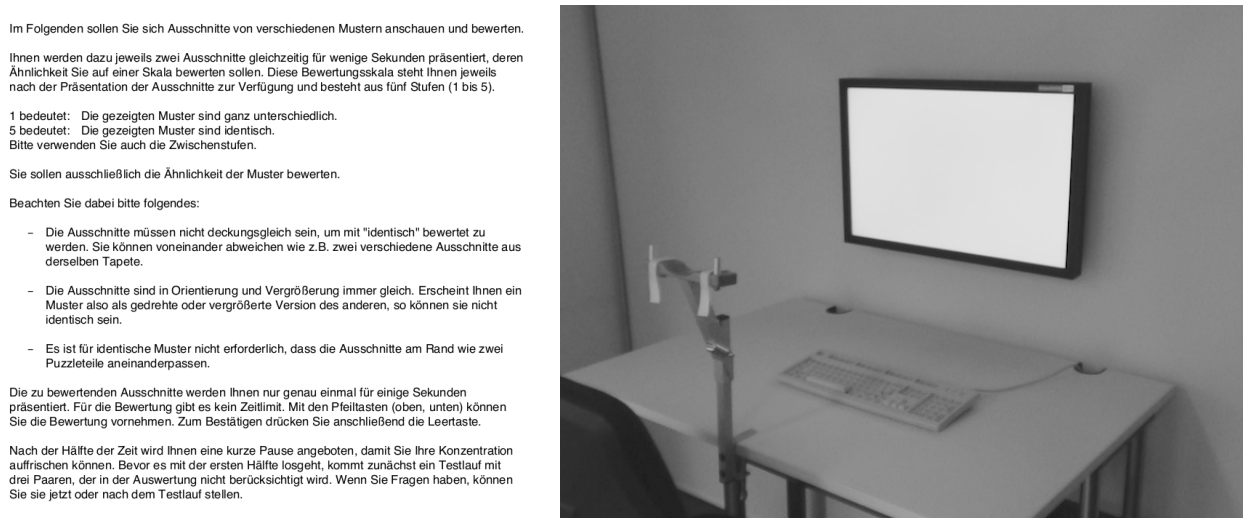
$$D_{\mathrm{WI}}(\boldsymbol{b}_s, \boldsymbol{b}_t) = \left| \ln \left(\int_{\square} W(\boldsymbol{f}) \frac{\left| 1 + A_s(e^{\jmath 2\pi f}) \right|^2}{\left| 1 + A_t(e^{\jmath 2\pi f}) \right|^2} \, \mathrm{d}\boldsymbol{f} \right) \right| \tag{3.55}$$

and

$$D_{\mathrm{WLSD}}(\boldsymbol{b}_s, \boldsymbol{b}_t) = 10 \sqrt{\int_{\square} W(\boldsymbol{f}) \left(\log_{10} \frac{\left| 1 + A_s(e^{\jmath 2\pi f}) \right|^2}{\left| 1 + A_t(e^{\jmath 2\pi f}) \right|^2} \right)^2 \mathrm{d}\boldsymbol{f}}, \tag{3.56}$$

where

$$W(\boldsymbol{f}) = \frac{1}{|\boldsymbol{f}|^k}, \qquad k \in \{0, 1, 2, 4\}. \tag{3.57}$$



(a) Introductory text (b) Physical setup

Figure 3.3 Experimental setup.

Note that the Itakura Distance is non-symmetric. Therefore, it was additionally evaluated with reversed arguments (i.e., with swapped numerator and denominator, designated 'A' and 'B' in what follows).

The STSIM was computed as defined in (3.45) using (3.48) through (3.51) and a logarithmic Gabor-like filterbank (c.f. Chapter 4), the steerable pyramid [Sim+92], with a number of scales (s) and orientations (o). Different combinations of scales and orientations were used. All metrics were evaluated by computing a 1024×1024 discretization of the PSDs of both texture models and approximating the integrals as sums.

3.6.1 Experimental setup and analysis

Subjective scores of texture similarity were acquired in a room with a uniform gray background with dim environmental lighting (Figure 3.3b), using an EIZO SX3031W display set to sRGB color space, gamma 2.1, connected to an Apple Mac Pro 2.66 GHz Quad-Core Intel Xeon. The test subjects were asked to rate the similarity of pairs of texture images on a scale of 1 to 5 using the up and down arrow keys and confirm their selection using the space bar. All keys except the arrow keys and the space bar were removed from the keyboard. A chin rest was used to ensure a controlled viewing distance of approximately 80 cm.

Before being introduced to the experiment, test subjects were checked for visual acuity using the Freiburg Vision Test [Bac96] version 3.7.1 using a viewing distance of 1.7 m. The decimal visual acuity of all test subjects was determined to be at least 1.2, such that the resolution of the display was just below the visual discrimination capabilities of the test subjects, or lower.

3.6.2 Prior distribution of texture model parameters

In earlier work [FB11], it was observed that the marginal prior distribution of GMRF texture parameters occurring in natural images resembles a double exponential (Laplace) distribution.

The texture models were obtained by employing a Monte Carlo technique. The first texture model of each pair was generated by randomly drawing GMRF parameters from a Laplace distribution. Since textures occurring in natural images do not contain arbitrarily high peaks in their power spectral density functions, a multi-start gradient descent method was used to estimate the global spectral maximum of each generated texture model. New candidate model parameters were randomly drawn until a model was obtained whose power spectral density is bounded by a threshold.

To find the second model of each pair, the parameters of the first model were perturbed by a vector of independent Gaussian noise, simulating an arbitrary quantization [FB11] of the parameters. The second texture model candidates were subjected to the same routine of spectral analysis to avoid high peaks in the second model, as well.

In an exploratory analysis, STSIM appeared to be the best objective measure. To ensure that the stimuli were approximately equally distributed on the objective and subjective scales, the following procedure was used: After a pair of texture models passed the spectral analysis, the STSIM between the two models was calculated and the pair was classified according to its STSIM value into 30 classes ranging from a STSIM of .85 to an STSIM of 1. The generation of new model pairs was iterated until there was at least one model pair in each class. All except the model pair closest to the central STSIM value of each class were discarded, such that the remaining model pairs encompassed approximately equally spaced STSIM values between .85 and 1. These 30 model pairs were used to generate all double stimuli.

3.6.3 Presentation of stimuli

Each double stimulus consisted of a pair of zero-mean homogeneous texture images of 1024×1024 pixels displayed side-by-side (with a gap in-between) for exactly 6 s. The background was plain gray (corresponding to the zero level of the images). There was no time limit for the rating, and a 2 s pause between the confirmation of each rating and the next double stimulus. Each stimulus was generated by computing the PSD of the corresponding texture model on a discrete grid the same size as the image, taking its square root, multiplying by the DFT of an independent Gaussian noise array (which was randomly generated for each stimulus), and inverse transformation. This is effectively the same algorithm as used in [GGM11].

To each test subject, 6 runs of double stimuli were presented, where each run consisted of 30 double stimuli. Within each run, each of the 30 model pairs were used exactly once, while their order was randomly permuted for each run. The test subjects were asked to rate the visual similarity of the double stimuli on a scale from 1 to 5, where they were explicitly asked not to determine whether the two images were the same, but rather whether they looked similar (Figure 3.3a reproduces the text shown to the subjects prior to the beginning of the rating).

3.6.4 Analysis of subjective scores

The experiment was conducted with 25 test subjects. Their responses were analyzed as follows. Firstly, the scores from the first run of double stimuli were discarded for each test subject in order to allow them to adapt to their task without having an effect on the results. Then, the average standard deviation of the remaining responses of each subject was analyzed to check for consistency.

A	1.00													
B*	0.90	1.00												
C	0.91	0.73	1.00											
D	0.93	0.85	0.94	1.00										
E	0.94	0.87	0.91	0.96	1.00									
F	0.95	0.83	0.95	0.95	0.94	1.00								
G	0.97	0.86	0.91	0.94	0.95	0.94	1.00							
H	0.97	0.88	0.92	0.96	0.96	0.95	0.95	1.00						
I	0.97	0.85	0.94	0.94	0.94	0.95	0.97	0.95	1.00					
J*	0.27	0.32	0.10	0.22	0.28	0.19	0.28	0.25	0.22	1.00				
K	0.93	0.79	0.90	0.92	0.93	0.91	0.96	0.90	0.95	0.31	1.00			
L	0.96	0.89	0.92	0.97	0.97	0.95	0.97	0.97	0.96	0.24	0.94	1.00		
M	0.93	0.78	0.91	0.92	0.95	0.93	0.95	0.91	0.94	0.30	0.97	0.94	1.00	
N	0.95	0.88	0.93	0.95	0.95	0.95	0.93	0.96	0.94	0.22	0.91	0.96	0.90	1.00
O	0.92	0.77	0.95	0.92	0.93	0.92	0.92	0.96	0.93	0.17	0.91	0.94	0.90	0.94
P	0.91	0.77	0.90	0.92	0.94	0.93	0.92	0.91	0.93	0.31	0.92	0.92	0.92	0.89
Q	0.91	0.86	0.90	0.95	0.93	0.94	0.92	0.92	0.90	0.24	0.88	0.95	0.90	0.94
R	0.97	0.90	0.91	0.95	0.94	0.95	0.95	0.98	0.94	0.26	0.89	0.96	0.89	0.96
S*	0.84	0.79	0.71	0.81	0.81	0.78	0.85	0.81	0.83	0.55	0.83	0.82	0.83	0.75
T	0.89	0.75	0.93	0.93	0.93	0.92	0.90	0.89	0.91	0.28	0.90	0.91	0.91	0.92
U*	0.87	0.85	0.86	0.92	0.88	0.90	0.84	0.91	0.89	0.10	0.81	0.89	0.82	0.92
V*	0.74	0.58	0.65	0.66	0.70	0.73	0.72	0.70	0.70	0.61	0.77	0.69	0.75	0.64
W	0.91	0.82	0.91	0.93	0.93	0.94	0.91	0.92	0.89	0.29	0.89	0.94	0.90	0.94
X*	0.89	0.98	0.73	0.85	0.86	0.81	0.86	0.87	0.83	0.33	0.79	0.89	0.78	0.85
Y	0.96	0.79	0.95	0.93	0.92	0.94	0.96	0.95	0.97	0.20	0.96	0.95	0.94	0.94
	A	B*	C	D	E	F	G	H	I	J*	K	L	M	N

O	1.00										
P	0.89	1.00									
Q	0.87	0.90	1.00								
R	0.93	0.88	0.94	1.00							
S*	0.71	0.84	0.77	0.79	1.00						
T	0.88	0.91	0.93	0.89	0.74	1.00					
U*	0.86	0.87	0.90	0.88	0.70	0.83	1.00				
V*	0.64	0.78	0.65	0.67	0.84	0.70	0.53	1.00			
W	0.89	0.88	0.97	0.94	0.75	0.93	0.85	0.68	1.00		
X*	0.75	0.77	0.87	0.90	0.79	0.76	0.79	0.58	0.82	1.00	
Y	0.96	0.92	0.90	0.93	0.80	0.89	0.86	0.72	0.91	0.78	1.00
	O	P	Q	R	S*	T	U*	V*	W	X*	Y

Table 3.2 Subject–subject correlation (Pearson's ρ). Due to symmetry, the upper half of the matrix has been omitted. Rows and columns marked with an asterisk indicate test subjects whose answers have been ignored.

A certain variance of the responses can be explained due to the quantization of the response scale. For example, a test subject may give alternate responses between two scores if the subjective similarity is in-between. In the worst case, this results in a standard deviation of 0.5. Three of the test subjects responded with an average standard deviation above this value. Further analysis also showed that the responses of these test subjects were badly correlated to the responses of all other subjects (Pearson's and Spearman's $\rho < 0.9$). These test subjects were therefore ignored (Table 3.2, subjects J, U, V). Another three subjects with bad correlation to the rest were also ignored (subjects B, S, X). Analysis showed that their responses were consistent and in principle along the same lines as the others, but either much more conservative or more radical (for example, answering with a score of 1 to more than 60% of all double stimuli).

To counteract singular outliers, the scores were averaged by removing the highest and the lowest score for each test subject and model pair (i.e., removing two of five responses for each subject and model pair) and then taking the arithmetic mean of the remaining values, either per-subject for the subject–subject correlation or across all 19 remaining subjects for the objective–subjective correlation. The mean subject–subject correlation coefficient was computed by taking the arithmetic mean of the correlation coefficients between the $^{19(19-1)}/_{2} = 171$ pairs of non-identical test subjects. It was found to be $\rho = 0.930$ (Pearson's), $\rho = 0.924$ (Spearman's), and $\tau = 0.838$ (Kendall's). Examples of textures and their average subjective scores are given in Figure 3.4.

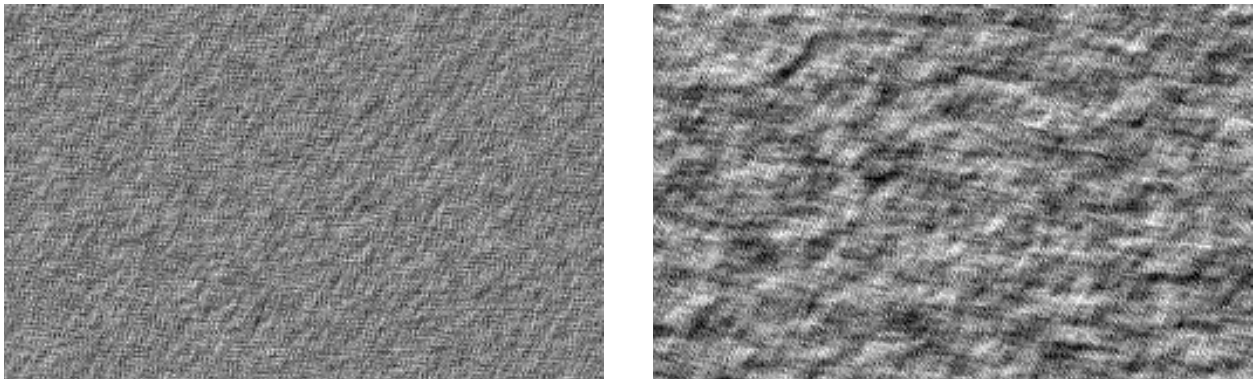
3.6.5 Results

The average subjective scores were compared to the objective metrics using Spearman's ρ and Kendall's τ (Table 3.3). The tables show that all metrics perform well capturing the characteristics of texture important to the human observers when the appropriate weighting or filterbank resolution is used. An exception is the RMSE of subband power, which does not appear to provide a good correlation regardless of the filterbank resolution.

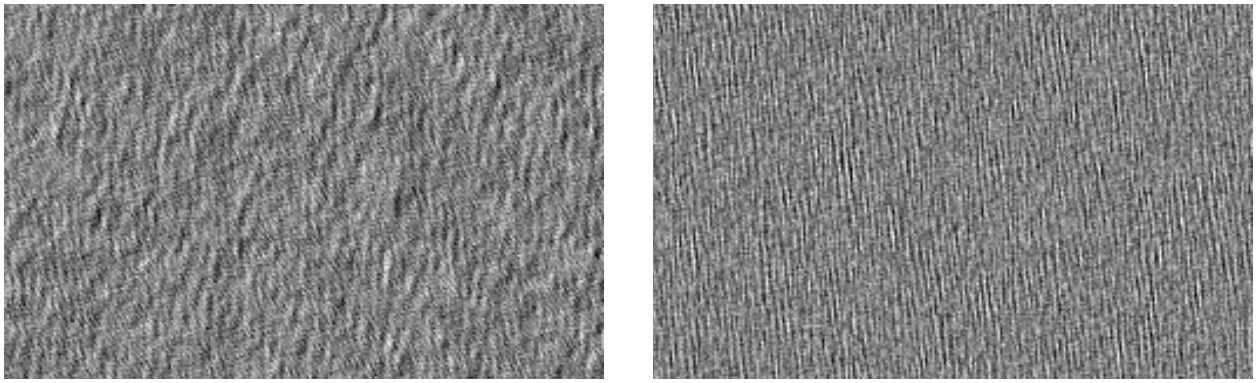
In Figure 3.5, the average subjective score is plotted against some of the metrics from the correlation tables. The $k = 2$ weighting should be expected to correlate well to the subjective scores, as this corresponds to the scale-invariant statistics of natural images and the properties of visual cortex cells [Fie87] (c.f. Chapter 4). This property is also reflected in the filterbank that was used.

Interestingly, the $k = 4$ weighting correlates even better to subjective scores. This implies that test subjects generally attributed an even greater importance to low-frequent components of texture. There is no obvious explanation for this. The prior selection of texture parameters may have some impact on the results. It is plausible that some of the less significant differences between correlation values (such as between different filterbank parameterizations or even between weighting parameters $k = 2$ and $k = 4$) may not be reproducible for spectra that are generated with a different method. More stimuli may be necessary to provide better distinctions between close figures. The fact that the Itakura distance achieves much worse correlation to subjective scores than the LSD, *except* for $k = 4$, may be due to similar reasons. To understand this effect, the study should be repeated using a different prior distribution of texture PSDs.

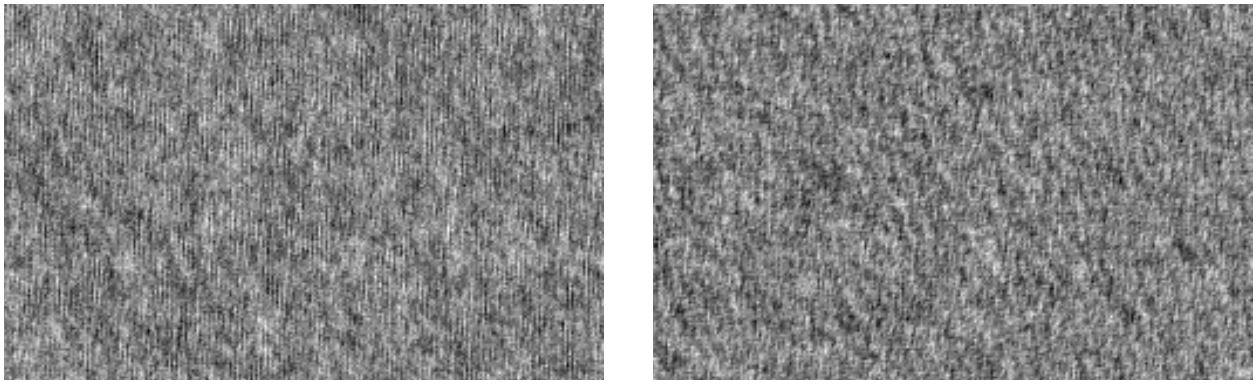
Nonetheless, we may draw three conclusions from this experiment:



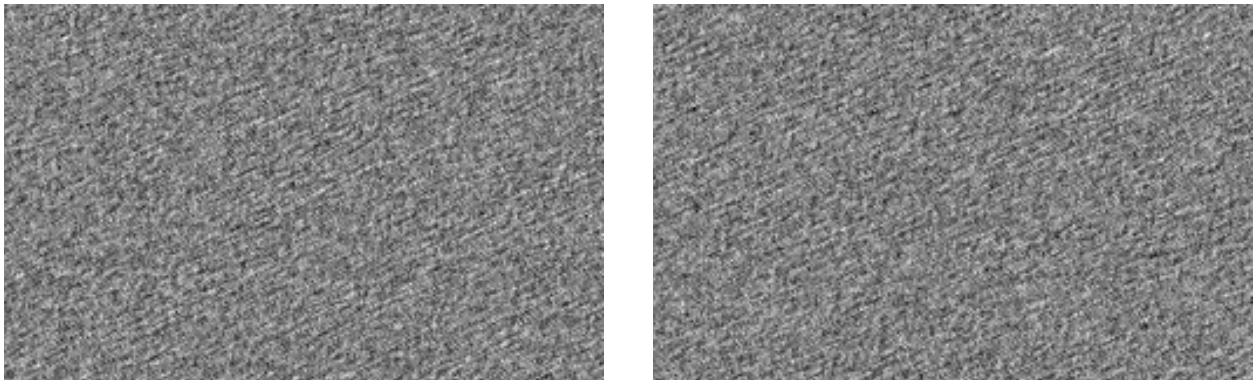
Avg. Subjective Score 1.04



Avg. Subjective Score 2.46



Avg. Subjective Score 4.19



Avg. Subjective Score 5.00

Figure 3.4 Samples of the texture model pairs and corresponding average subjective scores.

	LSD		Itakura A		Itakura B	
k	ρ	τ	ρ	τ	ρ	τ
0	0.086	0.060	0.213	0.157	0.090	0.078
1	0.689	0.520	0.387	0.285	0.108	0.120
2	0.957	0.833	0.788	0.626	0.730	0.575
4	**0.964**	**0.852**	**0.964**	**0.852**	**0.964**	**0.852**

(a) Weighted spectral distances

	STSIM		SSTSIM		Mag. RMSE		Pow. RMSE	
s/o	ρ	τ	ρ	τ	ρ	τ	ρ	τ
6/6	0.853	0.681	0.874	0.709	0.905	0.755	0.103	0.060
6/8	0.856	0.686	0.865	0.700	0.897	0.741	**0.113**	**0.069**
8/8	0.938	0.806	**0.940**	**0.815**	**0.913**	0.764	**0.113**	**0.069**
8/10	**0.939**	**0.810**	0.934	0.806	0.910	**0.769**	0.112	0.064

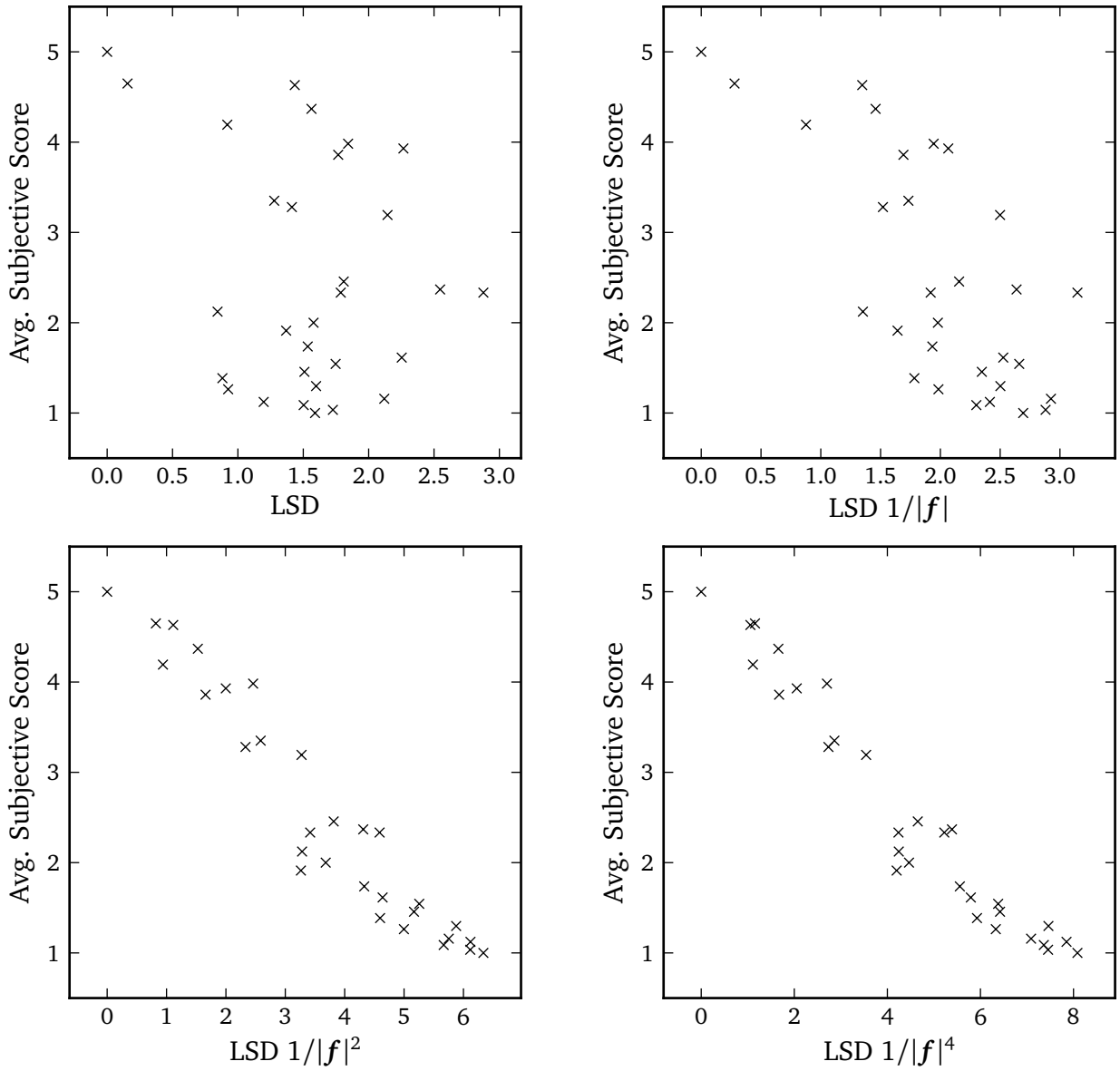
(b) Filterbank-based metrics

Table 3.3 Correlation between average subjective scores and similarity metrics.

- The fact that 22 of 25 test subjects responded very consistently with respect to stimuli that were generated from the same texture model, *but using a different driving random field*, suggests that the randomness introduced into each stimulus had no effect on subjective similarity. Thus, we have empirical evidence that the perception of a Gaussian random field is determined by its statistical structure – i.e., by its power spectral density – rather than by phase, which is an attribute of each instance of the field, in addition to the theoretical result of Section 3.1.

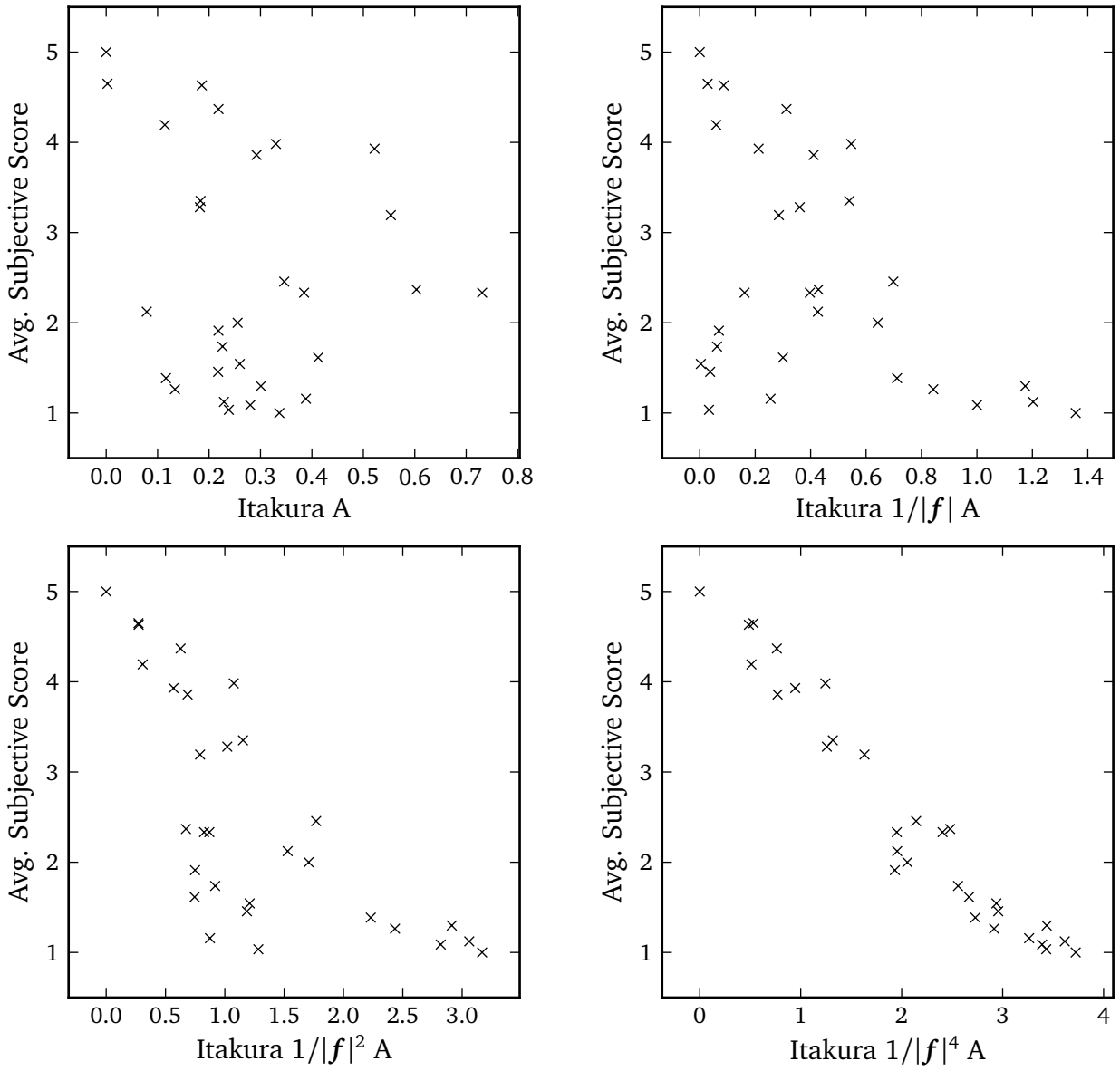
- The difference between the metrics we have considered is the way spectral magnitudes are mapped onto the metric value. If the metric covers the entire spectrum, it is capable of detecting the differences of two different random fields. Differences between two Gaussian random fields that do not manifest in a different PSD cannot exist. Therefore, we have the answer to Question 4 from the introduction. Of course, the exact metric that should be used must still be evaluated empirically.

- No matter which of the metrics is used, to reach good correlation to subjective scores, it appears to be essential to give a lower weight to the spatial frequencies of higher magnitude. This can be achieved by weighting of traditional spectral distances or by filterbank design – the octave-band structure of the steerable filterbank can be shown to be roughly equivalent to a weighting parameterized with $k = 2$ [Fie87]. We conclude that it is not advisable to rely on the $k = 4$ weighting as long there is no convincing interpretation for it.

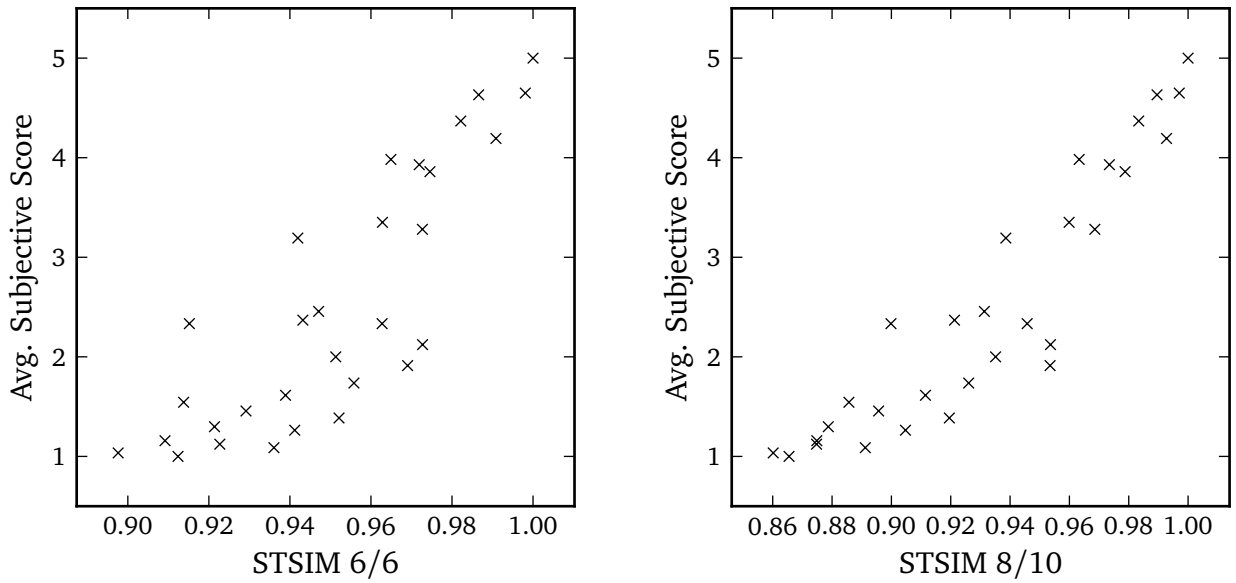


(a) Log-spectral distance

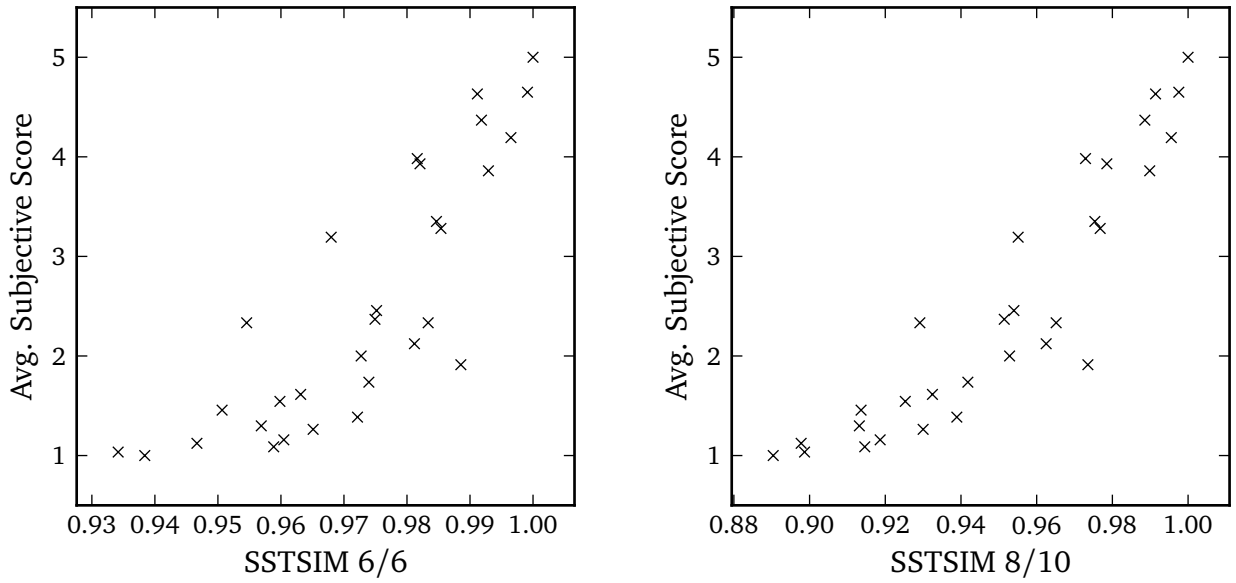
Figure 3.5 Scatter plots for subjective scores vs. similarity metrics (continued on next page).



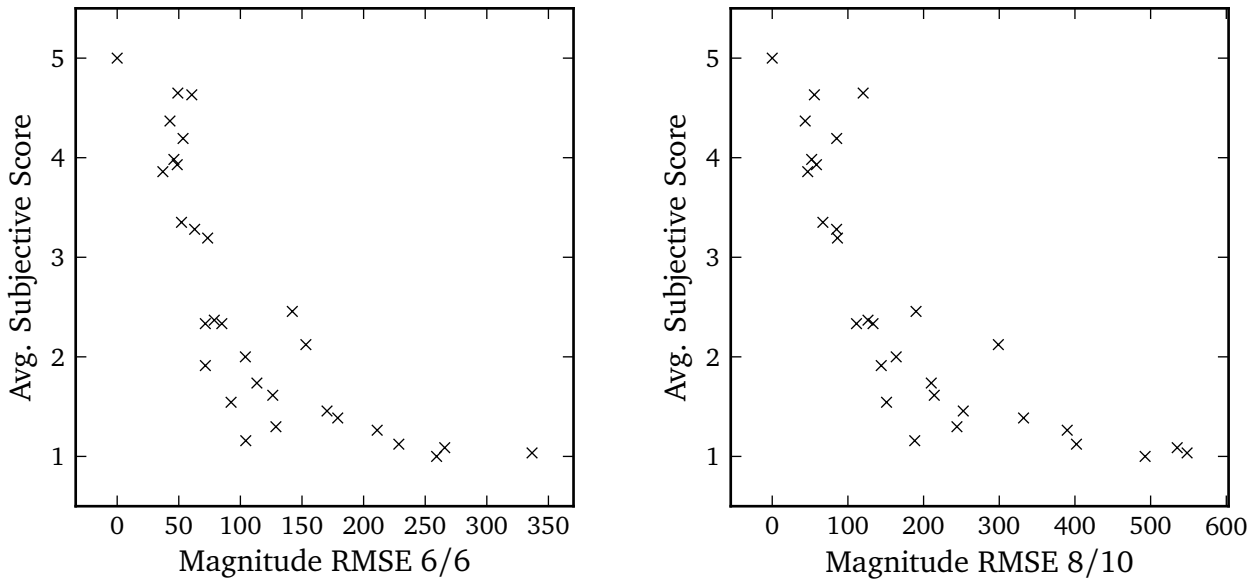
(b) Itakura Distance



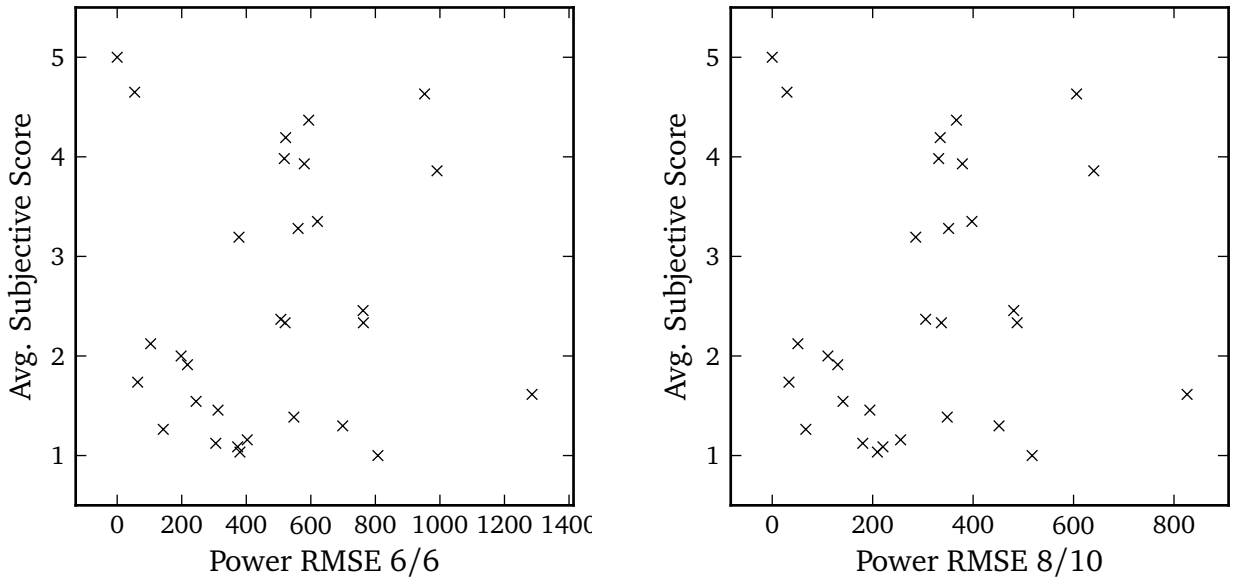
(c) STSIM



(d) Simplified STSIM



(e) Magnitude RMSE



(f) Power RMSE

4 Image analysis

In this chapter, we review some of the physiological fundamentals of the human visual system (HVS), as well as established methods for feature detection in natural images. What is interesting is the special role Gaussian fields take in the feature detection context: Gaussian texture, in a sense, is the "opposite" of a salient feature. This interpretation is important for the main purpose of this work, texture compression in natural images, as it equips us with a rationale for why Gaussian texture is unlikely to comprise semantic meaning – simply because it is extremely unlikely to even comprise a visual feature. Furthermore, we derive a method to specifically detect Gaussian or linear random fields in natural images, answering Question 1 from the introduction.

4.1 Biological vision

In the last few decades, there have been numerous studies on the neurophysiological workings of the human visual system. A particularly important insight that was gained is that specific properties of the HVS can be explained by the fact that the HVS was (and continually is) adapted to the statistics of the input it gets from our natural environment. The statistics of natural images appear to be one of the major factors that constitute the optimizing function for the HVS [Fie87; OF96]. Other factors, such as metabolic efficiency [GF08], may exist as well.

[Ser+05] provides a preliminary, but quantitative, computational model of the ventral stream pathway, which is believed to be responsible for visual recognition tasks, in an attempt to explain the most basic phenomena that have been observed. The model is limited to a feed-forward architecture, meaning that the processing of visual information is organized in a bottom-up fashion (from photoreceptor cells upward to the pre-frontal cortex which is associated with high-level tasks such as face recognition). It is believed that "backprojections," i.e., feedback from higher-level components down, are playing a role in biological vision, particularly for answering questions such as "Is the object in the scene an animal?" or determining sizes of objects [Ser+05, page 6]. However, a common conception is that the first 150 ms of visual processing following the onset of a stimulus can be sufficiently explained by forward-only models. Conceptually, this corresponds to Julesz's "pre-attentive vision" [Jul81; Jul91], i.e., vision that takes place without the subject devoting attention to a particular element or feature of the scene.

The structure of the model incorporates interleaved layers of two basic types of neurons, *simple cells* and *complex cells*, where simple cells constitute the first layer of processing in the visual cortex. The activity levels of both kinds of cells can, according to the model, be explained by the equation

$$y = g\left(\frac{\sum_{j=1}^{n} w_j x_j^p}{k + \left(\sum_{j=1}^{n} x_j^q\right)^r}\right), \tag{4.1}$$

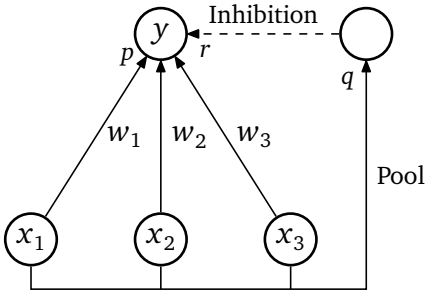


Figure 4.1 Model of neuronal gain control; after [Ser+05]. With $p \leq qr$, the model approaches tuning behavior.

where y, x, and w correspond to the output, the inputs, and the input weights, respectively, and

$$g(t) = \frac{1}{1 + e^{\alpha(t-\beta)}} \tag{4.2}$$

is a sigmoid function. The other variables are constants determining the behavior of the cell model. Generally, simple cells realize a *tuning* operation: The output achieves a maximum when the inputs are collinear to the weights. Therefore, the weights correspond to the *preferred stimulus* of the cell. Output activity of the cell is inhibited when there is elevated, "pooled" activity on the inputs (Figure 4.1). Equivalently, simple cells possess a neuronal gain control mechanism which is controlled through activity at the input cells. For example, to obtain an approximation to simple cell behavior, we may let $p = 1$, $r = 1$, $q = 2$. This particular choice results in the argument of the sigmoid function resembling a Pearson Correlation, provided k is chosen such that it corresponds to the energy of the weights. The first layer of simple cells in V1, the primary visual cortex, could thus be interpreted as a kind of normalized *matched filter*.

It should be stressed though, that it is difficult to select one "best" model. Rather, simulations of biological object recognition show that different cell models can achieve similar recognition quality [Ser+05]. The fact that the preferred stimuli of the first layer of simple cells possess remarkable similarity to Gabor Functions has been known since the early 1960s – for the visual cortex of cats and macaque monkeys. Later, these findings could be confirmed for the human brain [Ser+05]. Furthermore, it was discovered that simple cells are adapted to second-order statistics of images [Fie87] and optimized for sparsity [OF96].

Complex cells, which follow the first simple cell layer, realize a *soft-max* operation which can be modeled by letting $w_i = 1$ and $p = q + 1$ in (4.1), for sufficiently large q. This approximates a maximum operation across the output of simple cells of the same orientation, across different scales, and in spatial proximity. The purpose is to obtain a certain level of invariance with respect to scale and translation. The layer of complex cells is in turn followed by another layer of simple cells which tune to higher-level patterns (Figure 4.2).

4.2 Quadrature feature detection

Among the oldest and most well-known feature detection techniques is "the" Canny Edge Detector [Can86]. It should be noted that Canny's paper describes a mathematical framework

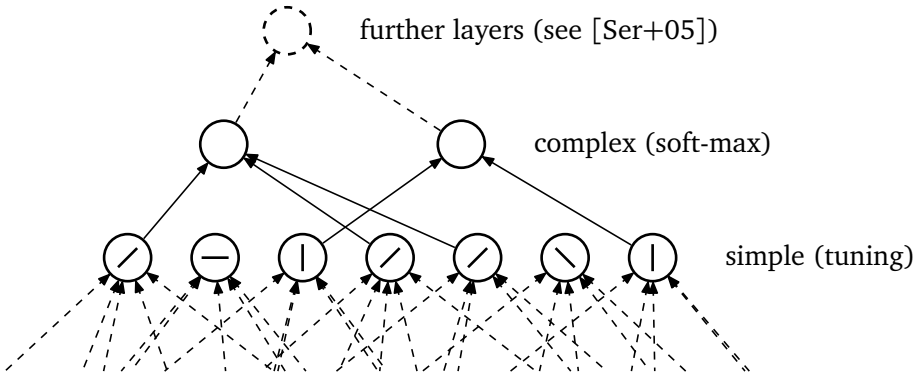


Figure 4.2 Model of first two layers of simple and complex cells in V1; loosely based on [Ser+05]. Simple cells tune to features of specific orientations (indicated by lines) and scales (not shown). Complex cells perform a soft-max operation on the outputs of simple cells of the same orientation and across scales.

to derive a linear filter such that its output will have a maximum wherever the image contains a desired feature: for example, a step edge. For practical purposes, it is often assumed that step edges are the most important visual feature in images, and that other features are negligible. However, using a linear filter that is optimized for detection of step edges blindly can result in systematic errors; for instance, a bar (line) will produce two maxima.

Physiological evidence suggests that feature detection in natural vision systems is more sophisticated. It was found that the perception of edges could be predicted better by filtering with two linear filters in quadrature – one being the Hilbert transform of the other – and then detecting maxima in the squared response:

$$e(x) = \left[(y * h)(x)\right]^2 + \left[(y * h')^2(x)\right]^2, \tag{4.3}$$

where h and h' is the filter pair and $e(\boldsymbol{x})$ is known as *local energy*. This concept generalizes the definition of a "visual feature" beyond step edges and bars; the local energy function attains a maximum for step edges, bars, but also for other luminance profiles. This model was first verified against human perception using one-dimensional sine-wave, step, and trapezoidal gratings [Mor+86; MO87], i.e. on synthetic, periodic, one-dimensional luminance profiles. Appropriately, Morrone and Owens based their theory on Fourier Analysis. They noted that maxima of the local energy function coincide with maxima of the *phase congruency* function,

$$p(x) = \max_{\bar{\phi}(x)} \frac{\sum_n A_n \cos(\phi_n(x) - \bar{\phi}(x))}{\sum_n A_n}, \tag{4.4}$$

where A_n is the amplitude of the nth component of the Fourier series expansion of y, and $\phi_n(x)$ is the local phase of the nth Fourier component at x. $p(x)$ is maximum when the Fourier components are locally in-phase. Once this generalized feature detector has found a maximum of the phase congruency function, the "type" of feature can be inferred from the phase at that location. In [VO90], Venkatesh and Owens give a classification of salient features which are characteristic for specific phase angles. It can be shown that

$$p(x) = \frac{\sqrt{e(x)}}{\sum_n A_n}, \tag{4.5}$$

if $y(x)$ is zero-mean, $h(x) = \delta(x)$, and $h'(x)$ is the Hilbert kernel; thus, Morrone and Owens chose digital filters to approximate this ideal pair.

A more applied approach to feature detection from phase congruency is presented in [Kov99a]. Kovesi replaces the Fourier components by wavelet components, such that there is a number of bandpass filters on different scales; furthermore, the filters are extended to two dimensions and a number of discrete orientations. Wavelet phase congruency (along one orientation) can then be defined as

$$p_o(\boldsymbol{x}) = \frac{\sqrt{\left[\sum_s (y * h_{o,s})(\boldsymbol{x})\right]^2 + \left[\sum_s (y * h'_{o,s})(\boldsymbol{x})\right]^2}}{\sum_s \sqrt{(y * h_s)^2(\boldsymbol{x}) + (y * h'_s)^2(\boldsymbol{x})}}, \tag{4.6}$$

where o and s are the orientation and scale indexes, respectively.[1] Using a wavelet representation is more appropriate than using Fourier analysis, as subband filters can be designed to be scale-invariant – although the human visual system is not exactly scale-invariant, the physics of natural image formation suggest that image features appear equivalently on all scales. Since it is likely that the HVS is adapted to these characteristics [Fie87], scale invariance is a reasonable abstraction of the properties of the HVS. Kovesi uses the logarithmic Gabor representation that was suggested by [Fie87], which models the preferred stimuli of the first layer of simple cells in V1. While $h_{o,s}(\boldsymbol{x})$ corresponds to the even-symmetric filters suggested by Field, $h'_{o,s}(\boldsymbol{x})$ is odd-symmetric. In Figure 4.3, cross sections of an even–odd symmetric filter pair are plotted, and Figure 4.4 demonstrates the variation of phase and magnitude of the subband coefficients across typical image features.[2] Kovesi implements both filters using a single convolution with a complex-valued filter $c_{o,s}(\boldsymbol{x}) = h_{o,s}(\boldsymbol{x}) + \jmath h'_{o,s}(\boldsymbol{x})$, which is easy to construct in the Fourier domain by setting one of the "humps" to zero. (4.6) then simplifies to

$$p_o(\boldsymbol{x}) = \frac{\left|\sum_s (y * c_{o,s})(\boldsymbol{x})\right|}{\sum_s \left|(y * c_{o,s})(\boldsymbol{x})\right|}. \tag{4.7}$$

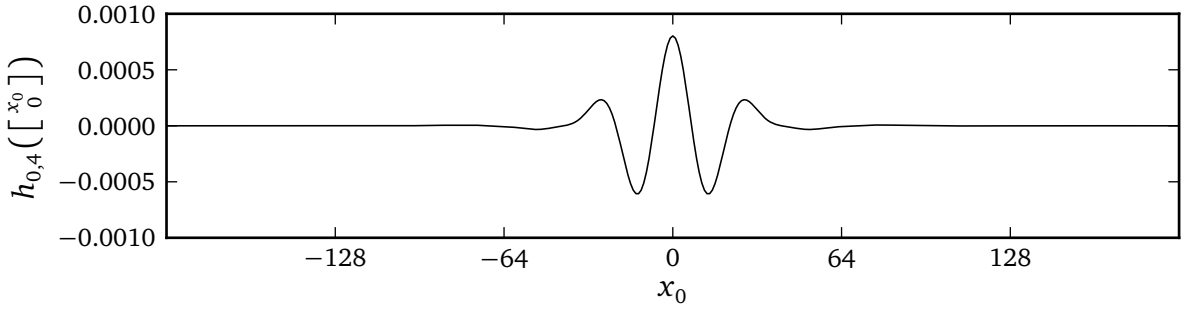
Interestingly, (4.6) and (4.7) bear some resemblance to the neuronal gain control mechanism of the previous section (4.1) in that they contain a normalization term (in the denominator) that is linked to overall activity in the respective subbands.

Although these methods achieve good results detecting and localizing features that cannot be reliably handled using simple linear filters such as the Canny detector, they do not necessarily model biological vision precisely. For instance, a recent study revealed that the edge localizations predicted by the Morrone–Owens Model do not always coincide with human judgements; i.e., depending on the phase of the feature, the model appears to make a systematic prediction error in the form of an angle dependent offset [Geo+07]. This may remind the computer vision expert that, even though edge detection with today's methods may appear to be a routine task – after decades of vision research, many aspects of the HVS have still not been fully uncovered and models that are useful in practice may, in the end, never satisfactorily explain physiological data.

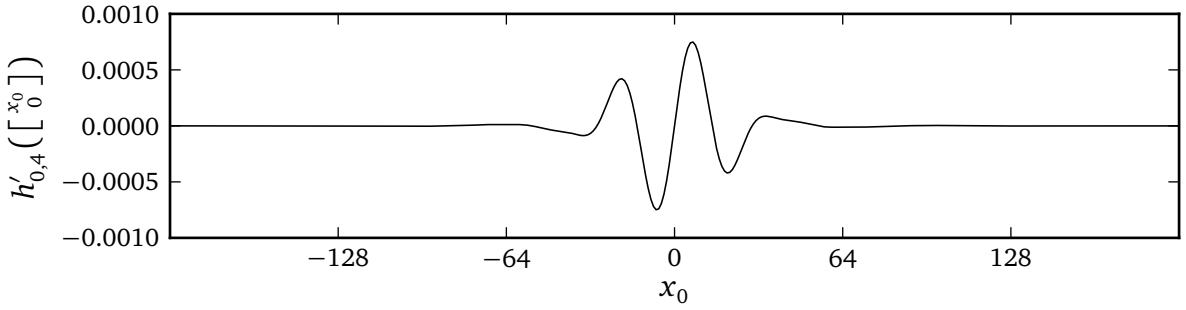
To that end, it is satisfying to note that the exact localization of salient features is not necessary for a compression system utilizing texture synthesis, as we will see in the following sections and chapter.

[1] Note that Kovesi applies further modifications to this formula which we do not reproduce here.

[2] Here, we use other bandpass filters than Gabor, defined in the next chapter, but the results are similar.



(a) symmetric filter



(b) anti-symmetric filter

Figure 4.3 Cross sections of bandpass impulse responses of the logarithmic subband transform used in this thesis; the filters are defined in the next chapter.

4.3 Statistical interpretation

The feature detection method presented in the previous section is purely motivated by physiological observations. However, since there exists a certain relationship to matched filtering, the question arises whether there is a statistical interpretation to the concept. The answer is yes – in fact, this interpretation is key to one of the contributions of this thesis.

Following the signal detection paradigm, we formulate a *null hypothesis*, under the intention to classify all observations as 'no feature present' unless we observe an event that is unlikely enough (under the null hypothesis) so as to exceed a given threshold. The null hypothesis is the following: "The image consists of instationary near-Gaussian noise of unknown power spectral density." The term "instationary" deserves a bit of elaboration here. While we allow the spectral density and variance to change across the image, we must, for all practical purposes, assume that the characteristics of noise are homogeneous if we choose a small enough window in space and/or frequency.

Since we assume near-Gaussian noise, the subband coefficients $(y * h_{o,s})(\boldsymbol{x})$ and $(y * h'_{o,s})(\boldsymbol{x})$ are approximately normally distributed under the null hypothesis (and zero-mean, as we are dealing with bandpass filters).[3] However, we do not know the variance of the coefficients. By assumption, it will vary with location and subband, but it will be approximately constant for

[3] We derive a quantitative prediction of the Gaussianity of the coefficients in Section 4.4.

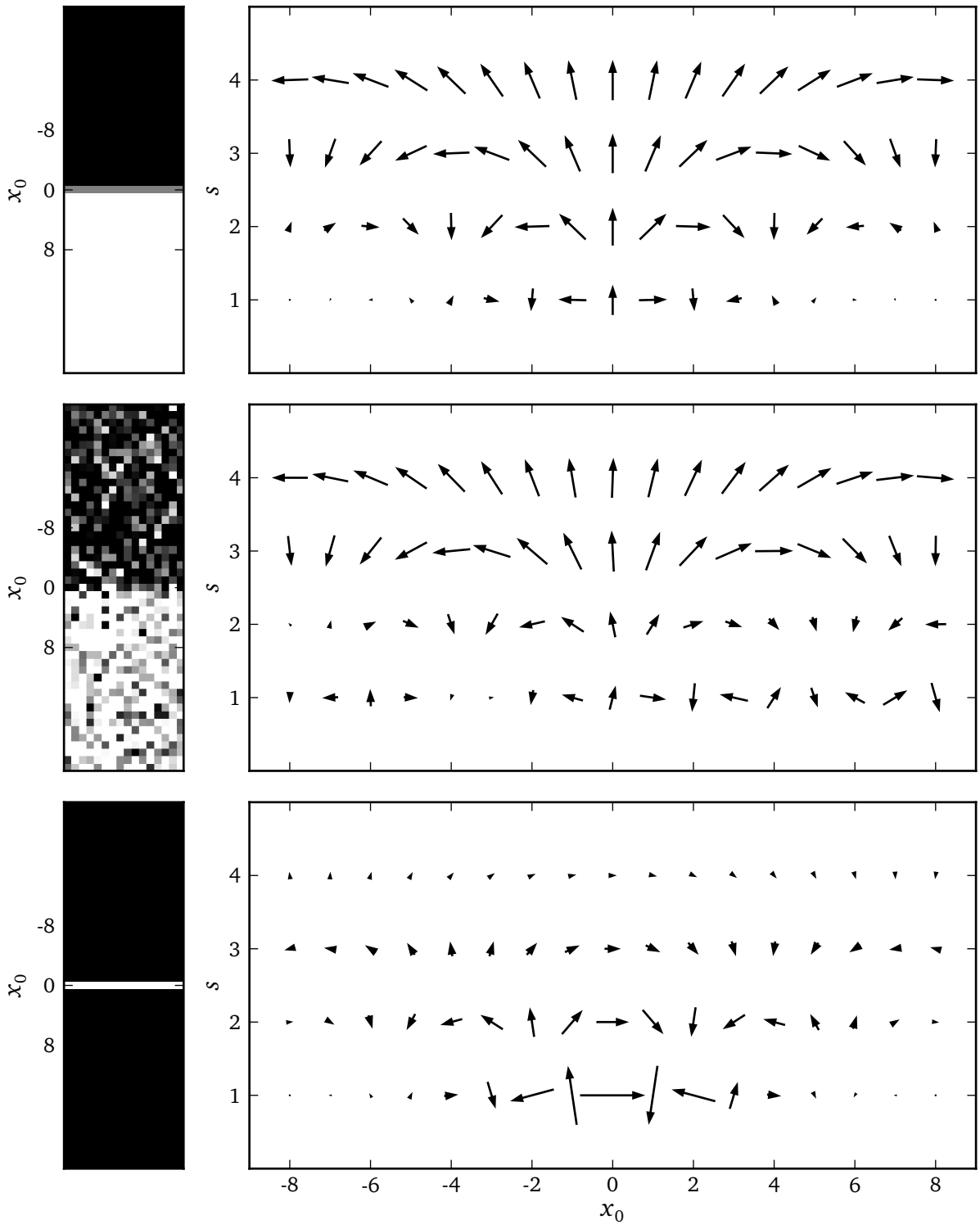


Figure 4.4 Complex bandpass filter responses to various synthetic image features along the orientation orthogonal to the feature. Top: step edge; middle: noisy step edge; bottom: bar. The orientation and length of the vectors correspond to phase and magnitude, respectively, of the complex-valued subband coefficients. Responses at $x_0 = 0$ are maximally phase congruent. The scale index s implies a bandpass center frequency of $|\boldsymbol{f}_c| = 2^{-(s+1)}$.

a small neighborhood in space and frequency. The variance of the even and odd symmetric filters is always the same, which follows from the even symmetry of the power spectral density and the Wiener–Lee theorem. Thus, we can state

$$(y * c_{o,s})(\boldsymbol{x}) \sim \frac{1}{2\pi\sigma_{o,s}^2(\boldsymbol{x})} \exp\left(\frac{-|t|^2}{2\sigma_{o,s}^2(\boldsymbol{x})}\right). \tag{4.8}$$

Because the coefficients are complex normally distributed, their magnitudes are Rayleigh distributed:

$$\left|(y * c_{o,s})(\boldsymbol{x})\right| \sim \frac{t}{\sigma_{o,s}^2(\boldsymbol{x})} \exp\left(\frac{-t^2}{2\sigma_{o,s}^2(\boldsymbol{x})}\right). \tag{4.9}$$

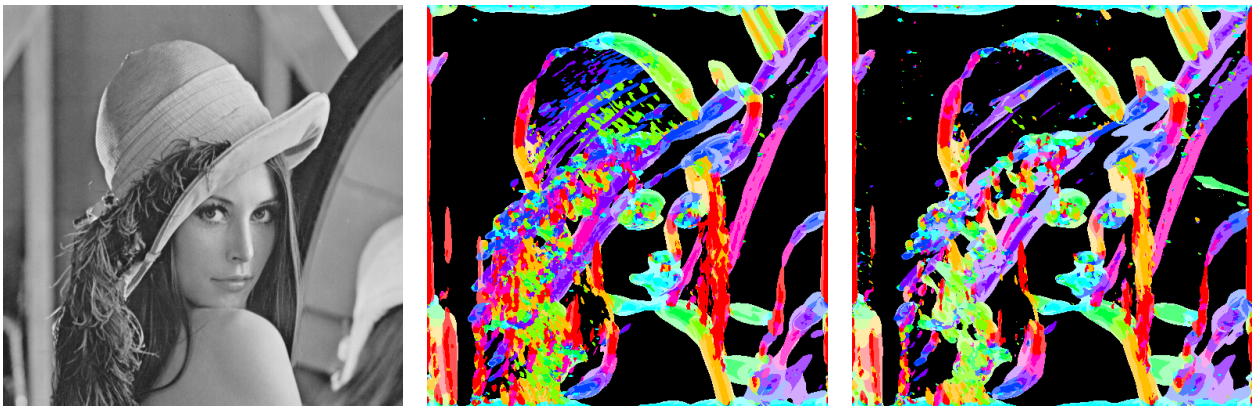
Under the null hypothesis, the magnitude of the standardized coefficient at a location $\boldsymbol{x}$ exceeds a threshold of T with a probability of

$$\mathrm{P}\left(\left|\frac{(y * c_{o,s})(\boldsymbol{x})}{\sigma_{o,s}(\boldsymbol{x})}\right| > T\right) = \int_T^\infty t \, \exp\left(\frac{-t^2}{2}\right) \mathrm{d}t = \exp\left(-\frac{T^2}{2}\right). \tag{4.10}$$

This observation appears to allow a p-test to be constructed: If we concluded that a feature is present on the event *the standardized coefficient exceeds the threshold*, the above probability would represent the probability of error of our conclusion. Unfortunately, there are two problems. Firstly, for standardization of the coefficients, a reasonable estimator for the variance is needed, as the variance cannot be known a priori. Secondly, this approach does not allow us to localize a feature, since y, by the null hypothesis, is correlated, and beyond that, filtering introduces further correlation: The two events *the threshold was exceeded at* $\boldsymbol{x}_0$ and *the threshold was exceeded at* $\boldsymbol{x}_1$ are not statistically independent. Therefore, we cannot call any of these events *statistically significant* in Fisher's sense.

The first problem, again, reminds us of the neuronal gain control mechanism introduced in Section 4.1. It has been tackled implicitly by many authors. Feature detection methods typically follow the assumption that a certain amount of noise in the image is due to the image formation process. "Undesired" noise of this kind, i.e. noise that is not considered part of the signal, leads to a loss of efficiency in most image processing methods. Because of this, many methods apply a thresholding of some kind to ignore the effects of additive noise. In [PM90], a more practical evaluation of the concept presented in [MO87], a heuristic threshold is applied after maxima detection. In [Kov99a], Kovesi applies a somewhat more sophisticated scheme similar to wavelet shrinkage: In the highest-frequency channels of the filterbank, the noise variance is estimated using a robust estimator, based on the assumption that images are composed mostly of untextured regions. He further assumes the image is corrupted by near-Gaussian, stationary white noise. In that case, the noise level in all other subbands can be inferred simply by taking the energy of the subband filters into account. Of course, if one of these assumptions fails, for example due to a high amount of high-frequency texture, the method will overestimate the noise level and consequently, will tend to produce false negatives.

For some applications, such as the one discussed in this thesis, it is, however, questionable whether the normalization of subband coefficients should be based on a whiteness assumption. Image compression methods are ultimately designed to preserve information that is relevant to the observer. If we require an image compression method to distinguish between relevant



(a) original (b) global variance estimation (c) local variance estimation

Figure 4.5 Visualization of features for the image LENNA. While black indicates that all standardized subband coefficient magnitudes are below threshold, exceedance is indicated by multiple colors. The threshold was selected using (4.13). Different hues stand for different orientations. For (b), global variance estimation was performed per subband. Global estimation leads to detection of features within textured regions (for instance, the hat), while local estimation effectively suppresses these features and allows detection of more subtle features in homogeneous regions (bar feature to the right of the forehead).

image features (to be preserved in the reconstruction) and irrelevant image features (to be dropped), what should be the decisive criterion – firstly, whether the image feature is an artifact of the image formation process, or secondly, whether the observer is able to perceive it? Ideally, the answer should be "both": Artifacts and imperceptible features should be ignored likewise. For the purpose of this thesis, we assume that artifacts of the image formation process do not exist; we merely strive to distinguish between perceptible and imperceptible features.[4] We should therefore select a normalization (i.e., a "noise" variance estimator) that follows the principles behind biological systems.

An idealization of the properties of biological systems is the scale-invariance property. After all, this property explains the logarithmic Fourier domain sensitivity of simple cell responses [Fie87] (although, like all practical systems, the HVS must have limits, so there must be some largest and some smallest scale response). The property implies that an image processing algorithm should be invariant with respect to a scaling of the image. If the sensitivity of the statistical feature detection method should be scale-invariant, we must select a variance estimator that is scale-invariant. Such an estimator follows naturally if we uphold the "unknown PSD" hypothesis, as, then, we cannot infer the "noise level" from one scale to the other.

A feasible estimator is

$$\hat{\sigma}_{o,s}(\boldsymbol{x}) = \frac{1}{\sqrt{\ln 4}} \operatorname*{med}_{\boldsymbol{x}' \in W(\boldsymbol{x},s)} \left| (y * c_{o,s})(\boldsymbol{x}') \right|, \tag{4.11}$$

where $W(\boldsymbol{x}, s)$ is a window that scales with the same factor the impulse responses $c_{o,s}$ scale with, and med denotes the median. Here, we use the fact that there is a direct relationship

[4] In the next chapter, we will see that most of the noise that images contain is regarded as texture by our method. Therefore, we can, to a certain extent, automatically handle noise artifacts of the image formation process.

between the median and the parameter of the Rayleigh distribution, like in [Kov99b]. The difference is that we use observations local in space and frequency, while Kovesi uses one global estimator. A very similar (local) estimator based on the median is used in the texture segmentation method by Hill et al. [HCB03] and later by O'Callaghan and Bull [OB05]. The authors then use the standardized coefficients as a basis for the so-called *texture gradient*, which in turn is used for a watershed algorithm – essentially, texture boundaries tend to be placed where the standardized coefficients attain large magnitudes. Of course, other estimators that use local observations could be used. Figure 4.5 demonstrates the basic difference between local and global variance estimation for feature detection.

The second problem quite frequently occurs in image processing methods that model human vision: formal p-tests are not applicable due to statistical dependence of the observations. Sensing a general pattern, Desolneux et al. [DMM00; DMM08] developed a mathematical framework to formalize the rules of human vision as well as decision making, which appears to be based on *expectation* rather than probability.

The cornerstone of this framework is what the authors call the *Helmholtz Principle*, after Hermann von Helmholtz, a german physician and physicist. Helmholtz postulated that human perception is based on the hypothesis of randomness – some event or feature is considered meaningful if and only if *it could not be expected*. According to Desolneux et al., this last phrase is to be taken literally: they propose to use the expected number of occurrences of an event under the randomness assumption (in their terminology, the *number of false alarms*, NFA), as the quantity to be subjected to a threshold. Since the expectation operator is linear, we can compute this number for a given observation, in our case:

$$\mathrm{NFA} = N \cdot \mathrm{P}\left(\left|\frac{(y * c_{o,s})(\boldsymbol{x})}{\sigma_{o,s}(\boldsymbol{x})}\right| > T\right) = N \exp\left(-\frac{T^2}{2}\right), \tag{4.12}$$

where N denotes the number of data samples that are observed at the same instant.[5] According to [DMM08], an event is ε-meaningful if the number of false alarms of such an event is less than ε, where ε should ultimately be set to one: We only consider events where we would expect less than one false alarm in the observation. To test for locations of 1-meaningful features, we solve for T:

$$T = \sqrt{2 \ln N}. \tag{4.13}$$

We can thus use the theory of *meaningfulness* to choose an appropriate threshold given physiological facts.

In spite of this development, we are still unable to locate the exact position of features, since the algorithm we have developed so far yields a number of *meaningful* feature locations at certain locations in space ($\boldsymbol{x}$) and frequency (o, s) without specifying where the perceived "centers" of the features are.[6] However, we will see in the following chapter, that for the purpose of this thesis, the exact localization of features is not needed.

[5] Desolneux et al. recommend to use the number of pixels in the image; however, for large images and to be physiologically correct, this should probably be an estimate of the number of pixels that are projected onto the fovea.

[6] In [DMM08], the concept of *maximal meaningfulness* is developed to "prune" the set of meaningful alignments. This is achieved by simply requiring that maximally meaningful alignments have the lowest likelihood of being observed under the randomness assumption of all meaningful alignments observed at the same location.

The essential idea that is important here is that feature detection in images can be thought of as a deviation from randomness – particularly, Gaussianity, as the Gaussian distribution is the continuous distribution of maximum entropy. In this sense, texture that is characterized by Gaussian random fields is special, as it is texture at the lowest possible level of perception. The human visual system extracts meaningful information from the natural environment we perceive, and the image features we have discussed above are the basic elements that carry this information. If no such features are present in an image, i.e., none of the simple cells of the first layer in V1 respond with elevated activity, it appears unlikely that an observer could extract a great deal of information from it. This is our answer to Question 2 from the introduction. Naturally, it can only hold with respect to our understanding of the human visual system at the present time.

4.4 Sparsity, kurtosis, and Gaussian texture

The fact that the preferred stimuli of the first layer of simple cells resemble logarithmic band-pass functions can be explained by the requirement of *sparsity*. Olshausen and Field [OF96] used a learning algorithm to obtain an array of linear filters that maximizes sparsity of its output. After verifying the algorithm by feeding it with synthetic training sets, for instance, sparse pixel noise, they could confirm that the algorithm produced Gabor-like functions on receiving segments of natural images as input.

Sparse image representations have been an integral part of research in the image processing community since about the same time. An early example is the matching pursuit algorithm due to Mallat and Zhang [MZ93], which is based on a greedy approach. More recent approaches establish links to the blind source separation problem [Fad+10]. Typically, these approaches work with overcomplete *dictionaries*, i.e., sets of basis functions whose cardinality exceeds the number of pixels in the image. There is a recent tendency to generalize beyond the set of basis functions that are plausible responses of the first simple cell layer. This may be used to achieve a level of modeling that includes a number of higher level layers (like S2 to S4 in [Ser+05]).

The empirical probability density functions of the subband coefficients that the authors of [OF96] observe for natural images happen to be symmetric. This is a good reason to use kurtosis, i.e. fourth-order cumulants, as a measure of sparsity (which the authors do).

We can relate this observation to the theory of random fields in the following way. Assume that the input to the filterbank is a stationary random field $y(\boldsymbol{x})$ (not necessarily white or Gaussian). There is a quantitative relationship between the normalized cumulants of the input field and the normalized cumulants of the subband coefficients $(y * h_{o,s})(\boldsymbol{x})$. Here, we may distinguish three cases:

1. The random field is (correlated) Gaussian: $y(\boldsymbol{x}) = (w * g)(\boldsymbol{x})$, where w is IID Gaussian and g is a linear filter.

2. The random field is linear: $y(\boldsymbol{x}) = (w * g)(\boldsymbol{x})$, where w is IID and g is a linear filter.

3. The random field is stationary, but not necessarily any of the above.

The kth standardized cumulant of the subband coefficients is

$$\begin{aligned}\frac{\kappa_{y*h,k}}{\left(\kappa_{y*h,2}\right)^{k/2}} &= \frac{\int\cdots\int_{\square}\Psi_{y*h,k}(e^{\jmath 2\pi f_1},\ldots,e^{\jmath 2\pi f_{k-1}})\,\mathrm{d}\boldsymbol{f}_1\cdots\,\mathrm{d}\boldsymbol{f}_{k-1}}{\left(\int_{\square}\Psi_{y*h,2}(e^{\jmath 2\pi f})\,\mathrm{d}\boldsymbol{f}\right)^{k/2}}\\ &= \frac{\int\cdots\int_{\square}\Psi_{y,k}(e^{\jmath 2\pi f_1},\ldots,e^{\jmath 2\pi f_{k-1}})\,\dot{\Phi}_{h,k}(e^{\jmath 2\pi f_1},\ldots,e^{\jmath 2\pi f_{k-1}})\,\mathrm{d}\boldsymbol{f}_1\cdots\,\mathrm{d}\boldsymbol{f}_{k-1}}{\left(\int_{\square}\Psi_{y,2}(e^{\jmath 2\pi f})\,\dot{\Phi}_{h,2}(e^{\jmath 2\pi f})\,\mathrm{d}\boldsymbol{f}\right)^{k/2}}\end{aligned}$$

due to the Wiener–Lee Theorem (2.44). H is maximum at $\pm\boldsymbol{f}_c$, its center frequency. Consequently, when k is even, $\dot{\Phi}_{h,k}(e^{\jmath 2\pi f_1},\ldots,e^{\jmath 2\pi f_i},\ldots,e^{\jmath 2\pi f_{k-1}})$ attains its maximum for $\boldsymbol{f}_i=\pm\boldsymbol{f}_c$, provided that the number of negative signs for all i is exactly one less or one more than the number of positive signs, due to (2.39). If H is narrow enough, we can continue by approximating $\Psi_{y,k}$ by its value at that point,

$$\begin{aligned}\frac{\kappa_{y*h,k}}{\left(\kappa_{y*h,2}\right)^{k/2}} &\approx \frac{\Psi_{y,k}(e^{\jmath 2\pi f_c},\ldots,e^{\jmath 2\pi f_c})\int\cdots\int_{\square}\dot{\Phi}_{h,k}(e^{\jmath 2\pi f_1},\ldots,e^{\jmath 2\pi f_{k-1}})\,\mathrm{d}\boldsymbol{f}_1\cdots\,\mathrm{d}\boldsymbol{f}_{k-1}}{\left(\Psi_{y,2}(e^{\jmath 2\pi f_c})\int_{\square}\dot{\Phi}_{h,2}(e^{\jmath 2\pi f})\,\mathrm{d}\boldsymbol{f}\right)^{k/2}}\\ &= \frac{\Psi_{y,k}(e^{\jmath 2\pi f_c},\ldots,e^{\jmath 2\pi f_c})}{\left(\Psi_{y,2}(e^{\jmath 2\pi f_c})\right)^{k/2}}\cdot\frac{\dot{\mu}_{h,k}}{\left(\dot{\mu}_{h,2}\right)^{k/2}}.\end{aligned}\tag{4.14}$$

Empirical observations suggest that the approximation also holds for odd k. In the first two cases, we can, again, apply (2.39):

$$\frac{\Psi_{y,k}(e^{\jmath 2\pi f_c},\ldots,e^{\jmath 2\pi f_c})}{\left(\Psi_{y,2}(e^{\jmath 2\pi f_c})\right)^{k/2}}=\frac{\kappa_{w,k}\,G^k(e^{\jmath 2\pi f_c})}{\left(\kappa_{w,2}\,G^2(e^{\jmath 2\pi f_c})\right)^{k/2}}=\frac{\kappa_{w,k}}{\left(\kappa_{w,2}\right)^{k/2}}.\tag{4.15}$$

Thus, the left hand factor of (4.14) reduces to the standardized cumulant of the driving noise field w. The right hand factor is a constant that only depends on the frequency response of the subband filter. If we design a filter that is narrow enough for the approximation in (4.14) to hold, the cumulants of the IID field w can be inferred by estimating the cumulants of the subband coefficients and then correcting using (4.14). However, we cannot use an arbitrarily narrow filter – the narrower the filter, the stronger will the correction lead to amplification of the variance of the kurtosis estimator. This can be seen in the following way: If we let $\tilde{h}(\boldsymbol{x})=h(a\boldsymbol{x})$, i.e., use a scaled version of the filter,

$$\begin{aligned}\dot{\mu}_{\tilde{h},k} &= \frac{1}{a^{2k}}\int\cdots\int_{\square}H(e^{\jmath\frac{2\pi}{a}f_1})\cdots H(e^{\jmath\frac{2\pi}{a}f_{k-1}})H(e^{-\jmath\frac{2\pi}{a}(f_1+\cdots+f_{k-1})})\,\mathrm{d}\boldsymbol{f}_1\cdots\,\mathrm{d}\boldsymbol{f}_{k-1}\\ &= \frac{1}{a^{2k}}\int\cdots\int_{\square}\dot{\Phi}_{h,k}(e^{\jmath\frac{2\pi}{a}f_1},\ldots,e^{\jmath\frac{2\pi}{a}f_{k-1}})\,\mathrm{d}\boldsymbol{f}_1\cdots\,\mathrm{d}\boldsymbol{f}_{k-1}\\ &= \frac{1}{a^{2k}}\,a^{2(k-1)}\,\dot{\mu}_{h,k}=\frac{1}{a^2}\,\dot{\mu}_{h,k},\end{aligned}\tag{4.16}$$

provided neither h nor $\tilde{h}$ are subject to alias. Therefore,

$$\frac{\dot{\mu}_{\tilde{h},k}}{\left(\dot{\mu}_{\tilde{h},2}\right)^{k/2}}=a^{k-2}\,\frac{\dot{\mu}_{h,k}}{\left(\dot{\mu}_{h,2}\right)^{k/2}}.\tag{4.17}$$

This is a variant of the central limit theorem: The greater the extent of the filter ($a \to 0$), the smaller are the magnitudes of the standardized cumulants of its output. In the limit, as the standardized cumulants approach zero, we arrive at a Gaussian distribution. This fact also serves as a physical explanation for natural occurrence of Gaussian texture. When the physical resolution of the imaging equipment (which can be modeled using a *point spread function*, a linear filter) is not fine enough, accumulations of small objects, such as sand, tend to produce Gaussian texture. A similar observation was made in [GGM11].

(4.14) allows us to conceive a statistical p-test to determine whether a given random field is Gaussian or near-Gaussian (Case 1), by thresholding the standardized kurtosis estimates of the subband coefficients: They are asymptotically zero under the null hypothesis. The principle of scale-invariance, again, demands that we choose the same estimator and threshold on all scales of the filterbank that we use.

The probability of error cannot be obtained analytically, as the probability distribution of estimators of standardized higher-order cumulants is typically intractable [SO87]. However, Monte Carlo simulations can be used. A test for linearity may be feasible, as well – in the second case, the corrected standardized kurtosis estimates of the subband coefficients of a filterbank are asymptotically equally valued.

What is the significance of this with with respect to texture? The development implies that some of the previous research on the essential qualities of visual texture may need to be reviewed. While more recent publications generally take an empirical, non-parametric approach to texture, one previous study used methods based on higher order statistics akin to Hinich's transient detector [Hin90] to determine whether, in general, visual texture could be classified as Gaussian, or linear [HG95]. In the light of the developments in this chapter, it seems, however, that this is not the question that should be asked.

Rather, we should ask "Is there visual texture that we *perceive* to be Gaussian, or linear?". The outcome of a statistical test always depends on the statistic that is used; and texture is a concept so inseparably connected with human perception that we must use models of the HVS to understand it. The authors of [HG95] directly estimate higher-order coherency functions (Section 2.2.6) to perform these tests, while the development above offers a model – an abstraction – of how "a typical human observer" may decide on this question. Basically, this is our answer to Question 1 from the introduction: We are able to detect Gaussian as well as near-Gaussian texture, in a manner that emulates human perception. The system designed in the next chapter can then be optimized to the statistical properties of that model.

A much more general discussion of the characteristics of visual texture is given in [PS00b]. In their work, Portilla and Simoncelli present convincing evidence that second-order statistics alone are not sufficient to capture the characteristics of what is generally considered visual texture.

There are three reasons why, in this thesis, we are still concerned with near-Gaussian – i.e., second-order – texture. Firstly, the fact that texture cannot generally be considered second-order does not imply that second-order texture does not occur in natural images. Secondly, as we have seen above, Gaussian texture is *maximum entropy* texture, while, thirdly, carrying the least possible information for the observer. Therefore, from an image compression standpoint, it should be expected that methods addressing this type of texture should achieve gains compared to conventional methods, while minimizing the risk of compromising semantics.

5 Compression of Gaussian texture in natural images

The development of the previous chapters allows us to design a coding system that

- detects Gaussian or near-Gaussian texture in natural images in a way that models human perception (Section 4.4),
- separates this texture from the remaining image (the *structure*),
- estimates texture parameters that are essential to the perceptual characteristics of it (Section 3.5),
- encodes and decodes structure using a conventional codec, as well as texture parameters, and
- reconstructs texture such that it satisfies the parameters (Section 3.4).

A high-level overview of the system design is given in Figure 5.1. Clearly, the efficiency of such a system cannot be evaluated by traditional quality assessment metrics such as the peak signal-to-noise ratio (PSNR) or the SSIM [Wan+04]. For an equitable comparison to conventional methods, a large-scale survey on subjective quality would be necessary. We attempt to provide a lightweight, but still plausible, substitute for such an evaluation in order to keep the efforts on a reasonable scale.

5.1 Structure–texture classification and decomposition

Many approaches to structure–texture, or "cartoon–texture," decompositions of images have been proposed before, like, for example, [Auj+06]. Here, we choose an approach that is consistent with the Gaussian model.

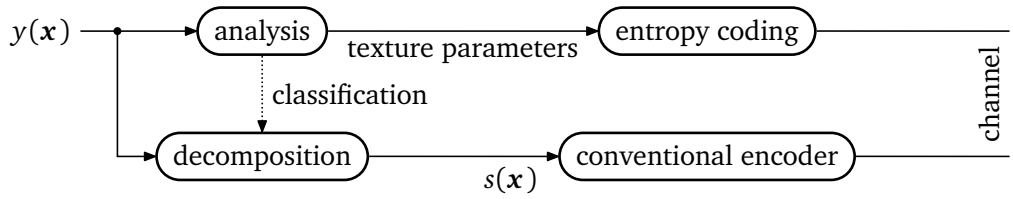
Gaussianity implies that statistical independence and linear independence are equivalent. To be more precise, let us consider two stationary random fields $s(\boldsymbol{x})$ and $t(\boldsymbol{x})$. The fields are linearly independent if and only if

$$\psi_{s,t}(\boldsymbol{x}) = 0 \text{ for all } \boldsymbol{x}, \tag{5.1}$$

where $\psi_{s,t}$ is the cross-covariance function between s and t, or equivalently,

$$\mathrm{E}\Big\{S(e^{\jmath 2\pi f})\, T(e^{\jmath 2\pi f})\Big\} = 0 \text{ for all } \boldsymbol{f}. \tag{5.2}$$

If the fields are additionally Gaussian, then they are also statistically independent. Statistical independence of the fields is important, because it implies that the composite field $y = s + t$



(a) encoder

Figure 5.1 High-level overview of system design.

has the same statistical structure, regardless of a phase difference, such as a shift $\boldsymbol{a}$, of one of the fields:

$$\mathrm{p}(y) = \mathrm{p}(s + t) = \mathrm{p}(s + \boldsymbol{z}^{\boldsymbol{a}} t). \tag{5.3}$$

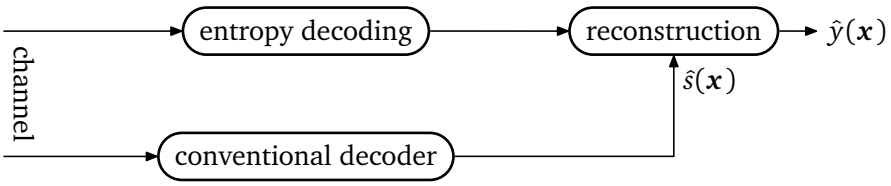
Moreover, we can sample from y by sampling from s and t independently and then simply adding the results. This is not appropriate if s and t are not statistically independent.

If a stationary random field y is Gaussian, we can theoretically use an ideal linear filter $g(\boldsymbol{x})$ to implement the reverse operation: split y into two statistically independent components $s = y * g$ and $t = y - y * g$. To satisfy (5.2), we need to ensure that $G(e^{\jmath 2\pi \boldsymbol{f}})$ is either one or zero for each $\boldsymbol{f}$. With respect to texture, this means we could decompose a texture image into a component of "fine-grained detail," t, and a "smooth component" s, provided that g is a lowpass filter. Using an ideal low-pass filter is of course not possible, as this would imply an infinitely extended impulse response. Approximate solutions need to be applied in practice – the decomposition is useful as long as sufficiently steep filters are applied.

This simple concept of structure–texture decomposition can be utilized to establish a useful feature of the proposed coding system. If texture analysis and synthesis is restricted to high-pass components of the image, such that the lowpass component is encoded conventionally, downscaled versions of the reconstruction can be made (nearly) identical to a downscaled version of the reconstruction using a conventional codec. Such a feature is desirable, as different applications of image compression methods have varying requirements with respect to pixel fidelity. For instance, imagery used in court or for medical purposes may have high requirements, while typical personal photographs may have lower requirements. Setting a maximum cut-off frequency for the lowpass filter allows to freely adjust the maximum amount of detail that is subjected to synthesis.

5.1.1 Filterbank

It should be noted that many possible solutions exist with respect to the shape of the decomposition filter, and using a single cut-off frequency of such a filter for an entire image would not be useful, as images are typically inhomogeneous. We thus need to analyze the image in neighborhoods that are localized in space as well as frequency, and determine a cut-off frequency that is adaptive across image space. Traditionally, this kind of analysis is achieved using filterbanks. Here, we take a simple approach: We let the cut-off frequency of the decomposition filter take values that are consistent with the radial edges of the subbands. Thus, we analyze the image subband-wise, and depending on the outcome of the analysis, we choose the cut-off frequency between two radial subbands of the filterbank.



(b) decoder

This work explicitly does not focus on implementational issues, such as cartesian separability of the filter or efficient implementation in the spatial domain. Instead, we strive to approximate models of the HVS. This implies polar separability rather than cartesian separability; there is no reason to prefer some orientation of image features over another. Another feature of the HVS that has been noted is that simple cell responses roughly cover an octave of radial bandwidth. The logarithmic Gabor filterbank suggested by [Fie87] incorporates both aspects. Further work which builds on these principles includes [Sim+92; SF95; Per95; Kov99a].

Although all of these publications use moderately overcomplete representations which tile the Fourier domain systematically using "hump"-shaped filters (Figure 5.2), there is no indication that the HVS is structured likewise if one takes the higher layers of simple cells into account. In fact, the HVS may realize a much higher level of overcompleteness to adapt to a broader range of different image features already at the lowest layer. Higher levels of overcompleteness have applications in image reconstruction [Gei08]. By using overcomplete representations with more complex basis functions, aspects of the higher cell layers may be modeled. However: These simple Gabor-like filters are the basis of processing in the HVS, and their success with respect to texture segmentation is well documented [Bov91]. Furthermore, we require a moderately compact representation for compression. Therefore, we use a filterbank that is Gabor-like and logarithmic.

The filterbank we use is defined by the radial component

$$R_s(r) = \begin{cases} \cos^2\left(\frac{\pi}{2}\max\left\{-1, \min\left\{0, \log_b 2r\right\}\right\}\right) & \text{if } s = 0, \\ \cos^2\left(\frac{\pi}{2}\min\left\{1, \max\left\{0, s + \log_b 2r\right\}\right\}\right) & \text{if } s = N_\mathrm{s} - 1, \\ \cos^2\left(\frac{\pi}{2}\min\left\{1, \left|s + \log_b 2r\right|\right\}\right) & \text{otherwise} \end{cases} \tag{5.4}$$

and the angular component

$$\begin{aligned} A_o(\theta) = &\frac{4}{5}\cos^6\left(\min\left\{\frac{N_\mathrm{o}}{4}\Delta\left(\theta, \frac{o\pi}{N_\mathrm{o}}\right), \frac{\pi}{2}\right\}\right) \\ &+ \frac{4}{5}\cos^6\left(\min\left\{\frac{N_\mathrm{o}}{4}\Delta\left(\theta - \pi, \frac{o\pi}{N_\mathrm{o}}\right), \frac{\pi}{2}\right\}\right) \end{aligned} \tag{5.5}$$

of the subband filter frequency response

$$H_{o,s}(e^{j2\pi f}) = R_s(|\boldsymbol{f}|) \cdot A_o(\arg \boldsymbol{f}). \tag{5.6}$$

Here, N_s and N_o indicate the number of scales and orientations, respectively, and $\Delta(\theta_1, \theta_2) \in [0, \pi)$ is the smaller of the two angles between the phase values θ_1 and θ_2. b corresponds to the radial bandwidth of each subband. Here, we use the value 2 (octave band filters), both because it is a biologically plausible choice and because it allows implementing

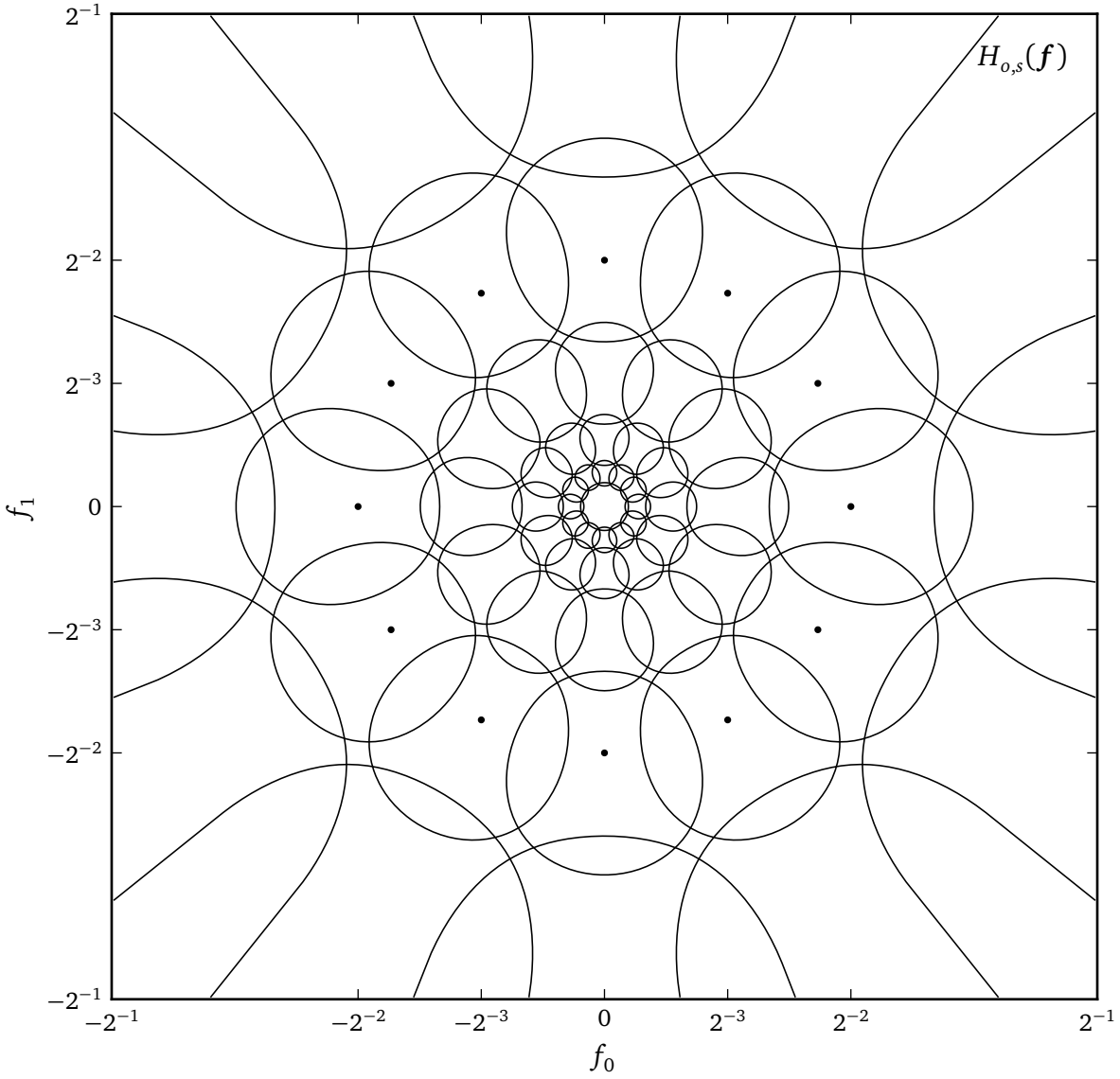


Figure 5.2 Contour lines and maxima of the filterbank defined by (5.4) through (5.6) with 6 scales and 6 orientations. Scales are numbered from highpass ($s = 0$) to lowpass ($s = N_s - 1$). The contour lines at the .3 level are shown for each subband filter. The dots indicate maxima of the subband filters at the first ($s = 1$) scale.

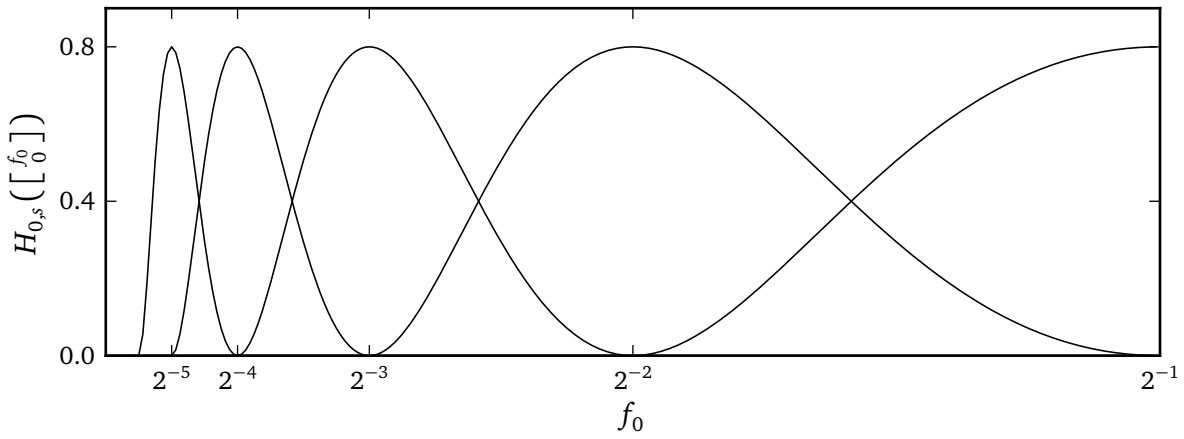


Figure 5.3 Radial cross section through all subband frequency responses of one orientation of the filterbank defined by (5.4) through (5.6) with 6 scales. The cross section shows scales $s = 0$ to $s = 4$ from right to left. The lowpass filter is not shown.

the subband filters more efficiently using decimation. The maximum radial frequency used by a bandpass filter on scale s is 2^{-s} (Figure 5.3), so with each scale ≥ 2, the computational complexity of each filtering operation on that scale can be roughly reduced by a factor of 4 compared to the previous scale.

While some designs require the squared subband filter responses to sum up to a constant, for example the *steerable pyramid* [Sim+92], this particular choice satisfies a similar constraint without the squares:

$$\sum_{o,s} H_{o,s}(e^{\jmath 2\pi f}) = 1. \tag{5.7}$$

In fact, the bandpass filters defined here are equivalent to the subband filters of the steerable pyramid squared. A cross section of one of the impulse responses is plotted in Figure 4.3a.

5.1.2 Space–frequency partitioning

We need to apply the decomposition in a way that is spatially local. One of the conceptually simplest approaches is a block partitioning of the image, a technique that, also, has proven useful for many image compression applications.

For each block (of block edge length d_b) at block position (i, j), we define a lowpass filter

$$G_{i,j}(e^{\jmath 2\pi f}) = 1 - \sum_{s=0}^{p(i,j)-1} \sum_{o} H_{o,s}(e^{\jmath 2\pi f}) = \sum_{s=p(i,j)}^{N_\mathrm{s}-1} R_s(|\boldsymbol{f}|). \tag{5.8}$$

For simplicity, we require the lowpass filter to be constant along the angular dimension, and the cutoff slope to be identical to one of the subband slopes. Thus, (5.8) is fully specified by an integer $p(i, j)$ for each block. This scheme gives rise to a *space–frequency partitioning* of the image.

To determine $p(i, j)$, we use the p-test from Section 4.4. In order to handle the highpass component as Gaussian texture, it must be established that all subbands on scales $s < p(i, j)$

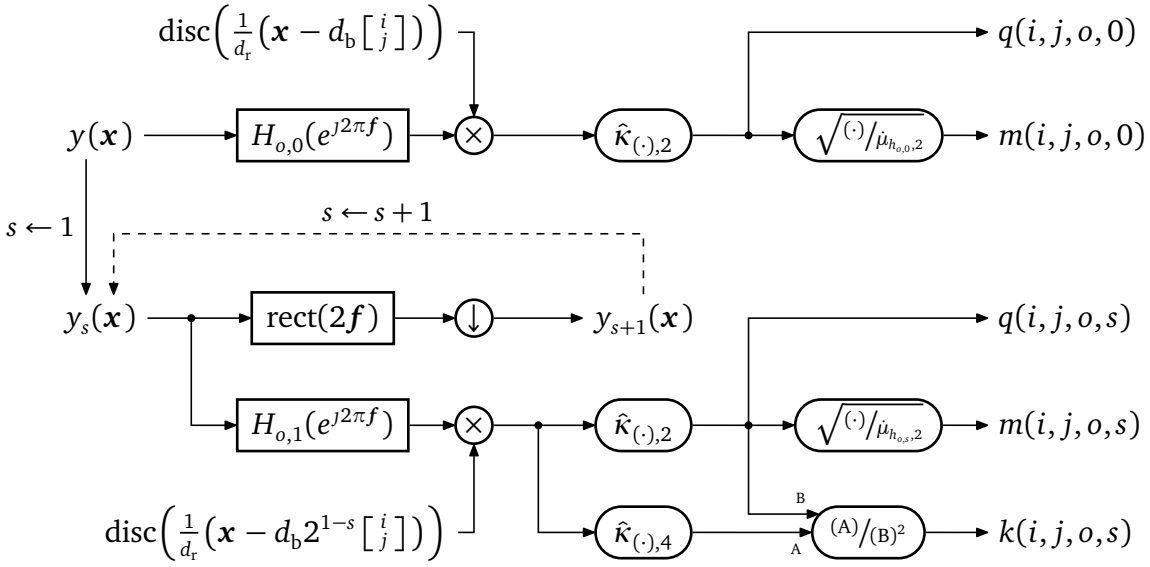


Figure 5.4 Analysis of the input image in the encoder. The processing begins with $y_1(\boldsymbol{x}) = y(\boldsymbol{x})$; the same processing is carried out iteratively on the decimated output $y_{s+1}(\boldsymbol{x})$. The lower branch of the block diagram is parameterized on the subband (o,s) and the block position (i,j); i.e., it is processed for each bandpass subband and block, sharing common inputs where possible.

are likely to be samples from a Gaussian random field (if they contain any significant energy). A special handling of object boundaries, region contours, etc., including appropriate models, is not necessary. It suffices to distinguish Gaussian random fields from all other types of signals, including transients (i.e., salient features).

To facilitate the selection of $p(i,j)$ using the p-test, local magnitude and kurtosis estimates are needed. These are obtained through the system illustrated in Figure 5.4. Scale 0 is not useful for kurtosis analysis, as the impulse responses do not correspond to luminance profiles that are meaningful for feature detection, and the highpass subbands mostly contain alias. Therefore, we estimate subband magnitude for highpass and bandpass subbands, and kurtosis only for the bandpass subbands. For accuracy and computational efficiency, the filters are implemented using the DFT. The input image $y(\boldsymbol{x})$ is fed into the highpass filters and into the bandpass filters of Scale 1. Standardized kurtosis $k(i,j,o,s)$ as well as average magnitude density $m(i,j,o,s)$ estimates are computed for each orientation o and block (i,j), including a disc-shaped neighborhood of a predefined radius d_r, centered on the block. For estimating the second and fourth-order cumulants, we use k-statistics [SO87, page 391]:

$$\hat{\kappa}_{(\cdot),2} = \frac{\sum^n (\cdot)^2}{n-1}, \tag{5.9}$$

$$\hat{\kappa}_{(\cdot),4} = \frac{n(n+1)\sum^n (\cdot)^4 - 3(n-1)\left[\sum^n (\cdot)^2\right]^2}{(n-1)(n-2)(n-3)}, \tag{5.10}$$

where the summation runs over all n elements that have not been masked by the disc. The bandpass filters on scales $s > 1$ are implemented by decimating the image and recursively feeding the output into the bandpass filters $h_{o,1}(\boldsymbol{x})$.

Using $m(i,j,o,s)$ and $k(i,j,o,s)$, the partitioning of the image, $p(i,j)$, can then be computed using Algorithm 5.1. The essential parameter of the algorithm is the kurtosis threshold t_1.

Algorithm 5.1 Partitioning

Input: $k(i,j,o,s)$, $m(i,j,o,s)$, t_m, t_1, t_2
Output: $p(i,j)$

$s \leftarrow 1$
$\forall (i,j) : v(i,j) \leftarrow \begin{cases} 1 & \text{for } |i| = |j| = 1 \\ 0 & \text{otherwise} \end{cases}$
$\forall (i,j) \in D : p(i,j) \leftarrow 0$
repeat
  $\forall (i,j) \in D : c(i,j) \leftarrow 0$ // classification on scale s. 0: structure, 1: texture
  for all $(i,j) \in D$ **do**
    if $(s = 1 \vee p(i,j) = s) \wedge (\forall o : m(i,j,o,s) \leq t_\mathrm{m} \vee k(i,j,o,s) \leq t_1)$ **then**
      $c(i,j) \leftarrow 1$
    end if
  end for
  $\forall (i,j) \in D : n(i,j) \leftarrow (c * v)(i,j)$ // number of neighbors classified as texture
  for all $(i,j) \in D$ **do**
    if $n(i,j) = 0$ **then**
      $c(i,j) \leftarrow 0$
    else if $(n(i,j) \geq 7) \wedge (s = 1 \vee p(i,j) = s) \wedge (\forall o : m(i,j,o,s) \leq t_\mathrm{m} \vee k(i,j,o,s) \leq t_2)$ **then**
      $c(i,j) \leftarrow 1$
    end if
  end for
  $s \leftarrow s + 1$
  for all $(i,j) \in D \mid c(i,j) = 1$ **do**
    $p(i,j) \leftarrow s$
  end for
until $s = N_\mathrm{s} - 1 \vee \forall (i,j) \in D : c(i,j) = 0$
return $p(i,j)$

$p(i,j)$ is progressively incremented if the standardized kurtosis of the oriented subbands at the corresponding scale and block are all below the threshold. This is combined with a very low threshold on subband magnitude t_m to prevent numerical problems of kurtosis estimation with extremely low-energetic subbands. A higher threshold generally implies a higher probability of incorrectly classifying structure as texture.

The kurtosis estimator, naturally, possesses a certain variance. This is illustrated in Figure 5.5: Under the null hypothesis, the kurtosis estimate generally yields values around zero, but depending on d_r, the threshold could be exceeded even though the null hypothesis is correct. To compensate for this effect, and to obtain partitioning maps that can be more efficiently predicted, we apply a simple heuristic that applies a higher threshold t_2 if enough neighboring blocks are classified as texture on the same scale. This can be interpreted as a – conditional – morphological image operation. Additionally, singular blocks that are classified as texture and surrounded by structure blocks are re-classified as structure.

The algorithm proceeds iteratively from Scale 1 upwards, incrementing $p(i,j)$ when the cor-

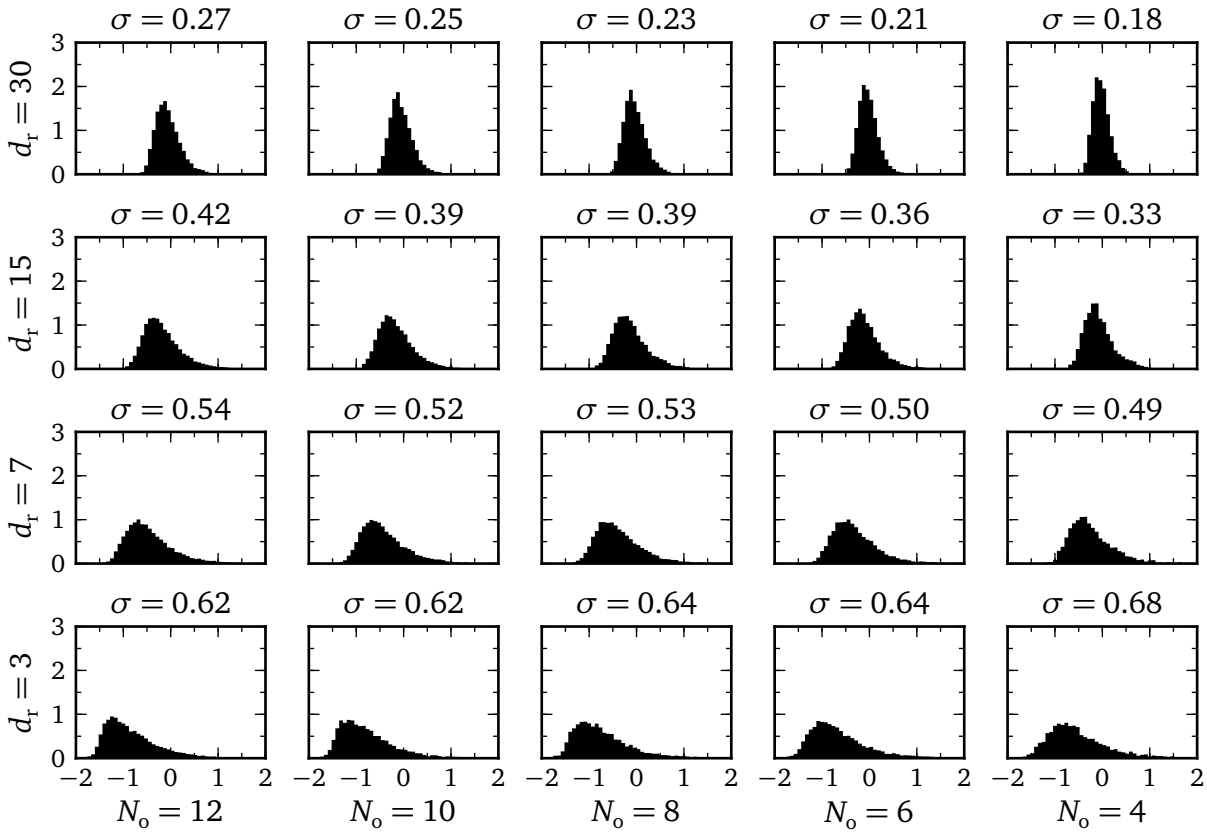


Figure 5.5 Normalized frequencies of the standardized kurtosis estimate $k(i,j,o,s)$ under the null hypothesis. Clearly, the estimator is biased, but consistent: As d_r increases, estimates concentrate around the value 0. Increasing the bandwidth of the filters has the same effect due to (4.14).

responding subbands can be classified as Gaussian. Scale 0 is subjected to synthesis whenever Scale 1 is (implying that $p(i,j)$ cannot attain the value 1). The processing ends if either all subbands on a scale are classified as non-texture or the pre-defined maximum scale $N_\mathrm{s}-1$ is reached. The maximum scale is bounded by the requirements with respect to pixel fidelity, or by image resolution; it does not make sense to use impulse responses $h_{o,s}(\boldsymbol{x})$ which approach the extents of the image.

Finally, the decomposition is carried out according to Figure 5.7. Since not many different lowpass filters need to be applied for the decomposition, the implementation filters the entire image and then selects each block from one of the outputs according to $p(i,j)$. An example of the estimates and the output of the algorithm is visualized in Figure 5.8. In the rightmost column of the figure, the unnormalized magnitude estimates $q(i,j,o,s)$ are illustrated: Here, it is obvious that the removed texture, particularly on the lower border of the image on Scale 1, carries a significant portion of the total energy.

A final example of decomposition is given in Figures 5.9 and 5.10. The removal of texture manifests in excessive blurring of some parts of the image, but the visually relevant features of the image are preserved.

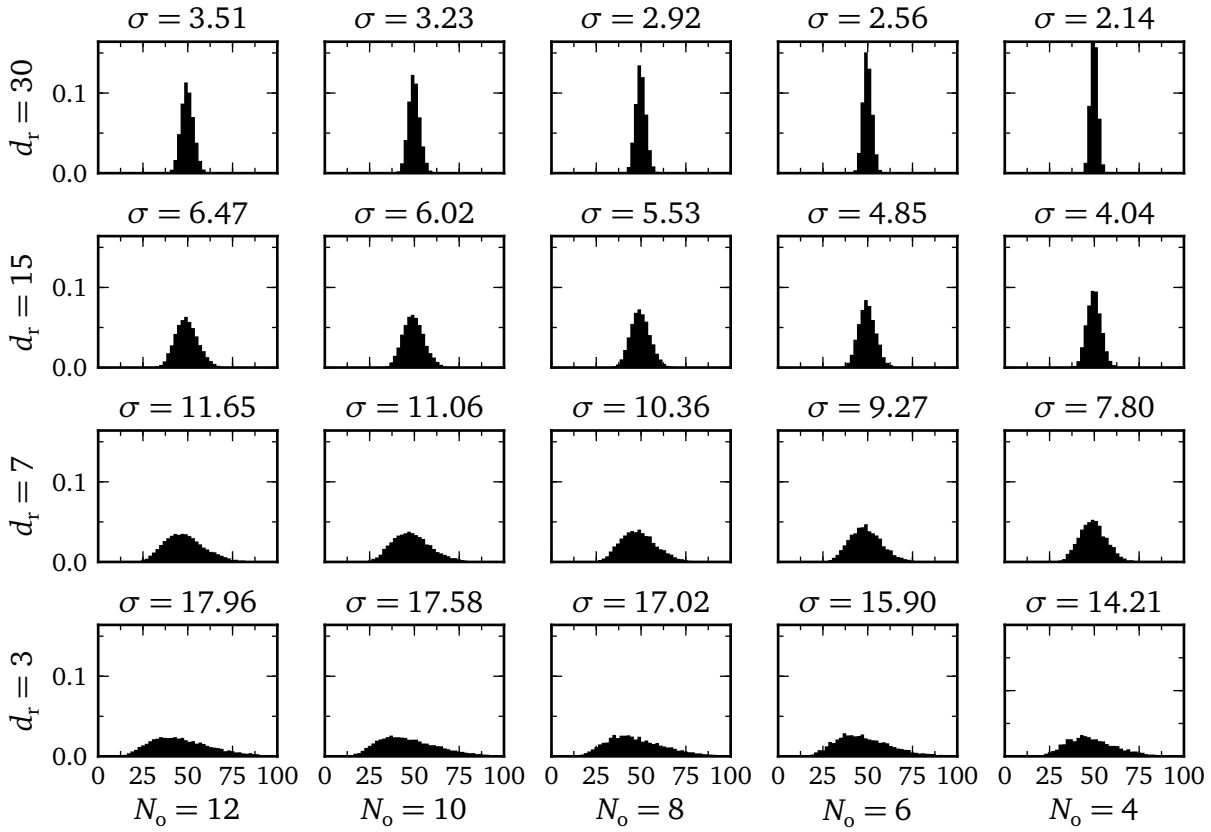


Figure 5.6 Normalized frequencies of magnitude density estimate $m(i, j, o, s)$ under the null hypothesis (the system was fed with Gaussian white noise of standard deviation 50 – with a dynamic range of 8 bit, this would imply slight amounts of clipping).

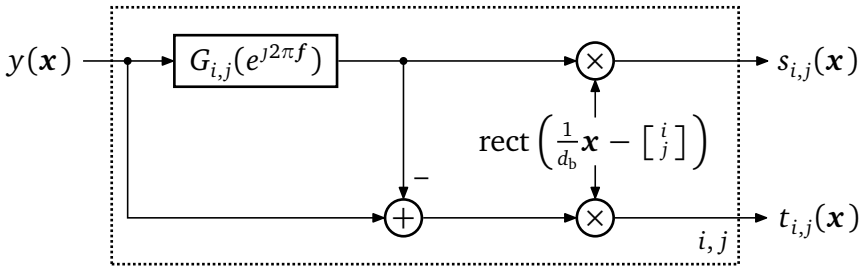
5.1.3 Parameter selection

The system outlined in the previous sections encompasses a number of parameters that need to be selected. Some of the parameters effectively determine the sensitivity of the system with respect to image features. For best results, these parameters should be optimized for physiological data. Since this is not available, we optimize the system for an important special case: the detection of a line feature in noise (Figure 5.11). In the rest of the section, we describe each of the parameters and justify the final choice.

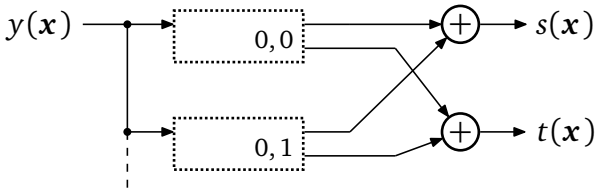
d_b Like in many other block-based image codecs, the optimal block size is content-dependent. A general rule of thumb is that with higher image resolution, larger blocks are helpful to collect and summarize the statistics of large similar areas.

N_s The number of filterbank scales, as mentioned above, needs to be adjusted to image resolution. Generally, Algorithm 5.1 terminates at some point, even if N_s is unrestricted, because it is unlikely that an image is exclusively composed of Gaussian texture. N_s is, therefore, mostly an implementation detail, unless it is used to deliberately restrict texture synthesis to a number of given scales (e.g., for application-dependent reasons).

N_o The number of filterbank orientations determine the shape of the image features that the



(a) Block level



(b) Image level

Figure 5.7 Structure–texture decomposition. (a) Processing carried out for a single block at position (i, j). This yields a structure component $s_{i,j}(\boldsymbol{x})$ and a texture component $t_{i,j}(\boldsymbol{x})$ for each block. (b) To retrieve the decomposition of the complete image, the block decomposition is obtained for each block and aggregated, sharing common operations where possible (such as the lowpass filtering).

system is sensitive to. As the number of orientations increases, the subband filters become narrower in the angular dimension, and, consequently, their impulse responses elongate (Figure 5.12). This leads to a better "signal-to-noise ratio" for features with low curvature. On the other hand, the sensitivity to curved boundaries diminishes. Considering this trade-off, it would be plausible for the HVS to employ angularly narrow and wide filters at the same time. In this application, however, we consider only a single relative angular bandwidth of the subband filters.

For optimization of this parameter, we therefore choose test images consisting of a line in additive noise which is near the perceptual detection limit (Figure 5.11a). The goal is to choose N_o as low as possible (to maintain response to curved features), while still detecting all of the present line feature. We can see from Figure 5.11b that with $N_\mathrm{o} = 12$, it is still possible to detect all blocks adjacent to or encompassing the line while maintaining a true negative rate of above 90%. We therefore choose $N_\mathrm{o} = 12$.

d_r Like N_o, this parameter determines the standard deviation of the magnitude density and standardized kurtosis estimators (Figures 5.5 and 5.6). The choice of d_r entails a trade-off between spatial resolution (for large d_r, features that are not actually close to the block may lead to classification as structure) and uncertainty of the classification (for small d_r, small perturbations may lead to incorrect classification decisions). In the absence of physiological data, we impose two restrictions:

1. The choice of d_r should be optimized in terms of the synthetic test image. Figure 5.11c shows the performance for $N_\mathrm{o} = 12$ and various values of d_r. With low values, such as $d_\mathrm{r} = 10$, there are too many random perturbations, and overall classification accuracy is bad. Too high values lead to many false positives in blocks

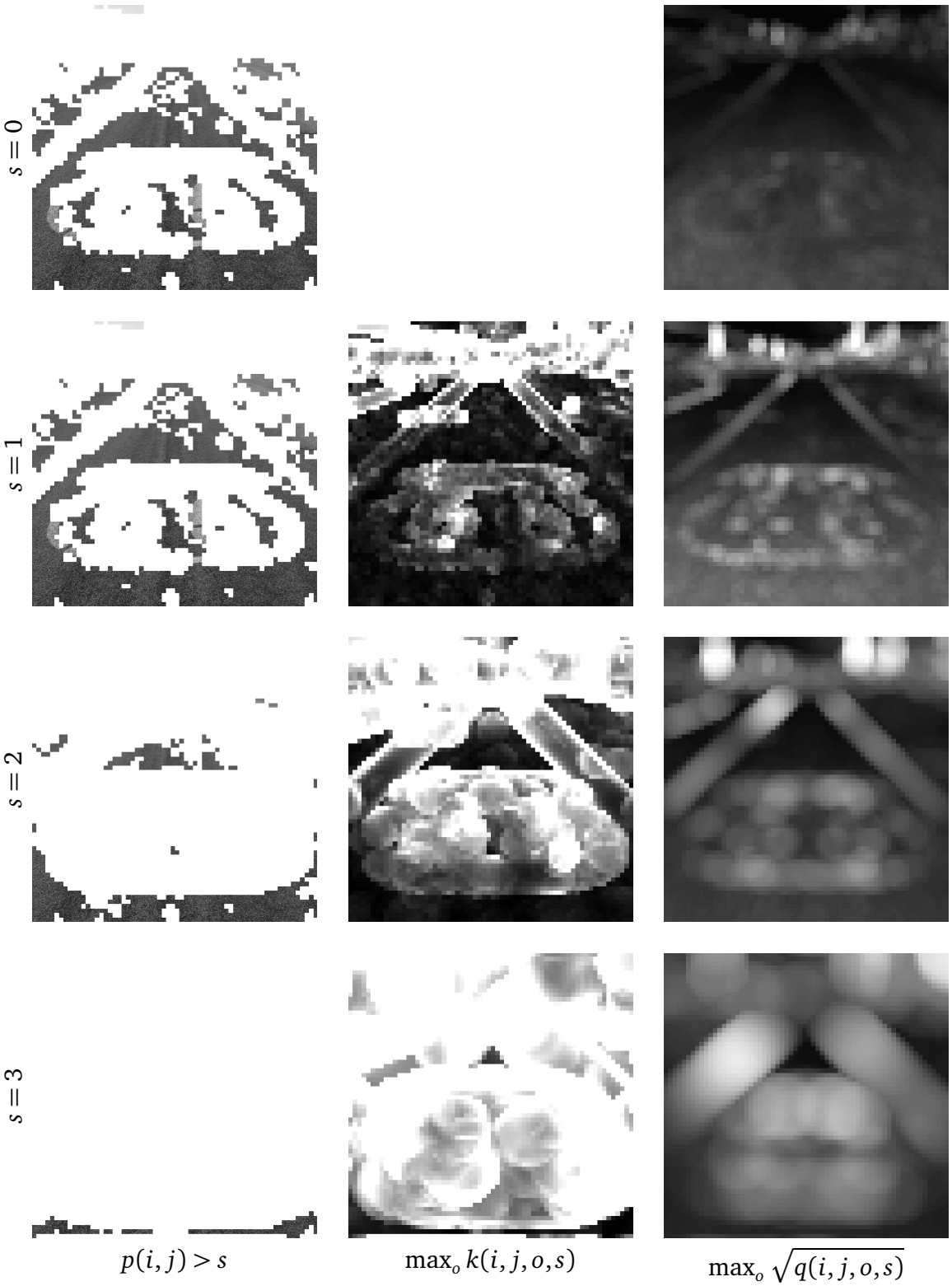
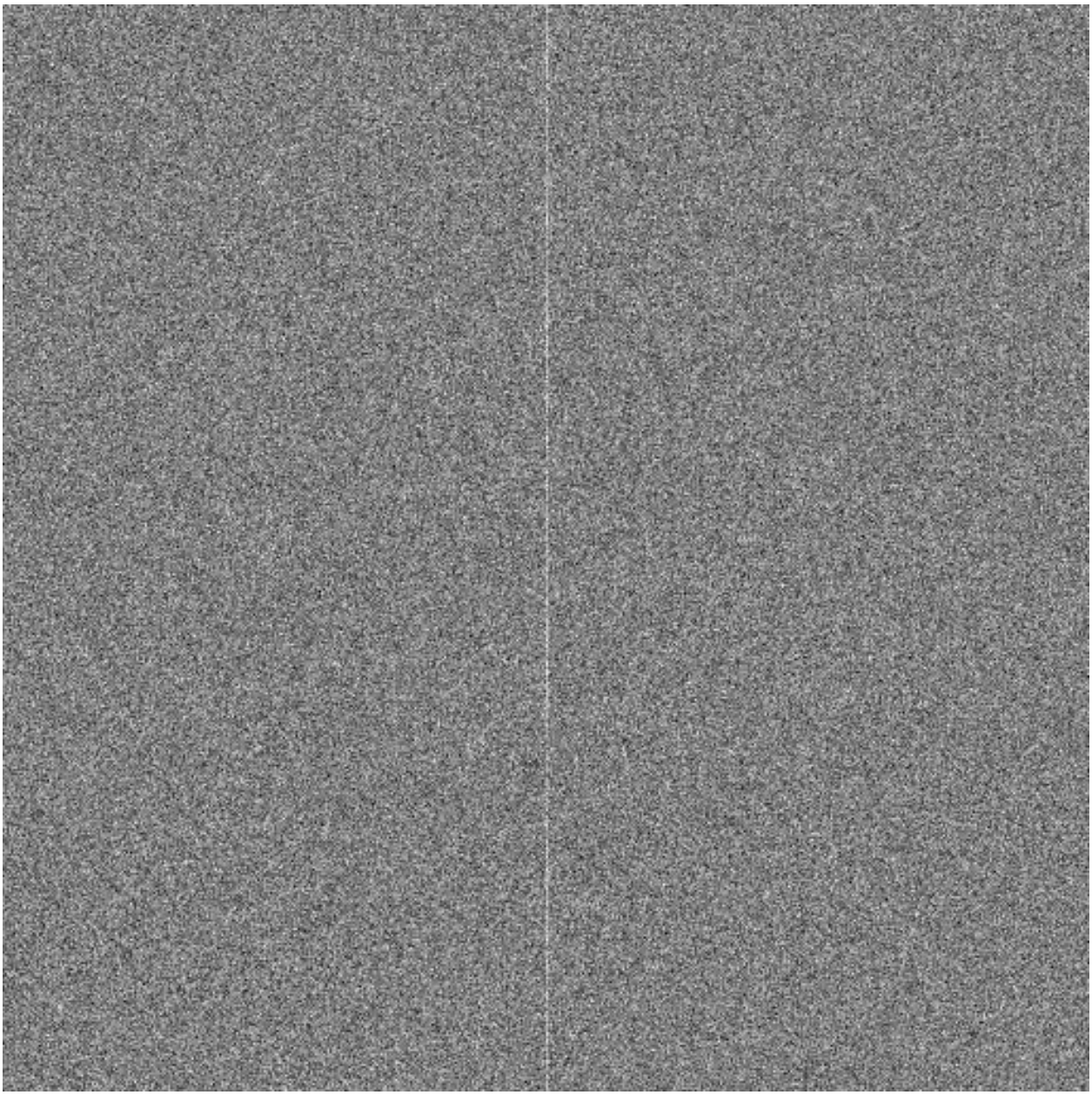


Figure 5.8 Example of space–frequency partitioning for ROUTE 66 image. The right-hand and middle column visualize magnitude and standardized kurtosis estimates, respectively. It is evident that the estimates become more correlated with increasing scale index. The left-hand column displays blocks in the image that are subjected to texture synthesis in the respective scale.

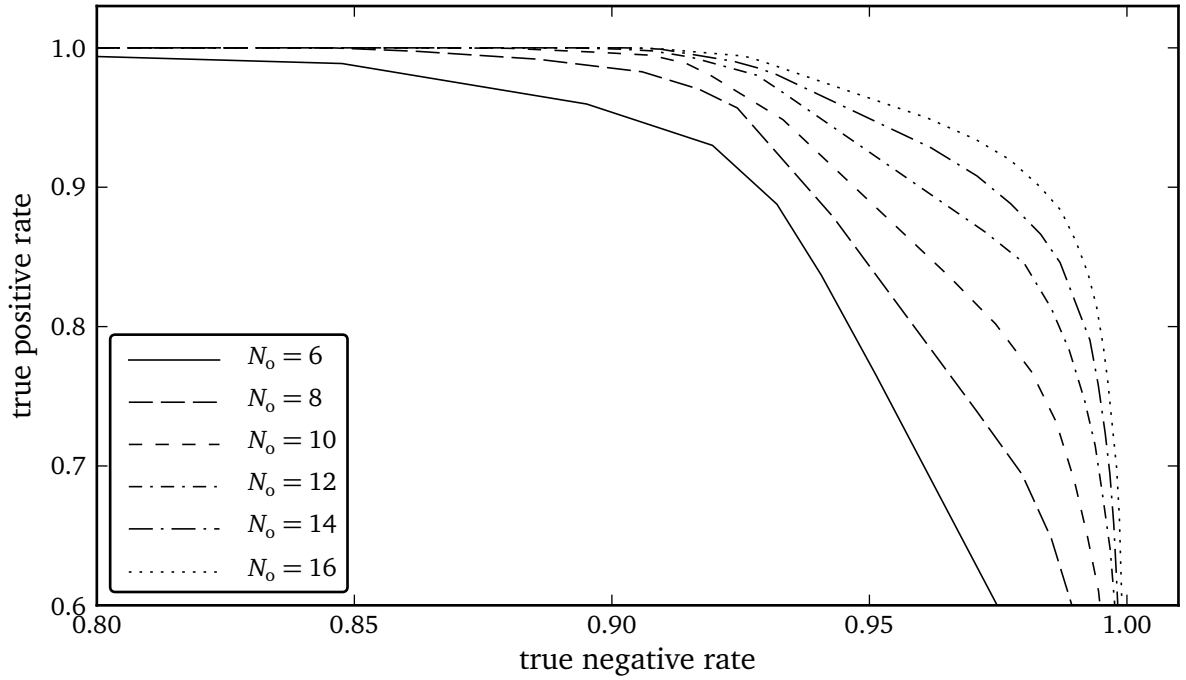
Figure 5.9 ROUTE 66 image.

Figure 5.10 Structure component of ROUTE 66 image.

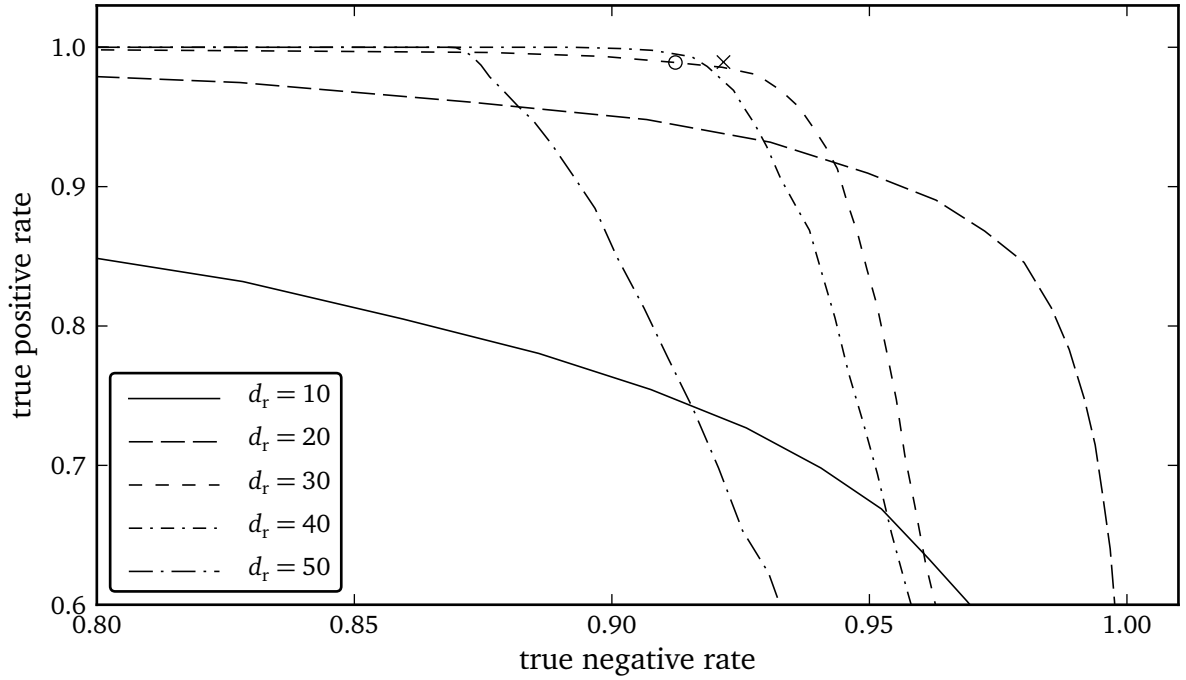


(a) Synthetic test image.

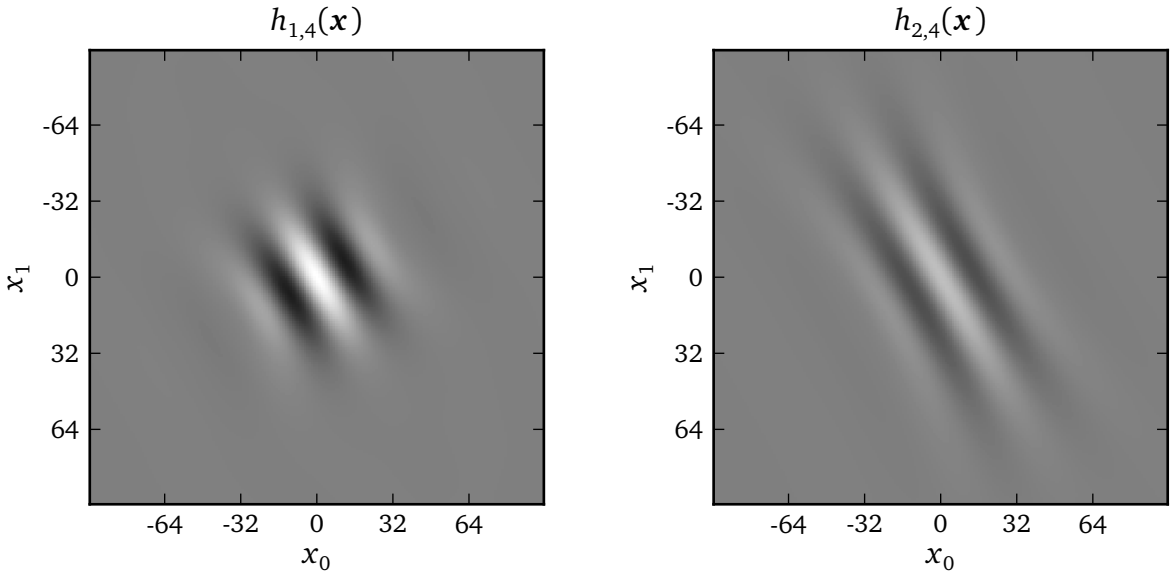
Figure 5.11 Evaluation of partitioning algorithm for synthetic test image. The parameter space for the evaluation is defined by $\{N_o = 2n \mid 6 \leq N_o \leq 16\} \times \{d_r = 10n \mid 10 \leq d_r \leq 50\} \times \{t_1 = 0.1n \mid 0 \leq t_1 \leq 3\} \times \{t_2 = 0.1n \mid 0 \leq t_2 \leq 6\}$. Hence, for each choice of parameter vector (N_o, d_r, t_1, t_2), Algorithm 5.1 was run against 50 test images as shown in (a), consisting of a one-pixel wide line, value 184, against gray background, value 128, subjected to additive white Gaussian noise of standard deviation 32. The blocks adjacent to the line were considered as "positives," all other blocks as "negatives." In (b), the best possible results for various choices of N_o are summarized by taking the convex hull with respect to d_r and t_1. In (c), $N_o = 12$ is fixed and results are plotted for variations of d_r. $d_r \in \{30, 40\}$ generally yield the best results. Morphological smoothing was disabled for both (b) and (c) (t_2 ineffective). It was enabled as a final improvement of the rate of true negatives, without adversely affecting the rate of true positives (indicated by circle and cross in (c)).



(b) Convex hulls with respect to d_r, t_1 for varying N_o (and t_2 ineffective).



(c) Results for $N_o = 12$ fixed. The circle indicates the parameter choice $(12, 30, 1.1, -)$, while the cross indicates the final choice $(12, 30, 1.1, 1.7)$ with morphological smoothing enabled.



(a) 6 orientations, 6 scales

(b) 12 orientations, 6 scales

Figure 5.12 Impulse response of a bandpass subband of two different filterbanks. Gray level reflects value, where black and white correspond to $-0.8 \cdot 10^{-3}$ and $+0.8 \cdot 10^{-3}$, respectively. (a) Subband of a filterbank with 6 orientations and 6 scales. (b) Equally oriented subband of a filterbank with 12 orientations and 6 scales. The increase in number of orientations implies that subband filters are narrower in the angular dimension, which makes the filters more selective with respect to oriented image features.

surrounding, but not directly next to the line feature ($d_r = 50$). A choice of $d_r = 30$ appears to be near optimality.

2. The uncertainty of the magnitude density estimator should – in the worst case – lead to a minor decrease in subjective quality. Figure 5.6 was generated by feeding the estimators with Gaussian white noise of standard deviation 50. A digital image with an 8 bit dynamic range cannot represent Gaussian noise with a standard deviation much larger than that, as clipping would distort the PDF. Since the standard deviation of the estimator increases with the standard deviation of the noise, this represents the worst case. Taking Figure 3.5e into account, we can conclude that a value of $d_r = 30$, which in this case leads to an estimator standard deviation of 3.51, would not significantly alter the subjective similarity score on average.

t_1, t_2 The thresholds ultimately determine the risk of misclassification. They should be chosen as large as possible, provided not too many false negatives occur. The larger the threshold, the more likely will faint image features in noise (as well as texture that is not Gaussian, but near-Gaussian) be subjected to texture synthesis. We choose $t_1 = 1.1$, which is close to the upmost quantiles of the estimator PDF, or even slightly above (Figure 5.5). This choice is also near the convex hull with respect to classification results (Figure 5.11c), and above the "knee" of the curve, where the number of false negatives is still low. Note that it is not absolutely necessary to achieve zero false negatives, as it

is, likewise, difficult for a human observer to distinguish the line from the background everywhere (Figure 5.11a). We further set $t_2 = 1.7$, which allows to increase the rate of true negatives while maintaining the same rate of true positives. In addition, the morphological smoothing removes singular outliers from the partition map $p(i,j)$, making it more amenable to predictive coding.

The above choice of parameters was cross-checked by the author through informal visual inspection on natural test images. Again, to ensure visually optimal results, more precise empirical data about perceptual detectability limits should be collected and utilized. Such a study may also reveal additional feasible points of operation that may be used for coding at very low bitrates.

5.2 Reconstruction

The decomposition algorithm of the preceding section removes blocks of texture from the image. The aggregation of all texture blocks constitute a finite Gaussian random field with rather complex boundary conditions: The boundary conditions of each such block is given by the neighboring blocks, which may either consist entirely of structure, or comprise a structure and a texture part separated by a linear filter. The equation system we need to solve is given by the following equations (at each of the blocks):

$$\forall \boldsymbol{x} \in B_{i,j} : (\hat{y} * g_{i,j})(\boldsymbol{x}) = s(\boldsymbol{x}), \tag{5.11}$$

$$\forall \boldsymbol{x} \in B_{i,j} : (\hat{y} * g'_{i,j})(\boldsymbol{x}) = (w * h_{i,j})(\boldsymbol{x}), \tag{5.12}$$

where $g'_{i,j}(\boldsymbol{x}) = \delta(\boldsymbol{x}) - g_{i,j}(\boldsymbol{x})$ is the complementary high pass filter and $B_{i,j}$ is the set of pixel positions in the block at position (i,j). The equation system is a linear one. Moreover, the solution is unique, as all spectral components of the solution are determined (as, for all (i,j), $G_{i,j}(e^{\jmath 2\pi f}) + G'_{i,j}(e^{\jmath 2\pi f}) = 1$).

We can employ Algorithm 2.1 as described in Section 3.4 to reconstruct the texture component. There are two complications over the conditions in Section 3.4: Firstly, the random field is no longer zero-mean; the mean for each block is a lowpass deterministic field with a block-dependent cutoff frequency. Secondly, the contents of any two blocks are generally not statistically independent; therefore, if applied naively, Algorithm 2.1 needs to deal with all texture in one step, i.e., on a very large precision matrix (for typical image sizes and desktop computers, it can easily exceed memory limits).

The first problem can be solved by applying texture synthesis separately on each scale of the filterbank. If we consider texture synthesis as a linear filtering operation

$$T(e^{\jmath 2\pi f}) = W(e^{\jmath 2\pi f}) H(e^{\jmath 2\pi f}), \tag{5.13}$$

where $H(e^{\jmath 2\pi f})$ is a linear filter and $W(e^{\jmath 2\pi f})$ is the driving IID Gaussian random field, as in (3.5), we can obtain the same result by performing, on each scale, the operation

$$T(e^{\jmath 2\pi f}) = \sum_s W(e^{\jmath 2\pi f}) H_s(e^{\jmath 2\pi f}). \tag{5.14}$$

For this equation to hold exactly, we require that the phase of H_s on each scale aligns with the other scales, i.e., $H_s(e^{\jmath 2\pi f}) = R_s(|\boldsymbol{f}|) H(|\boldsymbol{f}|)$. To satisfy this condition, we must modify

Step 3 of Algorithm 2.1. The back-substitution operation is equivalent to convolving a block-shaped cutout of w with a spatially inhomogeneous filter, whose phase is, in the original algorithm, arbitrarily determined by the Cholesky decomposition. Its output is $\boldsymbol{r}$, a finite Gaussian random field with the desired covariance, which is independent of its neighborhood. We can obtain the same result – a finite Gaussian random field with the desired covariance, *but also* with a desired phase – by filtering w, on each scale, with a linear filter $h_s(\boldsymbol{x})$. The phase of the other component, $\boldsymbol{d}$, is entirely determined by the neighborhood of the field and therefore does not need to be modified. To save computational complexity, we can perform the interpolation on decimated subbands.

The second problem can be solved by observing the conditional independence structure of the texture. Considering Figure 3.2, we can subdivide the set D of pixels further into disjoint sets B_i, such that $\sum_i B_i = D$. The conditional probability density of D with respect to M can be partitioned as

$$\begin{aligned} \mathrm{p}(\boldsymbol{t}_D \mid \boldsymbol{t}_M) &= \mathrm{p}(\boldsymbol{t}_{B_0}, \boldsymbol{t}_{B_1}, \ldots \mid \boldsymbol{t}_M) \\ &= \mathrm{p}(\boldsymbol{t}_{B_0} \mid \boldsymbol{t}_M)\, \mathrm{p}(\boldsymbol{t}_{B_1} \mid \boldsymbol{t}_M, \boldsymbol{t}_{B_0}) \cdots \\ &= \prod_i \mathrm{p}(\boldsymbol{t}_{B_i} \mid \boldsymbol{t}_M, \boldsymbol{t}_{B_0}, \ldots, \boldsymbol{t}_{B_{i-1}}). \end{aligned} \tag{5.15}$$

This suggests that we can sample $\boldsymbol{t}_D$ by successively sampling the sets $\boldsymbol{t}_{B_i}$. In the present reconstruction problem, each of the sets corresponds to one block $B_{i,j}$, where (i, j) is the block index. The computation of the quantity $\boldsymbol{Q}_{AB}\boldsymbol{t}_B$ in Algorithm 2.1 corresponds to a convolution of a region around each block with $q(\boldsymbol{x})$ corresponding to the block. A computationally efficient procedure is to fill the texture blocks with zeros in the image array and synthesize the blocks one by one, following the (modified) Algorithm 2.1. This produces exactly the required matrix product for each of the conditional densities in (5.15).

Algorithm 5.2 summarizes the resulting procedure. Note that – for readability – we state filtering operations in the domain that appears most suitable, and we omit decimation of the subbands in this description of the algorithm. It is understood that the Fourier transforms of fields are denoted by upper case letters ($t(\boldsymbol{x}) \circ\!\!-\!\!\bullet\ T(e^{j2\pi f})$, etc.). The algorithm works by first enforcing (5.12) for all blocks via Algorithm 2.1, and then (5.11) by direct assignment. Because the subband filtering operations required for the first step yield slightly distorted results due to the block-based texture removal in the encoder, we repeat the procedure, using the output of the first iteration as input to the second, and so forth. Effectively, the procedure represents an iterative solver for the linear equation system given by (5.11) and (5.12).

An example of texture reconstruction is given in Figure 5.13. Here, the state of the reconstruction after the first two iterations of the main loop of the algorithm is shown. Due to the quantitatively small error in the first iteration, the initial solution is already quite good: The difference between the output of the first and the second iteration is visually almost negligible, while further iterations have shown to be imperceptible and tend to produce changes below the quantization step size of the 8-bit image representation. We therefore end the reconstruction process after three iterations.[1] The result of this is depicted in Figure 5.14.

Note that this algorithm requires two redundant representations of the texture spectra: $q_{i,j}(\boldsymbol{x})$ and $h_{i,j}(\boldsymbol{x})$. Formally, these are related by the equation

$$q_{i,j} * h_{i,j} * h_{i,j}^{-} \equiv \delta. \tag{5.16}$$

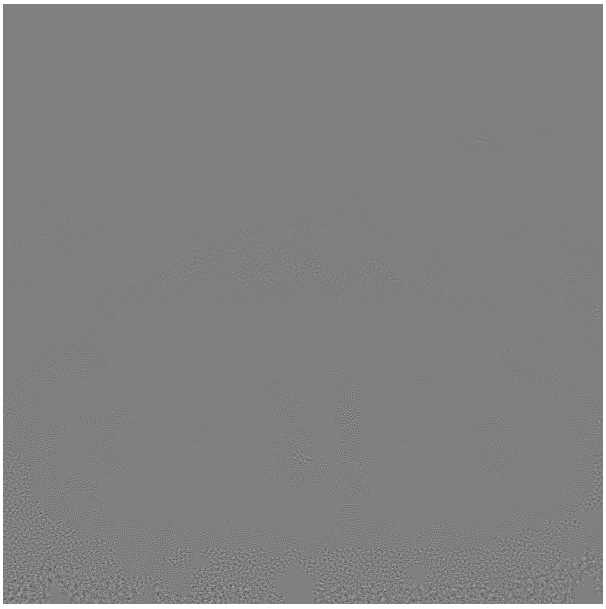
[1] Two iterations may suffice, but we add one as a measure of extra safety.

Algorithm 5.2 Texture reconstruction

Input: $p(i,j), q_{i,j}(\boldsymbol{x}), h_{i,j}(\boldsymbol{x}), s(\boldsymbol{x}), w(\boldsymbol{x})$
Output: $\hat{y}(\boldsymbol{x})$

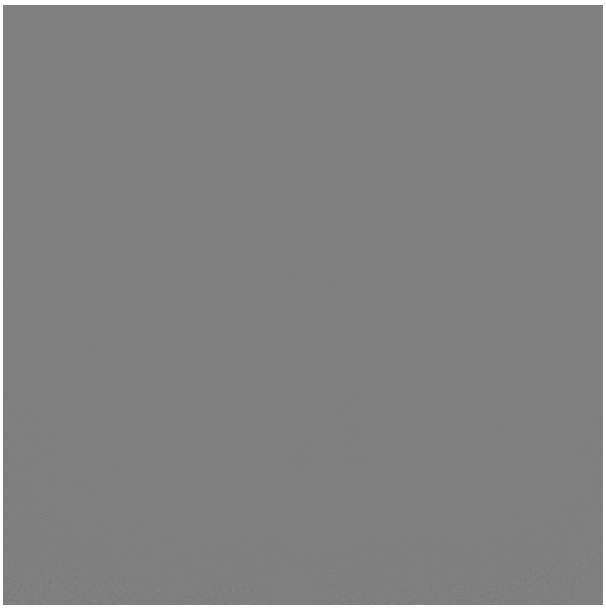
$\hat{Y}(e^{\jmath 2\pi f}) \leftarrow S(e^{\jmath 2\pi f})$ // start with structure component
repeat
  for all $s \in \{N_\mathrm{s}-1, \ldots, 2\}$ **do**
    if $s = 2$ **then**
      $M(e^{\jmath 2\pi f}) \leftarrow R_0(|\boldsymbol{f}|) + R_1(|\boldsymbol{f}|)$
    else
      $M(e^{\jmath 2\pi f}) \leftarrow R_{s-1}(|\boldsymbol{f}|)$
    end if
    $T(e^{\jmath 2\pi f}) \leftarrow \hat{Y}(e^{\jmath 2\pi f}) M(e^{\jmath 2\pi f})$
    for all $\boldsymbol{x} \in B_{i,j} \mid p(i,j) \geq s$ **do** // clear all blocks that contain texture
      $t(\boldsymbol{x}) \leftarrow 0$
    end for
    for all $(i,j) \mid p(i,j) \geq s$ **do** // apply Algorithm 2.1 for each block in succession
      $u(\boldsymbol{x}) \leftarrow (t * q_{i,j})\big(\boldsymbol{x} + d_\mathrm{b} \left[\begin{smallmatrix} i \\ j \end{smallmatrix}\right]\big)$
      $\boldsymbol{Q} \leftarrow \mathrm{mat}_\omega\, q_{i,j}$ // ω is a mapping defined precisely on the block $B_{0,0}$
      $\boldsymbol{u} \leftarrow \mathrm{vec}_\omega\, u$
      solve $\boldsymbol{Q}\,\boldsymbol{d} = -\boldsymbol{u}$ for $\boldsymbol{d}$
      $r(\boldsymbol{x}) \leftarrow (w * m)\big(\boldsymbol{x} + d_\mathrm{b} \left[\begin{smallmatrix} i \\ j \end{smallmatrix}\right]\big)$
      $\boldsymbol{r} \leftarrow (\mathrm{mat}_\omega\, h_{i,j})(\mathrm{vec}_\omega\, r)$
      $t(\boldsymbol{x}) \leftarrow t(\boldsymbol{x}) + \big(\mathrm{fld}_{\omega^{-1}}(\boldsymbol{d} + \boldsymbol{r})\big)\big(\boldsymbol{x} - d_\mathrm{b} \left[\begin{smallmatrix} i \\ j \end{smallmatrix}\right]\big)$
    end for
    $\hat{Y}(e^{\jmath 2\pi f}) \leftarrow \big(1 - M(e^{\jmath 2\pi f})\big)\,\hat{Y}(e^{\jmath 2\pi f}) + T(e^{\jmath 2\pi f})$ // merge subband
  end for
  for all $s \in \{N_\mathrm{s}-1, \ldots, 2, 0\}$ **do** // enforce structure component
    $T(e^{\jmath 2\pi f}) \leftarrow \hat{Y}(e^{\jmath 2\pi f}) \cdot \sum_{i=0}^{s-1} R_i(|\boldsymbol{f}|)$
    for all $\boldsymbol{x} \in B_{i,j} \mid p(i,j) = s$ **do**
      $\hat{y}(\boldsymbol{x}) \leftarrow s(\boldsymbol{x}) + t(\boldsymbol{x})$
    end for
  end for
until convergence
return $\hat{y}(\boldsymbol{x})$

(a) $\hat{y}(\boldsymbol{x})$ after first iteration



(b) difference of (a) to $s(\boldsymbol{x})$

(c) $\hat{y}(\boldsymbol{x})$ after second iteration



(d) difference of (c) to (a)

Figure 5.13 Reconstruction of ROUTE 66 image after each iteration of the main loop of Algorithm 5.2 and differences to the state before each iteration.

Figure 5.14 Reconstruction of ROUTE 66 image after three iterations of Algorithm 5.2.

This implies that, according to Table 3.1, one of them cannot be guaranteed to be finite. Therefore, we must apply some kind of approximative solution for either of the representations. Moreover, for the matrix $\boldsymbol{Q} = \mathrm{mat}_{\omega}\, q_{i,j}$ in Algorithm 5.2 to be positive definite (and the equation system to have a unique solution), Bochner's theorem (Section 2.1.3) demands, for each texture block (i, j),

$$Q_{i,j}(e^{\jmath 2\pi \boldsymbol{f}}) > 0 \qquad \forall \boldsymbol{f}. \tag{5.17}$$

Wherever (5.17) does not hold outside of the subband defined by $M(e^{\jmath 2\pi \boldsymbol{f}})$, we may simply assign a reasonable value to $Q_{i,j}(e^{\jmath 2\pi \boldsymbol{f}})$, since steps 14 and 17 of the algorithm correspond to forward and inverse filtering with $q_{i,j}$: There are no frequency components outside of the subband in the input field, so the filter frequency response is basically "don't care." When (5.17) is not fulfilled *inside the subband*, however, we must apply an approximation (e.g., add a constant to the spectral density function that, in line with Section 3.6, should be chosen to be visually irrelevant).

5.3 Texture parameterization

As we have seen in Section 3.2, there are a number of alternative ways to parameterize a PSD function. To be able to encode the essential qualities of near-Gaussian texture, we need to select one of them as the coding representation. A number of application-dependent requirements arise:

- The shape of the encountered PSD functions is generally not predictable: Texture may be strongly directional (peaked in the angular dimension) or isotropic (angularly flat); in the radial dimension, its shape depends on scale (in Figure 5.14, for example, the texture of the road gets more fine-grained in the distance). In general, it is reasonable to expect fairly smooth spectra – experimentation with randomly generated PSDs (Section 3.6) has shown that, while peaks are plausible (which would indicate the use of AR models as opposed to MA models), extremely high peaks are "unnatural" and unlikely to occur in natural images.

  A complication that has not been considered in Chapter 3 is caused by the decomposition introduced in Section 5.1: A texture PSD produced by the decomposition always contains a lowpass "hole" at the spatial frequencies that constitute the structure component. The coding representation needs to accommodate these characteristics of the PSD; ideally, it should be able to approximate any non-negative function, like the representations listed in Table 3.1.

- The estimation of the parameters should be computationally efficient. Furthermore, as we have seen in Section 3.6, similarity metrics for Gaussian texture need to consider the frequency-dependent sensitivity of the HVS. Ideally, the estimator of the parameters should optimize a weighted spectral distance like (3.40), or a filterbank-based metric like (3.52) or (3.54), considering the spatial frequencies below the cut-off frequency of the decomposition filter as "don't care."

- To exploit the fact that texture often appears in large image regions that do not coincide with block locations, it should be possible to re-use texture information across block

boundaries. This can either be achieved by vector quantization (VQ) [GG92], as in Ballé and Wien [BW09], or by prediction.

- A prerequisite for rate–distortion optimized codecs is that enough feasible points of operation exist. Ideally, the representation should allow a graceful trade-off between quality of reconstruction and bitrate.

In Ballé and Wien [BW09]; Ballé, Jurczyk, and Stojanovic [BJS09]; Feldmann and Ballé [FB11]; Ballé, Stojanovic, and Ohm [BSO11], the inverse filter model $(a(\boldsymbol{x}), \sigma)$ was combined with an on-line VQ approach. The resulting codebook was transmitted by way of scalar quantization. Transmission of codebook indexes must be done losslessly, which implies that their bitrate allocation is not scalable with quality of reconstruction. Despite substantial efforts to reduce the bitrate requirements, codebook indexes remain to take up substantial proportions of the total bitrate allocated for texture [FB11]. Clearly, this problem must intensify for off-line VQ unless effective countermeasures are found.

Regardless of whether a predictive or on-line VQ coding is used, some coefficients must be encoded using scalar quantization. Unfortunately, regardless of which spatial domain representation is used, it may happen that due to quantization, the positivity constraint of the spectrum is violated. The solution applied in the previous work is to choose a quantization step size that is small enough for all practical purposes [FB11]. A trade-off between quality of reconstruction and bitrate requirements can therefore only take place by adjusting the extents of the filter array, or the codebook size in the case of on-line VQ. A problem that applies to on-line VQ in particular is the selection of codebook size and coefficient array extents: optimization of these parameters is computationally expensive, as they are not independent.

Direct estimators for the representations from Table 3.1 are difficult to optimize for any of the similarity metrics that correlate well with human perception (Section 3.6). A linear prediction estimator which is optimized to a weighted Itakura distance is possible by pre- and postfiltering the signal before estimation and after reconstruction [BW05; BW07]. However, by pre-filtering with a potentially steep spectral weighting function such as $W(|\boldsymbol{f}|) = {}^{1}/_{|\boldsymbol{f}|^2}$, the spectrum that is due to be estimated becomes similarly steep. Since the inverse filter must be capable of representing the spectrum with finite support, this approach can easily become numerically unstable if the filter support is not increased. However, filter support must be limited if the filter array representation is used for coding.

Due to the limitations outlined above, a different approach was followed in this work. We use a simple parametric representation of the texture magnitude spectral density,

$$\sqrt{\Psi_{t_{i,j},2}(e^{j2\pi f})} = \sum_{o,s} m(i,j,o,s)\, H_{o,s}(e^{j2\pi f}). \tag{5.18}$$

This representation has several useful properties. Firstly, the representation of partial spectra (i.e., spectra with "holes" or "don't care" regions) is natural, as we can simply drop the information from subbands (o,s) which are irrelevant. Secondly, minimizing mean square quantization error of the model parameters $m(i,j,o,s)$ is equivalent to minimizing the MRMSE metric, which correlates well with human perception (Figure 3.5e). This suggests a predictive transform codec, which makes the coding of VQ indexes unnecessary. The quantization step size Δ_{q} of such a coder then is a parameter that Question 5 from the introduction calls for. Thirdly, no separate estimator is needed, as the magnitude spectral density is already estimated during the image analysis step (Figure 5.4).

For the reconstruction algorithm, we choose the zero-phase filters

$$h_{i,j}(\boldsymbol{x}) \overset{\mathrm{F}}{\circ\!\!-\!\!\bullet} \sum_{o,s} m(i,j,o,s) H_{o,s}(e^{\jmath 2\pi f}). \tag{5.19}$$

The filter support is limited by the filter support of the subband filters: We can therefore implement $h_{i,j}(\boldsymbol{x})$ using finite, fixed-length arrays (if we employ decimated subbands). To determine $q_{i,j}(\boldsymbol{x})$, we interpolate the lowpass "hole" of the magnitude spectral density heuristically by finding the least-squares solution to a set of difference equations forcing the interpolation to be as smooth as possible (thus minimizing the extent of the filter). The equation set is given by:

$$|N_{o,s}|\, m(i,j,o,s) - \sum_{(\tilde{o},\tilde{s})\in N_{o,s}} m(i,j,\tilde{o},\tilde{s}) = 0, \tag{5.20}$$

where $N_{o,s}$ is a set of neighbors for each subband. Generally, we choose $N_{o,s}$ to consist of the four immediate neighbors in scale and orientation:

$$N_{o,s} = \{(o+1 \bmod N_{\mathrm{o}}, s), (o-1 \bmod N_{\mathrm{o}}, s), (o, s+1), (o, s-1)\}, \tag{5.21}$$

except for the highpass and lowpass subbands. For the highpass subbands, the last of the above neighbors is omitted, while for the lowpass subband, the neighbors consist of all of the subbands of the last bandpass scale. Zeros occurring in $m(i,j,o,s)$ are interpolated by adding to the set the equation

$$10\, m(i,j,o,s) = 0. \tag{5.22}$$

The above set of equations is then solved (in a least-squares sense) for all subband magnitudes that are unavailable (i.e., comprise the "hole"), and for those that are zero.

Unfortunately, as predicted by the theory, $q_{i,j}(\boldsymbol{x})$ does not generally have limited support. As in [EW76], the magnitude of the array components wears off as we depart from the origin, so an approximation may be arbitrarily good (the extent of $q_{i,j}(\boldsymbol{x})$ would only be infinite if we would allow the magnitude spectral density to attain zero). However, to find a good tradeoff between computational complexity and precision, the array lengths should be handled dynamically. In our implementation, this problem has not been tackled; we simply use a large filter array yielding results that are precise enough for all images in the test set.

It should be noted that, by moving from filter array representations to this parametric representation of the PSD, we dispose of the ability to represent *arbitrary* spectra simply by increasing filter size, in contrast to the representations investigated in Chapter 3. We can only achieve greater spectral resolution by selecting narrower subband filters. Furthermore, since the subband filters are correlated, repeated estimation and synthesis will lead to degradation. This effect could only be minimized if the filters were orthogonal.[2] Then, however, they could not be Gabor-like. Clearly, the filterbank design requirements for feature detection and compression are in conflict, and we must select a tradeoff, or use two separate filterbanks, which would of course incur another separate estimation step.

[2]The error source that would then still remain, however, would be the variance of the magnitude estimator.

5.4 Coding

To encode the structure component, we use HEVC [HEVCDr12]. Since it is a block-based video codec, the removal of textured blocks will not lead to any significant artifacts if the block boundaries are aligned.

To encode the texture component, we use the CABAC (context adaptive binary arithmetic coding) framework [MSW03]. First, partitioning information $p(i,j)$ is losslessly encoded. Then, spectral magnitude information $m(i,j,o,s)$ is encoded using a predictive transform coding scheme.

5.4.1 Partitioning information

For coding $p(i,j)$, we exploit the fact that texture tends to occur on connected regions of the image. Often, $p(i,j)$ is constant across several blocks. Variations of $p(i,j)$ from one block to one of its neighbors are commonly of magnitude one (± 1). This is due to the fact that the extent of the subband filter impulse responses increases with increasing scale index: transients (i.e., object boundaries, etc.) on fine scales have a spatially narrower impact on the partitioning than transients on coarse scales.

Another peculiarity of the decomposition is that values of $p(i,j) = 1$ cannot exist. Thus, we can safely apply the coding process to its transformation

$$p'(i,j) = \begin{cases} 0 & \text{for } p(i,j) = 0, \\ p(i,j) - 1 & \text{otherwise.} \end{cases} \tag{5.23}$$

Naturally, we need to invert the transformation after decoding.

We encode $p'(i,j)$ in a raster-scan fashion. For each block (i,j), we designate two predictors, $p'(i-1,j)$ and $p'(i,j-1)$, where available. $\tilde{p}(i,j)$, the prediction of $p'(i,j)$, is determined as follows:

- If no predictor is available, then $\tilde{p}(i,j) = 0$.
- If only one predictor is available, then $\tilde{p}(i,j)$ takes the value of the available predictor.
- If two predictors are available and are equal, then $\tilde{p}(i,j)$ takes that value.
- If two predictors are available and are unequal, then $\tilde{p}(i,j)$ takes the value of the integer that is nearest to $1/2\,p'(i-1,j) + 1/2\,p'(i,j-1)$.

The prediction is subtracted from $p'(i,j)$. The final number encoded using a unary code, $u(i,j)$, is a transformation of the difference $\Delta p(i,j) = p'(i,j) - \tilde{p}(i,j)$. This number is determined as

$$u(i,j) = \begin{cases} 2\,\Delta p(i,j) & \text{if } \Delta p(i,j) \geq 0, \\ -2\,\Delta p(i,j) - 1 & \text{if } \Delta p(i,j) < 0. \end{cases} \tag{5.24}$$

Each bin of the unary codeword is encoded using a separate context. Additionally, a context increment is applied, such that disjoint context sets are used in each of the following cases:

- No predictor is available, or two are available and are unequal.
- Only one predictor is available, or two are available and are equal.

All contexts are initialized to uniform probability ($p = 1/2$).

5.4.2 Spectral magnitude information

Because spectral magnitudes are commonly smooth (not only due to the correlation of the subband filters), a transform coding scheme appears appropriate. However, since we do not know the true (prior) statistics of Gaussian texture spectra, we cannot know what transform maximizes sparsity of the transform coefficients. As a heuristic, we choose a harmonic transform.

Due to the symmetry of the magnitude spectrum, $m(i,j,o,s)$ is periodic with respect to orientation o. With respect to scale s, we do not know the boundary conditions, but we may assume that the spectrum is fairly smooth; a sensible choice is reflective boundary conditions.[3] Thus, we choose a discrete cosine transform (DCT, type II) along s and a real discrete Fourier transform (RDFT) along o. We normalize the sequence of both transforms to obtain a single orthonormal two-dimensional transform.

In the orientation dimension, we may assume that the length of the transform N_o is even, because it is natural to include both the vertical and the horizontal orientation in the filterbank. Due to the block-wise variation of the partitioning, the transform length is $p(i,j)$ in the scale dimension. Therefore, a variable-length transform is needed. We denote the transform as $\mathscr{T}_p$, where p is the length of the transform in the scale dimension, and the coefficients of the transform are numbered by the indexes $o' \in \{0,\dots,N_\mathrm{o}-1\}$ and $s' \in \{0,\dots,p-1\}$:

$$\mathscr{T}_p\{m(o,s)\} = \frac{1}{\sqrt{f_{o'}\,c_{s'}}} \sum_{o=0}^{N_\mathrm{o}-1} \sum_{s=0}^{p-1} m(o,s)\, F_{o'}(o)\, C_{s'}(s) \tag{5.25}$$

with the bases and normalization constants

$$F_{o'}(o) = \begin{cases} 1 & \text{for } o' = 0, \\ \cos\left(\frac{\pi}{N_\mathrm{o}}(o'+1)o\right) & \text{for } o' \in \{1,3,\dots,N_\mathrm{o}-1\}, \\ \sin\left(\frac{\pi}{N_\mathrm{o}}o'o\right) & \text{for } o' \in \{2,4,\dots,N_\mathrm{o}-2\}, \end{cases}$$

$$C_{s'}(s) = \cos\left(\frac{\pi}{p}s'(s+1/2)\right),$$

$$f_{o'} = \begin{cases} N_\mathrm{o} & \text{for } o' = 0 \text{ or } o' = N_\mathrm{o}-1, \\ N_\mathrm{o}/2 & \text{otherwise}, \end{cases}$$

$$c_{s'} = \begin{cases} p & \text{for } s' = 0, \\ p/2 & \text{otherwise}. \end{cases}$$

For entropy coding of the quantized transform coefficients, we use a method that imitates the entropy coding mechanism of the transform coefficients in H.264/AVC [AVC], but is not as sophisticated. As with $p(i,j)$, the coding is done in a block-wise fashion, following a raster-scan through the block positions (i,j). The coding process is characterized by "in-loop" predictive quantization, i.e., the prediction of the subband magnitudes of the current block is derived from the *reconstructed* subband magnitudes of the previously coded blocks.

[3] Although, if we assume that the image contains little alias, the highpass subbands should contain very little energy, so a possible improvement would be to enforce zero boundary conditions at $s = 0$.

Analogously to the coding of the partitioning $p(i,j)$, we designate two predictors for the subband magnitudes of a block $m(i,j,o,s)$: $\hat{m}(i-1,j,o,s)$ and $\hat{m}(i,j-1,o,s)$, where the hat indicates that these are the reconstructed values (i.e., the values of the subband magnitudes that are available in the decoder). Due to the variation of $p(i,j)$, it may happen that for a given subband, there exist two predictors, one predictor, or none at all. $\tilde{m}(i,j,o,s)$, the prediction of $m(i,j,o,s)$, is thus derived jointly for the entire block (for all o,s):

- For each subband (o,s) where only one predictor is available, $\tilde{m}(i,j,o,s)$ takes the value of that predictor.
- For each subband (o,s) where both predictors are available, $\tilde{m}(i,j,o,s) = {}^1\!/_2\,\hat{m}(i-1,j,o,s) + {}^1\!/_2\,\hat{m}(i,j-1,o,s)$.

The missing values of $\tilde{m}(i,j,o,s)$ are then derived using the same interpolation as in (5.20), unless no predictors are available for any subband; in that case, $\tilde{m}(i,j,o,s) = 0$ for all subbands.

Once the prediction $\tilde{m}(i,j,o,s)$ is available, the difference to the actual subband magnitude is computed and transformed. The transform coefficients are given by

$$M(i,j,o',s') = \mathscr{T}_{p(i,j)}\{m(i,j,o,s) - \tilde{m}(i,j,o,s)\}. \tag{5.26}$$

The transform coefficients are then subjected to uniform quantization. The absolute quantization levels are determined as

$$L(i,j,o',s') = \max\left\{0, \left\lfloor \frac{|M(i,j,o',s')| - {}^1\!/_2\,\Delta_{\mathrm{q},0}}{\Delta_{\mathrm{q}}} \right\rfloor\right\}, \tag{5.27}$$

where $\Delta_{\mathrm{q},0}$ and Δ_{q} denote the quantization step size for the reconstruction value 0 and for all other reconstruction values, respectively. In this work, we always let $\Delta_{\mathrm{q},0} = \Delta_{\mathrm{q}}$. The absolute quantization levels are encoded using an unary–exponential Golomb code concatenation as in [MSW03]. We (empirically) select 2 as the number of unary code bins, and, for the parameter of the exponential Golomb code, set $k = 0$. Each of the unary code bins is encoded using a separate context. Additionally, we distinguish contexts for different transform sizes and for predicted as opposed to unpredicted coefficients (the latter implying that $\tilde{m}(i,j,o,s) = 0$). Thus, the number of contexts to be reserved for the subband magnitude information is given by the quantity

$$2 \times \sum_{s=2}^{N_{\mathrm{s}}-1} N_{\mathrm{o}}\, s \times 2.$$

All contexts are initialized to uniform probability ($p = {}^1\!/_2$). For each non-zero quantization level, the sign of the corresponding transform coefficient is encoded using the CABAC bypass engine.

The reconstruction of the subband magnitudes, $\hat{m}(i,j,o,s)$, is finally obtained as:

$$\hat{m}(i,j,o,s) = \tilde{m}(i,j,o,s) + \mathscr{T}^{-1}_{p(i,j)}\left\{\operatorname{sgn}\left(M(i,j,o',s')\right)\left({}^1\!/_2\,(\Delta_{\mathrm{q},0} - \Delta_{\mathrm{q}}) + \Delta_{\mathrm{q}} L(i,j,o',s')\right)\right\}. \tag{5.28}$$

These values are used in the decoder for the reconstruction process (Section 5.2), as well as in the decoder and in the encoder for prediction of the following subband magnitudes.

5.5 Experimental results

The method outlined in the preceding sections of the chapter was evaluated against the intra codec of HEVC [HEVCDr12] as a reference. For both reference and structure compression, the default "intra" settings of HM 6.1, the evolving reference implementation of HEVC, were applied. To obtain a reasonable, but non-subjective evaluation of reconstruction quality, our approach here is two-fold: Firstly, we evaluate quality of structure and texture separately at varying bitrates using objective quality metrics, assuming that the partitioning (i.e., classification of blocks and spatial frequency regions into either structure or texture) *on the original image* is correct. Since there is no way of validating whether a human observer would classify a region of an image as either Gaussian texture, or anything else (to do this, the observer would have to be trained on what Gaussian means in terms of visual appearance, which defies the purpose of such an evaluation), we have no choice other than take its output as the truth if we want to work on natural images.

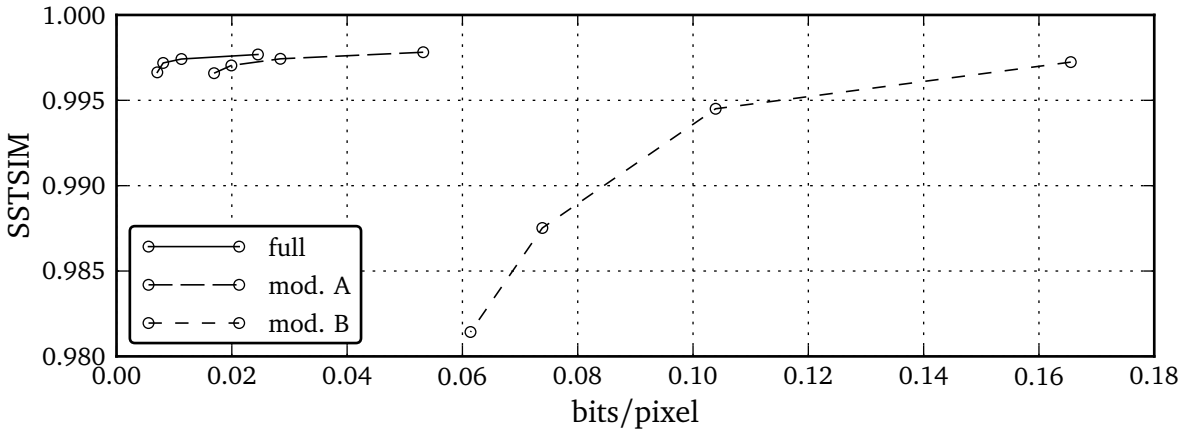
We can, however, observe whether the output of the system *after* synthesis "makes sense" to a human observer and, thus, indirectly evaluate the classification along with the coding system. This is the second aspect of our evaluation: We provide visual results for a number of selected test images to demonstrate the bandwidth of possible results. Appendix A provides further results for the complete Kodak set [Kodak], an established test set for still images.

The evaluation of texture reconstruction is fairly straight-forward: We apply the magnitude estimation of Section 5.1 again to the reconstruction of the reference codec as well as to the reconstruction $\hat{y}(\boldsymbol{x})$. As one of the metrics that provide the best correlation to the subjective scores (c.f. Table 3.3b), we select the SSTSIM to measure texture quality (although the system is optimized for Magnitude RMSE).

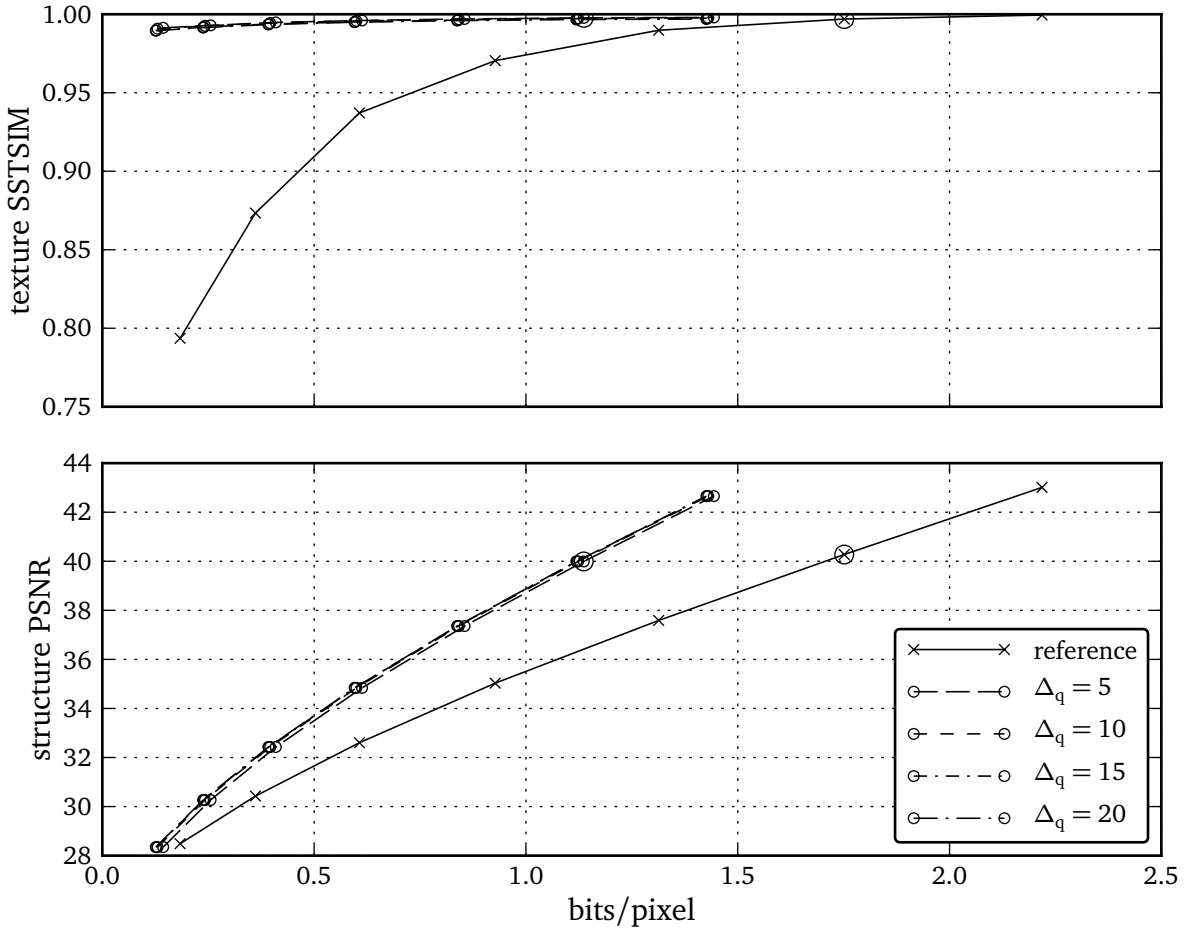
For objective evaluation of structure quality, we rely on the PSNR. Since the structure component is encoded separately, we simply measure the PSNR of its reconstruction against the structure component obtained from the original image. For the reference codec, we apply the identical decomposition (using the same $p(i,j)$) to the reconstruction, and then measure PSNR against the structure component from the original image, as well. From the plots, we can see that our method clearly improves rate requirements of images that contain a basic amount of GMRF texture.

One of the best examples that substantial amounts of bitrate can be saved for images containing Gaussian texture is given in Figure 5.15. This image contains a large amount of road texture, which changes due to perspective distortion. Since we do not attempt to segment the image into parts of homogeneous texture, this distortion is handled gracefully. Predictive coding of the subband magnitudes allows for a smooth transition between visually different texture in the back and in the front of the scene, as long as the chosen block size is small enough to accommodate it. This is in contrast to earlier work [FB11; BSO11], where coding of a single texture model (or, spectrum) requires on the order of 300 bytes. With that method, smooth transitions between blocks are still possible, but the number of different spectra is limited by the bitrate requirements for each model and the necessity to transmit codebook indexes.

With the approach followed in this work, the bitrate requirements are almost negligible with respect to the bitrate required for the structure component. Predictive transform coding of the spectral magnitude information provides substantial gains compared to coding the quantized spectral magnitude values directly. It is particularly efficient for non-directional texture, like



(a) ROUTE 66 texture-only bitrate vs. distortion for varying Δ_q $(5, 10, 15, 20)$ with some coding tools disabled. *Full* refers to the complete system. In *Modification A*, the spectral magnitude prediction is disabled. In *Modification B*, $\mathscr{T}_s$ is additionally replaced by the identity transform.



(b) ROUTE 66 overall bitrate vs. distortion at varying Δ_q and QP $(22, 25, \ldots, 40)$; Subfigures (c) and (d) correspond to the data points marked with large circles.

Figure 5.15 Results for ROUTE 66 image, © Lukas Ballé, 1024×1024 pixels (contd. on next page).

(c) Reference at QP = 25.

(d) Reconstruction at QP = 25, $\Delta_q = 5$.

for the ROUTE 66 image; this is evident from Figure 5.15a. The figure visualizes the bitrate for three cases: the full codec, when prediction is disabled, and when prediction as well as transform are disabled. The combination of both techniques provides for a texture bitrate saving of slightly more than 85% in the case of $\Delta_q = 5$. The percentage of total bitrate used for texture reduces from 13% to 2%. For images with strongly directional texture, i.e., angularly peaked spectra, such as BABOON (Figure 5.16), the gains are not as high, because the "energy compaction" due to the transform is not as strong in that case (Figure 5.16a). Additionally, the texture in that image is not homogeneous: The direction of hairs changes gradually across the blocks, so the prediction is less efficient than with the other image. Still, almost 70% of texture bitrate can be saved.

Because the bitrate requirements for coding texture with the predictive transform approach is so low compared to typical structure bitrates, an optimization of the entropy coding parameters (number of unary code bins and exponential Golomb parameter, context initialization) was not carried out. The block size was always chosen to be $d_b = 16$, and the quantization step size for the images shown here was always chosen as $\Delta_q = 5$, even though it is very likely that bitrate requirements could be further improved by optimizing these parameters.

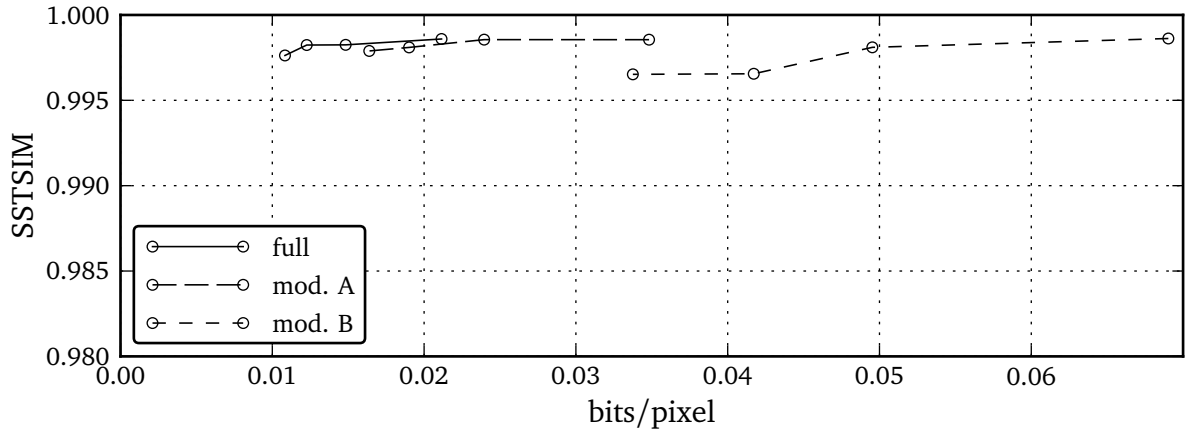
Obviously, bitrate savings vary with the amount of texture contained in the images (Figures 5.15, 5.16, 5.17, Appendix A). Noise introduced by the image formation process is typically near-Gaussian. When conventional codecs are used for high noise levels and high bitrate coding, almost two thirds of the total bitrate may be allocated for texture and noise, as in Figure 5.17. In practice, manual intervention like denoising is required. However, since the proposed coding system does not distinguish between texture and noise, this becomes unnecessary, and the encoding may be directly applied to noisy content. This can lead to almost grotesque bitrate saving figures, particularly for high resolution images – up to 64% of total bitrate saved for images with strong noise. Bitrate savings for non-noisy images are still up to 35% (Figure 5.15).

Of course, it is questionable whether the noise is a feature of the image that we intend to reproduce in the reconstruction. An interesting perspective is, though, that synthesis is always constrained to parts of the image that are low on saliency. Visually salient features that are mixed with noise are still very likely to be processed using the conventional codec. Performing a denoising prior to compression would not leave these image features unchanged; for some applications – for example, when noise levels are unknown – the behavior of our system may be desirable. For a small amount of extra bandwidth, it frees us from the need to define what we consider noise, and what we consider texture in the image.

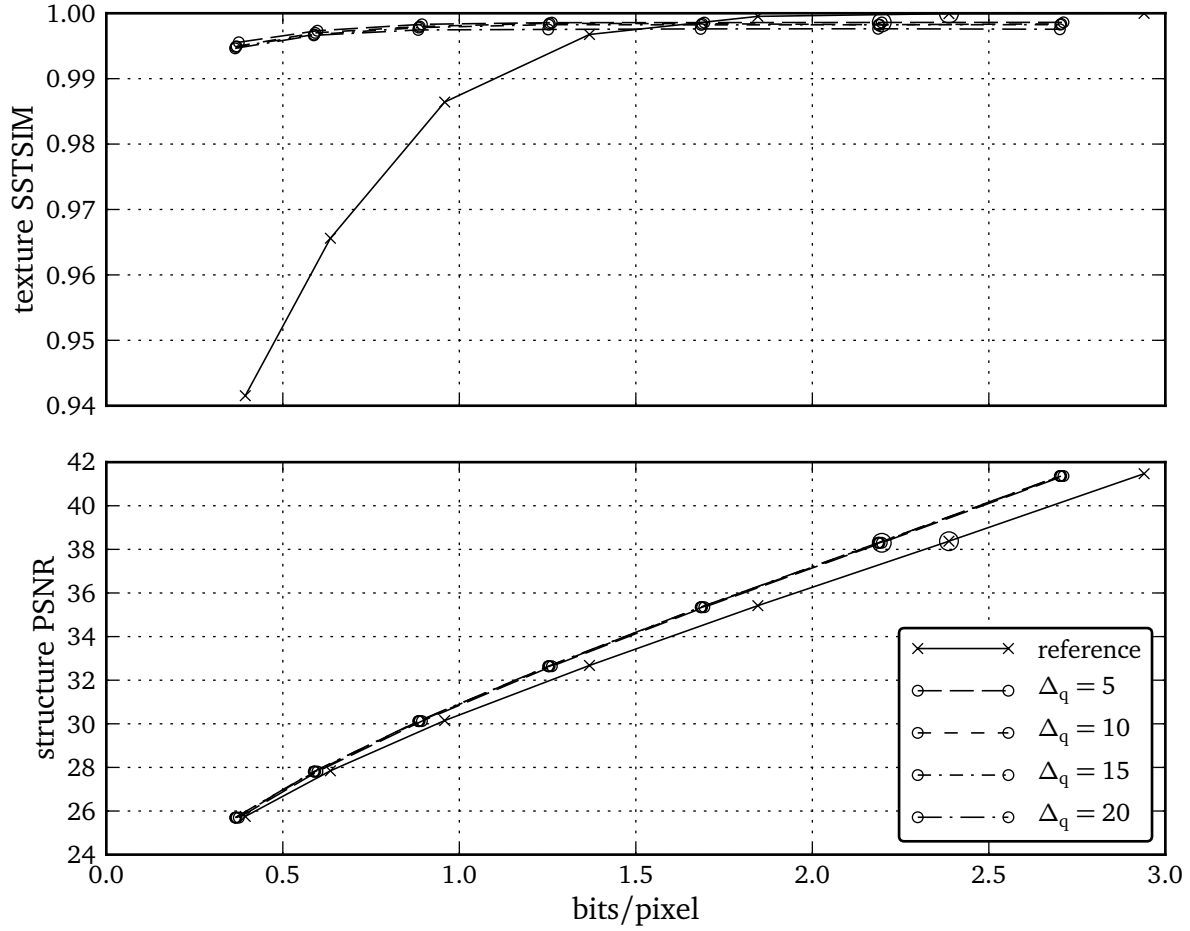
Images that contain neither noise nor texture typically result in a partitioning of $p(i, j) = 0$ for all or almost all blocks. In that case, the bitrate requirement for the texture information is extremely low, such that the results are almost equivalent to the reference results (for example, the first two of Figure A.2).

The careful reader may note that the SSTSIM figures for the suggested codec are mostly far better than the ones for the reference, but sometimes slightly worse (c.f. Figure 5.16). This can be attributed to the overlap of the filterbank subbands in the orientation dimension: When the PSD function of the original texture is strongly directional (like in that figure), subband filters neighboring to the directional component still capture a significant amount of power. This "leakage" is irreversible, but may be addressed by a more sophisticated filterbank design.

As we can see from the visual results, the structure–texture classification works quite reliably for the entire test set, including the Kodak set presented in the appendix. There are

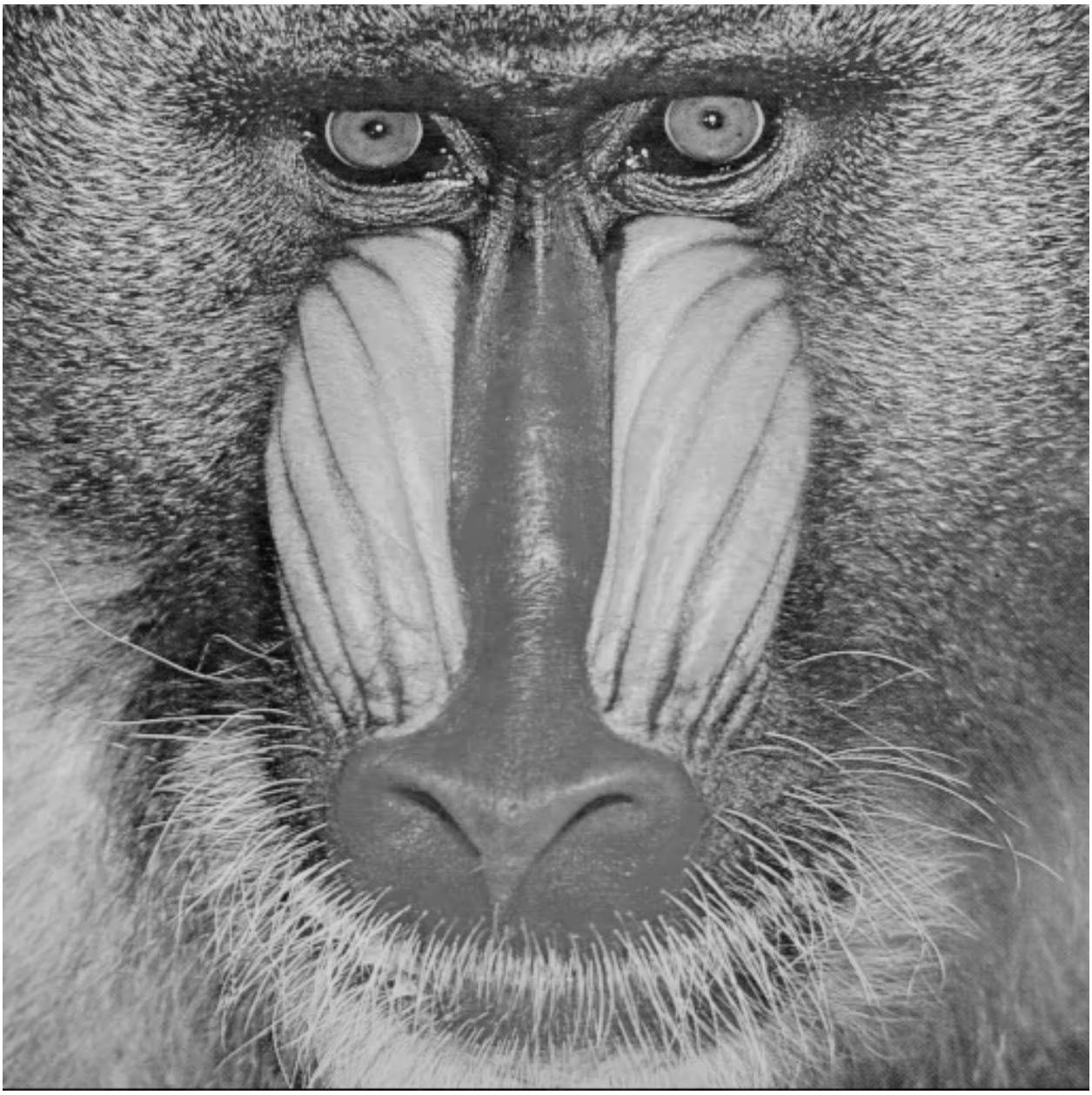


(a) BABOON texture-only bitrate vs. distortion for varying Δ_q $(5, 10, 15, 20)$ with some coding tools disabled. *Full* refers to the complete system. In *Modification A*, the spectral magnitude prediction is disabled. In *Modification B*, $\mathscr{T}_s$ is additionally replaced by the identity transform.

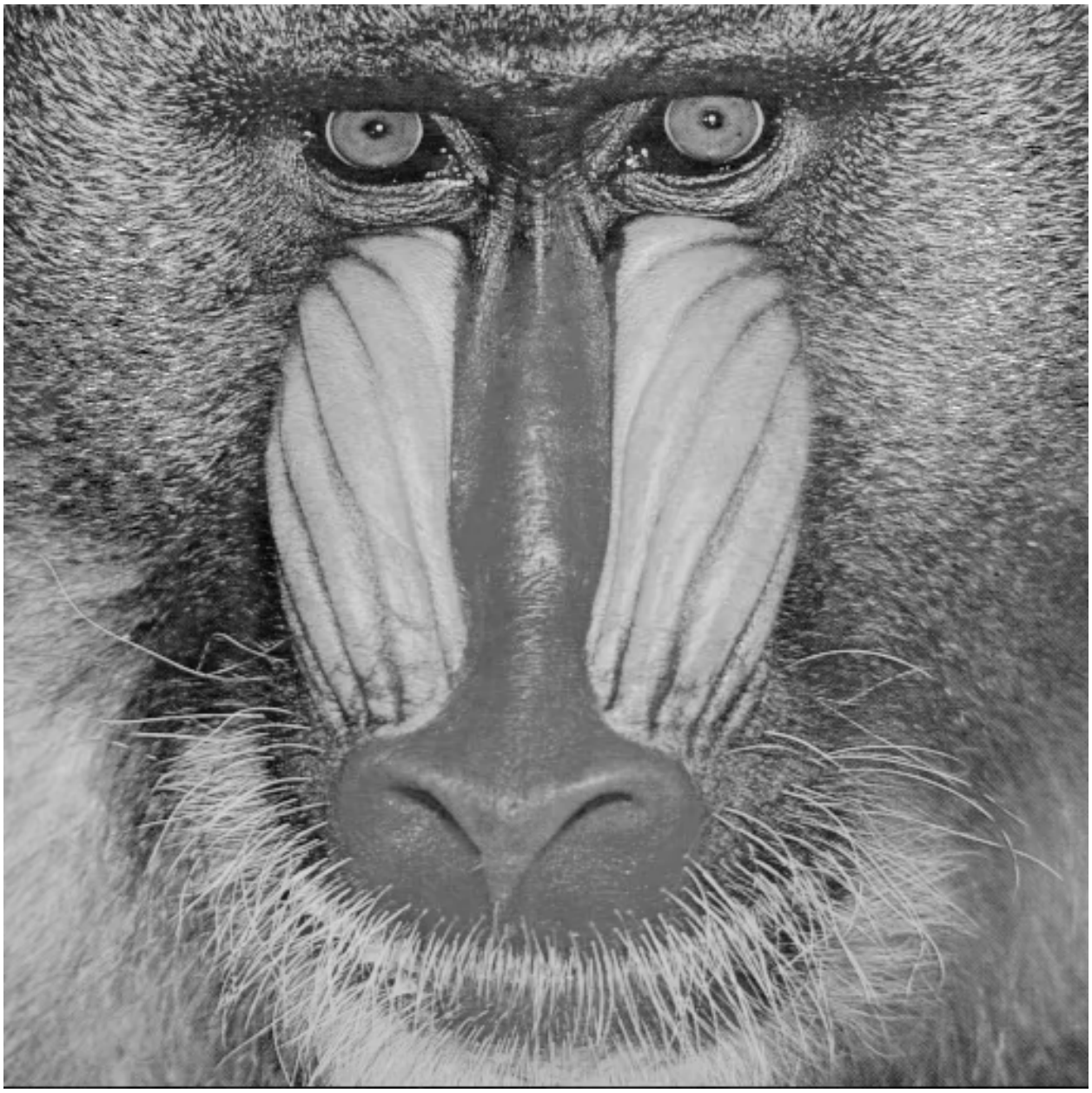


(b) BABOON overall bitrate vs. distortion at varying Δ_q and QP $(22, 25, \ldots, 40)$; Subfigures (c) and (d) correspond to the data points marked with large circles.

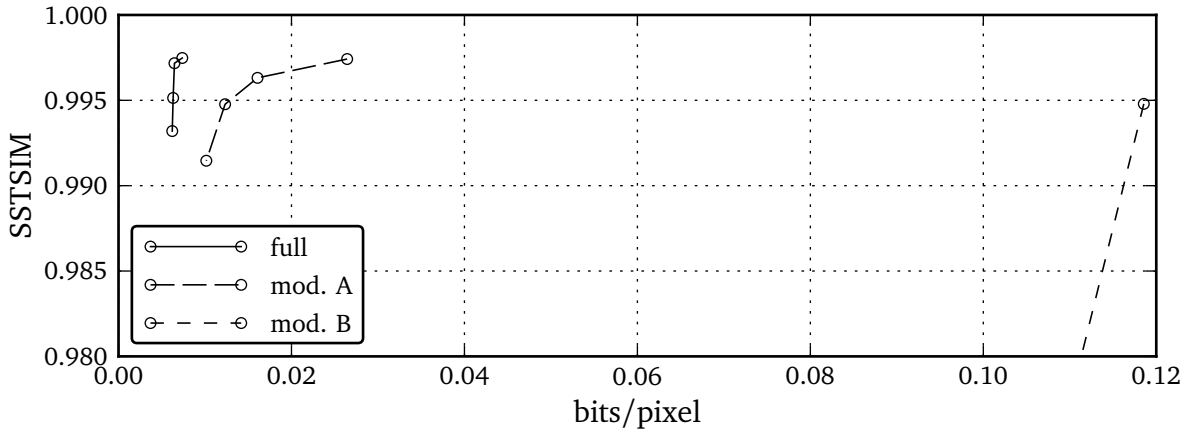
Figure 5.16 Results for BABOON image, 512×512 pixels (continued on next page).



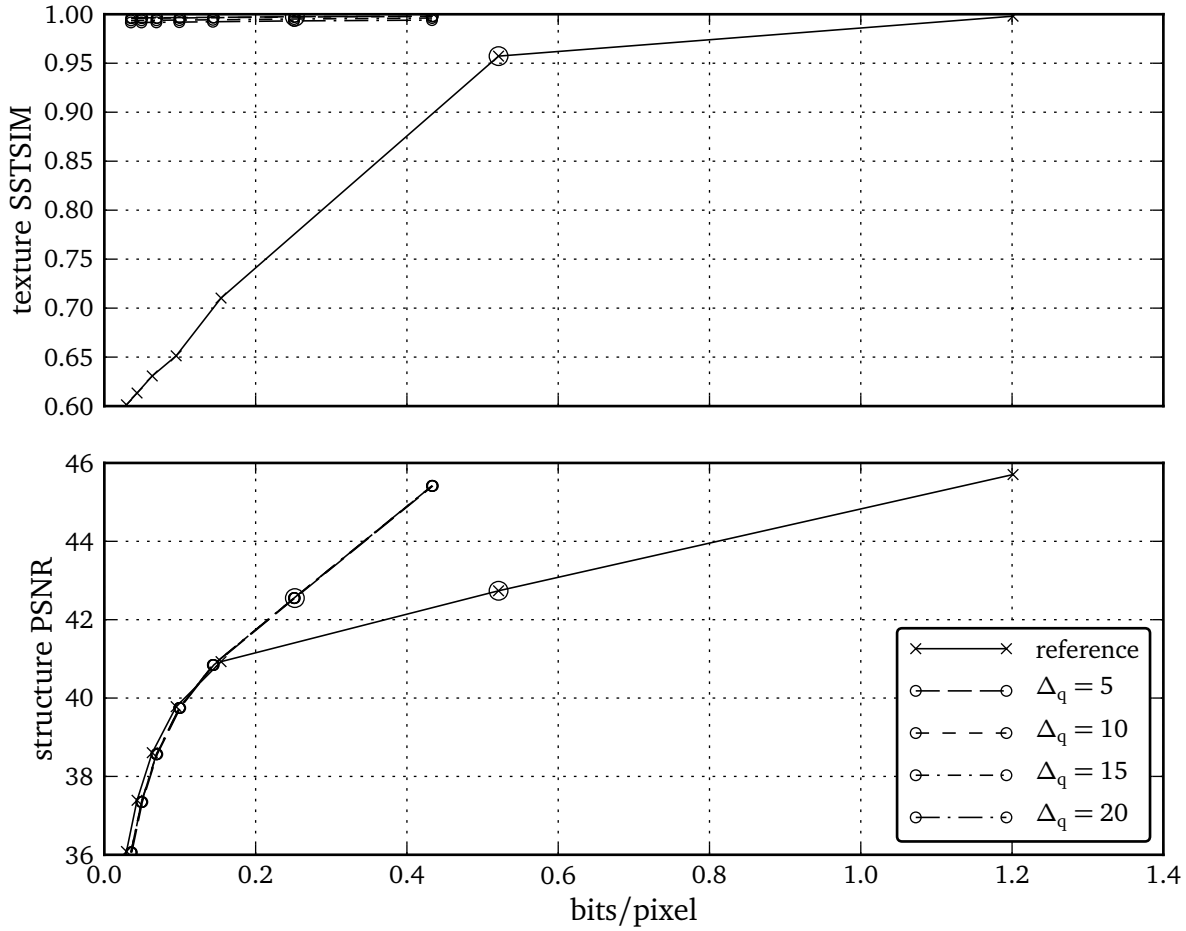
(c) Reference at QP = 25.



(d) Reconstruction at QP = 25, $\Delta_q = 5$.



(a) DEER texture-only bitrate vs. distortion for varying Δ_q $(5, 10, 15, 20)$ with some coding tools disabled. *Full* refers to the complete system. In *Modification A*, the spectral magnitude prediction is disabled. In *Modification B*, $\mathcal{T}_s$ is additionally replaced by the identity transform.



(b) DEER overall bitrate vs. distortion at varying Δ_q and QP $(22, 25, \ldots, 40)$; Subfigures (c) and (d) correspond to the data points marked with large circles.

Figure 5.17 Results for DEER image, [ImCoIn], 2016×1312 pixels.

(c) Reference at QP = 25.

(d) Reconstruction at QP = 25, $\Delta_q = 5$.

no instances of clear misclassifications (false negatives). According to the analysis in Section 5.1.3, this still involves a significant number of false positives, so the amount of texture subjected to synthesis could still be improved by using more empirical data to design the classification step. The parameters of the classification algorithm were identical for all images, as determined in Section 5.1.3.

The parameterization of the Gauss–Markov random field parameters is suitable to carry the perceptual characteristics of the texture – the visual appearance of texture is quite plausible and close to the reference, although in some cases with extremely regular texture (in Figure A.2d in the central part of the hat), we observe some clear artifacts – this may be a drawback of the filterbank-driven representation of the PSD. An inverse filter representation is clearly more suitable for strong peaks (or even singularities) in the spectrum. This may be resolved by modeling the intra subband correlations using a filter array representation such as a low-order AR model, similar to the STSIM metric from Section 3.5.3.

Another effect that we observe is the "Gaussianization" of texture that is subjected to synthesis. Taking a closer look at Figure 5.15, we can see that the statistics of asphalt fragments in the reference and the reconstruction are slightly different – the number of bright "specks" is different. This can be attributed to the fact that the classification algorithm must operate with a threshold, so some amount of textured regions that are not exactly Gaussian, but *near*-Gaussian, are subjected to synthesis, as well. Slight deviations of texture skewness and kurtosis are not recorded by the system and can therefore not be reproduced. Considering that the system in the presented form requires only a fraction of total bitrate for texture, this effect could be minimized by transmitting additional information about skewness and kurtosis of the synthesized texture.

One of the biggest conceptual problems of the presented system is the missing exchange of information between the structure and the texture codecs. In order to obtain visually convincing results, the point of operation of the system is, in its current form, restricted to high bitrates. The reason is that at lower bitrates, the conventional codec begins to "flatten" low-contrast texture due to transform coefficient quantization. If this occurs in blocks that have been classified as structure, for example, because they are close to a salient feature, this leads to unpleasing block-shaped artifacts – the reconstruction of texture is, technically, still better than with the reference, but it is constrained to the blocks that are classified as texture. The effect can be observed, for example, in some parts of the background in Figure 5.17 for the noise. This problem may be solved by aligning the bit allocation of both coders.

The computational complexity of the encoder and decoder is, naturally, higher than the one of the reference codec implementation, HM, as our method represents an "add-on" to it. However, it is not as high as might be expected. Both encoder and decoder, implemented in Python, a high-level interpretative language, take on the order of tens of seconds to run on a typical desktop computer, depending on image extent and amount of synthesized texture. This could be improved by making use of software optimization or dedicated hardware.

6 Summary and conclusion

In this thesis, we take a look at the Gauss–Markov random field model and the special role it takes – not only with respect to information entropy, but also with respect to human vision. In this context, we seek a statistical interpretation of feature detection in the human visual cortex.

We provide a dedicated investigation into the theoretical and practical benefits and limitations of a hybrid coding system designed for this particular model. This includes an investigation of the theoretical possibilities of transferring concepts known from speech and audio coding to random fields. We prove that it is possible to exploit the irrelevance inherent in such texture without employing models that are not backed by physiological evidence or that are based mostly on intuitive concepts like segmentation. This implies that we are able to address the 5 fundamental questions posed in the introduction in a mathematically concise manner.

In Section 5.5, we have seen that the presented system, at a very similar visual quality, is able to provide up to 35% of bitrate savings for natural images compared to a state-of-the-art codec, and up to 64% of bitrate savings if we compare compression of noisy content to the reference without pre-processing. An interesting perspective is that the proposed system of coding texture parameters provides the highest gains when texture magnitude spectral densities are flat – i.e., when the texture is maximum entropy. This is exactly the point where conventional transform coding systems must be, by principle, least efficient.

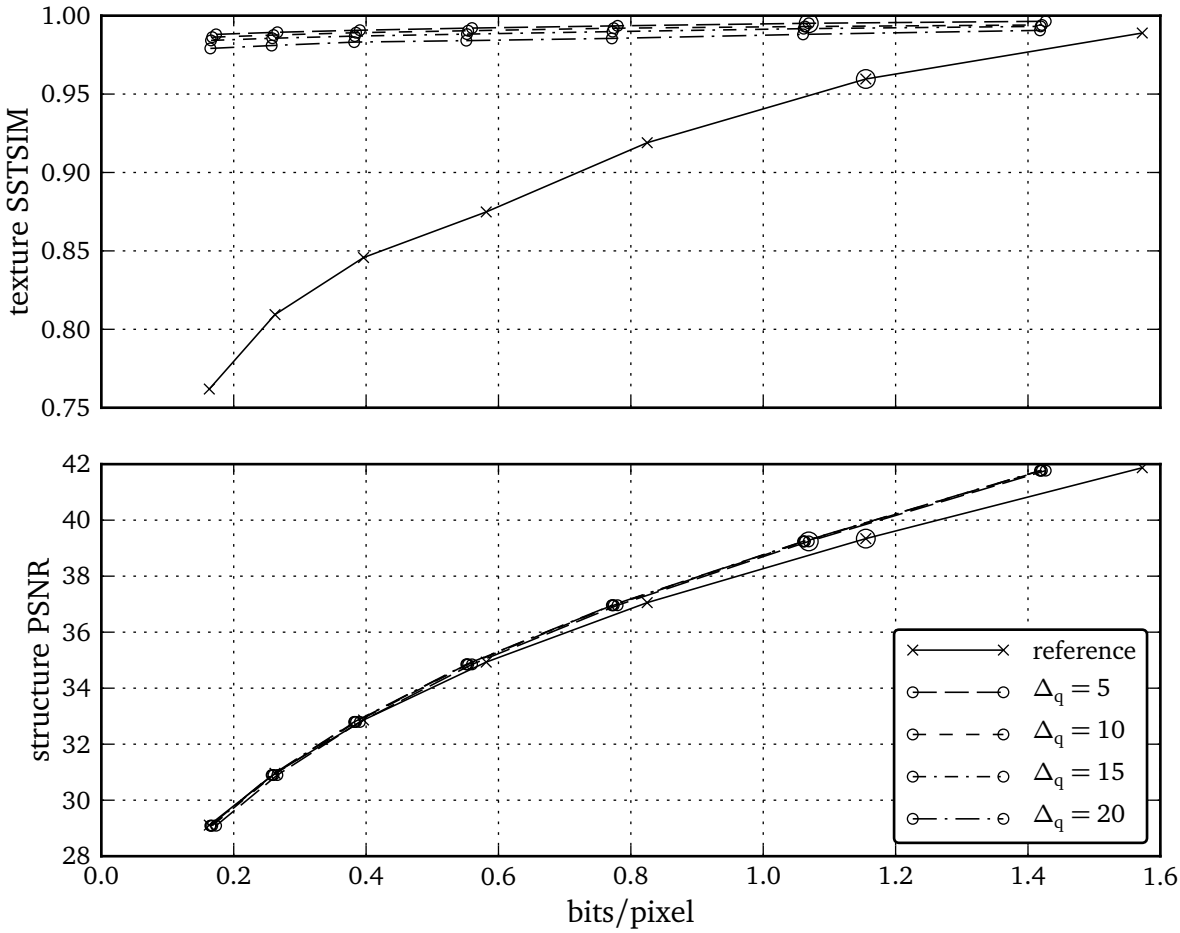
The presented thesis should be taken as a proof of concept and as a theoretical starting point. Several ideas for improvement of the system itself have been proposed in Section 5.5: To implement better rate control strategies, some flow of information between the conventional codec and the texture codec should be established. The filterbank design may be improved by examining more closely the diverging requirements of feature detection and compression of magnitude spectral density functions. Potentially, two different filterbanks may be applied. For extremely peaky spectra (like for some repetitive texture), it may even be useful to consider modeling intra subband correlation using inverse filter representations.

Furthermore, extensions of the theory presented in Chapter 3 to linear (i.e., non-Gaussian) random fields and to color texture can be obtained with small efforts. This may solve the problem of “Gaussianization” of texture (Section 5.5), as well as provide for a greater fraction of image energy subjected to texture synthesis. For linear texture, the higher-order cumulants of the driving IID random field must be modeled. They may be estimated via the relationship in (4.14). Additionally, since the equivalence of independence and linear independence (Section 5.1) breaks in this case, there is a need to model statistical dependency between the structure and texture components. For color texture, a similar problem arises: In the Gaussian case, Fourier phase itself remains to be irrelevant, but in any case, the phase coherence between the color channels is important. It can, however, be expected that such statistical dependencies, as well as the higher-order cumulants for linear texture, can be modeled using a small number of parameters.

A more difficult endeavor is the extension of the presented framework to moving images. It

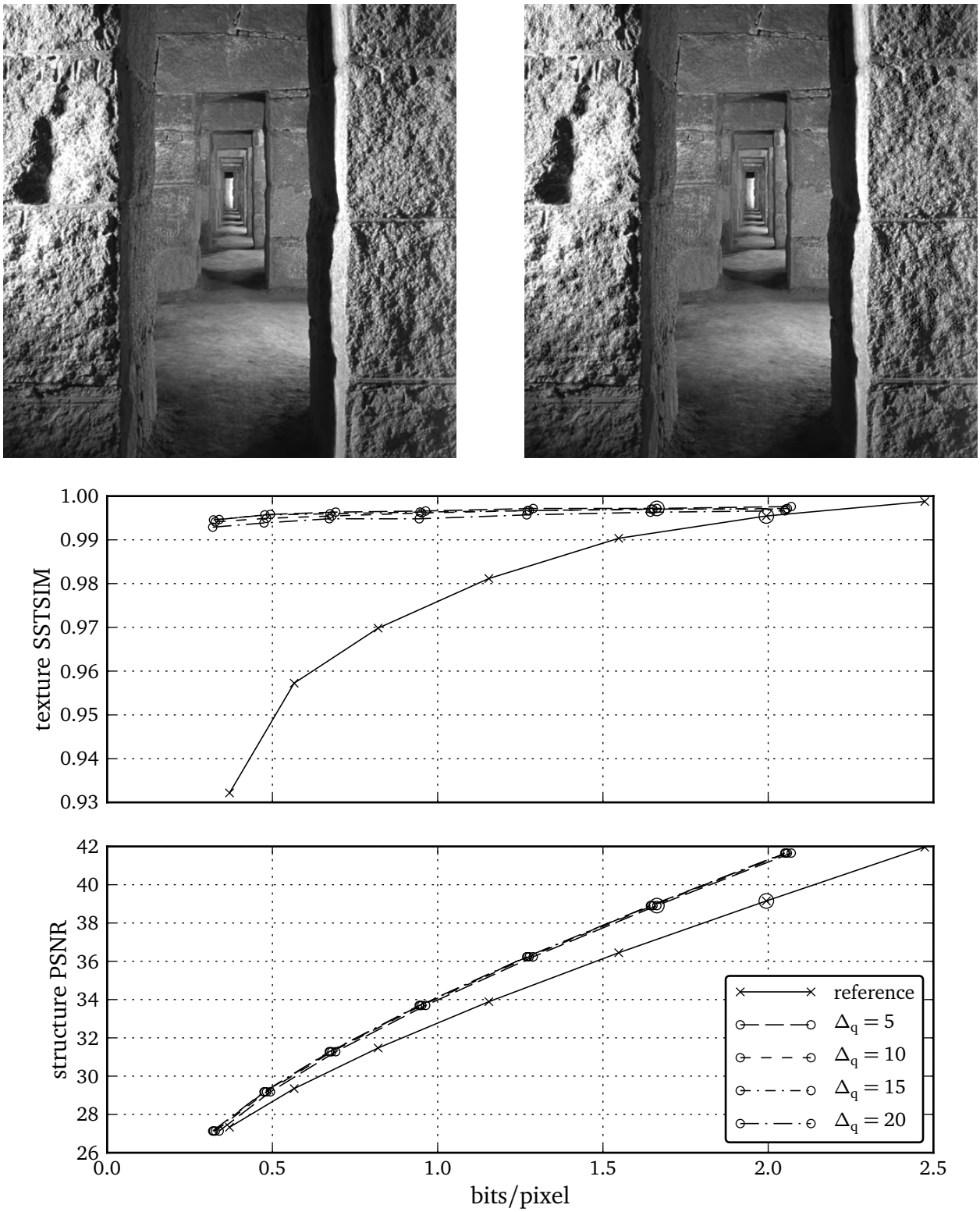
is not useful to naively combine it with the traditional frame-based approach to video coding, as this would fail to provide temporal consistency of texture. An approach more consistent with the methodology applied in this thesis could be conceived: This would require not only to model biological feature detection, but also the characteristics of motion perception – in particular, neurons that respond to temporal stimuli, but also, most likely, eye movement to some extent. An open question that remains is how a unification of the hybrid coding of structure and texture could be achieved.

A Further results

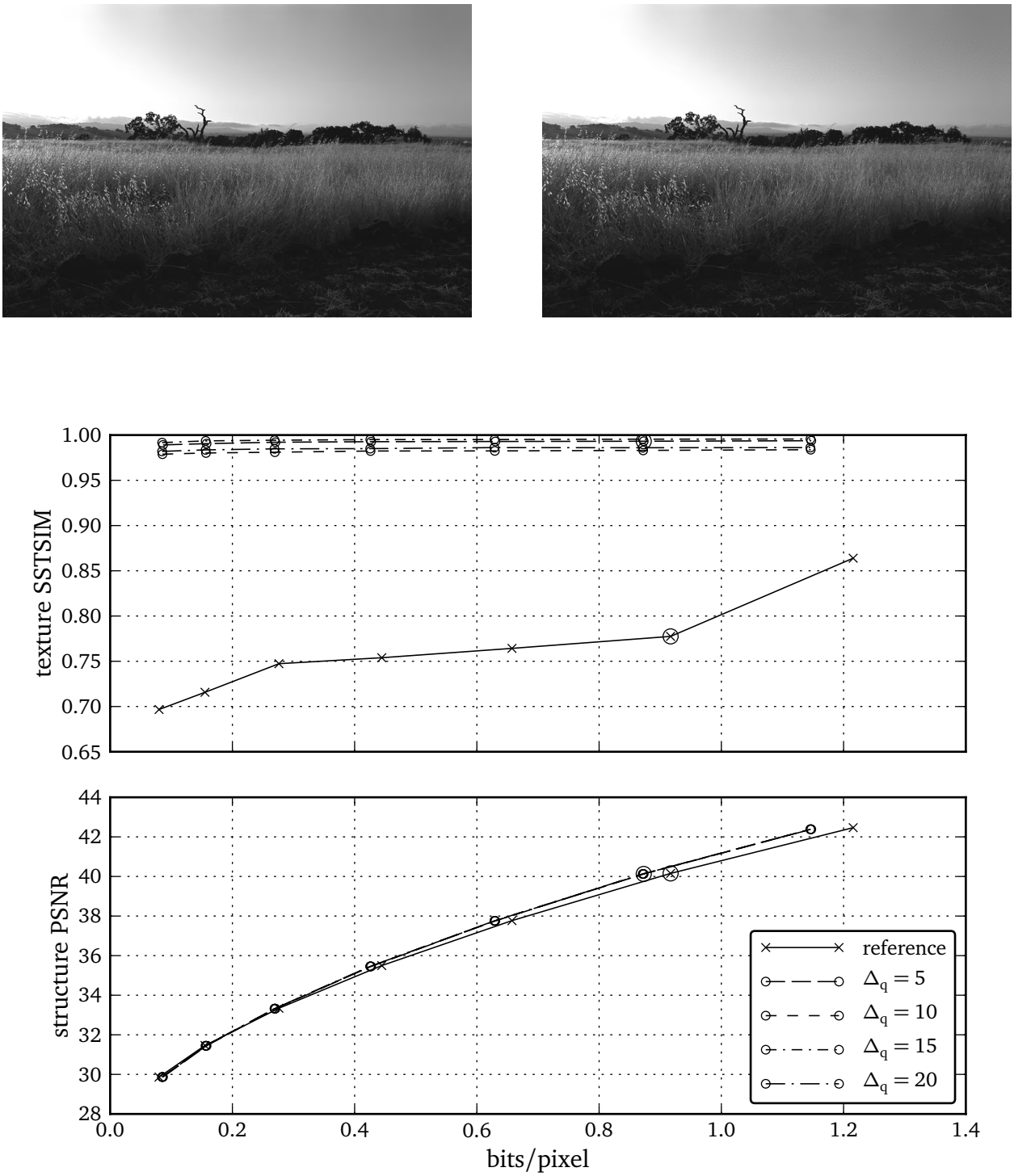


(a) Results for SYDNEY image, 1024 × 672 pixels. © author

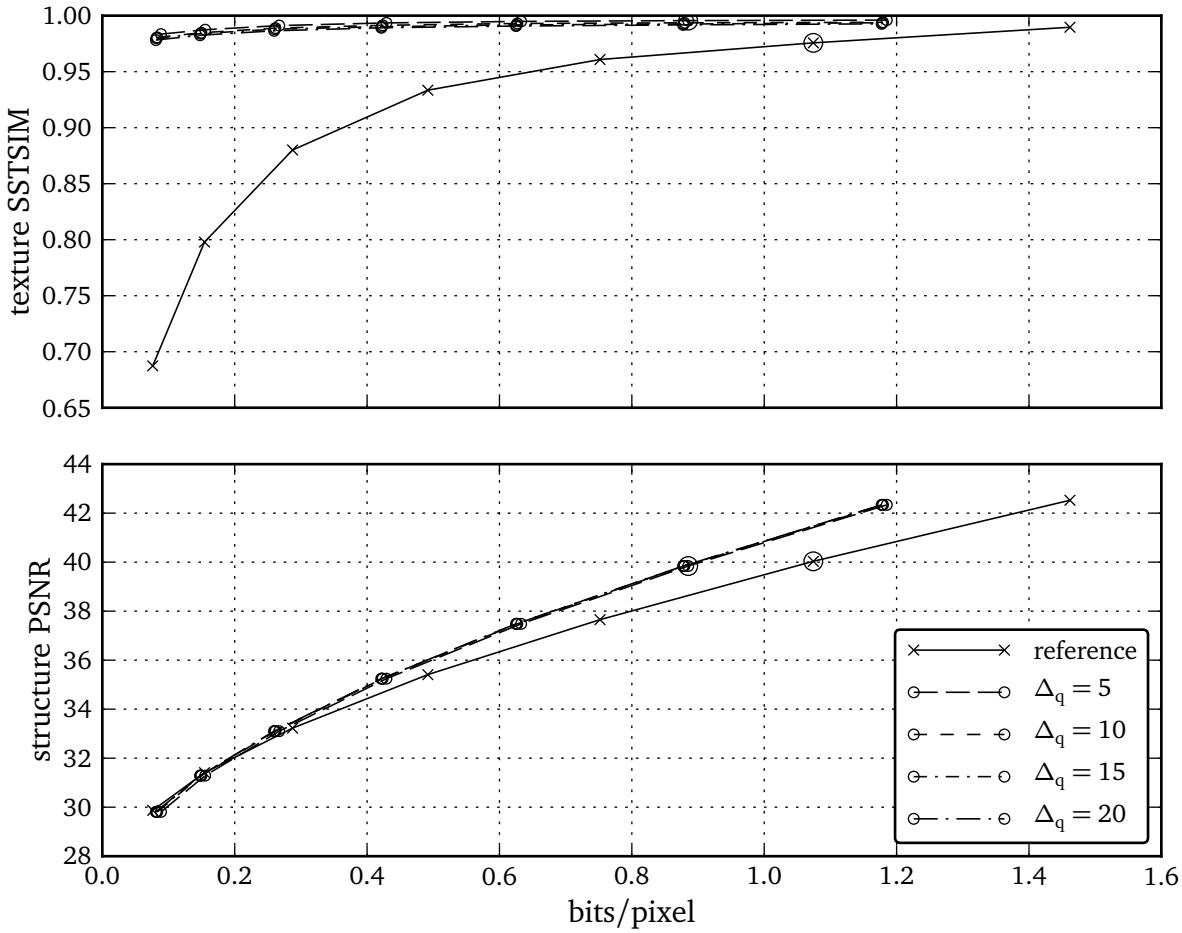
Figure A.1 Results for selected test images (continued on following pages). Top left: reference at QP = 25; top right: reconstruction at QP = 25, $\Delta_q = 5$; bottom: overall bitrate vs. distortion at varying Δ_q and QP $(22, 25, \ldots, 40)$. The images correspond to the data points marked with large circles.



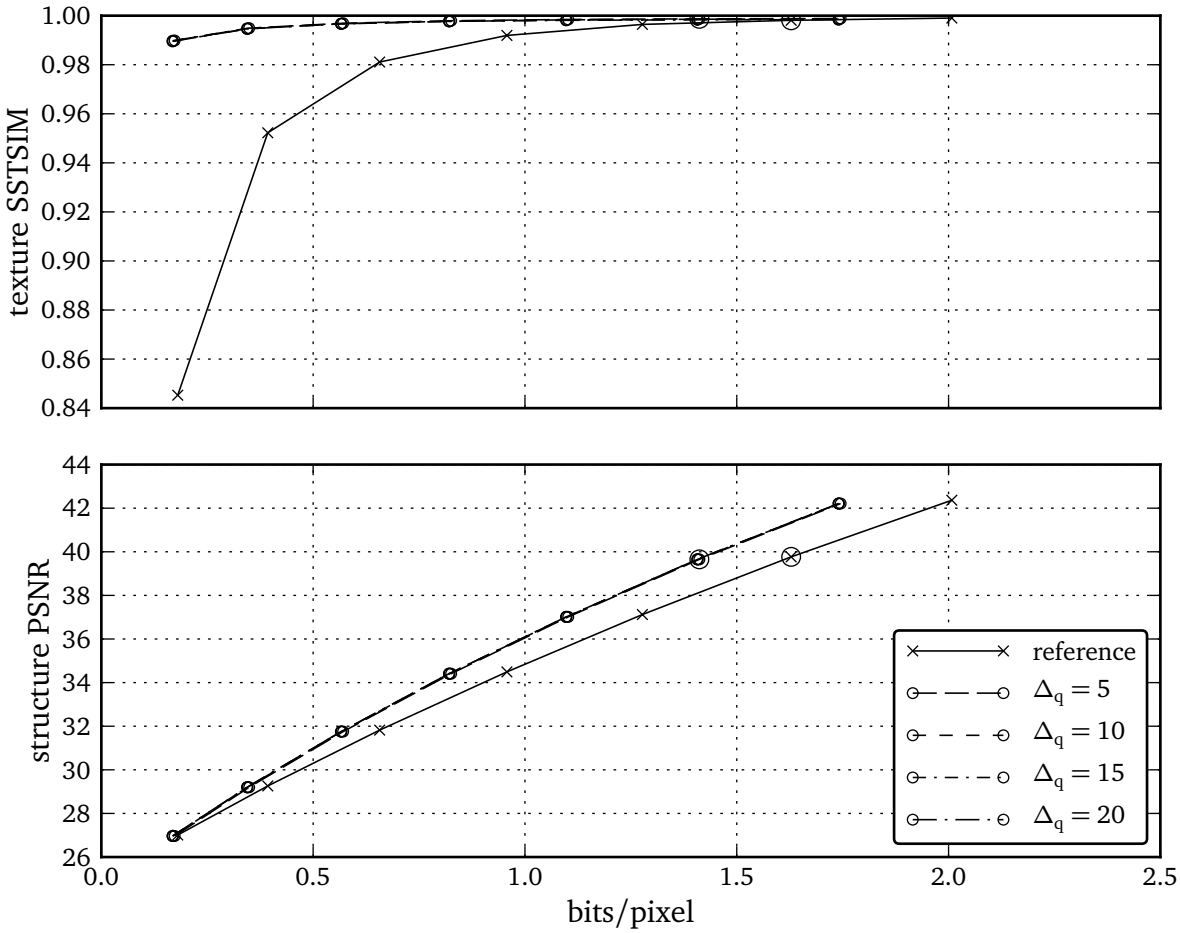
(b) Results for CORRIDOR image, 512 × 512 pixels. [VisTex]



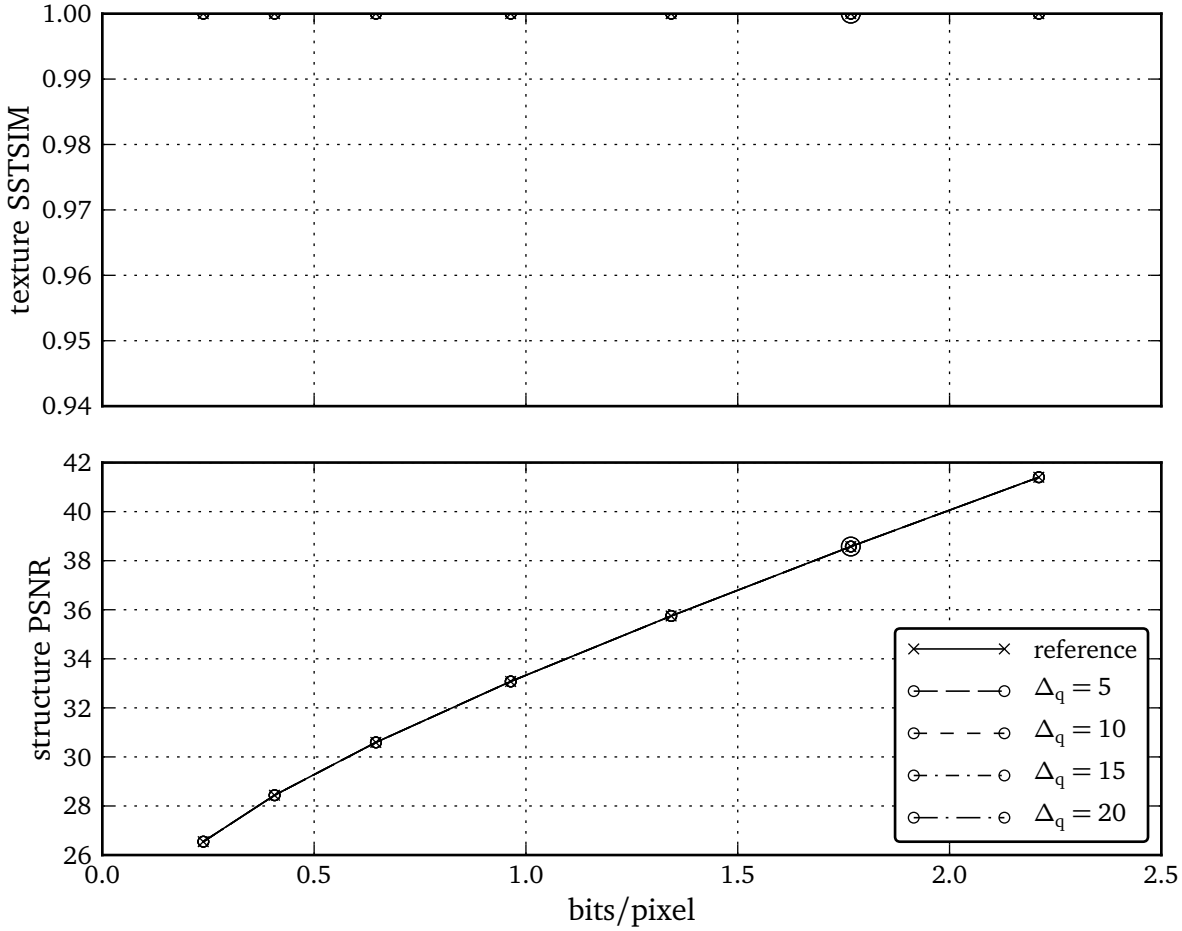
(c) Results for GRASS image, 768 × 512 pixels. [VisTex]



(d) Results for ULURU image, 768 × 1024 pixels. © author

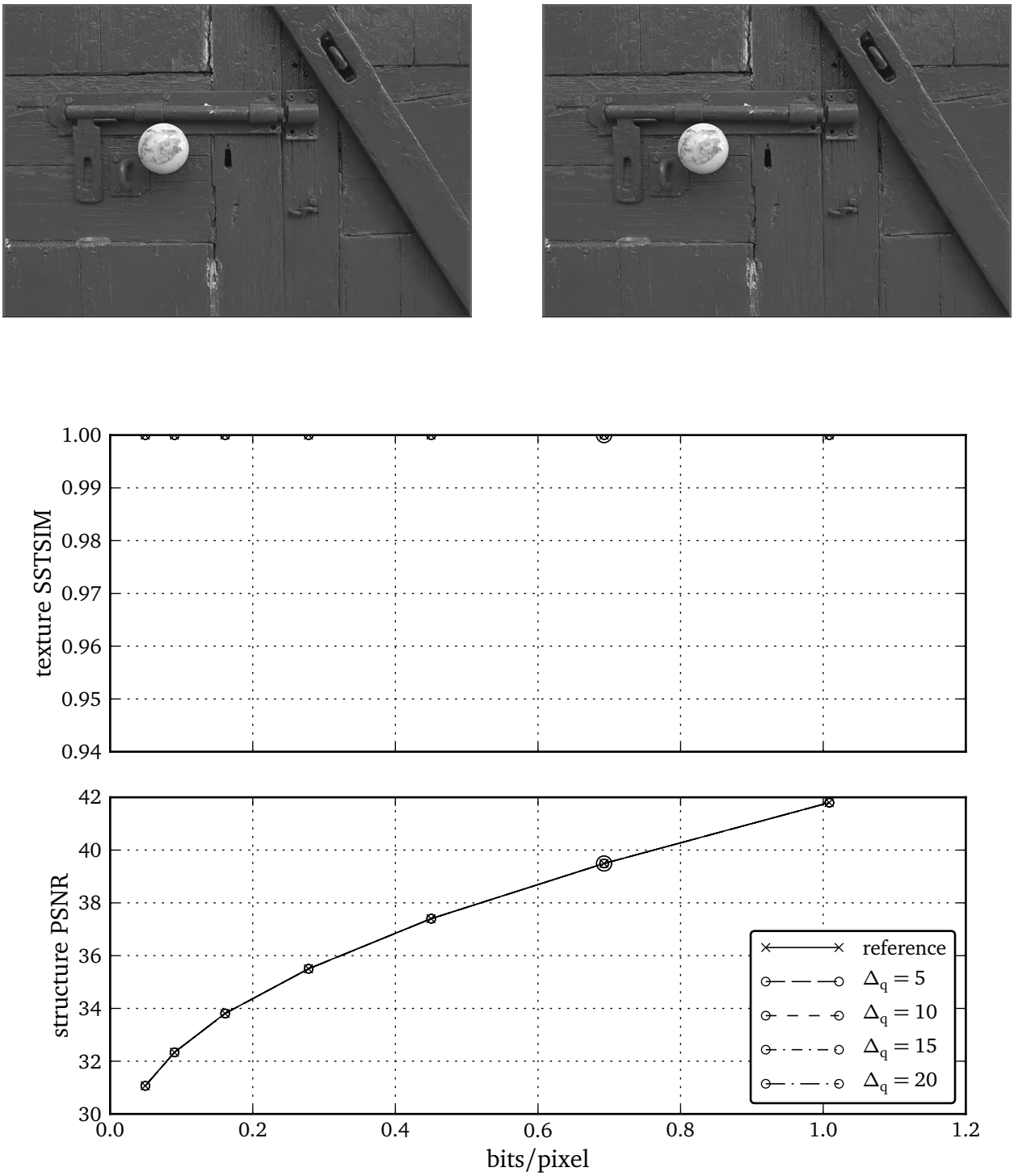


(e) Results for BIG BUILDING image, 1792 × 1344 pixels. [ImCoIn]

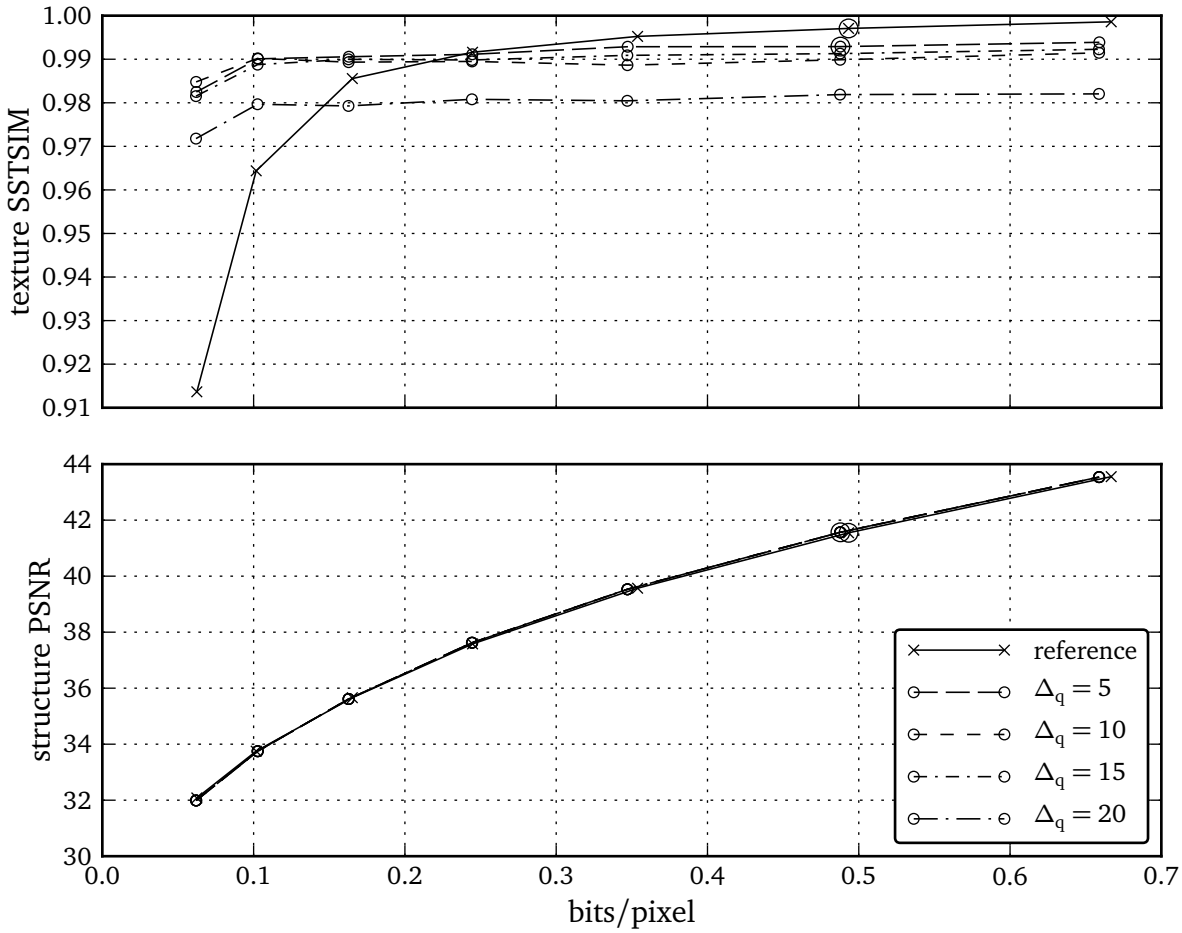


(a) Results for KODIM01 image, 768×512 pixels.

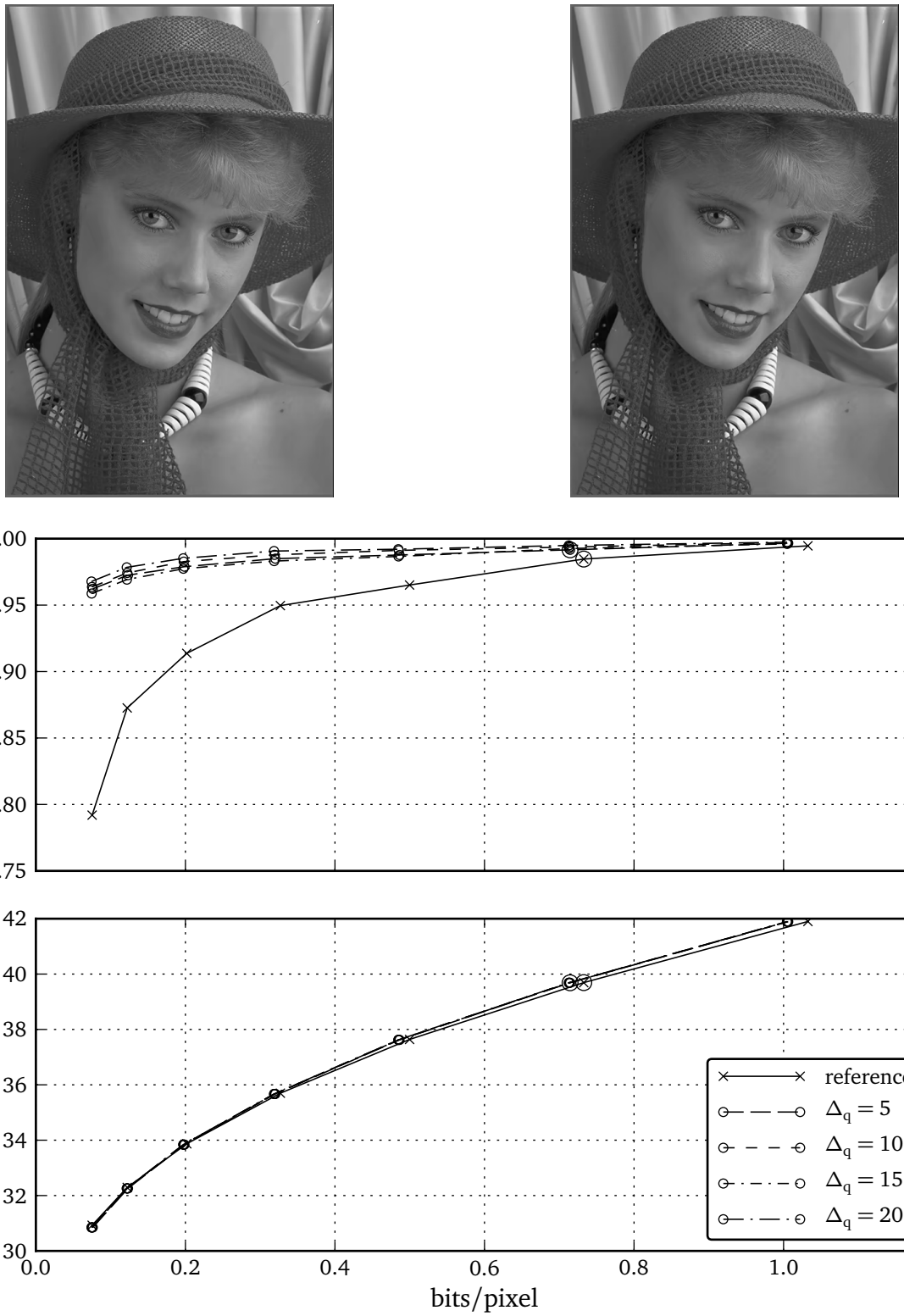
Figure A.2 Results for Kodak test set [Kodak] (continued on following pages). Top left: reference at QP $= 25$; top right: reconstruction at QP $= 25$, $\Delta_q = 5$; bottom: overall bitrate vs. distortion at varying Δ_q and QP $(22, 25, \ldots, 40)$. The images correspond to the data points marked with large circles.



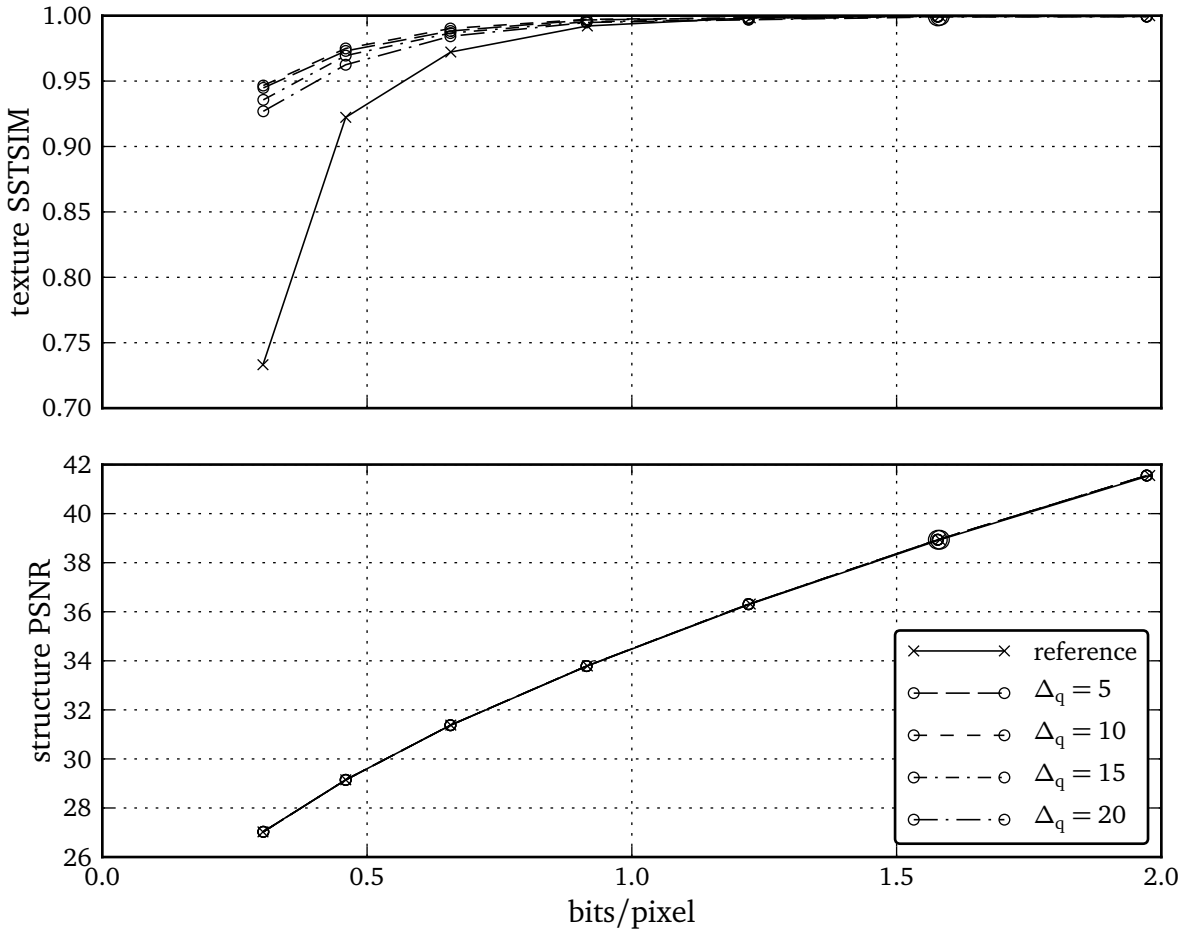
(b) Results for KODIM02 image, 768 × 512 pixels.



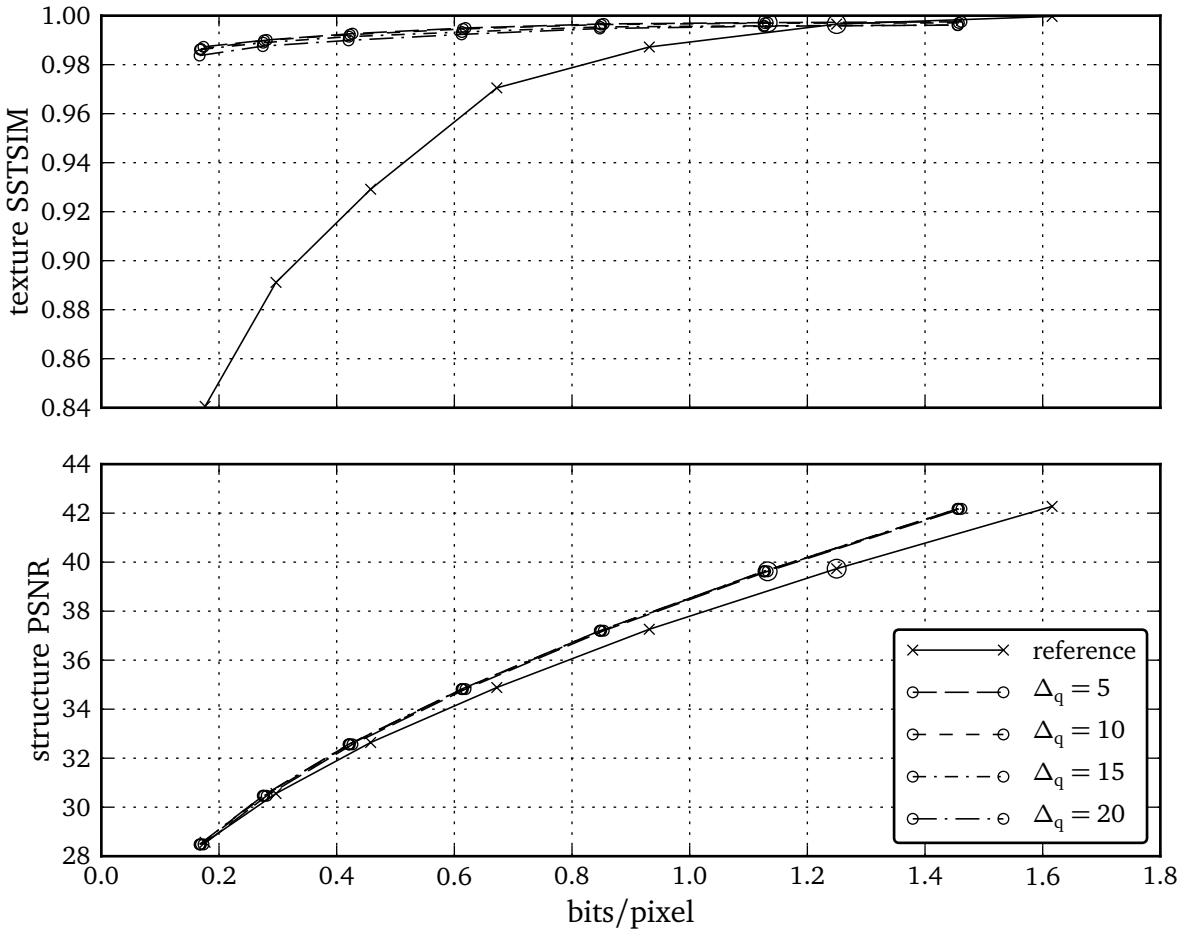
(c) Results for KODIM03 image, 768 × 512 pixels.



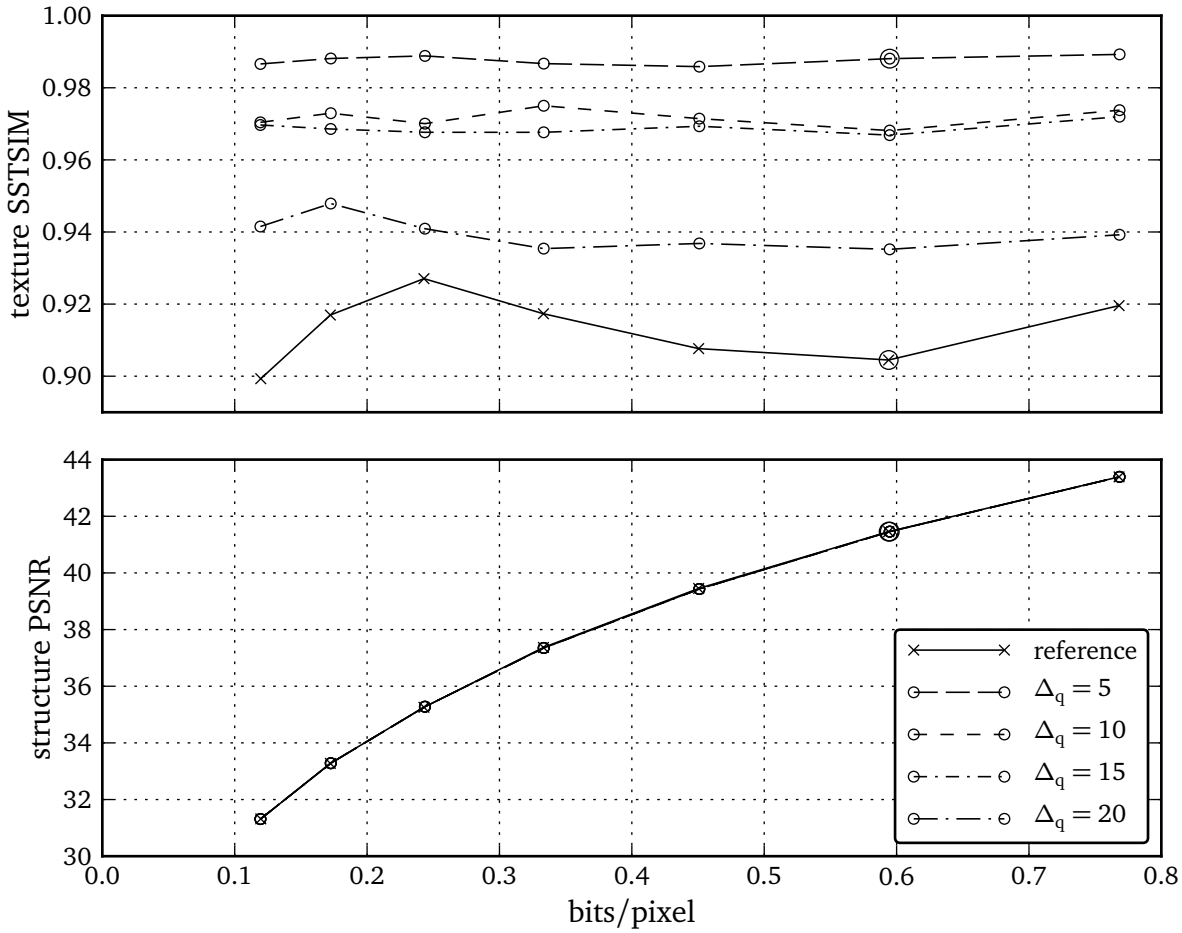
(d) Results for KODIM04 image, 512 × 768 pixels.



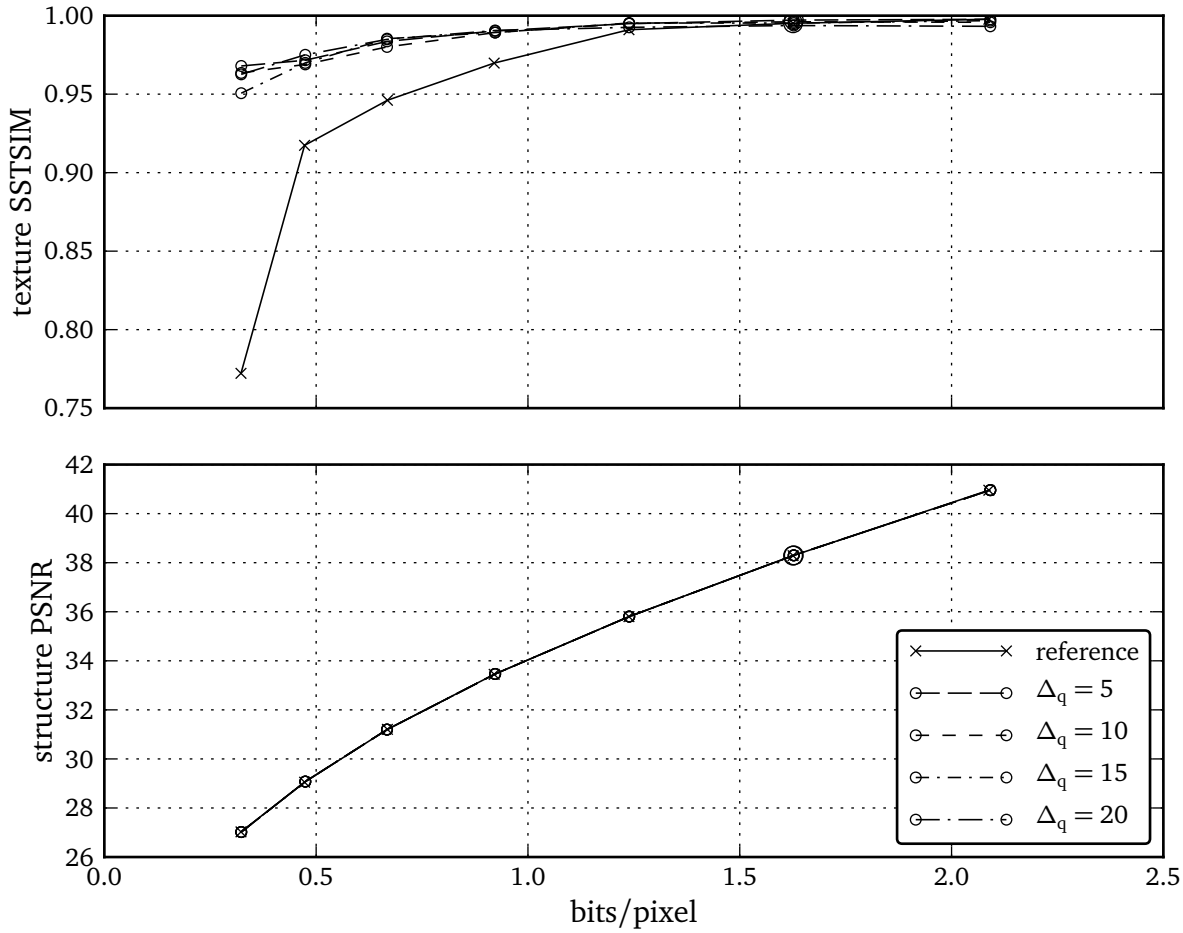
(e) Results for KODIM05 image, 768 × 512 pixels.



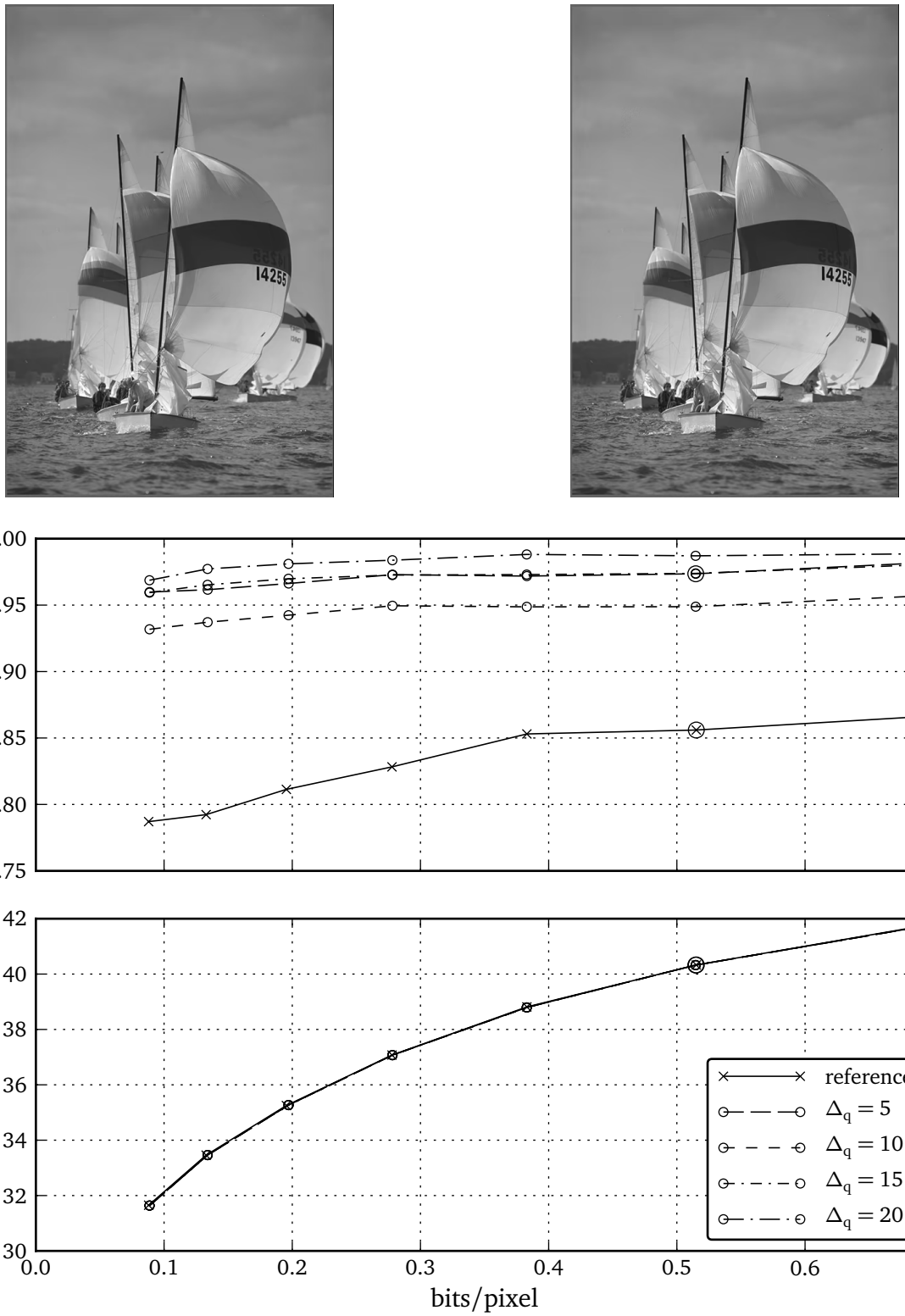
(f) Results for KODIM06 image, 768 × 512 pixels.



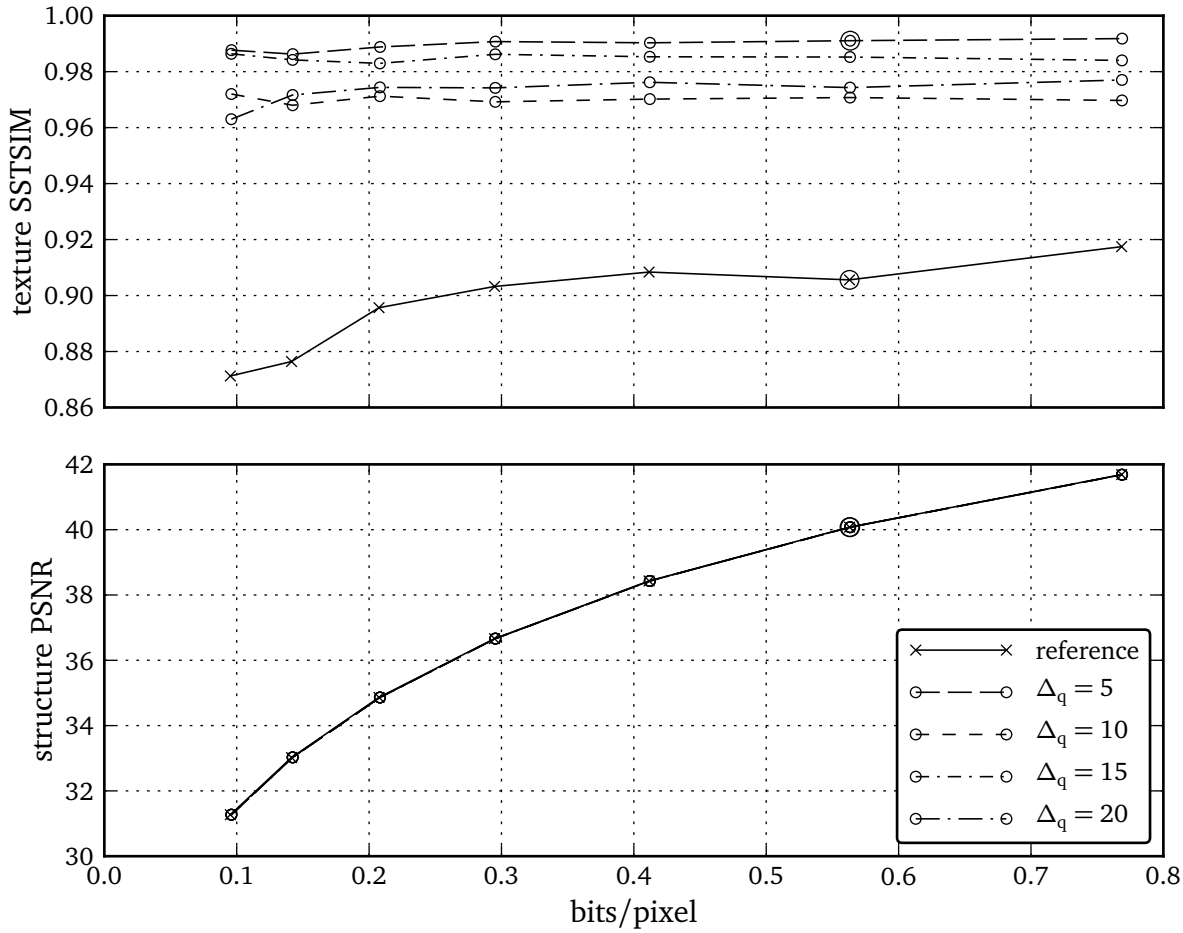
(g) Results for KODIM07 image, 768 × 512 pixels.



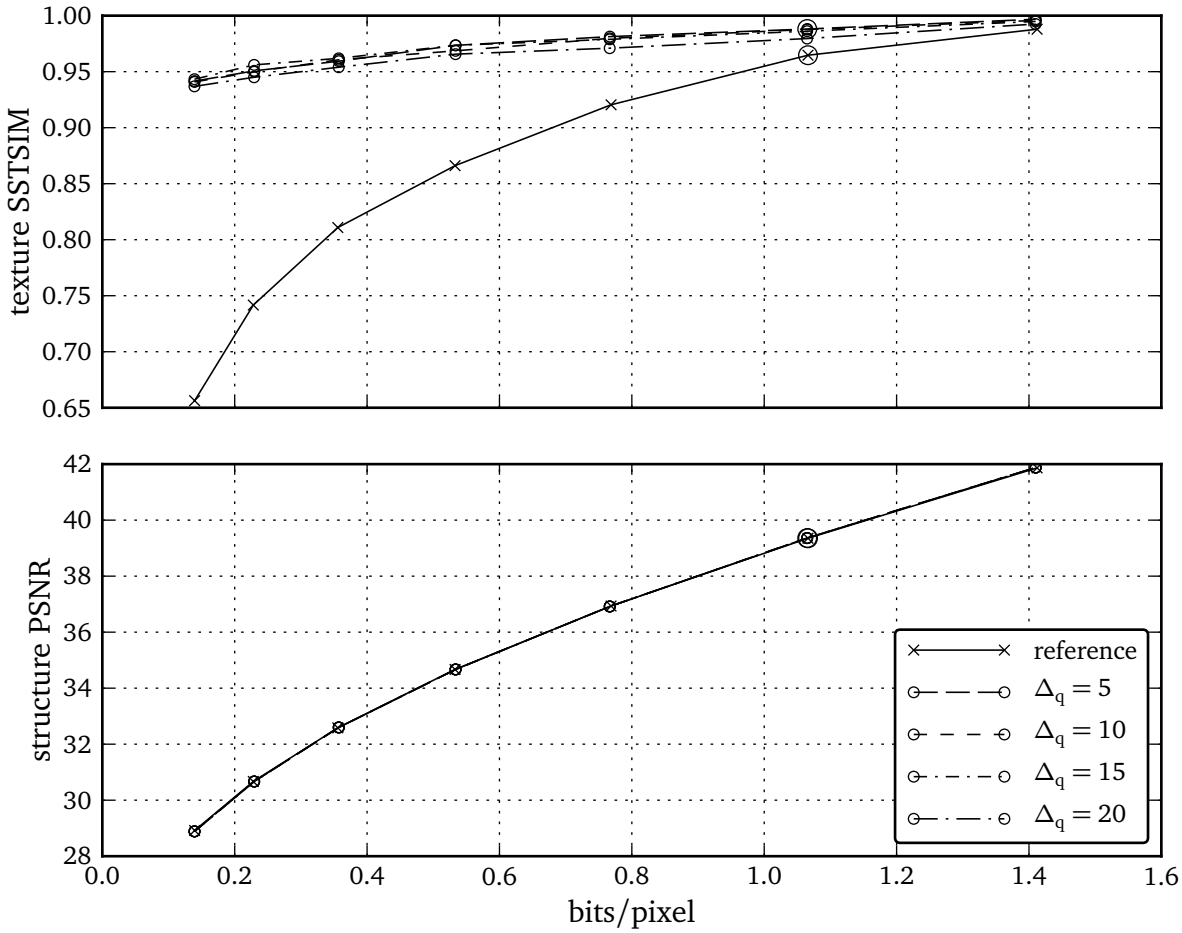
(h) Results for KODIM08 image, 768 × 512 pixels.



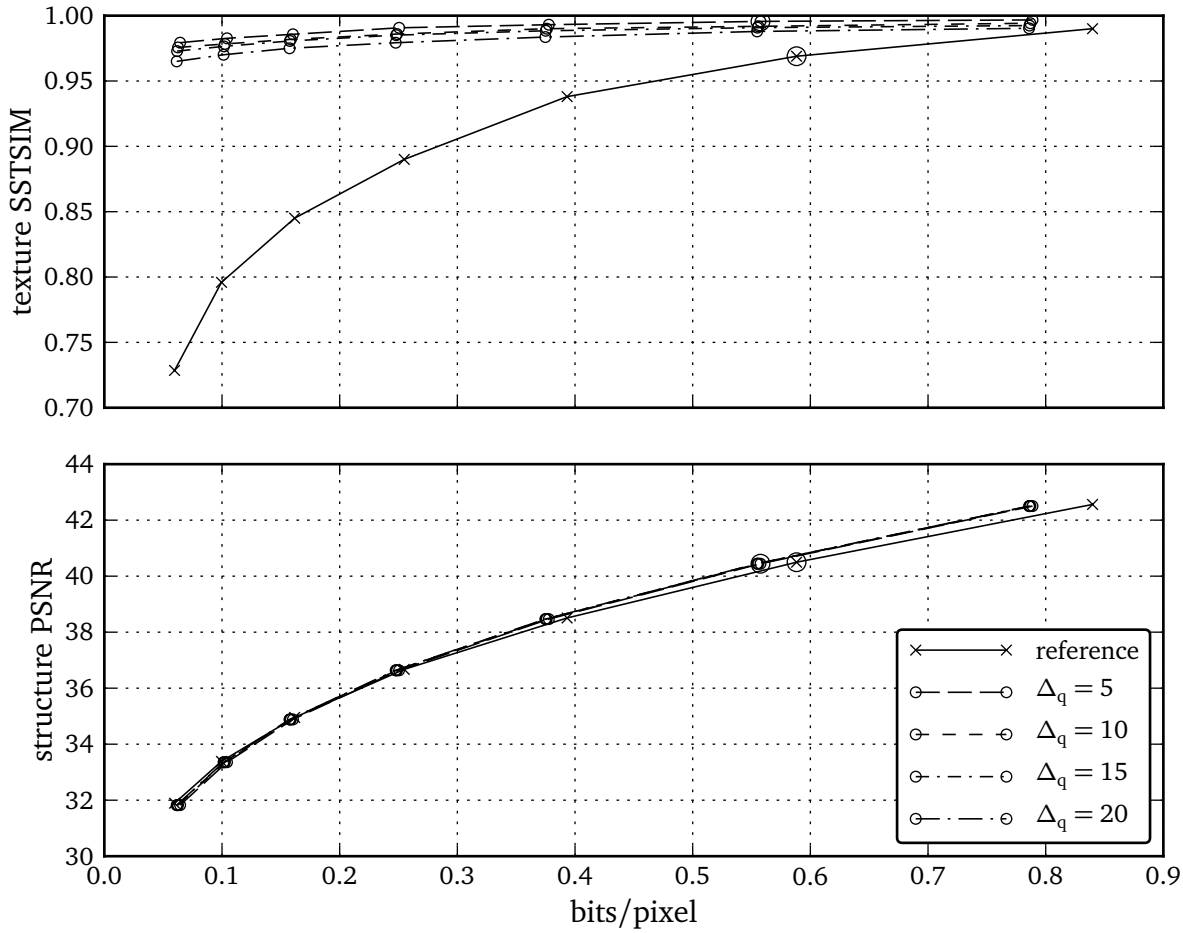
(i) Results for KODIM09 image, 512 × 768 pixels.



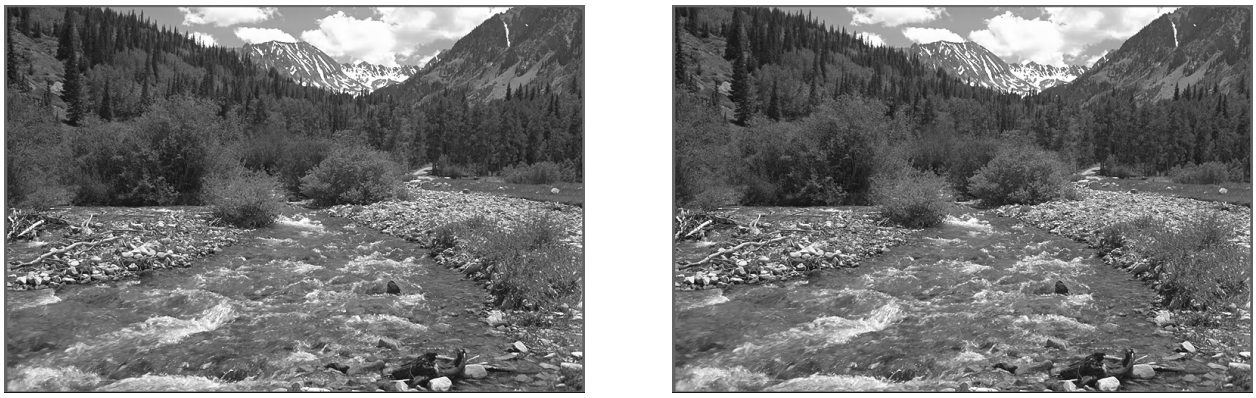
(j) Results for KODIM10 image, 512 × 768 pixels.

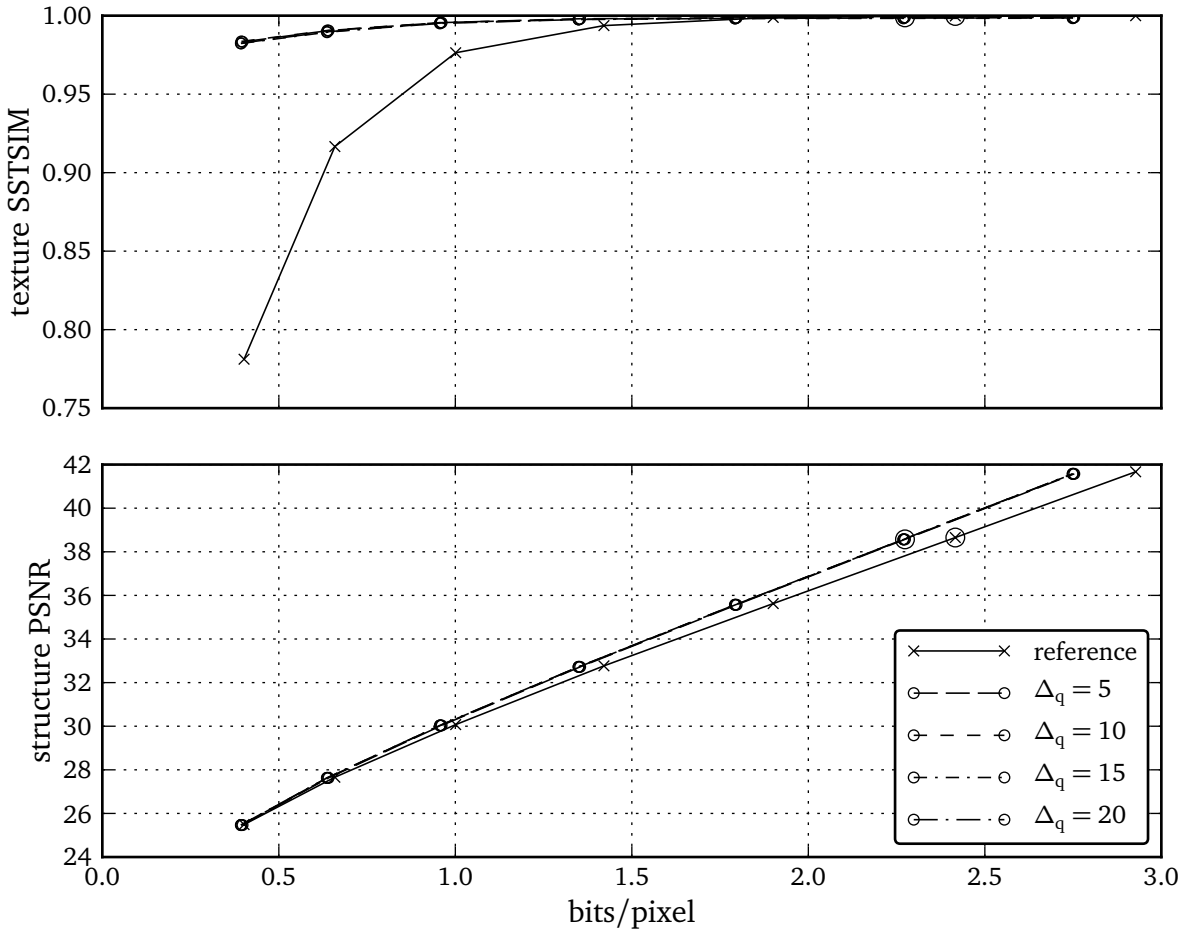


(k) Results for KODIM11 image, 768 × 512 pixels.

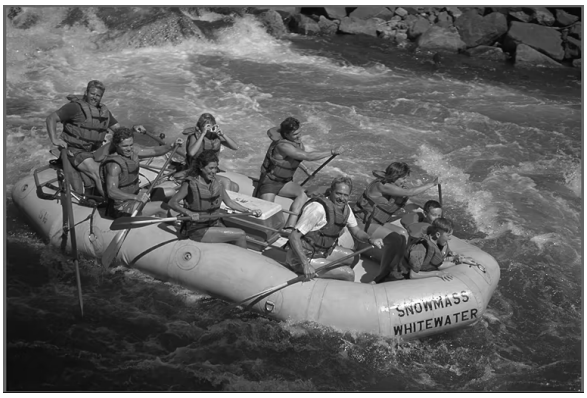


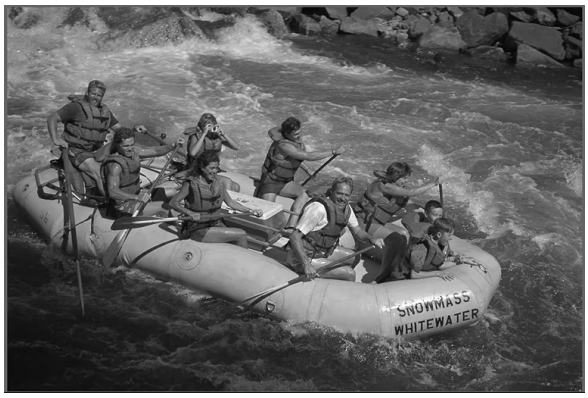
(l) Results for KODIM12 image, 768 × 512 pixels.

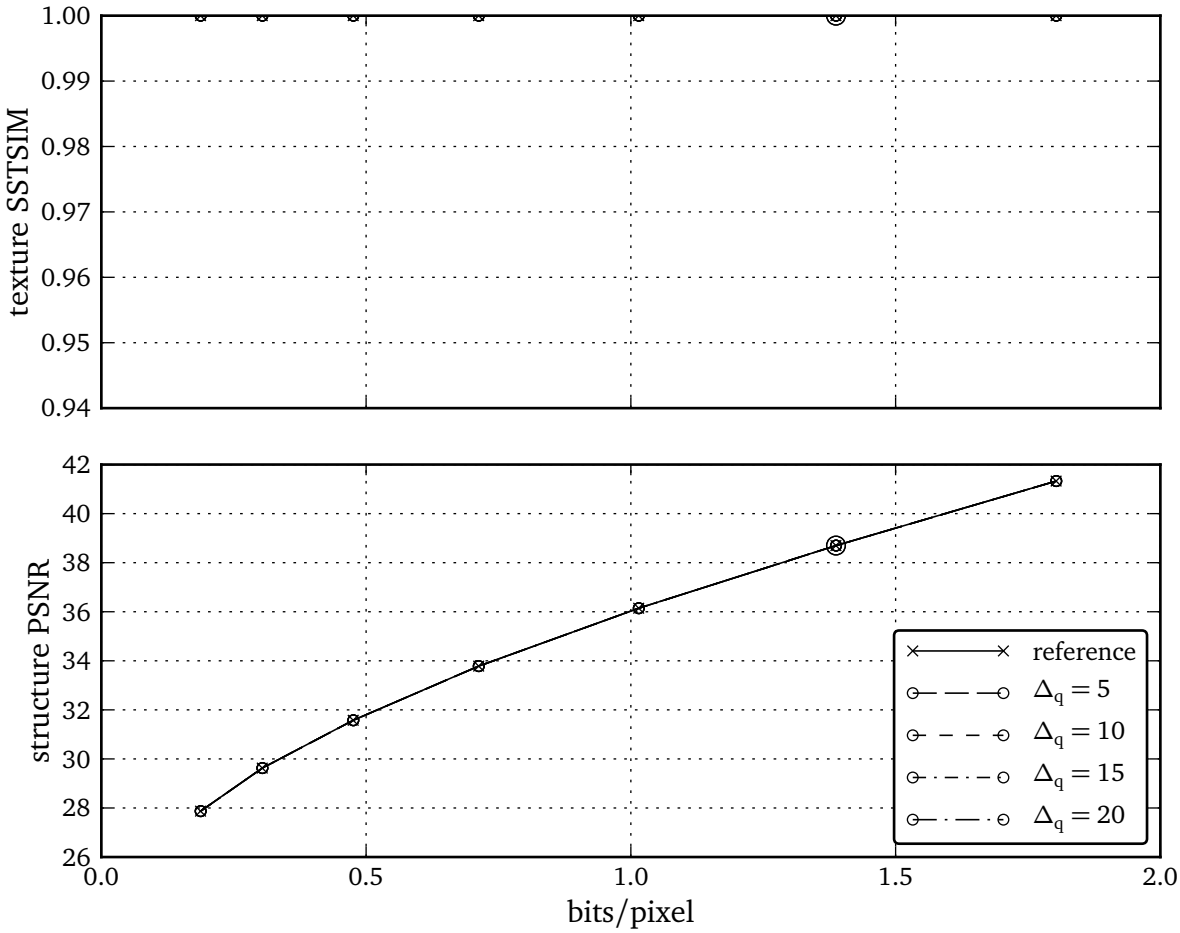




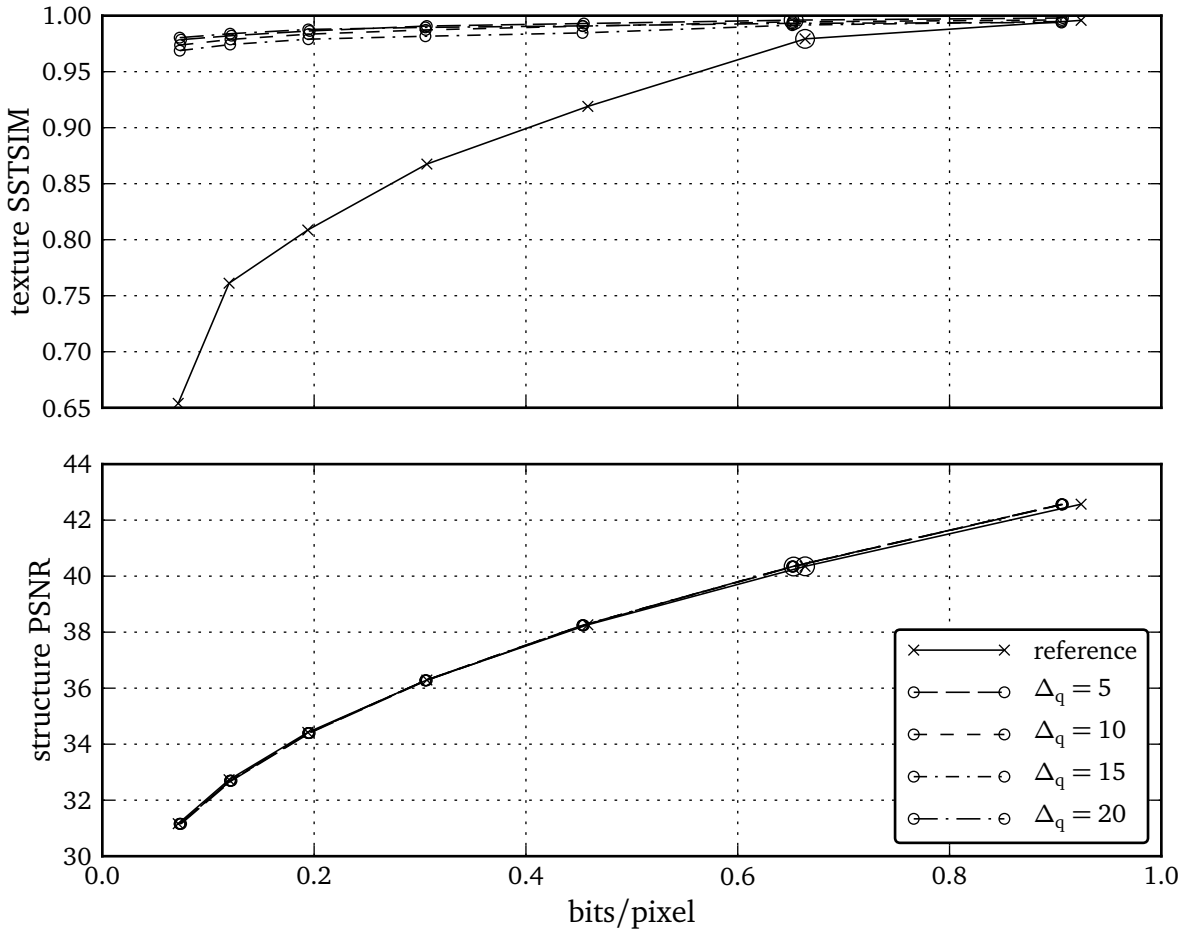
(m) Results for KODIM13 image, 768 × 512 pixels.



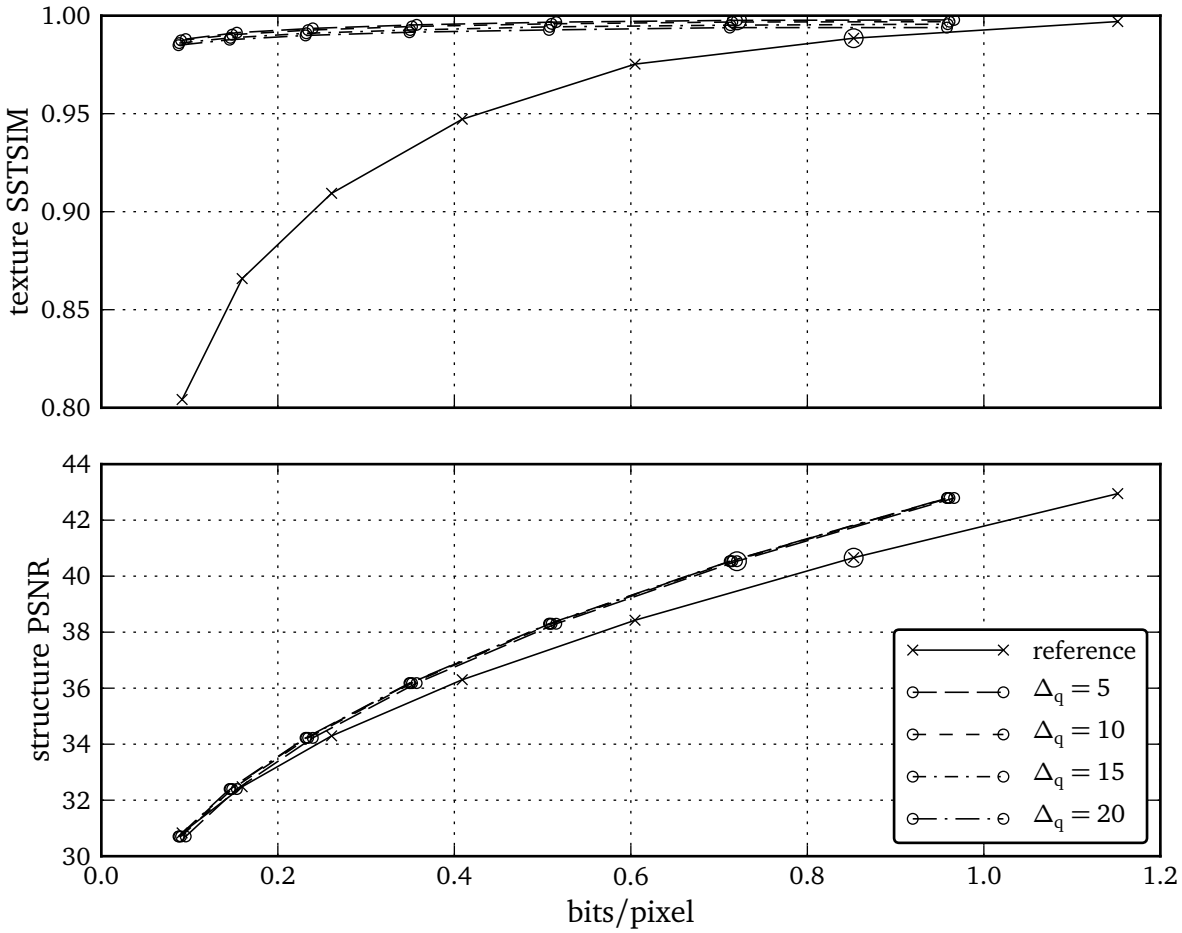




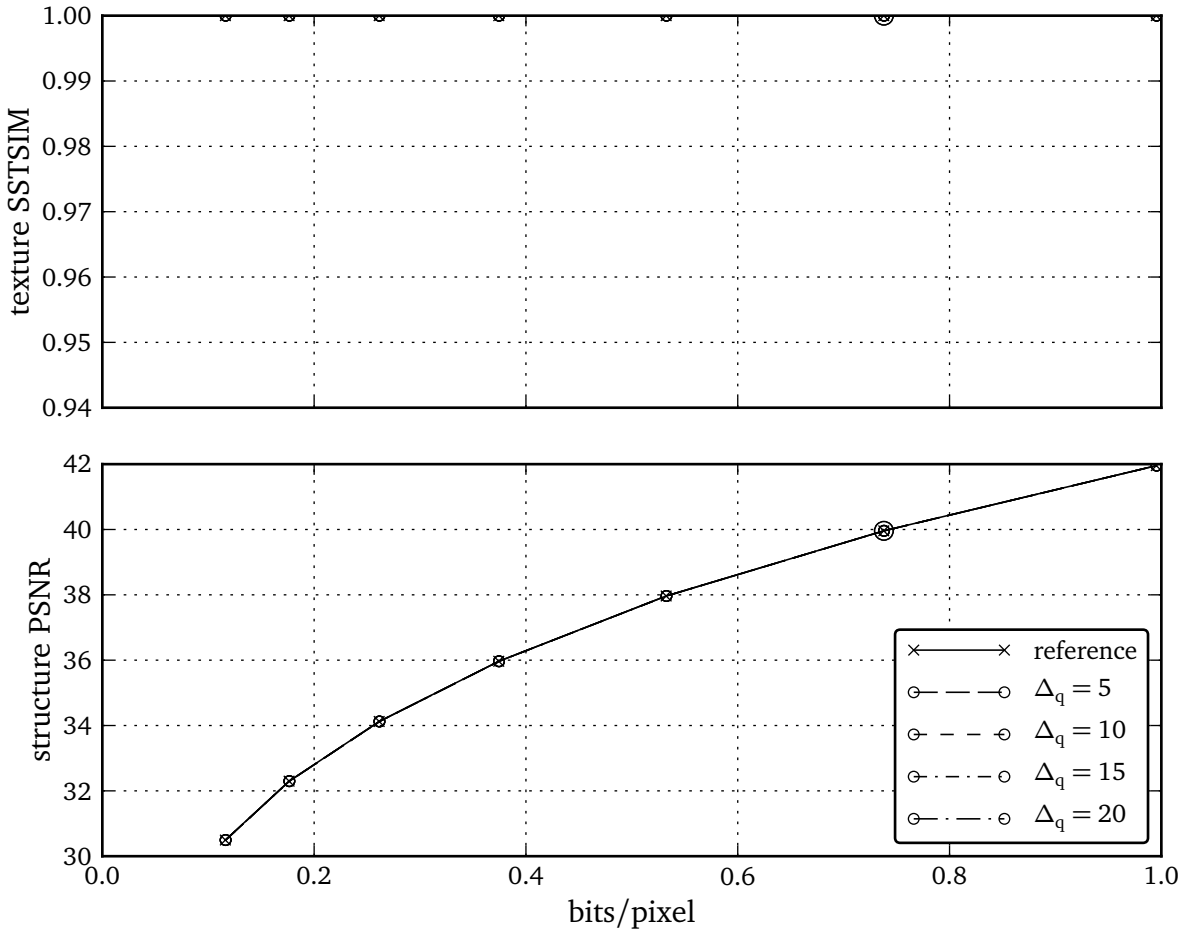
(n) Results for KODIM14 image, 768 × 512 pixels.



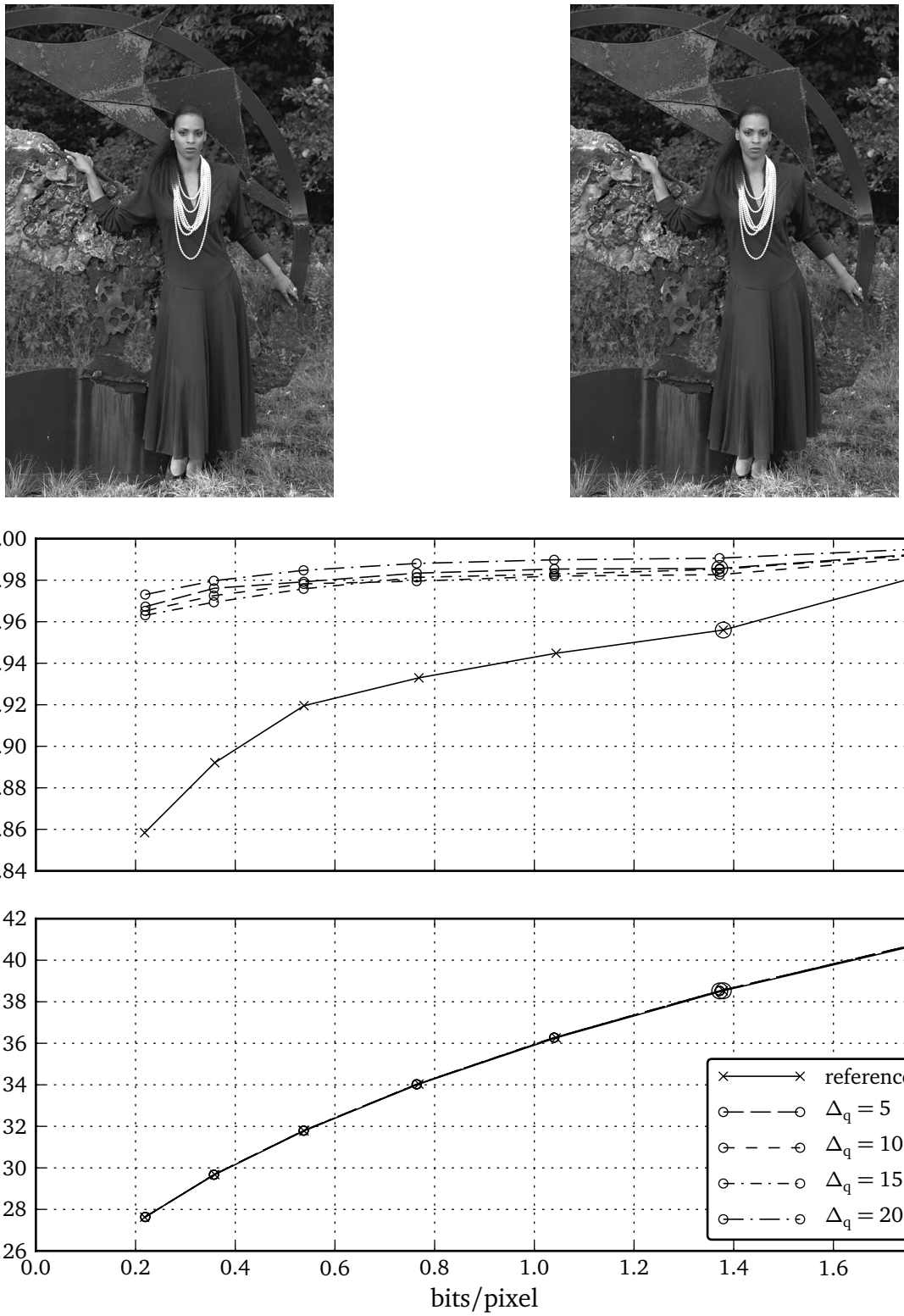
(o) Results for KODIM15 image, 768 × 512 pixels.



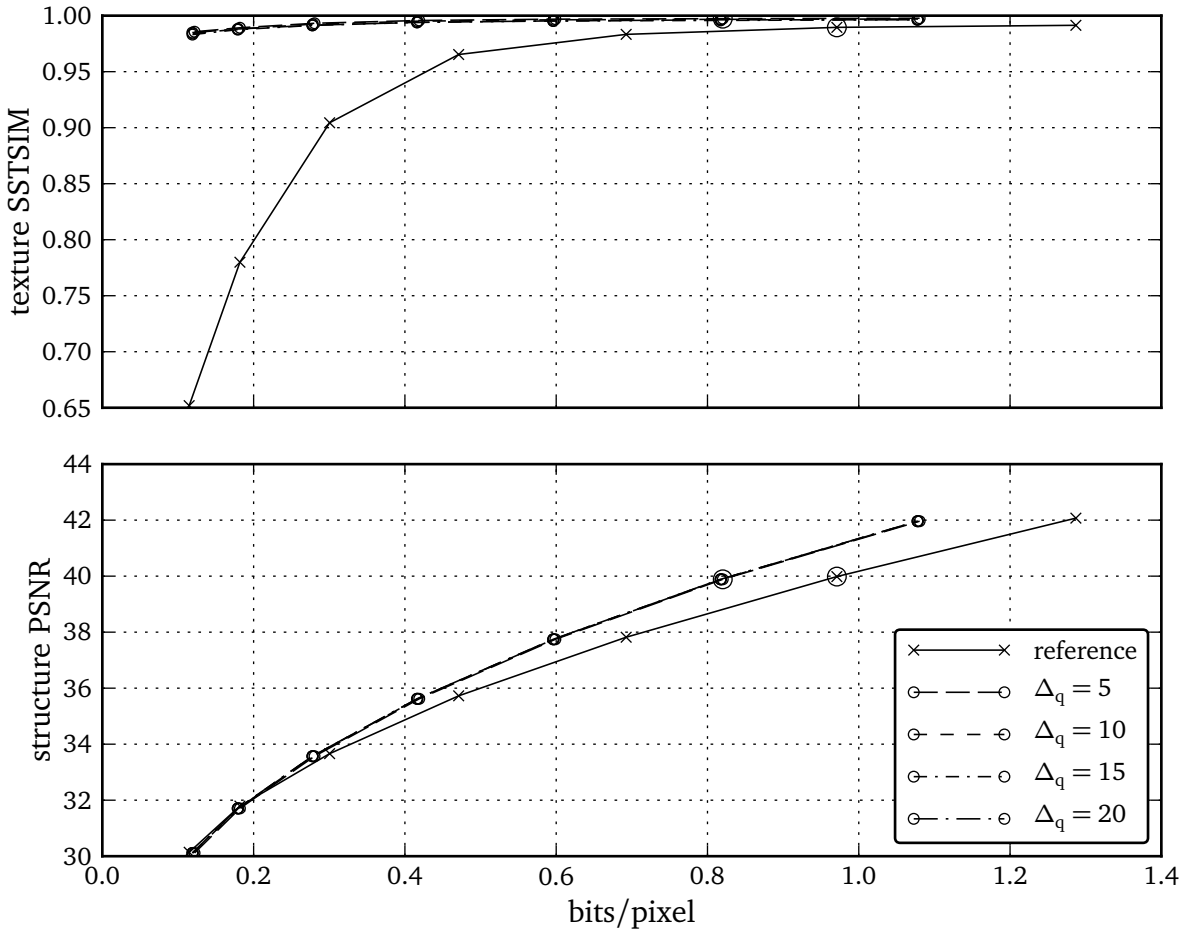
(p) Results for KODIM16 image, 768 × 512 pixels.



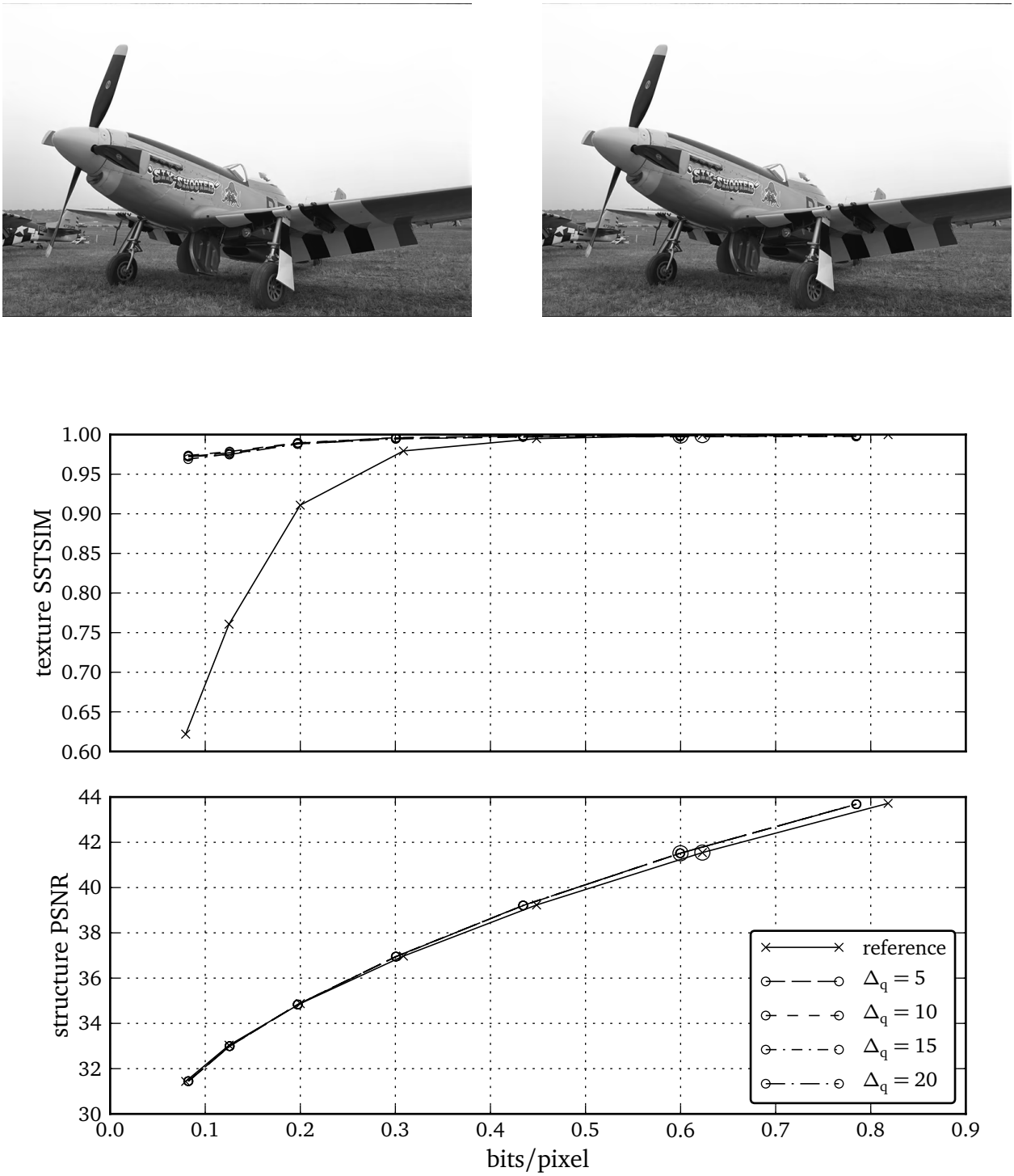
(q) Results for KODIM17 image, 512 × 768 pixels.



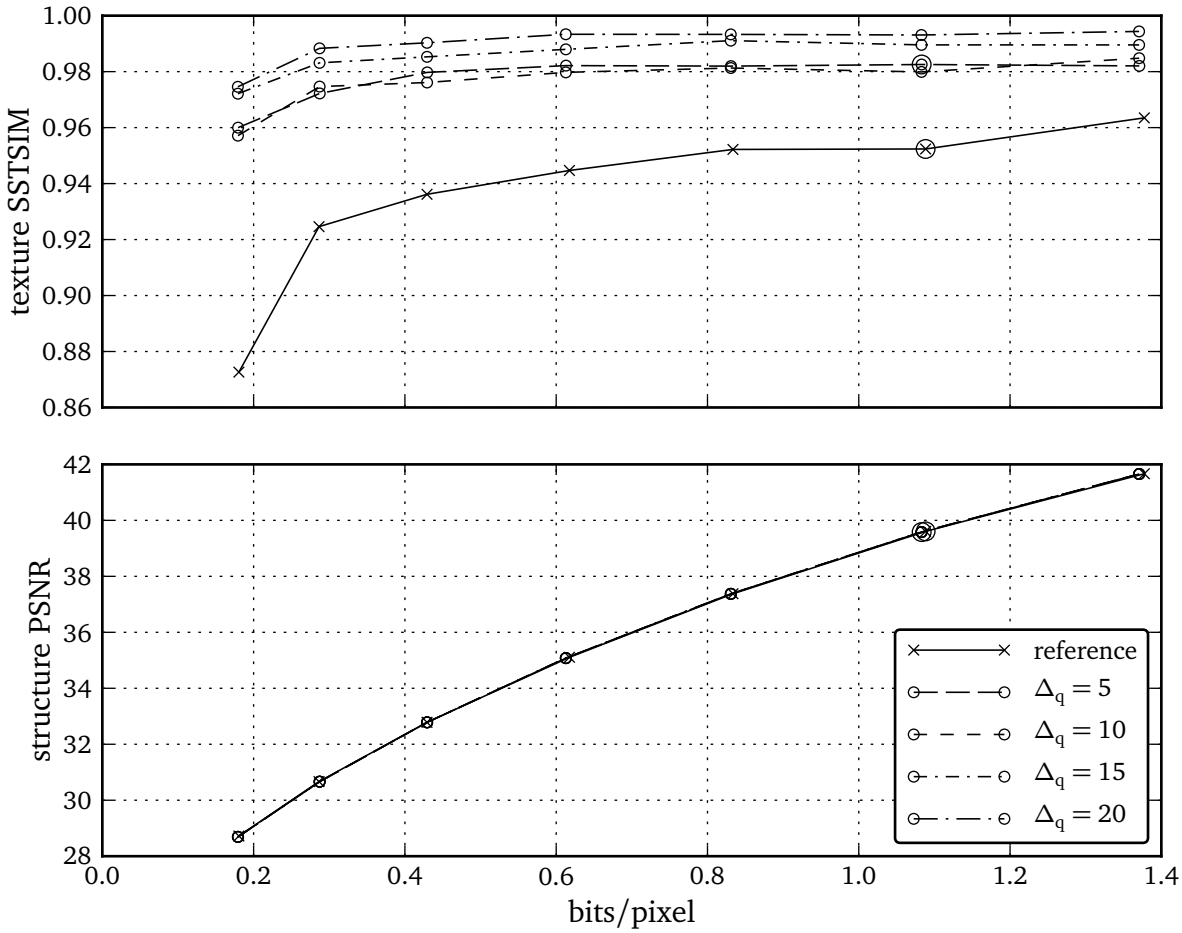
(r) Results for KODIM18 image, 512 × 768 pixels.



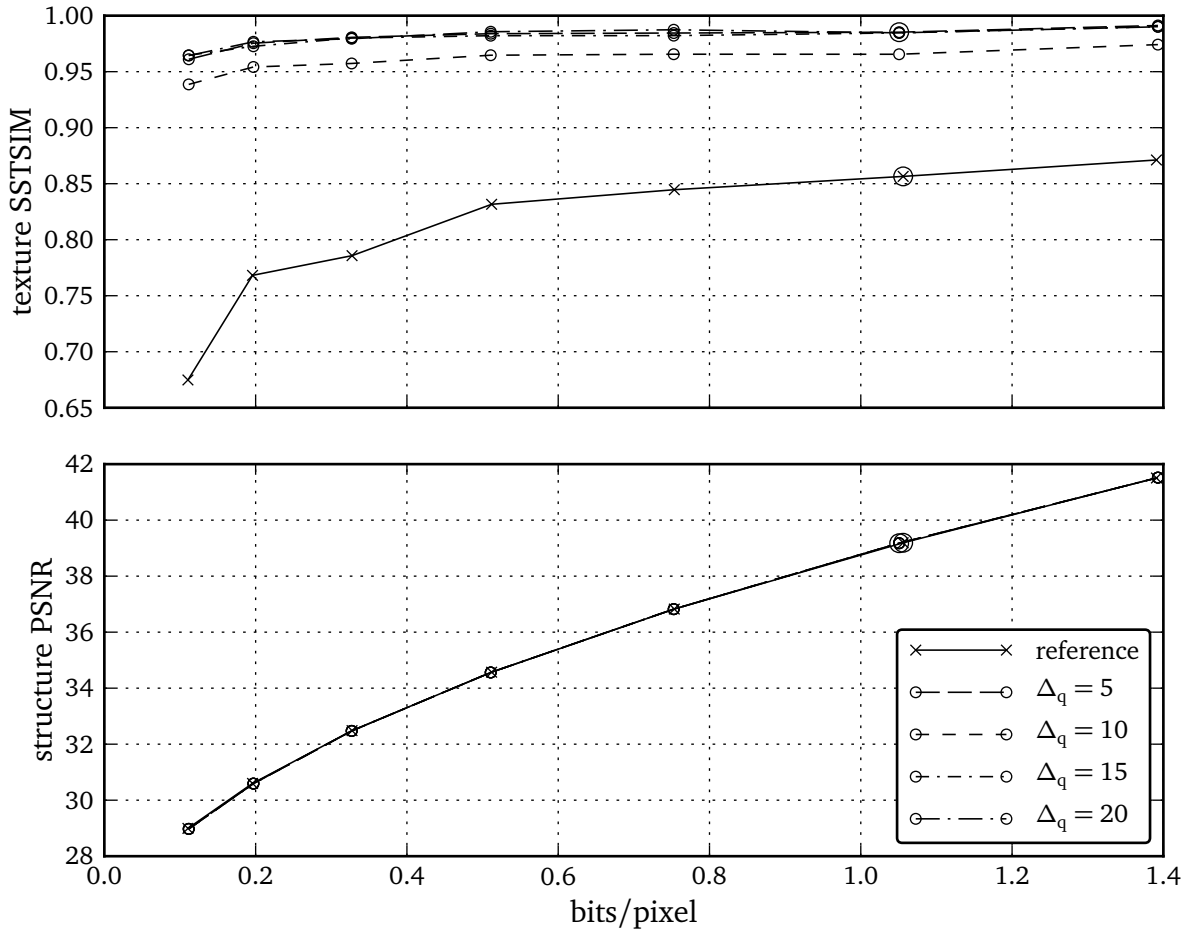
(s) Results for KODIM19 image, 512×768 pixels.



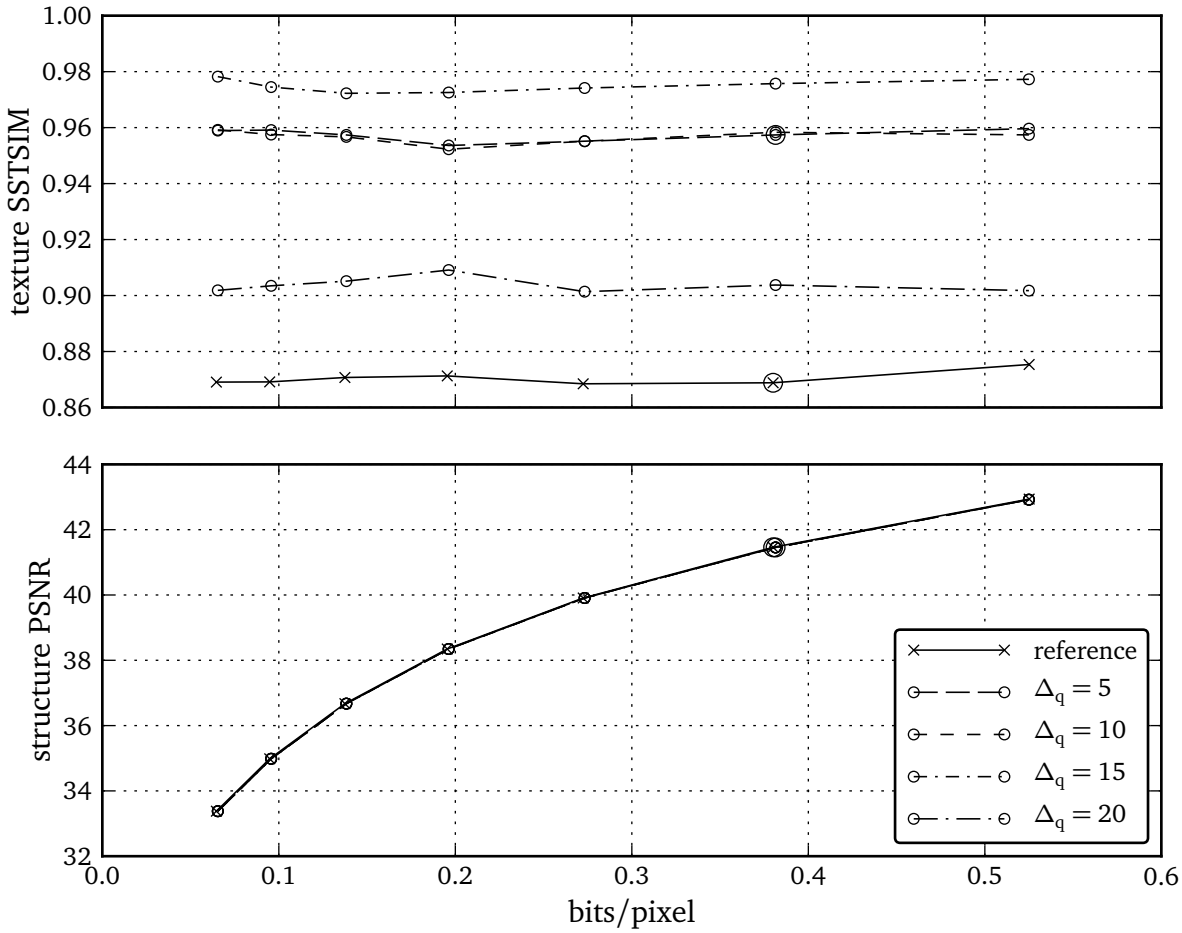
(t) Results for KODIM20 image, 768 × 512 pixels.



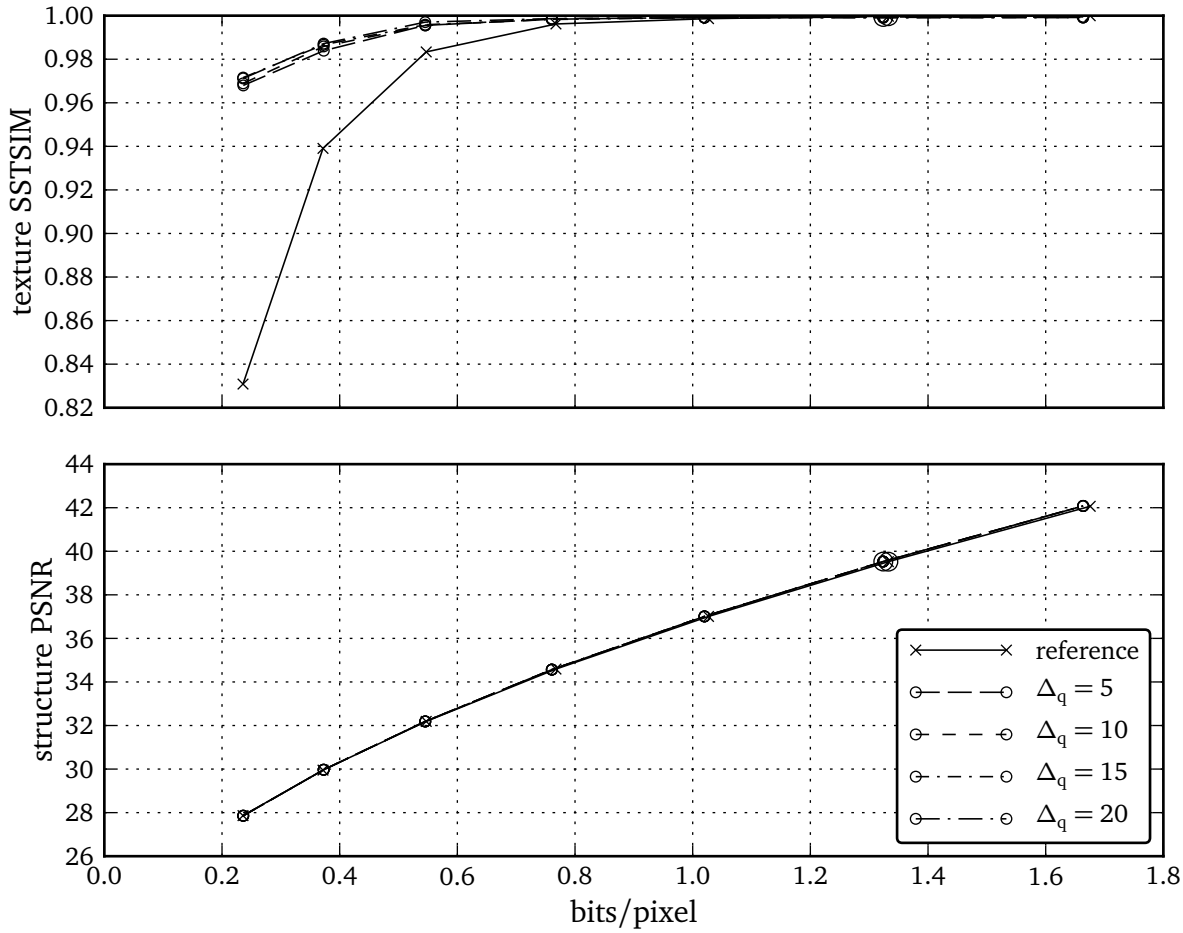
(u) Results for KODIM21 image, 768 × 512 pixels.



(v) Results for KODIM22 image, 768 × 512 pixels.



(w) Results for KODIM23 image, 768×512 pixels.



(x) Results for KODIM24 image, 768 × 512 pixels.

Bibliography

[AKZ11] Daniel Arfib, Florian Keiler, and Udo Zölzer. “Source–Filter Processing.” In: *DAFX — Digital Audio Effects*. Edited by Udo Zölzer. 2nd edition. Wiley, New York, 2011. ISBN: 978-0-470-66599-2 (cited on page 1).

[Ash01] Michael Ashikhmin. “Synthesizing Natural Textures.” In: *Proc. of Symposium on Interactive 3D Graphics SI3D*. 2001, pages 217–226. DOI: 10.1145/364338.364405 (cited on page 7).

[Auj+06] Jean-François Aujol, Guy Gilboa, Tony Chan, and Stanley Osher. “Structure–Texture Image Decomposition—Modeling, Algorithms, and Parameter Selection.” In: *International Journal of Computer Vision* 67.1 (2006), pages 111–136. DOI: 10.1007/s11263-006-4331-z (cited on page 59).

[AVC] *ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC: Advanced Video Coding for Generic Audiovisual Services* (cited on pages 1, 84).

[Bac96] Michael Bach. “The Freiburg Visual Acuity Test – Automatic measurement of visual acuity.” In: *Optometry & Vision Science* 73.1 (Jan. 1996), pages 49–53. ISSN: 1040-5488 (cited on page 37).

[Bal12] Johannes Ballé. “Subjective Evaluation of Texture Similarity Metrics for Compression Applications.” In: *Proc. of International Picture Coding Symposium PCS ’12*. (Kraków, Poland). IEEE, Piscataway, May 2012. DOI: 10.1109/PCS.2012.6213337 (cited on page 36).

[BJS09] Johannes Ballé, Bastian Jurczyk, and Aleksandar Stojanovic. “Component-Based Image Coding using Non-Local Means Filtering and an Autoregressive Texture Model.” In: *Proc. of IEEE International Conference on Image Processing ICIP ’09*. (Cairo, Egypt). IEEE, Piscataway, Nov. 2009, pages 1937–1940. ISBN: 978-1-4244-5653-6. DOI: 10.1109/ICIP.2009.5414524 (cited on page 81).

[Boc33] Salomon Bochner. “Monotone Funktionen, Stieltjessche Integrale und harmonische Analyse.” In: *Mathematische Annalen* 108 (1933). Edited by David Hilbert, pages 378–410 (cited on page 8).

[Bov91] Alan Conrad Bovik. “Analysis of Multichannel Narrow-Band Filters for Image Texture Segmentation.” In: *IEEE Transactions on Signal Processing* 39.9 (Sept. 1991), pages 2025–2043. ISSN: 1053-587X. DOI: 10.1109/78.134435 (cited on pages 35, 61).

[BSO11] Johannes Ballé, Aleksandar Stojanovic, and Jens-Rainer Ohm. “Models for Static and Dynamic Texture Synthesis in Image and Video Compression.” In: *IEEE Journal of Selected Topics in Signal Processing* 5.7 (Nov. 2011), pages 1353–1365. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2011.2166246 (cited on pages 3, 81, 86).

[BW05] Anders Blomqvist and Bo Wahlberg. “On Frequency Weighting in Autoregressive Spectral Estimation.” In: *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP ’05*. (Philadelphia, PA, USA). Mar. 2005. DOI: 10.1109/ICASSP.2005.1415991 (cited on page 81).

[BW07] Anders Blomqvist and Bo Wahlberg. “On the Relation Between Weighted Frequency-Domain Maximum-Likelihood Power Spectral Estimation and the Prefiltered Covariance Extension Approach.” In: *IEEE Transactions on Signal Processing* 55.1 (Jan. 2007), pages 384–389. DOI: 10.1109/TSP.2006.885780 (cited on page 81).

[BW09] Johannes Ballé and Mathias Wien. “A Quantization Scheme for Modeling and Coding of Noisy Texture in Natural Images.” In: *Proc. of IASTED Conference on Signal and Image Processing SIP ’09*. (Honolulu, HI, USA). ACTA Press, Calgary, Aug. 2009. ISBN: 978-0-88986-803-8 (cited on page 81).

[Byr+08] James Byrne, Stephen Ierodiaconou, David Bull, David Redmill, and Paul Hill. “Unsupervised Image Compression-by-Synthesis within a JPEG Framework.” In: *Proc. of IEEE International Conference on Image Processing ICIP ’08*. 2008. DOI: 10.1109/ICIP.2008.4712399 (cited on page 2).

[BZD11] Marc Bosch, Fengqing Zhu, and Edward J. Delp. “Segmentation Based Video Compression Using Texture and Motion Models.” In: *IEEE Journal of Selected Topics in Signal Processing* 5.7 (Nov. 2011), pages 1366–1377. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2011.2164779 (cited on page 2).

[Can86] John Canny. “A Computational Approach to Edge Detection.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.6 (Nov. 1986), pages 679–698. DOI: 10.1109/TPAMI.1986.4767851 (cited on page 48).

[DK89] Haluk Derin and Patrick A. Kelly. “Discrete-Index Markov-Type Random Processes.” In: *Proceedings of the IEEE* 77.10 (Oct. 1989), pages 1485–1510. DOI: 10.1109/5.40665 (cited on pages 18, 21).

[DM84] Dan E. Dudgeon and Russell M. Mersereau. *Multidimensional Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, 1984. ISBN: 978-0-13-604959-3 (cited on page 24).

[DMM00] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. “Meaningful Alignments.” In: *International Journal of Computer Vision* 40.1 (2000), pages 7–23. DOI: 10.1023/A:1026593302236 (cited on page 55).

[DMM08] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. *From Gestalt Theory to Image Analysis: A Probabilistic Approach*. Volume 34. Interdisciplinary Applied Mathematics. Springer, Heidelberg/Berlin, 2008. ISBN: 978-0-387-72635-9 (cited on page 55).

[EL99] Alexei A. Efros and Thomas K. Leung. “Texture Synthesis by Non-Parametric Sampling.” In: *Proc. of IEEE International Conference on Computer Vision ICCV*. Sept. 1999, pages 1033–1038. DOI: 10.1109/ICCV.1999.790383 (cited on pages 2, 7).

[EW76] Michael P. Ekstrom and John W. Woods. "Two-Dimensional Spectral Factorization with Applications in Recursive Digital Filtering." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.2 (Apr. 1976), pages 115–128. DOI: 10.1109/TASSP.1976.1162785 (cited on pages 23–26, 82).

[Fad+10] M. Jalal Fadili, Jean-Luc Starck, Jérôme Bobin, and Yassir Moudden. "Image Decomposition and Separation Using Sparse Representations: An Overview." In: *Proceedings of the IEEE* 98.6 (June 2010), pages 983–994. DOI: 10.1109/JPROC.2009.2024776 (cited on page 56).

[FB11] Christian Feldmann and Johannes Ballé. "Improved Entropy Coding for Component-Based Image Coding." In: *Proc. of IEEE International Conference on Image Processing ICIP '11*. (Bruxelles, Belgium). Sept. 2011. ISBN: 978-1-4577-1304-0. DOI: 10.1109/ICIP.2011.6116365 (cited on pages 37, 38, 81, 86).

[Fie87] David J. Field. "Relations Between the Statistics of Natural Images and the Response Properties of Cortical Cells." In: *Journal of the Optical Society of America A* 4.12 (Dec. 1987), pages 2379–2394. DOI: 10.1364/JOSAA.4.002379 (cited on pages 40, 42, 47, 48, 50, 54, 61).

[Gei08] Wilson S. Geisler. "Visual Perception and the Statistical Properties of Natural Scenes." In: *Annual Reviews of Psychology* 59 (Jan. 2008), pages 167–192. DOI: 10.1146/annurev.psych.58.110405.085632 (cited on page 61).

[Geo+07] Mark A. Georgeson, Keith A. May, Tom C. A. Freeman, and Gillian S. Hesse. "From filters to features: Scale–space analysis of edge and blur coding in human vision." In: *Journal of Vision* 7.13 (2007), pages 1–21. ISSN: 1534-7362. DOI: 10.1167/7.13.7 (cited on page 50).

[GF08] Daniel J. Graham and David J. Field. "Natural Images: Coding Efficiency." In: *Encyclopedia of Neuroscience*. Edited by Larry R. Squire. Volume 6. Elsevier, Nov. 2008, pages 19–27. ISBN: 978-0-08-045046-9. DOI: 10.1016/B978-008045046-9.00212-6 (cited on page 47).

[GG92] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Kluwer, 1992. ISBN: 978-0-7923-9181-4 (cited on pages 1, 81).

[GGM11] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. "Random Phase Textures: Theory and Synthesis." In: *IEEE Transactions on Image Processing* 20.1 (Jan. 2011), pages 257–267. DOI: 10.1109/TIP.2010.2052822 (cited on pages 38, 58).

[HCB03] Paul R. Hill, C. Nishan Canagarajah, and David R. Bull. "Image Segmentation Using a Texture Gradient Based Watershed Transform." In: *IEEE Transactions on Image Processing* 12.12 (Dec. 2003), pages 1618–1633. DOI: 10.1109/TIP.2003.819311 (cited on page 55).

[Her+01] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. "Image Analogies." In: *Proc. of International Conference on Computer Graphics and Interactive Techniques SIGGRAPH*. 2001, pages 327–340. DOI: 10.1145/383259.383295 (cited on page 7).

[HEVCDr12] Benjamin Bross, Woo-Jin Han, Gary J. Sullivan, Jens-Rainer Ohm, and Thomas Wiegand. *High Efficiency Video Coding (HEVC) text specification draft 6*. Doc. JCTVC-H1003. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, 2012 (cited on pages 83, 86).

[HG95] Thomas E. Hall and Georgios B. Giannakis. "Bispectral Analysis and Model Validation of Texture Images." In: *IEEE Transactions on Image Processing* 4.7 (July 1995), pages 996–1009. DOI: 10.1109/83.392340 (cited on page 58).

[HHL09] Sven J. Hammarling, Nicholas J. Higham, and Craig Lucas. "LAPACK-Style Codes for Pivoted Cholesky and QR Updating." In: *Lecture Notes in Computer Science*. Volume 4699. Springer, Heidelberg/Berlin, 2009, pages 137–146. ISBN: 978-3-540-75754-2. DOI: 10.1007/978-3-540-75755-9_17 (cited on page 29).

[Hig90] Nicholas J. Higham. "Analysis of the Cholesky Decomposition of a Semi-Definite Matrix." In: *Reliable Numerical Computation*. Edited by M. G. Cox and Sven J. Hammarling. Oxford University Press, 1990, pages 161–185. ISBN: 978-0-19-853564-5 (cited on page 29).

[Hin90] Melvin J. Hinich. "Detecting a Transient Signal by Bispectral Analysis." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38.7 (July 1990), pages 1277–1283. DOI: 10.1109/29.57556 (cited on pages 16, 58).

[ImCoIn] *The New Test Images*. URL: http://www.imagecompression.info/test_images/ (visited on 03/11/2012) (cited on pages 94, 104).

[Ita75] Fumitada Itakura. "Minimum Prediction Residual Principle Applied to Speech Recognition." In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 23.1 (Feb. 1975), pages 67–72. DOI: 10.1109/TASSP.1975.1162641 (cited on page 33).

[J2K] *ITU-T Rec. T.800 & ISO/IEC 15444-1: JPEG 2000 Image Coding System: Core Coding System* (cited on page 1).

[Jai89] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, 1989. ISBN: 978-0-13-336165-0 (cited on pages 23, 30).

[Jul81] Béla Julesz. "A Theory of Preattentive Texture Discrimination Based on First-Order Statistics of Textons." In: *Biological Cybernetics* 41 (1981), pages 131–138. DOI: 10.1007/BF00335367 (cited on page 47).

[Jul91] Béla Julesz. "Early vision and focal attention." In: *Reviews of Modern Physics* 63.3 (July 1991), pages 735–775. DOI: 10.1103/RevModPhys.63.735 (cited on page 47).

[Kay93] Steven M. Kay. *Fundamentals of Statistical Signal Processing*. Volume I: *Estimation Theory*. Prentice-Hall, Englewood Cliffs, 1993. ISBN: 978-0-13-345711-7 (cited on page 18).

[Kodak] *CIPR Still Images: Kodak*. URL: http://www.cipr.rpi.edu/resource/stills/kodak.html (visited on 06/31/2009) (cited on pages 86, 105).

[Kov99a] Peter Kovesi. “Image Features from Phase Congruency.” In: *Videre: Journal of Computer Vision Research* 1.3 (1999), pages 2–26. ISSN: 1089-2788 (cited on pages 50, 53, 61).

[Kov99b] Peter Kovesi. “Phase Preserving Denoising of Images.” In: *The Australian Pattern Recognition Society Conference: DICTA ’99*. Dec. 1999, pages 212–217. ISBN: 978-1-86342-838-5 (cited on page 55).

[Kra82] Steven George Krantz. *Function Theory of Several Complex Variables*. Wiley, New York, 1982. ISBN: 978-0-471-09324-4 (cited on page 24).

[Kwa+03] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. “Graphcut Textures: Image and Video Synthesis using Graph Cuts.” In: *Proc. of International Conference on Computer Graphics and Interactive Techniques SIGGRAPH*. July 2003, pages 277–286. DOI: 10.1145/1201775.882264 (cited on page 7).

[Kwa+05] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. “Texture Optimization for Example-Based Synthesis.” In: *Proc. of International Conference on Computer Graphics and Interactive Techniques SIGGRAPH*. 2005, pages 795–802. DOI: 10.1145/1073204.1073263 (cited on page 7).

[Mar78] Thomas L. Marzetta. “A Linear Prediction Approach to Two-Dimensional Spectral Factorization and Spectral Estimation.” PhD thesis. Cambridge: Massachusetts Institute of Technology, Feb. 1978 (cited on page 24).

[Mar80] Thomas L. Marzetta. “Two-Dimensional Linear Prediction: Autocorrelation Arrays, Minimum-Phase Prediction Error Filters, and Reflection Coefficient Arrays.” In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 28.6 (Dec. 1980), pages 725–733. DOI: 10.1109/TASSP.1980.1163468 (cited on pages 24, 25).

[Mar87] S. Lawrence Marple Jr. *Digital Spectral Analysis*. Prentice-Hall, Englewood Cliffs, 1987. ISBN: 978-0-13-214149-9 (cited on page 26).

[MMY06] Ricardo A. Maronna, R. Douglas Martin, and Victor J. Yohai. *Robust Statistics: Theory and Methods*. Wiley, New York, 2006. ISBN: 978-0-470-01092-1 (cited on page 19).

[MO87] Maria C. Morrone and Robyn A. Owens. “Feature Detection from Local Energy.” In: *Pattern Recognition Letters* 6.5 (Dec. 1987), pages 303–313. DOI: 10.1016/0167-8655(87)90013-4 (cited on pages 49, 53).

[Mor+86] Maria Concetta Morrone, David C. Burr, J. Ross, and Robyn A. Owens. “Mach Bands Are Phase Dependent.” In: *Nature* 324 (Nov. 1986), pages 250–253. DOI: 10.1038/324250a0 (cited on page 49).

[MSM84] Petros A. Maragos, Ronald W. Schafer, and Russell M. Mersereau. “Two-Dimensional Linear Prediction and Its Application to Adaptive Predictive Coding of Images.” In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 32.6 (Dec. 1984), pages 1213–1229. DOI: 10.1109/TASSP.1984.1164463 (cited on page 30).

[MSW03] Detlev Marpe, Heiko Schwarz, and Thomas Wiegand. “Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 13.7 (July 2003), pages 620–636. DOI: 10.1109/TCSVT.2003.815173 (cited on pages 83, 85).

[MZ93] Stéphane G. Mallat and Zhifeng Zhang. “Matching Pursuits with Time–Frequency Dictionaries.” In: *IEEE Transactions on Signal Processing* 41.12 (Dec. 1993), pages 3397–3415. DOI: 10.1109/78.258082 (cited on page 56).

[NBW09] Patrick Ndjiki-Nya, David R. Bull, and Thomas Wiegand. “Perception-Oriented Video Coding Based on Texture Analysis and Synthesis.” In: *Proc. of IEEE International Conference on Image Processing ICIP ’09*. 2009, pages 2273–2276. DOI: 10.1109/ICIP.2009.5414386 (cited on page 2).

[NHW07] Patrick Ndjiki-Nya, Tobias Hinz, and Thomas Wiegand. “Generic and Robust Video Coding with Texture Analysis and Synthesis.” In: *Proc. of IEEE International Conference on Multimedia and Expo ICME*. July 2007, pages 1447–1450. DOI: 10.1109/ICME.2007.4284933 (cited on page 2).

[NP93] Chrysostomos L. Nikias and Athina P. Petropulu. *Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework*. Prentice-Hall, Englewood Cliffs, 1993. ISBN: 978-0-13-678210-0 (cited on pages 9, 10, 13, 15).

[OB05] Robert J. O’Callaghan and David R. Bull. “Combined Morphological-Spectral Unsupervised Image Segmentation.” In: *IEEE Transactions on Image Processing* 14.1 (Jan. 2005), pages 49–62. DOI: 10.1109/TIP.2004.838695 (cited on pages 2, 55).

[OF96] Bruno Olshausen and David J. Field. “Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images.” In: *Nature* 381 (June 1996), pages 607–609. DOI: 10.1038/381607a0 (cited on pages 47, 48, 56).

[OL10] Jens-Rainer Ohm and Hans Dieter Lüke. *Signalübertragung*. 11th edition. Springer, Heidelberg/Berlin, 2010. ISBN: 978-3-642-10199-1 (cited on pages 12, 13).

[OS75] Alan V. Oppenheim and Ronald W. Schafer. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, 1975. ISBN: 978-0-13-214635-7 (cited on page 23).

[Per95] Pietro Perona. “Deformable Kernels for Early Vision.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.5 (May 1995), pages 488–499. DOI: 10.1109/34.391394 (cited on page 61).

[PM90] Pietro Perona and Jitendra Malik. “Detecting and Localizing Edges Composed of Steps, Peaks and Roofs.” In: *Proc. of IEEE International Conference on Computer Vision ICCV ’90*. Dec. 1990. DOI: 10.1109/ICCV.1990.139492 (cited on page 53).

[PS00a] Ted Painter and Andreas Spanias. “Perceptual Coding of Digital Audio.” In: *Proceedings of the IEEE* 88.4 (Apr. 2000), pages 451–513. DOI: 10.1109/5.842996 (cited on page 1).

[PS00b] Javier Portilla and Eero P. Simoncelli. “A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients.” In: *International Journal of Computer Vision* 40.1 (Oct. 2000), pages 49–71. DOI: 10.1023/A:1026553619983 (cited on page 58).

[QBC88] Schuyler R. Quackenbush, Thomas P. Barnwell III, and Mark A. Clements. *Objective Measures of Speech Quality*. Prentice-Hall, Englewood Cliffs, 1988. ISBN: 978-0-13-629056-8 (cited on page 34).

[RH05] Håvard Rue and Leonhard Held. *Gaussian Markov Random Fields*. Monographs on Statistics and Applied Probability 104. Chapman & Hall/CRC, 2005. ISBN: 978-1-58488-432-3 (cited on pages 16–18).

[Sam+09] Mehul P. Sampat, Zhou Wang, Shalini Gupta, Alan Conrad Bovik, and Mia K. Markey. “Complex Wavelet Structural Similarity: A New Image Similarity Index.” In: *IEEE Transactions on Image Processing* 18.11 (Nov. 2009), pages 2385–2401. DOI: 10.1109/TIP.2009.2025923 (cited on page 34).

[Ser+05] Thomas Serre, Minjoon Kouh, Charles Cadieu, Ulf Knoblich, Gabriel Kreiman, and Tomaso Poggio. *A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex*. Computer Science and Artificial Intelligence Laboratory Technical Report MIT-CSAIL-TR-2005-082. Massachusetts Institute of Technology, Dec. 2005 (cited on pages 1, 47–49, 56).

[SF95] Eero P. Simoncelli and William T. Freeman. “The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation.” In: *Proc. of IEEE International Conference on Image Processing ICIP ’95*. Volume 3. Oct. 1995, pages 444–447. DOI: 10.1109/ICIP.1995.537667 (cited on page 61).

[Sim+92] Eero P. Simoncelli, William T. Freeman, Edward H. Adelson, and David J. Heeger. “Shiftable Multiscale Transforms.” In: *IEEE Transactions on Information Theory* 38.2 (Mar. 1992), pages 587–607. DOI: 10.1109/18.119725 (cited on pages 37, 61, 63).

[SJ84] Frank K. Soong and Biing-Hwang Juang. “Line Spectrum Pair (LSP) and Speech Data Compression.” In: *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP ’84*. Mar. 1984. DOI: 10.1109/ICASSP.1984.1172448 (cited on page 23).

[SM05] Petre Stoica and Randolph Moses. *Spectral Analysis of Signals*. Prentice-Hall, Englewood Cliffs, Apr. 2005. ISBN: 978-0-13-113956-5 (cited on page 26).

[SM97] Petre Stoica and Randolph Moses. *Introduction to Spectral Analysis*. Prentice-Hall, Englewood Cliffs, Feb. 1997. ISBN: 978-0-13-258419-7 (cited on page 26).

[SO87] Alan Stuart and J. Keith Ord. *Kendall’s Advanced Theory of Statistics*. 5th edition. Volume 1. Charles Griffin & Co., London, 1987. ISBN: 978-0-85264-285-6 (cited on pages 58, 64).

[Stu27] Student. “Errors of Routine Analysis.” In: *Biometrika* 19.1/2 (July 1927), pages 151–164. DOI: 10.1093/biomet/19.1-2.151 (cited on page 11).

[The92] Charles W. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall, Englewood Cliffs, 1992. ISBN: 978-0-13-852112-7 (cited on page 29).

[VisTex] *Vision Texture*. URL: `http://vismod.media.mit.edu/vismod/imagery/VisionTexture/` (visited on 03/11/2012) (cited on pages 101, 102).

[VO90] Svetha Venkatesh and Robyn A. Owens. "On the Classification of Image Features." In: *Pattern Recognition Letters* 11.5 (May 1990), pages 339–349. DOI: 10.1016/0167-8655(90)90043-2 (cited on page 49).

[Wan+04] Zhou Wang, Alan Conrad Bovik, H. R. Sheikh, and Eero P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity." In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pages 600–612. DOI: 10.1109/TIP.2003.819861 (cited on pages 3, 34, 59).

[WL00] Li-Yi Wei and Marc Levoy. "Fast Texture Synthesis using Tree-Structured Vector Quantization." In: *Proc. of International Conference on Computer Graphics and Interactive Techniques SIGGRAPH*. July 2000, pages 479–488. DOI: 10.1145/344779.345009 (cited on page 7).

[ZB11] Fang Zhang and David R. Bull. "A Parametric Framework for Video Compression Using Region-Based Texture Models." In: *IEEE Journal of Selected Topics in Signal Processing* 5.7 (Nov. 2011), pages 1378–1392. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2011.2165201 (cited on pages 2, 3).

[Zha+08] Xiaonan Zhao, Matthew G. Reyes, Thrasyvoulos N. Pappas, and David L. Neuhoff. "Structural Texture Similarity Metrics for Retrieval Applications." In: *Proc. of IEEE International Conference on Image Processing ICIP '08*. (San Diego, CA, USA). Oct. 2008, pages 1196–1199. DOI: 10.1109/ICIP.2008.4711975 (cited on page 35).