Image Models for Segmentation and Recognition

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von
Dipl.-Inf. Matthias Heiler
aus Heidelberg.

Mannheim, 2006

Dekan:	Professor Dr. Matthias Krause, Universität Mannheim
Referent:	Professor Dr. Christoph Schnörr, Universität Mannheim
Korreferent:	Prof. Dr. Thomas Hofmann, Technische Universität Darmstadt
Tag der mündlichen Prüfung:	26. Juli 2006

Abstract

We examine image models for segmentation and classification that are based (i) on the statistical properties of natural images and (ii) on non-negative matrix or tensor codes.

Regarding (i) we derive a parametric framework for variational image segmentation. Using a model for the filter response statistics of natural images we build a sound probabilistic distance measure that drives level sets toward meaningful segmentations of complex textures and natural scenes. We show that the approach can be generalized from binary image segmentation to multiple image regions and is suitable for fast greedy optimization.

Regarding (ii) we use results from global deterministic optimization to obtain fast and practical algorithms for non-negative matrix and tensor factorization with sparsity constraints. Such problems were previously optimized using variations of projected gradient descent — a procedure that can be inefficient and difficult to generalize. In contrast, our approach uses highly efficient solvers from convex optimization to solve a sequence of quadratic or second-order conic programs. We show that this offers benefits in terms of efficiency and, in particular, extensibility: Our procedures are very easy to augment by additional convex constraints. We give numerous examples where such additional constraints yield improved results in image processing and recognition.

Zusammenfassung

Wir untersuchen Modelle zur Bild-Segmentierung und -Klassifikation, welche (i) auf der Statistik natürlicher Bilder sowie (ii) auf nicht-negativen Matrix- und Tensor-Codes basieren.

Bezüglich natürlicher Bildstatistiken (i) schlagen wir ein parametrisches Modell zur Segmentierung mit Variationsansätzen vor. Dabei leiten wir ein probabilistisches Abstandsmaß für Bildregionen aus der Filterantwort-Statistik natürlicher Bilder her. Integriert in einen Level-Set-Ansatz liefert dieses Abstandsmaß nichttriviale Segmentierungen komplexer Texturen und natürlicher Bilder. In einem Folgeschritt erweitern wir ein erstes binäres Segmentationsverfahren für den Mehrklassenfall und adaptieren es so, daß schnelle Optimierung möglich wird.

Bezüglich der nicht-negativen Codes (ii) verwenden wir Ergebnisse der deterministischen globalen Optimierung, um schnelle und praktikable Algorithmen für nicht-negative Matrix- und Tensor-Faktorisierung unter konkaven Nebenbedingungen, welche dünn besetzte Ergebnisse erzwingen, zu ermöglichen. Solche Probleme wurden bislang mit Varianten des Gradientenabstiegs-Verfahrens gelöst. Nachteil dieses Verfahrens ist, daß es langsam und schwierig zu verallgemeinern sein kann. Das in dieser Arbeit entwickelte Verfahren hingegen verwendet hocheffiziente Ansätze aus der mathematischen Programmierung und löst eine Folge von quadratischen bzw. konischen Optimierungsproblemen. Wir weisen nach, daß dieser Ansatz Vorteile bezüglich Rechenaufwands und Erweiterbarkeit hat. Insbesondere sind unsere Ansätze sehr einfach um zusätzliche konvexe Nebenbedingungen erweiterbar. Zahlreiche Beispiele belegen, dass solche Erweiterungen zur besseren Ergebnissen bei Aufgaben der Bildverarbeitung und -erkennung führen können.

Acknowledgments

I thank my supervisor Prof. Christoph Schnörr for introducing me to the field of computer vision and for giving me just the right amount of guidance during my time in his group. As graduate student I got to know him as a role model not only scientifically, but also in terms of integrity and straightforwardness. I also thank Prof. Thomas Hofmann for serving as an external referee of this thesis.

During my time in Mannheim I benefited from close interaction with the computer vision group, guests, and members of the department of mathematics. In particular, I thank Prof. Luis Álvarez for pointing me to literature on natural image statistics, Prof. Attila Kuba for introducing me to the SPECT factorization problem, and Prof. Jürgen Potthoff for his assistance with stable distributions. I am grateful to my fellow graduate students, namely to Martin Bergtholdt and Timo Kohlberger for many inspiring discussions, to Paul Ruhnau and Florian Becker for proofreading this thesis, to Stefan Weber and Thomas Schüle for sharing their experience with convex-concave programs, and to Christian Schellewald and Christian Gosch for not giving up on our computer network. I thank Jens Keuchel for the close and pleasant collaboration and Prof. Daniel Cremers for many fruitful discussions on computer vision and the computer vision community.

Finally, I thank my friends and family who did not complain when I was on conferences or basically disappeared before important deadlines: Your friendship, love, and patience pulled me through the more intense parts of the dissertation project.

Contents

Chapter 1

Introduction

In this chapter we give a general but brief overview on image segmentation and our contributions in this thesis.

1.1 Motivation

Image segmentation is the task of *partitioning* an image $I : \Omega \to \mathbb{R}^n$ into disjoint regions Ω_i, $i = 1, \ldots, k$, so that the Ω_i are visually distinct.

As a computational problem, image segmentation is interesting for at least three reasons: First, there is a multitude of potential applications. Medical image processing comes to mind, where doctors measure the size of organs and tissues using images obtained from their patients. Multimedia applications using image or video encoding can benefit from robust segmentation results, leading to more efficient image transfer and storage. In machine vision, image processing is usually a costly operation that should nevertheless perform in real time. An efficient segmentation algorithm can help to reduce the amount of visual information that needs to be processed.

Second, image segmentation is interesting because it serves as a testbed for ideas from other fields such as machine learning and pattern recognition, physics, or engineering. For instance, in its simplest form, image segmentation can be regarded as a clustering problem. Clustering algorithms can often be used in an image segmentation framework, and, conversely, algorithms developed for image processing can be very successful in clustering problems as well (Section 4).

Finally, image segmentation is interesting because it helps us to understand how we visually decipher the world: It is no coincidence that the objective formulated at the beginning of this section sounds so fuzzy. What, actually, constitutes a "visually distinct" region? In a sense, this is the core question to be solved, and historically, there was and still is a fruitful exchange of ideas between computational scientists, psychologists, and neural researchers [Gab46, Mar80, Dau85, Jul62, Tve77].

So by studying image segmentation we open a large field of important applications, we advance pattern recognition research, and have the chance to learn something about a fundamental problem of visual information processing.

1.1.1 Representation of Images

The unifying question behind the work in this thesis is that of image representation: *What is an image and how should it be modeled in the computer?*

In the past, various preliminary answers have been given, some more sophisticated than others. The two classes of image models we will be mostly interested in are based on parametric models of natural image statistics and the nonparametric statistics of Markov random fields.

1.1.2 Models from Natural Image Statistics

One approach toward image modeling is to utilize the statistics of natural images: Since non-trivial images are not just random collections of pixels in an array but pictures of a highly structured world their statistics are clearly non-trivial (cf. Section 3.1.3, [GK99, LPM03]). Analytical forms of filter responses of natural images have been observed very early [Kre52, RG83, HM99] and were frequently used in image processing applications [Mal98, JCF95, LR97, BS99].

For the slightly simplified case where images are modeled as superpositions of *translucent* objects the emergence of non-trivial statistics has been studied analytically [MG01, SLSZ03]. For occlusion models simulations of the *dead leaves model* have shown excellent correspondences [ÁGM99, LMH01]. We refer to [SLSZ03] for a survey on the field.

1.1.3 Models from Non-Negative Matrix Factorization

Above models are usually based on the statistics of pixel values and filter responses at individual image locations. The reason is that this yields one-dimensional histograms that are efficient to store and compute. However, in recent years models based on *image patches* have also become popular: Starting with MRF statistics collected from image samples for texture synthesis [EL99] and classification [PLL98, VZ03], patch-based approaches were used for segmentation [BU02, BSU04, BU04, AAR04], for object detection [Low99, WWP00, UVNS02, FPZ03, SK04, DS04b, JT05, NJT06], image categorization [SRE$^+$05], or to build priors for image-based rendering [FWZ05]. Interestingly, the best patch-based recognition algorithms offer state of the art performance even when object geometry or shape is not accounted for [DS04b, JT05].

Since image patches are high-dimensional and usually many patches need to be sampled, data compression techniques are important to reduce computational load and make inference possible. For instance, PCA was successful in providing improved scale invariant

feature description [KS04] and has been used for patch-based recognition [FPZ03]. Clustering is a critical step to simplify the statistical estimation problem [WWP00, SRE+05] and, especially when image patches are sampled densely, significantly influences the performance [JT05]. In unsupervised image categorization there is a third step where latent variables are estimated [SRE+05].

Interestingly, *all three steps*: dimensionality-reduction, clustering, and semantic analysis *fit into the same framework of non-negative matrix factorization (NMF).*

1. *Dimensionality reduction* was one of the tasks that motivated NMF. Originally developed for scientific data sets [SI89, PT94], it was soon applied to images [LS99, GSV02, WJHT05, BP04].

2. *Probabilistic and spectral clustering* was recently shown to fit into the NMF framework as well [DHS05, ZS05].

3. Along a similar line, a popular approach for *latent semantic analysis* is based on factorizing a matrix of (relative) frequencies [DDL+90, Hof99]. Since frequencies are non-negative, NMF offers benefits over classical approaches based on singular value decomposition [XLG03, SBPP06].

Thus, in the context of patch-based image models NMF can answer some key questions and may function as a building block to use for sophisticated implementations. When it comes to exercising precise control over a non-negative matrix factorization, *sparsity* is important: Sparse image codes, for instance, seem to be better suited for learning and have a strong tendency to separate images into parts [Ols96, Hoy04], yielding semantically meaningful image bases (Section 4.4). The optimization problems associated with sparse NMF are computationally intricate, which motivates our study of sparsity-constrained NMF problems.

1.1.4 Non-Negative Tensor Factorization Models

For image modeling matrix factorizations can be inefficient since they rely on *vectorizing* image data, a process which makes it difficult to capture spatial relationships between neighboring image locations. Tensor factorization has been proposed to address this problem. In tensor factorization base images are not vectors but outer products of vectors, i.e., real two-dimensional entities [WW01]. Depending on the specific class of images under consideration this can yield significant benefits: Since spatial correlations along the x- and y-axis are modeled explicitly, more efficient compression and improved performance of machine learning algorithms can be observed [HPS05]. Furthermore, higher-order decompositions are useful for modeling video or in clustering [DHS05, ZS05].

1.2 Overview

After this introduction we briefly present some results from mathematical programming and variational analysis (Chapter 2). Our exposure is concise and mainly aims at making this text self-contained. In Chapter 3 we present a novel parametric image model and derive segmentation algorithms for it. In Chapter 4 we approach the image segmentation problem from a different point of view and present new solvers for the non-negative matrix factorization (NMF) problem. Based on NMF, we develop image codes that yield powerful algorithms for image segmentation and categorization. In Chapter 5 we discuss the more general case when high-dimensional tensor factorizations are sought. We put our results to use in Chapter 6 where we tackle some intricate applications. Chapter 7 concludes the text with a brief outlook.

1.3 Prior Work

The work in this thesis is based on two complementary approaches in computer vision: One has its foundations in variational calculus and PDE-based image segmentation, the other in mathematical programming and non-negative models for images.

Concerning the variational approach we follow the *region-based* methods pioneered by Mumford and Shah [MS89] and made practical, e.g., by Chan and Vese [CV01] within a level set framework [TD79, OS88] or by Deriche and Paragios [PD02] in the geodesic active contour calculus [CKS97, KKO$^+$95]. Here, image regions are approximated by *piecewise smooth* functions, and discontinuities in the approximation correspond to boundaries between adjacent regions.

The statistical features we use in a region-competition framework pioneered by Zhu and Yuille [ZY96] are based on natural image statistics and filter response histograms which are popular for image modeling [ZWM97, WZL00, PHB99, LW03]. The corresponding parametric framework has so far only been used for image [Sim97, Mal98] and texture modeling [PS00]. Slightly more complex distributions have also been employed for clutter detection [SLG02].

The second part of this thesis builds on work on *non-negative matrix factorization (NMF)* which was first used to model transport processes in the atmosphere [SI89, PT94] and introduced to the computer vision and machine learning community in a seminal paper by Lee and Seung [LS99]. Several attempts have been made to further adapt this model to computer vision applications [Hoy02, GSV02, WJHT04, BP04]. An important development was the introduction of sparseness constraints [Hoy04] that allow precise control over sparseness in any given factorization. Akin to NMF, tensor factorization models where introduced to computer vision relatively recently [WW01, HPS05].

1.4 Contributions

Our main contributions are the following:

1. We develop an *image segmentation algorithm for natural images* that uses a statistically efficient parametric representation of natural images.
2. We present a *fast and mathematically sound optimization algorithm for NMF* problems that makes sparsity-constrained matrix factorization practical.
3. We *extend the previously used NMF models* by integrating constraints for prior knowledge, transformation invariance, missing values, or signed data. These extensions are important in applications where class labels are available for training, data is missing or corrupted, or sparsity-constrained PCA is needed.
4. We propose *sparsity-controlled non-negative tensor (NTF) models* and develop a solver for the corresponding optimization problem [WW01, HPS05]. This allows for the first time to compute tensor factorization with fully controlled sparseness.

We note that some results in this thesis have previously been published at conferences and in journals [HS03, HS05a, HS05b, HS05c, HS06a, HS06b].

1.5 Notation

$\mathbb{R}_+$	non-negative real numbers
$\mathbb{R}^n$	n-dimensional real vector space
$x^\top$	transpose of x
$\langle x, y \rangle$	inner product between x and y
$\otimes$	Kronecker's matrix product
$\odot$	element-wise matrix product: $(A \odot B)_{ij} = A_{ij} B_{ij}$
$\oslash$	element-wise matrix quotient: $(A \oslash B)_{ij} = A_{ij} / B_{ij}$
$\mathrm{vec}(M)$	concatenation of the columns of matrix M
$\mathrm{tr}(M)$	trace of matrix M: $\mathrm{tr}(M) = \sum_i M_{ii}$
$(M)_+$	remove negative entries: $((M)_+)_{ij} = \max(0, M_{ij})$
$\|x\|$	ℓ_2 norm of vector x: $\|x\|^2 = \langle x, x \rangle$
$\|x\|_p$	ℓ_p norm of vector x: $\|x\|_p = \sqrt[p]{\sum_i x_i^p}$
$\|M\|_F$	Frobenius norm of matrix M: $\|M\|_F = \sqrt{\mathrm{tr}(M^\top M)}$
e	vector of ones: $e = (1, \ldots, 1)^\top$
e_i	i-th unit column vector: $e_i = 1,\ j \neq i \Rightarrow e_j = 0$
E_{ij}	ij unit matrix: $E_{ij} = e_i e_j^\top$
$E^{m,n}$	$m \times n$ matrix with all entries equal to one
$M_{\bullet i}$	i-th column of matrix M
$M_{i\bullet}$	i-th row of matrix M
Ω	image domain: $\Omega \subset \mathbb{R}^2$ or $\Omega \subset \mathbb{N}^2$, bounded and open
$\mathcal{I}$	set of images: $\mathcal{I} = 2^{\Omega \to \mathbb{R}}$
I	individual image: $\mathcal{I} \ni I : \Omega \to \mathbb{R}$
ϕ	level set function: $\Omega \to \mathbb{R}$
V	non-negative $m \times n$ matrix of vectorized images
W	non-negative $m \times r$ matrix of basis functions
H	non-negative $r \times n$ matrix of coefficients
$\mathcal{L}^{n+1}$	n-dimensional second order cone

Chapter 2

Variational Models and Mathematical Programming

Throughout this text we will use results from variational analysis on the one hand and mathematical programming on the other. For easier reference and accessibility we summarize these in this chapter.

2.1 Variational Image Segmentation

In this section we present the central ideas of variational image segmentation as far as necessary to follow the results in this work. For a more complete overview and rigid mathematical treatment we refer to the excellent textbooks available such as [AK00, Sap01] or classics as [GF63]. We recommend [SZ91, DZ01] as a thorough treatment on shape optimization.

2.1.1 Level Sets

An important question for any segmentation algorithm is how to represent the *contours*, $\partial\Omega_i$, of image regions. Modeling contours as one-dimensional objects has inherent benefits in terms of memory utilization and computational efficiency. This is important when memory is scarce and CPU time expensive.

However, contour-based region representations are often difficult to handle from an implementation point of view: Usually, one will represent contours by some parametric curve such as a spline. Then, during curve evolution one has to ensure that the spline control points remain approximately evenly spaced, that singularities, such as merging or splitting of contours, are treated appropriately, that smoothness conditions are not violated, that information from the curve is propagated accurately to control points, etc. This can lead to numerically fragile solutions that are difficult to program and to use.

An important development to alleviate this problem is due to Osher and Sethian[1] [OS88]: They suggest to represent contours as *level sets* of surfaces, i.e., an one-dimensional object is described in terms of a two-dimensional entity. While this seems counter intuitive at first, the approach offers some important advantages such as greater numerical stability, invariance under topological changes of the contour, and natural generalization to multiple regions and higher dimensional objects.

Assume, a contour evolution $\mathcal{C}(t,q)$ is specified for time t and parametrization q in the following way: Let $N(t,q)$ denote the normal to $\mathcal{C}$ and let $F(t,q)$ be a flow describing how $\mathcal{C}$ evolves over time. Note that we are not interested in tangential flows which merely change the parametrization of the curve without affecting its geometry. Then, the evolution is given by [AK00]:

$$\begin{cases} \partial\mathcal{C}/\partial t = F(t,q)\cdot N(t,q) \\ \mathcal{C}(0,q) = \mathcal{C}_0(q). \end{cases} \tag{2.1}$$

Assume that at time t the curve $\mathcal{C}$ corresponds to the zero-level of a sufficiently smooth level set function $\phi_t : \mathbb{R}_+ \times \Omega \to \mathbb{R}$, i.e., $\mathcal{C}(t,\cdot) = \{x \in \Omega : \phi_t(x) = 0\}$. The question is then how to change ϕ_t in order to obtain curve evolution (2.1).

Assuming ϕ_t is differentiable we can insert $\mathcal{C}$ and obtain with (2.1)

$$\begin{aligned} & \phi_t(\mathcal{C}(t,q)) = 0 \\ \overset{\partial/\partial t}{\Rightarrow} \quad & \frac{\partial\phi}{\partial t} + \langle\nabla\phi, \frac{\partial\mathcal{C}}{\partial t}\rangle = 0 \\ \overset{(2.1)}{\Rightarrow} \quad & \frac{\partial\phi}{\partial t} + \langle\nabla\phi, F\cdot N\rangle = 0. \end{aligned} \tag{2.2}$$

In the level set formulation the curve normal is just the normalized and sign-corrected gradient of the level set function. By convention, ϕ is negative in the interior of $\mathcal{C}$ and positive outside. Then $N = -\frac{\nabla\phi}{|\nabla\phi|}$ and

$$\begin{aligned} & \partial\phi/\partial t = \langle\nabla\phi, F\cdot\frac{\nabla\phi}{|\nabla\phi|}\rangle \\ \Rightarrow \quad & \partial\phi/\partial t = F\,|\nabla\phi|\,. \end{aligned} \tag{2.3}$$

This is the result we sought: It describes how to express an evolution of $\mathcal{C}$ in terms of ϕ. With the appropriate initialization to a signed distance function $\bar{d}$ and Neumann boundary conditions the evolution is summarized in the *Hamilton-Jacobi equation*

$$\begin{cases} \partial\phi/\partial t = F\,|\nabla\phi| \\ \phi_0(x) = \bar{d}(x,\mathcal{C}_0) \\ \partial\phi/\partial N = 0 \qquad \forall x \in \partial\Omega. \end{cases} \tag{2.4}$$

Note that although derived only for $\mathcal{C} \subset \Omega$ the PDE is extended on Ω or on an neighborhood around $\mathcal{C}$ without difficulties.

[1] Similar ideas were developed independently in [TD79].

2.1.2 Region Based Segmentation

Classical *edge-based* algorithms examine the image locally, searching closed contours along regions where the image gradient ∇I is strong. Important representatives among these so-called *active contour models* are the *snake* model [KWT88] and its *geodesic* formulation [CKS97, KKO+95], or the *balloon* model [CC93].

Dually, *region-based* methods emerged where partitions of maximum homogeneity are sought. E.g., in the MUMFORD-SHAH *functional* [MS89] image regions are approximated by *piecewise smooth* functions. Discontinuities in the approximation correspond to boundaries between adjacent regions.

In this work we will adopt the region-based approach. Within this framework an image I is approximated by a piecewise-smooth function u, such that the following energy functional is minimized [MS89]:

$$E_{\mathrm{MS}}(u, \mathcal{C}) = \mu \cdot |\mathcal{C}| + \lambda \int_{\Omega} |I - u|^2 \, dx + \int_{\Omega \setminus \mathcal{C}} g(|\nabla u|) dx. \tag{2.5}$$

g is an arbitrary convex even function acting on the gradient magnitude of u. The first term ensures the length $|\mathcal{C}|$ of the contour $\mathcal{C}$ is minimized, the second term represents the approximation error, the last term ensures that the approximation is smooth except for the closed contour $\mathcal{C}$ where jumps can occur. Note that *no image gradient* is evaluated, i.e., the functional does not rely on detecting edge clues.

In the simplest case, u is assumed piecewise constant. Then, the last term can be omitted and an optimal u^* is simply [MS89]

$$u^*(x) = \begin{cases} \frac{1}{|\Omega_{\mathrm{in}}|} \int_{\Omega_{\mathrm{in}}} I(x) dx & x \text{ inside } \mathcal{C} \\ \frac{1}{|\Omega_{\mathrm{out}}|} \int_{\Omega_{\mathrm{out}}} I(x) dx & x \text{ outside } \mathcal{C}. \end{cases} \tag{2.6}$$

CHAN and VESE considered this model in a level set formulation [CV01]:

$$E(\phi) = \mu \int_{\Omega} |\nabla \phi| \, \delta(\phi) dx + \lambda_1 \int_{\Omega} |I - c_1|^2 \, H(\phi) dx + \lambda_2 \int_{\Omega} |I - c_2|^2 \, (1 - H(\phi)) dx. \tag{2.7}$$

Here, c_1 and c_2 are the mean gray values of the interior and the exterior region, respectively, and ϕ, H, and δ are the level set function, the unit step function and Dirac's delta. Thus, the first integral in (2.7) measures contour length while the other integrals measure how well c_1 and c_2 represent the interior and exterior region.

2.1.3 Shape Derivatives

To prepare for subsequent developments we slightly generalize (2.7) by introducing energy functions $k^{\mathrm{b}}(x)$, $k^{\mathrm{in}}(x, \phi)$, and $k^{\mathrm{out}}(x, \phi)$ representing the boundary, interior, and exterior energy contributions for a given level set function ϕ at location x. Note that

we allow the region energy functions to vary with ϕ, just as $c_{1/2}$ vary as the contour changes (eqn. (2.6)). Then, above energy functional reads [JBBA03]

$$E(\phi) = \int_\Omega k^{\mathrm{b}}(x) \left|\nabla\phi\right| \delta(\phi) dx + \lambda_1 \int_\Omega k^{\mathrm{out}}(x,\phi) H(\phi) dx + \lambda_2 \int_\Omega k^{\mathrm{in}}(x,\phi)(1-H(\phi)) dx. \tag{2.8}$$

Assuming E is sufficiently smooth, in particular k^{in} and k^{out} are weakly differentiable at least once [SZ91, Prop. 2.45], the variational update $\dot\phi = -\langle E'(\phi), \psi\rangle$, for all ψ sufficiently smooth, of this level set function reads[2]:

$$\frac{\partial E}{\partial \phi} = \frac{\partial}{\partial \phi}\left[\int_\Omega k^{\mathrm{b}} \left|\nabla\phi\right| \delta dx\right] + \int_\Omega (\lambda_1 k^{\mathrm{out}} - \lambda_2 k^{\mathrm{in}}) \delta\psi \; dx + \int_\Omega \left(\lambda_1 \frac{\partial k^{\mathrm{out}}}{\partial \phi} H + \lambda_2 \frac{\partial k^{\mathrm{in}}}{\partial \phi}(1-H)\right) \psi \; dx. \tag{2.9}$$

The third term, omitted in [CV01], originates formally from applying the product rule to the area integrals. It thus takes into account that k^{in} and k^{out} also depend on the level set function ϕ.

Let us take a closer look at the first term and develop it by the product rule:

$$\frac{\partial}{\partial \phi}\left[\int_\Omega k^{\mathrm{b}} \left|\nabla\phi\right| \delta dx\right] = \int_\Omega k^{\mathrm{b}} \left[\delta' \left|\nabla\phi\right| \psi + \delta \frac{\nabla\phi}{|\nabla\phi|} \nabla\psi\right] dx\,. \tag{2.10}$$

With Green's first theorem the second part becomes

$$\begin{aligned}\int_\Omega k^{\mathrm{b}} \delta \frac{\nabla\phi}{|\nabla\phi|} \nabla\psi \; dx &= -\int_\Omega \nabla\left(k^{\mathrm{b}} \delta \frac{\nabla\phi}{|\nabla\phi|}\right) \psi \; dx + \int_{\partial\Omega} \frac{k^{\mathrm{b}} \delta \partial\phi}{|\nabla\phi|\, \partial n} \psi ds \\ &= -\int_\Omega \left[\nabla k^{\mathrm{b}} \delta \frac{\nabla\phi}{|\nabla\phi|} + k^{\mathrm{b}} \nabla\delta \frac{\nabla\phi}{|\nabla\phi|} + k^{\mathrm{b}} \delta \nabla\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] \psi \; dx + \int_{\partial\Omega} \frac{k^{\mathrm{b}} \delta \partial\phi}{|\nabla\phi|\, \partial n} \psi \; ds \end{aligned} \tag{2.11}$$

which in connection with $\nabla\delta\frac{\nabla\phi}{|\nabla\phi|} = \delta' \left|\nabla\phi\right|$ and (2.10) yields

$$\frac{\partial}{\partial \phi}\left[\int_\Omega k^{\mathrm{b}} \left|\nabla\phi\right| \delta dx\right] = -\int_\Omega \left[\nabla k^{\mathrm{b}} \delta \frac{\nabla\phi}{|\nabla\phi|} + k^{\mathrm{b}} \delta \nabla\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] \psi \; dx + \int_{\partial\Omega} \frac{k^{\mathrm{b}} \delta \partial\phi}{|\nabla\phi|\, \partial n} \psi \; ds. \tag{2.12}$$

Note that we can replace each area integral containing the Dirac impulse into an integral over the region boundary $\mathcal{C} = \{x : \phi(x) = 0\}$:

$$\int_\Omega f(x,\phi)\delta(\phi) dx = \int_{\mathcal{C}} f(x,0) ds. \tag{2.13}$$

[2] To save horizontal space we abbreviate $\langle E'(\phi), \psi\rangle$ by $\partial E/\partial\phi$.

Hence we can write

$$\frac{\partial E}{\partial \phi} = \int_{\mathcal{C}\cap\partial\Omega} \frac{k^{\mathrm{b}}\partial\phi}{|\nabla\phi|\,\partial n}\psi \; ds + \int_{\mathcal{C}} \left[-\nabla k^{\mathrm{b}} \frac{\nabla\phi}{|\nabla\phi|} - k^{\mathrm{b}} \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \lambda_1 k^{\mathrm{out}} - \lambda_2 k^{\mathrm{in}} \right] \psi \; ds + \int_{\Omega} \left(\lambda_1 \frac{\partial k^{\mathrm{out}}}{\partial\phi} H + \lambda_2 \frac{\partial k^{\mathrm{in}}}{\partial\phi}(1-H) \right) \psi \; dx. \quad (2.14)$$

Assuming $\mathcal{C}\cap\partial\Omega = \emptyset$ and using the shorthands $n = \frac{\nabla\phi}{|\nabla\phi|}$ and $c = \operatorname{div}(n)$ we arrive at

$$\frac{\partial E}{\partial \phi} = \int_{\mathcal{C}} \left(-\nabla k^{\mathrm{b}} n - k^{\mathrm{b}} c + \lambda_1 k^{\mathrm{out}} - \lambda_2 k^{\mathrm{in}} \right) \psi \; ds + \int_{\Omega} \left(\lambda_1 \frac{\partial k^{\mathrm{out}}}{\partial\phi} H + \lambda_2 \frac{\partial k^{\mathrm{in}}}{\partial\phi}(1-H) \right) \psi \; dx \; . \quad (2.15)$$

This result from shape optimal design [SZ91] was first introduced to computer vision in [JBBA03]. Note also [Sch92] where shape derivatives of a more general class of functionals are used for motion estimation.

2.2 Convex Programming

From Chapter 4 on *convex programming* plays a dominant role in this text as the basic building blocks of some of our central algorithms are convex programs. Convex programming is attractive for multiple reasons: First, convex functions and sets share nice mathematical properties [Roc72]. In particular, there is a duality theory from which robust and highly efficient algorithms can be derived [Lue69, Min86, Wri97, Ber99, BV04]. Then, there exist a number of very efficient implementations that allow solving large-scale problems reliably and fast. In practice, they work almost as "black boxes", solving most problems without user intervention or additional parameter optimization. Since convex programs are also the core problems of important commercial applications, we can expect market forces to further motivate continuous, high-quality research in this field.

2.2.1 Quadratic Programming

Quadratic programming (QP) is concerned with minimizing a quadratic functional subject to linear constraints:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \frac{1}{2} x^\top Q x + f^\top x \\ \text{s.t.} \quad & Ax \leq b. \end{aligned} \quad (2.16)$$

Here, Q is required to be symmetric positive semi-definite, so that the problem is convex. Very efficient and robust solvers are available in software.

When Q is not positive semi-definite problem (2.16) is NP-hard [FV95]. In this case, only relatively small problem instances can be solved by methods of global optimization [HP95, HT96].

2.2.2 Second Order Conic Programs

Second order cone programming (SOCP) is a generalization of QP and concerned with minimizing a linear functional over convex quadratic cones intersected with affine sets [LVBL98]. The *second order cone* $\mathcal{L}^{n+1} \subset \mathbb{R}^{n+1}$ is the convex set

$$\mathcal{L}^{n+1} := \left\{ \begin{pmatrix} x \\ t \end{pmatrix} = (x_1, \ldots, x_n, t)^\top \,\middle|\, \|x\|_2 \leq t \right\} , \tag{2.17}$$

The problem of minimizing a linear objective function, subject to the constraints that several affine functions of the variables are contained in $\mathcal{L}^{n+1}$, is called a *second order cone program (SOCP)*:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f^\top x \\ \text{s.t.} \quad & \begin{pmatrix} A_i x + b_i \\ c_i^\top x + d_i \end{pmatrix} \in \mathcal{L}^{n+1} , \qquad i = 1, \ldots, m \end{aligned} \tag{2.18}$$

Note, that linear constraints and, in particular, the condition $x \in \mathbb{R}^n_+$ are important special cases. Our approach to sparsity-constrained factorizations, to be developed subsequently (Chapter 4 and 5), is based on this class of convex optimization problems for which efficient and robust solvers exist [Stu01, Mit03, Mos05].

2.3 Summary

In this short chapter we briefly covered the key results from mathematical programming and variational optimization used in the subsequent chapters.

Chapter 3

A Parametric Model for Variational Image Segmentation

In this chapter we present a model for natural images from which we derive a *parametric* distance measure on images useful for segmentation. While parametric models are in general limited in their descriptive power they are nevertheless attractive in applications: Properly used, they are efficient computationally and statistically, i.e., results are computed quickly and all the relevant information is employed.

3.1 Image Models and Filter Statistics

The first problem we are concerned with is that of assigning probabilities to given images: Let $p : \mathcal{I} \to [0, 1]$ be a probability density function describing for each image $I \in \mathcal{I}$ how often we will encounter I in a given application. Except for highly structured domains, e.g., for artificial or heavily preprocessed image data, it will be difficult to describe p accurately. The reasons are, first, that $\mathcal{I}$ is very large, e.g. there are 256^{256^2} moderately-sized gray-value images, ruling out many standard techniques from descriptive statistics. Second, while images are highly structured objects [GK99, PL02, LPM03] the variations experienced in the real world are overwhelming, and it is not clear at all how precisely describe their common characteristics.

An early attempt to deal with this complexity, that will also serve us in the context of more recent developments, is that of the *Markov random field* (MRF) [Bes74, GG84]: Instead of modeling the whole image domain Ω at once concentrate on small subdomains $\mathcal{N}_x \subset \Omega$ that can be modeled with sufficient accuracy. One then *assumes* the subdomains to be large enough to account for the relevant statistical dependencies in the image. By the Hammersley-Clifford theorem [Bes74] this leads, under some assumptions, to probability density functions that *factorize* over the cliques cl defined

by the neighborhoods $\mathcal{N}_x$:

$$p(I) \propto \prod_{cl(x)} \phi_{cl(x)}(x_{cl}). \tag{3.1}$$

A classical example is the ISING *model* from statistical physics [Isi25]. This model considers binary images, $I_x \in \{\pm 1\}$, and defines an image energy E, s.t. $p(I) \propto \exp(-E(I))$, which admits the form

$$E(I) = \sum_x \alpha_x I_x + \sum_{x,x' \in \mathcal{N}_x} \beta_{x,x'} I_x I_{x'}. \tag{3.2}$$

Here, β is an *interaction coefficient* measuring how strongly two image locations in a 4-neighborhood interact. α corresponds to a *prior term* determining how likely I_x equals one. Generalizations to non-binary images and to more complicated prior and interaction functions are generally summarized under *auto-models* for texture analysis [Bes74, Sec. 4.1].

The problem of course is that since we are capturing very basic pairwise pixel interactions only, the $\mathcal{N}_x$ need to be large for (3.2) to be a even remotely realistic model. Large subdomains, however, are difficult to model statistically: Thus, the problem we were trying to solve in the first place reappears. In practice, relatively small subdomains are used and addition assumptions, notably ergodicity, are made. Still, even crude approximations yield useful models for many applications ranging from image processing, to mid-level vision, or texture synthesis. More sophisticated approaches combine feature selection with second-order MRF representations and perform competitive in texture synthesis [ZG01]. In a different direction, clustering based on *textons* [MBLS01] is applied to allow MRF representations perform comparable to filter based methods in recognition [VZ03].

3.1.1 The FRAME Model

The previous example hinted that simple auto-models need to be large, making training difficult, especially with limited data. A natural approach to reduce the dimensionality of the learning problem is to apply a set of linear filters to the image and *instead of learning high-order statistics of the image values learn low-order statistics of the filter responses* [Jul62, ZWM97, WZL00, RB05]. This approach offers immediate benefits: Filters with large support naturally model medium- or large-scale dependencies in the image *without* rendering the learning problem more difficult. Also, depending on the application, one can design special-purpose filters that capture complicated geometries without risking parameter explosion during learning. And even in the generic case can filter banks often explain much image variation using relatively few coefficients only.

To implement this idea ZHU, WU, and MUMFORD [ZWM97] considered features $h : \mathcal{I} \to \mathbb{R}^n$ that collect some statistics on filter-response images. In particular, given a set of filters $f_i, i = 1, \ldots n$, operating on $\mathcal{I}$ the feature function h computes the *marginal histograms* of the filtered images

$$h(I) = (\text{hist}(f_1 * I), \text{hist}(f_2 * I), \ldots, \text{hist}(f_n * I)), \tag{3.3}$$

where $\text{hist}(I)_j = \sum_x \delta(z_j - I_x)/|\Omega|$ is the histogram with bins z_j over the image. Implementations with smoothed histograms are also used.

Now it was suggested to define a probability model p over image space that has *maximum entropy* among all models matching some observed statistics $\hat{h}_0$ *in expectation*. This yields the maximum entropy optimization problem [Jay57]:

$$\begin{aligned} \max_{p \geq 0} \quad & H(p) \\ \text{s.t.} \quad & \mathbb{E}_p[h(I)] = \hat{h}_0 \\ & \int p(I) dI = 1 \end{aligned} \tag{3.4}$$

where $H(p) = -\int p(I) \log(p(I)) dI$ denotes Shannon entropy. Examining the Lagrangean of (3.4) reveals that an optimal p admits the form of a Gibbs distribution: $p(I) \propto \exp(\langle \theta, h(I) \rangle)$, so that the only unknowns are the multipliers θ. These are readily available by noting that (3.4) is concave in θ and that the log-likelihood

$$\begin{aligned} L(\theta) &= \int \hat{p}(I) \log(p_\theta(I)) dI \\ &= \int \hat{p}(I) \left[\langle \theta, h(I) \rangle - \log \left(\int \exp(\langle \theta, h(J) \rangle) dJ \right) \right] dI \end{aligned} \tag{3.5}$$

has the derivative

$$\begin{aligned} \partial L / \partial \theta &= \int \hat{p}(I) h(I) dI - \int \hat{p}(I) \frac{\int \exp(\langle \theta, h(J) \rangle h(J)) dJ}{\int \exp(\langle \theta, h(J) \rangle) dJ} dI \\ &= \mathbb{E}_{\hat{p}}[h] - \mathbb{E}_{p_\theta}[h] \\ &= \hat{h}_0 - \mathbb{E}_{p_\theta}[h]. \end{aligned} \tag{3.6}$$

Thus, if we can compute $\mathbb{E}_{p_\theta}[h]$ we find a maximum-entropy image model by starting from any θ and following the gradient ascent rule. In general, this requires sampling from a Markov chain to find $\mathbb{E}_{p_\theta}[h]$: A usually intricate and time consuming process that nevertheless yields excellent results [ZWM97].

For texture synthesis using a set of handcrafted statistical features above time-consuming sampling procedure can reportedly be replaced by a more efficient alternative based on a sequence of projections on admissible sets [PS00]. In this setting, one starts from a random image and *sequentially* optimizes each individual constraint h_i until convergence on an image obeying all constraints is achieved. This is similar to the texture synthesis by Heeger and Bergen [HB95], however, more general image features, in particular, *parametric* features, are admissible. Although the approach is heuristic and convergence is not proven, excellent results are reported.

3.1.2 A Parametric Image Model

The FRAME model presented above accurately reproduces a wide range of synthetic and natural textures of varying complexity [ZWM97]. As such, it would be an ideal candidate

for use within an image segmentation algorithm. Unfortunately, training the model, i.e., determining suitable filters f_i and coefficients θ, is computationally expensive, and it is not clear whether training from very small image patches, such as small image regions, is feasible.

Fortunately, compared to synthesis, for segmentation less accurate models are usually sufficient. This motivates searching for simpler approximations to the FRAME model. A promising approach is to consider the *statistics of natural images*: Certain parametric statistics of natural images were frequently observed and are repeatedly mentioned in the literature. An early example is the rapidly declining autocorrelation function of television images [Kre52] or the $1/f^\alpha$ power spectra of natural images [TTC92, RB94, vdSvH96].

More recently, Bessel-K-Forms, axiomatically derived from a transparent dead-leaves image model, were empirically validated and successfully applied to clutter recognition and image coding [SLG02, SLSZ03]. Along a similar line are proposals such as, e.g., *scale mixtures of Gaussians* [PSS00, WS00], models derived from *stochastic geometry*, such as random superpositions of objects [MG01, SLG02] or the *dead leaves model* where occlusions are taken into account [LMH01]. For the latter and for the statistics of simple derivative filters on log-transformed images, excellent fit to the *generalized Laplacian* density

$$p(z) = \frac{\alpha}{2s\Gamma(1/\alpha)} \exp(-|z/s|^\alpha) \tag{3.7}$$

has been empirically established [LMH01].

Model (3.7) is not only particularly convenient to work with, it has also a long tradition: It was used for DCT coefficients of natural images [RG83] and derivative statistics of large databases of natural images [HM99]. In connection with different linear filters it has been successfully employed for image coding [Mal98, JCF95, LR97, BS99, HM99] and Bayesian image restoration [SA96].

3.1.3 Basic Physics for Images

At first thought, it is not clear how a simple parametric model as (3.7) can even partially account for the complex statistics of natural images. To understand this better we will derive an even more elementary image model from first principles.

A Simplified Image Model

In this model, an image represents, at each location x, the sum over many incident "light rays". Each light ray k incident at x has an initial energy $E_{k,x}$ and encounters objects with reflexion or transmission coefficients $\alpha_{i,k,x}$ until it finally arrives at the image plane:

$$I(x) \approx \sum_k^{n_k} E_{k,x} \prod_i^{n_i} \alpha_{i,k,x}. \tag{3.8}$$

Assumptions. Now we make the following assumptions:

1. Adjacent image locations are statistically independent.
2. $E_{k,x} = 1$.
3. $0 < c \leq \alpha_{i,k,x} \leq 1$.
4. n_k and n_i are large numbers.

Note that Assumption 1. is obviously not realistic. We will revisit this assumption later. Assumption 2. is used mainly for convenience. Assumption 3. is realistic: We use the minimum energy to get a sensor response in the imaging device (film, CCD, photo receptor cell) for c. Then assumption 3. says that we do not count the light rays that were essentially absorbed before they could reach the image plane.

Derivation. Let us introduce a random variable $z_{k,x}$ describing the accumulated reflection coefficients [Geu03]:

$$z_{k,x} = \prod_{i}^{n_i} \alpha_{i,k,x}. \tag{3.9}$$

As product of bounded RVs $z_{k,x}$ follows a power law distribution [Cha53, LS97, SC97]:

$$p(z_{k,x}) \sim z_{k,x}^{-\beta}. \tag{3.10}$$

Note that power law distributions may have infinite first and second moments.

We are interested in

$$I(x) \approx \sum_{k}^{n_k} z_{k,x}. \tag{3.11}$$

According to the *generalized limit theorem* by Gnedenko and Kolmogorov such a sum follows a *stable law*, also known as *α-stable Lévy distribution*, [Fel66, XVII.5, Thm. 2], [GK54, §35] a canonical representation of which is given by its characteristic function [GK54, §34]:

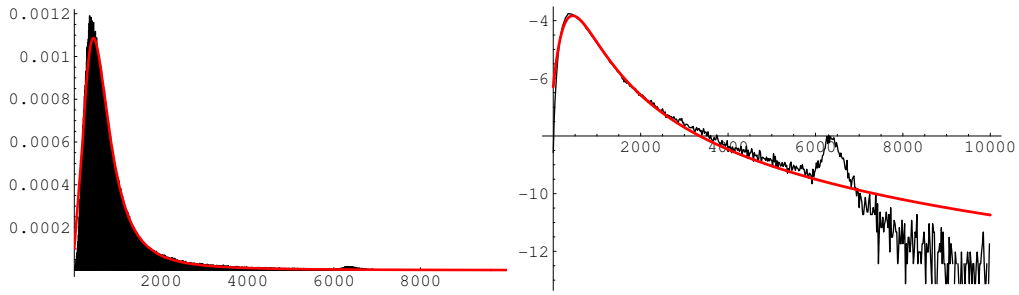
$$\log f(t) = i\gamma t - |ct|^{\alpha} \left(1 + i\beta \cdot \operatorname{sign}(t) \cdot \omega(t, \alpha)\right), \tag{3.12}$$

where

$$\omega(t, \alpha) = \begin{cases} \tan(\alpha\pi/2), & \alpha \neq 1 \\ \frac{2}{\pi} \log |t|, & \alpha = 1, \end{cases} \tag{3.13}$$

and $0 < \alpha \leq 2$ is called the *characteristic exponent*. For the other parameters we have the constraints $-1 \leq \beta \leq 1$, $\gamma \in \mathbb{R}$, and $c > 0$.

Special cases with closed form expressions are the Gaussian ($\alpha = 2$) and the Cauchy distribution ($\alpha = 1$, $\beta = 0$).



(a) Histogram & Model (b) Histogram & Model (log scale)

Figure 3.1: **Gray value statistics for natural images.** 500,000 pixels were independently sampled from the VAN HATEREN database [vHvdS98]. The corresponding gray value histogram is plotted along with the best fitting α-stable Lévy distribution (see text) on linear and on logarithmic scale: The theoretical model is an excellent fit for the data for pixel values up to 6000.

First Experiment: Lévy Fit to Gray Value Histogram

To test our results so far we compute the gray value histogram of 500,000 randomly sampled pixels from the VAN HATEREN natural image database [vHvdS98] fitted to α-stable Lévy distribution (3.12) with parameters

$$(\alpha, \beta, \gamma, \delta) \approx (1.01, 1.00, 249.93, 535.56)\,. \tag{3.14}$$

The resulting fit 3.1 is excellent except for very bright values ($I(x) > 6000$) where the Lévy distribution predicts a slower decline in frequency than empirically observed. It is not clear whether this discrepancy is due to limitations in the imaging device (saturation) or indicative for limitations of our mathematical model.

Brightness differences

Now we consider *random differences* between pixel values. That is, we look at $I(x)-I(x')$ where x and x' are random, not necessarily adjacent image locations. We need two propositions from [Nol05]:

Proposition 1. *For a stable distribution p we have:*

$$p(x, \alpha, -\beta, -\gamma, c) \stackrel{d}{=} p(-x, \alpha, \beta, \gamma, c). \tag{3.15}$$

Proof. The proof is elementary using the scaling property of the Fourier transformation on the characteristic function (3.12). □

Proposition 2. *If* $x_1 \sim p(x_1, \alpha, \beta_1, \gamma_1, c_1)$ *and* $x_2 \sim p(x_2, \alpha, \beta_2, \gamma_2, c_2)$ *then* $x_1 + x_2 \sim p(x, \alpha, \beta, \gamma, c)$ *with*

$$\beta = \frac{\beta_1 c_1^\alpha + \beta_2 c_2^\alpha}{c^\alpha}, \quad \gamma = \gamma_1 + \gamma_2, \quad c^\alpha = c_1^\alpha + c_2^\alpha. \tag{3.16}$$

Proof. Recall that $c > 0$. Multiplication of the characteristic functions in the Fourier domain yields the result. □

Thus, we find that the difference statistics computes to

$$(\alpha, \beta', \gamma', c') = (1.18187, 0, 0, 461.407) \tag{3.17}$$

which, again, is in good correspondence with the empirically observed statistics (Figure 3.2). For simplicity, we will drop the prime from β', c', γ' where there is no risk of confusion.

Writing out the characteristic function with the parameters above yields a *Laplacian-like characteristic* function:

$$\log g(t) = -\left|ct\right|^\alpha. \tag{3.18}$$

Integration of the characteristic function. To get a simpler form of the corresponding density we need compute the inverse Fourier transform:

$$\frac{1}{2\pi}\int_\infty^\infty \exp(-itx)\exp(-\left|ct\right|^\alpha)\ dt. \tag{3.19}$$

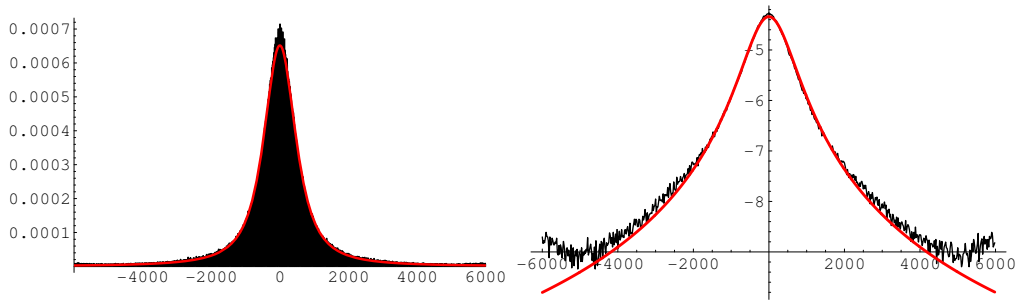
Using Taylor expansion of the integration kernel yields

$$\begin{aligned}\int_{-\infty}^{\infty} \exp(-itx)\exp(-\left|ct\right|^\alpha)\ dt &= \int_{-\infty}^{\infty}\left[\sum_{k=0}^{\infty}\frac{1}{k!}(-itx)^k\right]\exp(-\left|ct\right|^\alpha)\ dt \\ &\overset{(*)}{=} \sum_{k=0}^{\infty}\frac{1}{k!}(-ix)^k\int_{-\infty}^{\infty} t^k\exp(-\left|ct\right|^\alpha)\ dt,\end{aligned} \tag{3.20}$$

assuming uniform convergence at $(*)$. This table shows the first approximations to (3.19) for different choices of k:

k	Approximation
0	$\frac{\Gamma\left(1+\frac{1}{\alpha}\right)}{\pi c}$
1	$\frac{\Gamma\left(1+\frac{1}{\alpha}\right)}{\pi c}$
2	$\frac{\Gamma\left(1+\frac{1}{\alpha}\right)}{\pi c} - \frac{x^2\Gamma\left(\frac{\alpha+3}{\alpha}\right)}{6\pi c^3}$
3	$\frac{\Gamma\left(1+\frac{1}{\alpha}\right)}{\pi c} - \frac{x^2\Gamma\left(\frac{\alpha+3}{\alpha}\right)}{6\pi c^3}$
4	$\frac{\Gamma\left(\frac{\alpha+5}{\alpha}\right)x^4}{120\pi c^5} - \frac{\Gamma\left(\frac{\alpha+3}{\alpha}\right)x^2}{6\pi c^3} + \frac{\Gamma\left(1+\frac{1}{\alpha}\right)}{\pi c}$
5	$\frac{\Gamma\left(\frac{\alpha+5}{\alpha}\right)x^4}{120\pi c^5} - \frac{\Gamma\left(\frac{\alpha+3}{\alpha}\right)x^2}{6\pi c^3} + \frac{\Gamma\left(1+\frac{1}{\alpha}\right)}{\pi c}$
6	$-\frac{\Gamma\left(\frac{\alpha+7}{\alpha}\right)x^6}{5040\pi\ c^7} + \frac{\Gamma\left(\frac{\alpha+5}{\alpha}\right)x^4}{120\pi c^5} - \frac{\Gamma\left(\frac{\alpha+3}{\alpha}\right)x^2}{6\pi c^3} + \frac{\Gamma\left(1+\frac{1}{\alpha}\right)}{\pi c}$
7	$-\frac{\Gamma\left(\frac{\alpha+7}{\alpha}\right)x^6}{5040\pi\ c^7} + \frac{\Gamma\left(\frac{\alpha+5}{\alpha}\right)x^4}{120\pi c^5} - \frac{\Gamma\left(\frac{\alpha+3}{\alpha}\right)x^2}{6\pi c^3} + \frac{\Gamma\left(1+\frac{1}{\alpha}\right)}{\pi c}$

(3.21)



(a) Histogram & Model (b) Histogram & Model (log scale)

Figure 3.2: **Statistics for pairwise difference of random pixels.** $I(x_1) - I(x_2)$ is computed for randomly selected pixels x_1 and x_2. The fit is excellent in $[-2000, 2000]$. The empirical tails are slightly heavier than the model's.

We recognize the general rule

$$\sum_{k=1,3,\ldots} \frac{(-1)^{\frac{k-1}{2}} \Gamma(\frac{\alpha+k}{\alpha}) x^{k-1}}{\pi c^k k!} \tag{3.22}$$

and note a slight similarity to the expansion of $\cos(x)$.

To the best of our knowledge, there is no closed form representation for series (3.22). However, for some specific choices of α we recognize the series:

$$\begin{array}{lll} \alpha = 1: & \frac{c}{\pi(c^2+x^2)} & \text{Cauchy} \\ \alpha = 2: & \frac{1}{2c\sqrt{\pi}} \exp(-x^2/(4c^2)) & \text{Gaussian} \end{array} \tag{3.23}$$

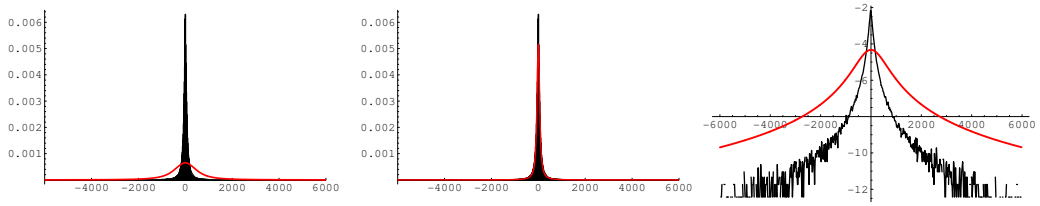
Unfortunately, convergence of (3.22) is rather slow. Even with $k \leq 100$ we have a radius of convergence for our estimated parameters α and c (eqn. (3.17)) of approximately $x \in [-600, 600]$, which expands to $[-1100, 1100]$ with $k \leq 1000$.

Second Experiment: Lévy Fit to Gray Value Differences

Using the samples obtained from the VAN HATEREN database we computed differences between random pixels and corresponding histograms. Again, our model fits the data well, except for the tails which are slightly heavier than predicted (Figure 3.2).

Comparison with Generalized Laplacians

The generalized Laplacian model was empirically shown to be an excellent fit for derivative statistics of the Haar-Wavelet type [HM99]. Such derivative statistics can be obtained when we observe $I(x_1) - I(x_2)$ where x_1 and x_2 are *neighboring* pixels as opposed to two randomly sampled locations.



(a) General Lévy (b) Fitted Lévy (c) Fitted Lévy (log scale)

Figure 3.3: **Statistics for pairwise difference of *neighboring* pixels.** $I(x_1) - I(x_2)$ is computed for neighboring pixels x_1 and x_2. Fig 3.3(a) shows the fit of the Lévy distribution for *general* (non-neighboring) pixel values. Even when the parameters of the distributions are fitted to the new histogram (Figure 3.3(b)) the results are not convincing as the log-scale plot (Figure 3.3(c)) reveals.

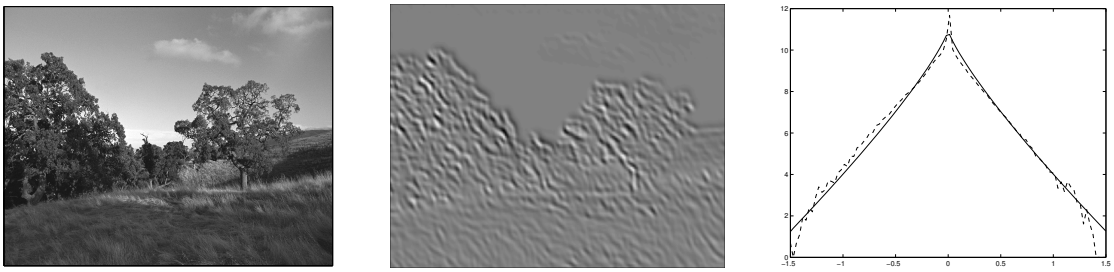
It is interesting to observe differences between the statistics of the generalized Laplacians and the Lévy statistics (3.22). For instance, it is easy to see that at the origin the first derivative of (3.22) exists and is zero. In contrast, the first derivative of the generalized Laplacian does not exist at the origin, and its upper and lower limits are non-zero.

In Figure 3.3 we show the statistics of gray value differences between *neighboring* pixels. The histogram is much stronger peaked at 0. For comparison, we show the Lévy distribution corresponding to the parameters computed above (3.17). Evidently, these are two very different distributions. Re-fitting the new data improves the results (Figure 3.3(b)), but the log-scaled plot (Figure 3.3(c)) shows that the Lévy model is qualitatively different from the observed histogram.

At this point we need to revisit the assumptions of our model (p. 17): The only difference between Figure 3.2 and Fig. 3.3 is that in the latter spatial correlations play a role while they are eliminated in the first experiment. In a sense, the Lévy model provides a baseline describing how images would look like if there was no highly structured world outside. Figure 3.3(a) visualizes: The entropy of the Lévy model (≈ 12.2 bit) is about 40% larger than the entropy of the corresponding generalized Laplacian (≈ 8.6 bit). If the Lévy model were accurate images would look a lot more random.

3.2 A Variational Approach to Image Segmentation

In the light of the ideas above, we decided to capture statistics of natural images using generalized Laplacians fitted to marginal histograms of linear filter responses. The Kullback-Leibler (KL) distance between the Laplacians then serves as a distance measure on the images (cf. Figure 3.4). The following section describes the approach in detail.



(a) Image (b) Filter response (c) Histogram

Figure 3.4: **Overview.** An image (a) is filtered by a linear filter. From the resulting filter response (b) the marginal histogram (dotted line) is extracted and a generalized Laplacian (solid line) is fitted to the histogram (c). The parameters (α, s) of the generalized Laplacian serve as image descriptors.

3.2.1 Parameter Estimation

As mentioned above, the basis of our approach is the statistical model

$$p(z) = \frac{\alpha}{2s\Gamma(1/\alpha)} \exp(-|z/s|^{\alpha})$$

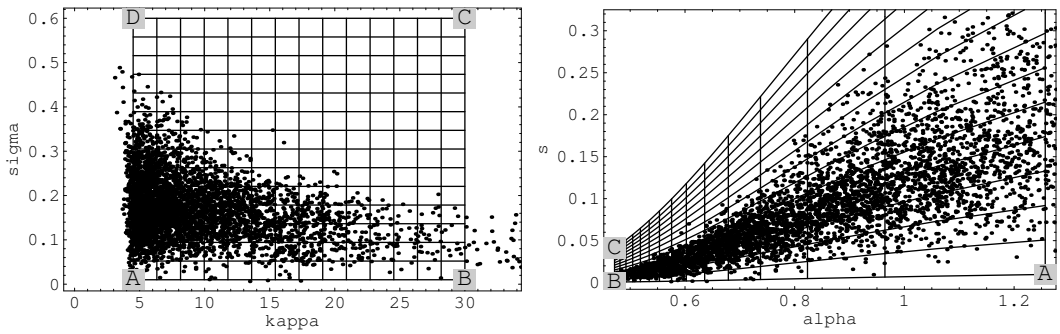
for the filter response z of a linear filter applied to natural images.

The generalized Laplacian model (3.7) has two parameters, s and α, which are related to variance σ^2 and kurtosis κ of the filter response by

$$\sigma^2 = \frac{s^2\Gamma(3/\alpha)}{\Gamma(1/\alpha)} \qquad \kappa = \frac{\Gamma(1/\alpha)\Gamma(5/\alpha)}{\Gamma^2(3/\alpha)}. \tag{3.24}$$

Figure 3.5 illustrates the nonlinear mapping from the measured statistics (σ, κ) to the model parameters (s, α) in (3.7). When $\kappa > 9/5$ we can solve the right equation numerically for α and determine s via the left equation. Mathematically, we cannot model distributions with $\kappa \leq 9/5$ as for $\alpha \to \infty$ the generalized Laplacian approaches the uniform distribution centered at 0, the kurtosis of which equals $9/5$. This is not a severe restriction, however: In the experimental section (Section 3.2.8 and Figure 3.8(a)) we show that such statistics are very rare in natural images.

We found experimentally (Section 3.2.8) that model (3.7) fits a large range of linear filter responses very well. In particular, we examined differences between steerable pyramid filters, quadrature mirror filters, or the well-known Haar wavelet and Daubechies wavelet of order 3. In the following, these filters are abbreviated by sp$n$, qmf$n$, haar, and daub3, where n is an integer encoding the number of filter orientations. These results are in line with findings on simple derivative filters reported by Huang and Mumford [HM99, Hua00].



(a) Filter response parameters.

(b) Density parameters.

Figure 3.5: **The role of nonlinear parameter mapping.** Standard deviation σ and kurtosis κ of the filter responses are mapped nonlinearly according to (3.24) to density parameters α and s. Note, that the left part of (a), where most points are located, is spread after mapping. Conversely, the area on the right of (a), where relatively few points are situated, is compressed. The points depicted are 4167 measurements collected from the van Hateren database using a linear derivative filter. The labeled grid visualizes the nonlinearity of the transformation.

3.2.2 Choice of Distance Measure

The general idea behind our segmentation approach is to compute two image densities, p_{in} and p_{out}, for each image. One describes the *interior* region of a segmentation, the other describes the *exterior* region. We will show later (Section 3.3) how to extend this approach to the multiclass case with more than two image regions.

If p_{in} and p_{out} are our basic image descriptors the question arises how to measure (dis)similarity between two given densities. This question is important, since in our variational framework it will be the similarity measure that drives the evolution of segmentation boundaries.

Natural candidates for probabilistic distances measures are the *Kullback-Leibler* divergence $D(p||q)$ which measures information loss when distribution p is approximated by a different distribution q [CT91]. χ^2 is a very popular statistical goodness-of-fit test that allows the density parameters to be estimated from data. The *Kolmogorov-Smirnov test* is a divergence measure between probability measures that reportedly works well even with small data samples, and the *Anderson-Darling test* is a variant which places equal weight along the cumulative probability function [AD52]. A priori it is not clear which of these distance measures should be preferred.

To answer this question we carried out a small-scale texture retrieval experiment: The Brodatz database of texture images [Bro66] was divided into patches. Specifically, we extracted 16 image patches of size 100×100 pixels non-overlappingly from 32 Brodatz images. These patches were filtered with different filter banks, and statistics were extracted. Then each patch was used as a query patch and the nearest neighbor patch

was found from the remaining ones using different distance measures. In particular, we computed σ, κ, α, and s for each image and measured distances using the ℓ_1 norm, KL divergence for the generalized Laplacian (KL), non-parametric KL divergence on the filter response histograms (kl), as well as Kolmogorov-Smirnov (ks), Anderson-Darling (ad) and χ^2, also on the filter response histograms. For each filter bank and each distance measure we used the max, mean, and median operator to map the distances between the individual filters in the filter bank to a single number.

The retrieval error rates are depicted in Table 3.1: It reveals that working on the raw variables σ, κ or α, s is not advisable. The probabilistic distance measures work much better. The parametric KL divergence (KL) performs very well, especially with the qmf16 and the sp3 filter bank. Surprisingly, non-parametric KL divergence (kl) fails to yield acceptable retrieval rates. A possible explanation is that we found it to be quite sensitive against histogram binning effects, and our particular binning might not have been optimal for the task at hand. On the other hand, Kolmogorov-Smirnov, Anderson-Darling, and χ^2 perform well on the histogram data.

We find that the parametric KL divergence measure performs competitive with the non-parametric distance measures. Given its simplicity this is quite surprising. Since the parametric KL divergence also leads to convenient variational formulations (Section 3.2.5) we adopt it for our further experiments.

3.2.3 MDL-Criterion for Segmentation

Our goal is to partition the image domain Ω into two, regions Ω_{in} and Ω_{out} separated by a contour $\mathcal{C}$ such that the local image statistics are "close" to the global statistics within Ω_{in} or Ω_{out}, respectively. More precisely, if p_x denotes the statistics of a small window W_x centered at image location x, and if p_{in} and p_{out} denote the statistics of the interior and exterior regions Ω_{in} and Ω_{out}, respectively, then we want to minimize

$$E_{\text{mdl}}(\Omega_{\text{in}}, \Omega_{\text{out}}) = \int_{\mathcal{C}} ds + \int_{\Omega_{\text{in}}} D(p_x||p_{\text{in}})dx + \int_{\Omega_{\text{out}}} D(p_x||p_{\text{out}})dx. \tag{3.25}$$

Here $D(p||q) = -\int p(z)\log(p(z)/q(z))dz$ is the Kullback-Leibler (KL) distance between densities p and q. Note that (3.25) fits into Zhu and Yuille's region competition framework [ZY96] when $P_{\text{in/out}} \propto \exp(-D(p_x||p_{\text{in/out}}))$ are the probabilities for the region models and the local image features I_x are given by the distributions p_x.

The motivation for energy (3.25) is that it can be linked to the length of a hypothetical image code [WB68, Ris78] based on two generalized Laplacians p_{in} and p_{out}: We encode the filter response image using either p_{in} or p_{out} as models. Assuming densities are truly Laplacian, encoding a pixel x with model p_x estimated from W_x has average length $H(p_x)$, H denoting Shannon entropy. Encoding it using the model for one of the regions Ω_{in} or Ω_{out} instead requires a code of average length $H(p_x) + D(p_x||p_{\text{in/out}})$. KL-distance is nonnegative, therefore (3.25) describes the additional coding effort we face when encoding x with the models for one of the regions. The first integral in (3.25)

	$\sigma\kappa/\ell_1$			$\alpha s/\ell_1$			KL			kl			ks			ad			χ^2		
	max	avg	med	max	avg	med	max	avg	med	max	avg	med	max	avg	med	max	avg	med	max	avg	med
sp0	98	86	108	126	128	285	62	60	94	196	290	298	44	40	56	41	37	53	66	**29**	53
sp1	54	25	46	55	40	91	25	20	26	40	185	255	31	18	**16**	28	24	23	44	21	21
sp3	60	23	31	54	36	40	22	**12**	18	28	193	230	18	16	14	21	17	18	30	17	14
sp5	73	26	25	54	28	35	31	23	19	35	164	220	31	18	**14**	29	21	**14**	34	20	**14**
qmf9	67	16	28	74	35	68	15	13	17	83	235	254	36	12	43	16	**11**	31	79	14	45
qmf12	71	14	29	69	32	71	15	12	19	71	233	234	32	14	55	20	**11**	29	75	14	39
qmf16	57	12	30	57	33	71	12	**8***	11	44	185	207	32	13	24	22	11	18	58	**8***	30
haar	64	19	31	77	36	106	24	**12**	26	119	309	317	51	**12**	41	26	17	30	74	14	26
daub3	74	18	39	74	40	79	22	15	18	88	209	218	45	16	42	30	**13**	30	76	14	35

Table 3.1: **Comparison of distance measures.** Errors for a texture retrieval experiment are reported for different choices of filters (rows) and different choices of distance measures (columns). See text for details.

measures the length of the separating contour $\mathcal{C}$, ensuring that the membership relation, that is, whether a specific point x belongs to Ω_{in} or to Ω_{out}, will be inexpensive to encode [Lec89].

The KL-distance between two generalized Laplacians p and q with parameters (s_p, α_p) and (s_q, α_q) can be computed conveniently: First, evaluate (3.24) for sample estimates on the left hand sides, then insert the resulting values for the parameters $(s_p, \alpha_p, s_q, \alpha_q)$ into the following expression:

$$D(p||q) = \frac{(\frac{s_p}{s_q})^{\alpha_q}\Gamma(\frac{1+\alpha_q}{\alpha_p})}{\Gamma(\frac{1}{\alpha_p})} + \log\left(\frac{s_q\Gamma(1+\frac{1}{\alpha_q})}{s_p\Gamma(1+\frac{1}{\alpha_p})}\right) - \frac{1}{\alpha_p}. \tag{3.26}$$

Note, that the hypothetical image code described above is only optimal if adjacent pixels in the filter response are statistically *independent*. Spatial correlations of filter responses at neighboring locations are not exploited. For an efficient real-world coding scheme this would be mandatory.

3.2.4 Combining Filter Responses

Given the statistics for a set of filter responses, how do we combine information gathered at different scales and orientations? In this work, we strive for a *generic* measure not optimized for any particular set of textures or filters, so feature selection schemes are not directly applicable.

We propose, as a first approximation, to treat the statistics of individual filter responses as statistically independent. Under this assumption the individual KL-distances simply add up so that we can minimize the average distance collected over all linear filters i:

$$E_{\text{mdl}}(\Omega_{\text{in}}, \Omega_{\text{out}}) = \int_{\mathcal{C}} ds \; + \sum_i \left[\int_{\Omega_{\text{in}}} D(p_{x,i}||p_{\text{in},i})dx + \int_{\Omega_{\text{out}}} D(p_{x,i}||p_{\text{out},i})dx\right] \tag{3.27}$$

Here $p_{\text{in/out},i}$ denotes the probability density function modeling the response of filter i in region $\Omega_{\text{in/out}}$ and $p_{x,i}$ is the corresponding density for a window W_x centered at location x in the image plane.

It is known that in reality the independence assumption does *not* hold. For orthogonal wavelet bases normalization schemes have been proposed to remove dependencies between filter responses at different scale and orientation [BS99, WSW01]. In this first implementation of our approach, however, we did not incorporate any such scheme. While in theory this is clearly suboptimal, our experiments (Section 3.2.8) suggest that the model is sufficiently accurate for many real-world scenes.

3.2.5 Level Set Formulation

In this section we incorporate our statistical distance measure into a level set formulation. The update equations determining the dynamics of the segmentation are rigorously

derived, taking into account *all* region-dependend terms, by computing the first variation of the corresponding area integrals.

Energy Functional

We minimize energy (3.27) within the region-based variational framework of Chan and Vese [CV01]. The framework applies to energy functionals of the form

$$E(\phi) = \int_\Omega k^{\mathrm{b}}(x)\,|\nabla\phi|\,\delta(\phi)dx + \lambda_1 \int_\Omega k^{\mathrm{out}}(x,\phi)H(\phi)dx + \lambda_2 \int_\Omega k^{\mathrm{in}}(x,\phi)(1-H(\phi))dx. \tag{3.28}$$

Here $\phi : \mathbb{R}^2 \to \mathbb{R}$ denotes the embedding level set function, the zero-level of which represents segmentation boundaries. $H : \mathbb{R} \to \{0,1\}$ is the step function and δ Dirac's delta function. $k^{\mathrm{b}}(x)$, $k^{\mathrm{in}}(x,\phi)$, and $k^{\mathrm{out}}(x,\phi)$ represent the boundary, interior, and exterior energy contributions at a location x and for a given level set function ϕ. Finally, λ_1 and λ_2 weight the relative importance of the interior and exterior energy terms against boundary energy. In the following we usually drop the arguments ϕ and x for brevity.

Chan and Vese's original gray-value based image model [CV01] fits into this framework as a special case with

$$\begin{cases} k^{\mathrm{b}} & = 1 \\ k^{\mathrm{in}} & = |u_0 - c_{\mathrm{in}}|^2 \\ k^{\mathrm{out}} & = |u_0 - c_{\mathrm{out}}|^2\,, \end{cases} \tag{3.29}$$

whereas with

$$\begin{cases} k^{\mathrm{b}} & = 1 \\ k^{\mathrm{in}} & = \sum_i D(p_{x,i}||p_{\mathrm{in},i}) \\ k^{\mathrm{out}} & = \sum_i D(p_{x,i}||p_{\mathrm{out},i}) \end{cases} \tag{3.30}$$

energy (3.27) is obtained.

First Variation and Boundary Update

The variational update $\dot\phi = -\langle E'(\phi), \psi\rangle$, $\forall\psi$, of the level set function reads[1] (Section 2.1.3, eqn. (2.9)):

$$\frac{\partial E}{\partial \phi} = \frac{\partial}{\partial\phi}\left[\int_\Omega k^{\mathrm{b}}\,|\nabla\phi|\,\delta dx\right] + \int_\Omega (\lambda_1 k^{\mathrm{out}} - \lambda_2 k^{\mathrm{in}})\delta\psi\; dx + \int_\Omega \left(\lambda_1 \frac{\partial k^{\mathrm{out}}}{\partial\phi} H + \lambda_2 \frac{\partial k^{\mathrm{in}}}{\partial\phi}(1-H)\right)\psi\; dx. \tag{3.31}$$

[1] To save horizontal space we abbreviate $\langle E'(\phi), \psi\rangle$ by $\partial E/\partial\phi$.

The third term, which is omitted in [CV01], originates from applying the product rule to the area integrals and thus takes into account that k^{in} and k^{out} also depend on the level set function ϕ. After some tedious calculations (Section 2.1.3) and with the shorthands $n = \frac{\nabla\phi}{|\nabla\phi|}$ and $c = \text{div}(n)$ we arrive at (eqn. (2.15))

$$\frac{\partial E}{\partial \phi} = \int_{\mathcal{C}} \left(-\nabla k^{\text{b}} n - k^{\text{b}} c + \lambda_1 k^{\text{out}} - \lambda_2 k^{\text{in}} \right) \psi \; ds + \int_{\Omega} \left(\lambda_1 \frac{\partial k^{\text{out}}}{\partial \phi} H + \lambda_2 \frac{\partial k^{\text{in}}}{\partial \phi} (1 - H) \right) \psi \; dx \; . \quad (3.32)$$

We point out that this formula was recently derived in a different way in [JBBA03] based on the calculus of shape optimal design [SZ91] which, in turn, relies on previous mathematical work like, e.g., [Sim80].

Derivation of the Model's Area Term

Let us examine more closely the area integral in (2.15). As mentioned above in eqn. (3.30) we model the local coding cost w.r.t. the interior region as

$$k^{\text{in}} = \sum_i D(p_{x,i} || p_{\text{in},i}). \quad (3.33)$$

Recall that the probability density functions are given as generalized Laplacians with two parameters $s = s(\alpha, \sigma^2)$ and $\alpha = \alpha(\kappa)$ which depend themselves on kurtosis κ and variance σ^2 measured both locally in W_x and globally in Ω_{in}. Therefore, we may write more precisely

$$k^{\text{in}} = \sum_i D(p(\alpha(\kappa_{x,i}), s(\alpha(\kappa_{x,i}), \sigma^2_{x,i})) || p(\alpha(\kappa_{\text{in},i}), s(\alpha(\kappa_{\text{in},i}), \sigma^2_{\text{in},i}))). \quad (3.34)$$

Here $\kappa_{\text{in},i}$ and $\sigma^2_{\text{in},i}$ depend on the area Ω_{in} and thus vary with the level set function ϕ. Let us drop the index i in the following discussion, thus focusing on a single filter response only.

With a slight abuse of notation, the derivative then reads

$$\frac{\partial k^{\text{in}}}{\partial \phi} = \frac{\partial D}{\partial \kappa_{\text{in}}} \frac{\partial \kappa_{\text{in}}}{\partial \phi} + \frac{\partial D}{\partial \sigma^2_{\text{in}}} \frac{\partial \sigma^2_{\text{in}}}{\partial \phi}, \quad (3.35)$$

where the computation of the partial derivatives $\partial D / \partial \kappa_{\text{in}}$ and $\partial D / \partial \sigma^2_{\text{in}}$ is long but nevertheless elementary: Starting from the analytical formulation of the KL-distance (3.26) and inserting the relations (3.24) solved for α and s it is easily obtained.

The statistics depending on the area form a hierarchy of region-dependent terms:

$$\kappa_{\text{in}} = \int_{\Omega_{\text{in}}} \frac{(x - \mu_{\text{in}})^4}{|\Omega_{\text{in}}| \, \sigma^4_{\text{in}}} dx \qquad \sigma^2_{\text{in}} = \int_{\Omega_{\text{in}}} \frac{(x - \mu_{\text{in}})^2}{|\Omega_{\text{in}}|} dx$$
$$\mu_{\text{in}} = \int_{\Omega_{\text{in}}} \frac{x}{|\Omega_{\text{in}}|} dx \qquad |\Omega_{\text{in}}| = \int_{\Omega_{\text{in}}} dx. \quad (3.36)$$

In the level set formulation (3.28) we replace the integrals over Ω_{in} by integrals over Ω weighted by the step function H. Now, taking the derivative w.r.t. ϕ yields (cf. Appendix 3.2.6)

$$\frac{\partial \sigma_{\text{in}}^2}{\partial \phi} = -\int_\Omega \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|^2} H \, dx \int_\Omega \delta\psi \, dx + \int_\Omega \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|} \delta\psi \, dx \tag{3.37}$$

and

$$\begin{aligned}\frac{\partial \kappa_{\text{in}}}{\partial \phi} &= \int_\Omega \frac{-4(x-\mu_{\text{in}})^3}{|\Omega_{\text{in}}|\,\sigma_{\text{in}}^4} H \, dx \left[\int_\Omega \frac{-x}{|\Omega_{\text{in}}|^2} H \, dx \int_\Omega \delta\psi \, dx + \int_\Omega \frac{x}{|\Omega_{\text{in}}|} \delta\psi \, dx \right] \\ &+ 2\sigma_{\text{in}}^2 \int_\Omega \frac{(x-\mu_{\text{in}})^4}{|\Omega_{\text{in}}|\,\sigma_{\text{in}}^8} H \, dx \left[\int_\Omega \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|^2} H \, dx \int_\Omega \delta\psi \, dx - \int_\Omega \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|} \delta\psi \, dx \right] \\ &- \int_\Omega \frac{(x-\mu_{\text{in}})^4}{|\Omega_{\text{in}}|^2\,\sigma_{\text{in}}^4} H \, dx \int_\Omega \delta\psi \, dx + \int_\Omega \frac{(x-\mu_{\text{in}})^4}{|\Omega_{\text{in}}|\,\sigma_{\text{in}}^4} \delta\psi \, dx.\end{aligned} \tag{3.38}$$

With (3.35) these terms form the area derivatives in (2.15).

3.2.6 Derivation of the Area Terms

We start from the relations (3.36) and replace the integrals over Ω_{in} by integrals over Ω weighted by the step function H. Taking the derivative w.r.t. ϕ yields

$$\frac{\partial \sigma_{\text{in}}^2}{\partial \phi} = \frac{\partial}{\partial \phi} \int_\Omega \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|} H \, dx = \int_\Omega \frac{\partial}{\partial \phi} \left[\frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|} \right] H + \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|} \delta\psi \, dx \tag{3.39}$$

and

$$\frac{\partial}{\partial \phi} \left[\frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|} \right] = 2\frac{\mu_{\text{in}} - x}{|\Omega_{\text{in}}|} \frac{\partial \mu_{\text{in}}}{\partial \phi} - \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|^2} \frac{\partial |\Omega_{\text{in}}|}{\partial \phi} \tag{3.40}$$

and finally

$$\frac{\partial |\Omega_{\text{in}}|}{\partial \phi} = \int_\Omega \frac{\partial H}{\partial \phi} dx = \int_\Omega \delta\psi \, dx. \tag{3.41}$$

Collecting these terms and using $\int_\Omega (\mu_{\text{in}} - x) H \, dx = 0$ yields (3.37).

The derivation of $\partial\kappa/\partial\phi$ proceeds in the very same manner:

$$\begin{aligned}\frac{\partial \kappa_{\text{in}}}{\partial \phi} &= \frac{\partial}{\partial \phi} \int_\Omega \frac{(x-\mu_{\text{in}})^4}{|\Omega_{\text{in}}|\,\sigma_{\text{in}}^4} H \, dx = \int_\Omega \frac{\partial}{\partial \phi} \left[\frac{(x-\mu_{\text{in}})^4}{|\Omega_{\text{in}}|\,\sigma_{\text{in}}^4} \right] H + \frac{(x-\mu_{\text{in}})^4}{|\Omega_{\text{in}}|\,\sigma_{\text{in}}^4} \delta\psi \, dx \\ &= \int_\Omega \frac{-4(x-\mu_{\text{in}})^3}{|\Omega_{\text{in}}|\,\sigma_{\text{in}}^4} \frac{\partial \mu_{\text{in}}}{\partial \phi} H - \frac{(x-\mu_{\text{in}})^4}{|\Omega_{\text{in}}|^2\,\sigma_{\text{in}}^8} \left(\sigma_{\text{in}}^4 \frac{\partial |\Omega_{\text{in}}|}{\partial \phi} + |\Omega_{\text{in}}| \frac{\partial \sigma_{\text{in}}^4}{\partial \phi} \right) H + \frac{(x-\mu_{\text{in}})^4}{|\Omega_{\text{in}}|\,\sigma_{\text{in}}^4} \delta\psi \, dx\end{aligned} \tag{3.42}$$

where

$$\begin{aligned}\frac{\partial \mu_{\text{in}}}{\partial \phi} &= \frac{\partial}{\partial \phi} \int_\Omega \frac{x}{|\Omega_{\text{in}}|} H \, dx = \int_\Omega \frac{-x}{|\Omega_{\text{in}}|^2} \frac{\partial |\Omega_{\text{in}}|}{\partial \phi} H + \frac{x}{|\Omega_{\text{in}}|} \delta\psi \, dx \\ &= -\int_\Omega \frac{x}{|\Omega_{\text{in}}|^2} H \, dx \int_\Omega \delta\psi \, dx + \int_\Omega \frac{x}{|\Omega_{\text{in}}|} \delta\psi \, dx\end{aligned} \tag{3.43}$$

Figure 3.6: **KL-Segmentation with Normalized Cut.** A similarity matrix, derived from KL-distances between locally fitted generalized Laplacians (see text), was treated within the classical normalized cut framework. No effort was undertaken to enforce particularly smooth partitions. The blocky segmentation boundaries are an artifact of our particular implementation.

and

$$\frac{\partial \sigma_{\text{in}}^4}{\partial \phi} = \frac{\partial}{\partial \phi}\left(\int_\Omega \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|} H \; dx\right)^2 = 2\sigma_{\text{in}}^2 \int_\Omega \frac{\partial}{\partial \phi}\left[\frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|}\right] H + \frac{(x-\mu_{\text{in}})^2}{|\Omega_{\text{in}}|}\delta\psi \; dx. \tag{3.44}$$

Inserting the various terms into each other, yields (3.38).

3.2.7 Relation to Established Segmentation Approaches

The image model we employ can potentially be useful within alternative segmentation frameworks based on graph cuts [SM00, KSSC03], density clustering [PHB99], or within the image parsing framework [TZ02]. For graph cut methods, the KL-distances $D(p_x||p_{x'})$ between generalized Laplacians p_x and $p_{x'}$ is easily translated into a similarity value $w_{xx'} = \exp(-D(p_x||p_{x'})/c)$ which can then be treated, for instance, by normalized cut. In Figure 3.6 we show, as a mere proof of concept, results obtained when such similarity graphs are partitioned by normalized cut. The scaling constant c was chosen as the mean KL-distance observed in the images, and the images were subsampled to reduce the size of the eigenvalue problem to approximately 1000×1000 matrix entries. The results roughly resemble those of the level set implementation and could further be improved by integrating a smoothing term in the similarity measure and by implementing a more sophisticated approximation method [FBCM04] to reduce block-artifacts.

Note, however, that in the graph cut framework there are no explicit models p_{in} and p_{out} for the complete interior and exterior image regions Ω_{in} and Ω_{out}. Only similarities between *locally* estimated image models p_x are used. In connection with our approach this might be a drawback as the global estimation over the larger image regions Ω_{in} and Ω_{out} is usually more reliable than the local models estimated from small image windows: During a typical PDE evolution the region models p_{in} and p_{out} are refined iteratively until they represent their corresponding image regions quite accurately. This is not possible for non-iterative optimization methods.

Conversely, our framework can benefit from employing alternative image models instead of parametric generalized Laplacians. Bessel K forms, derived from an image model based on weighted superposition of transparent objects, are suggested to represent broad image classes [SLG02]. Similarly, Weibull and power-law distributions were recently proposed and evaluated on thousands of natural images [GS03]. The beauty of these parametric models is that, while depending on few parameters only, they apply in rather broad contexts, and in some cases statistical goodness-of-fit tests are readily available.

When more flexibility in image modeling is needed, in particular for images with regular textures, mixture models [BCGM98, PD02, WCH03] and non-parametric models [TM01, RBD03] come into play. These can, in principle, model empirical densities to arbitrary precision. However, in order to avoid over-fitting within unsupervised settings, care must be taken that model complexity is kept under control. Also, KL-distances can in general no longer be evaluated analytically.

Empirical densities represented by histograms of filter responses also provide greater modeling capacity [ZWM98]. They fit into framework (3.28) when the parametric KL-distance in (3.30) is replaced by the discrete KL-distance between histograms. However, this solution might not be optimal as results can be sensitive to the chosen histogram bin-size. Therefore, more robust statistical measures, such as earth movers distance, χ^2, Kolmogorov-Smirnov, or the Anderson-Darling statistics seem more promising [PRTB99, RTG00, LW03, GS03].

Ideally, the user would not be required to decide for a particular image model or for the number of different image regions to expect *a priori*, but multiple models of different complexity would compete to explain the image during the course of optimization. This leads to a model selection problem which can in principle be treated within an MDL framework [Lec89, HY01]. While extensions of the Chan and Vese framework to multiple image regions [VC02] and models [CSS06] have been proposed, it is unclear if they generalize to a full MDL approach with multiple image models of different modeling capacity. Currently, methods from non-convex optimization are employed to handle such problems [Lec89, ZLW00].

3.2.8 Experiments

Now we describe extensive computational studies of the performance of our model. We validate the use of generalized Laplacian densities for steerable pyramid filter response statistics of natural images, perform experiments in texture retrieval and synthesis to understand what image features are captured by our model, and show sample segmentations on natural and artificial images. We compare our approach to a standard second-order variational model for image segmentation and demonstrate that it performs well.

	sp0	sp1	sp3	sp5	qmf9	qmf12	qmf16	haar	daub3
KL-dist	0.018	0.011	0.012	0.014	0.016	0.016	0.017	0.016	0.017
Entropy	2.274	2.070	1.996	2.004	1.933	1.932	1.938	1.994	1.966
KL/entropy	0.008	0.005	0.006	0.007	0.009	0.009	0.009	0.008	0.009

Table 3.2: **Model fit.** Medians of KL-distances between histograms and parametric model (3.7) measured over 4167 pictures from the van Hateren database [vHvdS98] for different sets of filters. For comparison, median entropies of the filter responses are also reported: Only a small fraction of the information present in the histograms is ignored (last row).

Filter Selection and Model Validation

Various linear transformations of images have been used in conjunction with the model: The discrete cosine transform [RG83], steerable pyramids [FA91, SF95, SA96], and different orthogonal wavelets [Mal98, HM99].

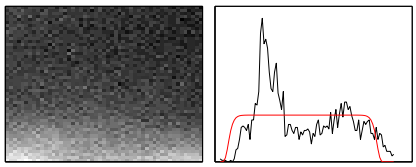
Before focusing on segmentation (Section 3.2.8) we conducted experiments to select a suitable filter bank and to verify that the restriction on the kurtosis of the filter response to be greater than 9/5 is met in practice (Section 3.2). Following [HM99] we used the van Hateren database of natural images [vHvdS98] for evaluation and removed multiplicative constants from the images by first log-transforming them and then subtracting their log-means.

Table 3.2 summarizes our results: We display the median of the KL-distance between the filter response histograms (20 bins) and a generalized Laplacian with identical variance and kurtosis. For comparison, we also report the histograms' average entropy and the median of the quotient of these values. The results show that almost all information in the histograms is captured by the parametric model. Importantly, the same holds for densities estimated *locally* from moderately small image patches (Figure 3.7). In the following, we perform all experiments using the steerable pyramid bank sp3 with four oriented sub-band filters and over three scales.

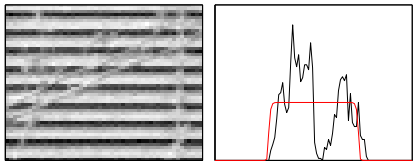
In Figure 3.8(a) we show the log-histograms of the kurtosis κ for each individual filter determined for all 4167 images of the database. Two things are remarkable: First, the distribution of κ follows closely a shifted exponential distribution. Second, the minimal values of κ encountered are well above the critical value of 9/5. Thus, *distributions that violate the kurtosis-constraint of our model do not occur in natural images.*

Clearly, during segmentation we also work with small *parts* of images for which small values for kurtosis *are* observed. Especially very homogeneous image regions like sky or plain street occasionally lead to untypical filter response histograms (Figure 3.7). To see how frequently this happens in reality, we randomly sampled over 700,000 image patches of size $10^2, 20^2, 30^2, 40^2, 50^2, 75^2$, and 100^2 pixels from the van Hateren database. For each patch size we counted how often the constraint $\kappa > 9/5$ was violated. The relative frequencies are shown in Figure 3.8(b): Only for the two smallest patch sizes, corre-

patch size	sp0	sp1	sp3	sp5
10×10	0.163	0.144	0.144	0.162
20×20	0.053	0.045	0.046	0.054
30×30	0.032	0.025	0.025	0.032
40×40	0.023	0.017	0.017	0.023
50×50	0.019	0.013	0.014	0.018
100×100	0.010	0.007	0.007	0.009



(a) unstructured (b) histogram



(c) periodic (d) histogram

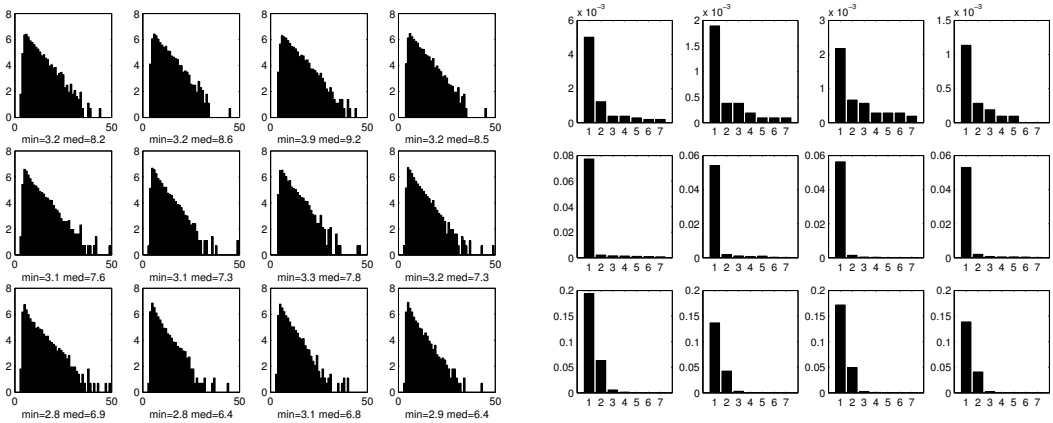
Figure 3.7: **Influence of window size.** How does the size of W_x influence the accuracy of the parametric model p_x in modeling the filter histogram? In the table the quotient KL-distance/entropy (cf. Table 3.2) is depicted for 50.000 image patches randomly selected from the van Hateren database. The bigger the images the more accurate is the fit between histogram and Laplacian. Two 50×50 image patches representative for very bad fits are depicted on the right: Unstructured areas like sky or street (Figure 3.7(a)) and areas with regular, periodic structure (Figure 3.7(c)) are most problematic.

sponding to the bars labeled "1" and "2", violations were found regularly. For patch sizes of size 30×30 or larger violations were very rare. In the segmentation experiments reported below we treated these cases as outliers, replacing κ with a default value slightly larger than 9/5. We found that this did not lead to a noticeable deterioration of segmentation quality.

Texture Synthesis Experiment

To get an intuition for which image features are captured by the generalized Laplacians we synthesized texture images using our model. For computational efficiency we did not resort to the Gibbs sampler but modified the fast pyramid-based algorithm of Heeger and Bergen [HB95] instead. This greedy algorithm enforces filter histogram similarity between a target image and a source image initialized to random noise over different scale and orientation bands of a steerable pyramid. In contrast to Heeger and Bergen we did not fit the complete filter histograms but only their generalized Laplacians. A similar approach was taken in [SLG02] where Bessel K forms and the Gibbs sampler were used to synthesize texture images from a larger number of linear filter responses.

Figure 3.9 shows some results: While our – from the viewpoint of image synthesis overly simple – method does *not* produce realistically looking textures, it appears subjectively that some discriminative information essential for image segmentation such as predominant orientation is retained.



(a) Log-histogram of κ for complete images. (b) Outlier statistics for small image patches.

Figure 3.8: **Check for pathological statistics.** Figure (a) shows the log-histogram of kurtosis κ measured over 4167 images from the van Hateren database [vHvdS98] for a steerable pyramid filter bank with three scales (rows) and four orientations (columns). Minimal and median values for κ are listed in the individual image captions. The histograms are *very regular*, and for each filter κ is well above 9/5, thus *no pathological cases* are present in the database. Figure (b) shows the relative frequency of outliers with $\kappa < 9/5$, measured over approximately 700,000 randomly sampled image *patches* of size $10^2, 20^2, 30^2, 40^2, 50^2, 75^2$, and 100^2, labeled 1 (size 10^2) to 7 (size 100^2). *Outliers are frequent with patch sizes smaller than* 30×30 *only.*

Supervised and Unsupervised Segmentation with Level Sets

To learn how our segmentation method performs on a set of standard images, we composed randomly selected textures from the BRODATZ database and arranged them in a texture collage with a cross-shaped inlay of one texture in another (Figure 3.10). We segmented 100 texture collages using (2.9) without area derivatives and *with fixed default parameters*: While in our experience the window size is an important parameter and should be chosen not too small, the choice of $\lambda_{1,2}$ is not critical. In the experiments we chose $\lambda_1 = \lambda_2 = 1$ and window size $|W| = 80 \times 80$ pixels. The texture collages were of size 512×512. For comparison, we implemented an image model based on second order statistics (cf. [ZY96, eqn. (20)]):

$$\begin{cases} k^{\mathrm{b}} = 1 \\ k^{\mathrm{in}} = \sum_i \log(\sigma^2_{\mathrm{in},i}) + \frac{(\mu_{x,i} - \mu_{\mathrm{in},i})^2}{\sigma^2_{\mathrm{in},i}} + \sigma^2_{x,i}/\sigma^2_{\mathrm{in},i} \\ k^{\mathrm{out}} = \sum_i \log(\sigma^2_{\mathrm{out},i}) + \frac{(\mu_{x,i} - \mu_{\mathrm{out},i})^2}{\sigma^2_{\mathrm{out},i}} + \sigma^2_{x,i}/\sigma^2_{\mathrm{out},i}. \end{cases} \tag{3.45}$$

This model should work well for images where the mean is the most important region descriptor (Figure 3.13(h)). Our BRODATZ-collages are of such type: The individual texture images usually are quite homogeneous, so filter response differences are likely to

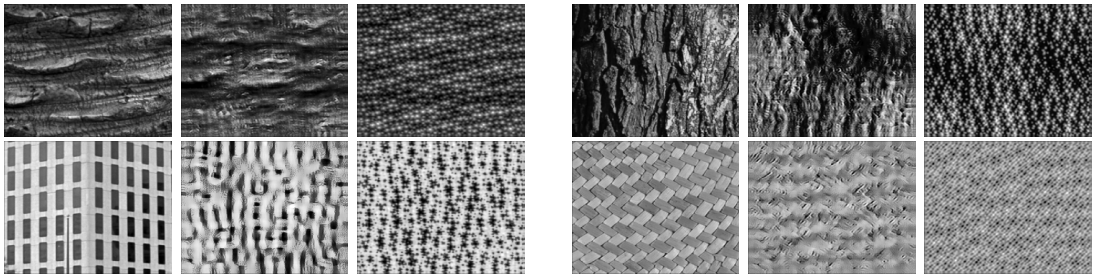


Figure 3.9: **Texture synthesis.** Textures from the VisTex database are reproduced using the histogram-based algorithm of Heeger and Bergen [HB95] and a simplified version which uses only image features captured by our model. All images were synthesized using identical filters and the same number of iterations. This illustrates how our model captures some structure of the image.

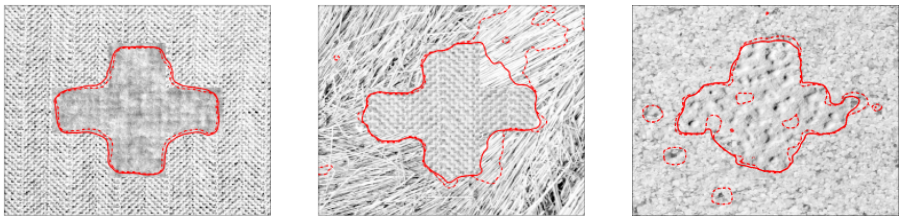


Figure 3.10: **Sample segmentations.** Brodatz texture collages segmented with KL-distance (solid line) and second order statistics (dotted line) with default parameters set. The KL-distance captures the cross-shaped inlay better than second order statistics. Here, we show some examples out of a large number of segmentation experiments, the statistics of which is given in Table 3.3. The image on the right is not successfully segmented by either method.

origin from texture boundaries.

We ran both image models for 100 iteration steps, i.e., well after we expected convergence, on each texture collage, *using the same variational framework* (Section 3.2.5) for energy minimization. For increased speed we computed the image statistics on a subsampled image and interpolated the result on the whole image. This makes the region boundaries look slightly smoother than one would otherwise expect. As both models are affected in exactly the same way this should not affect the model comparison. We finally determined the percentage of correctly segmented pixels. We found (see Table 3.3) that the average performance (median) as well as the performance on difficult images (25% quartile) of our model was significantly better than the performance of model (3.45).

We then evaluated the importance of the area derivatives, which are often omitted in variational segmentation implementations. We took the first 100 images from the

	reference model	proposed model	improvement
median	0.65	0.81	25%
q-25	0.47	0.69	47%
q-75	0.81	0.84	4%

Table 3.3: **Comparison of segmentation quality.** The percentage of correctly segmented pixels on a set of 100 randomly generated BRODATZ texture collages is reported for our model and for a reference model based on second order statistics. The median and both quartiles are shown. Our model clearly outperforms the reference model on average and shows much better performance on difficult images.

	q-10	q-90
KL-term	-4.0	2.6
Area-term	$-3.6 \cdot 10^{-5}$	$2.1 \cdot 10^{-5}$

Table 3.4: **Importance of the area term.** The 10% and the 90% quantiles of equations (3.46) and (3.47) evaluated on 100 images from the van Hateren database are reported. The contributions of the area term are five orders of magnitude smaller than the contributions of the KL-term, indicating that for our distance measure the area derivatives are negligible.

van Hateren database and computed the area derivative term from (2.9)

$$\lambda_1 \frac{\partial k^{\text{out}}}{\partial \phi} H + \lambda_2 \frac{\partial k^{\text{in}}}{\partial \phi} (1 - H) \tag{3.46}$$

for an initial segmentation consisting of equally spaced squares distributed over the whole image (Figure 3.13(a)). For comparison, we computed the KL-term

$$\lambda_1 k^{\text{out}} - \lambda_2 k^{\text{in}}, \tag{3.47}$$

and measured the influence over the whole image.

The results (Table 3.4) indicate that *for our choice of distance measure the area derivatives are negligible*. This validates common practice and allows for simpler implementations. Note, however, that this might not hold in general: Recently Jehan-Besson et al. [JBBA03] reported different results for a different choice of distance measure.

Figure 3.11 to 3.13 show some examples for supervised and unsupervised segmentation of natural images. In Figure 3.11 we examine an image from the Berkeley database [MFTM01]. The contour was initialized to equally spaced boxes. As stopping criterion we computed the improvement of the energy functional (3.28) for every time step and stopped as soon as it dropped below a previously determined threshold. The same threshold was used for all experiments. The zebra pattern is captured well by our model: The contour immediately locks onto the zebra pattern and energy (3.28) (not shown) drops sharply until the zebras are covered.

(a) t=0 (b) t=10 (c) t=20 (d) t=50

Figure 3.11: **Unsupervised segmentation.** Zebras are separated from the background. Contours were initialized to boxes, stopping was determined automatically according to $E(\phi)'$.

Figure 3.12 shows a more difficult case: A tree standing in front of a house, casting a sharp shadow on the house. With this image, unsupervised segmentation merely separates the irregular regions from the homogeneous sky and parts of the streets (Figure 3.12(e) – 3.12(h)). In contrast, if the contour is initialized in a supervised way (Figure 3.12(a)) the model captures the visually dominant tree. However, in the final segmentation (Figure 3.12(d)) relatively large parts of the shadowed house are captured as well.

In Figure 3.13 we compare our model with second order statistics (3.46) on an image from the MIT VisTex database [PGM+95]. The MDL criterion (3.25) separates the trees from the image fore- and background. This is sensible: The trees form an image region which is relatively expensive to encode while sky and grassland are comparatively homogeneous. Using one probability model for the trees and one for the rest of the image thus minimizes the expected coding length of the image. Second order statistics simply separates the bright sky from the rest of the image, yielding a less appealing segmentation.

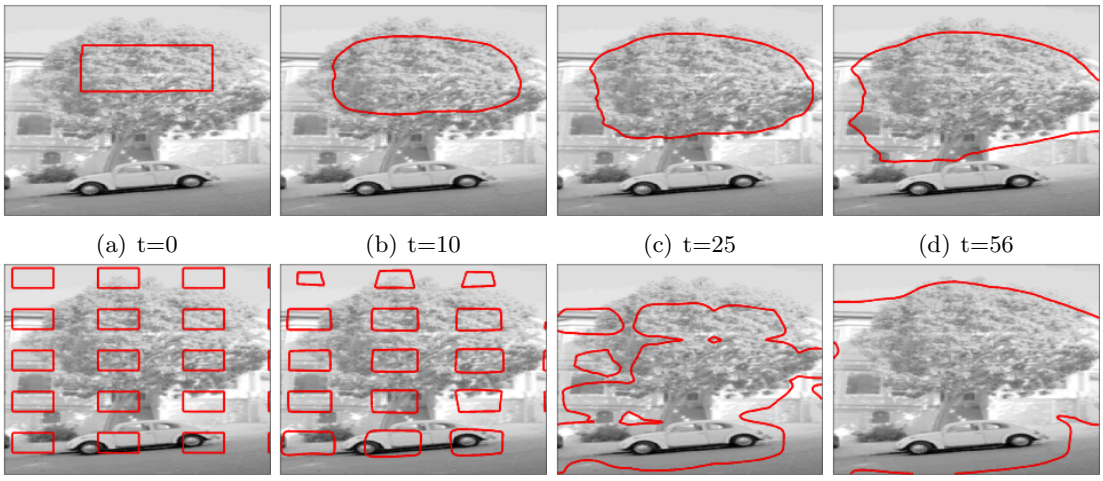
3.3 Multi-Class Segmentation

So far we were concerned with *binary* image segmentation. In this section we show how our approach generalizes to multiple classes. In this regard, a *multiphase extension* of the level set method has been suggested by Vese and Chan [VC02] which can readily be employed for minimizing our energy model.

However, performance considerations and modeling aspects make certain extensions desirable which we will discuss:

Proper boundary-length regularization. As illustrated in Figure 3.14(a), the multiphase level-set representation [VC02] may suffer from noisy boundaries due to the representation of multiple classes by only few level sets and the corresponding behavior of the boundary length regularization term (cf. Section 3.3.3).

Length discretization and fast greedy scheme. The greedy optimization scheme



(a) t=0 (b) t=10 (c) t=25 (d) t=56

(e) t=0 (f) t=5 (g) t=10 (h) t=22

Figure 3.12: **Supervised and unsupervised segmentation.** With supervised segmentation the tree is separated from house and car. Unsupervised segmentation fails in this case: Initialization of the filter response model is to unspecific, yielding a rather uninteresting segmentation into homogeneous (sky, street) and inhomogeneous regions (car, tree, house). Note the low image contrast in the lower left part of the tree.

suggested in [SC03] often exhibits good performance. However, due to length term discretization, it may get stuck in a local minimum (Figure 3.14(b)). We explain this and our solution in Section 3.3.2.

Emergence of nuisance classes. Approaches based on multiscale filter preprocessing, i.e., texture and motion analysis, commonly suffer from blurred parameter transitions at region boundaries, causing undesirable classifications (Figure 3.14(c)). We address this problem in Section 3.3.4.

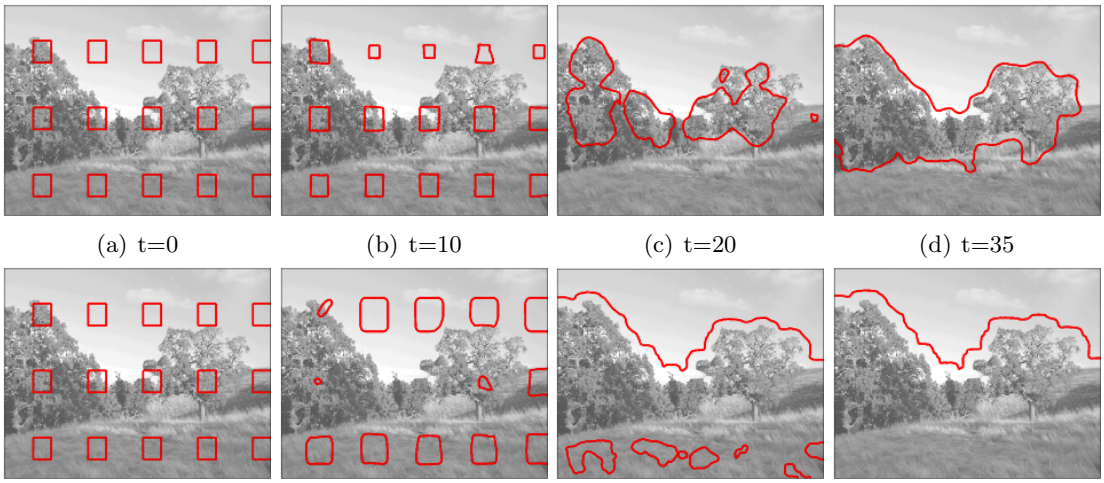
It is clear that *each* of these problems considerably hampers the design of robust and efficient segmentation schemes.

3.3.1 Multiphase Level Sets

An elegant possibility to generalize the Chan and Vese segmentation framework is to use multiple level set functions instead of just one. There are various possibilities how, precisely, to do so. In this section we review some and explain how our approach fits into the picture.

Original Model of Vese and Chan

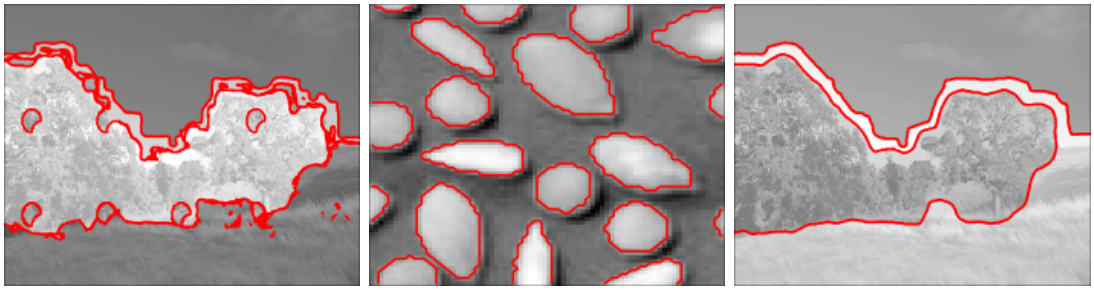
Vese and Chan extend their gradient-less image segmentation model [CV01] from binary segmentation with one level set function to 2^m-class segmentation using m level



(a) t=0 (b) t=10 (c) t=20 (d) t=35

(e) t=0 (f) t=4 (g) t=8 (h) t=12

Figure 3.13: **Unsupervised segmentation.** Unsupervised segmentation of a natural scene from the VisTex database [PGM+95]. Contours were initialized to boxes, stopping was determined automatically according to $E(\phi)'$. The contour evolution at different time steps is displayed for our model (Figure (a)–(d)) and for second order statistics (Figure (e)–(h)). The trees in the center of the image are the visually most dominant element which is reflected by the segmentation with our model. Second order statistics separates the bright sky from the darker rest of the image, failing to capture the visually dominant trees.



(a) regularization (b) early convergence (c) texture mixing

Figure 3.14: **Main technical problems addressed.** (a) Due to approximate length regularization in Vese and Chan's multiphase model *region boundaries are noisy*. (b) Caused by crude discretization in a greedy level set based optimization scheme contour evolution will always converge on a solution with *long region boundaries*. (c) *Mixing effects between adjacent textures* lead to nuisance classes at region boundaries: In the figure sky and trees are separated by a small band of grass. (See text for detailed explanations.)

set functions [VC02]. Formally, they introduce a vector of level set functions $\phi =$

$(\phi_1, \ldots, \phi_m)$ and the element-wise Heaviside step function $H : \mathbb{R}^m \to \{0,1\}^m$, $H(\phi) = (h(\phi_1), \ldots, h(\phi_m))$. Two pixels x_i and x_j belong to the same class iff $H(\phi(x_i)) = H(\phi(x_j))$. This class membership relation is encoded in 2^m characteristic functions $\chi_I : \{0,1\}^m \to \{0,1\}$ which return one for the $H(\phi(x))$ belonging to class I and zero otherwise.

For gray level image segmentation the energy functional to optimize reads

$$E = \sum_{1 \le I \le 2^m} \left[\int_\Omega (c_x - c_I)^2 \chi_I(H(\phi(x)))dx + \lambda \int_\Omega |\nabla \chi_I(H(\phi(x)))| \, dx \right]. \tag{3.48}$$

Unfortunately, this is quite intricate since the indicator functions $\chi_I : \{0,1\}^m \to \{0,1\}$ effectively compute conjunctions of thresholded level set functions:

$$\chi_1(H(\phi)) = h(\phi_1) \cdot h(\phi_2) \cdot h(\phi_3) \cdots \tag{3.49}$$

$$\chi_2(H(\phi)) = (1 - h(\phi_1)) \cdot h(\phi_2) \cdot h(\phi_3) \cdots \tag{3.50}$$

and so on. For the length term the first variation of the gradient of these functions is needed. This results in long expressions which are expensive to compute and numerically difficult to handle. As a workaround, Vese and Chan propose a *simplified length energy*

$$E_{\text{len}} = \sum_{1 \le k \le m} \int_\Omega |\nabla H(\phi_k)| \, dx. \tag{3.51}$$

This energy is easier to optimize: Instead of 2^m only m gradients must be computed, and the individual level set functions are decoupled [AK00]. For an alternative approach that uses m level set functions to represent m regions which compete *pairwise* only we refer to [BW06].

Parametric Multiphase Model

Using the notation introduced in the previous section we get a multiphase version of our parametric image energy (3.27) by

$$E = \sum_{I=1}^{n} \left[\int_\Omega D(p_x || p_I) \chi_I dx + \lambda \int_\Omega |\nabla \chi_I| \, dx \right]. \tag{3.52}$$

The first integral is a data term which captures how well point x is described by the model for region I. The second integral is proportional to the total length of the boundaries between regions and serves for regularization.

Song and Chan's Fast Optimization

Recently, Song and Chan suggested a discrete level set based optimization scheme to considerably speed up computation of image segmentations [SC03]. It is based on the

Algorithm 3.3.1 Greedy discrete optimization algorithm

Require: some initial segmentation (not necessarily informative)

1: **repeat**
2: **for all** classes I in the current segmentation **do**
3: estimate p_I using the filter response and eqn. (3.24)
4: **end for**
5: **for all** pixels x in the image **do**
6: **for all** classes I in the current segmentation **do**
7: preliminary assign class label I to pixel x
8: compute energy (3.52) locally at x using length term (3.55)
9: **end for**
10: finally assign the class label to x which yielded minimal energy in step 8
11: **end for**
12: **until** convergence

observation that energy (3.48) depends on the *sign* of the level set functions ϕ_k only. The magnitude of the level set functions has no influence.

The authors therefore propose to optimize (3.48) by greedily choosing the $\phi_k(x)$ from $\{\pm 1\}$ for each pixel x such that the energy is minimized. The length term (3.51) is discretized by finite forward differences:

$$E_{\text{len}} = \sum_{i,j,k} \sqrt{(u_{k,i+1,j} - u_{k,i,j})^2 + (u_{k,i,j+1} - u_{k,i,j})^2}, \tag{3.53}$$

where $u_{k,i,j} = H(\phi_k(x))$ represents the k-th level set function at $x = (i, j)$.

Song and Chan report that this algorithm is very efficient and, as no intricate gradients are computed, also easy to implement.

3.3.2 Greedy Variational Energy Minimization

In this section we present a discrete greedy optimization scheme to minimize energy (3.52) fast and reliably. The scheme is derived from the iterated conditional modes (ICM) algorithm [Bes86]. Of course, more recent optimization approaches would be applicable as well [BVZ01, YFW00, SSV+06]. Still, our scheme turns out to be a careful modification of the algorithm of Song and Chan (Section 3.3.1) and eliminates some major deficiencies (cf. Section 3.3.3) while preserving efficiency and performance.

We employ an EM-style algorithm (Alg. 3.3.1) to optimize energy (3.52). Starting from an initial segmentation it alternates between estimating the region densities p_I (Step 2–4) and greedily selecting image labels to minimize eqn. (3.52) (Step 5–11).

The speed improvement is significant: Typically, we need only about 5–10 iterations until a multiclass model converges as opposed to about 30–50 iterations in the original Vese

and CHAN model. As the inner loops of both algorithms are very similar this directly translates to a speedup of about 10. Of course, our straightforward level set implementation might benefit from algorithmic improvements such as implicit schemes [VC02], narrow-band methods [AS95], or AOS schemes [Wei98] which are not easily translated to the discrete setting.

A further important improvement to previous methods is that Alg. 3.3.1 directly supports an arbitrary number of classes. It is not restricted to exactly 2^m classes. This accelerates convergence and makes the model more robust as superfluous degrees of freedom are avoided.

3.3.3 Difficulties with the Boundary Term

The length term in (3.52) is crucial for good performance of our segmentation model: Since texture is an intrinsically non-local image property it changes smoothly over region boundaries. Without proper boundary regularization artifacts emerge at region boundaries (Figure 3.14(c)).

In this context, the models presented in Section 3.3.1 suffer from two problems: First, the approximation (3.51) of the length energy minimizes each level set contour length *independently*. This is *not* identical to minimizing the region boundary length of the resulting segmentation: Figure 3.14(a) shows an example where insufficient length regularization causes artifacts at the border between trees and sky. Second, discretization (3.53) is rather crude and causes segmentations to get stuck too early (Figure 3.16, first row).

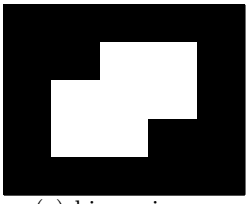
To solve the first problem we suggest computing the length term by

$$E_{\text{len}} = \sum_{i,j} \sqrt{\delta(y_{i+1,j} - y_{i,j}) + \delta(y_{i,j+1} - y_{i,j})}. \tag{3.54}$$

Here $y_{i,j}$ denotes the class label at location (i, j) in the image and $\delta(\cdot)$ is a function returning one iff its argument is zero and zero otherwise. This is a discrete version of the length term in eqn. (3.52) and a refinement of SONG and CHAN's proposal (eqn. (3.53)) which performs length regularization not on the labeled image but on individual level set functions.

While (3.54) ensures that the correct boundary length is minimized, the problem remains that contour evolution stops too early. Figure 3.15 illustrates this problem. Here, a small binary image is displayed along with the change in length energy when individual image pixels are modified. The result shows that no single change decreases length energy. Thus, no changes are made, and the image is not smoothed any further. Figure 3.16, first row, demonstrates that this effect is not restricted to artificial situations but appears in real-world scenarios.

In order to eliminate this problem we need to better estimate the gradient norm $|\nabla \chi_I|$ in (3.52), and we need to allow contours to be located in the image with sub-pixel



(a) binary image

0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	3.4	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0

(b) length pixel ← black

3.4	3.4	2.0	1.4	2.8
3.4	0.5	0.0	0.0	1.4
2.0	0.0	0.0	0.0	2.0
1.4	0.0	0.0	0.5	3.4
2.8	1.4	2.0	3.4	3.4

(c) length pixel ← white

Figure 3.15: **Problems with length discretization.** In Figure 3.15(a) a binary image is displayed. Figure 3.15(b) and (c) show the change in E_{len} according to eqn. (3.54) when the individual pixel at position (i, j) is turned black or white respectively. Note that the change in energy is non-negative over the whole picture. *Thus, no pixel is ever changed*, no matter how strong the boundary length energy is weighted.

accuracy. The first objective is achieved by using a length term

$$E_{\text{len}} = \sum_{1 \le I \le n} \int_{\Omega} |\nabla_{\varepsilon} \chi_I| \, dx, \tag{3.55}$$

where the gradient ∇_{ε} is slightly smoothed. Second, sub-pixel accuracy is achieved by locating contours not exactly on pixel boundaries, but on interpolated positions depending on the coding costs of adjacent pixels.

3.3.4 Experiments

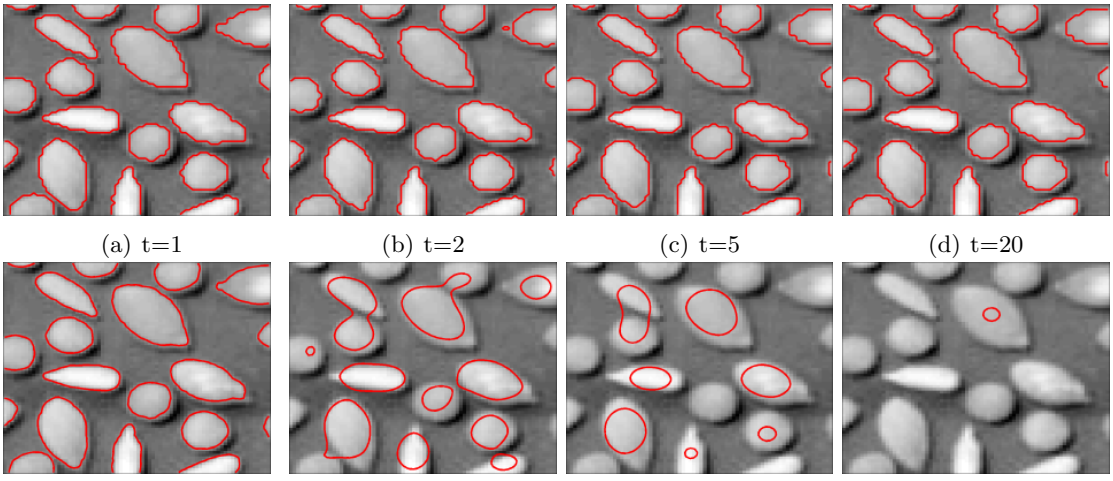
In this section we report some experiments that validate our analysis and the algorithm.

Length Regularization Term

In Figure 3.16 we show the contour evolution using the discrete formula for the length term (Figure 3.16(a) – (d)). Although a *large* length penalty of $\lambda = 100$ is chosen contours do not vanish. In contrast, with the proposed method (Figure 3.16(e) – (l)) contours look less jagged and steadily vanish with $t \to \infty$ for $\lambda = 1$ as it is expected.

Detecting Texture Boundaries

Texture-mixing effects on region boundaries regularly lead to nuisance classes causing problems when not explicitly accounted for in the model (Figure 3.14(c)). We avoid this problem by detecting texture boundaries explicitly: In our model, we compute image statistics not only for complete image windows W_x, but also for sub-windows consisting of the left and right and the upper and lower halves of W_x. If the KL-distance between two halves exceeds a predefined threshold the presence of a texture boundary is assumed. In this case the window half which fits worst to the current global models



(a) t=1 (b) t=2 (c) t=5 (d) t=20

(e) t=1 (f) t=5 (g) t=10 (h) t=20

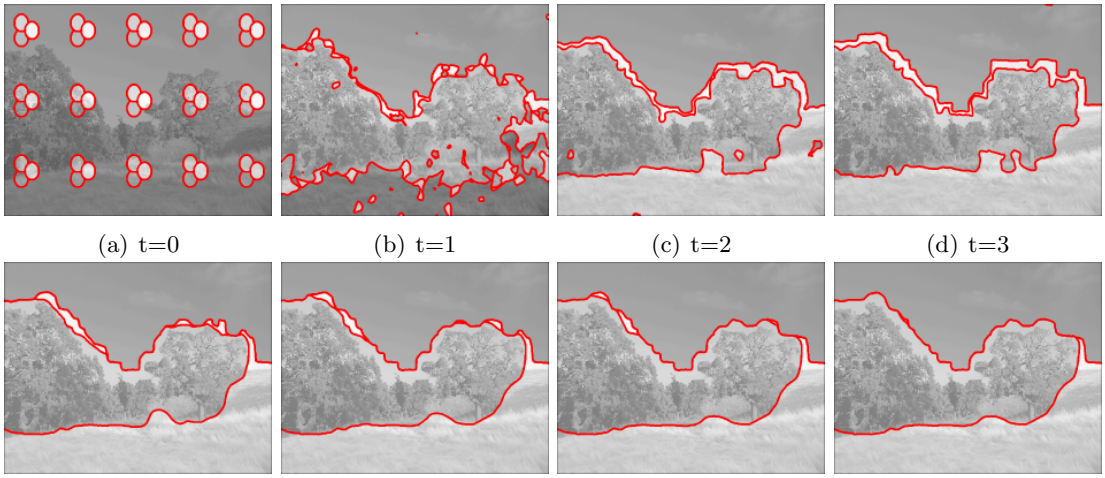
Figure 3.16: **Improved discretization of length term.** Graylevel segmentation of the Mathworks grains image using eqn. (3.54) (Figure 3.16(a) – 3.16(d)) and the proposed method (Figure 3.16(e) – 3.16(h)). We used $\lambda = 100$ (!) for the first method and $\lambda = 1$ for the proposed length term. The first length term converges after 5 iterations but gets stuck and fails to enforce proper length regularization (Figure 3.16(d)). In contrast, *the proposed method finds increasingly smoother segmentations* due to a proper boundary length regularization (bottom row).

p_I, measured in terms of KL-distance, is discarded. Together, improved length term and texture homogeneity criterion successfully eliminate misclassified texture boundaries (Figure 3.17).

3.3.5 Limitations and Further Work

The limitations of our model are illustrated in Figure 3.18, where a *completely unsupervised* segmentation starting from a random initialization fails.

In Figure 3.18(b), for instance, we need a meaningful initialization in order to find a segmentation of a van Hateren image into three classes. A completely unsupervised segmentation starting from an uninformative initialization as in Figure 3.17(a) would merely separate the right image half (trees) from the left half (street, building, sky). The same holds for the BRODATZ image collage depicted in Figure 3.18(d). In this case an uninformative initialization would completely miss the image structure which is easily recognized by a human observer. Figure 3.18(h) finally shows a case where even a meaningful initialization does not lead to satisfactory results. In this complex image the parametric model based on generalized Laplacians (eqn. (3.7)) is not discriminatory enough to distinguish the dominant objects tree, house (partly shadowed by tree), car, street, or sky.



(a) t=0 (b) t=1 (c) t=2 (d) t=3

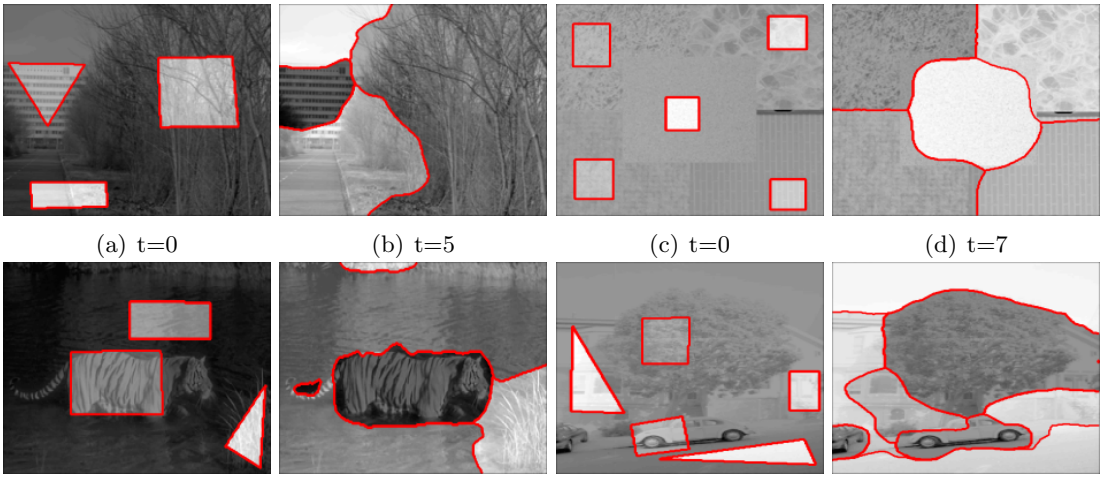
(e) t=4 (f) t=5 (g) t=6 (h) t=7

Figure 3.17: **Unsupervised segmentation.** The same scene as in Figure 3.14(a) is segmented into three classes. Depicted are seven ICM-style iterations starting from an uninformative initial configuration. After three iterations the length term is activated. Note that after the first iteration sky and grass are put into the same class (Figure 3.17(b)). One iteration later this error is corrected (Figure 3.17(c)). *The algorithm converged within seven iterations on a visually plausible segmentation.*

Concerning our statistical model, it is debatable how many more degrees of freedom are appropriate. While enhancing the descriptive power of the model, additional degrees of freedom might cause over-fitting effects in the unsupervised setting. More research is needed in this context. Images like 3.18(h), on the other hand, indicate that there are cases where integrating more complex models and corresponding prior knowledge becomes inevitable (cf. [TZ02]).

Concerning our fast optimization scheme, the reader might argue that the *greedy* strategy causes susceptibility to arbitrary initializations in situations as shown in Figure 3.18. According to our experience, this is *not* the case, however: Just like continuous level set implementations, we perform gradient descent on a well-defined objective functional and converge in most cases to a "good" local minimum. We are not aware of any rigorous approximation result of level set based optimization regarding the combinatorial problem of computing the *globally* optimal segmentation. Further work is needed to close this gap of theory.

Finally, a further limitation from our viewpoint is that our greedy scheme trivially converges in a *serial* (Gauss-Seidel) update mode, but that nothing is known about the convergence of *parallel* (Jacobian) updates. Due to the discrete formulation of our scheme, we expect that further work in this direction will elucidate possible connections between fast schemes motivated by PDE/level set representations and recent work on efficient MRF-based optimization algorithms.



(a) t=0 (b) t=5 (c) t=0 (d) t=7

(e) t=0 (f) t=7 (g) t=0 (h) t=7

Figure 3.18: **Limits and failure cases of the model.** In Figure (a) a completely unsupervised segmentation with random initialization fails. When starting with a reasonable initialization a useful segmentation into trees, street/grass/sky and building is attained within 5 iterations (Figure (b)). The BRODATZ collage and Berkeley tiger (Figure (d) and (f)) also require informative initialization. The conferta image with tree, house, and cars (Figure (h)) is not properly segmented: Its statistics are not described accurately enough by our model.

3.4 Summary

In this section we proposed a segmentation algorithm based on a parametric model for images. Using the generalized Laplacian to model filter response statistics of natural images (Section 3.1.2) we found that a simple and efficient image segmentation algorithm could be derived using a variational approach (Section 3.2). Although the image model was extremely compact and captured image statistics very approximately only (Section 3.2.8), good segmentation results have been obtained (Section 3.2.8). This suggests, that for a class of natural images, image segmentation is a surprisingly simple task, requiring only little information being processed.

On a more practical level, we extended the original binary segmentation to multiphase level sets and solve several boundary regularization problems arising with texture segmentation (Section 3.3).

Chapter 4

Image Models from Non-Negative Matrix Factorization

In the previous chapter we exploited the statistics of linear filter responses on natural images to derive a parametric image similarity measure that is useful for segmentation of a large class of images. The strengths of this approach were simplicity, statistical efficiency, and generality: In principle, the model applies to any image that looks sufficiently "natural".

However, this generality can be a burden in certain applications: Often, we already know what kind of image we will encounter, and almost always this will be a class much narrower than "natural images". We might expect images of people, of certain objects, or images coming from a given surveillance camera. In particular, we might expect images with a dominant periodic structure, such as buildings or cloth, that have a characteristic and atypical representation in the filter domain. For such applications, the general approach presented in Chapter 3 will not work optimally.

We thus need models that can be adapted to target more specific image domains. In this chapter we develop such models (Section 4.2) and their corresponding optimization algorithms (Section 4.3).

4.1 Linear Image Models

The approach we adopt in this chapter is to use linear models to represent image patches. In this section, we briefly review common methods. The key properties we are interested in are reconstruction accuracy, statistical independence properties, decomposability, and sparseness. All approaches considered have in common that by varying both, the number of basis functions to use and the algorithm for selecting them, we can tune the system to respond more specific or more general w.r.t. image classes.

4.1.1 Optimal Linear Reconstruction (PCA)

Principal components analysis, also known as *Karhunen-Loéve expansion*, is a widely established and reliable tool for visualization, statistics, and data analysis. The idea is to represent a data set $V \in \mathbb{R}^{m \times n}$ by *basis vectors* $W \in \mathbb{R}^{m \times r}$ and *coefficients* $H \in \mathbb{R}^{r \times n}$. A PCA basis is the optimal basis w.r.t. the mean-square error criterion:

$$W_{\mathrm{PCA}} = \arg\min_{W,H} \|V - WH\|_F^2 . \tag{4.1}$$

As it turns out [Fuk90, pp. 400], optimization problem (4.1) is solved by singular value decomposition of the centered covariance matrix of V. Alternatively, for large data sets, where computing the covariance matrix is impractical, direct methods can be used [Wib76, GH96, Row97, TB97]. Overall, PCA is a simple, well-motivated, and efficient tool for detecting linear structures in data.

4.1.2 Statistical Independence (ICA) and Sparseness

Independent components analysis was first introduced as a procedure to maximize *mutual-information* between input and output of non-linear information processing units [BS95]. Maximizing the flow of information through a *network* of such units as a side effect produces codes with *minimal redundancy* of the coding matrix W. This property justifies the term ICA.

Soon after the seminal information theoretic derivation of Bell and Sejnowski was published, a large body of literature appeared where different approaches toward ICA were explored (c.f. [Hyv99] for some references). In particular, *Bayesian ICA* [Mac96] was suggested as a probabilistic method which makes its underlying assumptions explicit.

The model assumes *zero noise*, such that W_{ICA} is found by maximizing

$$\begin{aligned} \max_{W,H} \quad & \prod_{ij} p(H_{ij}) \\ \text{s.t.} \quad & V = WH. \end{aligned} \tag{4.2}$$

Here, $p(H_{ij})$ is a prior probability. The original ICA algorithm is gradient ascent on the log-likelihood corresponding to (4.2) [Mac96].

Note that when p is a *sparse prior* and when we drop the zero noise assumption the *sparse coding* framework of Olshausen and Field results [OF97]. This connection, reported first in [Ols96], explains why image bases derived from both methods look strikingly similar [OF96, BS96].

4.1.3 Decomposability (NMF)

A desirable property of image codes is *decomposability* in the sense that images are *represented by parts*: Parts-based representations are inherently more robust against oc-

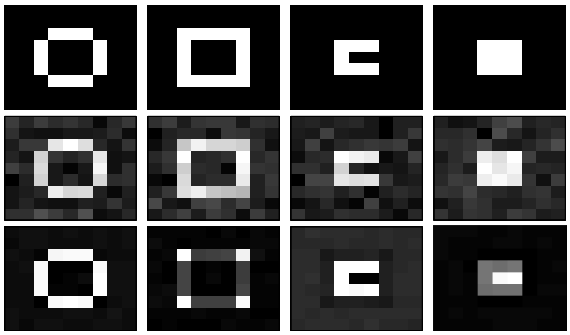


Figure 4.1: **Parts-based image representation.** Results of NMF on an artificial data set from [FJ03]. The first row shows the image primitives used. Noise corrupted versions of these were used for training (second row). NMF successfully constructs a suitable image base (third row). Note that the basis functions are *not* simply centroids in image space: They represent *parts* which must be *composed* to form the images. Thus, they are potentially more general without sacrificing image quality/sharpness.

clusion than are their global counterparts. Further, they are usually easier to understand and analyze since changes to image or model have *local* effects only.

Parts-based image representations are related, but not identical to statistically independent components. The key difference is that from a parts-based representation we expect to capture well-defined, localized, semantically relevant, "high-level" image features, while independent components can in principle be global or very simple local features. E.g., when modeling faces a parts-based representation would represent mouth, eyes, nose, etc. separately (Figure 4.2) while an ICA representation would still yield global features [BMS02, Figure 13] that have no immediate interpretation.

Regarding learning decomposable image codes we can work with image patches that are suitably selected for a given task [BU02]. This offers the benefit that images are always explained in terms of parts that occur in actual images. However, depending on the class of images considered, such methods might need a large library of parts or templates to work accurately enough.

On the other hand, we can extend the trusted PCA model (4.1) by a single constraint to encourage parts-based representations: Since images are non-negative we can force image bases and coefficients to be non-negative as well. This leads to the *non-negative matrix factorization* (NMF) problem:

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H \,. \end{aligned} \tag{4.3}$$

Restricting image bases and coefficients to be non-negative ensures that a contribution of a basis vector $W_{\bullet i}$ will not be canceled by other basis vectors' contributions. This encourages – but does not enforce – localized representations [LS99, DS04a]. E.g., in

Figure 4.1 the first two image primitives are decomposed (in frame and corners) while the second two primitives are not. Using additional sparsity constraints (Section 4.2.2) finally allows to completely rule out non-local basis functions (Figure 4.2).

4.2 Non-Negative Matrix Factorization and Extensions

In the previous introductory section we argued that non-negative constraints are a promising direction toward localized, parts-based image models. In this section we explore this direction further and propose various useful extensions to the basic NMF model.

NMF was originally introduced to model processes in the physical sciences [SI89, PT94]. In recent years, it has become increasingly popular in machine learning, signal processing, and computer vision as well [XLG03, HH02, SB03].

4.2.1 The Basic Model

As discussed above, the original NMF problem reads (Section 4.1.3, eqn. (4.3))

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H \,, \end{aligned}$$

and essentially is a principal components model where basis and coefficients are restricted to the non-negative cone. Although there are algorithms that compute the *global* optimum for such problems [FV93], these algorithms do not yet scale up to the large scale problems common in, e.g., machine learning, computer vision, or engineering. As a result, we will confine ourselves to *efficiently* compute a *local* optimum by solving a sequence of convex programs (Section 4.3.2).

4.2.2 Sparsity Control

Although NMF codes tend to be sparse [LS99], it has been suggested to control sparsity by more direct means. To this end, Hoyer [Hoy04] proposed recently to use the following sparseness measure for vectors $x \in \mathbb{R}_+^n$, $x \neq 0$:

$$\mathrm{sp}(x) := \frac{1}{\sqrt{n}-1}\left(\sqrt{n} - \frac{\|x\|_1}{\|x\|_2}\right). \tag{4.4}$$

Because of the relations:

$$\frac{1}{\sqrt{n}}\|x\|_1 \leq \|x\|_2 \leq \|x\|_1 \,, \tag{4.5}$$

the latter being a consequence of the Cauchy-Schwarz inequality, the sparseness measure is bounded:

$$0 \leq \mathrm{sp}(x) \leq 1 \,. \tag{4.6}$$

Figure 4.2: **Motivation of NMF with sparseness constraints.** Five basis functions (columns) with sparseness constraints ranging from 0.1 (first row, left block) to 0.8 (last row, right block) on W were trained on the CBCL face database. A moderate amount of sparseness encourages localized, visually meaningful base functions.

The bounds are attained for minimal sparse vectors with equal non-zero components where $\mathrm{sp}(x) = 0$ and for maximal sparse vectors with all but one vanishing components where $\mathrm{sp}(x) = 1$. Where appropriate, we will also write $\mathrm{sp}(M) \in \mathbb{R}^n$, meaning $\mathrm{sp}(\cdot)$ is applied to each column of matrix $M \in \mathbb{R}^{m \times n}$ and the results are stacked in a column vector.

Using this sparseness measure, the following constrained NMF problem was proposed in [Hoy04]:

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H \\ & \mathrm{sp}(W) = s_w \\ & \mathrm{sp}(H^\top) = s_h \,, \end{aligned} \tag{4.7}$$

where s_w, s_h are user parameters. The sparsity constraints control (i) to what extent basis functions are sparse, and (ii) how much each basis function contributes to the reconstruction of only a subset of the data V. In a pattern recognition application the sparsity constraints effectively weight the desired generality over the specificity of the basis functions.

Instead using of equality constraints, however, we will slightly generalize the constraints in this work to $s_w^{\min} \leq \mathrm{sp}(W) \leq s_w^{\max}$ and $s_h^{\min} \leq \mathrm{sp}(H^\top) \leq s_h^{\max}$. This disburdens the user from choosing exact parameter values s_w, s_h, which can be difficult to find in realistic scenarios. In particular, it allows for $s_h^{\max} = s_w^{\max} = 1$, which may often be useful.

Consequently, we define the *sparsity-constrained NMF problem* as follows:

$$\begin{aligned} \min_{0 \le W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & s_w^{\min} \le \mathrm{sp}(W) \le s_w^{\max} \\ & s_h^{\min} \le \mathrm{sp}(H^\top) \le s_h^{\max}, \end{aligned} \tag{4.8}$$

where $s_w^{\min}, s_w^{\max}, s_h^{\min}, s_h^{\max}$ are user parameters. See Figure 4.2 and Section 4.4 for examples.

Finally, note that the sparsity parameters need not necessarily be uniform for all entries in W or H: By choosing more stringent parameters for some basis functions than for others one can encourage a decomposition into global and local features at the same time. This resembles a multiscale approach to image coding. Of course, which specific parameters to choose depends strongly on the application at hand.

4.2.3 Soft Sparsity Constraints

The formulation (4.8) is often convenient for the user since it guarantees that sparseness constraints are enforced strictly and accurately. In some situations, however, the user may have no idea about good choices for the sparsity parameters. Instead of running elaborate and computationally expensive crossvalidation computations it might be preferred to specify a sparsity prior *in the objective function*. This way, one will in general loose any strict guarantee of the resulting sparseness, but it allows to *automatically balance achieved sparseness and reconstruction error*. The corresponding optimization problem is

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 + \lambda_h e^\top \mathrm{sp}(H^\top) + \lambda_w e^\top \mathrm{sp}(W) \\ \text{s.t.} \quad & 0 \le W, H, \end{aligned} \tag{4.9}$$

where $\lambda_{w,h}$ weight the relative importance of sparseness over the reconstruction error. Of course, λ_w and λ_h have to be specified by the user as well. However, in (4.9) the sparseness prior is always active while in (4.8) a overly lax sparseness constraint might not influence the resulting factorization at all. Overall, it depends strongly on the concrete application at hand which formulation is preferred.

4.2.4 Prior Knowledge

When NMF bases are used for recognition, it can be beneficial to introduce information about class membership in the training process. Doing so encourages NMF codes that not only describe the input data well, but also allow for good discrimination in a subsequent classification stage. We propose a formulation, similar to Fisher-NMF [WJHT05], that leads to particularly efficient algorithms in the training stage.

The basic idea is to restrict, for each class i and for each of its vectors j, the coefficients $H_{j\bullet}$ to a cone around the class center μ_i:

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H \qquad (4.10a) \\ & \|\mu_i - H_{j\bullet}\|_2 \leq \lambda \|\mu_i\|_1 \quad \forall i, \forall j \in \text{class}(i). \qquad (4.10b) \end{aligned}$$

Note that μ_i depends on H and is implicitly determined through the optimization process. As will be explained in Section 4.3, these additional constraints are no more difficult from the viewpoint of optimization than are the previously introduced constraints in (4.8). On the other hand, they offer greatly increased classification performance for some problems (Section 4.4). Of course, if the application suggests, supervised NMF (4.10) can be conducted with the additional sparsity constraints from (4.8).

4.2.5 Sparse PCA

For data that is non-negative by nature, e.g., image data, certain physical properties, probabilities, or equities, NMF is particularly well-suited. However, in situations where negative values occur we want to allow for negative bases and coefficient vectors as well. This leads to a sparsity-controlled setting similar to PCA [dGJL04, ZHT05, CJ01]. In particular, the problem considered reads

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & s_w^{\min} \leq \text{sp}(W) \leq s_w^{\max} \\ & s_h^{\min} \leq \text{sp}(H^\top) \leq s_h^{\max}, \end{aligned} \qquad (4.11)$$

which equals (4.8) except for the non-negativity constraints that are omitted.

4.2.6 Transformation Invariance

Much variation in images is due to perspective and affine transformations. For recognition we want to ignore such variation. A common solution is to integrate over the corresponding transformation. The problem then reads:

$$\begin{aligned} \min_{W,H,\theta} \quad & \|T_\theta(V) - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H. \end{aligned} \qquad (4.12)$$

where T is an operator, parametrized by θ, mapping V to transformed images.

4.2.7 Missing Values

In some applications missing data is a common annoyance. We model this by a binary $m \times n$ matrix E, s.t. E_{ij} is 0 if and only if the value V_{ij} is missing. The NMF optimization problem reads accordingly:

$$\begin{aligned} \min_{W,H} \quad & \|E \odot (V - WH)\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H. \end{aligned} \tag{4.13}$$

Recall that $\odot$ is the element-wise matrix product. Thus, hidden values are simply ignored in the objective function. This is similar to Wiberg's approach for PCA with missing values [Wib76].

4.3 Solving NMF Problems

In the previous sections, we presented various optimization problems, completely ignoring if and how they can be solved. Since most NMF problems are intricate non-convex problems it is not obvious that the models can be optimized efficiently. Fortunately, it turns out that the problems above are highly structured, so that an elegant geometric description can be found and efficient and robust algorithms can be devised.

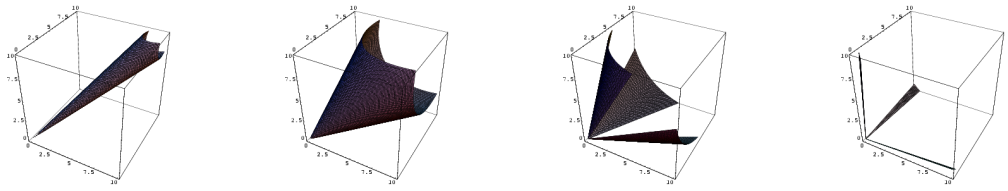
In this section we translate the sparsity-constrained NMF problems into the framework of second-order cone programming (Section 2.2.2). Within this framework, we develop sequential convex programs that solve the NMF problems quickly and to high accuracy.

4.3.1 Assumptions

Throughout this section we make the following assumptions:

1. The matrices $W^\top W$ and $HH^\top$ are positive definite.
2. $s_h^{\min} < s_h^{\max}$ and $s_w^{\min} < s_w^{\max}$ in (4.8).
3. The min-sparsity constraints in (4.8) are essential in the sense that all global optima of the problem without min-sparsity constraints violate at least one constraint on W and H.

The first assumption is introduced to simplify reasoning about convergence. In applications, it will regularly be satisfied as long as the number of basis functions r does not exceed size or dimension of the training data: $r \leq m, n$. Assumption two has been discussed above in connection with (4.8). Finally, assumption three is natural, because without the min-sparsity constraint problem (4.8) would essentially correspond to (4.3) which can be solved relatively easily.



(a) $s = 0.1$ (b) $s = 0.2$ (c) $s = 0.5$ (d) $s = 0.9$

Figure 4.3: Examples of sparsity cones in $\mathbb{R}^3$. As s increases the sparsity cone widens and intersects with the boundaries of the non-negative orthant.

4.3.2 Second Order Cone Programming and Sparsity

In this section we show how our sparsity measure relates to second order cones. Through their close link algorithms based on convex programming become useful for sparsity-controlled NMF.

On the non-negative cone we can model the sparseness-measure (4.4) using the family of *convex* sets parametrized by sparsity-parameter $s \in [0, 1]$:

$$\mathcal{C}(s) := \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{pmatrix} x \\ e^\top x / c_{n,s} \end{pmatrix} \in \mathcal{L}^{n+1} \right\} , \quad c_{n,s} := \sqrt{n} - (\sqrt{n} - 1)s. \tag{4.14}$$

Inserting the bounds $0 \leq \mathrm{sp}(x) \leq 1$ for s, we obtain

$$\mathcal{C}(0) = \left\{ \lambda e \,,\; 0 < \lambda \in \mathbb{R} \right\} \qquad \text{and} \qquad \mathbb{R}^n_+ \subset \mathcal{C}(1). \tag{4.15}$$

This raises the question as to when we must impose non-negativity constraints explicitly.

Proposition 3. *The set $\mathcal{C}(s)$ contains non-positive vectors $x \neq 0$ if:*

$$\frac{\sqrt{n} - \sqrt{n-1}}{\sqrt{n} - 1} < s \leq 1 \,, \quad n \geq 3 \tag{4.16}$$

Proof. We observe that if $x \in \mathcal{C}(s)$, then $\lambda x \in \mathcal{C}(s)$ for arbitrary $0 < \lambda \in \mathbb{R}$, because $\|\lambda x\|_2 - e^\top(\lambda x)/c_{n,s} = \lambda(\|x\|_2 - e^\top x/c_{n,s}) \leq 0$. Hence it suffices to consider vectors x with $\|x\|_2 = 1$. According to definition (4.14), such vectors tend to be in $\mathcal{C}(s)$ the more they are aligned with e. Therefore, w.l.o.g. put $x_n = 0$ and $x_i = (n-1)^{-1/2}$, $i = 1, \ldots, n-1$. Then $x \in \mathcal{C}(s)$ if $c_{n,s} < \sqrt{n-1}$, and the result follows from the definition of $c_{n,s}$ in (4.14). Finally, for $n = 2$ the lower bound for s equals 1, i.e., no non-positive vectors exist for all admissible values of s. □

This argument shows that (cf. Figure 4.3):

$$\mathcal{C}(s') \subseteq \mathcal{C}(s) \quad \text{for} \quad s' \leq s. \tag{4.17}$$

Therefore, to represent the feasible set of problem (4.7), we combine the convex non-negativity condition with the convex upper bound constraint:

$$\left\{ x \in \mathbb{R}^n_+ \;\middle|\; \mathrm{sp}(x) \leq s \right\} \;=\; \mathbb{R}^n_+ \cap \mathcal{C}(s) \,, \tag{4.18}$$

and impose the *non-convex* lower bound constraint by subsequently removing $\mathcal{C}(s')$:

$$\left\{x \in \mathbb{R}^n_+ \;\middle|\; s' \leq \mathrm{sp}(x) \leq s\,,\; s' < s\right\} \;=\; \left(\mathbb{R}^n_+ \cap \mathcal{C}(s)\right) \setminus \mathcal{C}(s')\,. \tag{4.19}$$

To reformulate (4.7), we define accordingly, based on (4.14):

$$\mathcal{C}_w(s) := \left\{W \in \mathbb{R}^{m\times r} \;\middle|\; W_{\bullet i} \in \mathcal{C}(s)\,,\; i = 1,\ldots,r\right\}, \tag{4.20}$$
$$\mathcal{C}_h(s) := \left\{H \in \mathbb{R}^{r\times n} \;\middle|\; H_{i\bullet} \in \mathcal{C}(s)\,,\; i = 1,\ldots,r\right\}. \tag{4.21}$$

As a result, the sparsity-constrained NMF problem (4.7) now reads:

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & W \in \left(\mathbb{R}^{m\times r}_+ \cap \mathcal{C}_w(s_w^{\max})\right) \setminus \mathcal{C}_w(s_w^{\min}) \\ & H \in \left(\mathbb{R}^{r\times n}_+ \cap \mathcal{C}_h(s_h^{\max})\right) \setminus \mathcal{C}_h(s_h^{\min})\,. \end{aligned} \tag{4.22}$$

This formulation makes explicit that enforcing sparse NMF solutions introduces a single additional *reverse-convex* constraint for W and H, respectively. Consequently, not only the joint optimization of W, H is non-convex, but individual optimization of W and H are also.

4.3.3 Optimality Conditions

We state the first-order optimality conditions for problem (4.22).

To this end, we define in view of (4.14) and (4.22):

$$f(W,H) := \|V - WH\|^2 \tag{4.23a}$$
$$Q := Q_w \times Q_h, \quad Q_w := \mathbb{R}^{m\times r}_+ \cap \mathcal{C}_w(s_w^{\max}), \quad Q_h := \mathbb{R}^{r\times n}_+ \cap \mathcal{C}_h(s_h^{\max}) \tag{4.23b}$$
$$G_w(W) := \left(\|W_{\bullet 1}\|_2 - \frac{1}{c_{n,s_w^{\min}}}\|W_{\bullet 1}\|_1\;,\;\ldots\;,\;\|W_{\bullet r}\|_2 - \frac{1}{c_{n,s_w^{\min}}}\|W_{\bullet r}\|_1\right)^\top \tag{4.23c}$$
$$G_h(H) := \left(\|H_{1\bullet}\|_2 - \frac{1}{c_{n,s_h^{\min}}}\|H_{1\bullet}\|_1\;,\;\ldots\;,\;\|H_{r\bullet}\|_2 - \frac{1}{c_{n,s_h^{\min}}}\|H_{r\bullet}\|_1\right)^\top. \tag{4.23d}$$

Note that $G_w(W)$ and $G_h(H)$ are non-negative exactly when sparsity is at least $s_w^{\min}$ and $s_h^{\min}$. For non-negative W and H computing the ℓ_1 norm is a linear operation, that is, $W \geq 0 \Rightarrow \|W_{\bullet i}\|_1 \equiv \langle W_{\bullet i}, e\rangle$.

Problem (4.22) then can be rewritten in standard form [RW98]:

$$\min_{(W,H)\in Q} f(W,H)\,, \quad G_w(W) \in \mathbb{R}^r_+\,, \quad G_h(H) \in \mathbb{R}^r_+ \tag{4.24}$$

With the corresponding Lagrangian L and multipliers λ_w, λ_h

$$L(W,H,\lambda_w,\lambda_h) = f(W,H) + \lambda_w^\top G_w(W) + \lambda_h^\top G_h(H) \tag{4.25}$$

the first-order conditions for a locally optimal point (W^*, H^*) are:

$$-\left(\frac{\partial L}{\partial W}, \frac{\partial L}{\partial H}\right)^\top \in N_Q(W^*, H^*) = N_{Q_w}(W^*) \times N_{Q_h}(H^*) \tag{4.26a}$$

$$G_w(W^*) \in \mathbb{R}^r_+ \,,\; G_h(H^*) \in \mathbb{R}^r_+ \tag{4.26b}$$

$$\lambda^*_w, \lambda^*_h \in \mathbb{R}^r_- \tag{4.26c}$$

$$\langle \lambda^*_w, G_w(W^*) \rangle = 0 \,,\; \langle \lambda^*_h, G_h(H^*) \rangle = 0 \,, \tag{4.26d}$$

where $N_X(x)$ denotes the normal cone to a set X at point x.

4.3.4 NMF by Quadratic Programming

Before we address the difficult multiply-constrained NMF problems let us first examine the basic form. Recall, that the original NMF problem (4.3) reads

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H. \end{aligned}$$

When we fix W and expand the objective function we obtain:

$$\begin{aligned} \|V - WH\|_F^2 &= \operatorname{tr}\big[(V - WH)^\top (V - WH)\big] \\ &= \operatorname{tr}(H^\top W^\top W H) - 2\operatorname{tr}(V^\top W H) + \operatorname{tr}(V^\top V) \end{aligned}$$

Together with the non-negativity constraints $H \geq 0$, this amounts to solving the QPs:

$$\mathrm{QP}(W^\top W, W^\top V_{\bullet i}) \,, \quad i = 1, \ldots n \tag{4.27}$$

for $H_{\bullet 1}, \ldots, H_{\bullet n}$. Conversely, fixing H yields:

$$\|V - WH\|_F^2 = \operatorname{tr}(WHH^\top W^\top) - 2\operatorname{tr}(VH^\top W^\top) + \operatorname{tr}(V^\top V) \,, \tag{4.28}$$

which leads to the QPs:

$$\mathrm{QP}(HH^\top, HV_{i\bullet}) \,, \quad i = 1, \ldots, m \,, \tag{4.29}$$

for $W_{1\bullet}, \ldots, W_{m\bullet}$.

We emphasize that by using a batch-processing scheme *problems of almost arbitrary size* can be handled this way. The only limitation is the number of basis vectors r, the dimension of the basis vectors m is irrelevant. This is particularly important for image and video processing applications where m represents the number of pixels which can be very large.

The algorithm is summarized in Alg. 4.3.1. Note, that *the same* target function (4.3) is optimized alternately with respect to H and W. As a result, the algorithm performs a *block coordinate descent* (cf. [Ber99]). Furthermore, we may assume that the QPs in (4.27) and (4.29) are strictly convex, because typically $r \ll m, n$.

Algorithm 4.3.1 QP-based NMF algorithm in pseudocode.

initialize $W^0,\ H^0 \geq 0$ randomly, $k \leftarrow 0$
repeat
  $H^{k+1} \leftarrow \text{QP-result}(W^k, V)$
  $W^{k+1} \leftarrow \text{QP-result}(H^{k+1}, V)$
  $k \leftarrow k+1$
until $\left| \left\|V - W^{k-1}H^{k-1}\right\|_F - \left\|V - W^kH^k\right\|_F \right| \leq \epsilon$

Proposition 4. *Under the assumptions of Section 4.3.1, Algorithm 4.3.1 converges to a local minimum of problem* (4.3).

Proof. See [Ber99, Prop. 2.7.1] □

4.3.5 Projected Gradient Descent

The first algorithm proposed for solving sparsity-constrained NMF was based on *projected gradient descent* [Hoy04]. It alternately optimizes for W and H by moving into negative gradient direction and subsequently projecting the solution on the feasible set. Using the shorthand $f(W,H) \equiv \|V - WH\|_F^2$ this reads

$$\begin{aligned} H^{k+1} &\leftarrow \pi[H^k - \alpha_h \nabla_H f(W^k, H^k)] \\ W^{k+1} &\leftarrow \pi[W^k - \alpha_w \nabla_W f(W^k, H^{k+1})]. \end{aligned} \tag{4.30}$$

The projection π is performed by solving a sequence of quadratic equations that fix the ℓ_2-norm of the vectors and change the ℓ_1-norm appropriately to achieve the desired sparseness.

Unfortunately, the stepsize α of the gradient descent scheme is not specified in [Hoy04], so it is difficult to comment on convergence. It is well known that projected gradient descent schemes can fail to converge to stationary points even in the case of convex constraints [Wol72, Min86]. Empirically, the algorithm in [Hoy04] seems to converge to good local optima, although sometimes at a very slow pace (Section 4.4.1).

4.3.6 Tangent-Plane Approach

In this section, we present an optimization scheme for sparsity-controlled NMF which relies on linear approximation of the reverse-convex constraint in (4.22). As in the case of unconstrained NMF, we alternately minimize (4.22) with respect to W and H. It thus suffices to concentrate on the H-step:

$$\begin{aligned} \min_H \quad & f(H) = \|V - WH\|_F^2 \\ \text{s.t.} \quad & H \in \left(\mathbb{R}_+^{r\times n} \cap \mathcal{C}_h(s_h^{\max})\right) \setminus \mathcal{C}_h(s_h^{\min}) \,. \end{aligned} \tag{4.31}$$

Recall the assumptions made in Section 4.3.1.

The Tangent Plane Constraints (TPC) Algorithm

The tangent-plane constraint algorithm solves a sequence of SOCPs where the convex max-sparsity constraints are modeled as second order cones and the min-sparsity cone is linearized: In an initialization step we solve SOCP ignoring the min-sparsity constrain and examine the solution. For the rows of H that violate the min-sparsity contraint we compute tangent planes to the min-sparsity cone and solve the SOCP again with additional tangent-plane constraints in place. This is repeated until all necessary tangent-planes are identified. During iteration we repeatedly solve this SOCP where the tangent planes are permanently updated to follow their corresponding entries in H: This ensures that they constrain the feasible set no more than necessary. This process of updating the tangent planes and computing new estimates for H is repeated until the objective function no longer improves.

In detail, the TPC algorithm consists of the following steps:

Initialization. The algorithm starts by setting $s_h^{\min} = 0$ in (4.31), and by computing the global optimum of the convex problem: $\min f(H),\ H \in \mathcal{C}_h(s_h^{\max})$, denoted by $\tilde{H}^0$. Rewriting the objective function:

$$\begin{aligned} f(H) &= \left\| V^\top - H^\top W^\top \right\|_F^2 \\ &= \left\| \mathrm{vec}(V^\top) - (W \otimes I)\mathrm{vec}(H^\top) \right\|_2^2 , \end{aligned} \tag{4.32}$$

we observe that $\tilde{H}^0$ solves the SOCP:

$$\min_{H,z} z\ , \quad H \in \mathbb{R}_+^{r\times n} \cap \mathcal{C}_h(s_h^{\max})\ , \quad \begin{pmatrix} \mathrm{vec}(V^\top) - (W \otimes I)\mathrm{vec}(H^\top) \\ z \end{pmatrix} \in \mathcal{L}^{r\times n+1} \tag{4.33}$$

Note that $\tilde{H}^0$ will be infeasible w.r.t. the original problem because the reverse-convex constraint of (4.31) is not imposed in (4.33). We determine the index set $J^0 \subseteq \{1,\dots,r\}$ of those vectors $\tilde{H}^0_{j\bullet}$ violating the reverse-convex constraint, that is $\tilde{H}^0_{j\bullet} \in \mathcal{C}(s_h^{\min})$.
Let $\pi(\tilde{H}^0_{j\bullet})$ denote the projections of $\tilde{H}^0_{j\bullet}$ onto $\partial\mathcal{C}(s_h^{\min})$, $\forall j \in J^0$. Further, let t_j^0 denote the tangent plane normals to $\mathcal{C}_h(s_h^{\min})$ at these points, and $H^0 \leftarrow \pi(\tilde{H}^0)$ a feasible starting point. We initialize the iteration counter $k \leftarrow 0$.

Iteration. Given J^k, $k = 0, 1, 2, \dots$, we once more solve (4.33) with additional linear constraints enforcing feasibility of each $H^k_{j\bullet}$, $j \in J^k$:

$$\begin{aligned} \min_{H,z} z\ , \quad & H \in \mathbb{R}_+^{r\times n} \cap \mathcal{C}_h(s_h^{\max})\ , \quad \begin{pmatrix} \mathrm{vec}(V^\top) - (W \otimes I)\mathrm{vec}(H^\top) \\ z \end{pmatrix} \in \mathcal{L}^{r\times n+1} \\ & \left\langle t_j^k, H_{j\bullet} - \pi(H^k_{j\bullet}) \right\rangle \geq 0\ , \quad \forall j \in J^k \end{aligned} \tag{4.34}$$

Algorithm 4.3.2 Tangent-plane approximation algorithm in pseudocode.

$H^0 \leftarrow$ solution of (4.33), $J^0 \leftarrow \emptyset$, $k \leftarrow 0$
repeat
  $\tilde{H}^k \leftarrow H^k$
  repeat
    $J^k \leftarrow J^k \cup \{j \in 1, \ldots, r : \tilde{H}^k_{j\bullet} \in \mathcal{C}(s_h^{\min})\}$
    $t_j^k \leftarrow \nabla \mathcal{C}_h(s_h^{\min})(\pi(\tilde{H}^k_{j\bullet})) \quad \forall j \in J^k$
    $\tilde{H}^k \leftarrow$ solution of (4.34) replacing H^k by $\tilde{H}^k$
  until $\tilde{H}^k$ is feasible
  $H^{k+1} \leftarrow \tilde{H}^k,\ J^{k+1} \leftarrow J^k,\ k \leftarrow k+1$
until $\left|f(H^k) - f(H^{k-1})\right| \leq \epsilon$

Let us denote the solution by $\tilde{H}^{k+1}$. It may occur that because of the additional constraints new rows $\tilde{H}^{k+1}_{j\bullet}$ of $\tilde{H}^{k+1}$ became infeasible for indices $j \notin J^k$. In this case we augment J^k accordingly, and solve (4.34) again until the solution is feasible.

Finally, we rectify the vectors $\tilde{H}^{k+1}_{j\bullet}$ by projection, as in the initialization, provided this further minimizes the objective function f. The result is denoted by H^{k+1}, and the corresponding index set by J^{k+1}. At last, we increment the iteration counter: $k \leftarrow k+1$

Termination criterion. We check whether H^{k+1} satisfies the termination criterion $\left|f(H^{k+1}) - f(H^k)\right| < \epsilon$. If not, we continue the iteration.

The algorithm is summarized in pseudocode in Alg. 4.3.2.

Convergence Properties

In the following discussion we use matrices $T^k = (t_j^k)_{j \in 1,\ldots,r}$ that have tangent plane vector t_j^k as j-th column when $j \in J^k$ and zeros elsewhere.

Proposition 5. *Under the assumptions stated in Section 4.3.1 Algorithm 4.3.2 yields a sequence $H^1, H^2, \ldots$ of feasible points, every cluster point of which is a local optimum.*

Proof. Our proof follows [Tuy87, Prop. 3.2]. First, note that for every $k > 0$ the solution H^k of iteration k is a feasible point for the SOCP solved in iteration $k+1$. Therefore, $\{f(H^k)\}_{k=1,\ldots}$ is a decreasing sequence, bounded from below and thus convergent. By assumption, no column of $W \geq 0$ equals the zero vector. Then, $\{H : f(H) \leq f(H^k)\}$ is bounded for each k. Consequently, the sequence $\{H^k\}_{k=1,\ldots}$ of solutions of (4.34) and the corresponding sequences $\{T^k\}_{k=1,\ldots}$ of tangent planes are bounded and contain converging subsequences. Let $\{H^{k_\nu}\}_{\nu=1,\ldots}$ and $\{T^{k_\nu}\}_{\nu=1,\ldots}$ be subsequences converging to cluster points $\bar{H}$ and $\bar{T}$.

Because H^{k_ν} is the global solution of a convex program we have

$$f(H^{k_\nu}) \leq f(H), \qquad \forall H \in \mathcal{C}(s_h^{\max}) \text{ with } T^{k_\nu\top} H \geq 0, \tag{4.35}$$

and in the limit $\nu \to \infty$

$$f(\bar{H}) \leq f(H), \qquad \forall H \in \mathcal{C}(s_h^{\max}) \text{ with } \bar{T}^\top H \geq 0. \tag{4.36}$$

We assumed the tangent plane constraints are regular. Then the constraints active in $\bar{T}$ correspond to entries $\bar{H}_{j\bullet} \in \partial\mathcal{C}_h(s_h^{\min}), j \in J$. According to (4.36) there is no feasible descent direction at $\bar{H}$ and, thus, it must be a stationary point. Since the target function is quadratic positive-semidefinite by assumption, $\bar{H}$ is an optimum. □

Thus, the TPC algorithm yields locally optimal W and H. However, this holds for the *individual* optimizations of W and H only. The same cannot be claimed for the *alternating sequence* of optimizations in W and H necessary to solve (4.8). Because of the intervening optimization of, e.g., W, we cannot derive a bound on $f(H)$ from a previously found locally optimal H. In rare cases, this can lead to undesirable oscillations. When this happens, we must introduce some damping term or simply switch to the convergent sparsity maximization algorithm described in Section 4.3.7.

On the other hand, if the TPC algorithm converges it does in fact yield a locally optimal solution.

Proposition 6. *If the TPC algorithm converges to a point (W^*, H^*) and the assumptions stated in Section 4.3.1 hold then (W^*, H^*) satisfies the first-order necessary optimality conditions 4.3.3 of problem* (4.22)*.*

Proof. For H^* we have from (4.34) using the notation from Section 4.3.3

$$H^* = \arg\min_{H \in Q_h} \quad \|V - W^* H\|^2 \tag{4.37a}$$

$$\text{s.t.} \quad \left\langle t_j^k,\ H_{j\bullet} - \pi(H_{j\bullet}^{*k}) \right\rangle \geq 0\,, \quad \forall j \in J^k. \tag{4.37b}$$

Since $t_j^k = \nabla\mathrm{sp}(\pi(H_{j\bullet}^{*k})^\top)$ constraint (4.37b) ensures that the min-sparsity constraint is enforced at H^* when necessary. Introducing Lagrange parameters $\lambda_f^*,\ \tilde{\lambda}_h^*$ for this convex problem yields that the result of (4.37) adheres to the first-order condition

$$\begin{aligned} -\lambda_f^* \frac{\partial}{\partial H} f(W^*, H^*) - \tilde{\lambda}_h^* \frac{\partial}{\partial H} \nabla\mathrm{sp}(\pi(H^{*k})^\top)(H^* - \pi(H^{*k})) \ &\in\ N_{Q_h}(H^*) \\ \Leftrightarrow \quad -\lambda_f^* \frac{\partial}{\partial H} f(W^*, H^*) - \tilde{\lambda}_h^* \nabla\mathrm{sp}(\pi(H^{*k})^\top) \ &\in\ N_{Q_h}(H^*) \\ \Leftrightarrow \quad -\frac{\partial}{\partial H}\Big(\lambda_f^* f(H^*) + \langle \tilde{\lambda}_h^*, G_h(H^*)\rangle\Big) \ &\in\ N_{Q_h}(H^*) \end{aligned} \tag{4.38}$$

which coincides with the condition on H in (4.26a). The W-part can be treated in the same way. □

Remarks

Problems (4.33), (4.34) are formulated in terms of the *rows* of H, complying with the sparsity constraints (4.21). Unfortunatly, matrix $W \otimes I$ in (4.33) is not block-diagonal,

so we cannot separately solve for each $H_{j\bullet}$. Nevertheless, the algorithm is quite efficient (Section 4.4).

Multiple tangent-planes with reversed signs can also be used to approximate the convex max-sparsity constraints. Then problem (4.34) reduces to a QP (Section 2.2.1). Except for solvers for linear programs, QP solvers are usually among the most efficient mathematical programming codes available. Thus, for a given large-scale problem some additional speed might be gained by using QP instead of SOCP solvers. In particular, this holds for the important special case when no non-trivial max-sparsity constraints are specified at all (i.e., $s_h^{\max} = s_w^{\max} = 1$).

A final remark concerns the termination criterion (Step 10 in Alg. 4.3.2). While in principle it can be chosen almost arbitrarily rigid, an overly small ϵ might not help in the overall optimization w.r.t. W *and* H. As long as, e.g., W is known only approximately, we need not compute the corresponding H to the last digit. In our experiments we chose relatively large ϵ so that the outer loop (Steps 2 to 10 in Table 4.3.2) was executed only once or twice before the variable under optimization was switched.

4.3.7 Sparsity-Maximization Approach

In this section we present an optimization scheme for sparsity-controlled NMF for which global convergence can be proven, even when W and H are optimized alternately. Here, global convergence means that the algorithm *always converges* to a *local* optimum. As in the previous sections, we assume our standard scenario (Section 4.3.1) and independently optimize for W and for H. Thus, it suffices to focus on the H-step.

Our algorithm is inspired by the reverse-convex optimization scheme suggested by Tuy [Tuy87]. This scheme is a *global* optimization algorithm in the sense that it finds a true global optimum. However, as already pointed out in [Tuy87], it does so at a considerable computational cost. Furthermore, it does not straightforwardly generalize to *multiple* reverse-convex constraints that are essential for sparsity-controlled NMF. We avoid these difficulties by confining ourselves to a *locally* optimal solution.

The general idea of our algorithm is as follows: After an initialization step, it alternates between two convex optimization problems. One maximizes sparsity subject to the constraint that the objective value must not increase. Dually, the other optimizes the objective function under the condition that the min-sparsity constraint may not be violated.

The Sparsity-Maximization Algorithm (SMA)

The sparsity-maximization algorithm is described below. A summary in pseudocode is outlined in Alg. 4.3.3

Initialization. For initialization we start with any point $H^0 \in \partial\mathcal{C}(s_h^{\min})$ on the boundary of the min-sparsity cone. It may be obtained by solving (4.31) without the min-sparsity constraints and projecting the solution onto $\partial\mathcal{C}(s_h^{\min})$. We set $k \leftarrow 0$.

First step. Given the current iterate H^k, we consider the program

$$\begin{aligned} \max_{H} \quad & g(H) = \min_j \{\mathrm{sp}(H_{j\bullet})\} \\ \text{s.t.} \quad & H \in \mathbb{R}_+^{r\times n} \cap \mathcal{C}_h(s_h^{\max}) \\ & f(H) \le f(H^k)\,, \end{aligned} \tag{4.39}$$

that maximizes sparsity of the least sparse $H_{j\bullet}$ subject to the constraint that the solution may not measure worse than H^k in terms of the target function f. This is a convex maximization problem on a bounded domain. As such, it can in principle be solved to global optimality [Tuy87]. However, practical algorithms exist for small-scale problems only.

Thus, we will content ourselves with a local improvement that is obtained by replacing $\mathrm{sp}(x)$ by its first order Taylor expansion at H^k, resulting in the SOCP

$$\begin{aligned} \max_{H,t} \quad & t \\ \text{s.t.} \quad & H \in \mathbb{R}_+^{r\times n} \cap \mathcal{C}_h(s_h^{\max}) && \text{(4.40a)} \\ & f(H) \le f(H^k) && \text{(4.40b)} \\ & t \le \mathrm{sp}(H_{j\bullet}^k) + \langle \nabla_{H_{j\bullet}} \mathrm{sp}(H_{j\bullet}^k), H_{j\bullet} - H_{j\bullet}^k \rangle\,, \quad j = 1,\dots,r\,, && \text{(4.40c)} \end{aligned}$$

where constraint (4.40b) ensures that the objective value will not deteriorate. In standard form this constraint translates to

$$\begin{pmatrix} \mathrm{vec}(V^\top) - (W \otimes I)\mathrm{vec}(H^\top) \\ f(H^k) \end{pmatrix} \in \mathcal{L}^{rn+1}. \tag{4.41}$$

We denote the result by H^{sp}. Note that this step maximizes sparsity in the sense that $\mathrm{sp}(H^k) \le \mathrm{sp}(H^{\mathrm{sp}})$, due to (4.40c) and the convexity of $\mathrm{sp}(\cdot)$.

Second step. While the intermediate solution H^{sp} satisfies the min-sparsity constraint, it may not be an optimal local solution to the overall problem. Therefore, in a second step, we solve the SOCP

$$\begin{aligned} \min_{H} \quad & f(H) \\ \text{s.t.} \quad & H \in \mathbb{R}_+^{r\times n} \cap \mathcal{C}_h(s_h^{\max}) && \text{(4.42a)} \\ & \left\| H_{j\bullet} - H_{j\bullet}^{\mathrm{sp}} \right\|_2 \le \min_{q \in \mathcal{C}(s_h^{\min})} \left\| q - H_{j\bullet}^{\mathrm{sp}} \right\|_2\,, \quad j = 1,\dots,r && \text{(4.42b)} \end{aligned}$$

which in standard form reads

$$\begin{aligned} \min_{H,t} \quad & t \\ \text{s.t.} \quad & \begin{pmatrix} \mathrm{vec}(V^\top) - (W \otimes I)\mathrm{vec}(H^\top) \\ t \end{pmatrix} \in \mathcal{L}^{rn+1} \\ & \begin{pmatrix} H_{j\bullet} - H^{\mathrm{sp}}_{j\bullet} \\ \min_{q \in \mathcal{C}(s_h^{\min})} \left\| q - H^{\mathrm{sp}}_{j\bullet} \right\|_2 \end{pmatrix} \in \mathcal{L}^{n+1} \quad \forall j \\ & H \in \mathbb{R}_+^{r \times n} \cap \mathcal{C}_h(s_h^{\max}). \end{aligned} \tag{4.43}$$

Here, the objective function f is minimized subject to the constraint that the solution must not be too distant from H^{sp}. To this end, the non-convex min-sparsity constraint is replaced by a convex max-distance constraint (4.42b), in effect defining a spherical *trust region*.

Termination. As long as the termination criterion $\left|f(H^k) - f(H^{k-1})\right| \leq \epsilon$ is not met we continue with the first step.

When the algorithm terminates a locally optimal H for the current configuration of W is found. In subsequent runs, we will not initialize the algorithm with an arbitrary H^0, but simply continue alternating between step one and step two using the current best estimate for H as a starting point[1]. This way, we can be sure that the sequence of H^k is monotonous, even when W is occasionally changed in between.

Remarks

The requirement that the feasible set has an non-empty interior is important. If $s_h^{\max} = s_h^{\min}$, the approximate approach in (4.40) breaks down, and each iteration just yields $H^k = H^{\mathrm{sp}} = H^{k+1}$. In this situation, it is necessary to temporarily weaken the max-sparsity constraint. Fortunately, max-sparsity constraints seem to be less important in many applications.

While convergence is guaranteed (see Prop. 7 below) and high-quality results are obtained (see Table 4.1 in Section 4.4.1), SMA can be slower than the tangent-plane method presented in the previous section. This is especially the case when $s_h^{\max} \approx s_h^{\min}$. Then, in order to solve a problem most efficiently, one will start with the tangent-plane method and only if it starts oscillating switch to sparsity-maximization mode.

Convergence Properties

We check the convergence properties of the SMA.

[1] Note that while such a scheme could be implemented with TPC as well, it would perform poorly in practice: Without proper initialization step TPC locks too early onto bad local optima.

Algorithm 4.3.3 Sparsity-maximization algorithm in pseudocode.

1: $H^0 \leftarrow$ solution of (4.33) projected on $\partial\mathcal{C}_h(s_h^{\min})$, $k \leftarrow 0$
2: **repeat**
3: $\quad H^{\text{sp}} \leftarrow$ solution of (4.40)
4: $\quad H^{k+1} \leftarrow$ solution of (4.42)
5: $\quad k \leftarrow k+1$
6: **until** $\left|f(H^k) - f(H^{k-1})\right| \leq \epsilon$

Proposition 7. *Under the assumptions stated in Section 4.3.1 and 4.3.3, the sparsity-maximization algorithm (Alg. 4.3.3) converges to a point* (W^*, H^*) *satisfying the first-order necessary optimality conditions of problem* (4.22).

Proof. Under the assumptions stated in Section 4.3.1, the feasible set is bounded. Furthermore, Alg. 4.3.3, alternately applied to the optimization of W and H, respectively, computes a sequence of feasible points $\{W^k, H^k\}$ that steadily decreases the objective function value. Thus, by taking a convergent subsequence, we obtain a cluster point (W^*, H^*) whose components separately optimize (4.40) when the other component is held fixed. It remains to check that conditions (4.26) are satisfied after convergence.

We focus on H without loss of generality. Taking into account the additional non-negativity condition, condition (4.40c) is equivalent to $t \leq \text{sp}(H_{j\bullet})$, because $\text{sp}(\cdot)$ is convex. Moreover, $t = s_h^{\min}$ because after convergence of iterating (4.40) and (4.42), the min-sparsity constraint will be active for some of the indices $j \in \{1, \ldots, r\}$. Therefore, using the notation (4.23), the solution to problem (4.40) satisfies

$$\max_{t, H \in Q_h} t^* = s_h^{\min}, \quad f(H^k) - f(H^*) \geq 0, \quad G_h(H^*) \in \mathbb{R}^r_+ \tag{4.44}$$

Using multipliers $\lambda_f^*, \tilde{\lambda}_h^*$, the relevant first-order condition with respect to H is

$$-\frac{\partial}{\partial H}\Big(\lambda_f^* f(H^*) + \big\langle \tilde{\lambda}_h^*, G_h(H^*)\big\rangle\Big) \;\in\; N_{Q_h}(H^*). \tag{4.45}$$

This corresponds to the condition on H in (4.26a). The W-part can be shown in the same way. □

4.3.8 Solving NMF Extensions

In this section we address the NMF extensions described above (Section 4.2.4 – 4.2.7). Any of the solvers presented so far can be adapted for any of the following extensions and variants. Consequently, we present only the *new* ideas for each problem and refer the reader to the previous descriptions of the basic algorithms. As before, we will describe how to optimize for H, assuming W constant.

Exploiting Information from Class Labels

The supervised variant (4.10) of the NMF problem is readily solved by above algorithms since (4.10b) translates, for each class i and for each coefficient vector $H_{j\bullet}$ belonging to class i, into a second order constraint

$$\begin{pmatrix} 1/n_i H_{(i)} e - H_{j\bullet} \\ \lambda/n_i e^\top H_{(i)} e \end{pmatrix} \in \mathcal{L}^{n+1}, \quad \forall i, \forall j \in \text{class}(i). \tag{4.46}$$

Here, the $r \times n_i$-matrix of coefficients belonging to class i is abbreviated $H_{(i)}$ and we recognize $\mu_i = 1/n_i H_{(i)} e$. Adding these constraints to, e.g., (4.40) and (4.42) yields an algorithm for solving supervised NMF.

Soft Sparsity Formulation

The relaxed form of sparsity-controlled NMF described in (4.9) is optimized by linearizing $\text{sp}(x)$ around H^k, yielding the SOCP

$$\begin{aligned} \min_{H,t,s} \quad & t - \lambda_h s \\ \text{s.t.} \quad & \begin{pmatrix} \text{vec}(V^\top) - (W \otimes I)\text{vec}(H^\top) \\ t \end{pmatrix} \in \mathcal{L}^{rn+1} \\ & s \leq \text{sp}(H^k_{j\bullet}) + \langle \nabla_{H_{j\bullet}} \text{sp}(H^k_{j\bullet})^\top, H_{j\bullet} - H^k_{j\bullet} \rangle \quad \forall j \\ & H \in \mathbb{R}^{r \times n}_+. \end{aligned} \tag{4.47}$$

Thus, in order to solve (4.9) for H we iteratively solve instances of (4.47) until convergence.

Sparse PCA

Next, we show how to optimize for H when both, W and H may contain negative entries. The idea is that we factorize any non-zero matrix $M \in \mathbb{R}^{m \times n}$ into $M_\pm = \text{sign}(M) \in \mathbb{R}^{m \times n}$ and $M_+ \in \mathbb{R}^{m \times n}_+$ s.t. $M = M_\pm \odot M_+$. Since sparsity is not affected by sign changes or multiplicative constants we observe

$$\text{sp}(M) = \text{sp}(M_+), \tag{4.48}$$

i.e., it is *sufficient to exercise sparsity control on the non-negative part* of x. Thus, the sparsity-controlled NMF algorithms presented above can be used on W_+ and H_+. Finally, for those entries in W and H that are close to 0 we subsequently optimize signs using convex programming.

First step. Considering H, we first optimize for H_+, by solving

$$\begin{aligned} \min_{H_+,t} \quad & t \\ \text{s.t.} \quad & \begin{pmatrix} \text{vec}(V) - ((I \otimes W) \odot H_S)\text{vec}(H_+) \\ t \end{pmatrix} \in \mathcal{L}^{rn+1} \\ & H_+ \in (\mathbb{R}_+^{r \times n} \cap \mathcal{C}_h(s_h^{\max})) \setminus \mathcal{C}_h(s_h^{\min}) \end{aligned} \tag{4.49}$$

using any of the techniques presented above. Note that (4.49) is identical to the NMF case, i.e., it minimizes the original problem for $H = H_\pm \odot H_+$ but the signs of H are not allowed to change. The matrix H_S is given by

$$H_S = (I_{r \times r} \otimes E_{n \times n}) \odot (\text{vec}(H_\pm) e^\top)^\top. \tag{4.50}$$

Second step. We then solve for $H_\pm$ using the convex program

$$\begin{aligned} \min_{t, H_\pm \in H_\epsilon} \quad & t \\ \text{s.t.} \quad & \begin{pmatrix} \text{vec}(V) - ((I \otimes W) \odot H_A)\text{vec}(H_\pm) \\ t \end{pmatrix} \in \mathcal{L}^{rn+1} \\ & -1 \leq H_\pm \leq 1, \end{aligned} \tag{4.51}$$

where H_A is constructed from H_+ analogously to (4.50):

$$H_A = (I_{r \times r} \otimes E_{n \times n}) \odot (\text{vec}(H_+) e^\top)^\top. \tag{4.52}$$

H_ϵ denotes those entries in H_+ that are within ϵ from 0. Entries in $H_\pm$ corresponding to larger entries in H_+ are *not* optimized in order to prevent an entry in $H_\pm$ with small norm cancel out an entry with large norm in H_+, thus possibly modifying sparseness of the product $H = H_\pm \odot H_+$.

Transformation Invariance

To approach problem (4.12) we assume a finite set of linear transformations mapping the input data $V \in \mathbb{R}^{m \times n}$ into $T_\theta(V) \in \mathbb{R}^{m \times n}$. θ_i specifies the transformations active for image $i \in \{1, \ldots, m\}$.

After each iteration we greedily replace the image data V by its most probable transformation, i.e., setting $V \leftarrow T_{\theta^*}(V)$ with

$$\theta^* = \arg\min_\theta \left\| T_\theta(V) - W^k H^k \right\|_F^2. \tag{4.53}$$

As long as the identity is part of the possible transformations this operation never increases the objective value to be minimized. However, for large images and many possible transformations it can be a very slow operation to compute. In this case, variational techniques and FFT offer greatly improved performance [FJ01].

Missing Values

All algorithms described above (Section 4.3.4, 4.3.6, 4.3.7) are directly applicable to the missing values case (4.13): Simply remove the terms with $E_{ij} = 0$ from the objective function. In case a complete row (or column) of E equals zero this row (or column) is completely removed from the problem. This is consistent with the intuition that an entirely unobserved image (or pixel location) should not influence the resulting factorization.

4.4 Evaluation

In this section we validate our algorithms on very well understood, simple data sets. The main objective is to draw precise conclusions about correctness and performance of the algorithms and our particular implementations. Realistic applications on real-world data sets are treated in Chapter 6.

4.4.1 Comparison with Established Algorithms

To see how our algorithms compare against an established method we computed sparsity-controlled decompositions into $r = 4$ basis functions for a subset of the USPS handwritten digits data set using our methods and projected gradient descent (pgd) as proposed in [Hoy04]. For different choices of sparseness we report mean and standard deviation of the runtime and mean residual error[2] averaged over 10 runs in Table 4.1. Note that the stopping criterion used was different for our algorithms and for pgd: We stopped when after a full iteration the objective value did not improve at least by a constant, the pgd implementation used[3] stopped as soon as the norm of the gradient was smaller than some ϵ. As the error measurements shown in Table 4.1 demonstrate, both stopping criteria yield comparable results. Regarding running time we see that the tangent-plane approach was usually fastest, followed by sparse-maximization. Also, our algorithms usually showed relatively small variation between individual runs while the runtime of pgd varied strongly, dependent on the randomly chosen starting points.

4.4.2 Large-Scale Factorization of Image Data

To examine performance on a larger data set we sampled 10 000 image patches of size 11×11 from the Caltech-101 image database [FFFP04]. Using a QP solver and the TPC algorithm we computed image bases with $r = 2, 4, \ldots, 10$ and $r = 50$ basis functions using $s_w^{\min} = 0.5$. In addition, we varied the stopping criterion from $\epsilon \in \{1, 0.5, 0.25\}$. Note

[2] Standard deviation of the residual error was equally negligible for all algorithms.

[3] We used the pgd code kindly provided by the author of [Hoy04], and removed all logging and monitoring parts to speed up calculation. Our SOCP solver was Mosek 3.2 from MOSEK ApS, Denmark, running under Linux.

sparsity	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
mean time tgp	34.35	35.46	41.30	64.97	66.80	60.86	56.33	51.82	42.50
mean time spm	94.81	106.20	133.52	159.30	173.14	167.06	114.36	78.42	74.96
mean time pgd	517.02	1038.99	218.17	70.24	177.35	189.48	167.94	430.36	322.88
stdv time tgp	3.17	2.64	3.84	5.50	8.51	7.05	4.25	0.76	0.38
stdv time spm	3.48	12.67	23.67	16.45	11.51	11.64	7.69	1.22	1.29
stdv time pgd	278.24	21.21	128.00	8.90	78.12	52.54	95.53	439.34	174.81
mean error tgp	0.82	0.76	0.73	0.72	0.78	0.89	0.99	1.08	1.12
mean error spm	0.81	0.77	0.74	0.73	0.78	0.89	1.00	1.08	1.13
mean error pgd	0.85	0.79	0.74	0.72	0.77	0.88	0.99	1.07	1.12

Table 4.1: **Performance Comparison.** Comparison of the tangent-plane (tgp) approach and the sparsity-maxmization algorithm (spm) with projected gradient descent (pdg). Sparse decompositions of the digit data set were computed. Statistics collected over 10 repeated runs are reported for runtime (sec.) and residual error $\|V - WH\|_F^2$.

ϵ	$r = 2$	$r = 4$	$r = 6$	$r = 8$	$r = 10$
1.00	5.99	49.32	98.91	222.01	278.76
0.50	5.97	54.67	103.93	230.45	256.98
0.25	10.22	72.75	133.23	224.62	363.62

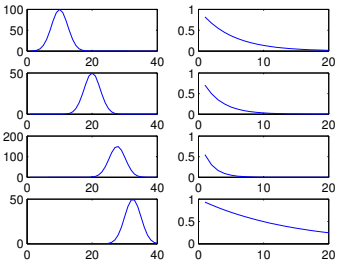
Table 4.2: **Large-scale performance.** A matrix containing $n = 10\,000$ image patches with $m = 121$ pixels was factorized using r basis functions and different stopping criteria for the TPC/QP algorithm (see text). The median CPU time (sec.) for three repeated runs is shown. Even the largest experiment with over 100 000 unknown variables is solved within 6 min.

that the corresponding QP instances contained roughly 100 000 to over half a million unknowns, so a stopping criterion of $\epsilon = 1$ translates to very small changes in the entries of W and H. We did *not* use any batch processing scheme but solved the QP instances directly, requiring between 100 MB and 2 GB of memory.

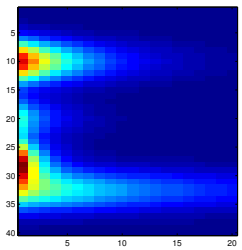
We show the median CPU time over three repeated runs for this experiment in Table 4.2: While the stopping criterion has only minor influence on the run time the number of basis functions is critical. All problems with up to 10 basis functions are solved within 6 min. For the large problem with 50 basis functions we measured a CPU time of 3, 5, and 7 hours for $\epsilon \in \{1, 0.5, 0.25\}$. Memory consumption was roughly 2 GB. We conclude that factorization problems with half a million unknowns can be comfortably solved on current office equipment.

4.4.3 Global Optimization

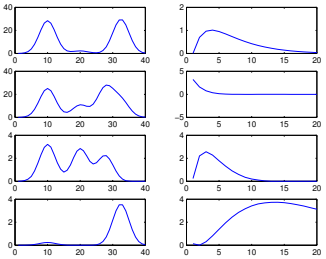
A potential source of difficulties with the sparsity-maximization algorithm is that the lower bound on sparsity is optimized only locally in (4.40). Through the proximity constraint in (4.42) the amount of sparsity obtained in effect limits the step size of the



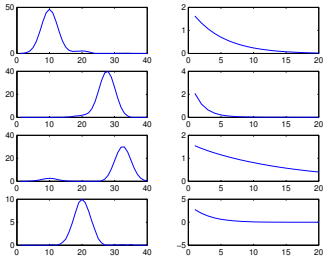
(a) True W and H.



(b) $V = WH + \eta$.



(c) NMF, $s_w^{\min} = 0.0$.



(d) NMF, $s_w^{\min} = 0.6$.

Figure 4.4: **Paatero experiments.** The data set is displayed in Figure 4.4(a): Gaussian and exponential distributions are multiplied to yield matrix V. In the experiments, a small amount of Gaussian noise $\eta \sim \mathcal{N}(0, 0.1)$ is added to the product. The results for different values of the min-sparsity constraint are shown in Figure 4.4(c) and 4.4(d): Only a non-trivial sparsity constraint makes recovery of W and H successful.

algorithm. Insufficient sparsity optimization may, in the worst case, lead to convergence to a bad local optimum.

To see if this worst-case scenario is relevant in practice, we discretized the problem by sampling the sparsity cones using rotated and scaled version of the current estimate H^k and then evaluated g in (4.39) using samples from each individual sparsity cone. Then we picked one sample from each cone and computed (4.40) replacing the starting point H^k by the sampled coordinates. For an exhaustive search on r cones, each sampled with s points, we have s^r starting points to consider.

For demonstration we used the artificial Paatero data set [Paa97] consisting of products of Gaussian and exponential functions (Figure 4.4). This data set is suitable since it is not overly large and sparsity control is crucial for its successful factorization (cf. [Paa97] and Figure 4.4).

In the sparsity-maximization algorithm we first sampled the four sparsity cones corresponding to each basis function of the data for $s_w \geq 0.6$ sparsely, using only 10 rotations on each cone. We then combined the samples on each cone in each possible way and evaluated g for all corresponding starting points. In a second experiment we placed

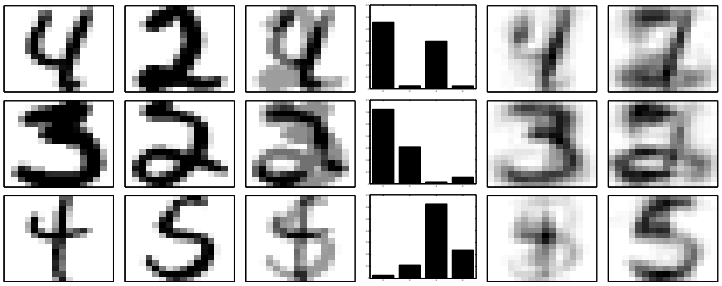


Figure 4.5: **Recognition and segmentation.** Based on sparse NMF, a model for the digits 2 – 5 (first columns) is built and presented with superimposed digits (third column) *after* training. The model consistently assigns the highest probabilities (fourth column) to the digits forming the image. This shows that NMF models can be relatively *stable against disturbances*. A subsequent local optimization retrieves the original digits (last columns).

1000 points on each sparsity cone, and randomly selected 10^4 combinations as starting points. The best results obtained over four runs and 80 iterations with our local linearization method and the sparse enumeration (first) and the sampling (second) strategy, are reported below:

Algorithm	min-sparsity	objective value
local linearization	0.60	0.24
sparse enumeration	0.60	0.26
sampling	0.60	0.26

We see that the local sparsity maximization yields results comparable to the sampling strategies. In fact, it is better: Over four repeated runs the sampling strategies each produced outliers with very bad objective values (not shown). This is most likely caused by severe under-sampling of the sparsity cones. This problem is not straightforward to circumvent: With above sampling schemes a run over 80 iterations takes about 24h of computing[4], so more sampling is not an option. In comparison, the proposed algorithm finishes in few seconds.

4.4.4 Recognition

Here we examine how NMF codes can be used for image recognition experiments and how supervised training can improve results.

[4]On machines with 3GHz P4, 2GB RAM, running Matlab under Linux.

Superimposed Digits

It has previously been reported that localized NMF is relatively robust against occlusions and disturbances [LHZC01]. To verify this claim for sparse NMF we repeated an image recognition experiment, previously approached with *credibility networks* [HGT00]: A model for the individual digits 2,3,4,5 from the USPS digit database was built, then new, more complicated, images were constructed by superimposing images from two different digits (Figure 4.5). Our model consisted of sparse NMF codes ($r = 20$, $s_w^{\min} = 0.6$) for the *single* digits and a *conditional maximum entropy* (cMaxEnt) classifier[5] $p(y|h)$ for class labels y and given coefficients h, using mean coefficient values and distance to randomly chosen reference coefficients as features.

For each combined image we computed h and evaluated $p(y|h)$ for $y \in \{2, 3, 4, 5\}$. On a test data set we counted how often the two top-ranking digits, w.r.t. $p(y|h)$, were the correct digits composing the image.

Five self-selected human subjects (students) achieved classification rates between 65% and 80% (mean 75%) on our data. The NMF-cMaxEnt classifier described above scored 77% correct on 500 test samples. For the more complex credibility networks a recognition rate of 78.3% is reported [HGT00] on a test set of 120 binarized images.

Once a decision is made for two digits y_1, y_2, we can visualize the corresponding segmentation by solving the non-convex program

$$\begin{aligned} \max_{h_1,h_2} \quad & p(y_1|h_1)p(h_1) \cdot p(y_2|h_2)p(h_2) \\ \text{s.t.} \quad & h = h_1 + h_2 \\ & 0 \leq h_1, h_2 \end{aligned} \tag{4.55}$$

i.e., we factor the reconstruction coefficient h into h_1 and h_2 such that the probability of the detected digits is maximized. Depending on the features used for training cMaxEnt and the form of the prior densities $p(h)$ this can be a very difficult problem. It turns out that for our choice of features and a Parzen estimator for $p(h)$ a conjugate gradient search already yields meaningful reconstructions (Figure 4.5).

[5] A cMaxEnt model is the solution to the convex optimization problem [NLM99]

$$\begin{aligned} \max_{p} \quad & H(p(y|x)) \\ \text{s.t.} \quad & \langle f \rangle = \mathbb{E}_p(f(x)) \\ & \int dp = 1 \end{aligned} \tag{4.54}$$

which defines the most general probability distribution, in terms on Shannon entropy H, on the class labels subject to the constraint that the feature statistics $\langle f \rangle \propto \sum_i f(x_i)$ measured on the training data are met.

Note, that (4.54) models the conditional distribution $p(y|x)$ as a function of y only. x is a known parameter as it can be measured from the given sample to classify. This is an important simplification over ordinary MaxEnt [Jay57]: By measuring x one needs to integrate over the label space $\mathcal{Y}$ only. It is thus not important how large the feature space $\mathcal{X}$ is. Often, $|\mathcal{Y}|$ is finite and small, so cMaxEnt obtains huge computational savings over traditional MaxEnt models.

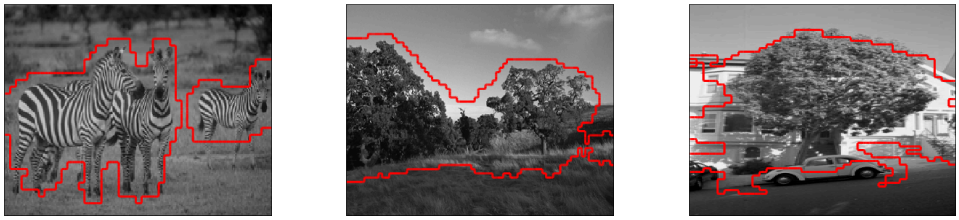


Figure 4.6: **KL-Segmentation with NMF.** A similarity matrix, derived from KL-distances between locally fitted generalized Laplacians (Section 3.2), was clustered using NMF. No effort was undertaken to enforce particularly smooth partitions. The blocky segmentation boundaries are an artifact of our particular implementation. The results are competitive with Normalized Cut (Figure 3.6).

Supervised Training

To show that the supervised label constraints (4.10b) can be useful we trained NMF codes ($r = 4$) on only 100 samples from the USPS handwritten digit data set. We used different values for the parameter λ and a very simple conditional maximum entropy model $p(y|h)$ with mean coefficient values $\mathbb{E}[h_i]$ as only features for classification. The number of errors on a 300 sample test dataset is given below:

λ	1e4	1e2	1	1e-2	1e-4	1e-6
#errors	108	82	75	60	58	56

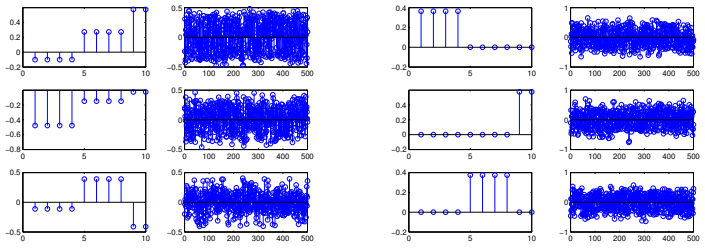
When λ is large, i.e., the supervised label constraint is inactive, the error is about 36% (108 out of 300 samples). This is slightly worse than a corresponding PCA basis (95 errors) would achieve. As the label constraint is strengthened the classification performance improves and finally is almost twice as good as in the unsupervised case.

4.4.5 Clustering and Segmentation

Since distance matrices are non-negative by nature, NMF can also beneficial for clustering: Let $D \in \mathbb{R}_+^{n \times n}$ be a distance matrix between n objects and let $D \approx WH$ be its non-negative matrix approximation. Then we cluster H using any standard algorithm like k-means or hierarchical clustering.

In Figure 4.6 we show segmentation results using NMF on the distance matrix employed for the NCut segmentation experiment (Section 3.2.7). The results are similar in segmentation quality. Note that since NMF works in a batch-processing scheme (Section 4.3.4) we would expect it to scale to larger problems easily.

Note that the k-means clustering step might not be strictly necessary: Recently, a symmetric variant of NMF has gained interest as a clustering algorithm on its own. In this context we refer the reader to [ZS05, DHS05].



(a) PCA (b) Sparse PCA

Figure 4.7: **Sparse PCA experiment.** Basis and coefficients for an artificial data set are shown. Only sparsity-controlled PCA successfully recovers the structure of the data.

4.4.6 Sparse PCA

As proof-of-concept we factorized the artificial data set examined in [dGJL04] using PCA and sparsity-controlled PCA. The data set consists of three factors sampled from $V_1 \sim \mathcal{N}(0, 290), V_2 \sim \mathcal{N}(0, 300), V_3 \sim -0.3V_1 + 0.925V_2 + \eta$ and additional Gaussian noise. The sparse PCA algorithm iteratively solved (4.49) and (4.51) using the relaxed optimization framework (4.47) with $\lambda_w = 0.6$ and a constraint limiting the admissible reconstruction error.

In Figure 4.7 we depict the factors and factor-loadings for PCA and sparsity-controlled PCA (best result out of three repeated runs). It is apparent that sparsity-controlled PCA correctly factorizes the data, while classical PCA fails.

4.4.7 Image Modeling

Now we approach image modeling tasks using the tools developed in this chapter.

Translation-Invariant Image Coding

In Figure 4.8 we show the results of transformation-invariant NMF (TNMF) applied to the artificial data described in [FJ03]: Four image primitives are translated using circular shifts in both image dimensions and Gaussian noise is added. The resulting training data set contains 1000 randomly translated images. For these, we learned image based using a feathering mask to encourage centered bases[6]. The resulting image basis not only models the data well, it also has a nice complementary structure: Even without additional sparseness constraints it tends to avoid modeling the same image location multiple times [LS99], leading to a true parts-based representation. A possible

[6] I.e., each basis function was weighted with a Gaussian s.t. pixels near the boundary had slightly less influence than pixels near the center.

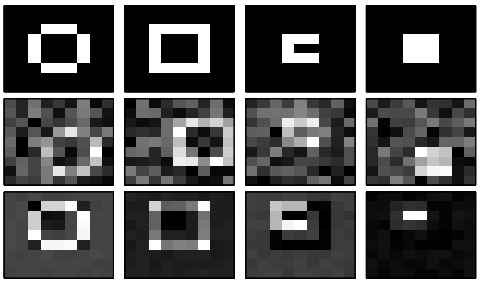


Figure 4.8: **Translation-invariant NMF.** Results of translation-invariant NMF on the artificial data set from [FJ03]. The first row shows the image primitives used. Shifted and noise corrupted versions of these were used for training (second row). NMF successfully constructs a suitable image base (third row). Note that the basis functions are *not* simply centroids in image space: they represent *parts* which must be *composed* to form the images. Thus, they are potentially more general without sacrificing image quality/sharpness.

explanation for this behavior is that the parts-based representation offers more degrees of freedom, making a better fit to the noisy data.

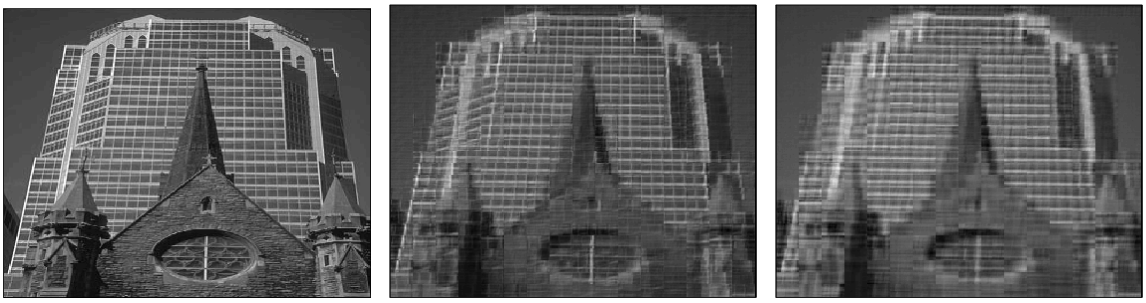
In a more realistic scenario we learned image bases for a skyscraper image (Figure 4.9): While NMF correctly captures the dominant horizontal and vertical lines, it is forced to model the same structure multiple times to fit the data well. TNMF removes this burden, allowing for much finer detail to appear in the basis functions. In fact, looking closely one can recognize parts of the building and key architectural structures being modeled.

We also used the PCA and the TNMF image bases for reconstruction: the original image was divided into 20×20 patches and for each patch we determined the optimal translation w.r.t. the given image bases. The patches were then reconstructed and assembled to form images 4.9(b) and 4.9(c). As expected, translation-invariance ensures that the TNMF reconstruction looks notably sharper. It is also closer to the original image: The Frobenius norm of the residual image was about 7% larger for the PCA reconstruction than for the TNMF reconstruction.

Modeling Images with Occlusions

As a test case for NMF with missing values, eqn. (4.13), we used a subset of the MIT/CBCL face data set [CBC00] and partially obstructed some of the images (Figure 4.10). Then we computed a NMF factorization ($r = 10$) for the image database. In one experiment we used standard NMF and replaced the obstructed image content by zeros. In a second experiment we used model (4.13), flagging the obstructed parts as missing data.

The results convincingly indicate that special treatment of missing data is beneficial:



(a) Original (b) TNMF reconstruction (c) PCA reconstruction

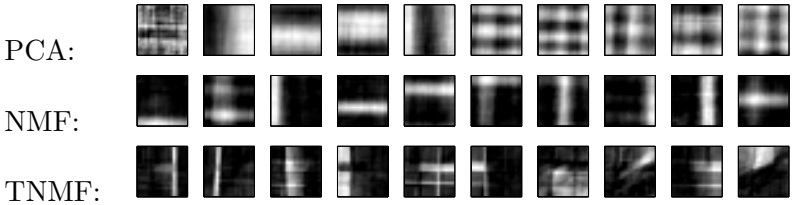


Figure 4.9: **Image modeling.** Different image bases learned for image 4.9(a) are shown. PCA learns global properties of image variation. The individual base images carry no apparent semantic meaning. NMF learns sparse, localized image features, but represents the dominant image elements (horizontal and vertical bars) multiple times. Transformation-invariant NMF (TNMF) is less redundant and captures very detailed image structure which can sometime be recognized as parts from the building. In Figure 4.9(b) and 4.9(c) reconstructions for TNMF and PCA are displayed. The TNMF reconstruction appears sharper and is slightly more accurate (see text).

Although we used a relatively small number of basis functions only, standard NMF actually models the obstructions as part of the dataset. It basically does not reconstruct the missing image information. We could force stronger regularization by using an even smaller image base, but then the results would look very blurry.

On the other hand, NMF with missing values correctly uses the information from the unobstructed images to inpaint the obstructed areas. Since the image database contains different views from each face, this task is quite feasible and the reconstructions look almost perfect.

Modeling a Low-Entropy Image Class

A sample application using real-world data is face modeling: Human faces, aligned, cropped and evenly lit, lead to highly structured images with relatively low entropy. With such images, sparse NMF appears robust against quantization: We trained a sparse image code ($r = 8$, $s_h^{\min} = 0.3$) for face images [CBC00] and a PCA code for comparison. Then we enumerated possible reconstructions by setting each entry of the coefficient

Figure 4.10: **Image reconstruction with occlusions.** Corrupted input data (first column) is reconstructed using standard NMF (second column) and NMF with missing variables (third column, NMF/MV). While standard NMF fails to recover the image, the missing variables approach delivers almost perfect results that are very similar to the ground truth (last column).

vector to 0 or to 1. The resulting $2^8 = 256$ images are shown in Figure 4.11 and 4.12: While most NMF "reconstructions" look remarkably natural the corresponding PCA images mostly suffer severe degradation.

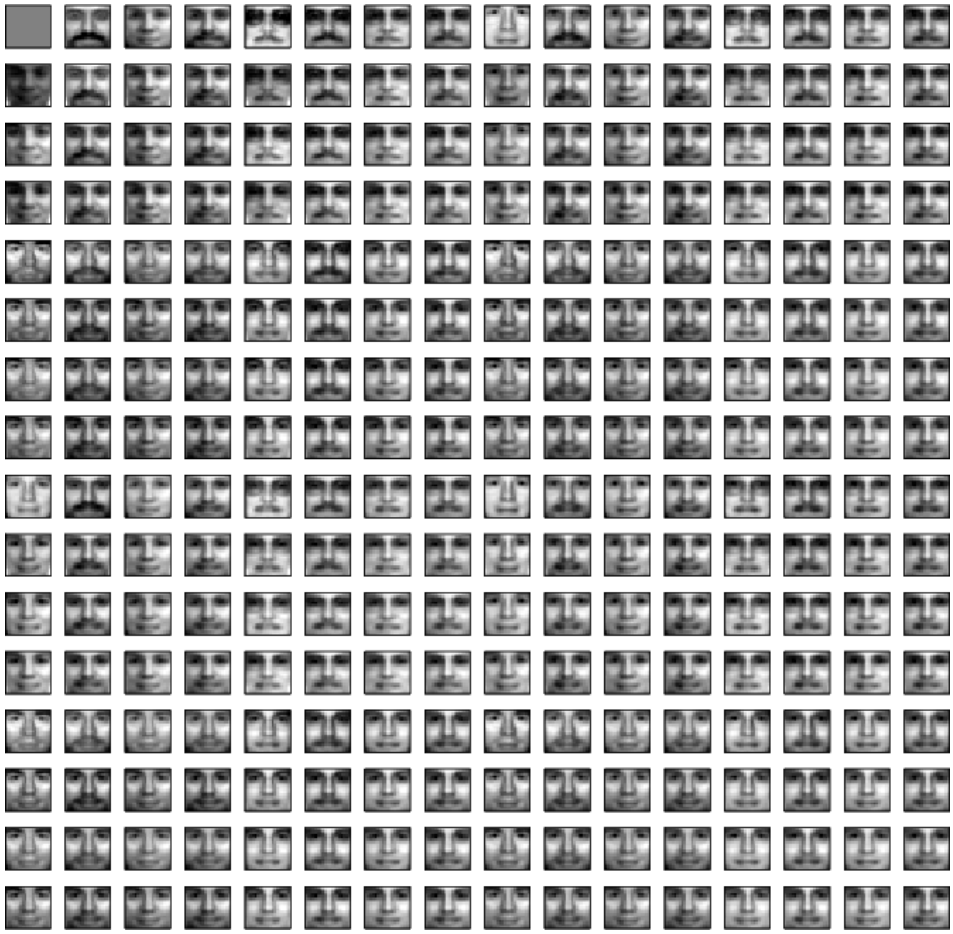
To measure how quantization affects reconstruction performance we used SVD and NMF to find a large image base ($r = 100$) on a subset of the face data. Then, we quantized the reconstruction coefficients H using k-means on each individual row of coefficients $H_{j\bullet}$. The results are shown in Figure 4.13: As expected, SVD offers slightly better reconstruction performance for large values of k. With stronger quantization, however, it looses its advantage and NMF performs better.

This is surprising as quantization robustness was not an original design goal of NMF codes. Adopting a Bayesian perspective we can explain this result by the fact that as quantization (or noise) increases the influence of prior information becomes more important. NMF models the prior information that images are non-negative. SVD has no such constraint and thus suffers more as quantization increases.

4.5 Summary

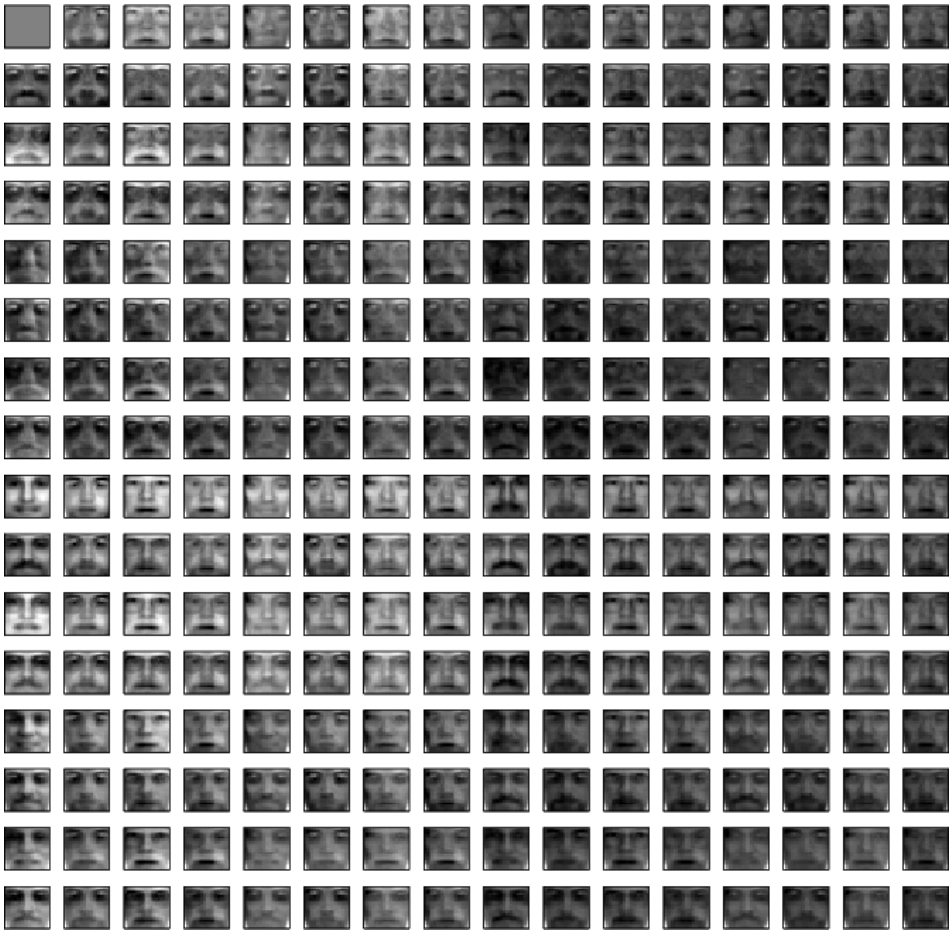
In this chapter we examined non-negative matrix factorization as a versatile tool for various image processing and machine learning tasks. We saw that precise sparsity control over NMF factors and coefficients can be exercised, leading to efficient and elegant sequential convex optimization algorithms. We extended the sparsity-constrained NMF

model in various dimensions, accounting for transformation invariance, prior knowledge about class labels, using hidden variables, or factorizations with negative entries. In effect, we built a flexible toolbox of useful models, all solved within the same optimization framework.



NMF: $W * \delta(h > \epsilon)$

Figure 4.11: **Robustness against quantization.** The 256 faces corresponding to previously learned 8bit NMF image code after quantization of the coefficients. The top left image corresponds to the binary coefficient vector 00000000_2, the bottom right image to 11111111_2. The NMF "reconstructions" remain very face-like.



PCA: $W * \delta(|h| > \epsilon)$

Figure 4.12: **Robustness against quantization.** The 256 faces corresponding to previously learned 8bit PCA image codes after quantization of the coefficients. The top left image corresponds to the binary coefficient vector 00000000_2, the bottom right image to 11111111_2. The PCA reconstruction produces many unnatural looking images.

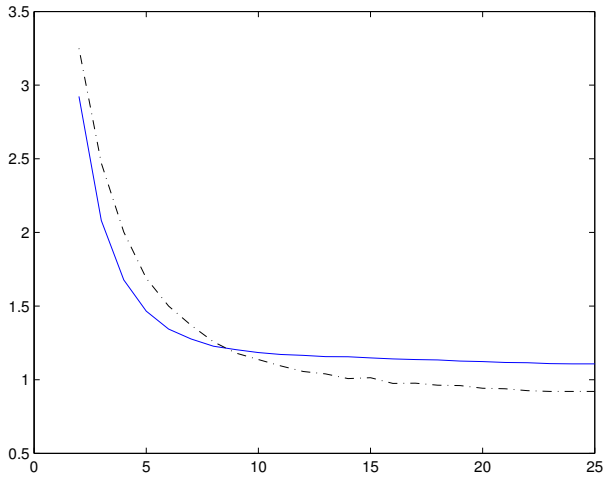


Figure 4.13: **Robustness against quantization.** An basis for face images was trained using NMF (solid blue line) and SVD (dashed black line). The reconstruction coefficients H where quantized using k-means for $k = 2, \ldots, 25$ (x-axis) and the reconstruction error $f(W, H)$ was determined (y-axis) for the training data set. SVD shows smaller reconstruction error with large values for k. NMF is more robust against strong quantization.

Chapter 5

Non-Negative Tensor Models

Non-negative tensor factorization (NTF) has recently been proposed as sparse and efficient image representation [WW01, SH05, HPS05]. Until now, sparsity of the tensor factorization has been empirically observed in many cases, but there was no systematic way to control it. In this chapter, we show that a sparsity measure introduced in Section 4.2.2 applies to NTF and allows precise control over sparseness of the resulting factorization. We devise an algorithm based on sequential conic programming and show improved performance over classical NTF codes on artificial and on real-world data sets.

5.1 Motivation

The main motivation for NTF from an computer vision perspective is twofold: First, in contrast to matrix factorization, where multiple minima are always of concern, tensor factorizations will be *unique* under few general conditions [Kru77, SB00]. Second, tensor factorization of image data can take *spacial correlations* into account: Unlike in NMF, where image data is vectorized and pixels are treated as statistically independent, NTF image factors are outer products of vectors. Thus, adjacent pixels are *not* assumed to be completely independent. This explains why it has been reported that compared to NMF tensor factorization shows a greater degree of sparsity, clearer separation of image parts, better recognition rates, and a tenfold increased compression ratio [HPS05].

However, until now it was not possible to exercise *explicit sparsity control* with NTF. In this chapter, we thus extend our algorithms for sparsity controlled NMF to allow for *fully sparsity-controlled NTF models*.

Notation. Unlike in the previous chapter we now represent image data as a *tensor* of order 3, e.g., $V \in \mathbb{R}_+^{d_1 \times d_2 \times d_3}$ denotes d_3 images of size $d_1 \times d_2$. We are not concerned about the transformation properties of V, so this simplified 3-way array notation is sufficient. The factorization is given by vectors $u_i^j \in \mathbb{R}^{d_i}$, where $j = 1, \ldots, k$ indexes k independent vectors. Where convenient, we omit indices of the factors, e.g. $u_i \in \mathbb{R}^{d_i \times k}$

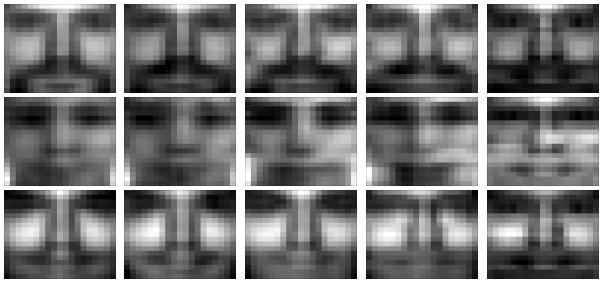


Figure 5.1: **Sparse NTF face model.** MIT CBCL faces are factorized ($k = 10$) and reconstructed using sparsity-control for horizontal factors u_1 (see text). The min-sparsity constraints were 0.0, 0.2, 0.4, 0.6, 0.8 (from left to right). Starting from $s_i^{\min} = 0.4$ reconstructions look increasingly generic and individual features disappear.

is the matrix of k factors corresponding to index i, and u alone is the ordered set of such matrices.

5.2 The NTF Optimization Problem and Sparseness

In this section we formally state the NTF optimization problem in its original form and extended by sparseness constraints.

5.2.1 Original NTF Model

The NTF optimization problem admits the general form

$$\begin{aligned} \min_{u_i^j \in \mathbb{R}^{d_i}} \quad & \left\| V - \sum_{j=1}^{k} \bigotimes_{i=1}^{3} u_i^j \right\|_F^2 \\ \text{s.t.} \quad & 0 \leq u_i^j. \end{aligned} \tag{5.1}$$

Here, image volume V is approximated by the sum of k rank-1 tensors that are outer products $u_1^j \otimes u_2^j \otimes u_3^j$. By using outer products with additional factors $u_i^j, i > 3$, this generalizes to higher-order tensors. In this work, however, we are concerned with image volumes only.

It is instructive to compare NTF with the more widespread NMF model: In NMF, image data is first vectorized, and the resulting non-negative matrix $V \in \mathbb{R}_+^{m \times d_3}$, $m = d_1 \cdot d_2$, is then factorized as the product of two non-negative matrices $W \in \mathbb{R}_+^{m \times k}$ and $H \in \mathbb{R}_+^{k \times d_3}$. In short, one optimizes (4.3)

$$\begin{aligned} \min_{W,H} \quad & \|V - WH\|_F^2 \\ \text{s.t.} \quad & 0 \leq W, H. \end{aligned}$$

It is clear that the vectorized representation does not take into account the spatio-temporal correlations of image data or video. In contrast, the NTF analogon to basis images are rank-one matrices $u_1^j \otimes u_2^j$ that nicely represent correlations along the x and y direction of the image plane. The drawback is that with NTF basis images are no longer arbitrary: The rank-one restriction rules out, e.g., basis images with diagonal structures. So images with predominantly diagonal structures are represented less efficiently.

5.2.2 Sparsity-Constrained NTF

It has early been reported that NTF codes tend do be *sparse*, i.e., many entries of the u_i^j equal zero [WW01]. Especially for pattern recognition applications, sparsity is a key property since it relates directly to learnability [LW86, HW02] and is biologically well motivated [OF97]. Sparsity also seems to act as a strong prior for *localized image representations* [Hoy04]. Such representations are desirable since they naturally focus on *parts* and thus are potentially *more robust against occlusion or noise* than are their global counterparts.

Thus, it is desirable to extend (5.1) by sparsity-controlling constraints as in the NMF case (Section 4.2.2), leading to the problem

$$\begin{aligned} \min_{u_i^j \in \mathbb{R}^{d_i}} \quad & \left\| V - \sum_{j=1}^{k} \bigotimes_{i=1}^{3} u_i^j \right\|_F^2 \\ \text{s.t.} \quad & 0 \leq u_i \\ & s_i^{\min} \leq \operatorname{sp}(u_i) \leq s_i^{\max}. \end{aligned} \tag{5.2}$$

The parameters $s_i^{\min}$ and $s_i^{\max}$ are again real numbers in $[0, 1]$ specified by the user for a given application. We propose solvers for (5.2) in Section 5.3 and validate the model on artificial and on real-world data in Section 5.4.

5.3 Solving Sparsity-Constrained NTF

In this section, we develop an algorithm for solving problem (5.2). In principle, any algorithm proposed in Section 4.3 can be adapted for the NTF case. We concentrate on the sparsity-maximization algorithm (Section 4.3.7).

First, let us rewrite (5.2) as

$$\begin{aligned} \min_{u_i^j} \quad & \left\| V - \sum_{j=1}^{k} \bigotimes_{i=1}^{3} u_i^j \right\|_F^2 \\ \text{s.t.} \quad & u_i^j \in (\mathbb{R}_+^{d_i} \cap C(s_i^{\max})) \setminus C(s_i^{\min}), \quad j = 1, \ldots, k. \end{aligned} \tag{5.3}$$

This notation makes explicit that as before the constraints consist of a *convex* part $u_i^j \in \{\mathbb{R}_+^{d_i} \cap C(s_i^{\max})\}$ and a *reverse-convex* part $u_i^j \notin C(s_i^{\min})$. The two fundamental challenges to address are thus, first, the non-convex objective function, and, second, the reverse-convex min-sparsity constraint.

5.3.1 The Sparsity Maximization Algorithm (SMA)

We use two strategies to cope with the basic challenges in problem (5.3): First, to address the non-convexity of the objective function, we apply again an *alternate minimization approach* where only one component $u_i,\ i \in \{1,2,3\}$, is optimized at a time while the other two components are held constant. The resulting objective function is convex quadratic in each step.

To deal with the second challenge, the reverse-convex min-sparsity constraint, we adopt the same general approach from global optimization [Tuy87] as in Section 4.3: Given a current estimate for u_i we compute the maximally sparse approximation subject to the constraint that the reconstruction error does not deteriorate, and, dually, given a maximally sparse approximation we minimize the reconstruction error subject to the constraint that the min-sparsity constraint may not be violated.

Let us assume that within the alternate minimization approach ("outer loop") we optimize component u_i, while the components $\bar{I} := \{1,2,3\} \setminus \{i\}$ remain fixed. Then the target function $f(V,u) := \left\| V - \sum_{j=1}^k \bigotimes_{i=1}^3 u_i^j \right\|_F^2$ can be written as

$$f(V,u_i) := \|\mathrm{vec}(V) - U\mathrm{vec}(u_i)\|_2^2 \,, \tag{5.4}$$

where U is a sparse matrix containing the corresponding entries $u_i,\ i \in \bar{I}$, that are not currently optimized.

Initialization. We start with any u_i that obeys the constraints of (5.3). A simple way to obtain such an initialization is to first solve the problem ignoring the min-sparsity constraint, i.e.,

$$\begin{aligned} \min_{u_i} \quad & f(V,u_i) \\ \text{s.t.} \quad & u_i^j \in \mathbb{R}_+^{d_i} \cap C(s_i^{\max}), \quad j = 1,\ldots,k \end{aligned} \tag{5.5}$$

which is a SOCP that reads in standard form

$$\begin{aligned} \min_{u_i,z} \quad & z \\ \text{s.t.} \quad & 0 \le u_i \\ & \begin{pmatrix} \mathrm{vec}(V) - U\mathrm{vec}(u_i) \\ z \end{pmatrix} \in \mathcal{L}^{kd_i+1} \\ & \begin{pmatrix} u_i^j \\ (c_{d_i,s_i^{\max}})^{-1} e^\top u_i^j \end{pmatrix} \in \mathcal{L}^{d_i+1}, \quad j = 1,\ldots,k. \end{aligned} \tag{5.6}$$

The resulting u_i can then be projected on the boundary of the min-sparsity cone. Accuracy is of no concern in this step, so simple element-wise exponentiation followed by normalization

$$\pi(u_i^j) \propto \frac{(u_i^j)^t}{\left\|(u_i^j)^t\right\|_2} \tag{5.7}$$

with suitable parameter t, yields a feasible initialization.

Step one. In the first step we maximize worst-case sparsity subject to the constraint that reconstruction accuracy may not deteriorate:

$$\begin{aligned} \max_{u_i} \quad & \min_j \operatorname{sp}(u_i^j) \\ \text{s.t.} \quad & u_i^j \in \mathbb{R}_+^{d_i} \cap C(s_i^{\max}), \quad j = 1, \ldots, k \\ & f(V, u_i) \leq f(V, \bar{u}_i), \end{aligned} \tag{5.8}$$

where $\bar{u}_i$ is the estimate for u_i before sparsity maximization. Problems similar to (5.8) have been solved using cutting plane methods, however, such solvers seem to perform well for small to medium-sized problems only [Tuy87, HT96]. For the large scale problems common in computer vision and machine learning, we must content ourselves with a local solution obtained by *linearization* of the sparsity cone around the current estimate $\bar{u}_i$. The resulting problem is a SOCP:

$$\max_{u_i, z} \quad z \tag{5.9a}$$

$$\text{s.t.} \quad u_i^j \in \mathbb{R}_+^{d_i} \cap C(s_i^{\max}), \quad j = 1, \ldots, k \tag{5.9b}$$

$$f(V, u_i) \leq f(V, \bar{u}_i) \tag{5.9c}$$

$$z \leq \operatorname{sp}(\bar{u}_i^j) + \langle \nabla \operatorname{sp}(\bar{u}_i^j), u_i^j - \bar{u}_i^j \rangle, \quad j = 1, \ldots, k. \tag{5.9d}$$

Note that $\operatorname{sp}(x)$ is convex, so the linearization (5.9) is valid in the sense that min-sparsity will never decrease in step one.

Step two. In the second step we improve the objective function while paying attention not to violate the min-sparsity constraints. Given the sparsity-maximized estimate $\bar{u}_i$ we solve the SOCP

$$\min_{u_i} \quad f(V, u_i) \tag{5.10a}$$

$$\text{s.t.} \quad u_i^j \in \mathbb{R}_+^{d_i} \cap C(s_i^{\max}), \quad j = 1, \ldots, k \tag{5.10b}$$

$$\left\|u_i^j - \bar{u}_i^j\right\|_2 \leq \min_{q \in C(s_i^{\min})} \left\|q - \bar{u}_i^j\right\|_2, \quad j = 1, \ldots, k \tag{5.10c}$$

which is straightforward to translate to standard form. Note that constraints (5.10c) make sure that the resulting u_i^j will not enter the min-sparsity cone. In effect, the reverse-convex min-sparsity constraint is translated in (5.10) into a convex proximity constraint. This is similar to *trust region* approaches common in nonlinear programming.

Algorithm 5.3.1 The sparsity maximization algorithm in pseudocode.

1: initialize all u_i^j using eqn. (5.6) and (5.7), set $\bar{u} \leftarrow u$
2: **repeat**
3: **for** $i = 1$ to 3 **do**
4: **repeat**
5: $u_{\text{old}} \leftarrow u$
6: $\bar{u}_i \leftarrow$ solution of (5.9)
7: $u_i \leftarrow$ solution of (5.10)
8: **until** $|f(V, u_i) - f(V, u_{\text{old},i})| \leq \epsilon$
9: **end for**
10: **until** no improvement found in loop 3–9

Termination. After the second step we check whether $f(V, u_i)$ improved more than ϵ. If it did we jump to step one, otherwise we switch in the outer loop to a different factor i. The whole algorithm is outlined in Alg. 5.3.1.

5.3.2 Convergence Properties

The convergence properties Alg. 5.3.1 are similar to those examined in Chapter 4. We therefore will be brief and just outline the general ideas.

Proposition 8. *The SMA algorithm (Alg. 1) terminates in finite time for any sparsity-constrained NTF problem.*

Proof (sketch). Revering to Section 4.3 for details we note that:

- Step 1 consists of solving three convex programs and subsequent projections. These operations will terminate in polynomial time.
- Any current estimate u is a feasible point for the convex programs (polynomial time) in the inner loop (steps 6 and 7). Thus, with each iteration of the inner loop the objective value $f(V, u)$ can only decrease or remain constant.
- Since $f(V, u)$ is bounded from below, the inner loop will eventually terminate (step 8).
- And so will the outer loop (step 10) for the same reason.

□

The algorithm conveniently converges on a stationary point if the constraints are *regular*. Following [Tuy87] we call constraints regular if their gradients are linearly independent and if removing one would allow for a new optimum with lower objective value. From a practical viewpoint, this means that in particular we assume $s_i^{\min} < s_i^{\max}$, i.e., the interior of the feasible set is not empty.

Proposition 9. *Under regular sparsity constraints, Alg. 5.3.1 converges on a stationary point of problem* (5.2)*.*

Proof. The first order optimality conditions for problem (5.2) read:

$$-\frac{\partial L}{\partial u_i^*} \in N_{Q_i}(u_i^*), \tag{5.11a}$$

$$G_i(u_i^*) \in \mathbb{R}_+^k, \tag{5.11b}$$

$$\lambda_i^* \in R_-^k, \tag{5.11c}$$

$$\langle \lambda_i^*, u_i^* \rangle = 0, \tag{5.11d}$$

where i runs from 1 to 3. Here,

$$L(u, \lambda_1, \lambda_2, \lambda_3) = f(V, u) + \sum_{i=1}^{3} \lambda_i^\top G_i(u_i) \tag{5.12}$$

is the Lagrangean of the problem and

$$G_i(u_i) = \left(\left\| u_i^1 \right\|_2 - (c_{d_i, s_i^{\min}})^{-1} \left\| u_i^1 \right\|_1, \cdots, \left\| u_i^k \right\|_2 - (c_{d_i, s_i^{\min}})^{-1} \left\| u_i^k \right\|_1 \right)^\top \tag{5.13}$$

encodes the min-sparsity constraints: $G_i(u_i)$ is non-negative if the min-sparsity constraints on u_i are adhered to. Finally, N_{Q_i} in (5.11a) is the normal cone [RW98] to the convex set $Q_i = R_+^{d_i \times k} \cap C(s_i^{\max}),\ i = 1, \ldots, 3$.

Now assume the algorithm converged (Prop. 8) on a point $\tilde{u}$. Because $\mathrm{sp}(\cdot)$ is convex and the constraints are regular we find that (5.9d) is locally equivalent to $z \leq \mathrm{sp}(\tilde{u}_i)$. In fact, $z = s_i^{\min}$ because the min-sparsity constraint is active for some vector $\tilde{u}_i^j$: Otherwise we could remove the constraint without changing the objective value of the solution.

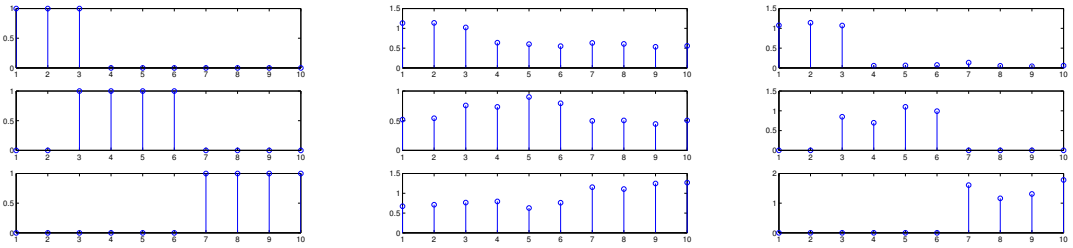
Overall, we find that the solution to (5.9) satisfies

$$\begin{aligned} \max_{z, u_i \in Q_i} \quad & z, \\ \text{s.t.} \quad & z = \min_i \mathrm{sp}(u_i), \\ & 0 \leq f(V, u_i) - f(V, \tilde{u}_i), \\ & G_i(u_i) \in \mathbb{R}_+^k. \end{aligned} \tag{5.14}$$

Then the solution obeys the corresponding first order condition

$$-\frac{\partial}{\partial u_i} \left(\hat{\lambda}_{fi} f(V, u_i) + \langle \hat{\lambda}_{ui}, G_i(u_i) \rangle \right) \in N_{Q_i}(u_i^*) \tag{5.15}$$

which is equivalent to (5.11). ∎



(a) Ground truth. (b) Classical NTF. (c) Sparse NTF.

Figure 5.2: **Ground truth experiment.** We created an artificial data set with known factors u_i (Figure 5.2(a)). We added noise (see text) and used NTF to recover the factors from $V = u_1 \otimes u_2 \otimes u_3 + |\nu|$. While the NTF model without sparsity constraints failed (Figure 5.2(b)), sparsity-controlled NTF successfully recovered the factors (Figure 5.2(c)).

5.3.3 Practical Considerations

The SOCP problems (5.9) and (5.10) are sparse but can become very large. Solvers with support for sparse matrices are crucial[1]. In applications where the convex max-sparsity constraints are not used, i.e., only min-sparsity constraints are specified, quadratic programming (QP) solvers can be used instead of SOCP solvers. Commercial QP solvers are usually highly optimized and may be faster than their SOCP counterparts.

5.4 Experiments

In this section we show that our optimization framework works robustly in practice. A comparison demonstrates that explicit sparsity-control leads to improved performance. Our results validate that sparsity-controlled NTF can be a useful model in real applications.

5.4.1 Ground Truth Experiment

To validate our approach we created an artificial data set with known ground truth. Specifically, we used three equally-sized factors u_i with $d_i = 10$ and all entries zero except for the entries shown in Figure 5.2(a). We computed $V = u_1 \otimes u_2 \otimes u_3 + |\nu|$, where $\nu \sim \mathcal{N}(0, 0.5)$ was i.i.d. Gaussian noise.

We found that over 10 repeated runs the classical NTF model without sparsity constraints was not able to recover any of the factors (Figure 5.2(b)). In contrast, sparsity-controlled NTF with $s_i^{\min} = 0.55$ yielded useful results in all 10 repeated runs (Figure 5.2(c)).

[1] In our experiments we used MOSEK 3.2.1.8 [Mos05].

feature	pixels	NMF	NTF	NTF	NTF	NTF	NTF	**NTF**	NTF
$s_1^{\min}$			0.0	0.3	0.4	0.6	0.7	**0.8**	0.9
ROC (trai)	0.997	0.995	1.000	1.000	0.997	0.997	0.994	**1.000**	0.991
ROC (test)	0.817	0.817	0.835	0.822	0.789	0.830	0.822	**0.860**	0.821
ACC-50 (test)	0.611	0.667	0.753	0.600	0.702	0.743	0.728	**0.761**	0.719

Table 5.1: **Recognition performance of sparse NTF codes.** We trained a SVM on a subset of the MIT CBCL face detection data set (see text). Features were raw pixels, a NMF basis, and a NTF basis with different min-sparsity constraints. We compared area under ROC for the MIT training data (first row), the MIT test data set (second row) and recognition accuracy for a balanced test data set with 50% face samples (last row). NTF with a relatively strong min-sparsity constraint $s_1^{\min} = 0.8$ performs best.

We conclude that in the presence of noise, sparsity constraints are crucial to successfully recover sparse factors. Further, we find that at least with the simple data set above the sparsity maximization algorithm converged on the correct factorization in 10 out of 10 repeated runs.

5.4.2 Face Detection

For the face detection problem, impressive results are reported in [HPS05] where NTF without sparsity constraints clearly outperformed NMF recognition rates on the MIT CBCL face data set [CBC00]. We demonstrate in this section that performance can further be improved by using sparsity-constrained NTF.

In our experiments we used the original training and test data sets provided by CBCL. In this data sets, especially the test data set is very imbalanced: A trivial classificator returning "non-face" for all input would obtain 98% accuracy. For this reason, we consider the *area under the ROC curve* as a more suitable performance measure: It is more meaningful for highly imbalanced data sets. We thus trained radial-basis function SVMs on small subsets (250 samples only) of the CBCL training data set. To determine the SVM and kernel parameters, we used 5-fold crossvalidation on the training data. For the resulting SVM we determined the area under the ROC on the test data set. In addition, we also created a data set ACC-50 consisting of all 472 positive samples in the test data set as well as of 472 randomly chosen negative test samples.

We compared the following feature sets:

1. the $19 \times 19 = 361$ raw image pixels as found in the CBCL data set,
2. coefficients for 10 NMF basis functions determined on a subset of the faces in the training data set,
3. coefficients for 10 NTF basis functions determined on a subset of the faces in the training data set using different values of the min-sparsity constraint on u_1.

Reconstructions using these features are shown in Figure 5.1. Note that the NTF basis corresponds to an about 10-fold higher compression ratio than the NMF basis.

The results are summarized in Table 5.1: NMF and raw pixel values perform similar in this experiment. NTF yields improved results, which is consistent with [HPS05]. Best results are obtained with NTF with strong sparsity constraint ($s_1^{\min} = 0.8$).

5.5 Summary

We extended the non-negative tensor factorization model for images [WW01, SH05, HPS05] by explicit sparseness constraint [Hoy04]. We found that compared to unconstrained NTF the extended model can be more robust against noise (Section 5.4.1) and the corresponding image codes can be more efficient for recognition, especially when training data is scarce (Section 5.4.2).

From an optimization point of view, we devised an algorithm based on sequential conic programming (Section 5.3.1) which has desirable convergence properties (Section 5.3.2) and works well in practice (Section 5.4). Because the algorithm's basic building blocks are convex programs, we believe the model could further be extended by additional convex constraints taking into account prior knowledge about the specific problem at hand, while still remaining in the sequential convex programming framework.

Chapter 6

Applications

In this chapter we describe some realistic applications that are not only intrinsically motivated by our research, but may be of interest to an audience outside the computer vision community as well.

6.1 Medical Imaging: Recovering SPECT Factors

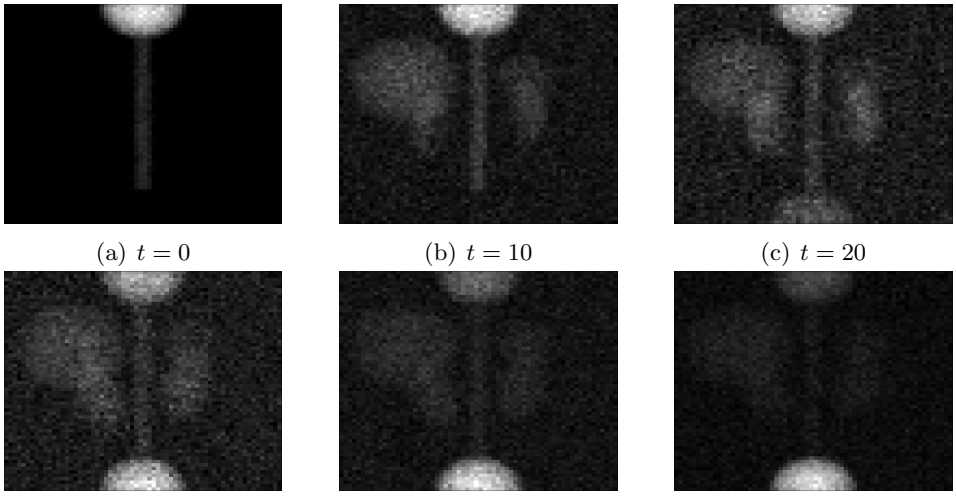
Single photon emission computed tomography (SPECT) is an imaging technology from nuclear medicine: Radioactive substances are injected in the blood stream and the resulting γ-rays are recorded. As the radio-pharmaceutical travels through the body, different organs become radioactive and visible for γ-cameras.

In a simplified model used by Nagy, Kuba, and Samal organs are represented by binary functions $f_k : \mathbb{R}^3 \to \{0, 1\}$ on the 3D space: $f_k(x)$ equals one iff x belongs to organ k [NKS05]. In a first approximation, one can model the SPECT imaging process by introducing coefficients $c_k : \mathbb{R}_+ \to \mathbb{R}_+$ describing the temporal behavior of the radio-pharmaceutical w.r.t. organ k. Overall, the 3D situation is then given by [NKS05]:

$$g(x,t) = \sum_k c_k(t) \cdot f_k(x) + \eta(x,t), \tag{6.1}$$

where η is noise. In our application both, f_k and c_k, are unknown.

The question, of course, is now: Given a sequence of images, can we recover c_k and f_k? To solve the full problem in 3D requires inverting the projection operation. This is not within the scope of this work. Rather, we will concentrate on how to factorize the organs *directly using the projection data* given by the γ-cameras. This is an important first step toward solving the 3D problem [NKS05].



(a) $t = 0$ (b) $t = 10$ (c) $t = 20$

(d) $t = 30$ (e) $t = 40$ (f) $t = 50$

Figure 6.1: **SPECT sequence.** One projection of the simulated SPECT sequence. As the radio-pharmaceutical travels through the body, different organs become visible and vanish over time. Note the relatively high amount of noise in the images.

6.1.1 The Dataset

For this study, we use an artificial data set, in medical image processing called *phantom*, developed by Backfrieder [BSB99]. It consists of 5 simplified organs corresponding to heart and aorta, liver and spleen, the renal parenchymas, the renal pelvises, and the bladder (Figure 6.1). The organs are modeled as homogeneous voxel volumes. We have four views (projections) of a 64×64 voxel volume recorded at $t = 120$ time steps.

Unfortunately, the original volume is not available, so there is no ground truth to test our factorizations against. However, we do have available a previous factorization of the data obtained using a combination of factor analysis, varimax rotation [Kai58], and supervised selection of background regions [ŠKS+87, NKS05].

In the simulation, a radio-pharmaceutical is injected at $t = 0$, and over time the different organs become visible. In addition, the dataset is corrupted by a fair amount of noise which is typical for images recorded by γ-cameras.

6.1.2 Experiments

In order to factorize the data we first ran standard NMF on each of the four projections independently. The results were not satisfying (Figure 6.2): The organs are not separated into different factors. In particular, the heart and aorta factor (Figure 6.2(a)) appears in two other factors (Figure 6.2(b) and 6.2(d)) and, consequently, the diffusion coefficients are incorrect (Figure 6.2(i)). The latter is particularly unfortunate since the diffusion coefficients contain important information about the health status of each organ.

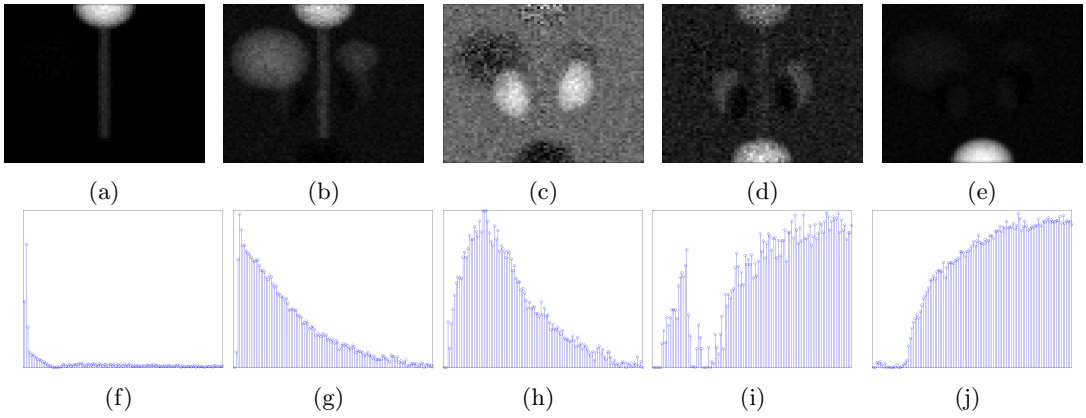


Figure 6.2: **Factorization of the SPECT sequence (failure).** Applying NMF alone straightforwardly does not yield a successful factorization: Factor (a) is also present in (b) and (d), factor (e) re-appears in (d). The coefficients (i) are highly unrealistic.

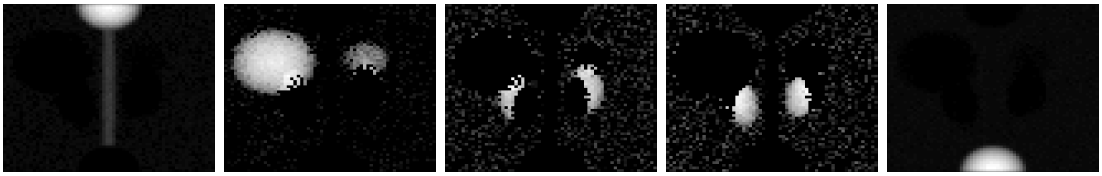
Projection	Error FA	Error QP	Error QP/sp
1	11416.7	3408.3	3409.0
2	11987.5	3307.5	3307.4
3	17891.7	3778.1	3792.8
4	3581.5	3581.5	3579.7

Table 6.1: **Reconstruction error for different projections.** The reconstruction error $f(W,H) = \|V - WH\|_F$ for the 4 different projections are listed for the factor analysis method (second column), NMF/QP (third column) and NMF/QP with sparse initialization (fourth column). For FA 6 factors have been used, for NMF 5 factors were sufficient. Note that except for projection 4, where the results are identical, the reconstruction error of the QP solver is about three times smaller than the error of the factor analysis method.

Note that the reconstruction error $f(W,H) \equiv \|V - WH\|_F$ is small and in particular much smaller than the reconstruction error of the previously used semi-supervised factor analysis method that nevertheless yielded acceptable factorizations (Table 6.1). So it seems that there are *multiple local mimima, only few of which are relevant* for the application at hand.

There are two principal ways to deal with this situation: First, extend the NMF objective function or the constraints to favor suitable factorizations. The second approach is to start not from arbitrary initializations but close to favorable solutions.

The characteristics of favorable factorizations are: First, no organ appears in more than one factor, and, second, the coefficients model the diffusion process realistically. In particular, we expect the coefficients to be continuous, unimodal functions over time. This leads directly to two possible additional energy terms: To prevent multiple appearances of organs we penalize the sum of the mutual inner products $e^\top W^\top W e$. To favor con-



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

Figure 6.3: **Sparsity-constrained SPECT factors.** With a sparsity constraint ($s_w = 0.6$) we obtain a factorization that separates the organs very clearly. These factors can be used directly or employed as initialization for a subsequent standard NMF optimization.

tinuous, unimodal coefficient functions we additionally penalize $\|\nabla H_{i\bullet}\|$. The resulting energy function reads

$$\hat{f}(W,H) = \|V - WH\|_F^2 + \lambda_1 e^\top W^\top W e + \lambda_2 \sum_i \|\nabla H_{i\bullet}\|_1 \,. \tag{6.2}$$

Note that the new terms are quadratic and linear, respectively, so they are easily integrated in our previously presented framework (Section 4.3). Unfortunately, from an application point of view, choosing the new parameters λ_1, λ_2 in (6.2) is *not* straightforward. This is a serious limitation when unsupervised operation is desired.

Surprisingly, the second approach, starting close to favorable solutions, is easier to use: By applying a sufficiently strong sparsity constraint on W we obtain initializations that are highly unlikely to model the same organ in multiple factors (Figure 6.3). Starting from such initializations a subsequent standard NMF optimization obtains factors that combine good reconstruction properties with good identification of the organs (Figure 6.5). Also, the reconstruction of the diffusion coefficients can improve significantly (cf. Figure 6.2(i) and 6.4(x)).

The advantage of the sparsity-constrained initialization over (6.2) is that the precise value of the min-sparsity constraint seems not critical: As long as it is large enough a sparse solution will be obtained and possibly missing pixels or parts will be restored in the subsequent standard NMF optimization.

Finally, note that it makes sense to factorize as many projections as possible *at once*: The diffusion coefficients are independent of the view and thus should be identical for each projection. With more views, however, more image information enters the optimization process, thus allowing views to be factorized correctly that would, on their own, not provide enough information for unambiguous factorization.

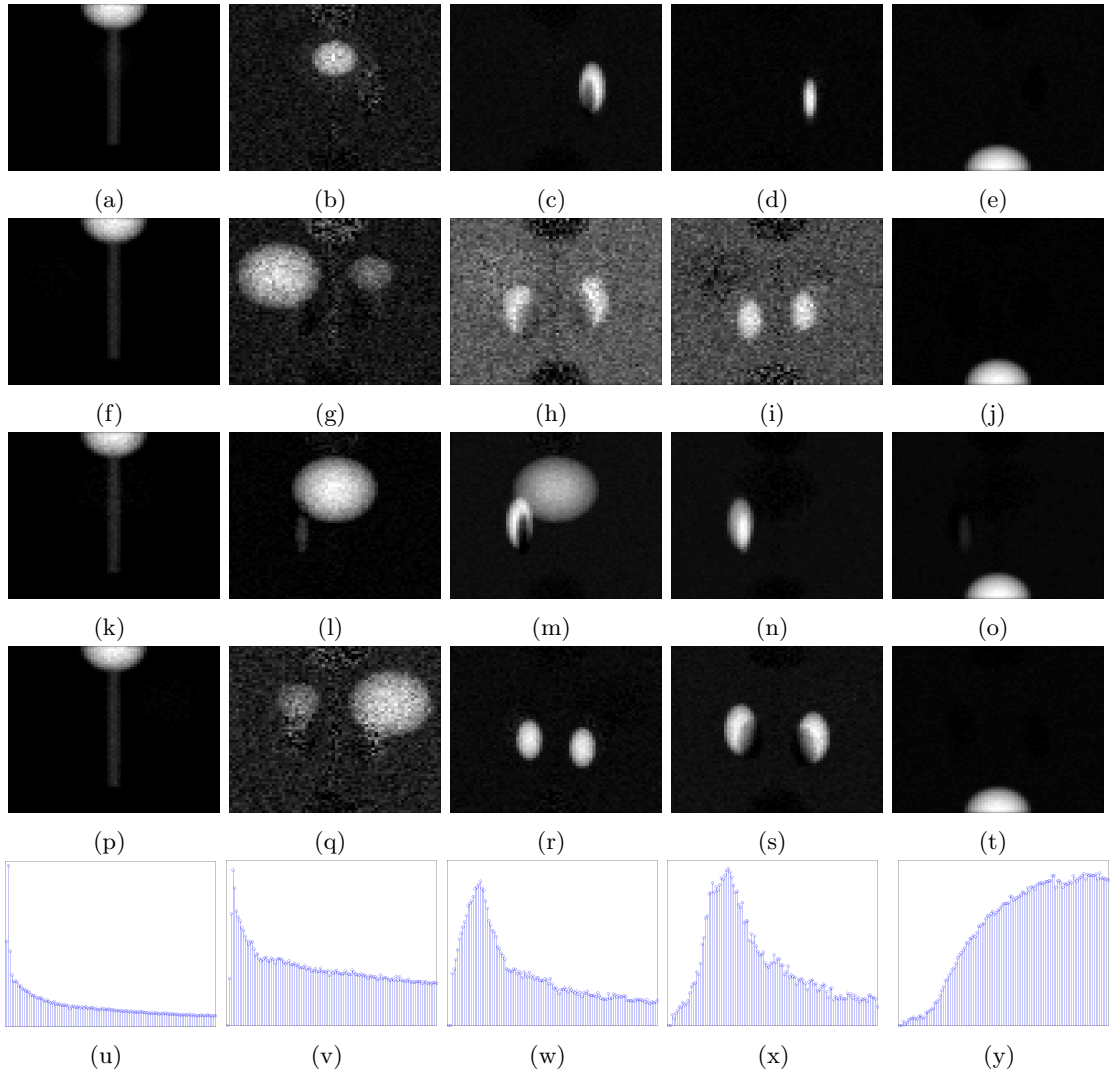
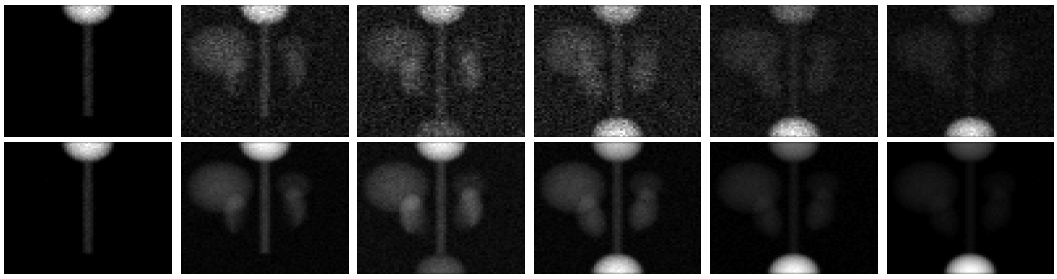


Figure 6.4: **Complete factorization of the SPECT sequence.** For each view (rows) five factors $f_k(x)$ are depicted using NMF with sparse initialization. Except for one duplicate in (l)-(m) the the factors correspond to anatomically meaningful regions and are well separated. In the bottom row we show corresponding diffusion coefficients $c_k(t)$ which look much more realistic than those in Figure 6.2. Note that for visualization some factors are amplified so that the noise level appears increased.



$t = 0$ $t = 10$ $t = 20$ $t = 30$ $t = 40$ $t = 50$

Figure 6.5: **SPECT sequence reconstruction.** Reconstruction of the SPECT sequence (second row). The original data is shown for comparison (first row). The data is almost perfectly reconstructed from the NMF factorization. Note that the noise level is significantly reduced in the reconstruction.

6.2 NMF-Based Image Classification

As outlined in the introduction (Section 1.1.3) non-negative matrix factorization (Section 4) is applicable to a number of key problems in computational vision. In this section, we develop as a proof-of-concept a system for weakly supervised learning of visual concepts.

6.2.1 Factorization for Semantic Analysis

Latent semantic analysis (LSA) is a model from statistical natural language processing (NLP) [DDL+90]. The idea is to treat text as a collection of *topics* that represent, in turn, *bags of words*. In analogy, Sivic et al. [SRE+05] propose *visual words* to explain images. Visual words are representative image patches obtained either through clustering [MBLS01, JT05] or by mere random sampling [NJT06]. *Visual topics* then correspond to collections of image patches which co-occur frequently. They can be found, e.g., by using LSA. Interestingly, although these models do, in their simplest form, not account for any geometric relations —there is no general agreement on a "visual grammar" yet— recognition performance is state-of-the-art as long as the vocabulary of image patches is chosen large enough [NJT06].

To understand the role of matrix factorizations in these applications, let $V \in \mathbb{N}_0^{m \times n}$ denote a matrix where V_{ij} records how often word (image patch) i occurs in document (image) j. Let the topics be stored in a $m \times k$ matrix of word (patch) frequencies W. The LSA model assumes that the documents can be explained by the weighted sum of a relatively small number of topics ($k \ll n$) so that with weights $H \in \mathbb{R}^{k \times n}$

$$V \approx WH. \tag{6.3}$$

So, in the bag-of-words model semantic analysis leads to a matrix factorization problem.

Classical Latent Semantic Analysis

The original algorithm for LSA is based on singular value decomposition (SVD): The matrix V of word counts is factorized to solve problem (4.1). This leads to a *low-dimensional* vector space representation of documents in the so-called *latent semantic space*.

Representing text documents in latent semantic space offers several benefits. A particularly important one is that machine learning is often easier to do in low-dimensional spaces. Also, the tools of differential geometry become available which can lead to improved classification algorithms [LL05].

Probabilistic Latent Semantic Analysis and Non-Negative Matrix Factorization

An undesirable property of latent semantic spaces is that since the basis vectors of a SVD are orthogonal some topics will in general have negative entries. This contradicts the notion that LSA aims at topics that are given by text semantics: A negative word-count in a topic can be difficult to comprehend [XLG03, Section 3].

Probabilistic latent semantic analysis (PLSA) is a probabilistic alternative to LSA that does not share this drawback [Hof99]. Here, the word count matrix $V_{w,d} \propto p(w,d)$ is described probabilistically where the topics are modeled as *latent variables*, such that relative word counts $p(w|z)$ and documents $p(d|z)$ are conditionally independent given the topics:

$$p(w,d) = \sum_z p(w|z) \cdot p(d|z) \cdot p(z). \tag{6.4}$$

To simplify notation, let us assume that w, d, and z are represented by integer ranges starting from one, so their values and their indices coincide. As pointed out by Hofmann [Hof99] (6.4) can be then written in matrix notation: With $W = (p(w|z))_{w,z}$, $H^\top = (p(d|z))_{d,z}$, and $\Sigma = \operatorname{diag}(p(z))$, we write

$$V = W\Sigma H. \tag{6.5}$$

Of course, all matrices in (6.5) contain probabilities and are thus non-negative. This notation makes explicit that *PLSA is a special NMF problem*[1].

From this perspective it is no longer surprising that NMF outperformed LSA in text classification [XLG03, SBPP06] and works well in probabilistic clustering [DHS05, ZS05].

6.2.2 Patch-Based Image Representations

Unlike with text, for images it is not trivial to decide what constitutes the "words" counted in the document count matrix V. Two alternatives are, first, to use *interest points* [FE87, HS88, SMB00, Low04, KS04, MS04] and locate sparsely sampled patches [BLP95, WWP00, LS03, FPZ03, AAR04, SRE+05]. The second alternative is to sample

[1] In fact, note the similarity between the EM update step for W which reads [Hof99, eqn. (4)]:

$$p_{\text{new}}(w|z) \propto \sum_d V_{w,d} p(z|d,w) = \sum_d V_{w,d} \frac{p(d|z)p(w|z)p(z)}{\sum_{z'} p(d|z')p(w|z')p(z')}\,. \tag{6.6}$$

and the NMF update for W proposed by Lee and Seung [LS00, eqn. (4)]:

$$W_{\text{new}} \propto (W \odot (VH^\top)) \oslash (WHH^\top) \tag{6.7}$$

which, using the probabilistic notation from above, can formally be written as

$$W_{w,z} \propto \sum_d V_{w,d} \frac{p(d|z)p(w|z)}{\sum_{d',z'} p(w|z')p(d'|z')p(d'|z)}. \tag{6.8}$$

patches or textons densely from the whole image plane [UVNS02, WCM05, NJT06]. In either case, the sampled image patches are then clustered and the resulting representatives serve as "words" to be recognized in the images. There is evidence that dense sampling can yield better recognition rates, although due to the statistics of natural images (cf. Section 3.1.3) clustering algorithms that rely on estimated mean and variance of the cluster distributions might be unreliable [JT05].

Next, we describe some experiments with our proof-of-concept system. To be able to fairly assess the performance of the NMF-based classifier we implement a brute-force method first. The brute-force method is based on direct and complete search of image patches in a database of labeled or unlabeled images. While this is wasteful from a computational point of view the resulting classification rates are important to put the NMF performance in perspective.

Brute-Force Classification

As base classifier we examine a supervised system that relies entirely on next-neighbor classification of independently sampled image patches.

Setup. Specifically, in each experiment we randomly sample k image patches $q_i, i = 1, \ldots, k$ from an image chosen randomly from a database of labeled images. Using the remaining images from the database, we then estimate the likelihood $p(q_i|c)$ that patch q_i is sampled from an image of class c. Then we combine this evidence by a max-likelihood estimator for the class c:

$$c_{\mathrm{ML}} = \arg\max_c \prod_{i=1}^{k} p(q_i|c) \ . \tag{6.9}$$

Here, we assume that the patches q_i are conditionally independent given class label c.

There are three open questions to be answered in order to implement this system: First, how do we model $p(q_i|c)$? Second, how large are the patches? Third, how many patches are to be used?

Concerning $p(q_i|c)$, we adopt a simple voting scheme where *normalized correlation* between each patch q_i and *each* of the remaining images in the database is computed. The database image where the largest normalized correlation for patch q_i is found casts a "vote" for its corresponding class[2]. The class most frequently voted for determines c_{ML}. Concerning size and number of patches we conducted various experiments where the patch size was chosen from $\{3, 11, 19, 27\}$ and k was either 5, 10, or 50 (Table 6.2).

[2]In a preliminary experiment we examined the top-n matches and used histograms to cast soft, probabilistic votes instead of the all-or-nothing approach. The results were similar.

k	3	11	19	27
5	0.52	0.68	0.73	0.76
10	0.55	0.79	0.82	0.84
50	0.53	**0.90**	0.89	0.87

Table 6.2: **Classification performance of brute force approach.** Images from the Caltech database were classified using varying number of patches k (rows) and varying patch sizes (columns). The overall classification performance is reported (see text for details). A patch size of 11 achieves best results.

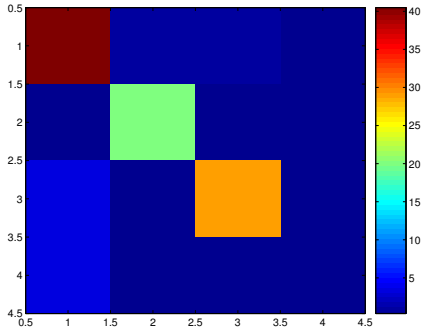


Figure 6.6: **Confusion matrix for brute force approach.** The rows correspond to true classes (1: motorcycles, 2: airplanes, 3: faces, 4: cars), the columns to the prediction of the brute force classifier with 50 patches of size 11×11 (cf. Table 6.2). The car class is most difficult to classify partly due to relatively little data available. The colors encode percentages.

Results. We used "Cars 1999 (Rear) 2", "Motorcycles 2001 (Side)", "Airplanes (Side)", and "Faces 1999 (Front)" from the Caltech-101 image database [FFFP04] for our experiments. Images were converted to gray scale and scaled to 120×300 pixels. Overall we randomly selected 500 images to classify and report the performance in Table 6.2 and Figure 6.6: We obtained classification rates up to 90%. In comparison, a classifier returning random labels would, since the class labels are uniformly distributed, obtain 33% accuracy on this data set.

While classification accuracy of the simple brute force approach is impressive the computational effort is outstanding: To create the results in Table 6.2 took over a week of continuous computation[3]. Huge computing resources would be needed to scale this approach to larger and more realistic databases.

[3]We run a Matlab 7.0.1 script on a 3GHz 64bit AMD CPU. Normalized correlation, the computationally most expensive operation, was implemented as a MEX extension in C.

NMF-based Classification

In the previous section we have seen that patch-based image representations combined with a basic statistical model can yield good classification performance on the Caltech data set. A drawback of this brute-force technique is that it requires a large number of normalized correlations that are computationally very expensive. In contrast, NMF factorization, the critical step in the following experiments, takes less than 10 minutes and often only 30 seconds to compute for our data.

In this section we adopt recent work by Sivic et al. and Nowak et al. to the NMF setting [SRE+05, NJT06] to solve the classification problem much more efficiently.

Setup. As in Section 6.2.2 we use four classes from the Caltech-101 database (2157 images) and sample 3000 patches on three different scales from each image. Specifically, the images are scaled by factors $\{0.7, 1.0, 1.3\}$ and each time 1000 patches of size 9×9 sampled. Unlike in the in the brute-force approach this is all the information we use from the images, i.e., we do *not* look for the globally best matches of the patches in the image database.

Then we divide the dataset into a test- and a training data set of equal size. Patches from the training images are clustered into 1000 representatives using only two iterations of k-means. Then we compute a word-frequency matrix V and *binarize* it using thresholds that maximize mutual information between class labels and the resulting binary vector [NJT06]. Note that the binarization is the *only* point where information about the true class labels enters the training process. The resulting binary matrix is factorized by NMF with varying number of basis functions.

Results. Using the NMF basis and the thresholds determined from the training data we compute reconstruction coefficients H for the test images. From these coefficients we obtain cluster labels by assigning each image i the index j that maximizes H_{ji}.

In Table 6.3 we show three different performance measures averaged over 10 repeated runs: We report the *adjusted rand index* [Ran71, HA85] and classification accuracy. The adjusted rand index measures the correspondence between two partitions of a set of objects. It equals one if the partitions are identical. The expectation of the adjusted rand index of two random partitions is zero. For the accuracy criterion we search for the optimal mapping from cluster label to true class label and then measure the classification error.

We find that with $r = 8$ NMF basis functions a classification accuracy of 82 percent can be obtained. Examining a confusion matrix (Figure 6.9) we see that—as in the brute-force approach—the car class is consistently misclassified. The most likely reason is that only 5.7% of the images are car images. Since information about the class labels enters our training process only in the form of thresholds for the binarization step the car images vanish in the noise of quite diverse backgrounds.

dimension r	adjusted rand index	accuracy
2	0.32	0.64
3	0.44	0.74
4	0.46	0.75
5	0.43	0.75
6	0.42	0.78
7	0.39	0.80
8	0.39	0.82
9	0.34	0.81
10	0.33	0.82

Table 6.3: **Performance of NMF categorization.** NMF image representations using a varying number of basis functions r. Different performance measures for the clusterings of a test data set, averaged over 10 repeated runs, are reported (see text). With $r = 8$ about 82% accuracy is obtained.

Looking at the most descriptive image patches (Figure 6.10) reveals a second drawback of our data set: Some images in the motorbike and airplane classes have white frames or borders and sometimes very homogeneous backgrounds. This background is already a strong clue for class membership. On the other hand, some features clearly concentrate on semantically relevant image parts as well.

To examine this further we ran a relatively high-dimensional NMF factorization on our training data set: Using $r = 50$ basis functions to represent the 1000-dimensional feature vectors resulted in a sparse factorization where only few patch clusters—those that frequently co-occur in the images—where active at a time. In Figure 6.11 we show some results: While many factors model noise or background we see that some face features (eyes, eyebrows, hair), and parts of the motorbike (wheels, seat) are distinguished. Using a much larger training data set would probably reduce the influence of background clutter. Alternatively, using *pairs* of image patches instead of single patches is also reported to lead to clearer labellings [SRE+05].

To see that the approach works also in the absence of white frames and homogeneous backgrounds we repeat our experiment using a database for human pose estimation [BKS06]. The resulting factorization (Figure 6.12) into $r = 4$ categories corresponds to people indoors, people outdoors, soccer players, and background (mostly pictures of an empty office environment).

6.3 Summary

In this chapter we described applications in medical image processing and image categorization where NMF plays a key role. In the first case, we saw that control over sparsity was crucially important to solve the application problem at hand. In the example of image categorization we found that a straightforward brute-force approach could

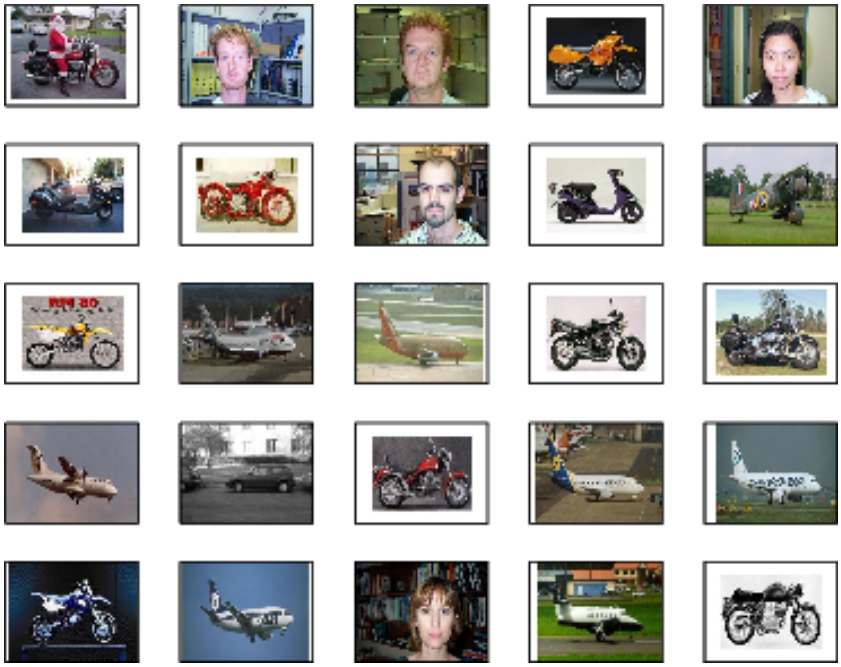


Figure 6.7: **Sample images from the database used.** A random sample of the subset of the Caltech-101 image database used. The classes were motorcycles, airplanes, cars (side view), and faces. Note that objects are roughly centered and some images have white frames. The car class is underrepresented compared to the other classes.

already obtain impressive results on a frequently used data set. However, it did so at a considerable computational cost. Using NMF reduced the processing time of a categorization from hours to minutes. In the future, it will be interesting to see if and how variations of NMF can further improve results and whether similar approaches will work for patch-based image segmentation.

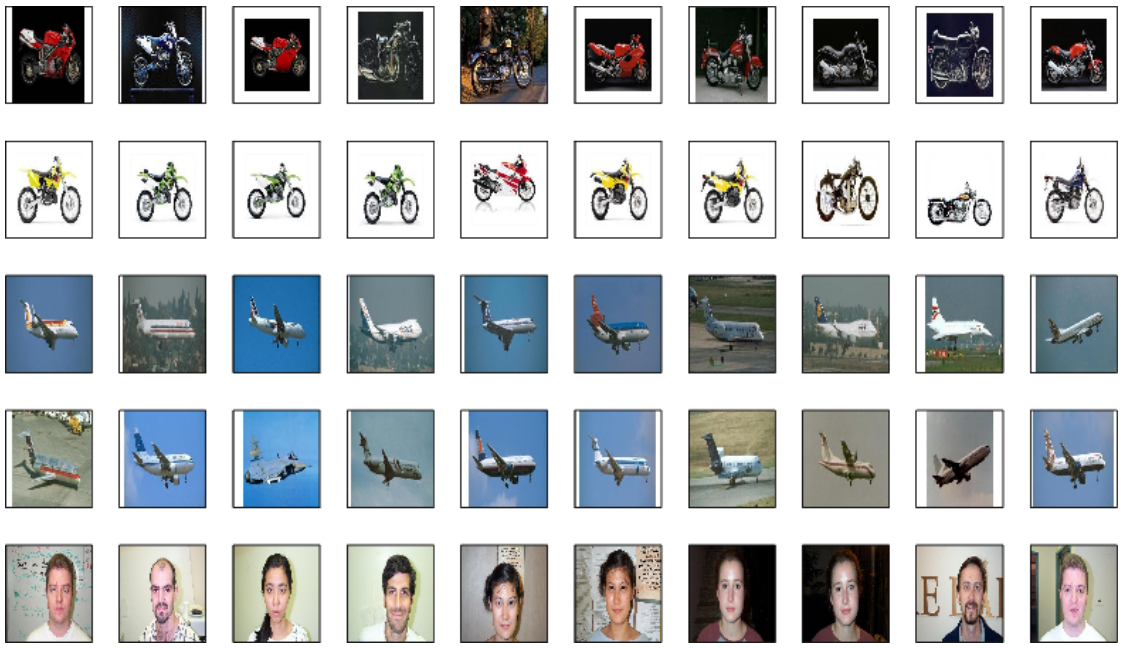


Figure 6.8: **NMF-based image categories.** A subset of the Caltech-101 database has been categorized using 5 NMF bases. The top-ranked images are depicted for each basis (rows). The first basis corresponds to motorcycles with dark background, the second basis models motorcycles with light background, basis three and four model airplanes, and basis 5 corresponds to faces. See text for details of the experiment.

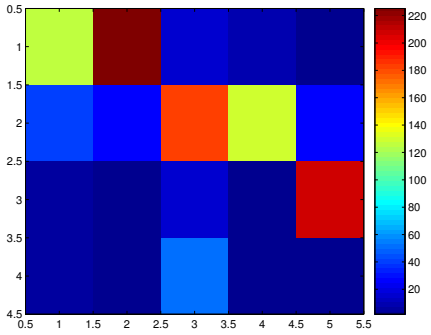


Figure 6.9: **Confusion matrix for NMF categories.** The rows correspond to true classes (1: motorcycles, 2: airplanes, 3: faces, 4: cars). The columns correspond to 5 NMF basis factors (cf. Figure 6.8 and text for details). The motorcycles and the planes are divided into two subcategories each. Faces are accurately modeled by basis number 5. The car class is confused with a plane class. Note that there were relatively few cars in the training data set.

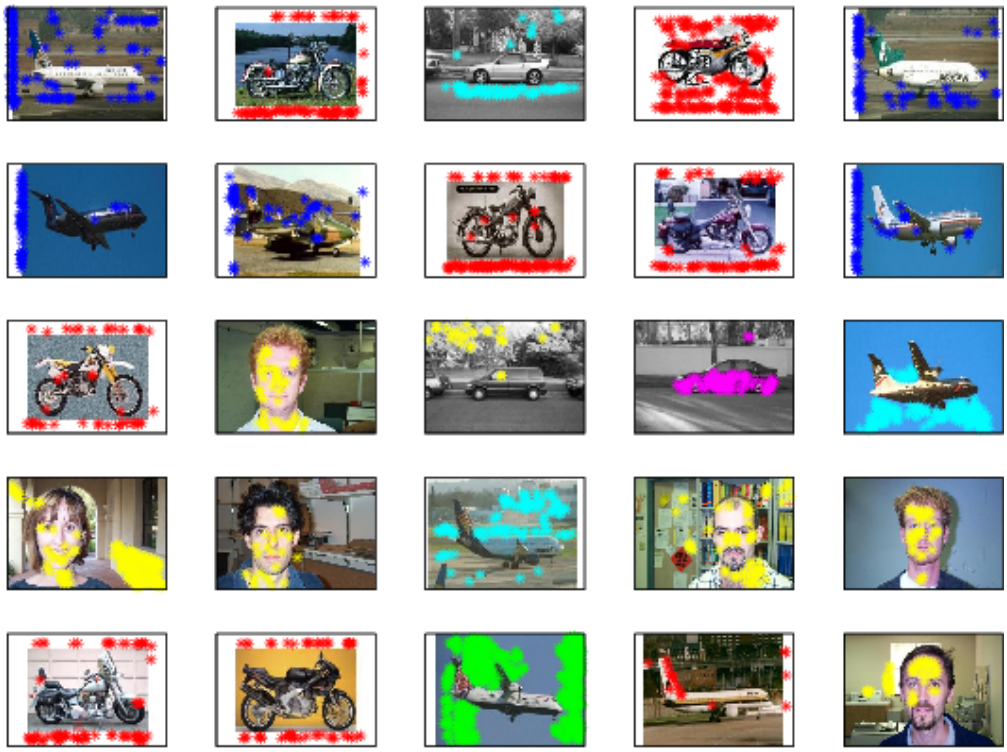
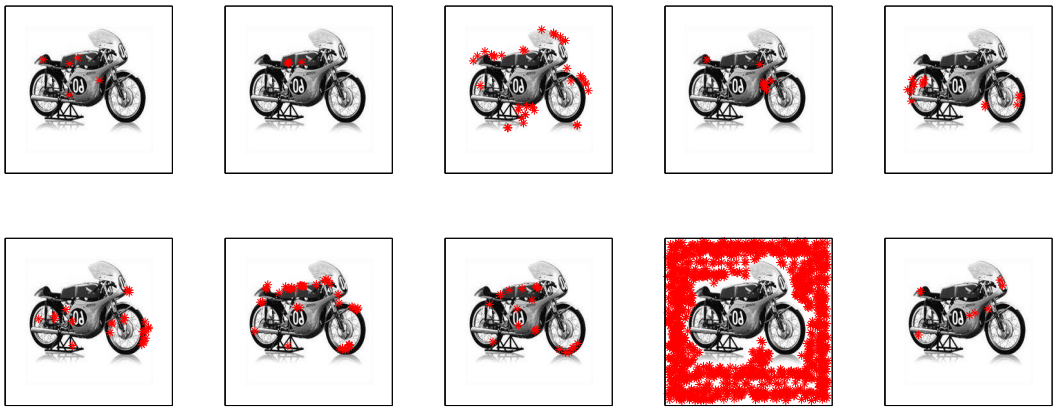
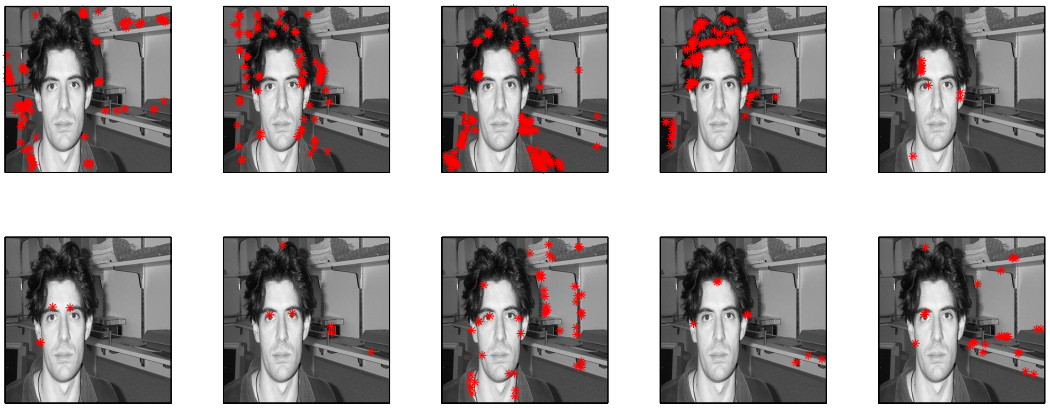


Figure 6.10: **Descriptive patches.** Descriptive image patches as selected by the NMF factorization (see text). Note that on the individual patch level categorization can be quite inconsistent. However, several intuitively relevant features are detected: For instance, face features are consistently selected, chassis and wheels are relevant, etc. White frames and borders, an artifact of the Caltech-101 dataset, are descriptive for the motorcycle image class. (See text for discussion.)



(a) Motorbike



(b) Face

Figure 6.11: **Co-occurring patches as extracted by NMF.** NMF with a relatively big number of basis function ($r = 50$) was used to find visual words that co-occur frequently in the database. While this completely unsupervised process captures much noise some semantically meaningful factors are modeled as well: For instance there is a group of words that roughly correspond to wheels of a motorbike, or to hair or to eyes.

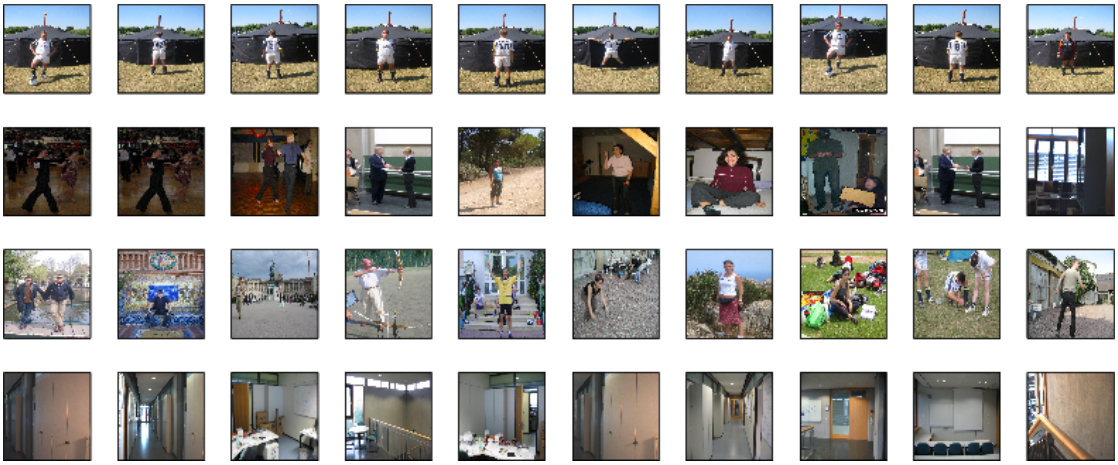


Figure 6.12: **Analyzing the people database.** A database used for human pose estimation [BKS06] was clustered using NMF: Images of humans and of an human-free office background are processed. The top-scoring images for four NMF basis functions are reported. Factor one (first row) models a series of shots of amateur soccer players, the other factors correspond to people indoors (second row), people outdoors (third row), and the office background (last row), respectively. No information about soccer players or indoor/outdoor scenes was provided during the training process.

Chapter 7

Summary and Outlook

In this thesis we considered image models for segmentation and classification and their associated optimization problems. Our work can roughly be grouped into two parts: In the first part we consider PDE-based, variational approaches; in the second part we adopt a mathematical programming viewpoint to solve sequences of convex programs.

From a practical point of view our main results are that (1) we can obtain surprisingly sophisticated segmentations of natural scenes using comparatively elementary statistical image models only (Chapter 3), and (2) that we can improve upon existing algorithms for matrix and tensor factorizations by adapting results from global optimization theory (Chapter 4 and 5), resulting in significant improvements in applications (Chapter 6).

Our work also motivates further research in various directions: For instance, we are not aware of a derivation of the generalized Laplacian distribution from physically realistic axiomatics. We have shown (Section 3.1.3) that explaining images as records of random light rays leads to statistics described by α-stable Lévy distributions. The parameters we found lead to distributions with have much larger entropy than the corresponding generalized Laplacians. It would be interesting to find the necessary additional assumptions that lead to the Laplacian image model.

Concerning the non-negative factorizations we have given a number of examples on how to extend the general framework to model prior knowledge in the form of additional constraints or slight changes to the objective function. This could be a fruitful direction for further research. For instance, one can ask how to model video in this framework, or whether latent semantic analysis can be improved by using objective functions that treat spurious and missing features asymmetrically.

From the optimization point of view we found the global deterministic optimization literature to be a rich field to mine for ideas. Of course, in most cases it will be inefficient to directly use a global deterministic method for the large-scale problems encountered in computer vision or machine learning. Still, approaches from global deterministic optimization can provide starting points to derive new optimization algorithms from. Most likely, one will introduce approximations or shortcuts at one point or another.

The challenge then is to gain a maximum in speed while paying a minimum in terms of undesirable properties of the resulting large-scale algorithms.

Finally, mathematical research is needed to unify the two dominant optimization frameworks used in computer vision: This work is quite characteristic for today's computer vision research in that it uses continuous, PDE-based methods on the one hand and discrete mathematical-programming based methods on the other quite independently. It is not yet clear how to translate systematically results from one framework to the other and what, precisely, the benefits and drawbacks of each approach are. We know from experience that the continuous as well as the discrete approach can yield impressive results. Bridging the two worlds is an exiting research agenda for the future.

Bibliography

[AAR04] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 26(11):1475–1490, nov 2004.

[AD52] T. W. Anderson and D. A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *Annals of Mathematical Stochastics*, 23:193–212, 1952.

[ÁGM99] Luis Álvarez, Yann Gousseau, and Jean-Michel Morel. The size of objects in natural images. *Advances in Imaging and Electron Physics*, 111:167–242, 1999.

[AK00] Gilles Aubert and Pierre Kornprobst. *Mathematical Problems in Image Processing*. Springer, New York, 2000.

[AS95] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. of Comp. Phys.*, 118:269–277, 1995.

[BCGM98] Serge Belongie, Chad Carson, Hayit Greenspan, and Jitendra Malik. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proc. of ICCV*, 1998.

[Ber99] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1999.

[Bes74] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *J. of the Royal Statistical Society*, 36(2):192–225, 1974.

[Bes86] Julian Besag. On the statistical analysis of dirty pictures. *J. of the Royal Statistical Society*, 48(3):259–302, 1986.

[BKS06] Martin Bergtholdt, Jörg Kappes, and Christoph Schnörr. Graphical knowledge representation for human detection. In *IEEE Intl. Workshop on the Representation and Use of Prior Knowledge in Vision*, Graz, Austria, May 2006.

[BLP95] M.C. Burl, T.K. Leung, and P. Perona. Face localization via shape statistics. In *Intl. Workshop on Autom. Face and Gesture Recogn.*, Zurich, Switz., 1995.

[BMS02] Marian Stewart Bartlett, Javier R. Movellan, and Terrence J. Sejnowski. Face recognition by independent components analysis. *IEEE Trans. on Neural Networks*, 13(6):1450–1464, 2002.

[BP04] Ioan Buciu and Ioannis Pitas. Application of non-negative and local non negative matrix factorization to facial expression recognition. In *Proc. of ICPR*, pages 288–291, 2004.

[Bro66] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, New York, 1966.

[BS95] Anthony J. Bell and Terrence J. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1004–1034, 1995.

[BS96] Anthony J. Bell and Terrence J. Sejnowski. Edges are the "independent components" of natural scenes. In *Adv. in NIPS*, pages 831–837, 1996.

[BS99] Robert W. Buccigrossi and Eero P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Trans. on Image Proc.*, 8(12):1688–1700, Dec. 1999.

[BSB99] W. Backfrieder, M. Samal, and H. Bergmann. Towards estimation of compartment volumes and concentrations in dynamic SPECT using factor analysis and limited number of projections. *Physica Medica*, 15, 1999.

[BSU04] Eran Borenstein, Eitan Sharon, and Shimon Ullman. Combining top-down and bottom-up segmentation. In *Proc. of CVPR*, 2004.

[BU02] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *Proc. of ECCV*, pages 109–122, 2002.

[BU04] Eran Borenstein and Shimon Ullman. Learning to segment. In *Proc. of ECCV*, pages 315–328, 2004.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[BW06] Thomas Brox and Joachim Weickert. Level set segmentation with multiple regions. *IEEE Trans. on Image Proc.*, 2006.

[CBC00] CBCL. CBCL face database #1. MIT Center For Biological and Computational Learning, http://cbcl.mit.edu/software-datasets, 2000.

[CC93] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(11):1131–1147, nov 1993.

[Cha53] D. G. Champernowne. A model of income distribution. *The Economic Journal*, 63:318–351, 1953.

[CJ01] Chakra Chennubholta and Allan Jepson. Sparse PCA extracting multi-scale structure from data. In *Proc. of ICCV*, pages 641–647, 2001.

[CKS97] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *Int. J. of Comp. Vision*, 22(1):61–79, 1997.

[CSS06] Daniel Cremers, Nir Sochen, and Christoph Schnörr. A multiphase dynamic labeling model for variational recognition-driven image segmentation. *Int. J. of Comp. Vision*, 66(1):67–81, 2006.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Interscience, New York, 1991.

[CV01] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Trans. on Image Proc.*, 10(2):266–277, February 2001.

[Dau85] J. Daugman. Uncertainty relation for resolution in space, spacial frequency, and orientation. *J. of the Optical Sco. of America*, 2(7), 1985.

[DDL+90] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[dGJL04] Alexandre d'Aspremont, Laurent El Ghaoui, Michael I. Jordan, and Gert R.G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. In *Adv. in NIPS*, 2004.

[DHS05] Chris Ding, Xiaofeng He, and Horst D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SIAM Int'l Conf. on Data Mining (SDM)*, apr 2005.

[DS04a] David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Adv. in NIPS*, volume 17, 2004.

[DS04b] Gyuri Dorkó and Cordelia Schmid. Object class recognition using discriminative local features. Submitted to IEEE PAMI, 2004.

[DZ01] M.C. Delfour and J.P. Zolésio. *Shapes and Geometries: Analysis, Differential Calculus, and Optimization.* SIAM, 2001.

[EL99] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proc. of ICCV*, pages 1033–1038, Corfu, Greece, September 1999.

[FA91] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(9):891–906, September 1991.

[FBCM04] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nyström method. *IEEE Trans. Patt. Anal. Mach. Intell.*, 26(3):214–225, February 2004.

[FE87] W. Förstner and E.Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Intercommission Conf. on Fast Processing of Photogrammetric Data*, pages 281–305, 1987.

[Fel66] William Feller. *An Introduction to Probability Theory and Its Applications*, volume II. John Wiley & Sons, Inc., 2 edition, 1966.

[FFFP04] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[FJ01] Brendan J. Frey and Nebojsa Jojic. Fast, large-scale transformation-invariant clustering. In *Adv. in NIPS*, 2001.

[FJ03] Brendan J. Frey and Nebojsa Jojic. Transformation-invariant clustering using the EM algorithm. *IEEE Trans. Patt. Anal. Mach. Intell.*, 45(1):1–17, January 2003.

[FPZ03] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. of CVPR*, 2003.

[Fuk90] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, NY, second edition, 1990.

[FV93] C. A. Floudas and V. Visweswaran. A primal-relaxed dual global optimization approach. *J. of Optim. Theory and Applic.*, 78(2), 1993.

[FV95] C. A. Floudas and V. Visweswaran. Quadratic optimization. In R. Horst and M Pardalos, editors, *Handbook of global optimization*, pages 217–269. Kluwer, Dordrecht-Boston-London, 1995.

[FWZ05] Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *Int. J. of Comp. Vision*, 63(2):141–151, 2005.

[Gab46] D. Gabor. Theory of communication. *J. of the Inst. of Electr. Eng. (IEE)*, 93(26):429–457, 1946.

[Geu03] J. M. Geusebroek. A scale-space analysis of multiplicative texture processes. In M. Chantler, editor, *Proc. 3rd Int. Workshop on Texture Anal. and Synthesis (Texture 2003)*, pages 37–40. Heriot-Watt University, 2003.

[GF63] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice-Hall, 1963.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(6):721–741, 1984.

[GH96] Zoubin Ghahramani and Geoffrey E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Dpt. of Comp. Sci, Univ. of Toronto, CA, 21 1996.

[GK54] B. V. Gnedenko and A. N. Kolmogorov. *Limit Distributions for Sums of Independent Random Variables*. Addison-Wesley, 1954.

[GK99] Donald Geman and Alexey Koloydenko. Invariant statistics and coding of natural microimages. In *Proc. IEEE Worksh. on Stat. and Compu. Theories of Vision*, 1999.

[GS03] J. M. Geusebroek and A. W. M. Smeulders. Fragmentation in the vision of scenes. In *Proc. of ICCV*, volume 1, pages 130–135, 2003.

[GSV02] David Guillamet, Bernt Schiele, and Jordi Vitrià. Analyzing non-negative matrix factorization for image classification. In *Proc. of ICPR*, 2002.

[HA85] L. Hubert and P. Arabie. Comparing partitions. *J. of Classification*, 2(1):193–218, 1985.

[HB95] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proc. of SIGGRAPH*, pages 229–238, 1995.

[HGT00] Geoffrey E. Hinton, Zoubin Ghahramani, and Yee Whye Teh. Learning to parse images. In *Adv. in NIPS*, pages 463–469, 2000.

[HH02] Patrik O. Hoyer and Aapo Hyvärinen. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, 42(12):1593–1605, 2002.

[HM99] Jinggang Huang and David Mumford. Statistics of natural images and models. In *Proc. of ICCV*, volume 1, pages 541–547, 1999.

[Hof99] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 289–296, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[Hoy02] Patrik O. Hoyer. Non-negative sparse coding. *Proc. of the IEEE Works. on Neur. Netw. for Sig. Proc.*, pages 557–565, 2002.

[Hoy04] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. of Mach. Learning Res.*, 5:1457–1469, 2004.

[HP95] Reiner Horst and Panos M. Pardalos, editors. *Handbook of Global Optimization*. Kluwer Academic Publisher, 1995.

[HPS05] Tamir Hazan, Simon Polak, and Amnon Shashua. Sparse image coding using a 3D non-negative tensor factorization. In *Proc. of ICCV*, 2005.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[HS03] Matthias Heiler and Christoph Schnörr. Natural image statistics for natural image segmentation. In *Proc. of ICCV*, 2003.

[HS05a] Matthias Heiler and Christoph Schnörr. Learning non-negative sparse image codes by convex programming. In *Proc. of ICCV*, 2005.

[HS05b] Matthias Heiler and Christoph Schnörr. Natural image statistics for natural image segmentation. *Int. J. of Comp. Vision*, 63(1):5–19, 2005.

[HS05c] Matthias Heiler and Christoph Schnörr. Reverse-convex programming for sparse image codes. In *Proc. of EMMCVPR*, volume 3757 of *LNCS*, pages 600–616. Springer, 2005.

[HS06a] Matthias Heiler and Christoph Schnörr. Controlling sparseness in non-negative tensor factorization. In *Proc. of ECCV*, 2006.

[HS06b] Matthias Heiler and Christoph Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *J. of Mach. Learning Res.*, (7):1385–1407, July 2006.

[HT96] Reiner Horst and Hoang Tuy. *Global Optimization*. Springer, Berlin, 1996.

[Hua00] Jinggang Huang. *Statistics of Natural Images and Models*. PhD thesis, Division of Applied Mathematics, Brown University, Rode Island, 2000.

[HW02] Ralf Herbrich and Robert C. Williamson. Algorithmic luckiness. *J. of Mach. Learning Res.*, 3:175–212, 2002.

[HY01] Mark H. Hansen and Bin Yu. Model selection and the principle of minimum description length. *J. of the Americ. Stat. Assoc.*, 96(454):746–774, 2001.

[Hyv99] Aapo Hyvärinen. Survey on independent components analysis. *Neural Computation Surveys*, 2:94–128, 1999.

[Isi25] Ernst Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925.

[Jay57] E. T. Jaynes. Information theory and statistical mechanics. *The Physical Review*, 106(4):620–630, may 1957.

[JBBA03] Stéphanie Jehan-Besson, Michel Barlaud, and Gilles Aubert. DREAM^{2}S: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. *Int. J. of Comp. Vision*, 53(1):45–70, 2003.

[JCF95] Rajan L. Joshi, Valerie J. Crump, and Thomas R. Fischer. Image subband coding using arithmetic coded trellis coded quantization. *IEEE Trans. on Circuits and Systems for Video Technology*, 5(6):515–523, December 1995.

[JT05] Fr'ed'eric Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *Proc. of ICCV*, 2005.

[Jul62] Bela Julesz. Visual pattern discrimination. *IRE Trans. on Information Theory*, 8(2):84–92, 1962.

[Kai58] Henry F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23:182–200, 1958.

[KKO$^+$95] Satyanad Kichenassamy, Arun Kumar, Peter J. Olver, Allen Tannenbaum, and Anthony J. Yezzi. Gradient flows and geometric active contour models. In *Proc. of ICCV*, pages 810–815, 1995.

[Kre52] E. R. Kretzmer. Statistics of television signals. *The Bell System Technical Journal, BSTJ*, pages 751–763, July 1952.

[Kru77] J. B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18:95–138, 1977.

[KS04] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. of CVPR*, volume 2, pages 506–513, 2004.

[KSSC03] Jens Keuchel, Christoph Schnörr, Christian Schellewald, and Daniel Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. Patt. Anal. Mach. Intell.*, 25(11):1364–1379, November 2003.

[KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *Int. J. of Comp. Vision*, 1(4):312–331, 1988.

[Lec89] Yvan G. Leclerc. Constructing simple stable descriptions for image partitioning. *Int. J. of Comp. Vision*, 3(1):73–102, 1989.

[LHZC01] Stan Z. Li, Xin Wen Hou, Hong Jiang Zhang, and Qian Sheng Cheng. Learning spatially localized, parts-based representation. In *Proc. of CVPR*, 2001.

[LL05] John Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *J. of Mach. Learning Res.*, 6:129–163, 2005.

[LMH01] Ann B. Lee, David Mumford, and Jinggang Huang. Occlusion models for natural images: A statistical study of a scale-invarian dead leaves model. *Int. J. of Comp. Vision*, 41(1/2):35–59, 2001.

[Low99] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of ICCV*, pages 1150–1158, 1999.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Comp. Vision*, 60(2):91–100, 2004.

[LPM03] Ann B. Lee, Kim S. Pedersen, and David Mumford. The nonlinear statistics of high-contrast patches in natural images. *Int. J. of Comp. Vision*, 54(1–3):83–103, 2003.

[LR97] Scott M. LoPresto and Kannan Ramchandran. Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework. In *Proc. of the Data Compr. Conf. (DCC)*, pages 221–230, 1997.

[LS97] Moshe Levy and Sorin Solomon. Power laws are logarithmic boltzmann laws. *Int. J. of Modern Physics C*, 7(4), 1997.

[LS99] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(21):788–791, October 1999.

[LS00] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Adv. in NIPS*, 2000.

[LS03] Bastian Leibe and Bernt Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference (BMVC'03)*, pages 759–768, Norwich, UK, 2003.

[Lue69] David G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, 605 Third Avenue, New York, NY, 1969.

[LVBL98] Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[LW86] N. Littlestone and Manfred Warmuth. Relating data compression, learnability, and the Vapnik-Chervonenkis dimension. Technical report, Univ. of Calif. Santa Cruz, 1986.

[LW03] Xiuwen Liu and DeLiang Wang. Texture classification using spectral histograms. *IEEE Transactions on Image Processing*, 12(6):661–670, 2003.

[Mac96] David J.C MacKay. Maximum likelihood and covariant algorithms for independent component analysis. University of Cambridge, Cavendish Lab, 1996. http://www.inference.phy.cam.ac.uk/mackay/BayesICA.html, 1996.

[Mal98] Stephane G. Mallat. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(7):674–693, 1998.

[Mar80] S. Marčelja. Mathematical description of the responses of simple cortical cells. *J. of the Optical Sco. of America*, 70(11):1297–1300, 1980.

[MBLS01] Jitendra Malik, Serge Belongie, Thomas K. Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *Int. J. of Comp. Vision*, 43(1):7–27, 2001.

[MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. of ICCV*, volume 2, pages 416–423, July 2001.

[MG01] David Mumford and Basilis Gidas. Stochastic models for generic images. *Quarterly Appl. Math.*, 59:85–111, 2001.

[Min86] M. Minoux. *Mathematical Programming – Theory and Algorithms*. John Wiley and Sons, 1986.

[Mit03] H.D. Mittelmann. An independent benchmarking of SDP and SOCP solvers. *Math. Programming, Series B*, 95(2):407–430, 2003.

[Mos05] MOSEK ApS, Denmark. *The MOSEK optimization tools version 3.2 (Revision 8) User's manual and reference*, 2005.

[MS89] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.

[MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant point detectors. *Int. J. of Comp. Vision*, 60(1):63–86, 2004.

[NJT06] Eric Nowak, Fr'ed'eric Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *Proc. of ECCV*, volume IV of *Springer LNCS 3964*, pages 490–503, 2006.

[NKS05] Antal Nagy, Attila Kuba, and Martin Samal. Reconstruction of factor structures using discrete tomography method. *Electronic Notes in Discrete Mathematics*, 20:519–534, 2005.

[NLM99] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61–67, 1999.*, 1999.

[Nol05] John P. Nolan. *Stable Distributions: Models for Heavy Tailed Data*. draft version, 2005.

[OF96] Bruno A. Olshausen and David J. Field. Natural image statistics and efficient coding. *Network: Computation in Neural Systems*, 7:333–339, May 1996.

[OF97] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, December 1997.

[Ols96] Bruno A. Olshausen. Learning linear, sparse, factorial codes. Technical Report AI Memo No. 1580, MIT AI Lab, 1996.

[OS88] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formuation. *J. of Comp. Physics*, 79:12–49, 1988.

[Paa97] Pentti Paatero. Least squares formulation of robust non-negative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 37, 1997.

[PD02] Nikos Paragios and Rachid Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. of Comp. Vision*, 46(3):223–247, 2002.

[PGM+95] Rosalind Picard, Chris Graczyk, Steve Mann, Josh Wachman, Len Picard, and Lee Campbell. VisTex vision texture database. MIT Media Lab, 1995.

[PHB99] Jan Puzicha, Thomas Hofmann, and Joachim M. Buhmann. Histogram clustering for unsupervised segmentation and image retrieval. *Pattern Recog. Letters*, 20:899–909, 1999.

[PL02] Kim Steenstrup Pedersen and Ann B. Lee. Toward a full probability model of edges in natural images. In *Proc. of ECCV*, pages 328–342, London, UK, 2002. Springer-Verlag.

[PLL98] R. Paget, I. D. Longstaff, and B. Lovell. Texture classification using non-parametric markov random fields. In *13th Intl. Conf. on Digit. Sign. Proc.*, 1998.

[PRTB99] Jan Puzicha, Yossi Rubner, Carlo Tomasi, and Joachim M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *Proc. of ICCV*, volume 2, pages 1165–1172, 1999.

[PS00] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. of Comp. Vision*, 40(1):49–71, 2000.

[PSS00] Lucas Parra, Clay Spence, and Paul Sajda. Higher-order statistical properties arising from the non-stationarity of natural signals. In *Adv. in NIPS*, 2000.

[PT94] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

[Ran71] W. M. Rand. Objective criteria for the evaluation of clustering methods. *J. of the Am. Stat. Assoc.*, 66(336):846–850, 1971.

[RB94] Daniel L. Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. *Physical Review Letters*, 73(6):814–817, August 1994.

[RB05] Stefan Roth and Michael J. Black. Fields of experts: A framework for learning image priors. In *Proc. of CVPR*, volume 2, pages 860–867, 2005.

[RBD03] Mikaël Rousson, Thomas Brox, and Rachid Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. of CVPR*, volume 2, pages 699–704, 2003.

[RG83] Randall C. Reininger and Jerry D. Gibson. Distributions of the two-dimensional DCT coefficients for images. *IEEE Trans. on Communications*, 31(6):835–839, June 1983.

[Ris78] Jorma Rissanen. Modelling by the shortest data description. *Automatica*, 14:465–471, 1978.

[Roc72] R. Tyrrell Rockafellar. *Convex Analysis*. Princton University Press, 2 edition, 1972.

[Row97] Sam Roweis. An EM algorithm for PCA and SPCA. In *Adv. in NIPS*, pages 626–632, 1997.

[RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth movers distance as a metric for image retrieval. *Int. J. of Comp. Vision*, 40(2):99–121, 2000.

[RW98] R.T. Rockafellar and R.J-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der math. Wissenschaften*. Springer, 1998.

[SA96] E P Simoncelli and E H Adelson. Noise removal via Bayesian wavelet coring. In *Third Int'l Conf on Image Proc*, pages 379–382, Lausanne, 1996. IEEE Sig. Proc. Soc.

[Sap01] Guillermo Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[SB00] Nicholas D. Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decompositions of N-way arrays. *J. of Chemometrics*, 14:229–239, 2000.

[SB03] Paris Smaragdis and Judith C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Appl. of Sign. Proc. to Audio and Acoustics*, pages 177–180, 2003.

[SBPP06] Farial Shahnaz, Michael W. Berry, V. Paul Pauca, and Robert J. Plemmons. Document clustering using nonnegative matrix factorization. *Journal on Information Processing & Management*, 42(2):373–383, mar 2006.

[SC97] Didier Sornette and Rama Cont. Convergent multiplicative processes repelled from zero: Power laws and truncated power laws. *J. Phys. I France*, 7:431–444, 1997.

[SC03] Bing Song and Tony Chan. A fast algorithm for level set based optimization. Technical report, Dept. of Mathematics, UCLA, 2003.

[Sch92] Christoph Schnörr. Computation of discontinuous optical flow by domain decomposition and shape optimization. *Int. J. of Comp. Vision*, 8(2):153–165, 1992.

[SF95] E P Simoncelli and W T Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Second Int'l Conf. on Image Proc.*, Washington, DC, 1995.

[SH05] Amnon Shashua and Tamir Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proc. of ICML*, 2005.

[SI89] J. Shen and G. W. Israël. A receptor model using a specific non-negative transformation technique for ambient aerosol. *Atmospheric Environment*, 23(10):2289–2298, 1989.

[Sim80] J. Simon. Differentiation with respect to the domain in boundary value problems. *Numerical Functional Analysis and Optimization*, 2(7):649–687, 1980.

[Sim97] Eero P. Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *31st Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, 1997. IEEE Sig. Proc. Soc.

[SK04] Henry Schneiderman and Takeo Kanade. Object detection using the statistics of parts. *Int. J. of Comp. Vision*, 56(3):151–177, 2004.

[ŠKS+87] M. Šámal, M. Kárný, H. Sürová, E. Maříková, and Z. Dienstbier. Rotation to simple structure in factor analysis of dynamic radionuclide studies. *Phys. Med. Biol.*, 32(3):371–382, 1987.

[SLG02] Anuj Srivastava, Xiuwen Liu, and Ulf Grenander. Universal analytical forms for modeling image probabilities. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(9):1200–1214, September 2002.

[SLSZ03] Anuj Srivastava, Ann B. Lee, Eero P. Simoncelli, and Song-Chun Zhu. On advances in statistical modeling of natural images. *Int. J. of Comp. Vision*, 18:17–33, 2003.

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22(8):888–905, 2000.

[SMB00] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *Int. J. of Comp. Vision*, 37(2):151–172, 2000.

[SRE+05] Josef Sivic, Bryan Russell, Alexei Efros, Andrew Zisserman, and William Freeman. Discovering object categories in image collections. In *Proc. of ICCV*, volume 1, pages 370–378, 2005.

[SSV+06] Rick Szeliski, Ramin Zabihand Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Mashall Tappen, and Carsten Rother. A comparative study of energy minimization for markov random fields. In *Proc. of ECCV*, 2006.

[Stu01] Jos F. Sturm. *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones (updated version 1.05).* Department of Econometrics, Tilburg University, Tilburg, The Netherlands, 2001.

[SZ91] Jan Sokolowski and Jean-Paul Zolésio. *Introduction to Shape Optimization.* Springer, New York, 1991.

[TB97] M. Tipping and C. Bishop. Probabilistic principal component analysis. Technical report, Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, September 1997.

[TD79] F. Thomasset and A. Dervieux. A finite element method for the simulation of a Rayleigh-Taylo instability. In *Springer Lecure Notes in Mathematics*, volume 771, pages 145–158, 1979.

[TM01] Ming Tang and Songde Ma. General scheme of region competition based on scale space. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(12):1366–1378, December 2001.

[TTC92] D. J. Tolhurst, Y. Tadmor, and Tang Chao. Amplitude spectra of natural images. *Ophthal. Pysiol. Opt.*, 12:229–232, April 1992.

[Tuy87] Hoang Tuy. Convex programs with an additional reverse convex constraint. *J. of Optim. Theory and Applic.*, 52(3):463–486, March 1987.

[Tve77] A. Tversky. Features of similarity. *Psychological Review*, 84:827–352, 1977.

[TZ02] Zhuowen Tu and Song Chun Zhu. Image segmentation by data-driven Markov Chain Monte Carlo. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(5):657–673, May 2002.

[UVNS02] Shimon Ullman, Michel Vidal-Naquet, and Erez Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, July 2002.

[VC02] Luminita A. Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. of Comp. Vision*, 50(3):271–293, 2002.

[vdSvH96] A. van der Schaaf and J. H. van Hateren. Modelling the power spectra of natural images: Statistics and information. *Vision Research*, 36(17):2759–2770, 1996.

[vHvdS98] J. H. van Hateren and A. van der Schaaf. Independent component filtering of natural images compared with simple cells in primary visual cortex. In *Proc. of the Royal Soc. London*, volume B 265, pages 359–366, 1998.

[VZ03] Manik Varma and Andrew Zisserman. Texture classification: Are filter banks necessary. In *Proc. of ICCV*, volume 2, pages 691–698, 2003.

[WB68] C. S. Wallace and D. M. Boulton. An information measure for classification. *Comp. J.*, 11(2):185–194, August 1968.

[WCH03] Yiming Wu, Kap Luk Chan, and Yong Huang. Image texture classification based on finite gaussian mixture models. In Mike Chantler, editor, *3rd Int. Workshop on Text. Anal. and Synth., ICCV*, pages 107–112, 2003.

[WCM05] John M. Winn, Antonio Criminisi, and Thomas P. Minka. Object categorization by learned universal visual dictionary. In *Proc. of ICCV*, pages 1800–1807, 2005.

[Wei98] J. Weickert. *Anisotropic diffusion in image processing.* Teubner, Stuttgart, 1998.

[Wib76] T. Wiberg. Computation of principal components when data are missing. In *Proc. Second Symp. Comp. Statistics*, pages 229–236, 1976.

[WJHT04] Yuan Wang, Yunde Jia, Changbo Hu, and Matthew Turk. Fisher non-negative matrix factorization for learning local features. In *Proc. Asian Conf. on Comp. Vision*, 2004.

[WJHT05] Yuan Wang, Yunde Jia, Changbo Hu, and Matthew Turk. Non-negative matrix factorization framework for face recognition. *Intl. J. of Patt. Recogn. and AI*, 19(4):495–511, 2005.

[Wol72] P. Wolfe. On the convergence of gradient methods under constraints. *IBM Journal of Research and Development*, 1972.

[Wri97] Stephen J. Wright. *Primal-dual interior-point methods*. SIAM, Society for Applied Mathematics, 3600 University City Science Center, Philadelphia, PA 19104-2688, 1997.

[WS00] Martin J. Wainwright and Eero P. Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In S. A. Solla, T. K. Lee, and K.-R. Müller, editors, *Adv. in NIPS*, pages 855–861, 2000.

[WSW01] Martin J. Wainwright, Eero P. Simoncelli, and Alan S. Willsky. Random cascades on wavelet trees and their use in analyzing and modeling natural images. *Applied and Computational Harmonic Analysis*, 11(1):89–123, July 2001.

[WW01] Max Welling and Markus Weber. Positive tensor factorization. *Pattern Recog. Letters*, 22(12):1255–1261, 2001.

[WWP00] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. of ECCV*, pages 18–32, 2000.

[WZL00] Ying Nian Wu, Song Chun Zhu, and Xiuwen Liu. Equivalence of Julesz ensembles and FRAME models. *Int. J. of Comp. Vision*, 38(3):247–265, 2000.

[XLG03] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proc. of the 26th Ann. Intl. ACM SIGIR Conf. on Res. and Developm. in Info. Retrieval*, pages 267–273. ACM Press, 2003.

[YFW00] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In *Adv. in NIPS*, pages 689–696, 2000.

[ZG01] Alexey Zalesny and Luc Van Gool. A compact model for viewpoint dependent texture synthesis. In *SMILE 2000 Workshop, LNCS Vol. 2018*, pages 124–143, 2001.

[ZHT05] Hui Zou, Trevor Hastie, and Rob Tibshirani. Sparse principal component analysis. *J. of Computational and Graphical Statistics*, 2005.

[ZLW00] Song Chun Zhu, Xiuwen Liu, and Ying Nian Wu. Exploring texture ensembles by efficient Markov Chain Monte Carlo-toward a 'trichromacy' theory of texture. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22(6):554–569, 2000.

[ZS05] Ron Zass and Amnon Shashua. A unifying approach to hard and probabilistic clustering. In *Proc. of ICCV*, volume 1, pages 294–301, oct 2005.

[ZWM97] Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8), November 1997.

[ZWM98] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modelung. *Int. J. of Comp. Vision*, 27(2):107–126, 1998.

[ZY96] Song Chun Zhu and Alan L. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(9):884–900, 1996.