

Tilburg University

Feature extraction from visual data

van der Maaten, L.J.P.

Publication date:
2009

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

van der Maaten, L. J. P. (2009). *Feature extraction from visual data*. TICC Dissertation Series 7.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Feature Extraction from Visual Data

Feature Extraction from Visual Data

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Tilburg,
op gezag van de rector magnificus,
prof. dr. Ph. Eijlander,
in het openbaar te verdedigen ten overstaan van een
door het college voor promoties aangewezen commissie
in de aula van de Universiteit
op dinsdag 23 juni 2009 om 14:15 uur

door

Laurentius Johannes Paulus van der Maaten
geboren op 20 april 1984 te Epe

Promotores:

Prof. dr. E.O. Postma
Prof. dr. H.J. van den Herik

Copromotor:

Dr. A.G. Lange

Beoordelingscommissie:

Prof. dr. A.P.J. van den Bosch
Dr. ir. R.P.W. Duin
Prof. dr. T.M. Heskes
Prof. dr. J.J. Koenderink
Prof. dr. E.J. Krahmer
Dr. M. Welling
Prof. dr. A.J. van Zanten



The research reported in this thesis has been funded by the Netherlands Organization for Scientific Research (NWO) in the project Reading Images for Cultural Heritage (RICH), grant number 640.002.401. The RICH project is part of the Continuous Access to Cultural Heritage (CATCH) research program.



SIKS Dissertation Series No. 2009-17

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.



TiCC Dissertation Series no. 06.

ISBN 978-90-8559-543-4.

Copyright © 2009, L.J.P. van der Maaten

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronically, mechanically, photocopying, recording or otherwise, without prior permission of the author.

Preface

Our brain is amazingly good at recognizing objects, faces, or scenes under a large number of variations. A true appreciation of these capabilities emerges when one attempts to develop computer vision systems and encounters the obstacles that our visual system seems to solve in a straightforward manner. The ability of the brain to analyze visual information reliably and fast is the result of a remarkable cooperation of elaborate attentional mechanisms, massive parallel processing, and sophisticated feature extraction mechanisms. Some of these feature extraction mechanisms are rather general, such as the initial processing of oriented contours occurring in the early stages of the visual system, but others are more specific and occur in brain areas, such as those involved in the recognition of faces.

Already in the research for my M.Sc. thesis, I focused on the extraction of features from visual data. Specifically, I investigated the extraction of features in handwriting that discriminate between the writers of the text. Since then, feature extraction has remained the main topic of my research, although there has been a slight shift from vision towards machine learning, as the chapters on dimensionality reduction in this thesis illustrate. The thesis presents my contribution to the solution of two fundamental problems in computer vision, i.e., the dimensionality problem and the variance problem.

The writing of this thesis (and the research presented therein) would have been impossible without the help and dedicated guidance of many people. First and foremost, I am greatly indebted to my supervisors Eric Postma and Jaap van den Herik. I thank Eric for his superb guidance, his unflagging energy and positivism, his great sense of humor, and his interest in a wide range of topics (pretty much everything except soccer and cars). Without Eric, I probably would not have withstood the large number of disappointments that are inevitable parts of the life of a scientific researcher. I thank Jaap for his great enthusiasm and support for my research, and in particular, for teaching me how to write scientific texts that are so clear that they can readily be understood by laymen. Even in chaotic times, Jaap always found time to point out the oddities in my writing in his own very special way.

Throughout the years, I have had the pleasure of working with many inspiring colleagues in Tilburg, Amersfoort, Maastricht, and Toronto. I would like to thank all colleagues for the lessons they taught me at some point. In particular, I would like to thank Guido de Croon, Jahn-Takeshi Saito, Stijn Vanderlooy, and Arnold Binas for the large number of discussions we have had on a wide range of topics. I am grateful to Ben Torben-Nielsen and Steven de Jong for designing the stylesheet of this thesis. Moreover, I would like to thank Joke Hellemons for her support, and Marc Ponsen, Jeroen Janssens, Ildikó Flesh, Guillaume Chaslot, Maarten Schadd, Niek Bergboer,

Igor Berezhnoy, Sander Bakkes, Joyca Lacroix, Nyree Lemmens, Erik Drenth, Rich Zemel, Iain Murray, Mark Palatucci, Graham Taylor, Vinod Nair, Ruslan Salakhutdinov, Andriy Mnih, and Ilya Sutskever for their participation in our discussions on a range of scientific topics.

A special word of thanks goes to Geoffrey Hinton for his hospitality, his enthusiasm and humor, and for being a great source of inspiration. Probably the most important lesson that Geoff taught me is never to give up on an idea (even if it takes 17 years to get it right), unless you completely understand why the idea is wrong.

My RICH team members also deserve a special word of gratitude. In particular, I would like to thank Guus Lange for his patience and for his support for my research, even if I was not digging into the past. In the end, our joint work forms a decent contribution to archaeology. I thank Paul Boon for his cooperation, his perseverance, and for successfully implementing some of our ideas into the cultural heritage (which, as we experienced, is actually much more challenging than it sounds). I recognize Hans Pajmans for our discussions, and for his sometimes alarmingly strange sense of humor.

Then, I would like to recognize Rene Cappers for generously providing the seeds dataset. I am grateful to the Van Gogh Museum and the Kröller-Muller Museum for providing the dataset of paintings by Van Gogh and his contemporaries. Louis Vuurpijl and Lambert Schomaker are acknowledged for creating the *Firemaker* dataset that formed the basis for the characters dataset. I am indebted to the Netherlands Organization for Scientific Research (NWO) and the Dutch State Service for Archaeology (RACM) for their support of my work.

Last and therefore most important, I would like to close by thanking my parents and Danique. I thank my parents for having supported me throughout my studies. Without their support, I would not have been where I am today. I thank Danique for her love and support. I admire her patience at the frequent times that I was distracted by my work. Fortunately, we understand each other better than anyone else.

Tilburg, May 2009.

Contents

Preface	vii
Contents	ix
1 Introduction	1
1.1 Feature extraction	3
1.2 Problem statement	4
1.3 Research methodology	5
1.4 Structure of the thesis	6
2 Dimensionality reduction	7
2.1 Dimensionality reduction	9
2.2 Convex techniques for dimensionality reduction	10
2.2.1 Full spectral techniques	10
2.2.2 Sparse spectral techniques	16
2.3 Non-convex techniques for dimensionality reduction	20
2.4 Characterization of the techniques	24
2.4.1 Relations	24
2.4.2 General properties	25
2.4.3 Out-of-sample extension	27
2.5 Experiments	28
2.5.1 Experimental setup	28
2.5.2 Experiments on artificial datasets	32
2.5.3 Experiments on natural datasets	32
2.6 Discussion	33
2.6.1 Full spectral techniques	34
2.6.2 Sparse spectral techniques	35
2.6.3 Non-convex techniques	36
2.6.4 Main weaknesses	37
2.7 Chapter conclusions	38

3	t-Distributed Stochastic Neighbor Embedding	39
3.1	Stochastic Neighbor Embedding	40
3.2	t-Distributed Stochastic Neighbor Embedding	43
3.2.1	Symmetric SNE	43
3.2.2	The crowding problem	44
3.2.3	Mismatched tails compensate for mismatched dimensionalities	45
3.2.4	Optimization methods for t-SNE	46
3.3	Experiments	48
3.3.1	Datasets	48
3.3.2	Experimental setup	48
3.3.3	Results	49
3.4	Applying t-SNE to large datasets	52
3.5	Discussion	54
3.5.1	Comparison with related techniques	54
3.5.2	Weaknesses	57
3.6	Chapter conclusions	58
4	Extensions of t-Distributed Stochastic Neighbor Embedding	59
4.1	Parametric t-SNE	60
4.1.1	Experiments	63
4.1.2	Discussion	66
4.2	Multiple-maps t-SNE	68
4.2.1	Formulating t-SNE using multiple maps	70
4.2.2	Experiments	72
4.2.3	Discussion	77
4.3	Chapter conclusions	81
5	Texture features	83
5.1	Graylevel co-occurrence features	85
5.2	Markov Random Fields	85
5.3	Filter-based features	87
5.3.1	Gabor filter bank	88
5.3.2	Maximum Response filter bank	89
5.3.3	Schmid filter bank	90
5.3.4	Steerable pyramids	91
5.3.5	Complex wavelet transform	91
5.4	Texton-based features	94
5.5	Chapter conclusions	95
6	Texton-based texture features	97
6.1	Feature construction	98
6.1.1	Codebook construction	99
6.1.2	Texton frequency histogram	99
6.2	Texton representations	99

6.2.1	Filter-based textons	100
6.2.2	Image-based textons	100
6.3	Experiments with filter-based textons	101
6.3.1	Experimental setup	101
6.3.2	Results	102
6.4	Invariant texton representations	103
6.4.1	Spin images	103
6.4.2	Polar Fourier features	104
6.4.3	Affine-invariant textons	105
6.5	Experiments with invariant textons	106
6.5.1	Experimental setup	106
6.5.2	Results	108
6.6	Discussion	113
6.7	Chapter conclusions	114
7	Applications to the cultural heritage	117
7.1	Painting analysis	118
7.1.1	Experimental setup	119
7.1.2	Results	120
7.1.3	Discussion	121
7.2	Seed analysis	122
7.2.1	Experimental setup	124
7.2.2	Results	124
7.2.3	Discussion	124
7.3	General discussion	126
7.4	Chapter conclusions	128
8	Conclusion	129
8.1	Answers to the research questions	130
8.2	Answer to the problem statement	131
8.3	Future research	131
	References	135
A	Image features	157
A.1	Local image features	157
A.1.1	SIFT features	157
A.1.2	RIFT features	157
A.2	Shape features	158
A.2.1	Zernike moments	159
A.2.2	Angular radial transform features	160
A.2.3	Curvature scale-space features	161
A.2.4	Shape contexts	162
A.3	Edge-based statistical features	163
A.3.1	Edge-hinge features	164

A.3.2 Edge angle-distance features 164

B Derivation of the t-SNE gradient 167

C Analytical solution to random walk probabilities 169

D Restricted Boltzmann Machines 171

E Derivation of the multiple maps t-SNE gradient 173

F Applications of edge-based statistical features 177

 F.1 Writer identification 177

 F.2 Coin classification 178

List of Figures 181

List of Tables 183

List of Abbreviations 185

Summary 187

Samenvatting 189

Curriculum Vitae 193

Publications 195

SIKS Dissertation Series 197

TiCC Ph.D. Series 205

Index 207

1 Introduction

Contents

Worldwide, the number of cheap digital image capturing devices is growing at a steady pace. This leads to the availability of vast amounts of visual data. The large collections of visual data pave the way for the development of new computer vision systems. The main problems that need to be addressed in the development of such systems are the dimensionality problem and the variance problem of image-space representations. The dimensionality problem is the result of the large number of pixels in an image. The variance problem is the result of the drastic changes of pixel values under small variations in the imaging conditions. Both problems may be addressed with success by extracting features from the visual data that are non-redundant and invariant under the variations in images. In the thesis, we study the extraction of features from visual data by investigating two research questions, which focus on the dimensionality problem and the variance problem, respectively. This chapter introduces the problem statement of the thesis, as well the two research questions.

Outline

In Section 1.1, we introduce feature extraction, which is an essential part of many artificially intelligent systems that process visual inputs. Section 1.2 presents the problem statement of the thesis, as well as the research questions that the thesis aims to answer. Section 1.3 presents the research methodology that is employed in order to answer the research questions. The chapter concludes by a description of the structure of the thesis in Section 1.4.

In the decades to come, the number of successful applications of artificially intelligent vision systems will rapidly increase. The emergence of cheap and portable digital image capturing devices facilitates the development of a wide range of new systems that assist humans in their everyday tasks. We mention five examples in different domains of applications.

- Archaeologists performing an excavation can be assisted by a system¹ that recognizes the age and origin of objects retrieved from the soil [van der Maaten *et al.*, 2008].
- Exchange offices and other financial institutes may benefit from a system that automatically sorts coins and banknotes based on their digital reproductions, extending the sorting capabilities of traditional money-sorting systems to a wide range of currencies [Huber *et al.*, 2005].
- Public health can be improved by the development of vision-based food quality assessment systems [Brosnan and Sun, 2004].
- Forensic research may be more effective by the development of new biometric applications for, e.g., writer recognition [Schomaker *et al.*, 2007].
- Public places such as airports may be made more secure by face recognition systems that automatically recognize wanted criminals in images from security cameras [Wolf *et al.*, 2002].

These systems rely on state-of-the-art computer vision and machine learning techniques which evaluate the input images that are captured by a camera. Machine learning comprises a collection of powerful approaches that allows for learning, e.g., underlying distributions, decision boundaries, or policies from sets of data [Bishop, 2006]. Vision systems usually train machine learning techniques on a large dataset of examples. For instance, OCR systems are trained on large datasets of character images that are labeled according to the depicted character [Mori *et al.*, 1999], and face detection systems are trained on large datasets of images in which the locations of all faces are marked [Viola and Jones, 2001]. In the training of the machine learning techniques, the input images can be represented in various ways.

Typically, grayscale and color images are represented by two-dimensional and three-dimensional matrices, respectively. Concatenating all elements of the matrix into a long vector gives rise to an image-space representation. The image is thus represented by a point (or vector) in a high-dimensional image space. The design of most computer vision systems is hampered by two main problems of image-space representations: (1) the dimensionality problem and (2) the variance problem. In what follows, we discuss both problems in more detail.

1) Dimensionality problem: The dimensionality problem follows from the exponential growth of the volume of the representation space with dimensionality. The large number of pixels in an image makes image-space representations very high-dimensional. As a result, image-space representations suffer from the curse of dimensionality and other undesired properties of high-dimensional spaces [Jimenez and Landgrebe, 1997].

¹We recognize that classification of archaeological objects may require contextual data next to visual data.

2) *Variance problem*: The variance problem is the result of the drastic changes in pixel values that occur under the influence of changes in lighting, contrast, camera settings, viewpoint, or under the presence of translations, three-dimensional rotations, and occlusions of depicted objects. An image-space representation is not a diagnostic representation for the class of the depicted object, as two completely different image-space representations may depict the same object under a small variation in imaging conditions.

The dimensionality problem and variance problem may be resolved by extracting *features* from the input images². Features are statistics that are computed from the input images. In order to resolve both problems successfully, the features should meet the following three requirements: (1) the features should be non-redundant to resolve the dimensionality problem as good as possible, (2) the features should be invariant to natural variations in the input images to resolve the variance problem, and (3) the features should be diagnostic for the object class under consideration. The extraction of features that meet the three requirements facilitates the successful training of machine learning techniques. The extraction of informative features from input images (or any other kind of data) is called *feature extraction*, and it is the main topic of this thesis.

The development of feature extraction techniques has a long tradition, see, e.g., [Yuille *et al.*, 1992; Reed and du Buf, 1993; Liu and Motoda, 1998; Forsyth and Ponce, 2003]. Our contribution is that we develop a new feature extraction technique that attempts to address the dimensionality problem by performing dimensionality reduction (see Chapter 2, 3, and 4), and new invariant texture features that aim to address the variance problem (see Chapter 5 and 6). Moreover, we apply the new feature extraction techniques in the challenging cultural heritage domain (see Chapter 7).

The outline of this chapter is as follows. Section 1.1 identifies and briefly discusses two main types of features: dimensionality reduction features and image features. In Section 1.2, we discuss the problem statement of the thesis and we present our two research questions. In Section 1.3, the methodology employed in addressing the research questions is discussed. Section 1.4 concludes the chapter by a description of the structure of the thesis.

1.1 Feature extraction

As described above, feature extraction is the process of extracting statistics from input images that are preferably (1) non-redundant, (2) invariant under natural image variations, and (3) diagnostic for the class of the depicted object. In this thesis, we distinguish two main types of features: (1) dimensionality reduction features and (2) image features. Dimensionality reduction features aim to address the dimensionality problem of image-space representations. Image features aim to address the variance problem of image-space representations. We briefly introduce both types of features below.

Dimensionality reduction features mitigate the undesired effects of the high dimensionality of image-space representations by exploiting the (non)linear relations between the pixel values in the input images. They do so by exploiting (non)linear relations between individual input

²We should note that some recent studies take a different approach, and construct invariant image classifiers without extracting features by training on massive datasets that are gathered by crawling the Internet [Torralba *et al.*, 2007].

variables (i.e., pixels), without explicitly using the spatial structure of images. As a result, dimensionality reduction features can be applied to virtually any type of data. We study dimensionality reduction features in the first part of this thesis. In particular, we identify the main weaknesses of state-of-the-art dimensionality reduction features and develop a technique for the extraction of new dimensionality reduction features that aims to address some of these weaknesses.

Image features mitigate the variance problem by constructing representations that are similar for images that depict the same object under different imaging conditions. For instance, image features may construct a representation for the texture of a surface in such a way that the representation is invariant under local affine transformations. In contrast to dimensionality reduction features, image features are explicitly designed to exploit the spatial structure in images.

Image features can be subdivided into local and global features. Local image features (such as SIFT features [Lowe, 2004]) represent small parts of an image, whereas global image features provide a representation for a complete image (or for the complete object depicted in the image). For some features, the assignment is ambiguous, for instance, global image features such as shape contexts (see Appendix A.2.4) may also be considered as local image features. Local image features are usually employed in object detection tasks (such as face detection), because they facilitate the use of matching algorithms that are invariant to occlusions. In contrast, global image features provide more detailed object representations, making them well suited for object classification tasks (such as face recognition). In the second part of this thesis, we investigate global image features. In particular, we develop novel features that provide invariant representations for the texture of an object's surface.

1.2 Problem statement

Above, we outlined the importance of features in addressing the two problems of image-space representations. We aim to develop non-redundant invariant image representations in an attempt to resolve both problems. This leads us to formulate the following problem statement.

How can we mitigate the problems of image-space representations?

To address the problem statement we focus on the development of two types of features: (1) dimensionality reduction features and (2) texture features. We opt for the investigation of dimensionality reduction features, because these features are well suited to address the high dimensionality of image-space representations, and because of the recent popularity of a large number of novel nonlinear dimensionality reduction techniques [Lee and Verleysen, 2007]. Texture features are investigated because they are important image features for which, in contrast to many other image features, the susceptibility to variations has not been the subject of much study (although [Lazebnik *et al.*, 2005; Mellor *et al.*, 2008] are notable exceptions). Moreover, the two selected types of features are well suitable for the two computer vision systems presented in Chapter 7. From the problem statement above, we derive two research questions.

- **Research question 1 (RQ1):** *How can we improve existing dimensionality reduction features?*

- **Research question 2 (RQ2):** *How can existing texture features be adapted to be invariant to variations that occur in uncontrolled environments, such as lighting changes, rotations, and affine transformations?*

The two main contributions of the thesis are (1) the development of a new technique for the extraction of dimensionality reduction features, called t-SNE, and (2) the development of a new affine-invariant texture feature.

1.3 Research methodology

The research methodology followed is based on review of the relevant literature, analysis of the findings from the literature, and development and evaluation of new features. We evaluate features in a quantitative and qualitative manner.

The quantitative evaluation is based on the determination of the generalization performance of classifiers trained with the developed features. By employing the cross-validation procedure [Bishop, 2006] we estimate the generalization performance as a measure of the quality of the features. The use of this validation procedure is commonplace in machine-learning research and provides a reliable estimate of the true generalization error. The generalization performance is defined as the average performance over all folds. The standard deviation of the performance over the folds may offer an indication of the reliability of the estimate of the generalization performance. When comparing the features under consideration, we simply compare the associated generalization performances. We do not employ statistical tests such as ANOVA [Lindman, 1974] to establish whether the difference in performance of different features is statistically significant, because the standard deviation of the generalization performance is typically very small (due to the large number of instances in each fold), which makes the use of statistical tests superfluous. Differences are considered to be significant whenever the average performances are separated by at least two standard deviations. The evaluation of the generalization performance allows us to determine whether the developed features capture information that is diagnostic to the class labels of the images [Cohen, 1995].

The qualitative evaluation is a visualization of the feature-based image representations in two-dimensional maps. It provides some intuition for which information in the image data is captured by the extracted features. Moreover, the visualizations may provide additional evidence for phenomena observed in the quantitative evaluation.

The image data on which we evaluate the features under investigation is selected in such a way that it meets the following three requirements: (1) the image data is labeled in order to facilitate the training and testing of classifiers, (2) the image data contains (some of) the variations that occur in natural images, and (3) the image data is publicly available in order to facilitate comparisons with results obtained in other studies. The only exception to the last requirement is in Chapter 7, where we employ the developed features in two computer vision systems that are trained on image data which is not publicly available.

1.4 Structure of the thesis

The remainder of this thesis consists of two main parts. The first part of the thesis attempts to answer research question RQ1 (in Chapter 2, 3, and 4). The second part of the thesis strives to answer research question RQ2 (in Chapter 5 and 6). The two main parts of the thesis are followed by a chapter in which we apply the developed features in the cultural heritage domain. Below, we briefly discuss the contents of each of the chapters of the thesis.

Chapter 1 presents the problem statement, research questions, and research methodology of the thesis. In Chapter 2, we compare state-of-the-art dimensionality reduction features in a range of classification tasks, and identify the main weaknesses of the underlying techniques. Chapter 3 presents a new dimensionality reduction technique called t-SNE that aims to address some of the weaknesses that were identified in Chapter 2. In Chapter 4, we extend t-SNE to two alternative learning settings, i.e., a learning setting in which a parametric mapping between the data space and the latent space is required and a learning setting in which the latent space is non-metric. In Chapter 5, our focus shifts towards texture features. The chapter presents a literature survey of state-of-the-art texture features, and concludes that so-called texton-based texture features form an interesting alternative to traditional texture features based on filter banks or Markov Random Fields. In Chapter 6, we present new texton-based color-texture features that are invariant to all main variations occurring in images that are captured in uncontrolled environments. Chapter 7 presents applications of the developed features in two applications. Chapter 8 concludes the thesis and presents the answers to the research questions posed in this chapter, as well as to the problem statement. Moreover, the chapter provides five directions of future research.

2 Dimensionality reduction

- Contents** The dimensionality problem of image-space representations may be addressed by extracting dimensionality reduction features from images. However, it is unclear which dimensionality reduction techniques are most appropriate for this task, and what the main limitations of the techniques are. Motivated by this observation, the chapter presents a comparative review of dimensionality reduction techniques. We identify the main weaknesses of current dimensionality reduction techniques in order to (partially) answer research question RQ1. The chapter presents recommendations for the development of future dimensionality reduction techniques, some of which we will implement in a new dimensionality reduction technique in Chapter 3.
- Based on** L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. Dimensionality Reduction: A Comparative Review. Submitted to *Journal of Machine Learning Research*.
- Outline** Section 2.1 gives a formal definition of dimensionality reduction. Section 2.2 describes and discusses nine convex techniques for dimensionality reduction. In Section 2.3, we describe and discuss four non-convex techniques for dimensionality reduction. Section 2.4 evaluates all techniques on theoretical characteristics. In Section 2.5, we present an empirical evaluation of techniques for dimensionality reduction on artificial and natural datasets. Section 2.6 discusses the results of the experiments and identifies weaknesses and points of improvement of the nonlinear techniques for dimensionality reduction. Section 2.7 concludes the chapter.

Dimensionality reduction is the transformation of high-dimensional data into a meaningful representation of reduced dimensionality. Ideally, the reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the data. The intrinsic dimensionality of data is the minimum number of parameters needed to account for the observed properties of the data [Fukunaga, 1990]. Dimensionality reduction is important in many domains, since it mitigates the curse of dimensionality and other undesired properties of high-dimensional spaces that are the result of the exponential growth of volume with dimensionality [Jimenez and Landgrebe, 1997]. As a result, dimensionality reduction facilitates, among others, classification, visualization, and compression of high-dimensional data. Traditionally, dimensionality reduction was performed using linear techniques such as Principal Components Analysis (PCA) [Pearson, 1901; Hotelling, 1933] and factor analysis [Spearman, 1904]. However, these linear techniques cannot adequately handle complex nonlinear data.

Therefore, in the last decade, a large number of nonlinear techniques for dimensionality reduction have been proposed (see for an overview, e.g., [Burges, 2005; Saul *et al.*, 2006; Lee and Verleysen, 2007; Venna, 2007]). In contrast to the traditional linear techniques, the nonlinear techniques have the ability to deal with complex nonlinear data. In particular for real-world data, the nonlinear dimensionality reduction techniques may offer an advantage, because real-world data is likely to be highly nonlinear. Previous studies have shown that nonlinear techniques outperform their linear counterparts on complex artificial tasks (see, e.g., [Roweis and Saul, 2000; Tenenbaum *et al.*, 2000]). For instance, the Swiss roll dataset comprises a set of points that lie on a spiral-like two-dimensional manifold that is embedded within a three-dimensional space. A vast number of nonlinear techniques are perfectly able to find this embedding, whereas linear techniques fail to do so. In contrast to these successes on artificial datasets, successful applications of nonlinear dimensionality reduction techniques on natural datasets are less convincing. Beyond this observation, it is not clear to what extent the performances of the various dimensionality reduction techniques differ on artificial and natural tasks (a comparison is performed by Niskanen and Silvén [2003], but this comparison is very limited in scope with respect to the number of techniques and tasks that are addressed).

Motivated by the lack of a systematic comparison of dimensionality reduction techniques, this chapter presents a comparative study of the most important linear dimensionality reduction technique (PCA), and twelve frontranked nonlinear dimensionality reduction techniques. The aims of the chapter are (1) to investigate to what extent novel nonlinear dimensionality reduction techniques outperform the traditional PCA on real-world datasets and (2) to identify the inherent weaknesses of the twelve nonlinear dimensionality reduction techniques. The investigation is performed by both a theoretical and an empirical evaluation of the dimensionality reduction techniques. The identification is performed by a careful analysis of the empirical results on specifically designed artificial datasets and on a selection of real-world datasets.

Next to PCA, the chapter investigates the following twelve nonlinear techniques: (1) Kernel PCA, (2) Isomap, (3) Maximum Variance Unfolding, (4) diffusion maps, (5) Locally Linear Embedding, (6) Laplacian Eigenmaps, (7) Hessian LLE, (8) Local Tangent Space Analysis, (9) Sammon mapping, (10) multilayer autoencoders, (11) Locally Linear Coordination, and (12) manifold charting. Although our comparative review includes the most important nonlinear techniques for dimensionality reduction, it is not exhaustive. The review does not include self-organizing maps [Kohonen, 1989] and their probabilistic extension GTM [Bishop *et al.*,

1998], because we consider these techniques to be clustering techniques. Techniques for Independent Component Analysis [Bell and Sejnowski, 1995] are not included in our review, because they were mainly designed for blind-source separation. Linear Discriminant Analysis [Fisher, 1936], Generalized Discriminant Analysis [Baudat and Anouar, 2000], and Neighborhood Components Analysis [Goldberger *et al.*, 2005; Salakhutdinov and Hinton, 2007] are not included in the review, because of their supervised nature. Furthermore, our comparative review does not cover a number of techniques that are variants or extensions of the thirteen reviewed dimensionality reduction techniques. These variants include factor analysis [Spearman, 1904], principal curves [Chang and Ghosh, 1998], kernel maps [Suykens, 2007], conformal eigenmaps [Sha and Saul, 2005], Geodesic Nullspace Analysis [Brand, 2004], various variants of multidimensional scaling [Faloutsos and Lin, 1995; Demartines and Hérault, 1997; Agrafiotis, 2003], techniques that (similarly to LLC and manifold charting) globally align a mixture of linear models [Roweis *et al.*, 2001; Verbeek, 2006; Sanguinetti, 2008], and linear variants of LLE [He *et al.*, 2005; Kokiopoulou and Saad, 2007], Laplacian Eigenmaps [He and Niyogi, 2004], and LTSA [Zhang *et al.*, 2007].

The outline of the remainder of this chapter is as follows. In Section 2.1, we give a formal definition of dimensionality reduction and subdivide the thirteen dimensionality reduction techniques into nine convex techniques and four non-convex techniques. Section 2.2 presents and discusses the nine convex dimensionality reduction techniques. Subsequently, Section 2.3 describes and discusses the four non-convex techniques for dimensionality reduction. Section 2.4 lists all techniques by theoretical characteristics. Then, in Section 2.5, we present an empirical comparison of all described techniques for dimensionality reduction on five artificial datasets and five natural datasets. Section 2.6 discusses the results of the experiments; moreover, it identifies weaknesses and points of improvement of the selected nonlinear techniques. Section 2.7 provides our conclusions.

2.1 Dimensionality reduction

The problem of (nonlinear) dimensionality reduction can be defined as follows. Assume we have a dataset represented in a $n \times D$ matrix \mathbf{X} consisting of n datavectors \mathbf{x}_i ($i \in \{1, 2, \dots, n\}$) with dimensionality D . Assume further that this dataset has intrinsic dimensionality d (where $d < D$, and often $d \ll D$). Here, in mathematical terms, intrinsic dimensionality means that the points in dataset \mathbf{X} are lying on or near a manifold with dimensionality d that is embedded in the D -dimensional space. Note that we make no assumptions on the structure of this manifold: the manifold may be non-Riemannian because of discontinuities (i.e., the manifold may consist of a number of disconnected submanifolds). Dimensionality reduction techniques transform dataset \mathbf{X} with dimensionality D into a new dataset \mathbf{Y} with dimensionality d , while retaining the geometry of the data as much as possible. In general, neither the geometry of the data manifold, nor the intrinsic dimensionality d of the dataset \mathbf{X} are known. Therefore, dimensionality reduction is an ill-posed problem that can only be solved by assuming certain properties of the data (such as its intrinsic dimensionality). Throughout the thesis, we denote a high-dimensional datapoint by \mathbf{x}_i , where \mathbf{x}_i is the i th row of the D -dimensional data matrix \mathbf{X} . The low-dimensional counterpart of

\mathbf{x}_i is denoted by \mathbf{y}_i , where \mathbf{y}_i is the i th row of the d -dimensional data matrix \mathbf{Y} . In the remainder of the thesis, we adopt the notation presented above, and we assume the dataset \mathbf{X} is zero-mean.

Figure 2.1 shows a taxonomy of techniques for dimensionality reduction. We subdivide techniques for dimensionality reduction into convex and non-convex techniques. Convex techniques optimize an objective function that does not contain any local optima, whereas non-convex techniques optimize objective functions that do contain local optima. The further subdivisions in the taxonomy are discussed in Section 2.2 (convex techniques) and Section 2.3 (non-convex techniques).

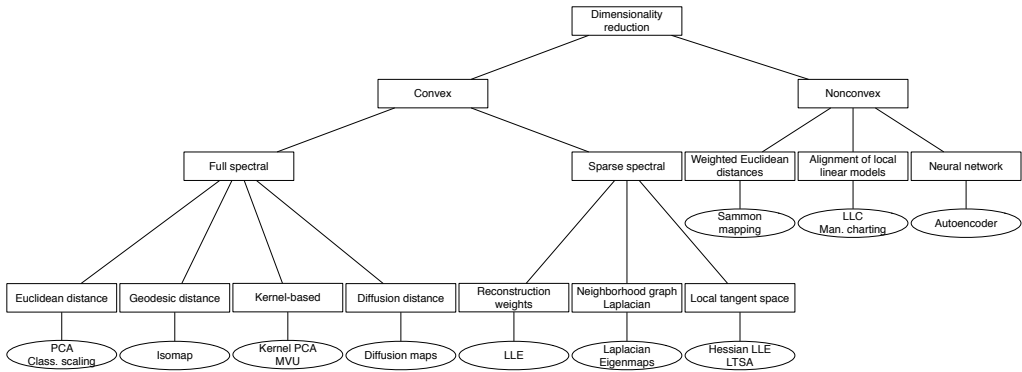


Figure 2.1 Taxonomy of dimensionality reduction techniques.

2.2 Convex techniques for dimensionality reduction

Convex techniques for dimensionality reduction optimize an objective function that does not contain any local optima (i.e., the solution space is convex) [Boyd and Vandenberghe, 2004]. Most of the selected dimensionality reduction techniques fall in the class of convex techniques. In these techniques, the objective function has the form of a (generalized) Rayleigh quotient: the objective function is of the form $\phi(\mathbf{X}) = \frac{\mathbf{X}^T \mathbf{A} \mathbf{X}}{\mathbf{X}^T \mathbf{B} \mathbf{X}}$. It is well known that a function of this form can be optimized by solving a generalized eigenproblem. One technique (Maximum Variance Unfolding) solves an additional semidefinite program using an interior point method. We subdivide convex dimensionality reduction techniques into techniques that perform a spectral analysis of a full matrix (subsection 2.2.1) and those that perform a spectral analysis of a sparse matrix (subsection 2.2.2).

2.2.1 Full spectral techniques

Full spectral techniques for dimensionality reduction perform an eigendecomposition of a full matrix that captures the covariances between dimensions or the pairwise similarities between datapoints (possibly in a feature space that is constructed by means of a kernel function). In this subsection, we discuss five such techniques: (1) PCA / classical scaling, (2) Isomap, (3) Kernel PCA, (4) Maximum Variance Unfolding, and (5) diffusion maps.

PCA / Classical scaling

Principal Components Analysis (PCA) [Pearson, 1901; Hotelling, 1933] is a linear technique for dimensionality reduction, which means that it performs dimensionality reduction by embedding the data into a linear subspace of lower dimensionality. Although there exist various techniques to do so, PCA is by far the most popular (unsupervised) linear technique. Therefore, in our comparison, we only include PCA.

PCA constructs a low-dimensional representation of the data that describes as much of the variance in the data as possible. This is done by finding a linear basis of reduced dimensionality for the data, in which the amount of variance in the data is maximal.

In mathematical terms, PCA attempts to find a linear mapping \mathbf{M} that maximizes $\mathbf{M}^T \text{cov}(\mathbf{X})\mathbf{M}$, where $\text{cov}(\mathbf{X})$ is the sample covariance matrix of the data \mathbf{X} . It can be shown that this linear mapping is formed by the d principal eigenvectors (i.e., principal components) of the sample covariance matrix of the zero-mean data¹. Hence, PCA solves the eigenproblem

$$\text{cov}(\mathbf{X})\mathbf{M} = \lambda\mathbf{M}. \quad (2.1)$$

The eigenproblem is solved for the d principal eigenvalues λ . The low-dimensional data representations \mathbf{y}_i of the datapoints \mathbf{x}_i are computed by mapping them onto the linear basis \mathbf{M} , i.e., $\mathbf{Y} = \mathbf{X}\mathbf{M}$.

PCA is identical to the traditional technique for multidimensional scaling called classical scaling [Torgerson, 1952]. The input into classical scaling is, like the input into most other multidimensional scaling techniques, a pairwise Euclidean distance matrix D of which the entries d_{ij} represent the Euclidean distance between the high-dimensional datapoints \mathbf{x}_i and \mathbf{x}_j . Classical scaling finds the linear mapping that minimizes the cost function

$$\phi(\mathbf{Y}) = \sum_{ij} (d_{ij}^2 - \|\mathbf{y}_i - \mathbf{y}_j\|^2), \quad (2.2)$$

in which $\|\mathbf{y}_i - \mathbf{y}_j\|^2$ is the squared Euclidean distance between the low-dimensional datapoints \mathbf{y}_i and \mathbf{y}_j . It can be shown [Torgerson, 1952; Williams, 2002] that the minimum of this cost function is given by the eigendecomposition of the Gram matrix $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ of the high-dimensional data. The entries of the Gram matrix can be obtained by double-centering the pairwise squared Euclidean distance matrix, i.e., by computing

$$k_{ij} = -\frac{1}{2} \left(d_{ij}^2 - \frac{1}{n} \sum_l d_{il}^2 - \frac{1}{n} \sum_l d_{jl}^2 + \frac{1}{n^2} \sum_{lm} d_{lm}^2 \right). \quad (2.3)$$

The minimum of the cost function in Equation 2.2 can now be obtained by multiplying the principal eigenvectors of the double-centered squared Euclidean distance matrix (i.e., the principal eigenvectors of the Gram matrix) with the square root of their corresponding eigenvalues. The similarity of classical scaling to PCA is the result of a relation between the eigenvectors of the

¹PCA maximizes $\mathbf{M}^T \text{cov}(\mathbf{X})\mathbf{M}$ with respect to \mathbf{M} , under the constraint that the L2-norm of each column \mathbf{m}_j of \mathbf{M} is 1, i.e., that $\|\mathbf{m}_j\|^2 = 1$. This constraint can be enforced by introducing a Lagrange multiplier λ . Hence, an unconstrained maximization of $\mathbf{m}_j^T \text{cov}(\mathbf{X})\mathbf{m}_j + \lambda(1 - \mathbf{m}_j^T \mathbf{m}_j)$ is performed. The stationary points of this quantity are to be found when $\text{cov}(\mathbf{X})\mathbf{m}_j = \lambda\mathbf{m}_j$.

covariance matrix and the Gram matrix of the high-dimensional data: it can be shown that the eigenvectors \mathbf{u}_i and \mathbf{v}_i of the matrices $\mathbf{X}^T \mathbf{X}$ and $\mathbf{X} \mathbf{X}^T$ are related through $\sqrt{\lambda_i} \mathbf{v}_i = \mathbf{X} \mathbf{u}_i$ [Chatfield and Collins, 1980]. The connection between PCA and classical scaling is described in more detail in, e.g., [Williams, 2002; Platt, 2005].

PCA and classical scaling have been successfully applied in a large number of domains such as face recognition [Turk and Pentland, 1991], coin classification [Huber *et al.*, 2005], and seismic series analysis [Posadas *et al.*, 1993]. PCA and classical scaling suffer from two main drawbacks.

First, in PCA, the size of the covariance matrix is proportional to the dimensionality of the datapoints. As a result, the computation of the eigenvectors might be infeasible for very high-dimensional data. In datasets in which $n < D$, this drawback may be overcome by performing classical scaling instead of PCA, because the classical scaling scales with the number of datapoints instead of with the number of dimensions in the data. Alternatively, iterative techniques such as Simple PCA [Partridge and Calvo, 1997] or probabilistic PCA [Roweis, 1997] may be employed.

Second, the cost function in Equation 2.2 reveals that PCA and classical scaling focus mainly on retaining large pairwise distances $d_{i,j}^2$, instead of retaining the small pairwise distances, which is much more important.

Isomap

Classical scaling has proven to be successful in many applications, but it suffers from the fact that it mainly aims to retain pairwise Euclidean distances, and does not take into account the distribution of the neighboring datapoints. If the high-dimensional data lies on or near a curved manifold, such as in the Swiss roll dataset [Tenenbaum *et al.*, 2000], classical scaling might consider two datapoints as near points, whereas their distance over the manifold is much larger than the typical interpoint distance. Isomap [Tenenbaum *et al.*, 2000] is a technique that resolves this problem by attempting to preserve pairwise geodesic (or curvilinear) distances between datapoints. Geodesic distance is the distance between two points measured over the manifold.

In Isomap [Tenenbaum *et al.*, 2000], the geodesic distances between the datapoints \mathbf{x}_i ($i = 1, 2, \dots, n$) are computed by constructing a neighborhood graph G , in which every datapoint \mathbf{x}_i is connected with its k nearest neighbors \mathbf{x}_{i_j} ($j = 1, 2, \dots, k$) in the dataset \mathbf{X} . The shortest path between two points in the graph forms an estimate of the geodesic distance between these two points, and can easily be computed using Dijkstra's or Floyd's shortest-path algorithm [Dijkstra, 1959; Floyd, 1962]. The geodesic distances between all datapoints in \mathbf{X} are computed, thereby forming a pairwise geodesic distance matrix. The low-dimensional representations \mathbf{y}_i of the datapoints \mathbf{x}_i in the low-dimensional space \mathbf{Y} are computed by applying classical scaling (see 2.2.1) on the resulting pairwise geodesic distance matrix.

An important weakness of the Isomap algorithm is its topological instability [Balasubramanian and Schwartz, 2002]. Isomap may construct erroneous connections in the neighborhood graph G . Such short-circuiting [Lee and Verleysen, 2005] can severely impair the performance of Isomap. Several approaches have been proposed to overcome the problem of short-circuiting, e.g., by removing datapoints with large total flows in the shortest-path algorithm [Choi and Choi, 2007] or by removing nearest neighbors that violate local linearity of the neighborhood

graph [Saxena *et al.*, 2004]. A second weakness is that Isomap may suffer from ‘holes’ in the manifold. This problem can be dealt with by tearing manifolds with holes [Lee and Verleysen, 2005]. A third weakness of Isomap is that it can fail if the manifold is non-convex [Tenenbaum, 1998]. Despite these three weaknesses, Isomap was successfully applied on tasks such as wood inspection [Niskanen and Silvén, 2003], visualization of biomedical data [Lim *et al.*, 2003], and head pose estimation [Raytchev *et al.*, 2004].

Kernel PCA

Kernel PCA (KPCA) is the reformulation of traditional linear PCA in a high-dimensional space that is constructed using a kernel function [Schölkopf *et al.*, 1998]. In recent years, the reformulation of linear techniques using the ‘kernel trick’ has led to the proposal of successful techniques such as kernel ridge regression and Support Vector Machines [Shawe-Taylor and Christianini, 2004]. Kernel PCA computes the principal eigenvectors of the kernel matrix, rather than those of the covariance matrix. The reformulation of PCA in kernel space is straightforward, since a kernel matrix is similar to the dot product of the datapoints in the high-dimensional space that is constructed using the kernel function. The application of PCA in the kernel space provides Kernel PCA with the property of constructing nonlinear mappings. Kernel PCA computes the kernel matrix \mathbf{K} of the datapoints \mathbf{x}_i . The entries in the kernel matrix are defined by

$$k_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j), \quad (2.4)$$

where κ is a kernel function [Shawe-Taylor and Christianini, 2004], which may be any function that gives rise to a positive-semidefinite kernel \mathbf{K} . Subsequently, the kernel matrix \mathbf{K} is double-centered using the following modification of the entries

$$k_{ij} = -\frac{1}{2} \left(k_{ij} - \frac{1}{n} \sum_l k_{il} - \frac{1}{n} \sum_l k_{jl} + \frac{1}{n^2} \sum_{lm} k_{lm} \right). \quad (2.5)$$

The centering operation corresponds to subtracting the mean of the features in traditional PCA: it subtracts the mean of the data in the feature space defined by the kernel function κ . As a result, the data in the features space defined by the kernel function is zero-mean. Subsequently, the principal d eigenvectors \mathbf{v}_i of the centered kernel matrix are computed. The eigenvectors of the covariance matrix \mathbf{a}_i (in the feature space constructed by κ) can now be computed, since they are related to the eigenvectors of the kernel matrix \mathbf{v}_i (see, e.g., [Chatfield and Collins, 1980]) through

$$\mathbf{a}_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{v}_i. \quad (2.6)$$

In order to obtain the low-dimensional data representation, the data is projected onto the eigenvectors of the covariance matrix \mathbf{a}_i . The result of the projection (i.e., the low-dimensional data representation \mathbf{Y}) is given by

$$\mathbf{y}_i = \left\{ \sum_{j=1}^n a_1^{(j)} \kappa(\mathbf{x}_j, \mathbf{x}_i), \dots, \sum_{j=1}^n a_d^{(j)} \kappa(\mathbf{x}_j, \mathbf{x}_i) \right\}, \quad (2.7)$$

where $a_1^{(j)}$ indicates the j th value in the vector \mathbf{a}_1 and κ is the kernel function that was also used in the computation of the kernel matrix. Since Kernel PCA is a kernel-based method, the mapping performed by Kernel PCA relies on the choice of the kernel function κ . Possible choices for the kernel function include the linear kernel (making Kernel PCA equal to traditional PCA), the polynomial kernel, and the Gaussian kernel that is given in Equation 2.8 [Shawe-Taylor and Christianini, 2004]. Notice that when the linear kernel is employed, the kernel matrix K is equal to the Gram matrix, and the procedure described above is thus identical to classical scaling (see 2.2.1).

An important weakness of Kernel PCA is that the size of the kernel matrix is proportional to the square of the number of instances in the dataset. An approach to resolve this weakness is proposed by Tipping [2000]. Kernel PCA has been successfully applied to, e.g., face recognition [Kim *et al.*, 2002], speech recognition [Lima *et al.*, 2004], and novelty detection [Hoffmann, 2007].

MVU

As described above, Kernel PCA allows for performing PCA in the feature space that is defined by the kernel function κ . However, it is unclear how the kernel function κ should be selected. Maximum Variance Unfolding (MVU, formerly known as Semidefinite Embedding) is a technique that attempts to resolve this problem by *learning* the kernel matrix. MVU learns the kernel matrix by defining a neighborhood graph on the data (as in Isomap) and retaining pairwise distances in the resulting graph [Weinberger *et al.*, 2004]. MVU is different from Isomap in that it explicitly attempts to ‘unfold’ the data manifold. It does so by maximizing the Euclidean distances between the datapoints, under the constraint that the distances in the neighborhood graph are left unchanged (i.e., under the constraint that the local geometry of the data manifold is not distorted). The resulting optimization problem can be solved using semidefinite programming.

MVU starts with the construction of a neighborhood graph G , in which each datapoint \mathbf{x}_i is connected to its k nearest neighbors \mathbf{x}_{i_j} ($j = 1, 2, \dots, k$). Subsequently, MVU attempts to maximize the sum of the squared Euclidean distances between all datapoints, under the constraint that the distances inside the neighborhood graph G are preserved. In other words, MVU performs the following optimization problem.

$$\begin{aligned} & \text{Maximize } \sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \text{ subject to (1), with:} \\ & (1) \|\mathbf{y}_i - \mathbf{y}_j\|^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \text{ for } \forall (i, j) \in G \end{aligned}$$

MVU reformulates the optimization problem as a semidefinite programming problem (SDP) [Vandenberghe and Boyd, 1996] by defining the kernel matrix K as the inner product of the low-dimensional data representation \mathbf{Y} . The optimization problem then reduces to the

following SDP, which learns the kernel matrix \mathbf{K} .

Maximize $\text{trace}(K)$ subject to (1), (2), and (3), with:

$$(1) k_{ii} + k_{jj} - 2k_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \text{ for } \forall(i, j) \in G$$

$$(2) \sum_{ij} k_{ij} = 0$$

$$(3) \mathbf{K} \succeq 0$$

The solution \mathbf{K} of the SDP is the kernel matrix that is used as input for Kernel PCA. The low-dimensional data representation \mathbf{Y} is obtained by performing an eigendecomposition of the kernel matrix \mathbf{K} that was constructed by solving the SDP.

MVU has a weakness similar to Isomap: short-circuiting may impair the performance of MVU, because it adds constraints to the optimization problem that prevent successful unfolding of the manifold. Despite this weakness, MVU was successfully applied to, e.g., sensor localization [Weinberger *et al.*, 2007] and DNA microarray data analysis [Kharal, 2006].

Diffusion maps

The diffusion maps (DM) framework [Lafon and Lee, 2006; Nadler *et al.*, 2006] originates from the field of dynamical systems. Diffusion maps are based on defining a Markov random walk on the graph of the data. By performing the random walk for a number of timesteps, a measure for the proximity of the datapoints is obtained. Using this measure, the so-called diffusion distance is defined. In the low-dimensional representation of the data, the pairwise diffusion distances are retained as good as possible. The key idea behind the diffusion distance is that it is based on integrating over all paths through the graph. This makes the diffusion distance more robust to short-circuiting than, e.g., the geodesic distance that is employed in Isomap.

In the diffusion maps framework, a graph of the data is constructed first. The weights of the edges in the graph are computed using the Gaussian kernel function, leading to a matrix \mathbf{W} with entries

$$w_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}, \quad (2.8)$$

where σ indicates the variance of the Gaussian. Subsequently, normalization of the matrix \mathbf{W} is performed in such a way that its rows add up to 1. In this way, a matrix $\mathbf{P}^{(1)}$ is formed with entries

$$p_{ij}^{(1)} = \frac{w_{ij}}{\sum_k w_{ik}}. \quad (2.9)$$

Since diffusion maps originate from dynamical systems theory, the resulting matrix $\mathbf{P}^{(1)}$ is considered a Markov matrix that defines the forward transition probability matrix of a dynamical process. Hence, the matrix $\mathbf{P}^{(1)}$ represents the probability of a transition from one datapoint to another datapoint in a single timestep. The forward probability matrix for t timesteps $\mathbf{P}^{(t)}$ is thus given by $(\mathbf{P}^{(1)})^t$. Using the random walk forward probabilities $p_{ij}^{(t)}$, the diffusion distance is defined by

$$D^{(t)}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_k \frac{(p_{ik}^{(t)} - p_{jk}^{(t)})^2}{\psi(\mathbf{x}_k)^{(0)}}}. \quad (2.10)$$

In Equation 2.10, $\psi(\mathbf{x}_i)^{(0)}$ is a term that attributes more weight to parts of the graph with high density. It is defined by $\psi(\mathbf{x}_i)^{(0)} = \frac{m_i}{\sum_j m_j}$, where m_i is the degree of node \mathbf{x}_i defined by $m_i = \sum_j p_{ij}$. From Equation 2.10, it can be observed that pairs of datapoints with a high forward transition probability have a small diffusion distance. Since the diffusion distance is based on integrating over all paths through the graph, it is more robust to short-circuiting than the geodesic distance that is employed in Isomap. In the low-dimensional representation of the data \mathbf{Y} , diffusion maps attempt to retain the diffusion distances. Using spectral theory on the random walk, it has been shown (see, e.g., [Lafon and Lee, 2006]) that the low-dimensional representation \mathbf{Y} that retains the diffusion distances $D^{(t)}(\mathbf{x}_i, \mathbf{x}_j)$ as good as possible (under a squared error criterion) is formed by the d nontrivial principal eigenvectors of the eigenproblem

$$\mathbf{P}^{(t)}\mathbf{v} = \lambda\mathbf{v}. \quad (2.11)$$

Because the graph is fully connected, the largest eigenvalue is trivial (viz. $\lambda_1 = 1$), and its eigenvector \mathbf{v}_1 is thus discarded. The low-dimensional representation \mathbf{Y} is given by the next d principal eigenvectors. In the low-dimensional representation, the eigenvectors are normalized by their corresponding eigenvalues. Hence, the low-dimensional data representation is given by

$$\mathbf{Y} = \{\lambda_2\mathbf{v}_2, \lambda_3\mathbf{v}_3, \dots, \lambda_{d+1}\mathbf{v}_{d+1}\}. \quad (2.12)$$

Diffusion maps have been successfully applied to, e.g., shape matching [Rajpoot *et al.*, 2007] and gene expression analysis [Xu *et al.*, 2007].

2.2.2 Sparse spectral techniques

In the previous subsection, we discussed five techniques that construct a low-dimensional representation of the high-dimensional data by performing an eigendecomposition of a full matrix. In contrast, the four techniques discussed in this subsection solve a sparse (generalized) eigenproblem. All presented sparse spectral techniques only focus on retaining local structure of the data. We discuss four sparse spectral dimensionality reduction techniques, viz. (1) LLE, (2) Laplacian Eigenmaps, (3) Hessian LLE, and (4) LTSA.

LLE

Local Linear Embedding (LLE) [Roweis and Saul, 2000] is a technique that is similar to Isomap (and MVU) in that it constructs a graph representation of the datapoints. In contrast to Isomap, it attempts to preserve solely local properties of the data. As a result, LLE is less sensitive to short-circuiting than Isomap, because only a small number of local properties are affected if short-circuiting occurs. Furthermore, the preservation of local properties allows for successful embedding of non-convex manifolds. In LLE, the local properties of the data manifold are constructed by writing the high-dimensional datapoints as a linear combination of their nearest neighbors. In the low-dimensional representation of the data, LLE attempts to retain the reconstruction weights in the linear combinations as good as possible.

LLE describes the local properties of the manifold around a datapoint \mathbf{x}_i by writing the datapoint as a linear combination \mathbf{W}_i (the so-called reconstruction weights) of its k nearest neighbors \mathbf{x}_{i_j} . Hence, LLE fits a hyperplane through the datapoint \mathbf{x}_i and its nearest neighbors, thereby

assuming that the manifold is locally linear. The local linearity assumption implies that the reconstruction weights \mathbf{W}_i of the datapoints \mathbf{x}_i are invariant to translation, rotation, and rescaling. Because of the invariance to these transformations, any linear mapping of the hyperplane to a space of lower dimensionality preserves the reconstruction weights in the space of lower dimensionality. In other words, if the low-dimensional data representation preserves the local geometry of the manifold, the reconstruction weights \mathbf{W}_i that reconstruct datapoint \mathbf{x}_i from its neighbors in the high-dimensional data representation also reconstruct datapoint \mathbf{y}_i from its neighbors in the low-dimensional data representation. As a consequence, finding the d -dimensional data representation \mathbf{Y} amounts to minimizing the cost function

$$\phi(\mathbf{Y}) = \sum_i \|\mathbf{y}_i - \sum_{j=1}^k w_{ij} \mathbf{y}_{i_j}\|^2 \text{ subject to } \|\mathbf{y}^{(k)}\|^2 = 1 \text{ for } \forall k, \quad (2.13)$$

where $\mathbf{y}^{(k)}$ represents the k th column of the solution matrix \mathbf{Y} . Roweis and Saul [2000] showed² that the coordinates of the low-dimensional representations \mathbf{y}_i that minimize this cost function are found by computing the eigenvectors corresponding to the smallest d nonzero eigenvalues of the inproduct $(\mathbf{I}_n - \mathbf{W})^T(\mathbf{I}_n - \mathbf{W})$, where \mathbf{W} is a sparse $n \times n$ matrix of which the entries are set to 0 if i and j are not connected in the neighborhood graph, and equal to the corresponding reconstruction weight otherwise. In this formula, \mathbf{I}_n is the $n \times n$ identity matrix.

The popularity of LLE has led to the proposal of linear variants of the algorithm [He *et al.*, 2005; Kokiopoulou and Saad, 2007], and to successful applications, e.g., to superresolution [Chang *et al.*, 2004] and sound source localization [DuraiSwami and Raykar, 2005]. However, there also exist experimental studies that report weak performance of LLE. Lim *et al.* [2003] report that LLE fails in the visualization of even simple synthetic biomedical datasets. Jenkins and Mataric [2002] claim that LLE performs worse than Isomap in the derivation of perceptual-motor actions. A possible explanation lies in the difficulties that LLE has when confronted with manifolds that contain holes [Roweis and Saul, 2000]. In addition, LLE tends to collapse large portions of the data close together in the low-dimensional space.

Laplacian Eigenmaps

Similar to LLE, Laplacian Eigenmaps find a low-dimensional data representation by preserving local properties of the manifold [Belkin and Niyogi, 2002]. In Laplacian Eigenmaps, the local properties are based on the pairwise distances between near neighbors. Laplacian Eigenmaps compute a low-dimensional representation of the data in which the distances between a datapoint and its k nearest neighbors are minimized. This is done in a weighted manner, i.e., the distance in the low-dimensional data representation between a datapoint and its first nearest neighbor contributes more to the cost function than the distance between the datapoint and its second nearest neighbor. Using spectral graph theory, the minimization of the cost function is defined as an eigenproblem.

The Laplacian Eigenmap algorithm first constructs a neighborhood graph G in which every datapoint \mathbf{x}_i is connected to its k nearest neighbors. For all points \mathbf{x}_i and \mathbf{x}_j in graph G that are

² $\phi(\mathbf{Y}) = (\mathbf{Y} - \mathbf{W}\mathbf{Y})^2 = \mathbf{Y}^T(\mathbf{I}_n - \mathbf{W})^T(\mathbf{I}_n - \mathbf{W})\mathbf{Y}$ is the function that has to be minimized. Hence, the eigenvectors of $(\mathbf{I}_n - \mathbf{W})^T(\mathbf{I}_n - \mathbf{W})$ corresponding to the smallest nonzero eigenvalues form the solution that minimizes $\phi(\mathbf{Y})$.

connected by an edge, the weight of the edge is computed using the Gaussian kernel function (see Equation 2.8), leading to a sparse adjacency matrix \mathbf{W} . In the computation of the low-dimensional representations \mathbf{y}_i , the cost function that is minimized is given by

$$\phi(\mathbf{Y}) = \sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 w_{ij}. \quad (2.14)$$

In the cost function, large weights w_{ij} correspond to small distances between the high-dimensional datapoints \mathbf{x}_i and \mathbf{x}_j . Hence, the difference between their low-dimensional representations \mathbf{y}_i and \mathbf{y}_j highly contributes to the cost function. As a consequence, nearby points in the high-dimensional space are put as close together as possible in the low-dimensional representation.

The computation of the degree matrix \mathbf{M} and the graph Laplacian \mathbf{L} of the graph \mathbf{W} allows for formulating the minimization problem in Equation 2.14 as an eigenproblem [Anderson and Morley, 1985]. The degree matrix \mathbf{M} of \mathbf{W} is a diagonal matrix, of which the entries are the row sums of \mathbf{W} (i.e., $m_{ii} = \sum_j w_{ij}$). The graph Laplacian \mathbf{L} is computed by $\mathbf{L} = \mathbf{M} - \mathbf{W}$. It can be shown that the following holds³

$$\phi(\mathbf{Y}) = \sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 w_{ij} = 2\mathbf{Y}^T \mathbf{L} \mathbf{Y}. \quad (2.15)$$

Hence, minimizing $\phi(\mathbf{Y})$ is proportional to minimizing $\mathbf{Y}^T \mathbf{L} \mathbf{Y}$ subject to $\mathbf{Y}^T \mathbf{M} \mathbf{Y} = \mathbf{I}_n$, a covariance constraint that is similar to that of LLE. The low-dimensional data representation \mathbf{Y} can thus be found by solving the generalized eigenvalue problem

$$\mathbf{L} \mathbf{v} = \lambda \mathbf{M} \mathbf{v} \quad (2.16)$$

for the d smallest nonzero eigenvalues. The d eigenvectors \mathbf{v}_i corresponding to the smallest nonzero eigenvalues form the low-dimensional data representation \mathbf{Y} .

Laplacian Eigenmaps (and its variants) have been successfully applied, e.g., to clustering [Weiss, 1999; Shi and Malik, 2000; Ng *et al.*, 2001], face recognition [He *et al.*, 2005], and the analysis of fMRI data [Brun *et al.*, 2003]. In addition, variants of Laplacian Eigenmaps may be applied to supervised or semi-supervised learning problems [Belkin and Niyogi, 2004; Costa and Hero, 2005]. A linear variant of Laplacian Eigenmaps is presented by He and Niyogi [2004].

Hessian LLE

Hessian LLE (HLLE) [Donoho and Grimes, 2005] is a variant of LLE that minimizes the ‘curviness’ of the high-dimensional manifold when embedding it into a low-dimensional space, under the constraint that the low-dimensional data representation is locally isometric. This is done by an eigenanalysis of a matrix \mathcal{H} that describes the curviness of the manifold around the datapoints. The curviness of the manifold is measured by means of the local Hessian at every datapoint. The local Hessian is represented in the local tangent space at the datapoint, in order to obtain a representation of the local Hessian that is invariant to differences in the positions of the datapoints.

³Note that $\phi(\mathbf{Y}) = \sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 w_{ij} = \sum_{ij} (\|\mathbf{y}_i\|^2 + \|\mathbf{y}_j\|^2 - 2\mathbf{y}_i \mathbf{y}_j^T) w_{ij} = \sum_i \|\mathbf{y}_i\|^2 m_{ii} + \sum_j \|\mathbf{y}_j\|^2 m_{jj} - 2 \sum_{ij} \mathbf{y}_i \mathbf{y}_j^T w_{ij} = 2\mathbf{Y}^T \mathbf{M} \mathbf{Y} - 2\mathbf{Y}^T \mathbf{W} \mathbf{Y} = 2\mathbf{Y}^T \mathbf{L} \mathbf{Y}$

It can be shown⁴ that the coordinates of the low-dimensional representation can be found by performing an eigenanalysis of \mathcal{H} .

Hessian LLE starts with identifying the k nearest neighbors for each datapoint \mathbf{x}_i using Euclidean distance. In the neighborhood, local linearity of the manifold is assumed. Hence, a basis for the local tangent space at point \mathbf{x}_i can be found by applying PCA on its k nearest neighbors \mathbf{x}_{i_j} . In other words, for every datapoint \mathbf{x}_i , a basis for the local tangent space at point \mathbf{x}_i is determined by computing the d principal eigenvectors $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_d\}$ of the covariance matrix $\text{cov}(\mathbf{x}_i)$. Note that the above requires that $k \geq d$. Subsequently, an estimator for the Hessian of the manifold at point \mathbf{x}_i in local tangent space coordinates is computed. In order to do this, a matrix \mathbf{Z}_i is formed that contains (in the columns) all cross products of \mathbf{M} up to the d^{th} order (including a column with ones). The matrix \mathbf{Z}_i is orthonormalized by applying Gram-Schmidt orthonormalization [Afken, 1985]. The estimation of the tangent Hessian \mathbf{H}_i is now given by the transpose of the last $\frac{d(d+1)}{2}$ columns of the matrix \mathbf{Z}_i . Using the Hessian estimators in local tangent coordinates, a matrix \mathcal{H} is constructed with entries

$$\mathcal{H}_{lm} = \sum_i \sum_j ((\mathbf{H}_i)_{jl} \times (\mathbf{H}_i)_{jm}). \quad (2.17)$$

The matrix \mathcal{H} represents information on the curviness of the high-dimensional data manifold. An eigenanalysis of \mathcal{H} is performed in order to find the low-dimensional data representation that minimizes the curviness of the manifold. The eigenvectors corresponding to the d smallest nonzero eigenvalues of \mathcal{H} are selected and form the matrix \mathbf{Y} , which contains the low-dimensional representation of the data. A successful application of Hessian LLE to sensor localization has been presented by Patwari and Hero [2004].

LTSA

Similar to Hessian LLE, Local Tangent Space Analysis (LTSA) is a technique that describes local properties of the high-dimensional data using the local tangent space of each datapoint [Zhang and Zha, 2004]. LTSA is based on the observation that, if local linearity of the manifold is assumed, there exists a linear mapping from a high-dimensional datapoint to its local tangent space, and that there exists a linear mapping from the corresponding low-dimensional datapoint to the same local tangent space [Zhang and Zha, 2004]. LTSA attempts to align these linear mappings in such a way, that they construct the local tangent space of the manifold from the low-dimensional representation. In other words, LTSA simultaneously searches for the coordinates of the low-dimensional data representations, and for the linear mappings of the low-dimensional datapoints to the local tangent space of the high-dimensional data.

Similar to Hessian LLE, LTSA starts with computing bases for the local tangent spaces at the datapoints \mathbf{x}_i . This is done by applying PCA on the k datapoints \mathbf{x}_{i_j} that are neighbors of datapoint \mathbf{x}_i . This results in a mapping \mathbf{M}_i from the neighborhood of \mathbf{x}_i to the local tangent space \mathbf{Z}_i . A property of the local tangent space \mathbf{Z}_i is that there exists a linear mapping \mathbf{L}_i from the local tangent space coordinates \mathbf{z}_{i_j} to the low-dimensional representations \mathbf{y}_{i_j} . Using this

⁴The derivation can be found in [Donoho and Grimes, 2005].

property of the local tangent space, LTSA performs the following minimization

$$\min_{\mathbf{Y}, \mathbf{L}_i} \sum_i \|\mathbf{Y}_i \mathbf{J}_k - \mathbf{L}_i \mathbf{Z}_i\|^2, \quad (2.18)$$

where \mathbf{J}_k is the centering matrix (i.e., the matrix that performs the transformation in Equation 2.5) of size k [Shawe-Taylor and Christianini, 2004]. Zhang and Zha [Zhang and Zha, 2004] have shown that the solution of the minimization is formed by the eigenvectors of an alignment matrix \mathbf{B} , that correspond to the d smallest nonzero eigenvalues of \mathbf{B} . The entries of the alignment matrix \mathbf{B} are obtained by iterative summation (for all matrices \mathbf{V}_i and starting from $b_{ij}^{(0)} = 0$ for $\forall i, j$)

$$\mathbf{B}_{N_i N_i}^{(t)} = \mathbf{B}_{N_i N_i}^{(t-1)} + \mathbf{J}_k (\mathbf{I} - \mathbf{V}_i \mathbf{V}_i^T) \mathbf{J}_k, \quad (2.19)$$

where N_i is the set of indices of the nearest neighbors of datapoint \mathbf{x}_i and t represents the number of the iteration. Subsequently, the low-dimensional representation \mathbf{Y} is obtained by computation of the eigenvectors corresponding to the d smallest nonzero eigenvectors of the symmetric matrix $\frac{1}{2}(\mathbf{B} + \mathbf{B}^T)$.

[Teng *et al.*, 2005] report on a successful application of LTSA to microarray data. A linear variant of LTSA is proposed by Zhang *et al.* [2007].

2.3 Non-convex techniques for dimensionality reduction

In the previous section, we discussed techniques that construct a low-dimensional data representation by optimizing a convex objective function by means of an eigendecomposition. In this section, we discuss four techniques that optimize a non-convex objective function. Specifically, we discuss a non-convex technique for multidimensional scaling that forms an alternative to classical scaling called Sammon mapping, a technique based on training multilayer neural networks (multilayer autoencoders), and two techniques that compute a mixture of local linear models and perform a global alignment of these linear models (LLC and manifold charting).

Sammon mapping

In subsection 2.2.1, we discussed classical scaling, a convex technique for multidimensional scaling [Torgerson, 1952], and noted that the main weakness of this technique is that it mainly focuses on retaining large pairwise distances, and not on retaining the small pairwise distances, which are much more important. Several multidimensional scaling variants have been proposed that aim to address this weakness [Sammon, 1969; Demartines and Hérault, 1997; Lee *et al.*, 2000; Hinton and Roweis, 2002; Agrafiotis, 2003; Nam *et al.*, 2004]. In this subsection, we discuss one of these variants called Sammon mapping [Sammon, 1969].

Sammon mapping (SM) adapts the classical scaling cost function (see Equation 2.2) by weighting the contribution of each pair (i, j) to the cost function by the inverse of their pairwise distance in the high-dimensional space d_{ij} . In this way, the cost function assigns roughly equal weight to retaining each of the pairwise distances, and thus retains the local structure of the data better than classical scaling. Mathematically, the Sammon cost function is given by

$$\phi(\mathbf{Y}) = \frac{1}{\sum_{i,j} d_{ij}} \sum_{i \neq j} \frac{(d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2}{d_{ij}}, \quad (2.20)$$

where d_{ij} represents the pairwise Euclidean distance between the high-dimensional datapoints \mathbf{x}_i and \mathbf{x}_j , and the constant in front is added in order to simplify the gradient of the cost function. The minimization of the Sammon cost function is generally performed using a pseudo-Newton method [Cox and Cox, 1994]. Sammon mapping is mainly used for visualization purposes.

Multilayer autoencoders

Multilayer autoencoders (AE) are feed-forward neural networks with an odd number of hidden layers [DeMers and Cottrell, 1993; Hinton and Salakhutdinov, 2006]. The middle hidden layer has d nodes, and the input and the output layer have D nodes. An example of an autoencoder is shown schematically in Figure 2.2. The network is trained to minimize the mean squared error between the input and the output of the network (ideally, the input and the output are equal). Training the neural network on the datapoints \mathbf{x}_i leads to a network in which the middle hidden layer gives a d -dimensional representation of the datapoints that preserves as much structure in \mathbf{X} as possible. The low-dimensional representations \mathbf{y}_i can be obtained by extracting the node values in the middle hidden layer, when datapoint \mathbf{x}_i is used as input. In order to allow the autoencoder to learn a nonlinear mapping between the high-dimensional and low-dimensional data representation, sigmoid activation functions are generally used (except in the middle layer, where a linear activation function is used).

Multilayer autoencoders usually have a high number of connections. Therefore, backpropagation approaches converge slowly and are likely to get stuck in local minima. Hinton *et al.* [2006] overcome this drawback by a learning procedure that consists of three main stages.

First, the recognition layers of the network (i.e., the layers from \mathbf{X} to \mathbf{Y}) are trained one-by-one using Restricted Boltzmann Machines (RBMs). An RBM is a Markov Random Field with a bipartite graph structure of visible and hidden nodes. Typically, the nodes are binary stochastic random variables (i.e., they obey a Bernoulli distribution) but for continuous data the binary nodes may be replaced by mean-field logistic or exponential family nodes [Welling *et al.*, 2004]. RBMs can be trained efficiently using an unsupervised learning procedure that minimizes the so-called contrastive divergence [Hinton, 2002]. We describe the training of RBMs in more detail in Appendix D. Second, the reconstruction layers of the network (i.e., the layers from \mathbf{Y} to \mathbf{X}') are formed by the inverse of the trained recognition layers. In other words, the autoencoder is unrolled. Third, the unrolled autoencoder is finetuned in a supervised manner using backpropagation as to minimize the mean squared error between the input and the output of the autoencoder.

Autoencoders have successfully been applied to problems such as missing data imputation [Abdella and Marwala, 2005] and HIV analysis [Betechuoh *et al.*, 2006].

LLC

Locally Linear Coordination (LLC) [Teh and Roweis, 2002] computes a number of locally linear models and subsequently performs a global alignment of the linear models. This process consists of two steps: (1) computing a mixture of local linear models on the data by means of an Expectation Maximization (EM) algorithm and (2) aligning the local linear models in order to obtain the low-dimensional data representation using a variant of LLE.

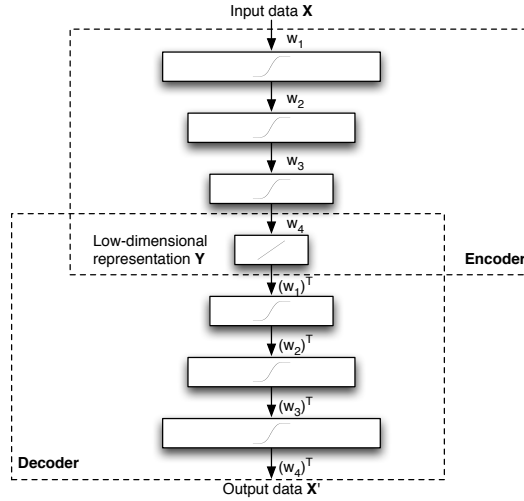


Figure 2.2 Schematic structure of an autoencoder.

LLC first constructs a mixture of m factor analyzers (MoFA)⁵ using the EM algorithm [Dempster *et al.*, 1977; Ghahramani and Hinton, 1996a; Kambhatla and Leen, 1997]. Alternatively, a mixture of probabilistic PCA model (MoPPCA) [Tipping and Bishop, 1999] could be employed. The local linear models in the mixture are used to construct m data representations \mathbf{z}_{ij} and their corresponding responsibilities r_{ij} (where $j \in \{1, \dots, m\}$) for every datapoint \mathbf{x}_i . The responsibilities r_{ij} describe to what extent datapoint \mathbf{x}_i corresponds to the model j ; they satisfy $\sum_j r_{ij} = 1$. Using the local models and the corresponding responsibilities, responsibility-weighted data representations $\mathbf{u}_{ij} = r_{ij}\mathbf{z}_{ij}$ are computed. The responsibility-weighted data representations \mathbf{u}_{ij} are stored in a $n \times mD$ block matrix \mathbf{U} . The alignment of the local models is performed based on \mathbf{U} and on a matrix \mathbf{M} that is given by $\mathbf{M} = (\mathbf{I}_n - \mathbf{W})^T(\mathbf{I}_n - \mathbf{W})$. Herein, the matrix \mathbf{W} contains the reconstruction weights computed by LLE (see 2.2.2), and \mathbf{I}_n denotes the $n \times n$ identity matrix. LLC aligns the local models by solving the generalized eigenproblem

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{B}\mathbf{v}, \quad (2.21)$$

for the d smallest nonzero eigenvalues⁶. In Equation 2.21, \mathbf{A} is the inner product of $\mathbf{M}^T\mathbf{U}$ and \mathbf{B} is the inner product of \mathbf{U} . The d eigenvectors \mathbf{v}_i form a matrix \mathbf{L} , that can be shown to define a linear mapping from the responsibility-weighted data representation \mathbf{U} to the underlying low-dimensional data representation \mathbf{Y} . The low-dimensional data representation is thus obtained by computing $\mathbf{Y} = \mathbf{U}\mathbf{L}$.

LLC has been successfully applied to face images of a single person with variable pose and expression, and to handwritten digits [Teh and Roweis, 2002].

⁵Note that the mixture of factor analyzers (and the mixture of probabilistic PCA model) is a mixture of Gaussians model with a restriction on the covariance of the Gaussians.

⁶The derivation of this eigenproblem can be found in [Teh and Roweis, 2002].

Manifold charting

Similar to LLC, manifold charting constructs a low-dimensional data representation by aligning a MoFA model or a MoPPCA model [Brand, 2002]. In contrast to LLC, manifold charting does not minimize a cost function that corresponds to another dimensionality reduction technique (such as the LLE cost function). Manifold charting minimizes a convex cost function that measures the amount of disagreement between the linear models on the global coordinates of the datapoints. The minimization of this cost function can be performed by solving a generalized eigenproblem.

Manifold charting first performs the EM algorithm to learn a mixture of factor analyzers, in order to obtain m low-dimensional data representations \mathbf{z}_{ij} and corresponding responsibilities r_{ij} (where $j \in \{1, \dots, m\}$) for all datapoints \mathbf{x}_i . Manifold charting finds a linear mapping from the data representations \mathbf{z}_{ij} to the global coordinates \mathbf{y}_i that minimizes the cost function

$$\phi(\mathbf{Y}) = \sum_{i=1}^n \sum_{j=1}^m r_{ij} \|\mathbf{y}_i - \mathbf{y}_{ij}\|^2, \quad (2.22)$$

where $\mathbf{y}_i = \sum_{k=1}^m r_{ik} \mathbf{y}_{ik}$, and $\mathbf{y}_{ij} = \mathbf{M} \mathbf{z}_{ij}$. The intuition behind the cost function is that whenever there are two linear models in which a datapoint has a high responsibility, these linear models should agree on the final coordinate of the datapoint. The cost function can be rewritten in the form

$$\phi(\mathbf{Y}) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^m r_{ij} r_{ik} \|\mathbf{y}_{ij} - \mathbf{y}_{ik}\|^2, \quad (2.23)$$

which allows the cost function to be rewritten in the form of a Rayleigh quotient. The Rayleigh quotient can be constructed by the definition of a block-diagonal matrix \mathbf{D} with m blocks by

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{D}_m \end{pmatrix}, \quad (2.24)$$

where \mathbf{D}_j is the sum of the weighted covariances of the data representations \mathbf{z}_{ij} . Hence, \mathbf{D}_j is given by

$$\mathbf{D}_j = \sum_{i=1}^n r_{ij} \text{cov}([\mathbf{z}_{ij} \quad \mathbf{1}]). \quad (2.25)$$

In Equation 2.25, the 1-column is added to the data representation \mathbf{z}_{ij} in order to facilitate translations in the construction of \mathbf{y}_i from the data representations \mathbf{z}_{ij} . Using the definition of the matrix \mathbf{D} and the $n \times mD$ block-diagonal matrix \mathbf{U} with entries $\mathbf{u}_{ij} = r_{ij} [\mathbf{z}_{ij} \quad \mathbf{1}]$, the manifold charting cost function can be rewritten as

$$\phi(\mathbf{Y}) = \mathbf{L}^T (\mathbf{D} - \mathbf{U}^T \mathbf{U}) \mathbf{L}, \quad (2.26)$$

where \mathbf{L} represents the linear mapping on the matrix \mathbf{Z} that can be used to compute the final low-dimensional data representation \mathbf{Y} . The linear mapping \mathbf{L} can thus be computed by solving the generalized eigenproblem

$$(\mathbf{D} - \mathbf{U}^T \mathbf{U}) \mathbf{v} = \lambda \mathbf{U}^T \mathbf{U} \mathbf{v}, \quad (2.27)$$

for the d smallest nonzero eigenvalues. The d eigenvectors \mathbf{v}_i form the columns of the linear combination \mathbf{L} from $[\mathbf{U} \quad \mathbf{1}]$ to \mathbf{Y} .

2.4 Characterization of the techniques

In Section 2.2 and 2.3, we provided an overview of techniques for dimensionality reduction. This section lists the techniques by three theoretical characterizations. First, relations between the dimensionality reduction techniques are identified (subsection 2.4.1). Second, we list and discuss a number of general properties of the techniques such as the nature of the objective function that is optimized and the computational complexity of the technique (subsection 2.4.2). Third, the out-of-sample extension of the techniques is discussed (subsection 2.4.3).

2.4.1 Relations

Many of the techniques discussed in Section 2.2 and 2.3 are highly interrelated, and in certain special cases even equivalent. In the previous sections, we already mentioned some of these relations, but in this subsection, we discuss the relations between the techniques in more detail. Specifically, we discuss three types of relations between the techniques.

First, traditional PCA is identical to performing classical scaling and to performing Kernel PCA with a linear kernel, due to the relation between the eigenvectors of the covariance matrix and the double-centered squared Euclidean distance matrix [Williams, 2002] (which is in turn equal to the Gram matrix). Autoencoders in which only linear activation functions are employed are very similar to PCA as well [Kung *et al.*, 1994].

Second, performing classical scaling on a pairwise geodesic distance matrix is identical to performing Isomap. Similarly, performing Isomap with the number of nearest neighbors k set to $n - 1$ is identical to performing classical scaling (and thus also to performing PCA and to performing Kernel PCA with a linear kernel). Diffusion maps are also very similar to classical scaling, however, they attempt to retain a different type of pairwise distances (the so-called diffusion distances). The main discerning property of diffusion maps is that its pairwise distance measure between the high-dimensional datapoints is based on integrating over all paths through the graph defined on the data.

Third, the spectral techniques Kernel PCA, Isomap, LLE, and Laplacian Eigenmaps can all be viewed upon as special cases of the more general problem of learning eigenfunctions [Hamm *et al.*, 2003; Bengio *et al.*, 2004a]. As a result, Isomap, LLE, and Laplacian Eigenmaps⁷ can be considered as special cases of Kernel PCA that use a specific kernel function κ . For instance, this relation is visible in the out-of-sample extensions of Isomap, LLE, and Laplacian Eigenmaps [Bengio *et al.*, 2004b]. The out-of-sample extension for these techniques is performed by means of a so-called Nyström approximation [Baker, 1977; Platt, 2005], which is known to be equivalent to the Kernel PCA projection [Schölkopf *et al.*, 1998] (see 2.4.3 for more details). Diffusion maps in which $t = 1$ are fairly similar to Kernel PCA with the Gaussian kernel function. There are two main differences between the two: (1) no centering of the Gram matrix is performed in diffusion maps (although centering is generally not essential in Kernel PCA [Shawe-Taylor and Christianini, 2004]) and (2) diffusion maps do not employ the principal eigenvector of the kernel matrix, whereas Kernel PCA does. MVU can also be viewed upon as a special case of Kernel PCA, in which the solution of the SDP is the kernel matrix. In turn, Isomap can be

⁷The same also holds for Hessian LLE and LTSA, but up to our knowledge, the kernel functions for these techniques have never been derived.

<i>Technique</i>	<i>Parametric</i>	<i>Parameters</i>	<i>Computational</i>	<i>Memory</i>
PCA	yes	none	$O(D^3)$	$O(D^2)$
Class. scaling	no	none	$O(n^3)$	$O(n^2)$
Isomap	no	k	$O(n^3)$	$O(n^2)$
Kernel PCA	no	$\kappa(\cdot, \cdot)$	$O(n^3)$	$O(n^2)$
MVU	no	k	$O((nk)^3)$	$O((nk)^3)$
Diffusion maps	no	σ, t	$O(n^3)$	$O(n^2)$
LLE	no	k	$O(pn^2)$	$O(pn^2)$
Laplacian Eigenmaps	no	k, σ	$O(pn^2)$	$O(pn^2)$
Hessian LLE	no	k	$O(pn^2)$	$O(pn^2)$
LTSA	no	k	$O(pn^2)$	$O(pn^2)$
Sammon mapping	no	none	$O(in^2)$	$O(n^2)$
Autoencoders	yes	net size	$O(inw)$	$O(w)$
LLC	yes	m, k	$O(imd^3)$	$O(nmd)$
Manifold charting	yes	m	$O(imd^3)$	$O(nmd)$

Table 2.1 Properties of techniques for dimensionality reduction.

viewed upon as a technique that finds an approximate solution to the MVU problem [Xiao *et al.*, 2006]. Evaluation of the dual MVU problem has also shown that LLE and Laplacian Eigenmaps show great resemblance to MVU [Xiao *et al.*, 2006].

As a consequence of the relations between the techniques, our empirical comparative evaluation in Section 2.5 does not include (1) classical scaling, (2) Kernel PCA using a linear kernel, and (3) autoencoders with linear activation functions, because they are similar to PCA. Furthermore, we do not evaluate Kernel PCA using a Gaussian kernel in the experiments, because of its resemblance to diffusion maps; instead we use a polynomial kernel.

2.4.2 General properties

In Table 2.1, the thirteen dimensionality reduction techniques are listed by four general properties: (1) the parametric nature of the mapping between the high-dimensional and the low-dimensional space, (2) the main free parameters that have to be optimized, (3) the computational complexity of the main computational part of the technique, and (4) the memory complexity of the technique. We discuss the four general properties below.

For property 1, Table 2.1 shows that most techniques for dimensionality reduction are non-parametric. This means that the technique does not specify a direct mapping from the high-dimensional to the low-dimensional space (or vice versa). The non-parametric nature of most techniques is a disadvantage for two main reasons: (1) it is not possible to generalize to held-out or new test data without performing the dimensionality reduction technique again and (2) it is not possible to obtain insight into how much information of the high-dimensional data was retained in the low-dimensional space by reconstructing the original data from the low-dimensional data representation and measuring the error between the reconstructed and true data.

For property 2, Table 2.1 shows that the objective functions of most nonlinear techniques for dimensionality reduction all have free parameters that need to be optimized. By free parameters, we mean parameters that directly influence the cost function that is optimized. The reader should note that non-convex techniques for dimensionality reduction have additional free parameters, such as the learning rate and the permitted maximum number of iterations. Moreover, LLE uses a regularization parameter in the computation of the reconstruction weights. The presence of free parameters has both advantages and disadvantages. The main advantage of the presence of free parameters is that they provide more flexibility to the technique, whereas their main disadvantage is that they need to be tuned to optimize the performance of the dimensionality reduction technique.

For properties 3 and 4, Table 2.1 provides insight into the computational and memory complexities of the computationally most expensive algorithmic components of the techniques. The computational complexity of a dimensionality reduction technique is of importance to its practical applicability. If the memory or computational resources needed are too large, application becomes infeasible. The computational complexity of a dimensionality reduction technique is determined by: (1) properties of the dataset such as the number of datapoints n and their dimensionality D and (2) parameters of the techniques, such as the target dimensionality d , the number of nearest neighbors k (for techniques based on neighborhood graphs), and the number of iterations i (for iterative techniques). In Table 2.1, p denotes the ratio of nonzero elements in a sparse matrix to the total number of elements, m indicates the number of local models in a mixture of factor analyzers, and w is the number of weights in a neural network. Below, we discuss the computational complexity and the memory complexity of each of the entries in the table.

The computationally most demanding part of PCA is the eigenanalysis of the $D \times D$ covariance matrix⁸, which is performed using a power method in $O(D^3)$. The corresponding memory complexity of PCA is $O(D^2)$. In datasets in which $n < D$, the computational and memory complexity of PCA can be reduced to $O(n^3)$ and $O(n^2)$, respectively (see subsection 2.2.1). Classical scaling, Isomap, diffusion maps, and Kernel PCA perform an eigenanalysis of an $n \times n$ matrix using a power method in $O(n^3)$. Because these full spectral techniques store a full $n \times n$ kernel matrix, the memory complexity of these techniques is $O(n^2)$.

In addition to the eigendecomposition of Kernel PCA, MVU solves a semidefinite program (SDP) with nk constraints. Both the computational and the memory complexity of solving an SDP are cube in the number of constraints [Borchers and Young, 2007]. Since there are nk constraints, the computational and memory complexity of the main part of MVU is $O((nk)^3)$. Training an autoencoder using RBM training or backpropagation has a computational complexity of $O(inw)$. The training of autoencoders may converge rather slowly, especially in cases where the input and target dimensionality are high (since this yields a high number of weights in the network). The memory complexity of autoencoders is $O(w)$.

The main computational part of LLC and manifold charting is the computation of the MoFA or MoPPCA model, which has computational complexity $O(imd^3)$. The corresponding memory complexity is $O(nmd)$. Sammon mapping has a computational complexity of $O(in^2)$. The corresponding memory complexity is $O(n^2)$, although the memory complexity may be reduced by computing the pairwise distances on-the-fly.

⁸In cases in which $n \gg D$, the main computational part of PCA may be the computation of the covariance matrix. We ignore this for now.

Similar to, e.g., Kernel PCA, sparse spectral techniques perform an eigenanalysis of an $n \times n$ matrix. However, for these techniques the $n \times n$ matrix is sparse which is beneficial, because it lowers the computational complexity of the eigenanalysis. Eigenanalysis of a sparse matrix (using Arnoldi methods [Arnoldi, 1951] or Jacobi-Davidson methods [Fokkema *et al.*, 1999]) has computational complexity $O(pn^2)$, where p is the ratio of nonzero elements in the sparse matrix to the total number of elements. The memory complexity is $O(pn^2)$ as well.

From the discussion of the four general properties of the techniques for dimensionality reduction above, we make four observations: (1) most nonlinear techniques for dimensionality reduction do not provide a parametric mapping between the high-dimensional and the low-dimensional space, (2) all nonlinear techniques require the optimization of one or more free parameters, (3) when $D < n$ (which is true in most cases), nonlinear techniques have computational disadvantages compared to PCA, and (4) a number of nonlinear techniques suffer from a memory complexity that is square or cube with the number of datapoints n . From these observations, it is clear that nonlinear techniques impose considerable demands on computational resources, as compared to PCA. Attempts to reduce the computational and/or memory complexities of nonlinear techniques have been proposed for, e.g., Isomap [de Silva and Tenenbaum, 2003; Law and Jain, 2006], MVU [Weinberger *et al.*, 2005, 2007], and Kernel PCA [Tipping, 2000].

2.4.3 Out-of-sample extension

An important requirement for dimensionality reduction techniques is the ability to embed new high-dimensional datapoints into an existing low-dimensional data representation. So-called out-of-sample extensions have been developed for a number of techniques to allow for the embedding of such new datapoints. They can be subdivided into parametric and nonparametric out-of-sample extensions.

In a parametric out-of-sample extension, the dimensionality reduction technique provides all parameters that are necessary in order to transform new data from the high-dimensional to the low-dimensional space (see Table 2.1 for an overview of parametric dimensionality reduction techniques). In linear techniques such as PCA, this transformation is defined by the linear mapping \mathbf{M} that was applied to the original data. For Kernel PCA, a similar transformation is available, although this transformation requires additional kernel function computations [Schölkopf *et al.*, 1998]. For autoencoders, the trained network defines the transformation from the high-dimensional to the low-dimensional data representation.

For the other nonlinear dimensionality reduction techniques, a parametric out-of-sample extension is not available, and therefore, a nonparametric out-of-sample extension is required. Nonparametric out-of-sample extensions perform an estimation of the transformation from the high-dimensional to the low-dimensional space. For instance, a nonparametric out-of-sample extension for Isomap, LLE, and Laplacian Eigenmaps has been presented by Bengio *et al.* [2004b], in which the techniques are redefined as kernel methods and the out-of-sample extension is performed using the Nyström approximation [Platt, 2005]. The Nyström approximation approximates the eigenvectors of a large $n \times n$ matrix based on the eigendecomposition of a smaller $m \times m$ submatrix of the large matrix. Similar nonparametric out-of-sample extensions for Isomap are proposed in [de Silva and Tenenbaum, 2003; Choi and Choi, 2007]. For MVU, an approximate out-of-sample extension has been proposed that is based on computing a linear

transformation from a set of landmark points to the complete dataset [Weinberger *et al.*, 2005]. An alternative out-of-sample extension for MVU finds this linear transformation by computing the eigenvectors corresponding to the smallest eigenvalues of the graph Laplacian (similar to Laplacian Eigenmaps) [Weinberger *et al.*, 2007]. A third out-of-sample extension for MVU approximates the kernel eigenfunction using Gaussian basis functions [Chin and Suter, 2008].

A nonparametric out-of-sample extension that can be applied to all nonlinear dimensionality reduction techniques is proposed by Li *et al.* [2005]. The technique finds the nearest neighbor of the new datapoint in the high-dimensional representation, and computes the linear mapping from the nearest neighbor to its corresponding low-dimensional representation. The low-dimensional representation of the new datapoint is found by applying the same linear mapping to this datapoint.

From the description above, we may observe that linear and nonlinear techniques for dimensionality reduction are quite similar in that they allow the embedding of new datapoints. However, for a number of nonlinear techniques, only nonparametric out-of-sample extensions are available, which leads to estimation errors in the embedding of new datapoints.

2.5 Experiments

In this section, a systematic empirical comparison of the performance of the linear and nonlinear techniques for dimensionality reduction is performed. We perform the comparison by measuring generalization errors in classification tasks on two types of datasets: (1) artificial datasets and (2) natural datasets. In addition to generalization errors, we measure the ‘trustworthiness’ of the low-dimensional embeddings as proposed by Venna and Kaski [2006].

The setup of our experiments is described in subsection 2.5.1. In subsection 2.5.2, the results of our experiments on five artificial datasets are presented. Subsection 2.5.3 presents the results of the experiments on five natural datasets.

2.5.1 Experimental setup

In our experiments on both the artificial and the natural datasets, we apply the thirteen techniques for dimensionality reduction on the high-dimensional representation of the data. Subsequently, we assess the quality of the resulting low-dimensional data representation by evaluating to what extent the local structure of the data is retained. The evaluation is performed by measuring the generalization errors of 1-nearest neighbor classifiers that are trained on the low-dimensional data representation. A similar evaluation scheme is employed by Sanguinetti [2008]. In addition, we measure the ‘trustworthiness’ that was proposed for the assessment of the quality of dimensionality reduction embeddings by Venna and Kaski [2006]. The trustworthiness measures the proportion of points that are too close together in the low-dimensional space. The trustworthiness measure is defined as

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^n \sum_{j \in U_i^{(k)}} (r(i, j) - k), \quad (2.28)$$

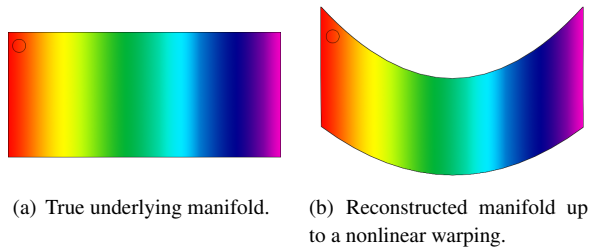


Figure 2.3 Two low-dimensional data representations.

where $r(i, j)$ represents the rank of the low-dimensional datapoint j according to the pairwise distances between the low-dimensional datapoints. The variable $U_i^{(k)}$ indicates the set of points that are among the k nearest neighbors in the low-dimensional space but not in the high-dimensional space. Both the generalization errors of the 1-nearest neighbor classifiers and the trustworthiness evaluate to what extent the local structure of the data is retained (the 1-nearest neighbor classifier does so because of its high variance). We opt for an evaluation of the local structure of the data, because for successful visualization or classification of data only its local structure needs to be retained. An evaluation of the quality based on generalization errors and trustworthiness has an important advantage over measuring reconstruction errors, because a high reconstruction error does not necessarily imply that the dimensionality reduction technique performed poorly. For instance, if a dimensionality reduction technique recovers the true underlying manifold in Figure 2.3(a) up to a nonlinear warping, such as in Figure 2.3(b), this leads to a high reconstruction error, whereas the local structure of the two manifolds is nearly identical (as the circles indicate). In other words, reconstruction errors measure the quality of the global structure of the low-dimensional data representation, and not the quality of the local structure. Moreover, for real-world datasets the true underlying manifold of the data is usually unknown, and as a result, reconstruction errors cannot be computed.

For all dimensionality reduction techniques except for Isomap, MVU, and sparse spectral techniques (the so-called manifold learners), we performed experiments without out-of-sample extension, because our main interest is in the performance of the dimensionality reduction techniques, and not in the quality of the out-of-sample extension. In the experiments with Isomap, MVU, and sparse spectral techniques, we employ out-of-sample extensions (see subsection 2.4.3) in order to embed datapoints that are not connected to the largest component of the neighborhood graph which is constructed by these techniques. The use of the out-of-sample extension of the manifold learners is necessary because the traditional implementations of Isomap, MVU, and sparse spectral techniques can only embed the points that comprise the largest component of the neighborhood graph.

The parameter settings employed in our experiments are listed in Table 2.2. Most parameters were optimized using an exhaustive grid search within a reasonable range, which is shown in Table 2.2. For two parameters (σ in diffusion maps and Laplacian Eigenmaps), we employed fixed values in order to restrict the computational requirements of our experiments. The value of k in the k -nearest neighbor classifiers was set to 1. We determined the target dimensionality in the experiments by means of the maximum likelihood intrinsic dimensionality estimator [Levina

and Bickel, 2004]. Note that for Hessian LLE and LTSA, the dimensionality of the actual low-dimensional data representation cannot be higher than the number of nearest neighbors that was used to construct the neighborhood graph. The generalization errors were obtained using leave-one-out validation.

<i>Technique</i>	<i>Parameter settings</i>
PCA	None
Isomap	$5 \leq k \leq 15$
Kernel PCA	$\kappa = (\mathbf{X}\mathbf{X}^T + 1)^5$
MVU	$5 \leq k \leq 15$
Diffusion maps	$10 \leq t \leq 100$ $\sigma = 1$
LLE	$5 \leq k \leq 15$
Laplacian Eigenmaps	$5 \leq k \leq 15$ $\sigma = 1$
Hessian LLE	$5 \leq k \leq 15$
LTSA	$5 \leq k \leq 15$
Sammon mapping	None
Autoencoders	Three hidden layers
LLC	$5 \leq k \leq 15$ $5 \leq m \leq 25$
Manifold charting	$5 \leq m \leq 25$

Table 2.2 Parameter settings for the experiments.

Five artificial datasets

We performed experiments on five artificial datasets, most of which are often used in the manifold learning literature (see, e.g., Roweis and Saul [2000]; Tenenbaum *et al.* [2000]). The datasets were specifically selected to investigate how the dimensionality reduction techniques deal with: (i) data that lies on a low-dimensional manifold that is isometric to the Euclidean space, (ii) data lying on a low-dimensional manifold that is not isometric to the Euclidean space, (iii) data that lies on or near a discontinuous manifold, and (iv) data forming a manifold with a high intrinsic dimensionality. The artificial datasets on which we performed experiments are: the Swiss roll dataset (addressing i), the helix dataset (addressing ii), the twin peaks dataset (addressing ii), the broken Swiss roll dataset (addressing iii), and the high-dimensional (HD) dataset (addressing iv). Figure 2.4 shows plots of the first four artificial datasets. The HD dataset consists of points randomly sampled from a 5-dimensional non-linear manifold embedded in a 10-dimensional space. In order to ensure that the generalization errors of the k -nearest neighbor classifiers reflect the quality of the data representations produced by the dimensionality reduction techniques, we assigned all datapoints to one of two classes according to a checkerboard pattern on the manifold. All artificial datasets consist of 5,000 samples. We opted for a fixed number of datapoints in each dataset, because in real-world applications, obtaining more training data is usually expensive.

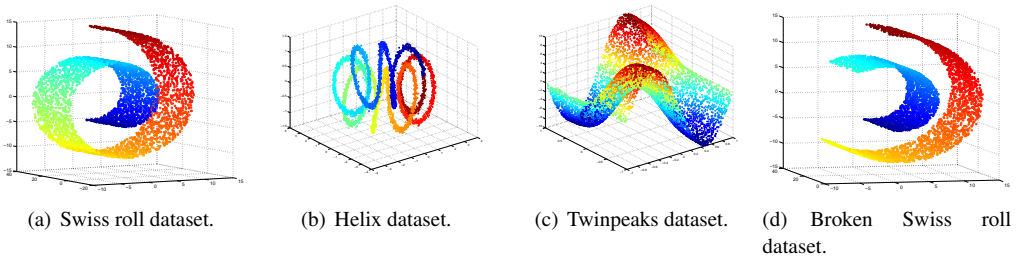


Figure 2.4 Four of the artificial datasets.

Dataset (d)	None	PCA	Isom.	KPCA	MVU	DM	LLE	LEM	HLLC	LTSA	SM	AE	LLC	MC
Swiss roll (2D)	3.68%	30.32%	2.94%	28.60%	5.90%	28.76%	7.32%	22.74%	3.38%	3.42%	24.44%	48.38%	20.86%	13.58%
Helix (1D)	1.24%	30.40%	6.18%	42.60%	3.86%	35.08%	26.72%	12.24%	52.22%	0.86%	52.22%	32.14%	29.22%	21.94%
Twinpeaks (2D)	0.40%	0.28%	0.40%	0.00%	0.50%	0.12%	1.16%	0.74%	0.12%	0.00%	0.22%	0.22%	8.66%	0.54%
Broken Swiss (2D)	2.14%	26.92%	14.43%	30.62%	30.84%	23.76%	40.86%	11.88%	4.64%	2.46%	28.00%	29.74%	39.38%	22.78%
HD (5D)	24.19%	21.73%	21.15%	27.87%	26.30%	38.30%	26.62%	42.34%	50.02%	40.93%	21.50%	32.03%	34.69%	20.06%

Table 2.3 Generalization errors of 1-NN classifiers trained on artificial datasets.

Five natural datasets

For our experiments on natural datasets, we selected five datasets that represent tasks from a variety of domains: (1) the MNIST dataset, (2) the COIL-20 dataset, (3) the NiSIS dataset, (4) the ORL dataset, and (5) the HIVA dataset. The MNIST dataset is a dataset of 60,000 handwritten digits. For computational reasons, we randomly selected 10,000 digits for our experiments. The images in the MNIST dataset have 28×28 pixels, and can thus be considered as points in a 784-dimensional space. The COIL-20 dataset contains images of 20 different objects, depicted from 72 viewpoints, leading to a total of 1,440 images. The size of the images is 32×32 pixels, yielding a 1,024-dimensional space. The NiSIS dataset is a publicly available dataset for pedestrian detection, which consists of 3,675 grayscale images of size 36×18 pixels (leading to a space of dimensionality 648). The ORL dataset is a face recognition dataset that contains 400 grayscale images of 112×92 pixels that depict 40 faces under various conditions (i.e., the dataset contains 10 images per face). The HIVA dataset is a drug discovery dataset with two classes. It consists of 3,845 datapoints with dimensionality 1,617.

Dataset (d)	None	PCA	Isom.	KPCA	MVU	DM	LLE	LEM	HLLC	LTSA	SM	AE	LLC	MC
Swiss roll (2D)	—	0.88	1.00	0.89	1.00	0.89	0.97	0.93	1.00	1.00	0.90	0.72	0.84	0.88
Helix (1D)	—	0.78	0.96	0.74	0.94	0.76	0.86	0.96	0.35	1.00	0.35	0.78	0.79	0.83
Twinpeaks (2D)	—	0.98	1.00	0.97	0.99	0.96	0.99	0.99	0.99	0.99	1.00	0.95	0.87	0.99
Broken Swiss (2D)	—	0.96	0.98	0.96	0.97	0.96	0.95	0.97	0.94	0.95	0.97	0.96	0.84	0.96
HD (5D)	—	1.00	0.99	1.00	0.98	0.99	0.99	0.92	0.26	0.95	1.00	0.92	0.88	1.00

Table 2.4 Trustworthinesses $T(12)$ on the artificial datasets.

2.5.2 Experiments on artificial datasets

In Table 2.3, we present the generalization errors of 1-nearest neighbor classifiers that were trained and tested on the low-dimensional data representations obtained from the dimensionality reduction techniques. We ran the experiments for all parameter settings described in Table 2.2, and for each technique, we report the best generalization error of all runs in Table 2.3. In the table, the left column indicates the name of the dataset and the target dimensionality to which we attempted to transform the high-dimensional data. The best performing technique for each dataset is shown in boldface. Table 2.4 presents the corresponding trustworthiness values of the low-dimensional embeddings (again, only the best trustworthiness of all runs is reported). From the results in Table 2.3 and Table 2.4, we make four observations.

First, the results reveal the strong performance of techniques based on neighborhood graphs (Isomap, MVU, LLE, Laplacian Eigenmaps, Hessian LLE, and LTSA) on artificial datasets such as the Swiss roll dataset. Of the techniques based on neighborhood graphs, Isomap slightly outperforms the other five techniques. Within the sparse spectral techniques, LTSA seems to be the best-performing technique. Techniques that do not employ neighborhood graphs (viz. PCA, diffusion maps, Kernel PCA, Sammon mapping, and autoencoders) perform poorly on artificial datasets such as the Swiss roll dataset. The performance of the two techniques that align local linear models (LLC and manifold charting) on the Swiss roll dataset are comparable to those of techniques that do not employ neighborhood graphs.

Second, from the results of the experiments on the helix dataset, we observe that Hessian LLE, a technique that performs strong on the Swiss roll dataset, may perform less well on manifolds that are not isometric to the Euclidean space. The performance of LLE on the helix dataset is also notably worse than its performance on the Swiss roll dataset. The other techniques based on neighborhood graphs (Isomap, MVU, LLE, Laplacian Eigenmaps, and LTSA) perform strong on the helix dataset, despite the non-isometric nature of the dataset.

Third, the high generalization errors on the broken Swiss roll dataset indicate that most nonlinear techniques for dimensionality reduction do not perform well under the presence of disconnected (i.e., non-smooth) manifolds in the data.

Fourth, from the results on the HD dataset, we observe that nonlinear techniques may have problems when they are faced with a dataset with a high intrinsic dimensionality. In particular, Hessian LLE performs disappointingly on the dataset with a high intrinsic dimensionality. On the HD dataset, the performance of PCA is surprisingly strong: it is only outperformed by Sammon mapping and manifold charting, which is the best performing technique on this dataset.

Taken together, the results show that manifold learners perform well on data that forms a low-dimensional manifold. However, the results also reveal that the strong performance on, e.g., the Swiss roll dataset does not always generalize to more complex datasets, such as datasets with disconnected manifolds, manifolds that are non-isometric to the Euclidean space, or manifolds with a high intrinsic dimensionality.

2.5.3 Experiments on natural datasets

Table 2.5 presents the generalization errors of 1-nearest neighbor classifiers that were trained on the low-dimensional data representations obtained from the dimensionality reduction techniques.

Dataset (d)	None	PCA	Isomap	KPCA	MVU	DM	LLE	LEM	HLLE	LTSA	SM	AE	LLC	MC
MNIST (20D)	5.11%	5.70%	12.36%	11.90%	12.12%	27.90%	9.99%	14.84%	69.54%	90.10%	65.32%	8.58%	16.70%	11.16%
COIL-20 (5D)	0.14%	3.82%	14.51%	7.78%	25.14%	4.51%	22.29%	95.00%	49.10%	5.63%	1.11%	15.83%	5.07%	38.26%
ORL (8D)	2.50%	4.75%	27.25%	6.25%	24.25%	49.00%	11.00%	97.50%	56.00%	12.75%	2.75%	6.25%	15.00%	60.05%
NiSIS (15D)	8.24%	7.95%	13.91%	9.50%	16.05%	22.97%	17.77%	47.59%	48.98%	24.74%	48.98%	8.57%	23.55%	19.02%
HIVA (15D)	4.63%	5.44%	4.81%	5.07%	4.81%	5.15%	4.89%	4.81%	3.51%	3.51%	3.51%	4.97%	3.51%	4.55%

Table 2.5 Generalization errors of 1-NN classifiers trained on natural datasets.

Dataset (d)	None	PCA	Isomap	KPCA	MVU	DM	LLE	LEM	HLLE	LTSA	SM	AE	LLC	MC
MNIST (20D)	–	1.00	0.98	0.99	0.94	0.93	0.96	0.89	0.64	0.56	0.78	1.00	0.93	0.98
COIL-20 (5D)	–	0.99	0.93	0.98	0.91	0.99	0.93	0.27	0.69	0.96	0.99	0.97	0.97	0.88
ORL (8D)	–	0.99	0.94	0.98	0.95	0.84	0.95	0.29	0.85	0.93	0.99	0.99	0.80	0.77
NiSIS (15D)	–	0.99	0.93	0.99	0.91	0.89	0.90	0.47	0.47	0.82	0.47	0.99	0.85	0.89
HIVA (15D)	–	0.97	0.94	0.89	0.84	0.76	0.80	0.78	0.42	0.54	0.42	0.99	0.91	0.95

Table 2.6 Trustworthinesses $T(12)$ on the natural datasets.

Table 2.6 presents the corresponding trustworthinesses. From the results in Table 2.5 and 2.6, we make two observations.

First, we observe that the performance of manifold learners on the natural datasets is disappointing compared to the performance of these techniques on the artificial datasets. In contrast, many techniques that do not employ neighborhood graphs such as PCA, Kernel PCA, Sammon mapping, and autoencoders perform well on (most of) the natural datasets. In particular, PCA and autoencoders outperform the other techniques on four of the five datasets (when the techniques are assessed based on the trustworthiness of their embeddings). On the COIL-20 dataset, the performance of autoencoders is slightly less strong, most likely due to the small number of instances that constitute this dataset, which hampers the successful training of the large number of weights in the network. Globally, the difference between the results of the experiments on the artificial and the natural datasets is remarkable: techniques that perform well on artificial datasets perform poorly on natural datasets, and vice versa.

Second, the results show that on some natural datasets, the classification performance of our classifiers was not improved by performing dimensionality reduction. Most likely, this is due to errors in the intrinsic dimensionality estimator we employed. As a result, the target dimensionalities may not be optimal (in the sense that they minimize the generalization error of the trained classifier). However, since we aim to compare the performance of dimensionality reduction techniques, and not to minimize generalization errors on classification problems, this observation is of no relevance to our study.

2.6 Discussion

In the previous sections, we presented a comparative study of techniques for dimensionality reduction. We observed that most nonlinear techniques do not outperform PCA on natural datasets, despite their ability to learn the structure of complex nonlinear manifolds. This section discusses the main weaknesses of current nonlinear techniques for dimensionality reduction that explain the results of our experiments. In addition, the section presents ideas on how to overcome these weaknesses. The discussion is subdivided into four parts. Subsection 2.6.1 discusses the main weaknesses of full spectral dimensionality reduction techniques. In subsection 2.6.2, we address

five weaknesses of sparse spectral techniques for dimensionality reduction. Subsection 2.6.3 discusses the main weaknesses of the non-convex dimensionality reduction techniques. Subsection 2.6.4 summarizes the main weaknesses of current nonlinear techniques for dimensionality reduction and presents some concluding remarks on the future development of dimensionality reduction techniques.

2.6.1 Full spectral techniques

Our discussion on the results of full spectral techniques for dimensionality reduction is subdivided into two parts. First, we discuss the results of the two neighborhood graph-based techniques, Isomap and MVU. Second, we discuss weaknesses explaining the results of the two kernel-based techniques, Kernel PCA and diffusion maps.

For the first part, we remark that full spectral techniques for dimensionality reduction that employ neighborhood graphs, such as Isomap and MVU, are subject to many of the weaknesses of sparse spectral techniques that we will discuss in subsection 2.6.2. In particular, the construction of the neighborhood graph is susceptible to the curse of dimensionality, overfitting, and the presence of outliers (see 2.6.2 for a detailed explanation). In addition to these three problems, Isomap suffers from short-circuiting: a single erroneous connection in the neighborhood graph may severely affect the pairwise geodesic distances, as a result of which the data is poorly embedded in the low-dimensional space. Moreover, Isomap uses classical scaling to construct a low-dimensional embedding from the pairwise geodesic distances. The cost function of classical scaling causes Isomap to focus on retaining the large geodesic distances, instead of on the small geodesic distance that constitute the local structure of the data. A possible solution to this problem is presented by Yang [2004]. MVU suffers from a similar problem as Isomap: a single short-circuit in the neighborhood graph may lead to an erroneous constraint in the semidefinite program that severely affects the performance of MVU.

For the second part, we remark that kernel-based techniques for dimensionality reduction (i.e., Kernel PCA and diffusion maps) do not suffer from the weaknesses of neighborhood graph-based techniques. However, the performance of Kernel PCA and diffusion maps on the Swiss roll dataset indicates that (similar to PCA) these techniques are incapable of modeling complex nonlinear manifolds. The main reason for this incapability is that kernel-based methods require the selection of a proper kernel function. In general, model selection in kernel methods is performed using some form of hold-out testing [Golub *et al.*, 1979], leading to high computational costs. Alternative approaches to model selection for kernel methods are based on, e.g., maximizing the between-class margins or the data variance using semidefinite programming (as in MVU) [Graepel, 2002; Lanckriet *et al.*, 2004]. Despite these alternative approaches, the construction of a proper kernel remains an important obstacle for the successful application of Kernel PCA. In addition, depending on the selection of the kernel, kernel-based techniques for dimensionality reduction may suffer from similar weaknesses as other manifold learners. In particular, when a Gaussian kernel with a small value of σ is employed, Kernel PCA and diffusion maps may be susceptible to the curse of intrinsic dimensionality (see 2.6.2). Diffusion maps largely resolve the short-circuiting problems of Isomap by integrating over all paths through a graph defined of the data, however, they are still subject to the second problem of Isomap: diffusion maps focus on

retaining large diffusion distances in the low-dimensional embedding, instead of on retaining the small diffusion distances that constitute the local structure of the data.

2.6.2 Sparse spectral techniques

The results of our experiments show that the performance of the popular sparse spectral techniques, such as LLE, is rather disappointing on many real-world datasets. Most likely, the poor performance of these techniques is due to one or more of the following five weaknesses.

First, sparse spectral techniques for dimensionality reduction suffer from a fundamental weakness in their cost function. For instance, the optimal solution of the cost function of LLE (see Equation 2.13) is the trivial solution in which the coordinates of all low-dimensional points \mathbf{y}_i are zero. This solution is not selected because LLE has a constraint on the covariance of the solution, viz., the constraint $\|\mathbf{y}^{(k)}\|^2 = 1$ for $\forall k$. Although the covariance constraint may seem to have resolved the problem of selecting a trivial solution, it is easy to ‘cheat’ on it. In particular, LLE often constructs solutions in which most points are embedded on the origin, and there are a few ‘strings’ coming out of the origin that make sure that the covariance constraint is met (at a relatively small cost). Moreover, the simple form of the covariance constraint in LLE may give rise to undesired rescalings of the manifold [Goldberg *et al.*, 2008]. The same problems also apply to Laplacian Eigenmaps, Hessian LLE, and LTSA, which have similar covariance constraints.

Second, all sparse spectral dimensionality reduction techniques suffer from the curse of dimensionality of the embedded manifold (i.e., the intrinsic dimension of the data) [Bengio and Monperrus, 2004; Weinberger *et al.*, 2005; Bengio and LeCun, 2007], because the number of datapoints that is required to characterize a manifold properly grows exponentially with the intrinsic dimensionality of the manifold. The susceptibility to the curse of dimensionality is a fundamental weakness of all local learners, and therefore, it also applies to learning techniques that employ Gaussian kernels (such as Support Vector Machines). For artificial datasets with low intrinsic dimensionality such as the Swiss roll dataset, this weakness does not apply. However, in most real-world tasks, the intrinsic dimensionality of the data is much higher. For instance, the face space is estimated to consist of at least 100 dimensions [Meytlis and Sirovich, 2007]. As a result, the performance of local techniques is poor on many real-world datasets, which is illustrated by the results of our experiments with the natural datasets.

Third, the inferior performance of sparse spectral techniques for dimensionality reduction arises from the eigenproblems that the techniques attempt to solve. Typically, the smallest eigenvalues in these problems are very small (around 10^{-7} or smaller), whereas the largest eigenvalues are fairly big (around 10^2 or larger). Eigenproblems with these properties are extremely hard to solve, even for state-of-the-art eigensolvers. The eigensolver may not be able to identify the smallest eigenvalues of the eigenproblem, and as a result, the dimensionality reduction technique might produce suboptimal solutions. The good performance of Isomap and MVU (that search for the largest eigenvalues) compared to sparse spectral techniques (that search for the smallest eigenvalues) may be explained by the difficulty of solving eigenproblems.

Fourth, local properties of a manifold do not necessarily follow the global structure of the manifold (as noted in, e.g., [Roweis *et al.*, 2001; Brand, 2004]) in the presence of noise around the manifold. In other words, sparse spectral techniques suffer from overfitting on the manifold.

Moreover, sparse spectral techniques suffer from folding [Brand, 2002]. Folding is caused by a value of k that is too high with respect to the sampling density of (parts of) the manifold. Folding causes the local linearity assumption to be violated, leading to radial or other distortions. In real-world datasets, folding is likely to occur because the data density may vary over the manifold (i.e., because the data distribution is not uniform over the manifold). An approach that might overcome this weakness for datasets with small intrinsic dimensionality is adaptive neighborhood selection. Techniques for adaptive neighborhood selection are presented in, e.g., [Wang *et al.*, 2005; Mekuz and Tsotsos, 2006; Samko *et al.*, 2006]. Furthermore, sparse spectral techniques for dimensionality reduction are sensitive to the presence of outliers in the data [Chang and Ghosh, 1998]. In local techniques for dimensionality reduction, outliers are connected to their k nearest neighbors, even when they are very distant. As a consequence, outliers degrade the performance of local techniques for dimensionality reduction. A possible approach to resolve this problem is the usage of an ϵ -neighborhood. In an ϵ -neighborhood, datapoints are connected to all datapoints that lie within a sphere with radius ϵ . A second approach to overcome the problem of outliers is preprocessing the data by removing outliers [Zhang and Zha, 2003; Park *et al.*, 2004].

Fifth, the local linearity assumption of sparse spectral techniques for dimensionality reduction implies that the techniques assume that the manifold contains no discontinuities (i.e., that the manifold is smooth). The results of our experiments with the broken Swiss dataset illustrate the incapability of sparse spectral dimensionality reduction techniques to model non-smooth manifolds. In real-world datasets, the underlying manifold is not likely to be smooth. For instance, a dataset that contains different object classes is likely to constitute a disconnected underlying manifold. In addition, most sparse spectral techniques cannot deal with manifolds that are not isometric to the Euclidean space, which is illustrated by the results of our experiments with the helix and twinpeaks datasets. This may be a problem, because for instance, a dataset of objects depicted under various orientations gives rise to a manifold that is closed (similar to the helix dataset).

In addition to these five weaknesses, Hessian LLE and LTSA cannot transform data to a dimensionality higher than the number of nearest neighbors in the neighborhood graph, which might lead to difficulties with datasets with a high intrinsic dimensionality.

2.6.3 Non-convex techniques

Obviously, the main problem of non-convex techniques is that they optimize non-convex objective functions, as a result of which they suffer from the presence of local optima in the objective functions. For instance, the EM algorithm that is employed in LLC and manifold charting is likely to get stuck in a local maximum of the log-likelihood function. In addition, LLC and manifold charting are hampered by the presence of outliers in the data. In techniques that perform global alignment of linear models (such as LLC), the sensitivity to the presence of outliers may be addressed by replacing the mixture of factor analyzers by a mixture of t-distributed subspaces (MoTS) model [de Ridder and Franc, 2003b,a]. The intuition behind the use of the MoTS model is that a t-distribution is less sensitive to outliers than a Gaussian (which tends to overestimate variances) because it has heavier tails.

For autoencoders, the presence of local optima in the objective function has largely been overcome by the pretraining of the network using RBMs. A limitation of autoencoders is that

they are only applicable on datasets of reasonable dimensionality. If the dimensionality of the dataset is high, the number of weights in the network is too large to find an appropriate setting of the network. This limitation of autoencoders may be addressed by preprocessing the data using PCA. Moreover, successful training of autoencoders requires the availability of sufficient amounts of data, as illustrated by our results with autoencoders on the COIL-20 dataset.

Despite the problems of the non-convex techniques mentioned above, our results show that convex techniques for dimensionality reduction do not necessarily outperform non-convex techniques for dimensionality reduction. In particular, multilayer autoencoders perform very well on all five natural datasets. Most likely, these results are due to the larger freedom in designing non-convex techniques, allowing the incorporation of procedures that circumvent many of the problems of (both full and sparse) spectral techniques mentioned above. In particular, multilayer autoencoders provide a deep architecture (i.e., an architecture with multiple nonlinear layers), as opposed to shallow architectures such of the convex techniques that we discussed [Bengio and LeCun, 2007]. The main advantage of such a deep architecture is that it requires exponentially less datapoints to learn the structure of highly varying manifolds, as illustrated for a d -bits parity dataset by Bengio [2007]. Hence, although convex techniques are much more popular in dimensionality reduction (and in machine learning in general), our results suggest that suboptimally optimizing a sensible objective function is a more viable approach than optimizing a convex objective function that contains obvious flaws. This claim is supported by the strong results of t-SNE, a non-convex multidimensional scaling variant, that we present in Chapter 3.

2.6.4 Main weaknesses

When collecting all the above observations, there is sufficient ground to state that the results of our experiments indicate that nonlinear techniques for dimensionality reduction do not yet clearly outperform the traditional PCA. This result agrees with the results of studies reported in the literature. On selected datasets, nonlinear techniques for dimensionality reduction outperform linear techniques [Niskanen and Silvén, 2003; Teng *et al.*, 2005], but nonlinear techniques perform poorly on various other natural datasets [Graf and Wichmann, 2002; Jenkins and Mataric, 2002; Hughes and Tarassenko, 2003; Lim *et al.*, 2003]. In particular, our results establish three main weaknesses of the popular sparse spectral techniques for dimensionality reduction: (1) flaws in their objective functions, (2) numerical problems in their eigendecompositions, and (3) their susceptibility to the curse of dimensionality. Some of these weaknesses also apply to Isomap and MVU.

From the first weakness, we may infer that a requirement for future dimensionality reduction techniques is that the minimum of the cost function is a non-trivial solution, even if this may prompt the use of a non-convex objective function. In the design of a non-convex technique, there is much more freedom to construct a sensible objective function that is not hampered by obvious flaws. The strong results of autoencoders support this claim, as well as the results we will present for t-SNE in Chapter 3.

The second weakness leads to exactly the same suggestion, but for a different reason: convex objective functions are often hard to optimize as well. In particular, sparse eigendecompositions are subject to numerical problems because it is hard to distinguish the smallest eigenvalues from the trivial zero eigenvalue. Moreover, interior point methods such as those employed to solve

the SDP in MVU require the computation of the Hessian, which may be prohibiting successful optimization for computational reasons (on medium-sized or large datasets, MVU can only be performed using a variety of approximations that result in suboptimal solutions).

From the third weakness, we may infer that a requirement for future techniques for dimensionality reduction is that they do not rely completely on local properties of the data. It has been suggested that the susceptibility to the curse of dimensionality may be addressed using techniques in which the global structure of the data manifold is represented in a number of linear models [Bengio and Monperrus, 2004]. However, the performance of LLC and manifold charting in our experiments is not sufficiently well to support this suggestion. The strong performance of autoencoders does suggest that it is beneficial to use deep architectures that contain more than one layer of nonlinearity.

2.7 Chapter conclusions

The chapter presented a review and comparative study of techniques for dimensionality reduction. From the results obtained, we may conclude that nonlinear techniques for dimensionality reduction are, despite their large variance, not yet capable of outperforming traditional PCA. In the future, we foresee the development of new nonlinear techniques for dimensionality reduction that (i) do not suffer from trivial optimal solutions, (ii) may be based on non-convex objective functions, and (iii) do not rely on neighborhood graphs to model the (local) structure of the data manifold. The other important concern in the development of novel techniques for dimensionality reduction is their optimization, which should be computationally and numerically feasible in practice.

3 t-Distributed Stochastic Neighbor Embedding

Contents

In the previous chapter we observed that, although dimensionality reduction may form a good approach to address the dimensionality problem of image-space representations, state-of-the-art techniques for dimensionality reduction are hampered by fundamental limitations that are often related to their convex nature. Motivated by our observations in the previous chapter, the chapter develops a new technique for dimensionality reduction, called t-Distributed Stochastic Neighbor Embedding (t-SNE), that aims to address the limitations of current state-of-the-art dimensionality reduction techniques. The results obtained with t-SNE provide novel insights into the answer to research question RQ1.

Based on

L.J.P. van der Maaten and G.E. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9(Nov):2431–2456, 2008.

Outline

Section 3.1 presents Stochastic Neighbor Embedding (SNE), which is a variant of multidimensional scaling that we use as a basis for our new technique. In Section 3.2, we present the new technique for dimensionality reduction called ‘t-Distributed Stochastic Neighbor Embedding’ or ‘t-SNE’, and explain why it is better than the original SNE technique. Section 3.3 presents our experiments with t-SNE, which reveal that the new technique outperforms existing techniques in the visualization of real-world data. Section 3.4 presents an extension of t-SNE that allows it to be performed on datasets that contain large numbers of datapoints, and applies t-SNE with the extension on a dataset of 60,000 datapoints. Section 3.5 discusses the results of our experiments, and explains why t-SNE outperforms the dimensionality reduction techniques that were investigated in Chapter 2. Section 3.6 concludes the chapter.

In the previous chapter, we compared current state-of-the-art techniques for dimensionality reduction. Our experiments revealed that in general, nonlinear techniques do not outperform a traditional linear dimensionality reduction technique such as PCA [Hotelling, 1933]. We also noted that PCA is identical to classical multidimensional scaling [Torgerson, 1952], as a result of which it focuses on keeping the low-dimensional representations of dissimilar datapoints far apart. For high-dimensional data that lies on or near a low-dimensional, non-linear manifold it is usually more important to keep the low-dimensional representations of similar datapoints close together, which is typically not possible with a linear mapping. This idea forms the basis of local dimensionality reduction techniques such as LLE and Laplacian Eigenmaps, however, for reasons discussed in subsection 2.6.2, these techniques do not work well.

In this chapter, we focus on reducing the dimensionality of data to only two dimensions, in order to facilitate visualization of the high-dimensional data¹. Not surprisingly, the nonlinear dimensionality reduction techniques we discussed in Chapter 2 are often not successful at visualizing high-dimensional data. In particular, most of the techniques are not capable of retaining both the local and the global structure of the data in a single low-dimensional map. For instance, a recent study reveals that even a semi-supervised variant of MVU is not capable of separating handwritten digits into their natural clusters [Song *et al.*, 2007].

The chapter describes a way of converting a high-dimensional dataset into a matrix of pairwise similarities and it introduces a new technique for dimensionality reduction called ‘t-SNE’. t-SNE is capable of capturing much of the local structure of high-dimensional data very well, while also revealing global structure such as the presence of clusters at several scales. We illustrate the strong performance of t-SNE by comparing it to PCA, Isomap, and LLE (the three most popular techniques for dimensionality reduction) on five datasets from a variety of domains. The maps that we present in the chapter are sufficient to demonstrate the superiority of t-SNE.

The outline of the chapter is as follows. In Section 3.1, we outline SNE as presented by [Hinton and Roweis, 2002], which forms the basis for t-SNE. In Section 3.2, we present t-SNE, which has two important differences when compared to SNE. In Section 3.3, we describe the experimental setup and the results of our experiments. Subsequently, Section 3.4 shows how t-SNE can be modified to visualize real-world datasets that contain many more than 10,000 datapoints. The results of our experiments are discussed in more detail in Section 3.5. The conclusions of the chapter are presented in Section 3.6.

3.1 Stochastic Neighbor Embedding

Stochastic Neighbor Embedding (SNE) models the similarity of datapoint \mathbf{x}_j to datapoint \mathbf{x}_i as the conditional probability, $p_{j|i}$, that \mathbf{x}_i would pick \mathbf{x}_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at \mathbf{x}_i . For nearby datapoints, $p_{j|i}$ is relatively high, whereas for widely separated datapoints, $p_{j|i}$ will be almost infinitesimal (for reasonable values of the variance of the Gaussian, σ_i). Mathematically, the conditional

¹In this chapter, we refer to the two-dimensional representations of the data as *map points*, and to the complete low-dimensional data representation $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$ as a *map*.

probability $p_{j|i}$ is given by

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}, \quad (3.1)$$

where σ_i is the variance of the Gaussian that is centered on datapoint \mathbf{x}_i . The method for determining the value of σ_i is presented later in this section. Because we are only interested in modeling pairwise similarities, we set the value of $p_{i|i}$ to zero. For the low-dimensional counterparts \mathbf{y}_i and \mathbf{y}_j of the high-dimensional datapoints \mathbf{x}_i and \mathbf{x}_j , it is possible to compute a similar conditional probability, which we denote by $q_{j|i}$. We set² the variance of the Gaussian that is employed in the computation of the conditional probabilities $q_{j|i}$ to $\frac{1}{\sqrt{2}}$. Hence, we model the similarity of map point \mathbf{y}_j to map point \mathbf{y}_i by

$$q_{j|i} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_i - \mathbf{y}_k\|^2)}. \quad (3.2)$$

Again, since we are only interested in modeling pairwise similarities, we set $q_{i|i} = 0$.

If the map points \mathbf{y}_i and \mathbf{y}_j correctly model the similarity between the high-dimensional datapoints \mathbf{x}_i and \mathbf{x}_j , the conditional probabilities $p_{j|i}$ and $q_{j|i}$ will be equal. Motivated by this observation, SNE aims to find a low-dimensional data representation that minimizes the mismatch between $p_{j|i}$ and $q_{j|i}$. A natural measure of the faithfulness with which $q_{j|i}$ models $p_{j|i}$ is the Kullback-Leibler divergence. SNE minimizes the sum of Kullback-Leibler divergences over all datapoints using a gradient descent method. The cost function C is given by

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}, \quad (3.3)$$

in which P_i represents the conditional probability distribution over all other datapoints given datapoint \mathbf{x}_i , and Q_i represents the conditional probability distribution over all other map points given map point \mathbf{y}_i . Because the Kullback-Leibler divergence is not symmetric, different types of error in the pairwise distances in the low-dimensional map are not weighted equally. In particular, there is a large cost for using widely separated map points to represent nearby datapoints (i.e., for using a small $q_{j|i}$ to model a large $p_{j|i}$), but there is only a small cost for using nearby map points to represent widely separated datapoints. This small cost comes from wasting some of the probability mass in the relevant Q distributions. In other words, the SNE cost function focuses on retaining the local structure of the data in the map (for reasonable values of the variance of the Gaussian in the high-dimensional space, σ_i).

The remaining parameter to be selected is the variance σ_i of the Gaussian that is centered over each high-dimensional datapoint, \mathbf{x}_i . It is not likely that there is a single value of σ_i that is optimal for all datapoints in the dataset because the density of the data is likely to vary. In dense regions, a smaller value of σ_i is usually more appropriate than in sparser regions. Any particular value of σ_i induces a probability distribution, P_i , over all of the other datapoints. This distribution has an entropy which increases as σ_i increases. SNE performs a binary search for the value of σ_i that produces a P_i with a fixed perplexity that is specified by the user³. The perplexity

²Setting the variance to another value only results in a rescaled version of the final map. Note that by using the same variance for every datapoint in the low-dimensional map, we lose the property that the data is a perfect model of itself if we embed it in a space of the same dimensionality.

³Note that the perplexity increases monotonically with the variance σ_i .

is defined as

$$\text{Perp}(P_i) = 2^{H(P_i)}, \quad (3.4)$$

where $H(P_i)$ is the Shannon entropy of P_i measured in bits

$$H(P_i) = - \sum_j p_{j|i} \log_2 p_{j|i}. \quad (3.5)$$

The perplexity can be interpreted as a smooth measure of the effective number of neighbors. The performance of SNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50.

The minimization of the cost function in Equation 3.3 is performed using a gradient descent method. The gradient has a surprisingly simple form

$$\frac{\delta C}{\delta \mathbf{y}_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(\mathbf{y}_i - \mathbf{y}_j). \quad (3.6)$$

Physically, the gradient may be interpreted as the resultant force created by a set of springs between the map point \mathbf{y}_i and all other map points \mathbf{y}_j . All springs exert a force along the direction $(\mathbf{y}_i - \mathbf{y}_j)$. The spring between \mathbf{y}_i and \mathbf{y}_j repels or attracts the map points depending on whether the distance between the two in the map is too small or too large to represent the similarities between the two high-dimensional datapoints. The force exerted by the spring between \mathbf{y}_i and \mathbf{y}_j is proportional to its length, and also proportional to its stiffness, which is the mismatch $(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})$ between the pairwise similarities of the data points and the map points.

The gradient descent is initialized by sampling map points randomly from an isotropic Gaussian with small variance that is centered around the origin. In order to speed up the optimization and to avoid poor local minima, a relatively large momentum term is added to the gradient. In other words, the current gradient is added to an exponentially decaying sum of previous gradients in order to determine the changes in the coordinates of the map points at each iteration of the gradient search. In addition, in the early stages of the optimization, Gaussian noise is added to the map points after each iteration. Gradually reducing the variance of this noise performs a type of simulated annealing that helps the optimization to escape from poor local minima in the cost function. If the variance of the noise changes very slowly at the critical point at which the global structure of the map starts to form, SNE tends to find maps with a better global organization. However, this requires sensible choices of the initial amount of Gaussian noise and the rate at which it decays. Moreover, these choices interact with the amount of momentum and the step size that are employed in the gradient descent. It is therefore common to run the optimization several times on a dataset to find appropriate values for the parameters⁴. In this respect, SNE is inferior to methods that allow convex optimization and it would be useful to find an optimization method that gives good results without requiring the extra computation time and parameter choices introduced by the simulated annealing.

⁴Picking the best map after several runs as a visualization of the data is not nearly as problematic as picking the model that does best on a test set during supervised learning. In visualization, the aim is to see the structure in the training data, not to generalize to held out test data.

3.2 *t*-Distributed Stochastic Neighbor Embedding

Section 3.1 discussed SNE as it was presented by Hinton and Roweis [2002]. Although SNE constructs reasonably good visualizations, it is hampered by a cost function that is difficult to optimize and by a problem we refer to as the ‘crowding problem’. In this section, we present a new technique called ‘*t*-Distributed Stochastic Neighbor Embedding’ or ‘*t*-SNE’ that aims to alleviate these problems. The cost function used by *t*-SNE differs from the one used by SNE in two ways: (1) it uses a symmetrized version of the SNE cost function with simpler gradients that was briefly introduced by Cook *et al.* [2007] and (2) it uses a Student-*t* distribution rather than a Gaussian to compute the similarity between two points *in the low-dimensional space*. In effect, *t*-SNE employs a heavy-tailed distribution in the low-dimensional space to alleviate both the crowding problem and the optimization problems of SNE.

In this section, we first discuss the symmetric version of SNE (subsection 3.2.1). Subsequently, we discuss the crowding problem (subsection 3.2.2), and the use of heavy-tailed distributions to address this problem (subsection 3.2.3). We conclude the section by describing our approach to the optimization of the *t*-SNE cost function (subsection 3.2.4).

3.2.1 Symmetric SNE

As an alternative to minimizing the sum of the Kullback-Leibler divergences between the conditional probabilities $p_{j|i}$ and $q_{j|i}$, it is also possible to minimize a single Kullback-Leibler divergence between a joint probability distribution, P , in the high-dimensional space and a joint probability distribution, Q , in the low-dimensional space

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \quad (3.7)$$

where again, we set p_{ii} and q_{ii} to zero. We refer to this type of SNE as symmetric SNE, because it has the property that $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$ for $\forall i, j$. In symmetric SNE, the pairwise similarities in the low-dimensional map q_{ij} are given by

$$q_{ij} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq l} \exp(-\|\mathbf{y}_k - \mathbf{y}_l\|^2)}. \quad (3.8)$$

The obvious way to define the pairwise similarities in the high-dimensional space p_{ij} is

$$p_{ij} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)}{\sum_{k \neq l} \exp(-\|\mathbf{x}_k - \mathbf{x}_l\|^2/2\sigma^2)}, \quad (3.9)$$

but this causes problems when a high-dimensional datapoint \mathbf{x}_i is an outlier (i.e., all pairwise distances $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ are large for \mathbf{x}_i). For such an outlier, the values of p_{ij} are extremely small for all j , so the location of its low-dimensional map point \mathbf{y}_i has very little effect on the cost function. As a result, the position of the map point is not well determined by the positions of the other map points. We circumvent this problem by defining the joint probabilities p_{ij} in the high-dimensional space to be the symmetrized conditional probabilities, i.e., we set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$. This ensures that $\sum_j p_{ij} > \frac{1}{2n}$ for all datapoints \mathbf{x}_i , as a result of which each datapoint \mathbf{x}_i makes

a significant contribution to the cost function. In the low-dimensional space, symmetric SNE simply uses Equation 3.8. The main advantage of the symmetric version of SNE is the simpler form of its gradient, which is faster to compute. The gradient of symmetric SNE is fairly similar to that of asymmetric SNE, and is given by

$$\frac{\delta C}{\delta \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(\mathbf{y}_i - \mathbf{y}_j). \quad (3.10)$$

In preliminary experiments, we observed that symmetric SNE seems to produce maps that are just as good as asymmetric SNE, and sometimes even a little better.

3.2.2 The crowding problem

Consider a set of datapoints that lie on a two-dimensional curved manifold which is approximately linear on a small scale, and which is embedded within a higher-dimensional space. It is possible to model the small pairwise distances between datapoints fairly well in a two-dimensional map, which is often illustrated on toy examples such as the ‘Swiss roll’ dataset. Now assume that the manifold has ten intrinsic dimensions⁵ and is embedded within a space of much higher dimensionality. There are several reasons why the pairwise distances in a two-dimensional map cannot faithfully model distances between points on the ten-dimensional manifold. For instance, in ten dimensions, it is possible to have 11 datapoints that are mutually equidistant and there is no way to model this faithfully in a two-dimensional map. A related problem is the different distribution of pairwise distances in the two spaces. The volume of a sphere centered on datapoint i scales as r^m , where r is the radius and m the dimensionality of the sphere. So if the datapoints are approximately uniformly distributed in the region around i on the ten-dimensional manifold, and we try to model the distances from i to the other datapoints in the two-dimensional map, we get the following ‘crowding problem’: the area of the two-dimensional map that is available to accommodate moderately distant datapoints will not be nearly large enough compared with the area available to accommodate nearby datapoints. Hence, if we want to model the small distances (fairly) accurately in the map, most of the huge number of points that are at a moderate distance from datapoint i will have to be placed much too far away in the two-dimensional map. In SNE, the spring connecting datapoint i to each of these too-distant map points will thus exert a very small attractive force. Although these attractive forces are very small, the large number of such forces crushes together the points in the center of the map, which prevents gaps from forming between the natural clusters. It should be remarked that the crowding problem is not specific to SNE, but that it also occurs in other techniques for multidimensional scaling such as Sammon mapping.

Cook *et al.* [2007] attempted to address the crowding problem by adding a slight repulsion to all springs. The slight repulsion is created by introducing a uniform background model with a small mixing proportion, ρ . So regardless of how far apart two map points are, q_{ij} can never fall below $\frac{\rho}{n(n-1)}$. As a result, for datapoints that are far apart in the high-dimensional space, q_{ij} will always be larger than p_{ij} , leading to a slight repulsion. This technique is called UNI-SNE and although it usually outperforms standard SNE, the optimization of the UNI-SNE cost function

⁵This is approximately correct for the images of handwritten digits we use in our experiments in Section 3.3.

is tedious. The best optimization method, so far, is to start by setting the background mixing proportion to zero (i.e., by performing standard SNE). Once the SNE cost function has been optimized using simulated annealing, the background mixing proportion can be increased to allow some gaps to be formed between natural clusters as shown by Cook *et al.* [2007]. Optimizing the UNI-SNE cost function directly does not work because two map points that are far apart will get almost all of their q_{ij} from the uniform background. So even if their p_{ij} is large, there will be no attractive force between them, because a small change in their separation will have a vanishingly small *proportional* effect on q_{ij} . This means that if two parts of a cluster become separated early on in the optimization, there is no force to pull them back together.

3.2.3 Mismatched tails compensate for mismatched dimensionalities

Since symmetric SNE is actually matching the joint probabilities of pairs of datapoints in the high-dimensional and the low-dimensional spaces rather than their distances, we have a natural way of alleviating the crowding problem. In the high-dimensional space, we convert distances into probabilities using a Gaussian distribution. In the low-dimensional map, we can use a probability distribution that has much heavier tails than a Gaussian to convert distances into probabilities, and as a result, eliminate the undesired attractive forces between dissimilar datapoints. This allows a moderate distance in the high-dimensional space to be faithfully modeled by a much larger distance in the map.

In *t*-SNE, we employ a Student *t*-distribution with one degree of freedom (which is the same as a Cauchy distribution) as the heavy-tailed distribution in the low-dimensional map. Using this distribution, the joint probabilities q_{ij} are defined as

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}. \quad (3.11)$$

We use a Student *t*-distribution with a single degree of freedom, because it has the particularly nice property that $(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}$ approaches an inverse square law for large pairwise distances $\|\mathbf{y}_i - \mathbf{y}_j\|$ in the low-dimensional map. This makes the map's representation of joint probabilities (almost) invariant to changes in the scale of the map for map points that are far apart. It also means that large clusters of points that are far apart interact in just the same way as individual points, so the optimization operates in the same way at all but the finest scales. A computationally convenient property is that it is much faster to evaluate the density of a point under a Student *t*-distribution than under a Gaussian because it does not involve an exponential, even though the Student *t*-distribution is equivalent to an infinite mixture of Gaussians with different variances.

The gradient of the Kullback-Leibler divergence between P (computed using Equation 3.9) and the Student-*t* based joint probability distribution Q (computed using Equation 3.11) is derived in Appendix B, and is given by

$$\frac{\delta C}{\delta \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(\mathbf{y}_i - \mathbf{y}_j) (1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}. \quad (3.12)$$

In Figure 3.1(a) to 3.1(c), we show the gradients between two low-dimensional datapoints \mathbf{y}_i and \mathbf{y}_j as a function of their pairwise distances in the high-dimensional and the low-dimensional

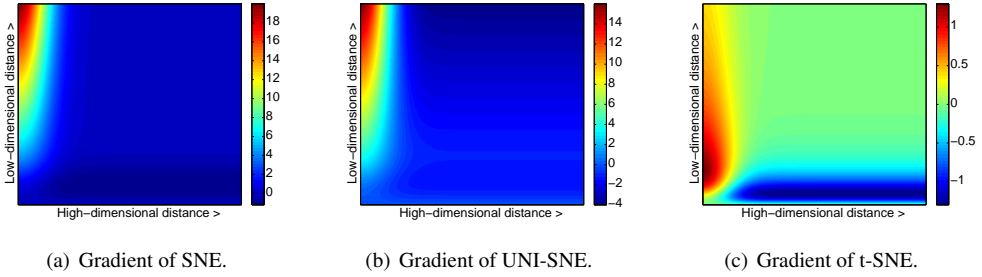


Figure 3.1 Gradients between two low-dimensional datapoints for three types of SNE as a function of their pairwise distance in the high-dimensional and low-dimensional data representation.

space (i.e., as a function of $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ and $\|\mathbf{y}_i - \mathbf{y}_j\|^2$) for the symmetric versions of SNE, UNI-SNE, and t-SNE. In the figures, positive values of the gradient represent an attraction between the low-dimensional datapoints \mathbf{y}_i and \mathbf{y}_j , whereas negative values represent a repulsion between the two datapoints. From the figures, we observe two main advantages of the t-SNE gradient over the gradients of SNE and UNI-SNE.

First, the t-SNE gradient strongly repels dissimilar datapoints that are modeled by a small pairwise distance in the low-dimensional representation. SNE has such a repulsion as well, but its effect is minimal compared to the strong attractions elsewhere in the gradient (the largest attraction in our graphical representation of the gradient is approximately 19, whereas the largest repulsion is approximately 1). In UNI-SNE, the amount of repulsion between dissimilar datapoints is slightly larger, however, this repulsion is only strong when the pairwise distance between the points in the low-dimensional representation is already large (which is often not the case, since the low-dimensional representation is initialized by sampling from a Gaussian with a very small variance that is centered around the origin).

Second, although t-SNE introduces strong repulsions between dissimilar datapoints that are modeled by small pairwise distances, these repulsions do not go to infinity. In this respect, t-SNE differs from UNI-SNE, in which the strength of the repulsion between very dissimilar datapoints is proportional to their pairwise distance in the low-dimensional map, which may cause dissimilar datapoints to move much too far away from each other.

Taken together, t-SNE puts emphasis on (1) modeling dissimilar datapoints by means of large pairwise distances, and (2) modeling similar datapoints by means of small pairwise distances. Moreover, as a result of these characteristics of the t-SNE cost function (and as a result of the approximate scale invariance of the Student t-distribution), the optimization of the t-SNE cost function is much easier than the optimization of the cost functions of SNE and UNI-SNE. In particular, good local optima can be found without resorting to simulated annealing.

3.2.4 Optimization methods for t-SNE

We start by presenting a relatively simple gradient descent procedure for optimizing the t-SNE cost function. This procedure uses a momentum term to reduce the number of iterations required and it works best if the momentum term is small until the map points have become moderately

well organized. Pseudocode for this simple algorithm is presented in Algorithm 1. The algorithm can be sped up using the adaptive learning rate scheme that is described by Jacobs [1988], which gradually increases the learning rate in directions in which the gradient is stable.

Algorithm 1: Simple version of *t*-Distributed Stochastic Neighbor Embedding.

Data: dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$,
 cost function parameters: perplexity $Perp$,
 optimization parameters: number of iterations T , learning rate η , momentum $\alpha^{(t)}$.
Result: low-dimensional data representation $\mathbf{Y}^{(T)} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$.

begin

- compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 3.1)
- set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
- sample initial solution $\mathbf{Y}^{(0)} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
- for** $t=1$ **to** T **do**
 - compute low-dimensional affinities q_{ij} (using Equation 3.11)
 - compute gradient $\frac{\delta C}{\delta \mathbf{Y}}$ (using Equation 3.12)
 - set $\mathbf{Y}^{(t)} = \mathbf{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathbf{Y}} + \alpha^{(t)} (\mathbf{Y}^{(t-1)} - \mathbf{Y}^{(t-2)})$
- end**

end

Although the algorithm described above produces visualizations that are often much better than those produced by other non-parametric dimensionality reduction techniques, the results can be improved further by using either of two tricks. The first trick, which we call ‘early compression’, is to force the map points to stay close together at the start of the optimization. When the distances between map points are small, it is easy for clusters to move through one another so it is much easier to explore the space of possible global organizations of the data. Early compression is implemented by adding an additional L2-penalty to the cost function that is proportional to the sum of squared distances of the map points from the origin. The magnitude of this penalty term and the iteration at which it is removed are set by hand, but the behavior is fairly robust across variations in these two additional optimization parameters.

A less obvious way to improve the optimization, which we call ‘early exaggeration’, is to multiply all of the p_{ij} ’s by, e.g., 4, in the initial stages of the optimization. This means that almost all of the q_{ij} ’s, which still add up to 1, are much too small to model their corresponding p_{ij} ’s. As a result, the optimization is encouraged to focus on modeling the large p_{ij} ’s by fairly large q_{ij} ’s. The effect is that the natural clusters in the data tend to form tight widely separated clusters in the map. This creates too much relatively empty space in the map, which makes it much easier for the clusters to move around relative to one another in order to find a good global organization.

In all the visualizations presented in this chapter and in the supporting material, we used exactly the same optimization procedure. We used the early exaggeration method with an exaggeration of 4 for the first 50 iterations. The number of gradient descent iterations T was set to 1000, and the momentum term was set to $\alpha^{(t)} = 0.5$ for $t < 250$ and $\alpha^{(t)} = 0.8$ for $t \geq 250$.

The learning rate η is initially set to 100 and it is updated after every iteration by means of the adaptive learning rate scheme described by Jacobs [1988].

3.3 Experiments

To evaluate t-SNE, we performed experiments in which t-SNE is compared to three other techniques for dimensionality reduction. We compare t-SNE with the three most popular dimensionality reduction techniques: (1) PCA, (2) Isomap, and (3) LLE. The three techniques were already introduced in Chapter 2, which is why we do not describe them here. We performed experiments on five datasets that represent a variety of application domains.

In subsection 3.3.1, the datasets that we employed in our experiments are introduced. The setup of the experiments is presented in subsection 3.3.2. In subsection 3.3.3, we present the results of our experiments.

3.3.1 Datasets

The five datasets we employed in our experiments are: (1) the MNIST dataset, (2) the ORL dataset, (3) the COIL-20 dataset, (4) the word-features dataset, and (5) the Netflix dataset. We briefly describe the five datasets below (note that the first three datasets were also employed in Chapter 2).

The MNIST dataset contains a training set of 60,000 grayscale images of handwritten digits. For our experiments, we randomly selected 6,000 of the digit images for computational reasons. The digit images have $28 \times 28 = 784$ pixels (i.e., dimensions). The ORL dataset consists of images of 40 individuals with small variations in viewpoint, large variations in expression, and occasional addition of glasses. The dataset consists of 400 images (10 per individual) of size $92 \times 112 = 10,304$ pixels. The COIL-20 dataset [Nene *et al.*, 1996] contains images of 20 different objects viewed from 72 equally spaced orientations, yielding a total of 1,440 images. The images contain $32 \times 32 = 1,024$ pixels. The word-features dataset [Mnih and Hinton, 2007] consists of 100-dimensional real-valued feature vectors for the 1,000 most common words in corpus of news articles from the period 1994-1996. The feature vectors were learned by trying to make the identity of the next word be as predictable as possible from the identities of the previous words when the predictions are made using the feature vectors (see [Mnih and Hinton, 2007] for details). The Netflix dataset contains ratings of 17,770 movies originating from over 400,000 movie viewers. A Restricted Boltzmann Machine with 30 hidden units was trained on these ratings (see Salakhutdinov *et al.* [2007] for details of the training), yielding a dataset of 30-dimensional movie-specific features. In our experiments on the Netflix dataset, we constructed visualizations for the 500 most popular movies.

3.3.2 Experimental setup

In all of our experiments, we start by using PCA to reduce the dimensionality of the data to 30. This speeds up the computation of pairwise distances between the datapoints and suppresses some noise without severely distorting the interpoint distances. We then use each of the dimensionality reduction techniques to convert the 30-dimensional representation to a two-dimensional

map and we show the resulting map as a scatterplot. For all of the datasets, there is information about the class of each datapoint, but the class information is only used to select a color and/or symbol for the map points. The class information is not used to determine the spatial coordinates of the map points. The coloring thus provides a way of evaluating how well the map preserves the similarities within each class.

The cost function parameter settings we employed in our experiments are listed in Table 3.1. In the table, $Perp$ represents the perplexity of a Gaussian kernel and k represents the number of nearest neighbors employed in a neighborhood graph. In the experiments with Isomap and LLE, we only visualize datapoints that correspond to vertices in the largest connected component of the neighborhood graph⁶.

<i>Technique</i>	<i>Cost function parameters</i>
t-SNE	$Perp = 40$
PCA	none
Isomap	$k = 12$
LLE	$k = 12$

Table 3.1 Cost function parameter settings for the experiments.

3.3.3 Results

In Figure 3.2, we show the results of our experiments with t-SNE, PCA, Isomap, and LLE on the MNIST dataset. The results reveal the strong performance of t-SNE compared to the other techniques. In particular, PCA constructs a ‘ball’ in which only three classes (representing the digits 0, 1, and 7) are somewhat separated from the other classes. Isomap and LLE produce solutions in which there are large overlaps between the digit classes. In contrast, t-SNE constructs a map in which the separation between the digit classes is almost perfect. Moreover, detailed inspection of the t-SNE map reveals that much of the local structure of the data (such as the orientation of the ones) is captured as well. This is illustrated in more detail in Section 3.4 (see Figure 3.6). The map produced by t-SNE contains some points that are clustered with the wrong class, but most of these points correspond to distorted digits that are often quite difficult to identify.

Figure 3.3 shows the results of applying t-SNE, PCA, Isomap, and LLE to the ORL dataset. Again, Isomap and LLE produce solutions that provide little insight into the class structure of the data. The map constructed by PCA seems slightly better, however, it does not reveal the natural classes in the data. In contrast, t-SNE does a much better job of revealing the natural classes in the data. Some individuals have their ten images split into two clusters, usually because a subset of the images have the head facing in a significantly different direction, or because they have a different expression or glasses. For these individuals, it is not clear that their ten images form a natural class when using the Euclidean distance in pixel space.

Figure 3.4 shows the results of applying t-SNE, PCA, Isomap, and LLE to the COIL-20 dataset. For many of the 20 objects, t-SNE accurately represents the one-dimensional manifold

⁶Isomap and LLE require data that gives rise to a neighborhood graph that is connected.

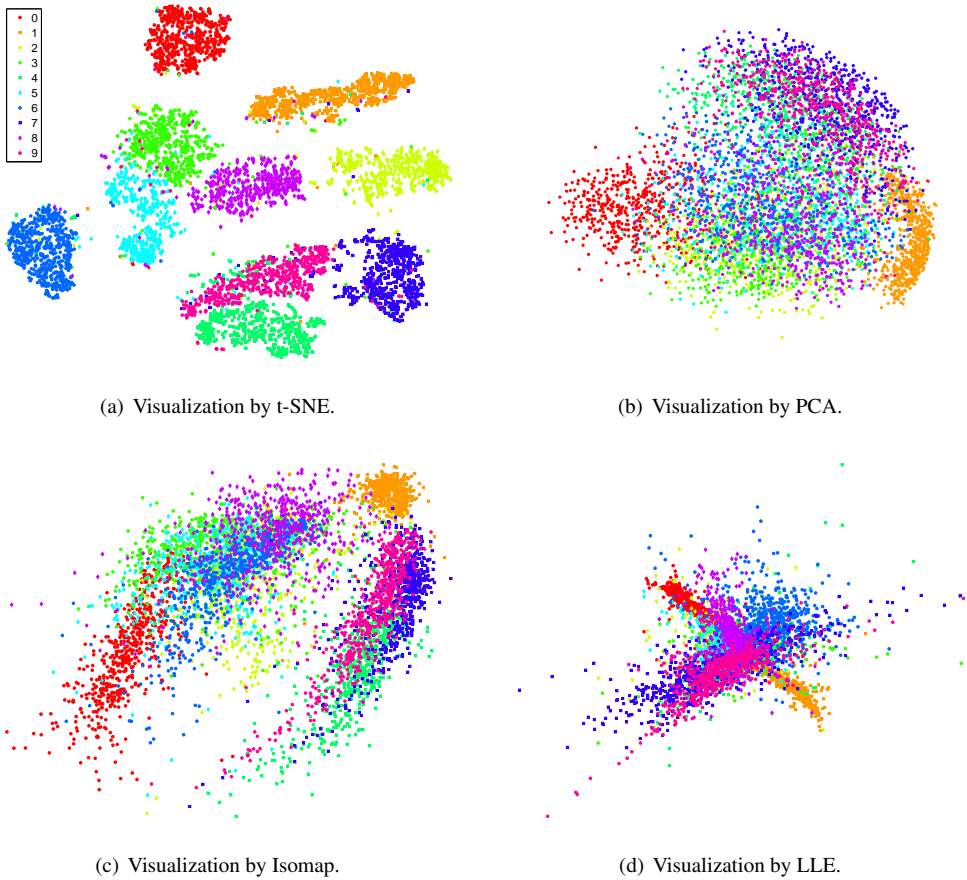


Figure 3.2 Visualizations of 6,000 handwritten digits from the MNIST dataset.

of viewpoints as a closed loop. For objects which look similar from the front and the back, t-SNE distorts the loop so that the images of front and back are mapped to nearby points. For the three types of toy cars in the COIL-20 dataset (the aligned ‘sausages’ in the bottom-left of the t-SNE map), the three rotation manifolds are aligned to capture the high similarity between different cars at the same orientation. This prevents t-SNE from keeping the four manifolds clearly separate. Figure 3.4 also reveals that the other three techniques are not nearly as good at cleanly separating the manifolds that correspond to different objects. In addition, Isomap and LLE only visualize a small number of classes from the COIL-20 dataset, because the dataset comprises a large number of widely separated submanifolds that give rise to small connected components in the neighborhood graph.

Because of space limitations⁷, the visualizations of the word-features dataset and the Netflix dataset are not presented in the thesis. The visualizations are available online in the supplemental

⁷If the words or the names of the movies are plotted in such a way that they do not overlap, the characters are too small to be easily read.

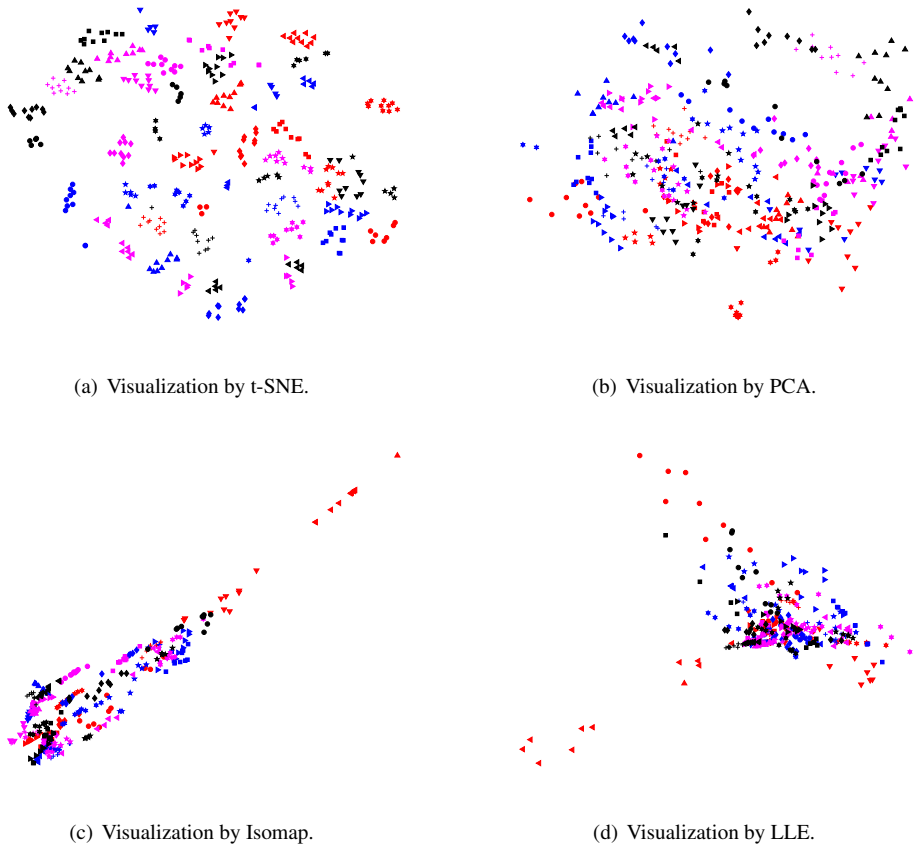


Figure 3.3 Visualizations of the ORL dataset.

material of [van der Maaten and Hinton, 2008]. In order to quantify the performance of t-SNE relative to the three other techniques, we measured the trustworthinesses (see 2.5.1) of the visualizations constructed by the dimensionality reduction techniques. The trustworthinesses are presented in Table 3.2. The best trustworthiness of each experiment is typeset in boldface. The results presented in the table confirm the observations we made from the visualizations, as t-SNE outperforms the three other dimensionality reduction techniques in all experiments.

<i>Technique</i>	<i>MNIST</i>	<i>ORL</i>	<i>COIL-20</i>	<i>Words</i>	<i>Netflix</i>
t-SNE	0.90	0.97	0.99	0.83	0.89
PCA	0.77	0.86	0.89	0.69	0.83
Isomap	0.52	0.30	0.88	0.65	0.42
LLE	0.69	0.80	0.76	0.59	0.70

Table 3.2 Trustworthinesses $T(12)$ of the visualizations of the five datasets.

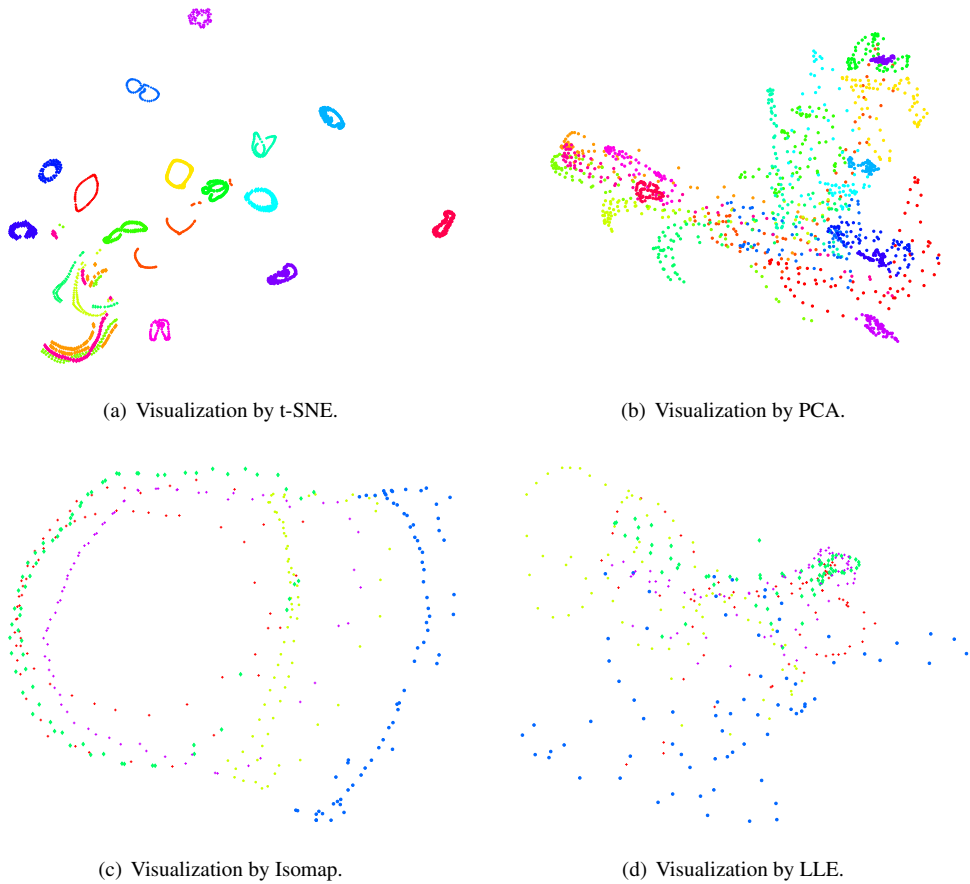


Figure 3.4 Visualizations of the COIL-20 dataset.

3.4 Applying t-SNE to large datasets

Like many other visualization techniques, t-SNE has a computational and memory complexity that is quadratic in the number of datapoints. This makes it infeasible to apply the standard version of t-SNE to datasets that contain many more than, say, 10,000 points. Obviously, it is possible to pick a random subset of the datapoints and display them using t-SNE, but such an approach fails to make use of the information that the undisplayed datapoints provide about the underlying manifolds. Assume, for example, that A, B, and C are all equidistant in the high-dimensional space. If there are many undisplayed datapoints between A and B and none between A and C, it is much more likely that A and B are part of the same cluster than A and C. This is illustrated in Figure 3.5. In this section, we show how t-SNE can be modified to display a random subset of the datapoints (so-called landmark points) in a way that uses information from the entire (possibly very large) dataset.

We start by choosing a desired number of neighbors and creating a neighborhood graph for all of the datapoints. Although this is computationally intensive, it is only done once. Then, we

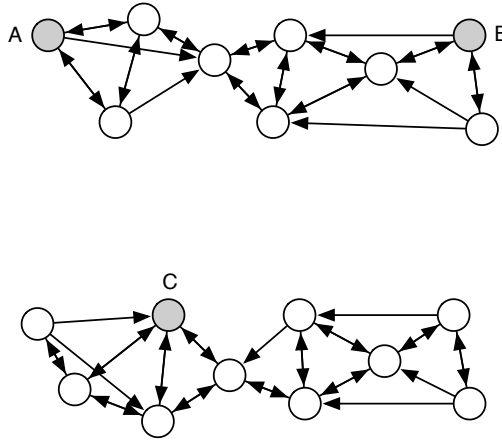


Figure 3.5 An illustration of the advantage of the random walk version of t-SNE over a standard landmark approach. The shaded points A, B, and C are three (almost) equidistant landmark points, whereas the non-shaded datapoints are non-landmark points. The arrows represent a directed neighborhood graph where $k = 3$. In a standard landmark approach, the pairwise affinity between A and B is approximately equal to the pairwise affinity between A and C. In the random walk version of t-SNE, the pairwise affinity between A and B is much larger than the pairwise affinity between A and C, and therefore, it reflects the structure of the data much better.

define a random walk on the neighborhood graph for each of the landmark points, starting at that landmark point and terminating as soon as it lands on another landmark point. During a random walk, the probability of choosing an edge emanating from node \mathbf{x}_i to node \mathbf{x}_j is proportional to $e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}$. We define⁸ $p_{j|i}$ to be the fraction of random walks starting at landmark point \mathbf{x}_i that terminate at landmark point \mathbf{x}_j . This has some resemblance to the way Isomap measures pairwise distances between points. However, as in diffusion maps [Lafon and Lee, 2006; Nadler *et al.*, 2006], rather than looking for the shortest path through the neighborhood graph, the random walk-based affinity measure integrates over all paths through the neighborhood graph. As a result, the random walk-based affinity measure is much less sensitive to ‘short-circuits’ [Lee and Verleysen, 2005], in which a single noisy datapoint provides a bridge between two regions of dataspace that should be far apart in the map. Similar approaches using random walks have also been successfully applied to, e.g., semi-supervised learning [Szummer and Jaakkola, 2001; Zhu *et al.*, 2003] and image segmentation [Grady, 2006].

The most obvious way to compute the random walk-based similarities $p_{j|i}$ is to perform the random walks explicitly on the neighborhood graph, which works very well in practice, given that one can easily perform one million random walks per second. Alternatively, Grady presents an analytical solution to compute the pairwise similarities $p_{j|i}$ that involves solving a sparse linear system [Grady, 2006]. The analytical solution to compute the similarities $p_{j|i}$ is sketched in Appendix C. In preliminary experiments, we did not find significant differences between performing the random walks explicitly and the analytical solution. In the experiment presented below, we explicitly performed the random walks because this is computationally less expensive.

⁸Note that t-SNE estimates the joint probabilities p_{ij} by symmetrizing: $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$.

However, for very large datasets in which the landmark points are sparse, the analytical solution may be more appropriate.

Figure 3.6 shows the results of an experiment, in which we applied the random walk version of t-SNE to 6,000 randomly selected digits from the MNIST dataset, using all 60,000 digits in the trainingset to compute the pairwise affinities $p_{j|i}$. In the experiment⁹, we used a neighborhood graph that was constructed using a value of $k = 20$. The inset of the figure shows the same visualization as a scatterplot in which the colors represent the labels of the digits. In the t-SNE map, all classes are clearly separated and the ‘continental’ sevens¹⁰ form a small separate cluster. Moreover, t-SNE reveals the main dimensions of variation within each class, such as the orientation of the ones, fours, sevens, and nines, or the ‘loopiness’ of the twos. The strong performance of t-SNE is also reflected in the generalization error of nearest neighbor classifiers that are trained on the low-dimensional representation. Whereas the generalization error (measured using 10-fold cross validation) of a 1-nearest neighbor classifier trained on the original 784-dimensional datapoints is 5.75%, the generalization error of a 1-nearest neighbor classifier trained on the two-dimensional data representation produced by t-SNE is only 5.13%. The computational requirements of random walk t-SNE are reasonable: it took only one hour of CPU time to construct the map in Figure 3.6.

3.5 Discussion

The results presented in Section 3.3 and 3.4 illustrate the strong performance of t-SNE on a wide variety of datasets. In this section, we discuss the performance of t-SNE relative to other non-parametric techniques (subsection 3.5.1), and we discuss a number of weaknesses and possible improvements of t-SNE (subsection 3.5.2).

3.5.1 Comparison with related techniques

PCA [Pearson, 1901; Hotelling, 1933] and classical scaling [Torgerson, 1952] find a linear mapping that minimizes the squared error of the pairwise Euclidean distances in the low-dimensional map (see subsection 2.2.1 for details). This leads to two main weaknesses of PCA and classical scaling: (1) the techniques can only project data onto a linear subspace of the original high-dimensional space and (2) the cost function of the techniques assigns relatively large importance to retaining large pairwise distances. In other words, PCA and classical scaling are not capable of identifying data that lies on or near complex nonlinear manifolds in the original high-dimensional space, and do not preserve the local structure of the data (that is generally more important than the global structure of the data that PCA and classical scaling retain).

In contrast to PCA and classical scaling, the Gaussian kernel employed in the high-dimensional space by t-SNE defines a soft border between the local and global structure of the data. For pairs of datapoints that are close together relative to the standard deviation of the Gaussian, the importance of modeling their separations is almost independent of the magnitudes of those separations. Moreover, t-SNE determines the local neighborhood size for each datapoint

⁹In preliminary experiments, we found the performance of random walk t-SNE to be robust under changes of k .

¹⁰A ‘continental’ seven is a seven that has a horizontal cross-bar.

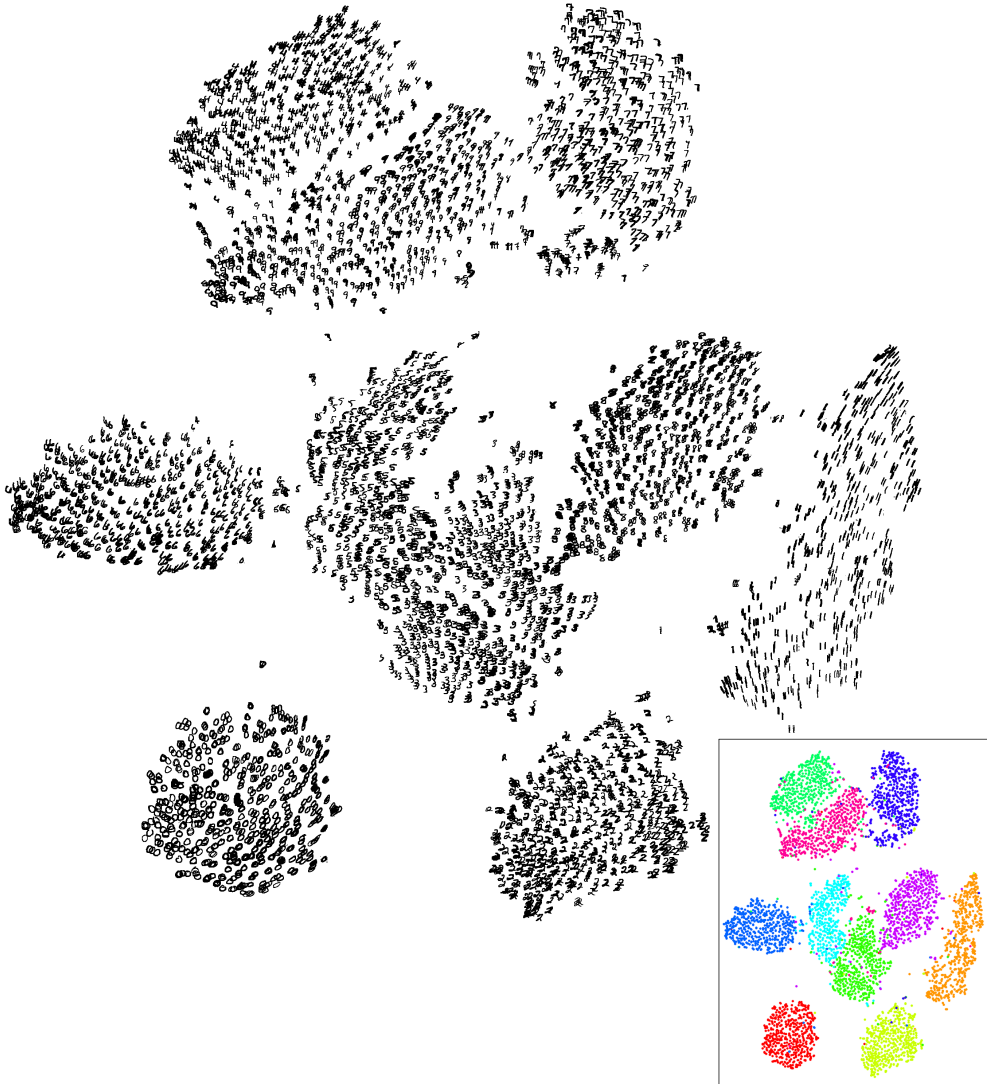


Figure 3.6 Visualization of 6,000 digits from the MNIST dataset produced by the random walk version of t-SNE (employing all 60,000 digit images).

separately based on the local density of the data (by forcing each conditional probability distribution P_i to have the same perplexity).

The experiments presented in the chapter reveal that t-SNE outperforms multidimensional scaling techniques such as Sammon mapping, and that it also outperforms more recent manifold learners such as Isomap and LLE. The strong performance of t-SNE compared to Isomap is partly explained by Isomap’s susceptibility to ‘short-circuiting’ (see subsection 2.6.1). Moreover, Isomap mainly focuses on retaining large geodesic distances instead of on retaining small geodesic distances. The strong performance of t-SNE compared to LLE is due to the weaknesses of local dimensionality reduction techniques as discussed in subsection 2.6.2. The results of our experiments reveal that the poor performance of LLE is mainly due to the cost function problem of LLE: the only thing that prevents all datapoints from collapsing onto a single point is a constraint on the covariance of the low-dimensional representation. In practice, this constraint is often satisfied by placing most of the map points near the center of the map and using a few widely scattered points to create large covariance (see Figure 3.2(d) and 3.3(d)). For neighborhood graphs that are almost disconnected, the covariance constraint can also be satisfied by a ‘curdled’ map in which there are a few widely separated, collapsed subsets. Moreover, neighborhood-graph based techniques (such as Isomap and LLE) are not capable of visualizing data that consists of two or more widely separated submanifolds, because such data does not give rise to a connected neighborhood graph. It is possible to construct a map for each connected component, but this loses information about the similarities between the separate components.

Like Isomap and LLE, the random walk version of t-SNE employs neighborhood graphs, but it does not suffer from short-circuiting problems because the pairwise similarities between the high-dimensional datapoints are computed by integrating over all paths through the neighborhood graph. Because of the diffusion-based interpretation of the conditional probabilities underlying the random walk version of t-SNE, it is useful to compare t-SNE to diffusion maps (see subsection 2.2.1). Recall that diffusion maps define a ‘diffusion distance’ on the high-dimensional datapoints that is given by

$$D^{(t)}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_k \frac{(p_{ik}^{(t)} - p_{jk}^{(t)})^2}{\psi(\mathbf{x}_k)^{(0)}}}, \quad (3.13)$$

where $p_{ij}^{(t)}$ represents the probability of a particle traveling from \mathbf{x}_i to \mathbf{x}_j in t timesteps (through a graph on the data with Gaussian emission probabilities). The term $\psi(\mathbf{x}_k)^{(0)}$ is a measure for the local density of the points, and serves a similar purpose to the fixed perplexity Gaussian kernel that is employed in SNE. The diffusion map is formed by the principal non-trivial eigenvectors of the Gaussian kernel. It can be shown that when all $(n - 2)$ non-trivial eigenvectors¹¹ are employed, the Euclidean distances in the diffusion map are equal to the diffusion distances in the high-dimensional data representation [Lafon and Lee, 2006]. Mathematically, diffusion maps minimize

$$C = \sum_i \sum_j \left(D^{(t)}(\mathbf{x}_i, \mathbf{x}_j) - \|\mathbf{y}_i - \mathbf{y}_j\| \right)^2. \quad (3.14)$$

¹¹Notice that both the major and the minor eigenvalues are trivial: the major eigenvalue is 1, whereas the minor eigenvalue is 0.

As a result, diffusion maps are susceptible to the same problems as PCA and classical scaling: they assign much higher importance to modeling the large pairwise diffusion distances than the small ones and as a result, they are not good at retaining the local structure of the data. Moreover, in contrast to the random walk version of t-SNE, diffusion maps do not have a natural way of selecting the length, t , of the random walks.

In preliminary experiments, we also performed experiments in which we compared t-SNE to CCA [Demartines and Hérault, 1997], MVU [Weinberger *et al.*, 2004], and Laplacian Eigenmaps [Belkin and Niyogi, 2002]. The results with these techniques are not shown here because of space limitations (the plots are shown in the supplemental material of [van der Maaten *et al.*, 2009]). On all datasets, we found t-SNE to outperform these techniques. For CCA and the closely related CDA [Lee *et al.*, 2000], these results can be partially explained by the hard border λ that these techniques define between local and global structure, as opposed to the soft border of t-SNE. Moreover, within the range λ , CCA suffers from the same weakness as Sammon mapping: retaining a pairwise distance of, say, 0.001 is much more important than retaining a pairwise distance of 0.003, due to the quadratic cost function. For MVU, the results may be partially explained from the fact that MVU, just like Isomap, suffers from short-circuiting: a single erroneous constraint may severely affect the performance of MVU. Also, MVU makes no attempt to model longer range structure. For Laplacian Eigenmaps, the results may be explained from the weaknesses discussed in subsection 2.6.2. Most importantly, Laplacian Eigenmaps have the same covariance constraint as LLE, and it is easy to cheat on this constraint.

3.5.2 Weaknesses

Although we have shown that t-SNE compares favorably to other techniques for data visualization, t-SNE has three potential weaknesses: (1) it is unclear how t-SNE performs on general dimensionality reduction tasks, (2) the relatively local nature of t-SNE makes it sensitive to the curse of the intrinsic dimensionality of the data, and (3) t-SNE is not guaranteed to converge to a global optimum of its cost function. Below, we discuss the three weaknesses in more detail.

1) Dimensionality reduction for other purposes. It is not obvious how t-SNE will perform on the more general task of dimensionality reduction (i.e., when the dimensionality of the data is not reduced to two or three, but to $d > 3$ dimensions). To simplify evaluation issues, this chapter only considers the use of t-SNE for data visualization. The behavior of t-SNE when reducing data to two or three dimensions cannot readily be extrapolated to $d > 3$ dimensions because of the heavy tails of the Student-t distribution. In high-dimensional spaces, the heavy tails comprise a relatively large portion of the probability mass under the Student-t distribution, which might lead to d -dimensional data representations that do not preserve the local structure of the data as well. In Chapter 4, we discuss this issue in more detail and we present an approach that addresses this weakness.

2) Curse of intrinsic dimensionality. t-SNE reduces the dimensionality of data mainly based on local properties of the data, which makes t-SNE sensitive to the curse of the intrinsic dimensionality of the data [Bengio, 2007]. In datasets with a high intrinsic dimensionality and an underlying manifold that is highly varying, the local linearity assumption on the manifold

that t-SNE implicitly makes (by employing Euclidean distances between near neighbors) may be violated. As a result, t-SNE might be less successful if it is applied on datasets with a high intrinsic dimensionality (for instance, a recent study estimates the face space to be constituted of approximately 100 dimensions [Meytlis and Sirovich, 2007]). Manifold learners such as Isomap and LLE suffer from exactly the same problems (see, e.g., [Bengio, 2007]). A possible way to (partially) address this issue is by performing t-SNE on a data representation obtained from a model that represents the highly varying data manifold efficiently in a number of nonlinear layers such as an autoencoder [Hinton and Salakhutdinov, 2006]. Such deep-layer architectures can represent complex nonlinear functions in a much simpler way, and as a result, require fewer datapoints to learn an appropriate solution (as is illustrated for a d -bits parity task by [Bengio, 2007]). Performing t-SNE on a data representation produced by, e.g., an autoencoder is likely to improve the quality of the constructed visualizations, because autoencoders can identify highly-varying manifolds better than a local method such as t-SNE. However, the reader should note that it is by definition impossible to fully represent the structure of intrinsically high-dimensional data in two or three dimensions.

3) *Non-convexity of the t-SNE cost function.* A nice property of most state-of-the-art dimensionality reduction techniques (such as classical scaling, Isomap, LLE, and diffusion maps) is the convexity of their cost functions. A major weakness of t-SNE is that the cost function is not convex, as a result of which several optimization parameters need to be chosen. The constructed solutions depend on the choices of the optimization parameters and may be different each time t-SNE is run from an initial random configuration of map points. We have demonstrated that the same choice of optimization parameters can be used for a variety of different visualization tasks, and we found that the quality of the optima does not vary much from run to run. Therefore, we believe that the weakness of the optimization method is insufficient reason to reject t-SNE in favor of methods that lead to convex optimization problems but produce noticeably worse visualizations. A local optimum of a cost function that accurately captures what we want in a visualization is often preferable to the global optimum of a cost function that fails to capture important aspects of what we want. Moreover, the convexity of cost functions can be misleading, because their optimization is often computationally infeasible for large real-world datasets, prompting the use of approximation techniques [de Silva and Tenenbaum, 2003; Weinberger *et al.*, 2007].

3.6 Chapter conclusions

The chapter presented a new technique for the extraction of dimensionality reduction features, called t-SNE, that is capable of retaining local structure of the data while also revealing some important global structure of the data (such as clusters). We showed the strong performance of t-SNE in a number of experiments on five datasets. Both the computational and the memory complexity of t-SNE are $O(n^2)$. Yet, we presented an approach that makes it possible to visualize successfully large real-world datasets with limited computational demands. From the experimental results, we may conclude that t-SNE is a valuable new technique for the extraction of dimensionality reduction features.

4 Extensions of t-Distributed Stochastic Neighbor Embedding

- Contents** Even though the strong performance of t-SNE is of high value in visualization tasks, t-SNE cannot readily be employed to resolve the dimensionality problem in many computer vision systems, because these systems often require a parametric mapping between the data space and the latent space. Moreover, the performance of t-SNE may be hampered by the metric nature of the low-dimensional map, as a result of which it is not possible to model, e.g., asymmetric similarities in the map. To resolve these problems, the chapter develops two new variants of t-SNE in order to gain more insight into the answer of research question RQ1. The first variant of t-SNE provides a parametric mapping between the data space and the low-dimensional latent space, as is required in many computer vision systems. The second variant of t-SNE employs a latent space with a non-metric nature, which provides the capability to model non-metric similarities between objects.
- Based on** L.J.P. van der Maaten. Learning a Parametric Embedding by Preserving Local Structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR W&CP 5:384-391, 2009.
- Outline** In Section 4.1, we present the parametric version of t-SNE. In Section 4.2, we present a variant of t-SNE that constructs multiple maps instead of a single map, as a result of which the latent space is non-metric. Section 4.3 concludes the chapter.

In this chapter, we extend t-SNE to two learning settings that were not addressed in the previous chapter. First, we present a parametric version of t-SNE that can be used in learning settings in which generalization to unseen test data is required (Section 4.1). Second, we present a variant of t-SNE that can embed objects whose pairwise similarities do not obey the metric axioms, such as semantic similarities, by constructing a collection of multiple maps (Section 4.2).

4.1 Parametric t-SNE

Like other techniques for multidimensional scaling, t-SNE is a non-parametric technique for dimensionality reduction. Therefore, it cannot readily be applied in learning settings in which the goal is to generalize to held-out test data or to new datapoints. Generalization to held-out or new datapoints may be desirable if not all datapoints are available at training time, for instance, in typical classification tasks or when rapid visualization of new data is required. In this section, we describe a parametric version of t-SNE that allows for rapid generalization by providing a parametric mapping from the high-dimensional space to the low-dimensional space.

In parametric t-SNE, we parametrize the mapping $f : X \rightarrow Y$ from the high-dimensional data space X to the low-dimensional space Y by means of a feed-forward neural network with weights W . We opt for the use of a (deep) neural network, because a neural network with sufficient hidden layers (with nonlinear activation functions) is capable of parametrizing arbitrarily complex functions. The main drawback of the use of deep neural networks is that the millions of weights in the network cannot be learned successfully using backpropagation, as backpropagation tends to get stuck in poor local minima due to the complex interactions between the layers in the network. In order to circumvent this problem, we use a training procedure that is inspired by the training of autoencoders that we described in Section 2.3.

The training procedure of a parametric t-SNE network consists of three main stages. First, a stack of Restricted Boltzmann Machines (RBMs) is trained. Second, the stack of RBMs is used to construct a pretrained feed-forward neural network. Third, the pretrained feed-forward neural network is finetuned in order to minimize the t-SNE cost function. The training procedure is illustrated in Figure 4.1. We describe the three main stages of the training of a parametric t-SNE network separately below.

First, the multilayer network is pretrained by greedily training a stack of Restricted Boltzmann Machines (RBMs) using the procedure initially proposed by Hinton and Salakhutdinov [2006]. This greedy training procedure consists of three steps that are repeated for each layer in the neural network: (i) the RBM that corresponds to the first layer is trained on the input data (the training of RBMs is described in Appendix D), (ii) the most likely values for the hidden nodes of the RBM are inferred for each datapoint, and (iii) these values are used as input data to train the RBM that corresponds to the second layer. This process is repeated for all layers in the network. The RBMs that correspond to the bottom layers of the neural network have Bernoulli distributed hidden units and a linear energy function, because this gives rise to a sigmoid activation function in the network (see Appendix D). The RBM that corresponds to the top layer of the neural network uses Gaussian distributed hidden units and a quadratic energy function, because this gives rise to a linear activation function in the network. The top layer of a neural network typically has a linear activation function to make the outputs of the network more stable.

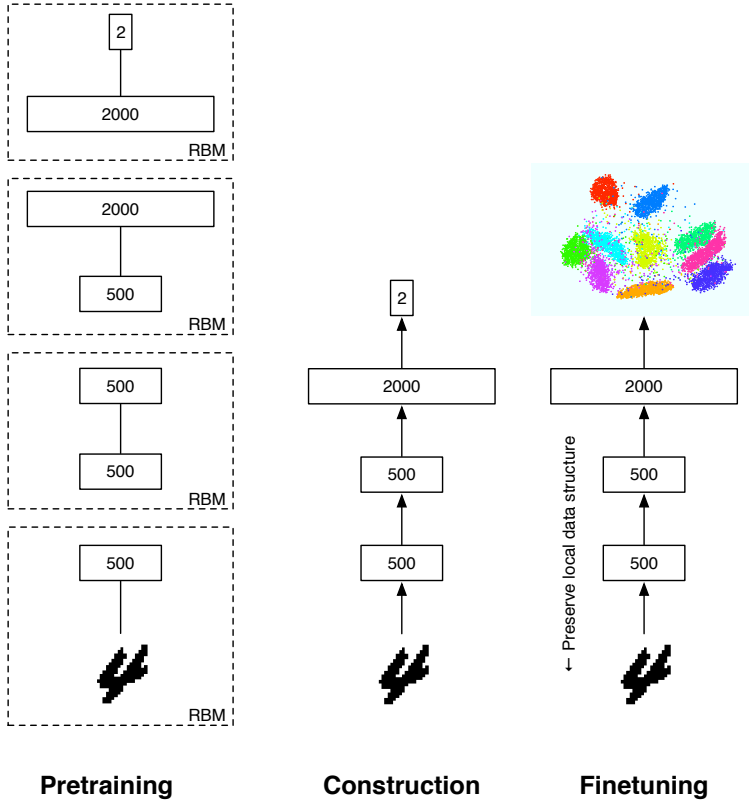


Figure 4.1 Overview of the three-stage training procedure of a parametric t-SNE network.

Second, the undirected weights of the RBMs are untied and the biases on the visible units of the respective RBMs are dropped, which transforms the stack of RBMs into a pretrained feed-forward network. Even though the resulting neural network was trained in a completely unsupervised manner, it usually forms an appropriate initialization for backpropagation approaches. For instance, the pretrained neural network can be used to initialize autoencoders (as we explained in 2.3) or nonlinear variants of Neighborhood Components Analysis [Salakhutdinov and Hinton, 2007]. Here, we use the pretrained network as an initial solution to the procedure that minimizes the t-SNE cost function with respect to the weights of the neural network.

Third, the weights of the pretrained feed-forward network are finetuned in such a way that the network minimizes the t-SNE cost function that was given in Equation 3.7. The introduction of the parametric mapping requires the pairwise similarities q_{ij} to be redefined. We denote the mapping from the high-dimensional to the low-dimensional space, which depends on the setting of the weights W of the neural network, as $f : X \rightarrow Y$. Using the mapping f , the pairwise similarities q_{ij} in the low-dimensional space are redefined as

$$q_{ij} = \frac{(1 + \|f(\mathbf{x}_i|W) - f(\mathbf{x}_j|W)\|^2/v)^{-\frac{v+1}{2}}}{\sum_{k \neq l} (1 + \|f(\mathbf{x}_k|W) - f(\mathbf{x}_l|W)\|^2/v)^{-\frac{v+1}{2}}}, \quad (4.1)$$

where v represents the number of degrees of freedom of the Student *t*-distribution. Note that we treat the degrees of freedom v as a free parameter here, whereas in Chapter 3, we simply set $v = 1$. As we will explain in detail later in this section, a higher value of v may be more appropriate if we reduce the data dimensionality to, say, 30 dimensions.

The new definition of the low-dimensional pairwise similarities q_{ij} allows us to minimize the ‘normal’ *t*-SNE cost function C (see Equation 3.7) with respect to the network weights W . Because of its large resemblance to the gradient of non-parametric *t*-SNE, the derivation of the gradient of parametric *t*-SNE is not given here. The required gradient $\frac{\delta C}{\delta W}$ is given by

$$\frac{\delta C}{\delta W} = \frac{\delta C}{\delta f(\mathbf{x}_i|W)} \frac{\delta f(\mathbf{x}_i|W)}{\delta W}, \quad (4.2)$$

where $\frac{\delta f(\mathbf{x}_i|W)}{\delta W}$ is computed using standard backpropagation, and $\frac{\delta C}{\delta f(\mathbf{x}_i|W)}$ is given by

$$\frac{\delta C}{\delta f(\mathbf{x}_i|W)} = \frac{2v+2}{v} \sum_j (p_{ij} - q_{ij}) (f(\mathbf{x}_i|W) - f(\mathbf{x}_j|W)) (1 + \|f(\mathbf{x}_i|W) - f(\mathbf{x}_j|W)\|^2/v)^{-\frac{v+1}{2}}. \quad (4.3)$$

The minimization of the cost function usually has to be performed using batches of a few thousand points, as the number of p_{ij} ’s and q_{ij} ’s grows quadratically with the number of datapoints in the batch.

A potential weakness of parametric *t*-SNE is that the tails of the Student-*t* distribution that is used in the low-dimensional space may contain a large portion of the probability mass under the distribution, because the volume of the low-dimensional space Y grows exponentially with its dimensionality. This problem may be addressed by setting the degrees of freedom v as to correct for the exponential growth of the volume of the low-dimensional space, because increasing the degrees of freedom v gives rise to a distribution with lighter tails. In fact, the parameter v determines to what extent the low-dimensional space is ‘filled up’: lower values of v lead to larger separations in the low-dimensional space between the natural clusters in the data, because they give rise to stronger repulsive forces between dissimilar datapoints. In contrast, higher values of v lead to smaller separations between the natural clusters in the data, as a result of which more space is available in the low-dimensional space to appropriately model the local structure of the data. Below, we discuss two approaches to set the degrees of freedom of the Student-*t* distribution that is used to measure pairwise similarities in the low-dimensional space.

1) *Linear dependency.* As the thickness of the tail of a Student-*t* distribution decreases exponentially with the degrees of freedom v , it seems likely that the parameter setting for degrees of freedom v should be linearly dependent on the dimensionality of the low-dimensional space d . Hence, it seems reasonable to set $v = d - 1$ in order to obtain a single degree of freedom in two-dimensional maps (as in Chapter 3).

2) *Learning v .* A potential problem of the approach presented above is that the appropriate value of v does not only depend on the dimensionality of the low-dimensional space. In fact, the most appropriate setting of v depends on the magnitude of the crowding problem, which in turn depends on the ratio between the *intrinsic dimensionality* of the data and the dimensionality of

the low-dimensional space. For instance, if the intrinsic dimensionality is equal to the dimensionality of the low-dimensional space, the crowding problem does not occur at all, and the most appropriate value is thus $v = \infty$ (note that a Student-t distribution with infinite degrees of freedom is equal to a Gaussian distribution). As the intrinsic dimensionality of the data at hand is usually unknown, it seems reasonable to treat v as a free parameter that should be optimized with respect to the cost function as well. The required gradient of the cost function C with respect to v is given by

$$\frac{\delta C}{\delta v} = \sum_{i \neq j} \left(\frac{(-v-1)d_{ij}^2}{2v^2 \left(1 + \frac{d_{ij}^2}{v}\right)} + \frac{1}{2} \log \left(1 + \frac{d_{ij}^2}{v}\right) \right) (p_{ij} - q_{ij}), \quad (4.4)$$

where d_{ij}^2 represents $\|f(\mathbf{x}_i|W) - f(\mathbf{x}_j|W)\|^2$. In the following section, we present experiments in which we used three different settings for v : (i) $v = 1$, (ii) $v = d - 1$, and (iii) learn v using the gradient presented above.

4.1.1 Experiments

This subsection describes experiments in which we compare the performance on two datasets of parametric t-SNE with two other unsupervised parametric techniques for dimensionality reduction, viz., PCA and autoencoders. The subsection separately describes (i) the setup of the experiments and (ii) the results of the experiments.

Experimental setup

We performed experiments on two handwritten character datasets, one of which we already used in Chapter 2 and 3: (1) the MNIST dataset and (2) the characters dataset. The MNIST dataset contains 70,000 images of handwritten digits of size 28×28 pixels. The dataset has a fixed division into 60,000 training images and held out 10,000 test images. The characters dataset consists of 40,121 grayscale images of handwritten upper-case characters and numerals of size 90×90 pixels [van der Maaten, 2009], of which we used 35,000 images as training data and the remainder as test data. The characters dataset comprises 35 classes, viz. 10 numeric classes and 25 alpha classes (the character ‘X’ is missing in the dataset).

In our experiments, we compared parametric t-SNE with two other unsupervised parametric techniques for dimensionality reduction, viz., PCA and multilayer autoencoders [Hinton and Salakhutdinov, 2006]. We evaluated the performance of the three techniques by means of plotting two-dimensional visualizations, measuring generalization performances of nearest-neighbor classifiers, and evaluating the trustworthiness [Venna and Kaski, 2006] of the low-dimensional embeddings. In order to make the comparison between parametric t-SNE and autoencoders as fair as possible, we used the same layout for both neural networks (where it should be noted that a parametric t-SNE network does not have the decoder part of an autoencoder). Motivated by the experimental setup employed by Salakhutdinov and Hinton [2007], we used $28 \times 28 - 500 - 500 - 2000 - 2$ parametric t-SNE networks and autoencoders in our experi-

ments on the MNIST dataset¹. In our experiments on the characters dataset, we used networks with a similar structure, viz. $90 \times 90 - 500 - 500 - 2000 - 2$ networks. The autoencoders were trained using the same three-stage training approach as parametric t-SNE, but the autoencoder is finetuned by performing backpropagation as to minimize the sum of squared errors between the input and the output of the autoencoder (see [Hinton and Salakhutdinov, 2006] for details).

We used exactly the same procedure and parameter settings in the pretraining of the parametric t-SNE networks and the autoencoders. In the training of the RBMs of which the hidden units have sigmoid activation functions (the RBMs in the first three layers), the learning rate is set to 0.1, and the weight decay is set to 0.0002. The training of the RBMs with a linear activation function in the hidden units (the RBMs in the fourth layer) is performed using a learning rate of 0.01 and a weight decay of 0.0002. In the training of all RBMs, the momentum is set to 0.5 for the first five iterations, and to 0.9 afterwards. The RBMs are all trained using 50 iterations of contrastive divergence with one complete Gibbs sweep per iteration (CD-1).

Both parametric t-SNE and the autoencoders were finetuned using 30 iterations of backpropagation using conjugate gradients on batches of 5,000 datapoints. The subdivision of training data into batches was fixed in order to facilitate the precomputation of the P matrices that are required in parametric t-SNE. In the experiments with parametric t-SNE, the variance σ_i of the Gaussian distributions was set such that the perplexity of the conditional distributions P_i was equal to 30.

Results

In Figure 4.2, we present the visualizations of the MNIST dataset that were constructed by PCA, the $28 \times 28 - 500 - 500 - 2000 - 2$ autoencoder, and the $28 \times 28 - 500 - 500 - 2000 - 2$ parametric t-SNE network. The visualizations were constructed by transforming the MNIST test images, that were held out during training, to two dimensions using the trained models. The results reveal the strong performance of parametric t-SNE compared to PCA and autoencoders. In particular, the PCA visualization mixes up most of the natural classes in the data. The autoencoder outperforms PCA, but cannot successfully separate the classes 4, 9, 6, and 8. In contrast, parametric t-SNE clearly separates all classes (although the visualization contains some debris that is mainly due to the presence of distorted digits in the data). Parametric t-SNE does not only outperform other parametric techniques, it even outperforms the non-parametric techniques for which we presented visualizations in Figure 3.2.

In Figure 4.3, we present visualizations of the characters dataset that were constructed by PCA, the multilayer autoencoder, and the parametric t-SNE network. Again, the visualization only depicts test images that were held out during the training of the dimensionality reduction techniques. The results reveal that parametric t-SNE reveals the natural clusters in the data much better than PCA and autoencoders. The separation between the classes on the parametric t-SNE visualization is not perfect, but this is mainly due to the fact that it is impossible to discriminate between, for instance, the character 'O' and the numeral '0' if no context is available. The strong performance of parametric t-SNE compared to PCA and autoencoders is also revealed by gen-

¹In the notation of the network structure, each number represents the number of units in a layer. The numbers are ordered such that the first number corresponds to the number of input units, whereas the last number indicates the number of units in either the output layer of the parametric t-SNE network, or the middle layer of the autoencoder.

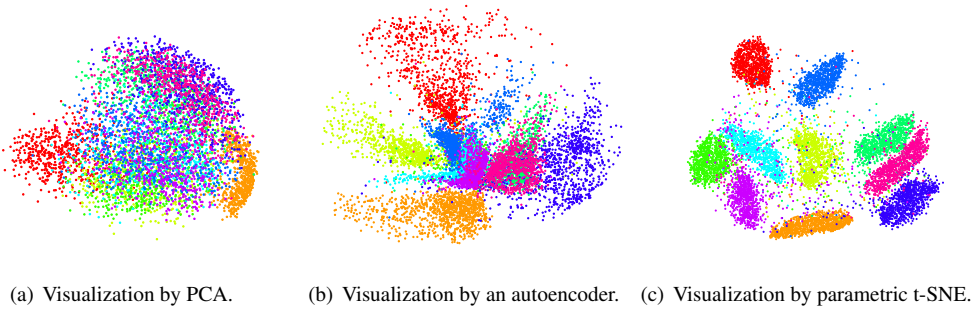


Figure 4.2 Visualizations of 10,000 digits from the MNIST dataset by parametric dimensionality reduction techniques.

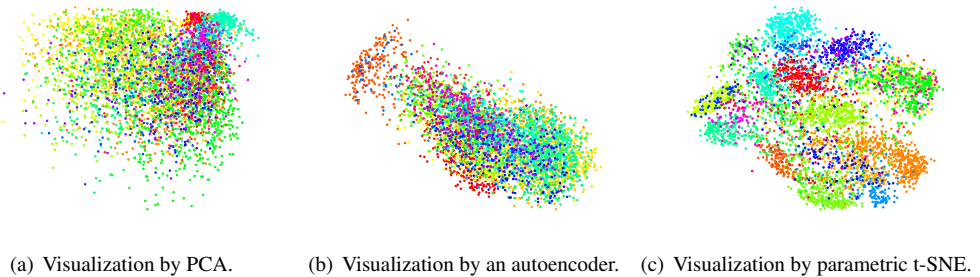


Figure 4.3 Visualizations of 5,000 characters from the characters dataset by parametric dimensionality reduction techniques.

eralization errors of nearest neighbor classifiers trained and tested on the low-dimensional data representations.

In Table 4.1, we present the generalization errors of nearest neighbor classifiers that were trained on the low-dimensional representations obtained from the three parametric dimensionality reduction techniques (using three different dimensionalities for the low-dimensional space). The generalization errors were measured on test data that was held out during the training of both the dimensionality reduction techniques and the classifiers. The corresponding trustworthinesses $T(12)$ of the embeddings are presented in Table 4.2. In both tables, the best performance in each experiment is typeset in boldface. From the results presented in Table 4.1 and 4.2, we can make the following two observations.

First, we observe that parametric *t*-SNE performs better or on par with the other techniques in most experiments. In particular, the performance of parametric *t*-SNE is strong if the dimensionality of the low-dimensional space is not sufficiently large to accommodate for all properties of the data. In this case, the heavy tails of the distribution of parametric *t*-SNE in the low-dimensional space push the natural clusters in the data apart, whereas PCA and autoencoders construct embeddings in which these natural clusters (partially) overlap. The high trustworthinesses of the parametric *t*-SNE embeddings indicate that parametric *t*-SNE preserves the local structure of the data in the low-dimensional space well.

	MNIST			Characters		
	2D	10D	30D	2D	10D	30D
PCA	78.16%	43.03%	10.78%	86.72%	60.73%	20.50%
Autoencoder	66.84%	6.33%	2.70%	82.93%	17.91%	11.11%
Par. t-SNE, $v = 1$	9.90%	5.38%	5.41%	43.90%	26.01%	23.98%
Par. t-SNE, $v = d - 1$	9.90%	4.58%	2.76%	43.90%	17.13%	13.55%
Par. t-SNE, learned v	12.68%	4.85%	2.70%	44.78%	17.30%	14.31%

Table 4.1 Generalization errors of 1-nearest neighbor classifiers on low-dimensional representations of the MNIST and characters dataset.

	MNIST			Characters		
	2D	10D	30D	2D	10D	30D
PCA	0.744	0.991	0.998	0.735	0.971	0.994
Autoencoder	0.729	0.996	0.999	0.721	0.976	0.992
Par. t-SNE, $v = 1$	0.926	0.983	0.983	0.866	0.957	0.959
Par. t-SNE, $v = d - 1$	0.927	0.997	0.999	0.866	0.988	0.995
Par. t-SNE, learned v	0.921	0.996	0.999	0.861	0.988	0.995

Table 4.2 Trustworthiness $T(12)$ of low-dimensional representations of the MNIST and characters dataset.

Second, we observe that it is disadvantageous to use a single degree of freedom in the low-dimensional space if that low-dimensional space has more than, say, two dimensions. Our results reveal that it is better to make the number of degrees of freedom v linearly dependent on the dimensionality of the low-dimensional space d , for reasons we already stated above. The results also show that learning the appropriate number of degrees of freedom v leads to similar results. In fact, the learned value of v was often close to $d - 1$ in our experiments.

4.1.2 Discussion

From the results of our experiments, we observe that parametric t-SNE often outperforms two other unsupervised parametric dimensionality reduction techniques, in particular, if the dimensionality of the low-dimensional space is relatively low. These results are due to the main differences of parametric t-SNE compared to PCA and autoencoders, which we discuss below.

The strong performance of parametric t-SNE compared to PCA can be explained from the two main problems of PCA (that were also discussed in Chapter 2 and 3). First, the linear nature of PCA is too restrictive for the technique to find appropriate embeddings for non-linear real-world data. Second, PCA focuses primarily on retaining large pairwise distances in the low-dimensional space (which can be understood from its relation to classical scaling [Williams, 2002]), whereas it is more important to retain the local structure of the data in the low-dimensional space.

The strong performance of parametric t-SNE compared to autoencoders, especially if the low-dimensional space has a relatively low dimensionality, can be understood from the following difference between parametric t-SNE and autoencoders. Parametric t-SNE aims to model

the local structure of the data appropriately in the low-dimensional space, and it attempts to create separation between the natural clusters in the data (by means of the heavy-tailed distribution in the low-dimensional space). In contrast, autoencoders mainly aim to maximize the variance of the data in the low-dimensional space, in order to achieve low reconstruction errors. As a result of the maximization of the variance, autoencoders generally do not construct low-dimensional data representations in which the natural classes in the data are widely separated (as this would decrease the variance of the low-dimensional data representation, and increase the reconstruction error). The relatively poor separation between natural classes in low-dimensional data representations constructed by autoencoders leads to inferior generalization performance of nearest neighbor classifiers compared to parametric t-SNE; in particular, if the dimensionality of the low-dimensional space is relatively low. Moreover, parametric t-SNE provides computational advantages over autoencoders. An autoencoder consists of an encoder part and a decoder part, whereas parametric t-SNE only employs an encoder network. As a result, errors have to be back-propagated through half the number of layers in parametric t-SNE (compared to autoencoders), which gives it an computational advantage over autoencoders (even though the computation of the errors is somewhat more expensive in parametric t-SNE).

A notable advantage of autoencoders is that they provide the capability to reconstruct the original data from its low-dimensional representation in the low-dimensional space. In other words, autoencoders do not only provide a parametric mapping from the data space to the low-dimensional space, but also the other way around. A possible approach to address this shortcoming is to use the decoder part of an autoencoder as a regularizer on the parametric t-SNE network, i.e., to minimize a weighted sum of the parametric t-SNE cost function and the reconstruction error (as is done for non-linear NCA by Salakhutdinov and Hinton [2007]).

As the number of parameters in parametric t-SNE and autoencoders is larger than in PCA, these techniques are likely to be more susceptible to overfitting. However, we did not observe overfitting effects in our experiments, probably because of the relatively large number of instances in our training data. If parametric t-SNE or autoencoders are trained on smaller datasets, it may be necessary to use early stopping [Caruana *et al.*, 2001]

The results of our experiments not only reveal the strong performance of parametric t-SNE compared to PCA and autoencoders, but also provide insight into the nature of the crowding problem. In particular, the results reveal that the severity of the crowding problem depends on the ratio between the intrinsic dimensionality of the data and the dimensionality of the low-dimensional space. The number of degrees of freedom v should thus be set accordingly. We suggested to treat v as a parameter that has to be learned as well, and although competitive, learning v does not always outperform a setting in which v depends linearly on the dimensionality of the low-dimensional space. Presumably, this observation is due to the following. When v is learned, it is set in such a way as to ‘fill up’ the low-dimensional space. This decreases the Kullback-Leibler divergence that parametric t-SNE minimizes, because it provides more space to model the local structure of the data appropriately (recall that the cost function focuses on retaining local structure). Although the ‘filling up’ of the space is advantageous for modeling the local structure of the data (as is illustrated by the high trustworthinesses which were obtained when v is learned), it has a negative influence on the generalization performance of nearest neighbor classifiers on the low-dimensional data representation, as it decreases the separation between the natural clusters in the data.

4.2 Multiple-maps t-SNE

In our discussions on (1) dimensionality reduction techniques in Chapter 2, (2) t-SNE in Chapter 3, and (3) parametric t-SNE in Section 4.1, we made an assumption that has remained implicit until now. We assumed that the space in which the dimensionality reduction techniques construct their low-dimensional embeddings has a metric nature. In other words, we assumed that the low-dimensional space in which we embed the data obeys the four metric axioms: (1) non-negativity of distances, (2) identity of indiscernibles, (3) symmetry of distances, and (4) the triangle inequality. If we denote the distance between object A and object B by $d(A, B)$, the four metric axioms [Munkres, 2000] are given by

$$d(A, B) \geq 0, \quad (4.5)$$

$$d(A, B) = 0 \text{ iff } A = B, \quad (4.6)$$

$$d(A, B) = d(B, A), \quad (4.7)$$

$$d(A, C) \leq d(A, B) + d(B, C). \quad (4.8)$$

Until now, the assumption that the low-dimensional data representations reside in a metric space was no limitation, since the input data (such as images) resided in a metric space itself. However, the input into a variant of multidimensional scaling such as t-SNE may equally well be a collection of objects of which the pairwise similarities do not obey the four metric axioms. For instance, the collection of objects may be a set of words, and the pairwise similarities may be co-occurrences or association values between the words. Such semantic similarities are likely to be non-metric, as a result of which they cannot successfully be embedded in a low-dimensional space that obeys the four metric axioms. Traditional multidimensional scaling techniques such as classical scaling [Torgerson, 1952], Sammon mapping [Sammon, 1969], and t-SNE will thus perform inferior when used to model, e.g., semantic similarities².

The metric axioms give rise to three limitations of metric spaces in terms of the similarities that can be represented in the space: (1) the triangle inequality induces transitivity of similarities, (2) the number of points that can have the same point as their nearest neighbor is limited³, and (3) similarities have to be symmetric. We discuss the three limitations of metric spaces in more detail below. In the discussion, we assume that the input objects are words that are described in terms of their semantic similarities to other words.

The first limitation of metric spaces is the result of the triangle inequality. The triangle inequality basically states that in a metric space, if point A is close to point B and B is close to point C , point A has to be close to C as well. In practice, this constraint may well be violated by objects such as words. Consider, for instance, the word *tie*, which is semantically similar to a word such as *tuxedo*. In a low-dimensional metric map of words, the two words should thus be modeled close to each other. However, the word *tie* has more than one meaning, as a result of which it also is semantically similar to a word such as *knot*. The word *tie* should thus be modeled

²The same observation holds for the similarity choice model [Shepard, 1957; Luce, 1963], which is actually very similar to SNE.

³This is not the only limitation on the neighborhood relations of points in a metric space. For instance, the maximum number of equidistant points in a metric space is limited as well.

close to *knot* as well. As a result, the words *tuxedo* and *knot* are modeled close together in the low-dimensional metric map, even though the words exhibit no obvious semantic similarity. The triangle inequality thus induces transitivity of semantic similarities, which may be undesired.

The second limitation of a metric space is that only a limited number of points can have the same point as their nearest neighbor. For instance, in a two-dimensional space, maximally five points can have the same point as their nearest neighbor (by arranging them in a pentagon that is centered onto the point). As a result, it is not possible to model the large number of similarities of ‘central’ objects with other objects appropriately in a low-dimensional metric map. This is problematic because many collections of objects are characterized by a high ‘centrality’, i.e., by the presence of objects that are similar or related to a large portion of the other objects [Tversky and Hutchinson, 1986]. For instance, a collection of words is typically characterized by a high centrality. The high centrality of word collections can be understood from the properties of semantic networks: like many other networks, semantic networks are scale-free networks that are characterized by a high clustering coefficient [Steyvers and Tenenbaum, 2005]. The high clustering coefficient indicates the presence of ‘central’ words. For instance, large numbers of mammals are semantically similar to the word *mammal*, as a result of which all mammals would like to have the word *mammal* as a near neighbor in a low-dimensional map. However, because in a two-dimensional metric map only five points can have the same nearest neighbor, it is impossible to model the large number of mammals in such a way that they all have the word *mammal* as a near neighbor.

The third limitation of metric spaces is that similarities in these spaces are symmetric, whereas the similarities between objects in the world are often asymmetric. Tversky illustrated this problem with a famous example on the similarity between China and North Korea [Tversky and Hutchinson, 1986]: “People typically have the intuition that North Korea is more similar to China than China is to North Korea”. The reason for the asymmetry in these similarities is that a person’s representation of China typically comprises a large number of features, of which only some features are shared with North Korea, whereas the representation of North Korea involves a small number of features, most of which are shared with China. Loosely speaking, we could state that ‘specific’ objects are more similar to ‘general’ objects than the other way around. A metric map cannot represent such asymmetric similarities appropriately.

The three limitations of low-dimensional metric spaces discussed above led Tversky to argue against techniques for multidimensional scaling (such as t-SNE), since the fundamental limitations of metric space make multidimensional scaling techniques not suitable as computational models for semantic representation [Tversky and Hutchinson, 1986]. In the remainder of this section, we present a variant of t-SNE that constructs multiple maps that complement each other. We show that the resulting technique, called multiple-maps t-SNE, can avoid the three limitations of low-dimensional metric spaces. The presented technique thus resolves the arguments of Tversky against multidimensional scaling techniques, and as a result, it gives rise to an interesting computational cognitive model for semantic representation. The section compares the characteristics of multiple-maps t-SNE to that of three alternative computational models for semantic representation.

The outline of the remainder of this section is as follows. In 4.2.1, we present the multiple maps variant of t-SNE and we explain why it is not hampered by the three limitations of metric spaces discussed above. In 4.2.2, we present our experiments with multiple maps t-SNE, which

show that our technique is capable of addressing all three problems of multidimensional scaling techniques.

4.2.1 Formulating t-SNE using multiple maps

The probabilistic nature of t-SNE allows for natural extensions to variants that construct multiple maps, and not a single map. This is a desirable property, because the use of multiple maps allows for the three limitations of metric spaces to be avoided [Cook *et al.*, 2007]. In this section, we first present a multiple maps version of *asymmetric* t-SNE. Subsequently, we discuss how the presented technique avoids the three limitations of low-dimensional metric spaces.

Multiple-maps t-SNE constructs a collection of M low-dimensional maps, all of which contain all N datapoints. In each map with index m , the point with index i has a so-called mixing proportion $\pi_i^{(m)}$ that measures the ‘weight’ of point i in map m . Because of the probabilistic interpretation of the technique, we require that the mixing proportions of a single point in all maps have to sum up to 1. In other words, the mixing proportions $\pi_i^{(m)}$ are constrained to make sure that $\sum_m \pi_i^{(m)} = 1$. We define the conditional probability distribution $q_{j|i}$, which represents the similarity between the objects with index i and j under the model, as the weighted sum of the pairwise similarities between the points corresponding to the objects i and j over all M maps. Mathematically, we define $q_{j|i}$ in multiple maps t-SNE as

$$q_{j|i} = \frac{\sum_m \pi_i^{(m)} \pi_j^{(m)} \left(1 + \|\mathbf{y}_i^{(m)} - \mathbf{y}_j^{(m)}\|^2\right)^{-1}}{\sum_{m'} \sum_{k \neq i} \pi_i^{(m')} \pi_k^{(m')} \left(1 + \|\mathbf{y}_i^{(m')} - \mathbf{y}_k^{(m')}\|^2\right)^{-1}}. \quad (4.9)$$

Note that in the remainder of this section, we use the *asymmetric* definition of pairwise similarity $q_{j|i}$, and not the symmetric q_{ij} that we used in Chapter 3. The cost function of multiple maps t-SNE is given by the cost function presented in Equation 3.3. However, it is now optimized with respect to the $N \times M$ low-dimensional map points $\mathbf{y}_i^{(m)}$ and with respect to the $N \times M$ mixing proportions $\pi_i^{(m)}$.

Because the mixing proportions $\pi_i^{(m)}$ for a single point i should sum to 1 over all maps, direct optimization of the cost function C with respect to the parameters $\pi_i^{(m)}$ is tedious. To avoid this problem, we represent the mixing proportions $\pi_i^{(m)}$ in terms of mixing weights using an idea that is similar to that of softmax units, which are commonly used in neural networks [Bridle, 1989]. The mixing proportions $\pi_i^{(m)}$ are represented in terms of the mixing weights $w_i^{(m)}$ as follows

$$\pi_i^{(m)} = \frac{e^{-w_i^{(m)}}}{\sum_{m'} e^{-w_i^{(m')}}}. \quad (4.10)$$

By defining the mixing proportions in this way, they are guaranteed to be positive and to sum up to 1, as a result of which the minimization of the cost function can be performed with respect to the unconstrained mixing weights $w_i^{(m)}$. This significantly simplifies the optimization of the cost function using a gradient descent method.

The gradients that are necessary to perform the minimization of the cost function are derived in Appendix E. In our experiments, we used the same optimization procedure as in Chapter 3,

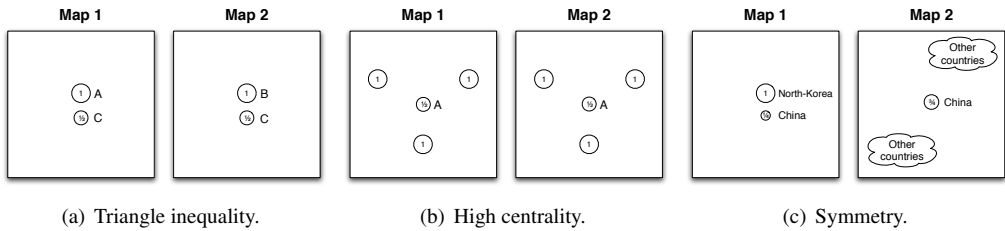


Figure 4.4 Illustration of how multiple-maps t-SNE can resolve the three weaknesses of metric spaces.

i.e., we used a simple gradient descent method that employs: (i) an additional momentum term to stabilize the gradient search and (ii) the early exaggeration method described in 3.2.4.

Multiple-maps t-SNE has three main advantages over single map multidimensional scaling techniques such as t-SNE: (1) it can represent similarities for which the triangle inequality does not hold, (2) it can represent data with high centrality, and (3) it can represent asymmetric similarities. We separately discuss the three advantages of multiple-maps t-SNE over single-map multidimensional scaling techniques below.

1) *Triangle inequality*. Consider our introductory example with the word *tie*, which is semantically similar to *tuxedo* and to *knot*. The word *tie* should be modeled close to *tuxedo* and *knot*, but the words *tuxedo* and *knot* should not be modeled close to each other. In contrast to single-map multidimensional scaling techniques, multiple-maps t-SNE can appropriately model this example as follows.

Assume we have three datapoints A , B , and C that are embedded into two maps (see Figure 4.4(a)). Multiple-maps t-SNE can give point A a mixing proportion of 1 in the first map, point B a mixing proportion of 1 in the second map, and point C a mixing proportion of $\frac{1}{2}$ in both maps, and it can give all three points have the same spatial location in both maps. Then, the pairwise similarity between point A and C is equal to $1 \times \frac{1}{2} = \frac{1}{2}$, and the pairwise similarity between point B and C is also equal to $\frac{1}{2}$. However, the pairwise similarity between point A and B is 0, because the points A and B have no mixing proportion in each others maps. Hence, the representation constructed by multiple-maps t-SNE does not satisfy the triangle inequality, as a result of which it can model intransitive semantic similarities such as our example with *tie*, *tuxedo*, and *knot*.

2) *High centrality*. In a metric space, only a limited number of points can have the same point as their nearest neighbor, as a result of which it is not possible to model the large number of similarities of ‘central’ objects with other objects appropriately in a low-dimensional metric map. Data with high centrality can be modeled appropriately by multiple-maps t-SNE, essentially, because multiple maps provide much more space than a single map. We illustrate the capability of multiple-maps t-SNE to model data with high centrality by an example.

Assume we have six objects that all have the same ‘central’ object A as their most similar object. In a single map, only five of the objects can be modeled in such a way that they have the low-dimensional model of object A as their nearest neighbor. In contrast, when two maps are

available, the data can be modeled in such a way that the low-dimensional models of the all six objects have the model of object *A* as their nearest neighbor. For instance, this can be achieved by giving *A* a mixing proportion of $\frac{1}{2}$ in both maps, modeling the first three objects close to the model of object *A* in the first map with mixing proportion 1, and modeling the remaining three objects close to the model of object *A* in the second map with mixing proportion 1. This example is illustrated in Figure 4.4(b).

Multiple-maps t-SNE can thus successfully model ‘central’ objects in the data, such as the *mammal* in our introductory example. The number of points that can have the same point as their nearest neighbor in multiple-maps t-SNE depends on the number of maps and on the dimensionality of these maps.

3) *Symmetry*. A metric low-dimensional map constructed by a single map multidimensional scaling technique (such as t-SNE) cannot appropriately model asymmetric similarities, such as the similarity between China and North Korea. In contrast, asymmetric similarities between objects can be modeled by multiple maps t-SNE. We illustrate this capability using Tversky’s famous example on the similarity between China and North-Korea, which may be modeled by multiple maps t-SNE as follows.

Assume (1) that we have two maps, (2) that North Korea has a mixing proportion of 1 in the first map and a mixing proportion of 0 in the second map, and (3) that China has a mixing proportion of $\frac{1}{4}$ in the first map and a mixing proportion of $\frac{3}{4}$ in the second map. In addition, assume (4) that North Korea and China are mapped close to each other in map 1, and (5) that China is modeled close to other countries in map 2. This example is illustrated in Figure 4.4(c). In the example, North Korea is modeled as very similar to China, whereas China is much less similar to North Korea, because it shares a large number of features with other countries as well. The actual similarity between China to North Korea under the model depends on the locations and mixing proportions of the other countries in both maps, i.e., on the amount of features that China shares with North Korea, relative to the amount of features that China shares with other countries. Nevertheless, the representation constructed by multiple-maps t-SNE successfully models asymmetric similarities.

4.2.2 Experiments

Above, we introduced multiple-maps t-SNE and we explained how multiple-maps t-SNE can overcome the limitations of metric spaces that hamper techniques for multidimensional scaling. In this subsection, we present experiments with multiple-maps t-SNE, in which we employ the technique to visualize a set of word association data. We selected a word association dataset for our experiments, because word associations cannot be modeled well by single-map multidimensional scaling techniques: word association data typically contains intransitive semantic relations, central concepts, and asymmetric semantic similarities. Below, we discuss the setup of the experiments and the results of the experiments separately.

Experimental setup

In the evaluation of the performance of multiple-maps t-SNE, we performed experiments in which we visualize the Florida State University word association dataset [Nelson *et al.*, 1998]. The Florida State University word association dataset contains association data for 10,617 words, of which 5,019 were used as input stimuli. The word association data was gathered as follows. Human subjects were given one of the 5,019 words as input stimulus and were instructed to write down the first word that came to mind that they associated with the input stimulus. In total, over 6,000 participants produced approximately 750,000 responses to the 5,019 words. Each subject was presented with 100 to 120 randomly selected words, as a result of which on average 149 subjects produced a response to a single word. In the processing of the responses, the counts for singular and plural forms of the same word were pooled and the majority response was employed as a label for the pooled counts. After normalization of the word association counts per word, a condition probability $p_{j|i}$ is obtained that measures the probability that a human subject produces word j as response after being presented with word i as input stimulus. In other words, the conditional probability $p_{j|i}$ represents the probability that a human subject associates word j with word i . The conditional probabilities $p_{j|i}$ are used as the input for multiple-maps t-SNE. In the computation of the conditional probabilities $p_{j|i}$, we ignored all words that were given as a response but that were not used as an input stimulus (because multiple-maps t-SNE expects a square similarity matrix as input).

The word association dataset has three characteristics that make it difficult to visualize the data using single-map multidimensional scaling techniques. First, it contains numerous examples of intransitive semantic relations, such as our introductory example with *tie*, *tuxedo*, and *knot*. Second, it contains a number of fairly ‘general’ words that have semantic relations with many other words. The high centrality of the Florida State University word association dataset is reflected in the high clustering coefficient of the dataset [Steyvers and Tenenbaum, 2005]. The most central word in the data is the word *field*, which has a semantic similarity to 33 other words in the data. Third, the word association data contains numerous examples of asymmetric similarities. For instance, the probability that a human thinks of the word *cut* after being presented with the word *scissors* is 0.879, whereas the probability that a human thinks of *scissors* after being presented with the word *cut* is only 0.034.

In our experiments, we set the number of maps to 40. The dimensionality of each map is set to 2 in order to facilitate the visualization of the resulting maps. The optimization is performed using 2,000 iterations of gradient descent, in which we employed an additional momentum term. The momentum term was set to 0.5 in the first 250 iterations, and to 0.8 afterwards. The initial learning rate was set to 0.1, and the learning rate was updated after every iteration using the adaptive learning weight scheme described by Jacobs [1988]. The results are visualized in an annotated scatter plot, in which the size of a dot represents the mixing proportion of a word in a specific map. To prevent the visualizations from being too cluttered, datapoints with a small mixing proportion (below 0.1) were removed from the visualization. To increase the readability of the plots, the annotations in the scatter plot were manually aligned to minimize the overlap between the annotations, while keeping the word labels close to their corresponding point in the map.

Results

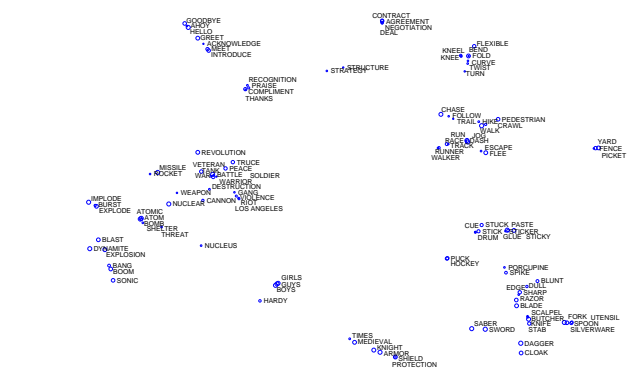
Figure 4.5 presents the results of our experiments on the word association data. The figure shows 6 of the 40 maps that were constructed by multiple-maps t-SNE. The results reveal that the maps retain the similarity structure of the association data well⁴. Because the data contains too many ‘topics’, a single map does not generally visualize a single topic. Instead, most maps reveal two or three main topics, as well as some very small separate structures. For instance, map 4.5(d) visualizes the topics *sports* and *clothing*, and it shows small local structures that are related to, e.g., the Statue of Liberty: *monument - statue - liberty - freedom*.

The results reveal how multiple-maps t-SNE avoids the limitations of low-dimensional spaces. In particular, multiple-maps t-SNE successfully models intransitive similarities of words. For instance, the semantic relation of the word *tie* with words such as *suit*, *tuxedo*, and *prom* is modeled in map 4.5(a), whereas in map 4.5(d), the semantic relation of the word *tie* with *rope* and *knot* is modeled. In addition, map 4.5(e) reveals the semantic relation of *tie* with words such as *ribbon* and *bow*. As a second example, the semantic relation of the word *cheerleader* with various kinds of sports is modeled in map 4.5(d), whereas map 4.5(f) reveals the association of the word *cheerleader* with words such as *gorgeous*, *beauty*, and *sexy*. A third example is the word *monarchy*, which is modeled close to words that are related to royalty such as *king*, *queen*, *crown*, and *royal* in map 4.5(c). In map 4.5(f), the word *monarchy* is modeled close to other governmental forms such as *oligarchy*, *anarchy*, *democracy*, and *republic*.

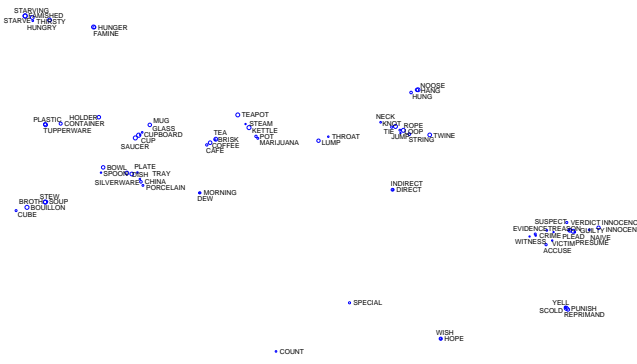
From the results of the experiments, it is hard to assess whether multiple maps t-SNE was successful in modeling concepts with high centrality. We believe that establishing whether multiple maps t-SNE successfully model central concepts can best be done on artificial data, such as the example in Figure 4.4(b). We leave such an experiment for future work.

The results of our experiments do reveal how multiple-maps t-SNE represents asymmetric pairwise similarities. For instance, map 4.5(c) reveals that the word *dynasty* is more often associated with the word *China* than the other way around. In map 4.5(c), the representations of both words are close to one another, however, the word *China* has a much smaller mixing proportion than *dynasty* in map 4.5(c). As a result, the denominator of Equation 4.9 is higher for *China* than for *dynasty*, which implies that *dynasty* is closer to *China* than the other way around.

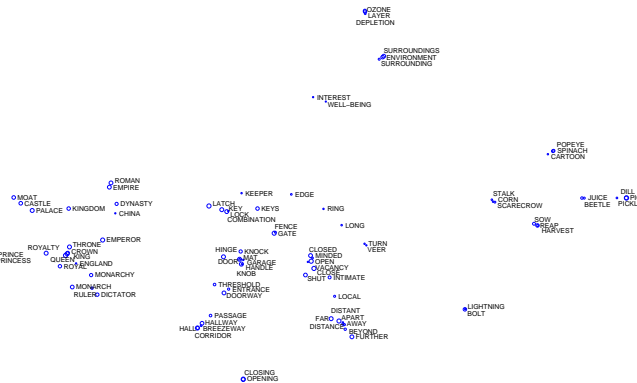
⁴Please note that the word association data does not exactly capture semantic similarity. For instance, in map 4.5(f), the word *beauty* shown next to the word *beast*, revealing the word association that results from a famous Disney movie.



(a) Map 1.



(b) Map 2.



(c) Map 3.

Figure 4.5 Maps of the word association dataset constructed by multiple-maps t-SNE (a-c). Because of space limitations, we only show 6 of the original 40 maps.

4.2.3 Discussion

In the previous subsection, we presented the results of experiments that reveal the merits of multiple-maps t-SNE over single-map multidimensional scaling techniques such as t-SNE. Multiple-maps t-SNE may have applications in information retrieval and visualization. Moreover, the quality of the visualizations constructed by multiple maps t-SNE suggests that it may provide a basis for computational cognitive models of semantic representation that overcome many of the problems of cognitive models based on semantic spaces [Shepard, 1968; Tversky and Hutchinson, 1986; Landauer and Dumais, 1997]. Below, we compare the theoretical properties of multiple maps t-SNE with those of three alternative cognitive models for semantic representation: (1) semantic space models, (2) semantic networks, and (3) Bayesian latent variable models.

1) *Semantic space models.* Semantic space models are similar to multiple maps t-SNE in that they represent semantic concepts as points in a space in such a way, that similar concepts are represented close together in the space. In other words, semantic space models are based on the idea of implementing a second-order isomorphism between the representation space and the concepts in the world [Edelman and Duvdevani-Bar, 1997], which means that words with similar semantics should have a similar representation in the space.

Traditionally, multidimensional scaling models have been the most popular semantic space models [Torgerson, 1952; Shepard, 1968; Sammon, 1969], but these models are hampered by the limitations of metric spaces that we discussed in Section 4.2. For classical scaling (see 2.2.1), it is possible to exploit structure from the eigenvectors that correspond to the *negative* eigenvalues of the Gram matrix when modeling non-metric similarities, as these eigenvectors contain structural information on the metricity violations in the pairwise dissimilarity matrix [Laub and Müller, 2004; Laub *et al.*, 2007]. However, such an approach is limited in that it can only construct two maps: one map that corresponds to the positive part of eigenspectrum and one map that corresponds to the negative part of the eigenspectrum.

More recently, the Latent Semantic Analysis (LSA) model has gained popularity. LSA is a model for semantic representation that was originally designed for use in information retrieval systems [Landauer and Dumais, 1997]. It computes a low-rank approximation of a word association or word co-occurrence matrix by means of singular value decomposition (SVD). The most important output of LSA is formed by the k principal left-singular vectors of the low-rank approximation, where the importance of the singular vectors is determined by their corresponding singular values. The left-singular vectors provide a spatial representation for the words in the data in an orthogonal basis spanned by k vectors, hence, they represent words as points in the k -dimensional metric space \mathbb{R}^k . Semantic similarity in this space is typically represented in terms of the cosine distance between word vectors, as a result of which semantic similarities under the LSA model obey all metric axioms. Therefore, LSA (and its probabilistic counterpart [Hofmann, 1999]) is not fundamentally different from other semantic representation models that rely on second-order isomorphic representations. LSA is thus subject to all of the objections against multidimensional scaling that were formulated by Tversky⁵ [Tversky and Hutchinson, 1986], as

⁵We are not the first authors to note the limitations of Latent Semantic Analysis. See for a more extensive coverage of the limitations of LSA, e.g., [Griffiths *et al.*, 2007].

a result of which multiple-maps *t*-SNE has important advantages over (probabilistic) LSA. In contrast to (probabilistic) LSA, multiple-maps *t*-SNE can successfully model intransitive similarities and asymmetric similarities between objects.

Other important semantic space models are models based on distributed representations that are typically employed in connectionist models of semantic representation [McClelland and Rumelhart, 1981; Kawamoto, 1993; Plaut, 1997; Rodd *et al.*, 2004]. Distributed representations are fairly similar to multiple-maps *t*-SNE in that they allow an object to be represented by multiple points. However, an important problem of the distributed representations is that automatically extracting a distributed semantic representation from text involves significant computational challenges, such as deciding how many senses each word should have and when those senses are being used. Until now, these problems have been alleviated by constructing the networks based on data that consists of labeled pairs of words and their meanings. In contrast, multiple-maps *t*-SNE provides a way to learn automatically a semantic representation from word associations (that can, in turn, be automatically extracted from text corpora), and infers from the data how many senses each word has. Herein, the only restriction is that the number of senses for a single word cannot exceed the predefined number of maps, but it is unlikely that this restriction is violated if a sensible number of maps is employed. This property of multiple-maps *t*-SNE gives it an important advantage over current connectionist models for semantic representation.

2) *Semantic networks.* Semantic associative networks provide an intuitive way to model semantic similarities, and they provide simple solutions to problems such as word prediction, word disambiguation, and gist extraction [Collins and Loftus, 1975]. A semantic network consists of nodes that represent the words, and edges that represent the semantic similarities between the two words that the edges connect. When a word is observed, the node that corresponds to this word is activated. The resulting activation spreads through the semantic network, thereby activating nodes that are nearby in terms of the diffusion distance through the network. The strength of the activations in the nodes represents the semantic similarity of their corresponding words with the observed word.

Activations in undirected semantic networks can readily be represented in a distributed semantic representation [Hinton, 1981; Shastri and Ajjanagadde, 1993], and as a result, an undirected semantic network can be converted into a semantic space model using a bijective mapping. The semantic space corresponding to an undirected semantic network typically has a very high dimensionality, as a result of which the model has no problems with representing ‘central’ concepts. However, undirected semantic networks cannot represent asymmetric or intransitive semantic relations, because they obey the symmetry axiom and the triangle inequality, respectively. The former problem can be overcome by defining semantic networks as directed graphs, in which the weight of an edge from *A* to *B* may be different from the weight of the edge between *B* and *A*, causing similarities in the network to become asymmetric. However, this does not resolve problems with intransitive similarities. If node *A* has a strong connection to node *B*, and node *B* has a strong connection to node *C*, activation from node *A* will spread to node *C*, which makes *A* and *C* semantically related under the model. Multiple-maps *t*-SNE thus has significant advantages over models based on semantic networks, in particular, because it can represent asymmetric similarities.

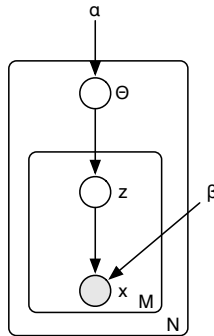


Figure 4.6 Generative process of Latent Dirichlet Allocation.

3) *Bayesian latent variable models.* Recently, Bayesian latent variable models that originate from information retrieval have been proposed as computational cognitive models for semantic representation [Griffiths *et al.*, 2007]. The most important examples of such models are the so-called *topic models*. Recently proposed topic models include Latent Dirichlet Allocation [Blei *et al.*, 2003], the author model [McCallum, 1999], the author-topic model [Rosen-Zvi *et al.*, 2004], and the author-topic-recipient model [McCallum *et al.*, 2004]. Because of the popularity of Latent Dirichlet Allocation (LDA)⁶, we will focus on that model here. However, our discussion also holds for many other Bayesian latent variable models.

LDA was originally developed to model large text corpora. The key idea of LDA is that each word x has a topic z that is drawn from a topic distribution θ that is specific for a document. The graphical model of LDA is shown in Figure 4.6. The corresponding underlying generative process is given by

- For each of the N documents in the corpus:
 - Choose a topic distribution $\theta \sim \text{Dirichlet}(v)$
 - For each of the M words in the document:
 - * Choose a topic $z \sim \text{Multinomial}(\theta)$
 - * Choose a word $x \sim \text{Multinomial}(\beta_z)$

The latent variables in LDA are formed by: (i) k multinomial distributions z over all words and (ii) a distribution θ over these multinomial distributions⁷. The k multinomial distributions z can be viewed upon as topics, and each topic has its own multinomial distribution over words. The variable k is a parameter that sets the number of topics that is employed in the semantic representation. It may either be set by the user, or it may be learned from the data using non-parametric Bayesian techniques [Blei *et al.*, 2004; Teh *et al.*, 2004].

Under a topic model, two words can be viewed upon as semantically related if they both have a high probability under at least one of the k topics [Griffiths *et al.*, 2007]. This provides topic

⁶Please note that here, the abbreviation LDA refers to Latent Dirichlet Allocation, and not to Linear Discriminant Analysis.

⁷The distribution over the multinomial distributions over all words is parametrized by means of a Dirichlet distribution, which is the conjugate prior of the multinomial distribution (see, e.g., [Gelman *et al.*, 1995]).

models with the same desirable properties that multiple-maps t-SNE has. In particular, a topic model is capable of modeling intransitive semantic similarities in different topics. Analogous to our example with *tie*, *tuxedo*, and *knot*, in LDA, *tie* and *tuxedo* could be given a high probability in one topic and *tie* and *knot* could be given high probability in another topic, which would not make *tuxedo* similar to *knot* under the model. In the same way, LDA can model ‘central’ objects by giving them a high probability in a large number of topics, which automatically gives rise to asymmetric similarities. The only requirement is that (as in multiple-maps t-SNE) sufficient topics are available to model the required centrality. The topics in LDA can be thought of as an equivalent for the maps in multiple maps t-SNE.

The main difference between topic models and multiple-maps t-SNE is that, in contrast to LDA, multiple-maps t-SNE can (1) be used to model word association data and (2) capture subtle semantic structure in the spatial structure of the maps. The first capability may be relevant depending on the input data that is available. The merits of the second capability are illustrated, for instance, in the ‘sports’ cluster in Figure 4.5(d), where the subtle semantic difference between physical sports such as *football*, *baseball*, and *volleybal*, and mental sports such as *chess*, *checkers*, and *poker* is captured in the spatial structure of the cluster (from left to right). In addition, multiple-maps t-SNE has the advantage that it can model small semantic structures that are not closely related to other semantic structures, such as the *Popeye - spinach - cartoon* cluster in Figure 4.5(c), without resorting to the construction of a new map or topic.

A minor disadvantage of multiple-maps t-SNE is that, like all other second-order isomorphic models, it implicitly assumes that every concept is at least similar to some other concept. In multiple-maps t-SNE, an object that is not similar to any other object can only be modeled by placing its corresponding map points infinitely far away from the other map points, or by constructing a map in which all other concepts have zero mixing proportion. In contrast, topic models can easily give a concept zero probability under all topics, as a result of which they have a more natural way to model objects that are not similar to any other object.

A remaining relevant question is to what types of data multiple-maps t-SNE can be applied. Clearly, multiple-maps t-SNE is good at visualizing word association data, and we surmise it performs equally well on other datasets that have a high clustering coefficient. However, multiple-maps t-SNE is not very well capable of modeling, e.g., the handwritten character datasets we employed in Section 4.1. On a handwritten characters dataset, multiple-maps t-SNE will exploit the additional space that the multiple maps provide to model the local structure of the data better, because the cost function focuses on modeling the local data structure. As a result, all maps will have a similar global layout, but each of the maps will model only parts of the local structure of the data, which is not very informative for human observers. Multiple-maps t-SNE is thus primarily tailored to modeling datasets that comprise large numbers of relatively small clusters (i.e., on modeling data that gives rise to a scale-free similarity network), such as human similarity judgements that are often collected in cognitive psychology.

4.3 Chapter conclusions

In the chapter, we presented experiments with two variants of t-SNE that are applicable in learning settings in which the original t-SNE does not apply. First, we presented a parametric version of t-SNE that can be employed in learning settings in which generalization to unseen test data or reconstruction is required. The results of our experiments reveal the strong performance of parametric t-SNE compared to two other unsupervised parametric dimensionality reduction techniques. Second, we presented a variant of t-SNE that constructs multiple maps instead of a single map, as a result of which it is capable of modeling objects of which the similarities are intransitive or asymmetric, or that may have a high centrality. Our experiments revealed the strong performance of the multiple maps t-SNE model in representing semantic similarities, which suggests it is a suitable computational cognitive model for semantic representation. Then, we performed a theoretical comparison of multiple-maps t-SNE with alternative computational models for semantic representation, from which we may conclude that multiple-maps t-SNE has significant advantages over semantic space models, and shares many desirable properties with the recently proposed topic models.

5 Texture features

Contents Up to this point, the thesis has focused on resolving the dimensionality problem of image-space representations, but the variance problem has remained unaddressed. In this chapter, we describe how the variance problem can be addressed in images that contain textured surfaces. In order to (partially) answer research question RQ2, the chapter provides an overview of the four main types of texture features. We describe the rationale of the four types of texture features and we discuss their main advantages and disadvantages. The features presented in this chapter form the basis for the new texture features that we develop in Chapter 6.

Outline The chapter discusses the four main types of texture features in four separate sections. First, we present and discuss graylevel co-occurrence features (in 5.1). Second, features based on Markov Random Fields are discussed (in 5.2). Third, we discuss the large class of filter-based features (in 5.3). Fourth, the chapter presents texton-based features (in 5.4). Section 5.5 concludes the chapter.

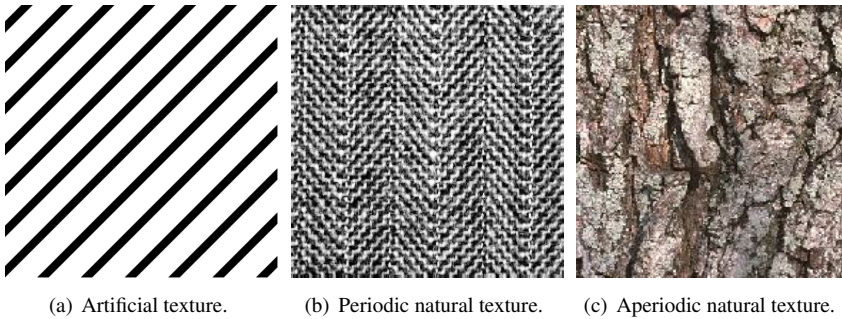


Figure 5.1 Examples of textures.

In this chapter, we shift our focus from feature extraction using dimensionality reduction techniques towards feature extraction using texture features. Texture features aim to model the (potentially colored) texture of the surface of an object. We define texture as a homogeneous structure on a surface that consists of repeated elements which may be subject to randomness in, e.g., their location and orientation, and which may contain noise (often assumed to be additive Gaussian noise). Texture can be subdivided into two main types: (i) artificial texture and (ii) natural texture. In artificial textures, the repeated elements are neither subject to randomness in location or orientation, nor are they distorted by noise. An example of such an artificial texture is given in Figure 5.1(a). In contrast, randomness plays an important role in natural textures. Natural textures can be further subdivided into periodic and aperiodic textures. Periodic textures contain regularly repeating elements, whereas in aperiodic textures the elements do not occur regularly. Examples of periodic and aperiodic natural textures are given in Figure 5.1(b) and 5.1(c).

In natural textures, color and texture are highly interrelated. Changing the texture of a surface usually changes the perceived color of a surface, and a change of the color of texture often changes the visual appearance of the texture. Moreover, the visual appearance of texture is highly dependent on, for instance, changes in lighting of the textured surface. This is illustrated for a natural texture in Figure 5.2. Ideally, the texture features extracted from both images in Figure 5.2 are identical (because they represent the same texture), which makes the extraction of features from textured surfaces a challenging task.

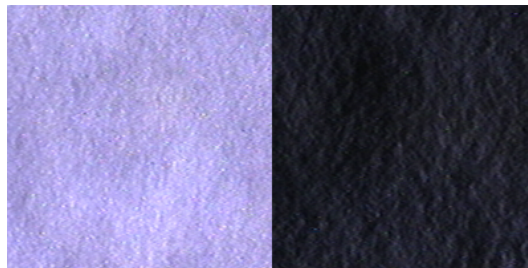


Figure 5.2 Visual appearance of a texture photographed under different lighting conditions.

The main feature extraction approaches for texture modeling can roughly be subdivided into four main types: approaches based on (1) measuring graylevel co-occurrences, (2) Markov Random Field models, (3) statistics of high-pass filter responses, and (4) small texture patches (so-called *textons* [Julesz, 1981]). In this chapter, we perform a literature survey of these four approaches to the extraction of texture features from texture images. We discuss the four approaches separately in Section 5.1 to 5.4. Admittedly, there are more texture features. They include, e.g., blob features [Xu and Chen, 2006], and autocorrelation or autoregressive features [Kashyap and Khotanzed, 1986; Kang *et al.*, 2005], but because they are relatively less popular, we do not discuss them in this chapter. For other reviews of texture features, we refer to, e.g., [Tuceryan and Jain, 1998; Zang and Tan, 2002; Blunsden, 2004].

5.1 Graylevel co-occurrence features

Graylevel co-occurrence features represent a feature using statistics from the co-occurrence of a graylevel at a specific location with a graylevel at a location relative to that location [Haralick and Dinstein, 1973]. In order to compute graylevel co-occurrence features, first, a graylevel co-occurrence matrix (GLCM) \mathbf{C} is computed. The entries C_{ij} of the GLCM counts how often two pixels with relative angle θ and pairwise distance d (which are two free parameters) have the respective values i and j . Typically, GLCMs are computed for a large number of angle-distance pairs (θ, d) . After normalization, the matrices represent the joint probability of grayvalues i and j occurring in the image (for a given angle-distance pair).

From the normalized GLCMs, descriptive statistics are computed that form the feature representation of the texture. Although many different statistics have been proposed, the most important statistics are angular second moment, contrast, correlation, and entropy [Haralick and Dinstein, 1973; Strand and Taxt, 1994].

The main disadvantage of graylevel co-occurrence features is that they require the computation of a large number of graylevel co-occurrence matrices, which is computationally expensive, in particular, for large images. Empirical results indicate that, e.g., features based on filter-banks slightly outperform graylevel co-occurrence features [Randen, 1997].

5.2 Markov Random Fields

A Markov Random Field (MRF) is an undirected probabilistic graphical model. In order to make inference in Markov Random Fields tractable (up to a constant), it is necessary to restrict the MRFs to have a structure in which most of the nodes are conditionally independent, because the computational complexity of the model grows exponentially with the maximum clique size in the graph. Examples of such restricted structures are chains (as employed in Hidden Markov Models [Viterbi, 1967; Rabiner and Juang, 1986] and linear dynamical systems [Kalman, 1963; Ghahramani and Hinton, 1996b]) or bipartite graphs (the Restricted Boltzmann Machines [Ackley *et al.*, 1985] we employed in Chapter 4).

In texture modeling, Markov Random Fields are often assumed to have a grid structure, in which the intensity value of a pixel is governed by the intensity values of its directly neighboring pixels only [Cross, 1980]. The pixel values are usually quantized into k bins, and the nodes

are assumed to represent k -nomial distributions. An example of such an MRF model (using a 4-connected neighborhood) is shown schematically in Figure 5.3. In the figure, a circle represents a single pixel and the lines indicate the dependencies between the pixel values. Texture might be viewed upon as a Markov Random Field that generates pixel values according to the corresponding conditional probability distributions. As a result of the Hammersley-Clifford theorem [Clifford, 1990], the joint distribution $P(\mathbf{x})$ over the MRF model is given by

$$P(\mathbf{x}) = \frac{1}{Z} \prod_c V_c(\mathbf{x}_c), \quad (5.1)$$

in which c represents a maximal clique, \mathbf{x}_c represents the nodes in this clique, and $V_c(\mathbf{x}_c)$ represents the potential function defined over this clique, which usually takes the form of the exponential of the negative of an energy function $E(\mathbf{x}_c)$ (a so-called Boltzmann distribution). Notice that because the multinomial distribution belongs to the exponential family, the product $P(\mathbf{x})$ of the cliques $V_c(\mathbf{x}_c)$ is a Boltzmann distribution as well. The variable Z indicates the partition function of the model that makes sure that $P(\mathbf{x})$ is a valid probability distribution, and is given by

$$Z = \sum_{\mathbf{x}} \prod_c V_c(\mathbf{x}_c). \quad (5.2)$$

In general, it is not possible to compute the likelihood of a data vector under an MRF model due to the presence of the partition function: the evaluation of the partition function of an MRF model with n k -nomial nodes requires summing over k^n states. This prohibits the comparison of different MRF models based on the likelihood of the training data under the model, but it is possible to compare the likelihood of two data vectors under a single model based on their density (because under a single model, the partition function Z is constant).

Evaluation of this density in an MRF is relatively straightforward, since it requires evaluation of the conditional probability distribution $P(x_i | \forall x_j \in \mathcal{N}_i)$, where \mathcal{N}_i indicates the neighborhood set of x_i . The maximum likelihood estimate for this conditional probability distribution can readily be computed by normalizing the co-occurrence statistics of the cliques in the input images. Note that when a 4-connected neighborhood and k states per node are employed, the conditional probability distribution $P(x_i | \forall x_j \in \mathcal{N}_i)$ is parameterized by $4k^2$ parameters (because it involves 4 cliques of size 2). Because $P(x_i | \forall x_j \in \mathcal{N}_i)$ specifies the conditional probability of a single node given all other nodes, sampling from the joint distribution of the model (e.g., to perform texture synthesis) can be performed using, e.g., Gibbs sampling.

MRF-based texture models have two main advantages: (1) they are well suited for the modeling of artificial textures and (2) since they are probabilistic models, generating new texture images from a trained model can be performed by sampling from the joint distribution $P(x)$. The latter makes MRF models well suitable for texture synthesis and inpainting [Efros and Leung, 1999; Zalesny and van Gool, 2001]. An important disadvantage of MRF models is that they are often not well capable of modeling natural textures, because an MRF's decision on a pixel value depends solely on the pixel values of the surrounding pixels. This weakness can be addressed by enlarging the neighborhood of a pixel, however, this may lead to computational problems due to the rapid growth of the number of cliques in the MRF. These computational problems of MRF models are typically addressed by assuming that the Markov Random Field is homogeneous, i.e., that all potential functions V_c are identical to a single (shared) function V . Another disadvantage

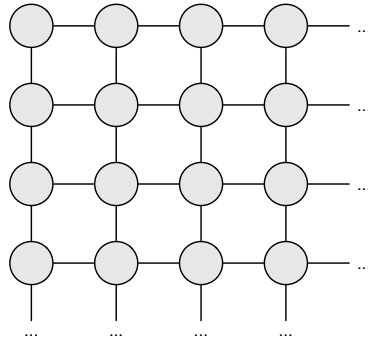


Figure 5.3 Graphical model of a Markov Random Field with a grid structure (using a 4-connected neighborhood). In line with the conventions for probabilistic graphical models, the nodes are shaded because they represent observed variables (i.e., pixels).

of MRF models is that it is unclear how the potential functions V_c should be defined. Recent work suggests to resolve this problem by learning the shared potential function V from the data, for instance, by defining it as the energy of a product of experts model¹ [Roth and Black, 2005].

5.3 Filter-based features

Filter-based texture features measure the presence of high spatial frequencies in a texture image at certain scales and orientations. The intuition behind filter-based features is that fine image details, which contain most information on the texture, are contained in high spatial frequencies. In contrast, low spatial frequencies carry information on the global structure of the surface. Hence, if we are interested in texture representations, we should mainly focus extracting statistics from the high spatial frequencies. Hence, filter-based approaches to texture modeling apply a filter bank that contains high-pass filters on the texture image. An additional advantage of applying a high-pass filter bank is that it gives rise to sparse image representations, because a filter response is only nonzero when the local structure of the image closely resembles the structure of the filter². Sparse image representations are advantageous because they lead to more efficient codes of the input images [Olshausen and Field, 1996; Hyvärinen *et al.*, 2008].

After the filter bank is applied on the texture images, statistics are extracted from the resulting activation images. Such an approach is motivated by the Julesz conjecture, which states that textures with similar higher-order (filter response) statistics are perceptually similar [Julesz, 1962]. As a result, higher-order statistics of texture images convolved with high-pass filters are informative texture features. The Julesz conjecture has been proven to be incorrect by Julesz himself by the construction of a set of texture images with identical second-order and third-order statistics that are perceptually different [Caelli and Julesz, 1978; Julesz *et al.*, 1978]. However, the Julesz conjecture still appears to hold well for real-world textures [Portilla and Simoncelli,

¹The product of experts model is a generalization of the Restricted Boltzmann Machine that is described in Appendix D [Hinton, 2002].

²In natural images, the Fourier amplitude is approximately inversely proportional to the frequency. Hence, high spatial frequencies are rare in natural images.

2000]. The human brain extracts statistics from a range of filtered images as well. For instance, the primal visual cortex (V1) consists of neurons that respond to edges with a certain scale and orientation [Jones and Palmer, 1987]; there exists evidence that the skewness of the resulting responses (a third-order statistic) plays an important role in the human brain [Motoyoshi *et al.*, 2007]. Computer vision systems generally use statistics such as image histograms, autocorrelations, and standard deviations [Portilla and Simoncelli, 2000].

Over the years, a large number of filter banks have been proposed in order to compute filter-based texture features. Below, we discuss five important filter banks that we use in our experiments in Chapter 6: (1) the Gabor filter bank, (2) the Maximum Response (MR) filter banks, (3) the Schmid filter bank, (4) steerable pyramids, and (5) the complex wavelet transform. The five types of filter banks are discussed separately in Section 5.3.1 to 5.3.5. For a more extensive overview of filtering approaches to texture classification, we refer to [Randen and Husoy, 1999].

5.3.1 Gabor filter bank

A Gabor filter bank is formed by a collection of Gabor filters with various orientations and scales. The Gabor filter is a high-pass filter that responds to edges with the same spatial frequency and orientation as the filter. Mathematically, the Gabor filter is the product of a Gaussian envelope and a complex sinusoid. It is given by the equation

$$G(x, y) = \frac{1}{2\pi} \underbrace{e^{-\frac{1}{2}(x'^2+y'^2)+i\kappa x'}}_{\text{complex sinusoid}} - \underbrace{e^{-\frac{\kappa^2}{\sigma^2}}}_{\text{envelope}}. \quad (5.3)$$

In this equation, the variable σ indicates the variance of the Gaussian, whereas the variable κ is given by

$$\kappa = \sqrt{2 \ln(2)} \frac{2^\phi + 1}{2^\phi - 1}, \quad (5.4)$$

in which ϕ is the bandwidth in octaves. The value of ϕ is typically $0.5 < \phi < 1.5$. The variables x' and y' define the orientation of the sinusoid, and thereby of the function response. They are defined by the equations

$$x' = x \cos \theta + y \sin \theta, \quad (5.5)$$

$$y' = -x \sin \theta + y \cos \theta. \quad (5.6)$$

In these equations, θ is the orientation of the filter in radians. The real and imaginary parts of a Gabor filter with $\theta = 0$ are shown in Figure 5.4.

A Gabor filter bank is typically formed by a set of Gabor filters with a number of orientations and scales. For instance, if eight orientations and three scales are employed, the Gabor filter bank consists of $8 \times 3 = 24$ filters. The use of a collection of filters with various scales and orientations allows for the measurement of the presence of edges at these scales and orientations, and thereby, it provides more information than simple edge-detecting filters such as the Laplace and Sobel filters. The human primal visual cortex V1 consists of neurons that respond to edges at a certain scale and orientation as well [Daugman, 1985; Jones and Palmer, 1987], and thereby, the use of Gabor filter banks may be motivated biologically.

Throughout the years, the Gabor filter bank has been applied in many variations. For instance, Bovik [1991] suggests the use of narrow-band Gabor filters of which the central frequencies are

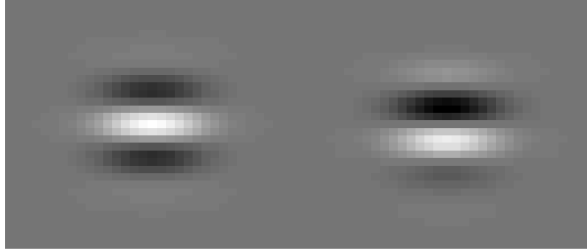


Figure 5.4 Real and imaginary parts of a Gabor filter.

tuned to the spectral peaks of the textures. In other words, the central frequency of the Gabor filters is set equal to the spatial frequency corresponding to the main spectral peaks in the image. Hereby, the image representation may be optimized. A Gabor filter design scheme that optimizes feature separation between two texture classes is proposed by Dunn and Higgins [1995]. In the scheme, the optimal central frequency of the filter is determined by the evaluation of a large number of frequencies, from which the frequency that minimizes the generalization error on a classification task is selected. A generalization of this scheme to texture classification problems with multiple classes is proposed by Weldon and Higgins [1996a,b]. Bianconi and Fernández [2007] investigate the effect of various parameter settings on the performance of Gabor filters in texture classification. The most important conclusion of this study is that the number of scales and orientations that is used in the filter bank is of limited influence on the performance on texture classification, whereas smoothing of x' and y' (by dividing them by fixed values $\gamma_{x'}$ and $\gamma_{y'}$ that are determined empirically) may improve the performance of Gabor filter banks significantly.

5.3.2 Maximum Response filter bank

One of the main problems with Gabor filter banks is the large number of filter responses it produces, which gives rise to a feature space of very high dimensionality. Maximum Response (MR) filter banks address this problem by identifying the maximum filter responses (produced by orientation-sensitive filters) over all orientations.

The MR filter bank consists of a collection of (1) edge filters and box filters at three scales and six orientations, (2) a Gaussian filter, and (3) a Laplacian of Gaussian filter [Varma and Zisserman, 2005]. After the filter bank is applied on the texture images, the responses of the edge filters and the box filters are combined by identifying the maximum response at each location over all orientations. This leads to the responses of the so-called MR8 filter bank, which constructs eight filter responses from the $2 \times 3 \times 6 + 1 + 1 = 38$ filter responses. Two of these filter responses are obtained from the rotation-invariant Gaussian and Laplacian of Gaussian filters. The remaining six filter responses are formed by the maximal responses of the anisotropic filters across all orientations. The advantage of such an approach over, e.g., a traditional Gabor filter bank is twofold: (1) the resulting filter coefficients are rotation-invariant and (2) the dimensionality of the filter responses is reduced. An important disadvantage of the MR filter banks is the large number of filters that is employed, leading to high computational costs. An alternative to

the MR8 filtering is the MR4 filter bank, in which a single scale filtering is performed, leading to just four filter responses.

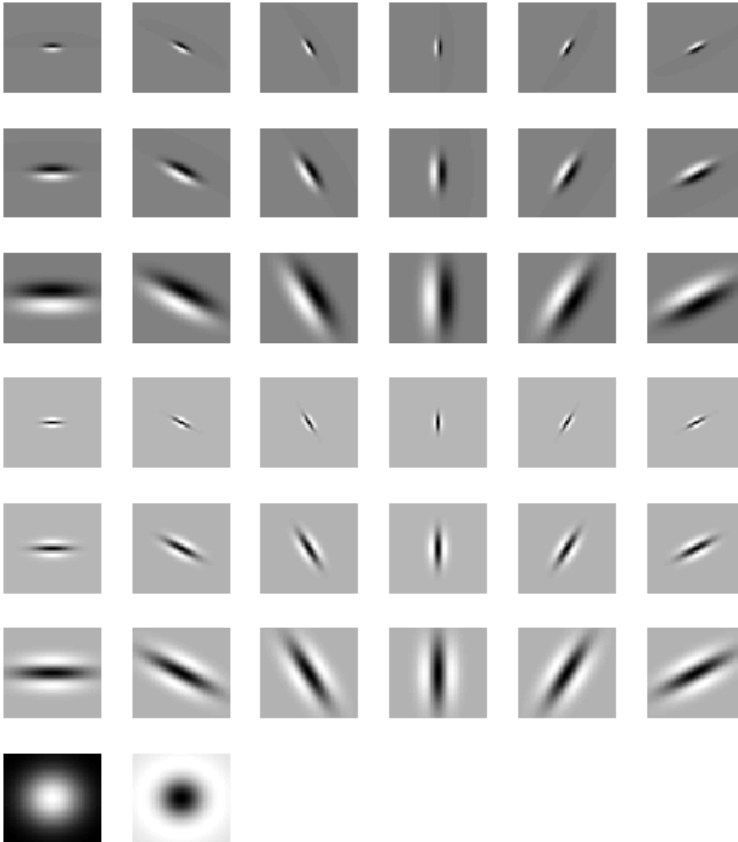


Figure 5.5 The basis of the Maximum Response (MR) filter banks.

5.3.3 Schmid filter bank

The Schmid filter bank consists of 13 circular filters that are rotationally invariant [Schmid, 2001]. All filters are of the form

$$F(r, \sigma, \tau) = F_0(\sigma, \tau) + \cos\left(\frac{\pi\tau r}{\sigma}\right) e^{-\frac{r^2}{2\sigma^2}}. \quad (5.7)$$

in which r controls the radius of the filter. The term $F_0(\sigma, \tau)$ is added to the filter in order to obtain a zero DC component. The 13 filters in the Schmid filter bank are constructed by setting the pair (σ, τ) to $(2, 1)$, $(4, 1)$, $(4, 2)$, $(6, 1)$, $(6, 2)$, $(6, 3)$, $(8, 1)$, $(8, 2)$, $(8, 3)$, $(10, 1)$, $(10, 2)$, $(10, 3)$ and $(10, 4)$. The resulting filters are shown in Figure 5.6. The main advantage of the Schmid filter bank is its rotation-invariance. Furthermore, the Schmid filter bank is computationally more efficient than, e.g., the Maximum Response filter banks, due to the limited number of filters that is applied on the image.

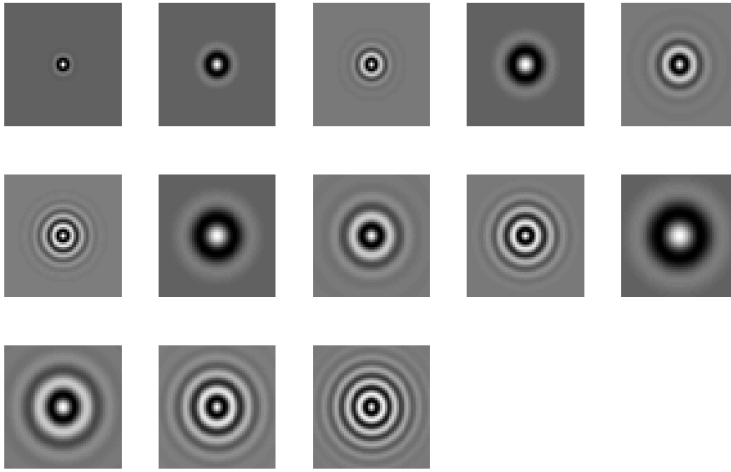


Figure 5.6 The Schmid filter bank.

5.3.4 Steerable pyramids

Steerable pyramids [Simoncelli and Freeman, 1995; Portilla and Simoncelli, 2000] are circular filters with multiple scales and orientations. The main advantage of steerable pyramids over other orientation-sensitive filters (such as Gabor filters and MR filters) is that steerable pyramids provide an image *decomposition*³, as a result of which the original image can be reconstructed from the filter responses. The main disadvantage of steerable pyramids are the computational and memory requirements, that result from the strong overcompleteness of the transform: the transform is $\frac{4k}{3}$ times complete when k orientations are employed. The steerable pyramid filters are illustrated in Figure 5.7.

One of the main advantages of steerable pyramids over other orientation-sensitive filters, is that they are *steerable* [Freeman and Adelson, 1991]. Steerability means that given the responses of a few filters with different orientations, it is possible to compute the response of a filter with any other orientation without actually convolving the input image with this filter. A filter is steerable if it can be expressed as the product of an orientation-invariant filter (such as the Schmid filters) and an angular weighting function. The minimum number of oriented filters required is equal to the number of nonzero coefficients of the Fourier expansion of the angular weighting function. The filter bank shown in Figure 5.7 reveals that steerable pyramids are indeed such a product.

5.3.5 Complex wavelet transform

The wavelet transform expands a signal into a collection of frequency components (similar to the Fourier transform). Unlike the Fourier transform, the wavelet transform does so by using a

³In wavelet literature, a transform that provides a decomposition is often referred to as a ‘tight frame’. This means that the transform obeys Parseval’s inequality: the L2-norm of the coefficients is equal to the L2-norm of the input image.

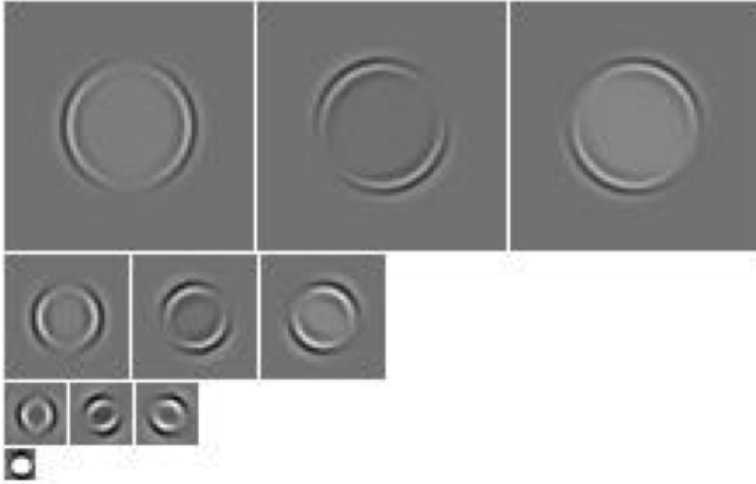


Figure 5.7 A steerable pyramid filter bank with three levels and three orientations ($k = 3$).

collection of localized basis functions. In this way, the wavelet transform resolves the Gibbs phenomenon [Wilbraham, 1848; Gibbs, 1898] from which the Fourier transform suffers. The Gibbs phenomenon occurs when the Fourier transform is applied on a discrete signal; it is illustrated in Figure 5.8. In practice, the wavelet transform is implemented as a dyadic filter tree in which a low-pass filter g and a high-pass filter h are employed. Both filters are applied on the signal, the low-pass filter response is downsampled, both filters are applied on the result, and this process is iterated. If both filters meet certain requirements (such as orthogonality of the filters), the responses of the high-pass filters provide the wavelet coefficients. An extensive introduction on wavelet theory can be found in [Chui, 1992; Daubechies, 1992].

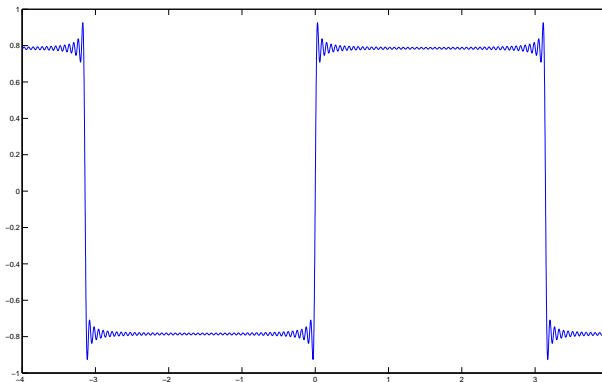


Figure 5.8 Illustration of the Gibbs phenomenon. The figure shows a reconstruction of a square wave using 50 sinusoids. The Gibbs phenomenon is visible around the discrete changes in the signal.

The complex wavelet transform (CWT) is capable of capturing more phase information than the traditional wavelet transform by the use of complex filters, and thereby, it provides approximate shift invariance to the wavelet transform [Kingsbury, 2001]. The CWT is implemented by means of a dual dyadic filter tree, of which a one-dimensional version is shown schematically in Figure 5.9. In the figure, square boxes indicate a filtering with either the high-pass filter h_i or the low-pass filter g_i , and $\downarrow 2$ indicates a downsampling of the signal by 2.

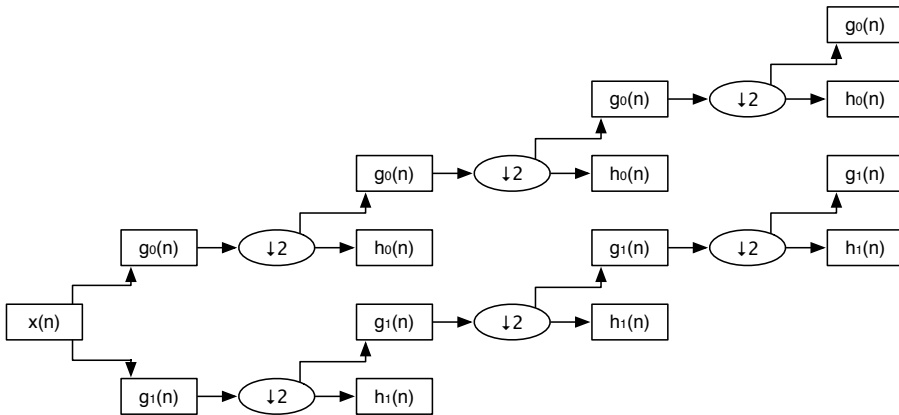


Figure 5.9 Complex wavelet transform filter tree.

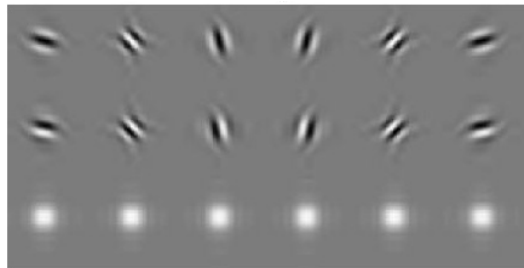


Figure 5.10 Wavelets corresponding to the complex wavelet transform. The upper row represents the real parts of the six wavelets, whereas the middle row represents the imaginary parts of the wavelets. The magnitude of the filters is depicted in the bottom row, revealing that the real and imaginary parts of the wavelets are 90° phase-shifted, and thus orthogonal. The wavelets were obtained using the filters proposed by Abdelnour and Selesnick [2001].

In addition to the restrictions on the filters in the traditional wavelet transform, the filters in the two branches of the filter tree should form Hilbert pairs. In other words, filter g_1 should be the Hilbert transform⁴ of filter g_0 , and filter h_1 should be the Hilbert transform of filter h_0 . If this requirement is met, the responses of the filters can be shown to complement each other, leading to a lower susceptibility of the wavelets transform to shifts in the signal (i.e., small translations

⁴The Hilbert transform [Hilbert, 1953] of a function $f(x)$ is the convolution of the function with $\frac{1}{\pi}x$, which leads to a shift of $+90^\circ$ to the phase of negative frequency components, and a phase shift of -90° of positive frequency components.

in the image). In the 2D case, the wavelets in the CWT show great resemblance to orientation-sensitive filters such as Gabor filters, as is illustrated in Figure 5.10. The wavelets in Figure 5.10 were obtained using the filters proposed by Abdelnour and Selesnick [2001]. Although the resulting wavelets look similar to Gabor filters, the CWT has three important advantages over other orientation-sensitive filters. First, CWT coefficients are less redundant than Gabor wavelet coefficients (the 2D CWT is only four times complete), leading to an image representation of lower dimensionality that can be computed more efficiently. Second, the support of the filters that are used in the CWT is generally small, allowing for a better estimation of the texton generation distribution and for an additional computational advantage. Third, similar to steerable pyramids, the CWT has an inverse transform, which implies that the original image can be reconstructed from the wavelet coefficients for, e.g., visualization purposes.

5.4 Texton-based features

In MRF-based texture features (see Section 5.2), texture is viewed upon as a probabilistic generator of pixel values. In contrast, in texton-based features, texture is viewed upon as a generator of small texture patches [Leung and Malik, 2001; Varma and Zisserman, 2002, 2003; Cula and Dana, 2004; Caputo *et al.*, 2005; Varma and Zisserman, 2005; Xie and Mirmehdi, 2007]. The representations of these texture patches (e.g., by means of filter bank responses or as a concatenation of pixel values) are called textons, and can be viewed upon as the fundamental building blocks of texture [Julesz, 1981]. Texture generates textons according to some underlying probability distribution (assuming neighboring textons are independent), which can be estimated by means of a texton frequency histogram. The texton frequency histogram measures the relative frequency of textons from a texton codebook in a texture image. A texton codebook is constructed by applying vector quantization on a set of randomly selected textons. An example of a texton codebook is shown in Figure 5.11. A texton frequency histogram is constructed from a texture image by scanning over the texture image and extracting small texture patches. The small texture patches are converted into the representation that is used in the codebook in order to obtain a collection of textons. For each texton in the collection, the texton is compared to the textons in the codebook in order to identify the most similar texton from the codebook, and the texton frequency histogram bin corresponding to this texton is incremented. After normalization, the texton frequency histogram forms a feature vector that models the texture.

The main advantages of texton-based features over MRF-based features and filter-based features are (1) its simplicity and (2) its computational efficiency. Furthermore, texton-based features have been shown to perform strongly on well-known texture datasets [Varma and Zisserman, 2007]. Successful applications of texton-based features are reported in, e.g., anomaly detection [Xie and Mirmehdi, 2007] and the classification of hematologic malignancies [Tuzel *et al.*, 2007]. The main disadvantage of texton-based features is their susceptibility to the presence of rotations, rescalings, or other affine transformations in the texture images. In Chapter 6, we discuss texton-based texture features in more detail, and we present approaches to address their susceptibility to, e.g., affine transformations.

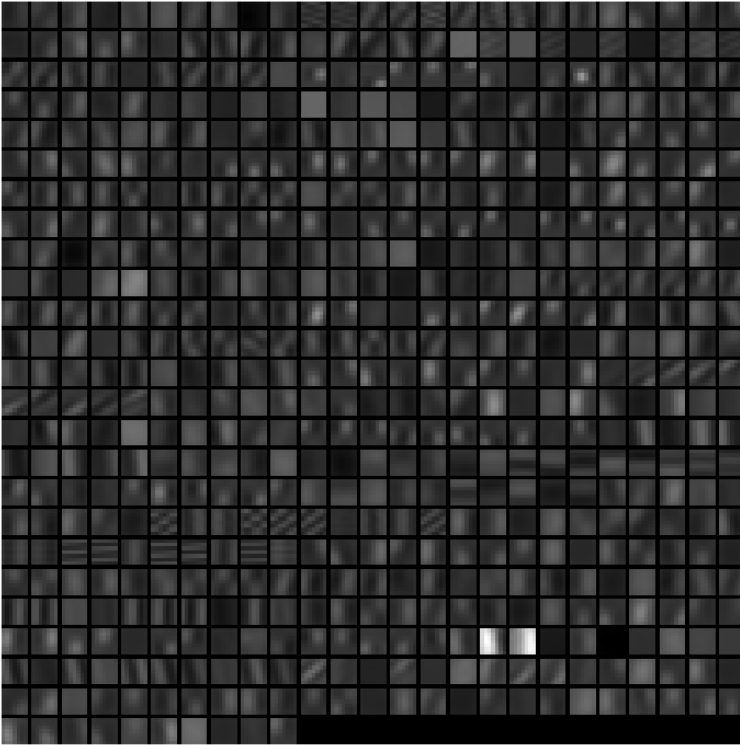


Figure 5.11 An example of a (pixel-based) texton codebook with pixel-based textons of size 7×7 pixels.

5.5 Chapter conclusions

In the chapter, we presented a literature review of state-of-the-art texture features, and we discussed the main limitations and weaknesses of the features. The texture features discussed can be subdivided into techniques based on graylevel co-occurrences, techniques based on Markov Random Fields, techniques based on statistics of filter responses, and techniques based on textons. In Chapter 6, we discuss texton-based texture models in more detail, and we present approaches to overcome the main weakness of texton-based texture models: their susceptibility to (local) affine transformations of the texture. In Chapter 7, we present applications of texton-based texture features to the analysis of Van Gogh paintings and to the automatic classification of seeds.

6 Texton-based texture features

- Contents** Over the last decade, filter-based features have dominated the field of texture modeling and analysis. The previous chapter briefly discussed, among others, texton-based texture features that do not employ filter-based images representations, but are claimed to perform on par with filter-based features. The current chapter investigates this claim in comparative experiments, the results of which challenge the supremacy of filter-based features. The success of the texton-based texture features opens up the way for the development of new types of invariant features. In order to answer research question RQ2, the chapter develops three new invariant texture features based on textons, two of which are invariant to rotations and one of which is invariant to local affine transformations. We investigate the invariance properties of the new texture features in a collection of texture classification experiments.
- Based on** L.J.P. van der Maaten and E.O. Postma. Texton-Based Texture Features with Local Affine Invariance. Submitted to *British Machine Vision Conference*.
- L.J.P. van der Maaten and E.O. Postma. Texton-Based Texture Classification. In Dastani, M. and de Jong, E., editors, *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence*, pages 213–220, 2007.
- Outline** In Section 6.1, we discuss the extraction of texton-based texture features in more detail. Section 6.2 discusses image-based and filter-based texton representations. We empirically compare image-based and filter-based texton representations in Section 6.3. We present two rotation-invariant texton representations and one affine-invariant texton representation in Section 6.4. In Section 6.5, we present our experiments with the newly developed texton representations. The results of these experiments are discussed in more detail in Section 6.6. Section 6.7 concludes the chapter.

In this chapter, we focus on texton-based texture features (which were already briefly discussed in Chapter 5). In particular, we investigate which texton representations are most appropriate for the construction of texton-based texture features that are invariant to, for instance, lighting changes, rotations, and changes in viewpoint. In most studies on texton-based texture features, the textons are represented as a collection of filter bank responses to obtain invariance to lighting changes [Leung and Malik, 2001; Varma and Zisserman, 2002; Cula and Dana, 2004; Caputo *et al.*, 2005]. However, the use of filter-based textons was recently challenged [Varma and Zisserman, 2003]. Instead, Varma and Zisserman [2003] advocate the use of image-based textons, which is controversial, because image-based representations are generally considered to be inappropriate for image modeling.

In this chapter, we perform texture classification experiments that compare image-based textons with filter-based textons, and establish the strong performance of image-based textons. An advantage of the use of image-based textons is that they allow for the development of new texton representations that are invariant to rotations or affine transformations, as a result of which they may address the variance problem of many state-of-the-art texture features. In particular, the development of affine-invariant texton representations is of high interest because such a texton representation leads to texture features that are invariant under *local* affine transformations, as a result of which they can be used to model the texture on non-planar surfaces that typically constitute real-world objects.

The remainder of this chapter consists of two main parts. First, in Section 6.1 to 6.3, we compare image-based textons with textons based on five different filter banks: (1) the Leung-Malik filter bank, (2) a Maximum Response filter bank, (3) the Schmid filter bank, (4) a steerable pyramid filter bank, and (5) a complex wavelet transform. The last four filter banks were discussed in Chapter 5, whereas the Leung-Malik filter bank is described by Leung and Malik [2001]. Second, in Section 6.4 and 6.5, we address the susceptibility to rotations and affine transformations of image-based texton representations by developing and investigating (1) two new rotation-invariant texton representations and (2) one texton representation that is invariant to the affine transformations (as a result of which the resulting texture feature is invariant to local affine transformations). The results of our experiments are discussed in more detail in Section 6.6. The main conclusion of the chapter is presented in Section 6.7, and reads that image-based texton representations are an appropriate alternative to filter-based texton representations that open up the way for the development of texton-based texture features that are not hampered by the variance problem.

6.1 Feature construction

As we already discussed in Section 5.4, texton-based texture features represent texture images by means of a texton frequency histogram. The construction of texton-based texture features is discussed in more detail below. It consists of two main stages: (1) the construction of a texton codebook and (2) the construction of a texton frequency histogram. We discuss the two stages separately in subsection 6.1.1 and 6.1.2.

6.1.1 Codebook construction

A texton codebook is a collection of textons that contains prototypes of the texture patches that occur in a set of texture images. The codebook of prototypical textons can be used to extract statistics from texture images in a similar way as, for instance, grapheme codebooks are used in the extraction of writer-specific features from handwriting [Schomaker *et al.*, 2007].

The construction of a texton codebook consists of three main steps. First, small texture patches are extracted from random positions in a collection of texture images that correspond to a specific texture class. Second, textons are obtained by converting the extracted texture patches into an appropriate image representation (such as a collection of filter bank responses or a concatenation of normalized pixel values). Third, vector quantization is performed on the resulting collection of textons using, e.g., k -means clustering [Bishop, 2006], Kohonen maps [Kohonen, 1989], or affinity propagation [Frey and Dueck, 2007] to obtain prototypical textons for the texture class at hand. The process is repeated for every texture class in the texture dataset, and the texton codebook is formed by gathering all prototypical textons of each of the texture classes. If the texture dataset is a representative subset of real-world textures, the texton codebook contains the most important textons that occur in real-world textures.

6.1.2 Texton frequency histogram

The rationale behind texton-based texture features is that texture is viewed upon as a probabilistic generator of textons. The underlying probability distribution of the generator can be estimated with the help of the texton codebook. Specifically, it can be estimated by means of a texton frequency histogram that measures the relative frequency of textons from the codebook in a texture image.

The texton frequency histogram of a texture image is computed by sliding a window over the texture image and extracting a texture patch at each location of the window. The small texture patches are converted to the same image representation that was used in the construction of the texton codebook (for instance, a collection of filter responses). The resulting textons are compared to the textons in the codebook in order to identify the most similar texton from the codebook, for instance, in terms of their pairwise Euclidean distance, and the texton frequency histogram bin corresponding to this texton is incremented. After normalization, the texton frequency histogram forms an estimator of the texton probability distribution that underlies the texture image at hand, as a result of which is a suitable feature representation of the texture image. In the experiments in this chapter, we compute texton frequency histograms using an overcomplete texture patch basis, i.e., there is overlap in the texture patches that are extracted from the texture images.

6.2 Texton representations

In the previous section, we discussed (i) the construction of texton codebooks and (ii) the computation of texton frequency histograms. Our discussion of the two topics is independent of the image representation that is employed in order to represent the textons (although the employed texton representation may be of relevance to the distance metric that is used to assess the simi-

larity of textons). In the remainder of the chapter, we focus on the texton representations that can be used in texton-based texture features.

As discussed above, most studies on texton-based texture features [Leung and Malik, 2001; Varma and Zisserman, 2002; Cula and Dana, 2004] represent textons by means of a collection of filter bank responses obtained from large filter banks (such as those discussed in Section 5.3). We discuss these filter-based texton representations in subsection 6.2.1. In [Varma and Zisserman, 2003], the supremacy of filter-based textons was questioned and the use of image-based textons was proposed. We discuss image-based textons in subsection 6.2.2.

6.2.1 Filter-based textons

Most studies on texton-based texture features employ a texton representation based on a collection of filter bank responses. For instance, Leung and Malik [2001] employ texton representations based on Leung-Malik filter bank responses, Varma and Zisserman [2002] use textons based on Maximum Response filter responses, and Cula and Dana [2004] employ a subset of the filters in the Leung-Malik filter bank to represent textons. Motivated by the lack of comparisons between these filter-based texton representations, we investigate texton representations that are based on five filter banks, four of which we discussed in Chapter 5: (1) the Leung-Malik filter bank, (2) the Maximum Response filter bank, (3) the Schmid filter bank, (4) the steerable pyramid filter bank, and (5) the complex wavelet transform. Our selection of these filter banks is motivated by their use in previous studies and by their different characteristics (which are described in Chapter 5). As a result, we believe that our selection of filter banks provides a good basis for a comparison of various filter-based texton representations.

The construction of filter-based textons is rather straightforward, and consists of two main steps. First, the input image is convolved with all filters that constitute the filter bank at hand. Second, the texton representation is constructed by gathering the responses of the filters at all scales and orientations that are measurements at the same spatial location in the texture image.

The only exception to this approach is formed by textons based on the complex wavelet transform, as the complex wavelet transform provides an expansion of the texture image at hand. In the construction of these textons, we extract a small image patch and we transform the extracted image patch to the complex wavelet domain by applying the complex wavelet transform on the image patch. Because a wavelet decomposition requires the length of a signal to be a power of two, the dimensions of the extracted image patch should be powers of two (such as 4×4 or 8×8 pixels).

6.2.2 Image-based textons

Image-based textons are small image patches extracted from a texture image of which the pixel values were normalized by (i) making them zero-mean and (ii) dividing them by their variance or standard deviation. The strong performance of image-based textons reported by Varma and Zisserman [2003] leads to questions about the necessity of applying filter banks for the analysis of texture. Varma and Zisserman [2003] suggest three main reasons for the relative strong performance of image-based textons [Varma and Zisserman, 2003].

First, the use of filter banks reduces the number of textons that can be extracted from a texture image. This reduction is a consequence of the large support of filter banks; the number of patches that can be extracted from a, say, 200×200 pixel texture image is significantly reduced when this image is convolved with a 50×50 filter. The presence of a reduced number of textons affects the quality of the texton frequency histogram estimations, leading to inferior generalization performances. Second, the large support of filter banks leads to small errors in the localization of edges. Imprecise edge localization may significantly change the geometry of the textons, leading to errors in the estimation of the texton frequency histogram. Third, the application of most filters leads to some blurring on the texture images, which is the result of the Gaussian envelope in these filters. The blurring might remove local details in the textons, that are of interest in the classification of the texture.

Although the three suggestions presented by Varma and Zisserman [2003] are interesting, the general claim that image-based textons outperform filter-based textons remains controversial [Mellor *et al.*, 2008]. In particular, there are two main reasons why image-based textons are not expected to lead to informative texture features: (1) image-based textons do not contain information on the presence of different orientations in the texture despite the fact that the measurement of edge orientations is known to be important in human vision [Jones and Palmer, 1987] and (2) image-based textons are relatively sensitive to the presence of noise in the image.

6.3 Experiments with filter-based textons

As a sequel to the controversial claim in [Varma and Zisserman, 2003] that image-based textons outperform filter-based textons, we investigate this claim below by performing experiments in which we use texton-based texture features in a texture classification task. The setup of the experiments is discussed in subsection 6.3.1. The results of the experiments are presented in subsection 6.3.2.

6.3.1 Experimental setup

We evaluated the quality of the texton-based texture features (constructed using filter-based or image-based textons) in texture classification experiments on the CURET texture dataset [Dana *et al.*, 1999]. The CURET dataset contains images of 61 different materials that were photographed under 205 different viewpoints. The differences in viewpoints led to a large variability in the visual appearance of the same material (as illustrated in Figure 5.2). From the 205 images of each texture class, we selected the 116 images that allow for the extraction of a texture image of 200×200 pixels. The selection of a part of the image is required as the images in the CURET dataset do not only reveal the texture of the photographed surfaces, but also their environment. Because we are interested in the texture of the surface, we extract image parts of size 200×200 pixels that only contain the texture of the surface. Examples of the selected image parts for all 61 texture classes of the CURET dataset are depicted in Figure 6.1. Because the color in the images provides too much information on the texture class, and our aim is to evaluate the quality of our texture descriptors, we converted all images in the dataset to grayscale images.

In our experiments, we constructed texton codebooks by performing k -means clustering on $116 \times 500 = 58,000$ textons from each texture class, that were obtained by random selection

from the training images. In our experiments, we used a value of $k = 10$, leading to texton codebooks consisting of $61 \times 10 = 610$ textons. All experiments except those with textons based on the CWT were performed with texture patches of size 3×3 to 8×8 pixels. The experiments with textons based on CWT features were performed with texture patches of 4×4 and 8×8 pixels, because wavelet transforms require a signal length that is a power of 2. In our experiments, the classification is performed by a 1-nearest neighbor classifier. The generalization performance of the classifiers is evaluated using 10-fold cross validation. Our experimental setup is roughly similar to the setup employed by Varma and Zisserman [2003].

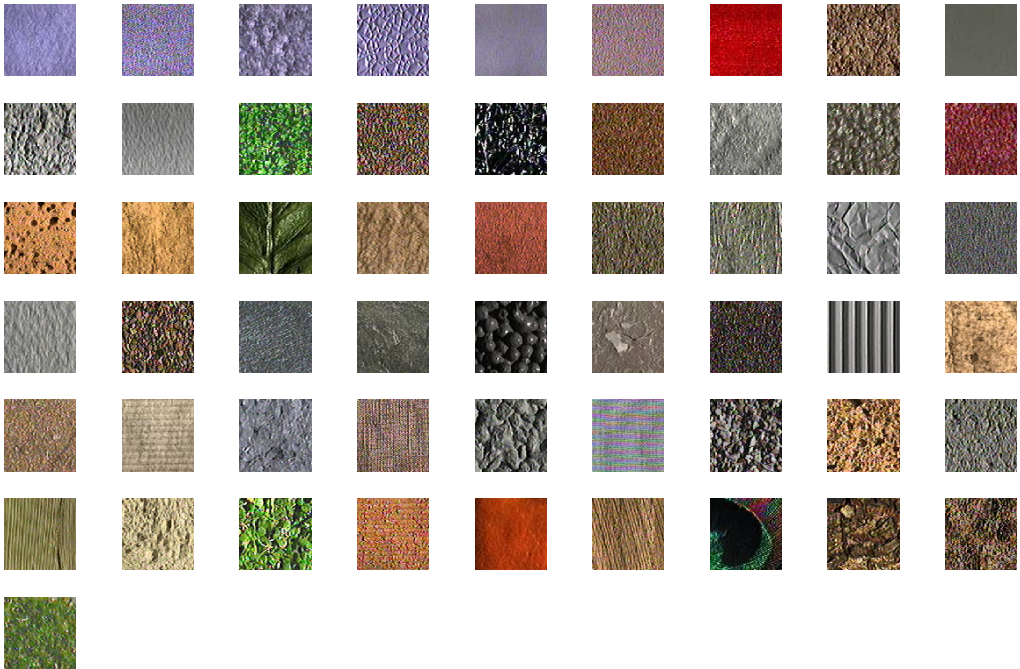


Figure 6.1 The 61 texture classes in the CURET texture dataset.

6.3.2 Results

In Table 6.1, we present the generalization errors of 1-nearest neighbor classifiers that were trained on texton frequency histograms using six texton representations. The table compares the generalization errors of classifiers trained using image-based textons with that of the five filter-based textons: (1) textons based on the Leung-Malik filter bank, (2) textons based on the Maximum Response-8 filter bank, (3) textons based on the Schmid filter bank, (4) textons based on steerable pyramids, and (5) textons based on the complex wavelet transform. The best generalization error for each texton size is typeset in boldface. From the results presented in the table, we make three main observations.

First, we observe the relatively strong performance of texton-based texture features that employ image-based texton representations. In three of the six experiments, the image-based texton representations outperform most filter-based textons (although the differences are often not statistically significant). In the other three experiments, the image-based textons perform only slightly worse than the best-performing textons. The best performance of all experiments was obtained using image-based textons of size 6×6 pixels. The results in Table 6.1 thus confirm the controversial claims about the performance of image-based textons by Varma and Zisserman [2003] that we discussed above.

Second, we observe that the best generalization performance of filter-based textons in our experiments was obtained using CWT-based textons of size 8×8 pixels. This result illustrates the potential advantage of the use of filters with small support (that are employed in the complex wavelet transform). Unlike other filter-based texton features, the CWT-based textons are not hampered by information loss at the borders or by the smoothing problem.

Third, we observe that filter-based textons that use steerable pyramids perform disappointing compared to image-based and other filter-based textons. This result may be due to the high sensitivity of steerable pyramids to their parameter settings.

Texton size	Image	Leung-Malik	MR8	Schmid	Steerable pyramids	CWT
3×3	0.0264 ± 0.0053	0.0259 ± 0.0061	0.0205 ± 0.0060	0.0229 ± 0.0052	0.1197 ± 0.0076	–
4×4	0.0206 ± 0.0064	0.0222 ± 0.0044	0.0212 ± 0.0033	0.0245 ± 0.0057	0.1146 ± 0.0133	0.0260 ± 0.0056
5×5	0.0204 ± 0.0062	0.0243 ± 0.0060	0.0201 ± 0.0057	0.0235 ± 0.0082	0.1112 ± 0.0096	–
6×6	0.0177 ± 0.0044	0.0246 ± 0.0054	0.0208 ± 0.0071	0.0248 ± 0.0045	0.1065 ± 0.0103	–
7×7	0.0195 ± 0.0057	0.0236 ± 0.0049	0.0223 ± 0.0058	0.0249 ± 0.0079	0.1195 ± 0.0115	–
8×8	0.0187 ± 0.0051	0.0257 ± 0.0054	0.0219 ± 0.0066	0.0283 ± 0.0078	0.1158 ± 0.0135	0.0179 ± 0.0038

Table 6.1 Generalization errors of 1-nearest neighbor classifiers trained on texton-based texture features on the CURET dataset.

6.4 Invariant texton representations

In the previous section, we obtained results that support the claim by Varma and Zisserman [2003] that image-based textons perform on par with filter-based textons, and even tend to outperform filter-based textons in texture classification experiments. The success of image-based texton representations provides new ways in which invariant texture features can be developed.

In this section, we develop three new image-based texton representations: two of them are invariant under rotations of the texture images and one of them is invariant under affine transformations. The rotation-invariant representations are based on spin images (subsection 6.4.1) and on polar Fourier features (subsection 6.4.2). The affine-invariant texton representation is based on an eigenanalysis of the second-order matrix (subsection 6.4.3).

6.4.1 Spin images

Spin images estimate the joint intensity-radius distribution of an image in a coarse histogram [Johnson and Hebert, 1999; Schmid *et al.*, 2004]. In the construction of a spin image, the distance of every pixel to the center of the image (i.e., the radius) is computed. The radiuses

and the corresponding pixel values are quantized and binned in a joint histogram. The construction of spin images is illustrated in Figure 6.2. The main advantage of the use of spin images is that they are invariant to changes in the orientation of the image.

In our texon-based texture features, we construct spin images with 8 intensity bins from the normalized textons that were extracted from the texture images. The number of radius bins is set to the width (or height) of the texture patches in pixels.

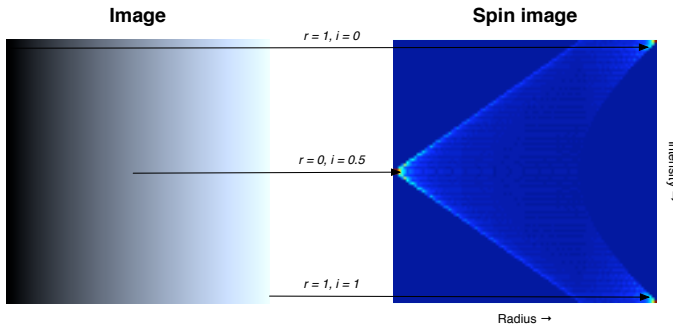


Figure 6.2 Illustration of the construction of a spin image.

6.4.2 Polar Fourier features

Polar Fourier features start by converting the texture patch to the polar space. In the polar space, one axis represents the distance to the center of the image, whereas the other axis represents the angle from the baseline (which is the horizontal line through the center of the image). As a result, a rotation of the original image leads to a circular shift in the ‘distance bands’ of the polar image. Subsequently, polar Fourier features employ the property that the magnitude of the Fourier transform of a histogram is invariant under circular shifts, because all phase information is in the sign of the Fourier coefficients [Szoplik and Arsenault, 1985]. Polar Fourier features make the polar image rotation-invariant by computing the magnitude of the Fourier transform of every distance band in the polar image.

The construction of polar Fourier features is illustrated in Figure 6.3. The middle image in Figure 6.3 shows that rotations in the original texture patch are reflected by circular shifts in the distance bands of the polar image. The right image in Figure 6.3 reveals that the magnitude of the Fourier transform of the distance bands is invariant under these circular shifts.

An important difference between spin images and polar Fourier features is that polar Fourier features implicitly assign more weight to the center of the texture patch, because pixels are ‘added’ by means of interpolation in the construction of the polar image representation. It is unclear whether assigning more weight to the center of the texture patch is advantageous or not, but the use of the (log)polar transform has been proven useful in other domains, such as image registration [Wolberg and Zokai, 2000].

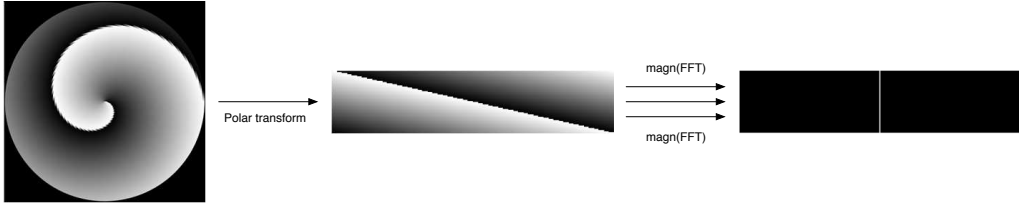


Figure 6.3 Illustration of the construction of polar Fourier features. The right image contains a single line, because the input image was designed to contain a single frequency.

6.4.3 Affine-invariant textons

The construction of a texton representation that is invariant to affine transformations can be performed using an eigenanalysis of the so-called second-order matrix. The construction of the affine-invariant texton representation consists of three main stages: (1) the computation of the second-order matrix, (2) the identification of an affine-covariant image region, and (3) the construction of the texton representation. We discuss the three stages separately below.

In order to construct the affine-invariant texton representation, in the first stage, we compute the second-order matrix $\mathbf{M}_{x,y}$ at location (x, y) . The second-order matrix is a 2×2 matrix that is computed from the horizontal and vertical image derivatives \mathbf{I}_X and \mathbf{I}_Y . It is defined as

$$\mathbf{M}_{x,y} = \begin{bmatrix} \sum (\mathbf{W}_{x,y} \cdot \mathbf{I}_X^2) & \sum (\mathbf{W}_{x,y} \cdot \mathbf{I}_X \cdot \mathbf{I}_Y) \\ \sum (\mathbf{W}_{x,y} \cdot \mathbf{I}_X \cdot \mathbf{I}_Y) & \sum (\mathbf{W}_{x,y} \cdot \mathbf{I}_Y^2) \end{bmatrix}, \quad (6.1)$$

where \cdot represents the element-wise or Hadamard product of two matrices, $\mathbf{W}_{x,y}$ is a matrix (with the same size as \mathbf{I}_X and \mathbf{I}_Y) that weights the image derivatives, and the sum is over all elements of the weighted product of the image derivatives. Typically, the weight matrix $\mathbf{W}_{x,y}$ is selected as to contain a localized Gaussian kernel (with a relatively small variance σ) that is centered onto the spatial location (x, y) .

In the second stage, we identify an affine-covariant image region by employing properties of the eigenvectors of the matrix $\mathbf{M}_{x,y}$. The second-order matrix $\mathbf{M}_{x,y}$ may be viewed upon as the (weighted) local covariance of the horizontal and vertical image derivatives, as a result of which the principal eigenvector of the second-order matrix represents the dominant direction of the image derivative; the corresponding eigenvalue determines how dominant this direction is. The eigenvectors and eigenvalues of the second-order matrix $\mathbf{M}_{x,y}$ can thus be visualized as an ellipse, as illustrated in Figure 6.4. In Figure 6.5, we show a grayscale image in which the second-order matrix at each spatial location is visualized by means of such an ellipse. The ellipse that is constructed from the eigenvectors and eigenvalues of the second-order matrix can be viewed upon as the ‘characteristic region’ of the image at that specific spatial location. This means that if an affine transformation is applied to the image, the ellipse will be transformed accordingly (for the proof, we refer to [Mikolajczyk and Schmid, 2004]). In other words, the ellipse covaries with affine transformations, and thus forms an *affine-covariant image region*. If the contents of the ellipsoid affine-covariant image region are normalized to the unit circle, the result is thus invariant up to rotations of the unit circle and errors due to interpolation effects. This is illustrated by an example in Figure 6.6.

In the third and final stage, we construct the affine-invariant texton representation (after having normalized the contents of the affine-covariant region to the unit circle) by transforming the unit circle to polar space and computing the magnitude of the Fourier coefficients of all distance bands of the polar representation (as we did in polar Fourier features). The magnitude of the resulting coefficients is invariant to affine transformations.

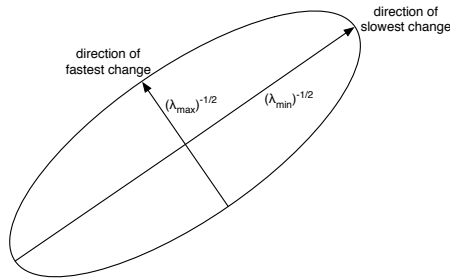


Figure 6.4 Illustration of the second-order ellipse.

6.5 Experiments with invariant textons

In the previous section, we developed three new texton representations that overcome the susceptibility of texton representations to rotations and/or affine transformations. This section investigates the performance of the new texton representations on four texture classification tasks: (1) a task in which there are no transformations applied on the textures, (2) a task in which the classifier has to deal with rotations of the textures, (3) a task in which the classifier has to deal with affine transformations of the texture, and (4) a task in which the input images contain natural local affine transformations of the depicted textures. The setup of our experiments is described in subsection 6.5.1. Subsection 6.5.2 presents the results of our experiments.

6.5.1 Experimental setup

In order to evaluate the performance of the invariant textons discussed above, we performed experiments on the CURET dataset. The texture images were preprocessed as described in 6.3.1. Because the main aim of our experiments is to investigate the invariance properties of the new texture features, we performed experiments in which we apply rotations and affine transformations on the test images (but not on the training images). In particular, we performed three different experiments on the CURET dataset: (1) an experiment in which there are no artificial transformations, (2) an experiment in which the test images are rotated by 90° , and (3) an experiment in which the test images are transformed using an affine transformation. In particular, we used an affine transformation that rotates the test images by 90° and scales up the test images by a factor of 2 (using bicubic interpolation). We opt for rotations of the test images of 90° , because such

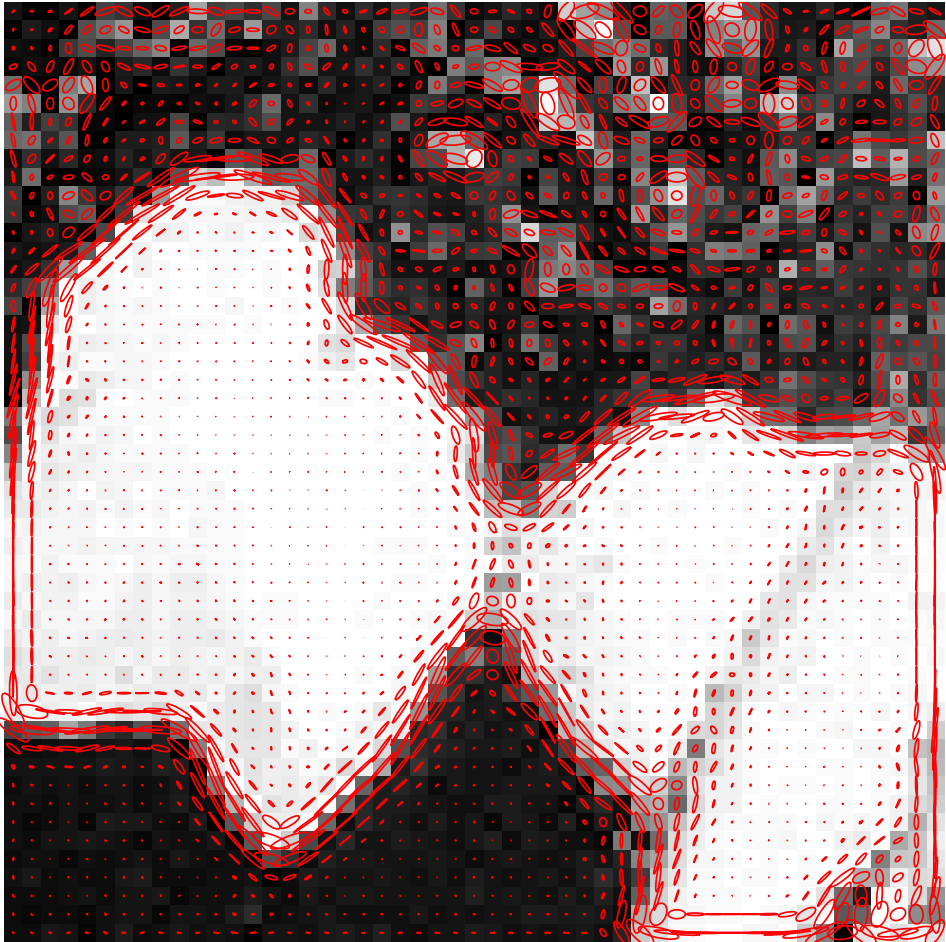


Figure 6.5 Second-order ellipses drawn onto a grayscale image.

a rotation does not degrade the quality of the test images as a result of interpolation and border artefacts. The remainder of the experimental setup of our experiments on the CURET dataset is identical to that described in 6.3.1.

Next to the experiments on the CURET dataset, we performed experiments on the UIUCTex dataset [Lazebnik *et al.*, 2005]. The UIUCTex dataset contains 40 images for each of 25 texture classes, giving rise to a dataset with 1,000 images. The images in the dataset have size 640×480 pixels. The 25 texture classes are shown in Figure 6.7. The main difference between the UIUCTex dataset and the CURET dataset is that the textures that are depicted in the UIUCTex dataset are subject to local affine transformations. These local affine transformations are the result of viewpoint variations and the bending of some of the textured surfaces. As our texton-based texture features are theoretically invariant to local affine transformations, they are likely to outperform other texton-based texture features on the UIUCTex dataset.



(a) Original image with affine-covariant region.



(b) Affine transformed image with affine-covariant region.



(c) Polar representation of affine-covariant region of original image.



(d) Polar representation of affine-covariant region of affine transformed image.

Figure 6.6 Illustration of affine-covariant image regions. Notice that the affine transformation of the image is reflected in a vertical shift in the polar images.

In the experiments on the UIUCTex dataset, we constructed texton codebooks by performing k -means clustering on $40 \times 500 = 20,000$ randomly selected textons for each texture class. As in the experiments on the CURET dataset, we used a setting of $k = 10$, as a result of which the final codebooks contain $25 \times 10 = 250$ textons. The experiments with the image-based textons, spin image-based textons, and polar Fourier textons were performed using textons of size 3×3 to 8×8 pixels. In the experiments with the affine-invariant textons, we used settings of the scale s of the second-order ellipses between 1 and 7. The quality of the resulting texture features is evaluated by measuring the generalization error of 1-nearest neighbor classifiers using 10-fold cross validation.

6.5.2 Results

In Table 6.2, we present the generalization errors of 1-nearest neighbor classifiers on the CURET dataset. The 1-nearest neighbor classifiers were trained on texton frequency histograms using image-based textons and the three invariant texton representations: (1) textons based on spin images, (2) textons based on polar Fourier features, and (3) affine-invariant textons. In the experiments, the texture images were not rotated, rescaled, or affinely transformed. In Table 6.3,

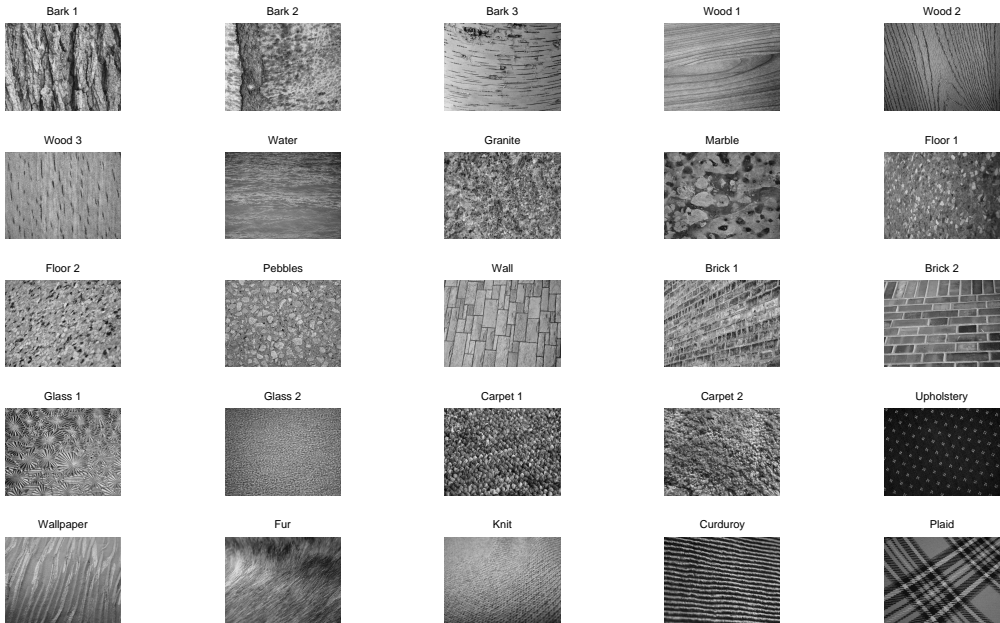


Figure 6.7 The 25 texture classes in the UIUCTex texture dataset.

we present the results of similar experiments in which 1-nearest neighbor classifiers were trained on normal texture images, but tested on texture images that were rotated 90° in a clockwise direction. Table 6.4 presents the results of similar experiments in which the 1-nearest neighbor classifiers were trained on normal texture images, but tested on texture images that were scaled up by a factor of 2 (using bicubic interpolation), and rotated by 90 degrees in a clockwise direction. In the tables, we do not typeset any generalization errors in boldface, as the setting of the patch size cannot readily be compared to the setting of the scale parameter of affine-invariant textons. From the results presented in Table 6.2, 6.3, and 6.4, we make the following three main observations.

First, we observe that the use of invariant textons slightly degrades the performance of the classifiers (compared to image-based textons) in experiments in which no transformations are applied on the test images. For spin images, the degradation of the performance of the classifiers is approximately 3%. For the affine-invariant textons, the degradation of the generalization performance lies between 6 and 10%. Our polar Fourier features are least hampered by the loss of information that is the result of the additional invariance: the generalization performance of classifiers trained using polar Fourier-based textons is only degraded by 1 to 2% (compared to image-based textons).

Second, from the results in Table 6.3 and 6.4, we observe that the presence of rotations severely degrades the performance of image-based textons. Specifically, the presence of rotations degrades the performance of image-based textons by 50 to 60%. In contrast, textons based on spin images and polar Fourier features are not hampered at all by the presence of rotations in the test

images: the performance of these textons in Table 6.2 and 6.3 is approximately equal. The same observation holds for affine-invariant texton representations.

Third, we observe that the performance of texture features that employ image-based textons, textons based on spin images, and textons based on polar Fourier features is severely deteriorated due to the presence of affine transformations in the test images. The generalization error obtained using these three texton representations is approximately 90% in all experiments. In contrast, the performance of classifiers trained on texture features that employ affine-invariant texton representations is more robust under the presence of affine transformations in the test images. The best generalization error we obtained on the CURET dataset (with affine transformations applied to the test images) was 26.42%. We do note that the results show that the quality of affine-invariant texton representations depends strongly on the scale parameter that is used to determine the size of the affine-covariant regions.

<i>Texton size</i>	<i>Image</i>	<i>Spin image</i>	<i>Polar Fourier</i>	<i>Scale</i>	<i>Affine-invariant</i>
3×3	0.0264 ± 0.0053	0.0649 ± 0.0100	–	1	0.2320 ± 0.0152
4×4	0.0206 ± 0.0064	0.0546 ± 0.0073	0.0466 ± 0.0072	2	0.0921 ± 0.0124
5×5	0.0204 ± 0.0062	0.0509 ± 0.0121	–	3	0.0873 ± 0.0071
6×6	0.0177 ± 0.0044	0.0530 ± 0.0123	–	4	0.0810 ± 0.0098
7×7	0.0195 ± 0.0057	0.0516 ± 0.0102	–	5	0.0810 ± 0.0153
8×8	0.0187 ± 0.0051	0.0530 ± 0.0085	0.0243 ± 0.0070	6	0.0767 ± 0.0068

Table 6.2 Generalization errors of 1-nearest neighbor classifiers trained on invariant texton-based features on the CURET dataset (no transformations).

In order to investigate the performance of the texton-based texture features under the presence of local affine transformations, we also performed experiments on the UIUCTex dataset. Table 6.5 presents the generalization errors of 1-nearest neighbor classifiers on this dataset. Here too, we performed experiments with image-based textons and the three invariant texton representations. From the results presented in Table 6.5, we make two main observations.

First, we observe that affine-invariant textons perform strongly compared to the other texton representations in the experiments on the UIUCTex dataset. However, the best generalization performance of 9.30% was obtained using texton representations based on spin images (of size

<i>Texton size</i>	<i>Image</i>	<i>Spin image</i>	<i>Polar Fourier</i>	<i>Scale</i>	<i>Affine-invariant</i>
3×3	0.5369 ± 0.0143	0.0622 ± 0.0099	–	1	0.2734 ± 0.0150
4×4	0.5597 ± 0.0119	0.0541 ± 0.0082	0.0630 ± 0.0121	2	0.1000 ± 0.0173
5×5	0.6104 ± 0.0153	0.0540 ± 0.0061	–	3	0.0902 ± 0.0081
6×6	0.6512 ± 0.0050	0.0543 ± 0.0093	–	4	0.0826 ± 0.0090
7×7	0.6778 ± 0.0144	0.0523 ± 0.0062	–	5	0.0823 ± 0.0074
8×8	0.6971 ± 0.0179	0.0520 ± 0.0107	0.0253 ± 0.0062	6	0.0837 ± 0.0142

Table 6.3 Generalization errors of 1-nearest neighbor classifiers trained on invariant texton-based features on the CURET dataset (rotations).

<i>Texton size</i>	<i>Image</i>	<i>Spin image</i>	<i>Polar Fourier</i>	<i>Scale</i>	<i>Affine-invariant</i>
3×3	0.9050 ± 0.0093	0.9741 ± 0.0061	–	1	0.7410 ± 0.0124
4×4	0.8979 ± 0.0094	0.9632 ± 0.0067	0.9686 ± 0.0064	2	0.4911 ± 0.0157
5×5	0.8963 ± 0.0144	0.9496 ± 0.0108	–	3	0.3827 ± 0.0183
6×6	0.8911 ± 0.0079	0.9464 ± 0.0053	–	4	0.3093 ± 0.0143
7×7	0.8973 ± 0.0154	0.9315 ± 0.0105	–	5	0.2741 ± 0.0103
8×8	0.8909 ± 0.0057	0.9385 ± 0.0106	0.9635 ± 0.0075	6	0.2642 ± 0.0130

Table 6.4 Generalization errors of 1-nearest neighbor classifiers trained on invariant texton-based features on the CURET dataset (affine transformations).

8×8 pixels). Presumably, the affine-invariant textons do not clearly outperform spin images due to two main problems: (1) the affine-invariant texton representation may be hampered by the presence of areas in the texture images that are somewhat out-of-focus and (2) the affine-invariant textons appear to be quite sensitive to the setting of the scale parameter. The first problem is the result of the image gradients being relatively small in out-of-focus image regions, as a result of which the affine-covariant ellipses are somewhat smaller than they would have been if the texture-images were in-focus. This may give rise to larger differences in texton frequency histograms of texture images that correspond to the same class. The second problem is the result of the influence of the scale parameter on the identified affine-covariant regions. If the scale parameter is too small or too large¹, the identified regions are not perfectly affine-covariant. As a result, affine-invariant textons with a scale of, say, 2.5 may obtain a lower generalization error on the UIUCTex dataset than the generalization errors reported for affine-invariant textons in Table 6.5.

Second, we observe that the performance of spin image-based textons on the UIUCTex dataset is strong compared to, e.g., the performance of texton representations based on polar Fourier features. This result is most likely due to that spin images are hampered less by the presence of out-of-focus regions. Spin images represent intensity value measurements in a coarse histogram, as a result of which they are less susceptible to errors that occur in image regions that are somewhat out-of-focus. The spin images constructed from out-of-focus regions are blurred versions of their counterparts constructed from in-focus image regions. The blurring of spin images does not have a strong negative influence on the matching with the texton codebook (blurring histograms may even slightly increase the performance of codebook approaches [Faichney and Gonzalez, 2002]).

In addition to the classification experiments on the UIUCTex dataset, we performed an experiment in which we visualized the UIUCTex dataset in a two-dimensional map by performing t-SNE on the affine-invariant texture features extracted from the texture images. The result of this experiment is shown in Figure 6.8. The insets show magnifications of parts of the visualization.

From the visualization in Figure 6.8, we can make two observations. First, we observe that the affine-invariant texture features separate the texture classes in the dataset quite well. For instance, the *plaid* texture images are widely separated from the other texture classes. Second, we observe that the texture features are clearly invariant under affine transformations. For instance, the *brick*

¹We also note that the optimal value for the scale parameter may vary between texture classes or imaging conditions.



Figure 6.8 Map of the UIUCTex dataset constructed by performing t-SNE on affine-invariant texture features. The insets show magnifications of parts of the map.

<i>Texon size</i>	<i>Image</i>	<i>Spin image</i>	<i>Polar Fourier</i>	<i>Scale</i>	<i>Affine-invariant</i>
3×3	0.2020 ± 0.0343	0.2710 ± 0.0318	–	1	0.1480 ± 0.0388
4×4	0.1830 ± 0.0495	0.1820 ± 0.0305	0.3410 ± 0.0570	2	0.1100 ± 0.0306
5×5	0.1680 ± 0.0266	0.1610 ± 0.0407	–	3	0.1210 ± 0.0484
6×6	0.1740 ± 0.0306	0.1390 ± 0.0256	–	4	0.1230 ± 0.0306
7×7	0.1810 ± 0.0357	0.1260 ± 0.0395	–	5	0.1270 ± 0.0411
8×8	0.1830 ± 0.0275	0.0930 ± 0.0267	0.2950 ± 0.0490	6	0.1370 ± 0.0313

Table 6.5 Generalization errors of 1-nearest neighbor classifiers trained on invariant texton-based features on the UIUCTex dataset.

texture images are modeled close together despite the large variation in the viewpoints under which the bricks were photographed.

6.6 Discussion

In the previous sections, we presented the results of experiments with one image-based, five filter-based and three invariant texton representations on two texture datasets. In this section, we discuss four observations made from the results of our experiments.

First, the performance of textons based on the filter-based responses supports the claim by Varma and Zisserman [2003] that, in contrast to popular belief, it is not required to measure filter responses in order to extract informative texture features. We surmise that the main disadvantage of the filter-based textons that are employed by Varma and Zisserman [2003] is the reduction of the number of textons (due to the large support of the filters). Since textons based on the complex wavelet transform do not suffer from this weakness, they outperform the other filter-based textons. This result is due to the small support of the filters we applied in the complex wavelet transform, and due to the low redundancy of the complex wavelet transform. Despite the advantages of the complex wavelet transform, CWT-based textons do not significantly outperform image-based textons, which suggests that imprecise edge localization is a problem in all filter-based textons.

Second, next to the arguments presented by Varma and Zisserman [2003], we believe that there is an important additional reason for the strong performance of image-based textons. The main aim of the application of filters in texture analysis is to extract information on the high spatial frequencies in the image, and to discard information on the low spatial frequencies. Although this aim of filters is relevant when, for instance, the extracted texture features are formed by simple statistics of the filter responses such as an intensity histogram, it is not so relevant when textons are employed. When textons are employed, most low spatial frequency information is already discarded anyway, thanks to the small size of the employed image patches. The selection of high spatial frequencies using filters is thus superfluous in texton-based texture features.

Third, the results of our experiments show that current texton-based texture features are very sensitive to the presence of rotations or (local) affine transformations in the texture images. The results of our experiments show that invariant textons may degrade the accuracy of the texture features somewhat. This degradation is the result of the loss of information that necessarily oc-

curs when invariances are built into image representations. For instance, our rotation-invariant texture features model each ‘distance band’ separately in a rotation-invariant model, as a result of which the information on the alignment of the distance bands is lost. However, our experiments showed that the degradation of the performance as a result of the information loss is limited. For instance, polar Fourier features almost perform comparable to their image-based counterparts that are sensitive to changes in the orientation of the texture in experiments on the CuRET dataset.

Fourth, we observe that the invariance under local affine transformations of our texture features based on affine-invariant textons leads to strong generalization performances on the UIUCTex dataset. However, the results also suggest that affine-invariant textons are sensitive to (1) the setting of the scale of the affine-covariant regions and (2) the presence of out-of-focus regions in the texture images. We discuss potential solutions to both problems below.

The first problem may be resolved by (1) performing the feature extraction using multiple scales and comparing the features of two images across all combinations of scales, or (2) by employing automatic scale selection techniques such as those presented in [Lindeberg, 1998; Kadir and Brady, 2001]. Typically, automatic scale selection techniques analyze multi-scale filter responses to obtain a series of local scale estimates. For instance, Lindeberg [1998] finds local maxima in the scale space pyramid of Laplacian of Gaussian (LoG) responses, which results in a series of local scale estimates of which the median may provide an estimate for the salient scale of the texture image. An alternative approach selects the scale for which the entropy of a collection of local Fourier and wavelet descriptors as the salient scale of the image [Kadir and Brady, 2001].

The second problem may be addressed by one of the following three approaches. The problem may be addressed by (1) only extracting patches at keypoint locations in the texture images, as is done in, e.g., [Lazebnik *et al.*, 2005]. A possible drawback of this approach is that homogeneous textures typically contain a small number of keypoint locations, as a result of which the resulting texton frequency histograms may be subject to large errors. The problem may also be addressed by (2) identifying out-of-focus regions in the texture images by locally comparing filter responses (see, e.g., [Shoa *et al.*, 2004]). The identified out-of-focus regions can then be ignored in the construction of the texton frequency histograms using affine-invariant textons (or the weight of the corresponding textons can be set accordingly). Another alternative to address the problem may be to (3) represent the normalized affine-covariant image regions using spin images instead of polar Fourier features. The relatively strong performance of texton representations based on spin images on the UIUCTex dataset suggests the spin image representation may outperform the polar Fourier representation.

6.7 Chapter conclusions

Texton-based texture features form an interesting alternative to traditional texture classification approaches such as Markov Random Fields or filter bank models. The results in this chapter suggest the use of (invariant) image-based textons over filter-based textons. We may conclude that the extraction of texture features from images can be performed well without the computa-

tionally expensive application of large filter banks on the texture images. Moreover, the use of image-based textons facilitates the development of invariant texton representations that give rise to the texture features that are invariant under local transformations.

We developed three invariant texton representations, two of which are invariant under rotations, and one of which is invariant under affine transformations. The results of our experiments on the CURET dataset revealed the invariance properties of our texture features, which come at a relatively small degradation in performance. We showed that it is possible to construct texture features that are invariant to local affine transformations by employing affine-invariant textons. The results of our experiments revealed that affine-invariant textons outperform all other texton representations on the UIUCTex dataset.

Future work focuses on (1) developing an approach that resolves the susceptibility of affine-invariant textons to the setting of the scale parameter and (2) developing an approach that is more robust to the presence of out-of-focus image regions. Moreover, future work may focus on the development of invariant texton-based texture features that employ the color information in the texture images, because color (and its interplay with texture) is an important feature of many textures. Color may be used in texton-based texture features, for instance, using one of the schemes proposed by Burghouts and Geusebroek [2008].

7 Applications to the cultural heritage

Contents

The previous chapters have presented features that address the redundancy and variance problems of image-space representations in an attempt to improve the performance of state-of-the-art computer vision systems. In order to investigate whether our features are effective in real-world computer vision systems, we investigate the performance of the newly developed features in the challenging cultural heritage domain. Specifically, we employ the developed features in the analysis of paintings that are (allegedly) painted by Van Gogh and some of his contemporaries, and we use the new features in the analysis of digital photographs of seeds.

Outline

In Section 7.1, we present the application of textron-based texture features and t-SNE in the analysis of paintings. The application of these features in seed analysis is presented in Section 7.2.

In the previous chapters, we investigated dimensionality reduction features and texture features, and we developed new features of both types. In this chapter, we apply the developed features on two real-world visual tasks from the cultural heritage domain. Specifically, we investigate the applicability of the developed features in (1) painting analysis and (2) seed analysis. We discuss the two applications separately in Section 7.1 and 7.2. The conclusions of this chapter are presented in Section 7.4. In Appendix F, we present an additional application of computer vision to the cultural heritage domain. There, we apply edge-based statistical features to the classification of ancient coins.

7.1 Painting analysis

The analysis of paintings is an important task in the cultural heritage that aims to give information about the attribution and the creation process of a painting. In particular, the attribution of a painting to an artist (i.e., artist identification) is of high importance to the monetary value of the painting. Currently, artist identification is performed by art experts who have considerable experience with the works by one or more painters. Although experts have a variety of techniques at their disposal – such as canvas weave count, dendrochronological analysis of the wood of the frame, chemical analysis of the pigments, and x-radiography of the painting or support – visual assessment of the painting is still one of their most important tools. An important cue in identifying the artist of a painting is the “handwriting” of the painter: the brushstrokes and brushstroke configurations that reveal the painter’s style. Despite the variations in form and appearance of a painter’s brushstrokes, the artist’s handwriting can be recognized by skilled art experts, although it is hard to explicate to laymen what the characteristic elements of a painter’s handwriting are. Intelligent image analysis and machine learning techniques that are sensitive to the brushstroke texture may support the art expert in detecting and visualizing painter-specific brushstrokes, and provide objective evidence for the attribution of a painting to an artist.

Previous work on artist identification and painting analysis using digital analysis techniques focused on the assessment of color use in (Van Gogh) paintings using filter-based approaches [Berezhnoy *et al.*, 2007], and on capturing statistical information from segmented or outlined brushstrokes [Sablattig *et al.*, 1998]. In this section, we build an effective representation of the brushwork of paintings using the texton-based texture features developed in Chapter 6. Moreover, we visualize the representations by means of t-SNE (which was developed in Chapter 3). We motivate our approach to painting analysis below.

Given that the segmentation of individual brushstrokes from a painting is unfeasible [Johnson *et al.*, 2008], from an image analysis perspective, brushstroke analysis corresponds to the analysis of the texture of the painting. The wildly overlapping brushstrokes form a textural cue of the painter’s handwriting. As we already discussed in Chapter 5, texture analysis is typically performed by applying a bank of filters that respond to intensity transitions in the input image. Various studies also use texture features based on filter responses in painting analysis [Sablattig *et al.*, 1998; Johnson *et al.*, 2008]. As we explained in Chapter 6, one of the main arguments against the use of filter-based approaches is that they employ filters with a Gaussian envelope, as a result of which the filters *smooth* the image before they measure the presence of (oriented) high spatial frequencies. The smoothing is necessary to remove noise that masks image gradients.

However, the smoothing may distort or remove pivotal information of relevance to the texture analysis task. In the case of the brushstroke texture in paintings, such details may correspond to individual hairs in the brush used by the painter that contain valuable cues for artist identification. For instance, the extent to which such hairs are visible in the painting may provide information on the amount of pressure the artist exerted on the brush in the various stages of a brushstroke. The image-based texton features described in 6.2.2 are not hampered by the smoothing problem, as a result of which they are well suitable for the analysis of brushstroke texture.

Our approach to painting analysis consists of two stages. We analyze the brushstroke texture of the paintings by means of image-based texton features (stage 1), as motivated above. The texture analysis results in high-dimensional feature vectors, which we reduce to two dimensions using t-SNE in order to facilitate visualization in a scatter plot (stage 2). Even though alleged attributions by art experts are available, we opt for the visualization of the results of the brushstroke analysis instead of for an automatic artist classification experiment, because the number of negative examples (i.e., non-Van Gogh paintings) in our dataset is too small to facilitate such a classification experiment.

Below, we present our experiments with the approach described above on a dataset of digital reproductions of paintings by Van Gogh and his contemporaries. The setup of these experiments is described in 7.1.1. The results of the experiments are presented in 7.1.2, and are discussed in more detail in 7.1.3.

7.1.1 Experimental setup

We apply the approach that was outlined above on a recently released dataset of 117 high-resolution digital reproductions of Van Gogh paintings. The dataset contains high-resolution digital 48-bit color reproductions of 117 paintings attributed to Van Gogh and related painters, which we transformed to 8-bit grayscale images for our experiments. The reproductions were created using ektachromes made available by the Van Gogh Museum and the Kröller-Müller Museum (both in the Netherlands). The paintings were normalized in such a way that a square inch of the painting is represented by 196.3×196.3 pixels. Of the 117 paintings, 13 are known not to be painted by Van Gogh and 6 are of disputed authorship. The remaining 98 paintings are generally accepted as authentic Van Gogh paintings. Each painting is labeled with its authenticity (Van Gogh, non Van Gogh, or disputed), and all authentic Van Gogh paintings are labeled by their creation date (ranging from 1884 to 1890) and creation place.

In order to evaluate our approach to painting analysis, we extracted texton histograms for image-based textons of six different sizes: 25×25 , 35×35 , 45×45 , 55×55 , 65×65 , and 75×75 pixels. The six texton codebooks employed in the experiment were constructed using affinity propagation, and contained approximately 500 textons each. Figure 7.1 shows one of the constructed texton codebooks. Altogether, the six texton histograms form feature vectors with approximately 3,000 dimensions, which were first reduced to 50 dimensions using PCA. Subsequently, we use t-SNE to reduce the dimensionality of the resulting feature vectors to 2 dimensions. In the high-dimensional space, we set the variance parameters σ_i in such a way that the conditional distributions P_i had a perplexity of 10.

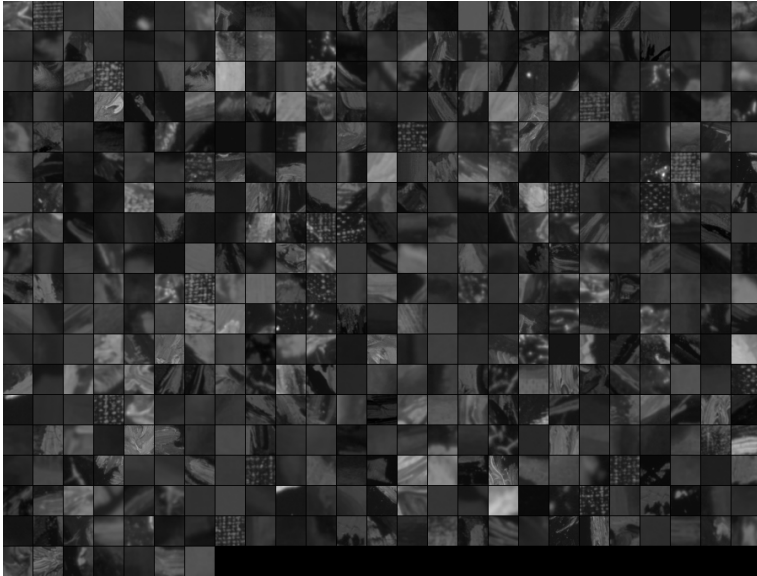


Figure 7.1 An example of one of the texton codebooks. This codebook was constructed using affinity propagation on textons of size 35×35 pixels.

7.1.2 Results

In Figure 7.2, we present one of the visualizations obtained with our approach. Each dot represents a single texton histogram (i.e., a single painting). The green dots represent Van Gogh's paintings, whereas the red dots represent established non Van Gogh paintings. Paintings of which the attributions are disputed are indicated by blue dots. The visualization reveals that all-but-two non Van Gogh paintings are depicted in the periphery of the visualization. Apparently, the brush-stroke texture in these paintings is appropriately captured by the texture histograms and offers an effective, albeit crude, indication of textural differences and similarities. The two paintings that do not stand out in the visualization are the so-called Wacker forgery and a painting by Gauguin. The Wacker forgery is one of a series of forgeries, which fooled renowned Van Gogh experts for years [Koldehoff, 2002]. The Wacker forgery in our collection is quite easy to discriminate from the genuine Van Gogh paintings using global texture analysis [Johnson *et al.*, 2008]: the Wacker forgery contains more high spatial frequencies than the genuine Van Gogh's. Presumably, the local textons do not capture these global statistics. The same may apply to the painting created by Gauguin. Further analysis is needed to establish this. Despite these two anomalies, the visualization places 11 of the 13 non-Van Gogh paintings in the periphery of the visualization. This is quite a remarkable result, given that our approach is completely data-driven and does not rely on any domain knowledge. The visualizations obtained also suggest attributions of the disputed paintings: some are located in the middle of a cluster of genuine Van Gogh paintings, whereas others are located close to the non Van Gogh paintings in the periphery.

Figure 7.3 shows a visualization obtained by applying our approach to established Van Gogh paintings only. The dots are colored according to the two main periods in Van Gogh's oeuvre:

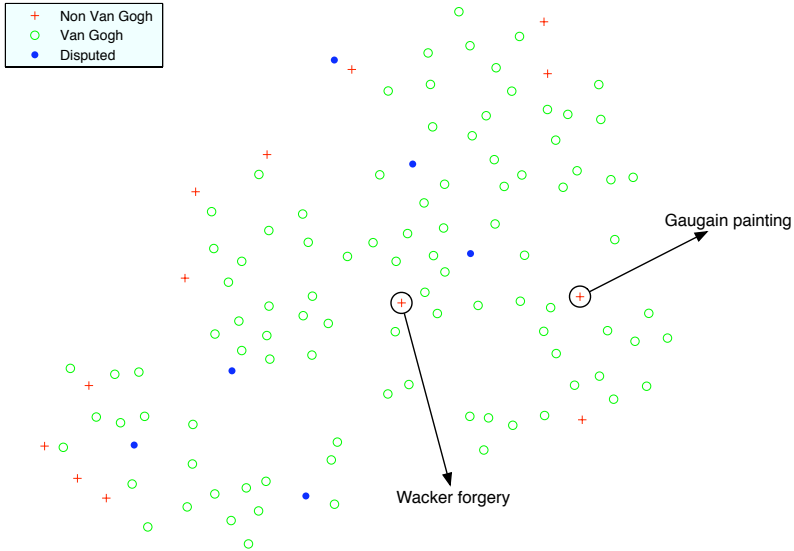


Figure 7.2 Visualization of the dataset of Van Gogh, non Van Gogh, and disputed Van Gogh paintings. The points in the scatter plot are labeled according to the authenticity of the paintings.

the Dutch period (1883-1886; red dots) and the French period (1886-1890; blue dots). The visualization shows a clear separation between the paintings from both periods (all Dutch paintings are captured in one of three small clusters), and thus captures diagnostic textural elements of the development of Van Gogh's paintings style from his originally sober style (the Dutch period) to his later lively impressionistic painting style (the French period). Art historians may use our approach to create visualizations of subsets of paintings to examine more subtle textural differences.

7.1.3 Discussion

The results presented above illustrate the potential of our approach to support art historians in their analysis of paintings. Of course, our approach only offers an initial crude characterization of paintings. A complete approach to computer-assisted artist identification should integrate more information than just the local texture characteristics that our texton histograms capture. For instance, the interaction between brushstrokes should be captured, prompting the use of textural features that are less local than our texton features. Moreover, the color use by Van Gogh (and more specifically, the use of complementary colors) should be captured in global painting features such as those proposed by Berezhnoy *et al.* [2007]. The development of such a combined approach is the most viable way to obtain clear separation between paintings that are created by different artists based on the visual assessment of the paintings. Typically, the numerical re-

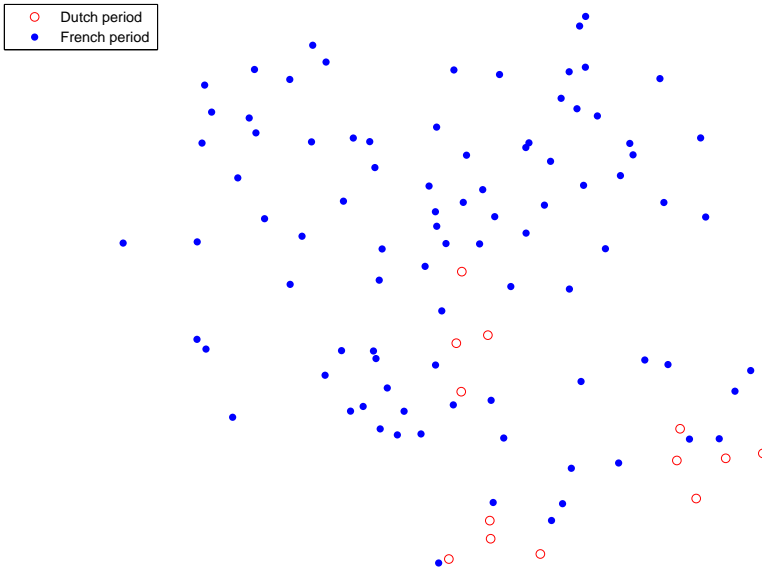


Figure 7.3 Visualization of the authentic Van Gogh paintings in the dataset. The points in the scatterplot are labeled according to the period in which they are painted.

sults obtained with our approach and other image analysis methods on digital reproductions of paintings will complement the results obtained by other types of analysis (such as provenance analysis, canvas weave count, and dendrochronological measurements).

Also, additional work is needed in order to present the results of the image analysis in intuitive ways to the art expert, because the true value for art historians is in the visualization and understanding of the visual characteristics (e.g., textons or configurations of textons) that give rise to the visualization. We envisage the future development of software that allows art experts to map and visualize subsets of paintings and selected regions of paintings. In that respect, the software that incorporates our approach may become one of the many tools at the disposal of the art historian.

7.2 Seed analysis

A seed is a small embryonic plant that is usually enclosed in a so-called seed coat, and is essential in the reproduction of plants. Four examples of seeds are shown in Figure 7.4. The analysis and classification of seeds is of relevance to, among others, biological, geological, and climatological research. Moreover, the analysis of seeds may be of interest to archaeological research, as seeds that are found in archaeological excavations provide information on the food habits of prehistoric humans. Since the analysis of seeds is a specialistic task, it is typically performed by biologists



Figure 7.4 Four examples of seeds.

or archaeobotanic experts. The analysis of seeds is a time-consuming and error-prone process, in particular, due to the large number of seed species that exist worldwide. In the Netherlands alone, the number of different seeds that are found in nature exceeds 2,000 species [Cappers *et al.*, 2006]. As a result, biologists and archaeobotanic experts would benefit from systems that assist in the analysis and classification of seeds. As state-of-the-art microscopes are often equipped with built-in digital photocaleras, such systems can now readily be integrated into the analysis process.

From a computer vision perspective, two types of features are distinctive between seed classes: (1) color-texture features and (2) shape features. In this section, we investigate the performance of an approach that uses image-based texton features, shape context features, and t-SNE in the analysis of digital photographs of seeds. Our approach in this section is similar to the approach we used in the analysis of paintings, but consists of three stages. First, we extract image-based texton features (see 6.2.2) to capture color and texture information in the seeds, and compute pairwise Euclidean distances between the texton features. Second, we supplement the pairwise distances based on texton features with pairwise dissimilarities that are based on matching shape context features (see Appendix A.2.4 for details on shape contexts) that capture the shape information in the seeds. Herein, the shape dissimilarities are given the same weight as the texture dissimilarities. Third, we construct a two-dimensional representation of the seed data by using the sum of the two pairwise dissimilarity matrices as input into t-SNE. We opt for performing visualization based on the pairwise dissimilarities and not classification, because in our seed dataset only one image per taxon (i.e., seed class) is typically available.

Below, we evaluate our approach on a dataset of digital seed photographs. We discuss the setup of the experiments in 7.2.1. The results of the experiments are presented in 7.2.2, and discussed in more detail in 7.2.3.

7.2.1 Experimental setup

In the evaluation of our approach to seed analysis, we performed experiments on a dataset of 2,434 photographs of seeds. For all seeds, various types of class information (i.e., taxon names) are available, but the class information is not used in our experiments because for most taxons only one seed image is available in the dataset. We employed image-based texton features using textons of size 7×7 pixels. The pairwise Euclidean distances between the texton features form the pairwise dissimilarity matrix D_1 . In the shape matching, we employed shape contexts features (see A.2.4 for an explanation of shape contexts) using 200 shape context descriptors with 12 radius bins and 5 distance bins (on a logarithmic scale). In the matching of the shape contexts, we employed 20 dummy nodes and 5 matching iterations (i.e., we performed the thin-plate spline warping and the Hungarian matching five times [Belongie *et al.*, 2001]). The result of the shape context matching is a pairwise dissimilarity matrix D_2 . We normalized both matrices D_1 and D_2 to lie in the range between 0 and 1 (i.e., we subtracted the minimum of the matrix values and we divided them by their maximum values). The sum of both normalized dissimilarity matrices is used as input into t-SNE. In t-SNE, the perplexity of that was used to compute the pairwise similarities in the data space was set to 20.

7.2.2 Results

In Figure 7.5, we present a visualization that was constructed using the approach described above. The insets show magnifications of the indicated regions of the visualization that illustrate the structure in the data that was captured by our approach. From the results, we observe that our approach successfully captured some of the structure in the shapes and textures of the seeds. For instance, the upper right inset shows that our approach successfully identified seeds from the *Orchidaceae* family, which are seeds that consist of a brown core within a transparent wrapping. The lower right inset shows that our approach successfully captured the members of the *Carex* family. The lower left inset shows that our approach identified seeds from the *Pinaceae* family, i.e., seeds with the form of a single helicopter rotor. Of course, the current visualization is far from perfect, as it is unlikely that all differences within the visual appearance of seeds can be captured within a mere two dimensions. However, finer structure within the visual appearance of seeds should become visible if our approach is applied onto subsets of the seed dataset, for instance, on a single family of seeds.

7.2.3 Discussion

The results obtained by a combination of texton features, shape contexts, and t-SNE illustrate the potential of our approach to the analysis of seeds. The approach may be seen as a first step towards the development of a system for the automatic classification of archaeological seeds. In order for the development of such a system to be successful, three problems need to be addressed.

First, the texture of many seeds is non-homogeneous. For instance, the texture of the periphery of the seed is often different from the texture of the center of the seed. An example of the variety in texture on a single seed surface is shown in the second seed in Figure 7.4, which consists of green surface with a brown navel. The texture features discussed in Chapter 5 and 6 are not tailored to work on non-homogeneous textures, because in the construction of the features,

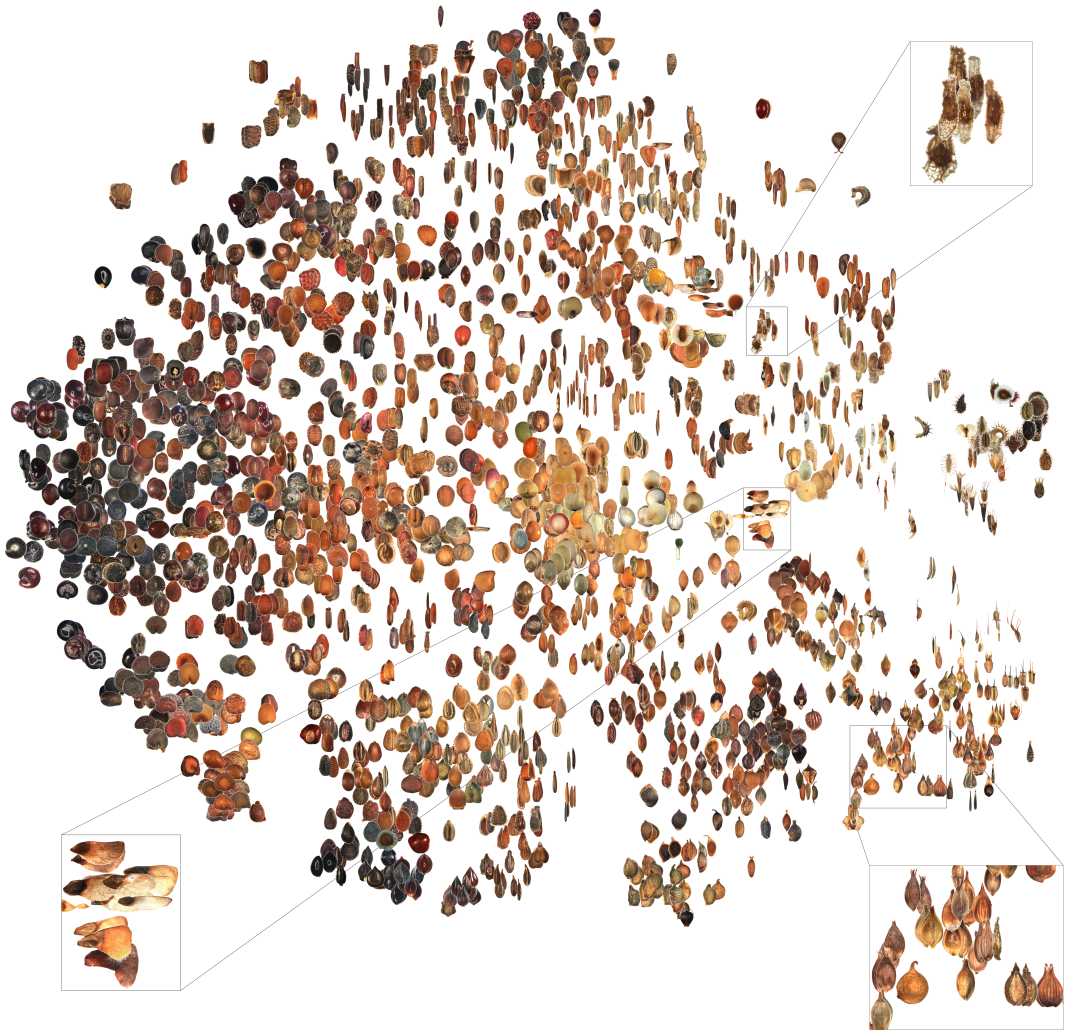


Figure 7.5 A seed map constructed by t-SNE based on texton and shape context features. The insets show magnifications of parts of the map.

they simply sum or average over all locations in the texture images. A possible solution to this problem may be to perform texture segmentation first, and model each part of the segmented texture image separately. A drawback of this approach is that it is unclear how the separate models should be combined, as summing or averaging would re-introduce the homogeneity problem. Moreover, such an approach cannot deal successfully with gradual changes in the seed texture. A more viable solution to the homogeneity problem may be to use an image model that is based on the topic models discussed in 4.2.3. Examples of such image topic models can be found in, e.g., [Fei-Fei and Perona, 2005; Sudderth *et al.*, 2005; He and Zemel, 2008]. The future success of these models in texture modeling largely depends on how well they can be trained in practice: inference in most topic models is intractable, which prompts the use of variational or Monte Carlo approximations. Also, it may be a good idea to add dependencies between neighboring pixels in the models.

Second, it is unlikely that classification of seeds can successfully be performed without additional domain knowledge, especially, since datasets with thousands of examples per seed class are not likely to become available in the near future. Such domain knowledge should include information on (1) the size and weight of the seed, (2) the order of the various dissection stages of the seed, (3) the three-dimensional shape of the seed (which can usually not be observed well from a single two-dimensional image), and (4) the relevance of the respective elements of the seed to its class. For instance, the shape and position of the navel of a seed may be of high relevance to the seed class.

Third, the system should be able to work on seeds that are found in archaeological excavations. Archaeological seeds are often charred or highly degraded due to being buried in the soil, which significantly alters their visual appearance. In particular, the shape of seeds is altered when seeds are (partially) burnt. The classification of degraded or charred seeds is challenging, even for archaeobotanic experts. A system for the automatic classification of such seeds should thus also rely on non-visual information such as domain knowledge, and knowledge about the context in which the seed was found. In this thesis, we were not able to develop such a system due to a lack of (digitized) archaeological seed data.

7.3 General discussion

The results presented in this chapter (but also those in the previous chapters) raise questions about which characteristics of the data lead to the observed structure. Is the structure in the map of Figure 7.5 the best way to represent the structure in the visual appearance of seeds on a two-dimensional plane? In Figure 7.2, the prevailing question reads: are the paintings by contemporaries separated from Van Gogh paintings because they have a different brushstroke texture? And if so, what are the main textural elements that cause the difference between Van Gogh paintings and paintings by his contemporaries?

These questions amount to a single more fundamental question that is not addressed hitherto in this thesis: *how should we evaluate unsupervised learning or feature extraction techniques?* Although visualizations such as those presented in this thesis may be informative, they do not give a quantitative measure for the quality of the extracted features. A simple quantitative evaluation that is used in the thesis is to use the labels that are assigned to the data instances to train a

classifier, and to measure the generalization performance of the trained classifier. However, this approach is not generally applicable for two main reasons. First, class labels may not be available for the data and obtaining them may be hard or costly. Second, the measured generalization performances are not necessarily informative on the quality of the features that were extracted from the data. This is illustrated by our results with parametric t-SNE in Chapter 4: the use of a (relatively) high number of degrees of freedom v sometimes leads to inferior generalization performances compared to the use of a single degree of freedom. However, in terms of the trustworthiness (which measures the extent to which the local structure of the data is retained), using a high number of degrees of freedom is beneficial compared to using a single degree of freedom. In other words, parametric t-SNE with a high number of degrees of freedom retains the local structure of the data better, i.e., it extracts *better* features, but the extracted features give rise to a lower generalization performance.

As a counterargument, one may argue that trustworthiness is probably a poor evaluation criterion, but up to the best of our knowledge, there are no obvious reasons why trustworthiness is a bad measure. The main disadvantage of the trustworthiness measure is that it is biased towards techniques that retain the local structure of the data, as a result of which PCA will typically perform inferior in terms of trustworthiness compared to t-SNE. In contrast, if the amount of variance that is retained is used as an evaluation measure, PCA will most probably outperform t-SNE (in particular, because the variance in a t-SNE map does not depend on the ‘scale’ of the data). In other words, there is not a single most appropriate evaluation measure. In fact, if such an evaluation measure existed, this would give rise to a single most appropriate technique, because the evaluation criterion can be optimized directly using techniques for non-convex optimization (assuming it is a continuous measure). Because the no-free-lunch theorem states that a single most appropriate technique does not exist [Wolpert and Macready, 1997], consequently, a single most appropriate evaluation criterion does not exist either. It thus seems unlikely that there will ever be consensus on what the most appropriate evaluation criterion for unsupervised learning or feature extraction is.

An interesting alternative to evaluating the quality of extracted features is to project the features back into the original data space. Whether this is possible depends on the feature extraction technique at hand. For instance, for PCA the backprojection can readily be performed, and for parametric t-SNE it can be performed by training decoder layers on top of the network (as suggested in 4.1.2). Texton-based texture features also facilitate backprojection into the data space. In particular, it is possible to compute the difference between the texton histograms of two texture images, and to highlight regions in one of the texture images that occur often in that texture image, but not in the other texture image (using a sliding scale). In Figure 7.6, we demonstrate this way of backprojecting texton-based texture features into the dataspace. In the example, we compare a Van Gogh painting with a painting by one of his contemporaries (viz., Claude Monet), and highlight the regions in the contemporary painting that are not very ‘Van Gogh-like’. The resulting visualization indicates that the sky of the contemporary painting is very different from the Van Gogh painting in terms of texton measurements, a result that was acknowledged by art experts.

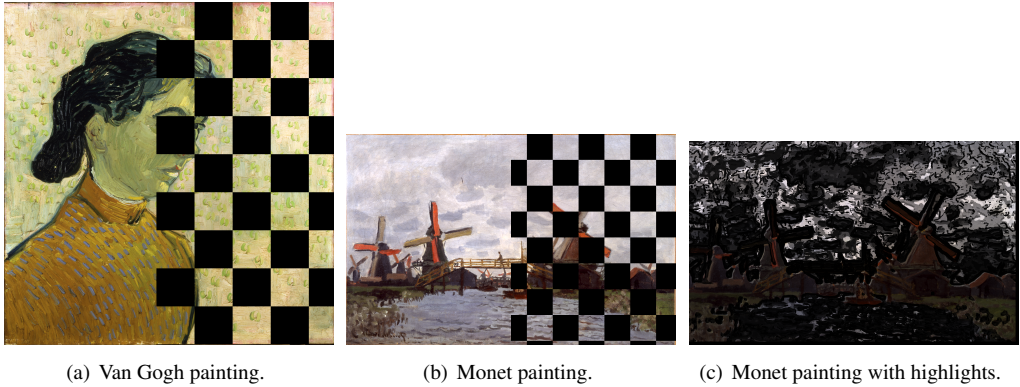


Figure 7.6 Illustration of the backprojection of texton-based texture features onto the data space. The right painting is a highlighted version of the middle painting, in which only textons are highlighted that do not occur frequently in the Van Gogh painting on the left.

7.4 Chapter conclusions

The chapter presented two applications of the texton and t-SNE features developed in this thesis to the cultural heritage domain: (1) a painting analysis application and (2) an application to the analysis of seeds. From the results obtained with the texton and t-SNE features in these two domains, we may conclude that there is a large potential for using these features in the automatic visual analysis of our cultural heritage. However, it should be noted that the successful development of classification systems for the cultural heritage also requires the incorporation of relevant domain knowledge.

8 Conclusion

Contents Our investigations in the previous chapters have led to many observations and new insights. This chapter discusses the main observations and provides answers to the two research questions of the thesis. The chapter also addresses the problem statement, which leads to a general conclusion. Finally, the chapter presents five interesting directions for future research.

Outline In Section 8.1, we answer the two research questions. Section 8.2 addresses the problem statement and draws the general conclusion of the thesis. Five directions for future research are presented in Section 8.3.

In this chapter, we first answer our two research questions (in 8.1). Subsequently, we address the problem statement, and draw a general conclusion from the work presented (in 8.2). We conclude the chapter by indicating directions for future research (in 8.3).

8.1 Answers to the research questions

Research question 1: *How can we improve existing dimensionality reduction features?*

Obviously, existing dimensionality reduction features can be improved by identifying and addressing their main weaknesses. An important weakness of many existing dimensionality reduction features is that they are designed to give rise to convex optimization problems, which is a commandable objective in itself, but often leads to structural problems in the resulting objective functions. In particular, sparse spectral dimensionality reduction techniques suffer from the simple form of their covariance constraint on the solution, which is often not capable of preventing the techniques of identifying solutions that are close to the trivial solution. Full spectral techniques focus too much on retaining large pairwise distances between datapoints, as a result of which they are not capable of preserving the local structure of the data, which is much more important.

When convexity is not a requirement in the design of a dimensionality reduction technique, it is possible to develop dimensionality reduction techniques that address the problems mentioned above. If the resulting technique is not hampered too much by the presence of poor local optima in the objective function, it may obtain superior results, as is illustrated by the strong performance of t-SNE. In addition, the development of non-convex dimensionality reduction may facilitate (1) a nonlinear parametrization of the mapping between the data space and the latent space and (2) the use of a non-metric latent space.

Research question 2: *How can existing texture features be adapted to be invariant to variations that occur in uncontrolled environments, such as lighting changes, rotations, and affine transformations?*

Invariance to lighting changes is usually obtained by employing a bank of filters on the input images. Even though these filter-based texture features have dominated over the last decade, our results illustrate that invariance to lighting changes can also be obtained by simply normalizing the pixel values of the input images. This facilitates the development of image-based texture features that are invariant to rotations and (local) affine transformations.

Rotation invariance of image-based texture features can be obtained by (1) the construction of coarse rotation-invariant intensity histograms such as spin images or (2) by exploiting the invariance properties of the Fourier coefficients of polar image representations.

Invariance to affine transformations can be obtained by performing an eigenanalysis of the second-order matrix to identify affine-covariant image regions. The affine-covariant image regions can be normalized and represented using rotation-invariant features. This gives rise to texture features that are invariant to local affine transformations.

8.2 Answer to the problem statement

On the basis of the answers to the research questions, we are now able to answer the problem statement.

How can we mitigate the problems of image-space representations?

The dimensionality problem of image-space representations that hampers computer vision systems can successfully be addressed by means of dimensionality reduction techniques such as t-SNE. This conclusion is supported by, among others, our results on the handwritten digit dataset. We showed that it is possible to reduce the hundreds of dimensions that constitute handwritten digit images to only two dimensions by exploiting the redundancy in the image-space representations, despite the fact that dimensionality reduction techniques do not exploit the spatial structure of images.

The variance problem of image-space representations that hampers computer vision systems may be addressed by exploiting the characteristics of, among others, the Fourier coefficients of polar image representations and the eigenvectors of the second-order matrix. This conclusion is supported by the results of our invariant texture representations, but readily extends to image representations in general.

Did our research contribute to achieving the general goal mentioned in Chapter 1? The goal is to improve the performance of computer vision systems. We believe that t-SNE and, in particular, its parametric counterpart form an important contribution that may help developers of computer vision systems to resolve the dimensionality problem by exploiting the strong (non)linear relations between pixel values in their input images. We specifically mention the parametric version of t-SNE, as computer vision systems generally require parametric transformations to facilitate rapid processing of input images. Our contribution to the solution of the variance problem is that we showed that the use of image-based representations facilitates the development of image features that are invariant under local affine transformations.

Based on our research findings, we arrive at the following two conclusions.

Conclusion 1: We may conclude that dimensionality reduction (e.g., by t-SNE) forms an important approach to address the dimensionality problem, in particular, when it is combined with feature extraction approaches that exploit the spatial structure of natural images.

Conclusion 2: We may conclude that the variance problem can be addressed by image representations that are invariant to lighting changes and to local affine transformations by making use of normalization, affine-covariant regions, and the Fourier coefficients of polar image representations.

8.3 Future research

Below, we mention five directions for future research.

First, future work should focus on investigating variants of t-SNE. Even though we gave some theoretical reasons for selecting the use of a Student-t distribution in the low-dimensional map

(i.e., its relation to the Gaussian distribution and its approximate scale invariance), it is not clear whether the Student-t distribution is the most suitable distribution to address the crowding problem. It would be interesting to investigate whether it is possible to resolve the crowding problem the other way around: by using a light-tailed distribution (such as the raised cosine distribution) in the high-dimensional space and a Gaussian distribution in the low-dimensional space. A possible advantage of such an approach is that it might facilitate the use of convex optimization machinery: if a Gaussian distribution is used in the low-dimensional map, the resulting minimization problem is convex with respect to the Gram matrix of the solution¹ [Globerson *et al.*, 2007].

Second, an interesting direction for future work is to investigate models that combine parametric t-SNE with other dimensionality reduction techniques based on deep-layer neural networks such as (i) autoencoders or (ii) nonlinear NCA [Salakhutdinov and Hinton, 2007]. In combination (i), the autoencoder may serve as a regularizer that forces parametric t-SNE to maximize the variance of the data in the low-dimensional representation, i.e., to exploit optimally the low-dimensional latent space available. In combination (ii), nonlinear NCA may serve as an additional learning signal to parametric t-SNE in semi-supervised learning settings, which is advantageous, because it facilitates the use of both the complete set of (labeled and unlabeled) data and the available class information.

Third, future work may focus on the development of various adaptations of multiple maps t-SNE. For instance, it is possible to develop a *mixture of maps* model, in which the similarity between two points under the model is equal to a weighted sum of the similarities between the two points in the maps. Another interesting idea is to learn multiple maps in a two-stage approach: (1) cluster the datapoints based on their pairwise similarities to obtain a collection of soft cluster assignments and (2) use these soft cluster assignments as fixed mixing proportions in the optimization of multiple maps t-SNE. The advantage of such an approach is that it may make the optimization of the multiple maps t-SNE model easier, because there are no interactions between the maps anymore (since the mixing proportions are fixed). Alternatively, one may encourage the construction of clean clusterings (or topics) in the maps by introducing a “background” map for each of the maps in which all points have pairwise distance 0 (as in UNI-SNE [Cook *et al.*, 2007]). As a result, each pair of points that have high mixing proportions in the same map are slightly similar under the model, which may lead to the emergence of cleaner data clusterings in the maps.

Fourth, future work should focus on the development of texture features that are capable of modeling non-homogeneous textures. Today, the most promising approaches for modeling non-homogeneous texture are topic models that are tailored to work on image data, such as the models presented by Fei-Fei and Perona [2005]; Sudderth *et al.* [2005]; He and Zemel [2008]. Currently, the most important problems of these models that need to be addressed are (i) their intractable nature and (ii) their lack of sufficient dependencies between neighboring pixel values in the model. Also, it is unclear if these topic models (which are essentially mixture models) are more appropriate probabilistic models for images than product models such as the field of experts model [Roth and Black, 2005].

¹An additional difficulty is that there is no constraint on the rank of the Gram matrix of the solution, as a result of which a trivial solution may be selected. This problem may be addressed by introducing an L1-regularizer on the trace of the Gram matrix.

Fifth, interesting future work can be performed in incorporating color information into our affine-invariant texture features. Even though color is an important cue in human vision, it has largely remained unexplored in texture modeling and image modeling in general. A notable exception is the study that is presented by Burghouts and Geusebroek [2008], which develops a number of color invariants using the Gaussian opponent color model [Geusebroek *et al.*, 2001] that are invariant to (i) regional intensity variation, (ii) Lambertian reflectance, and (iii) Fresnel reflectance². Burghouts and Geusebroek [2006] developed color invariants for use in texton-based texture features, as a result of which they can readily be incorporated into our affine-invariant texton-based texture features.

²Invariance to Lambertian reflectance implies that the representations are invariant under shadows and shading. Invariance to the Fresnel coefficient implies that the representations are invariant to “highlights” in the image.

References

- Abdella, M. and Marwala, T. (2005). The use of genetic algorithms and neural networks to approximate missing data in database. In *Proceedings of the IEEE International Conference on Computational Cybernetics*, pages 207–212. Cited on page 21.
- Abdelnour, A. and Selesnick, I. (2001). Design of 2-band orthogonal near-symmetric CQF. In *Proceedings of the IEEE International Conference Acoustic, Speech, and Signal Processing*, volume 6, pages 3693–3696. Cited on pages 93 and 94.
- Ackley, D., Hinton, G., and Sejnowski, T. (1985). A learning algorithm for Boltzmann Machines. *Cognitive Science*, 9:147–169. Cited on pages 85 and 171.
- Afken, G. (1985). *Gram-Schmidt Orthogonalization*. Academic Press, Orlando, FL. Cited on page 19.
- Agrafiotis, D. (2003). Stochastic proximity embedding. *Journal of Computational Chemistry*, 24(10):1215–1221. Cited on pages 9 and 20.
- Anderson, W. and Morley, T. (1985). Eigenvalues of the Laplacian of a graph. *Linear and Multilinear Algebra*, 18:141–145. Cited on page 18.
- Arnoldi, W. (1951). The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–25. Cited on page 27.
- Baker, C. (1977). *The numerical treatment of integral equations*. Clarendon Press, Oxford, UK. Cited on page 24.
- Balasubramanian, M. and Schwartz, E. (2002). The Isomap algorithm and topological stability. *Science*, 295(5552):7. Cited on page 12.
- Baudat, G. and Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404. Cited on page 9.
- Belkin, M. and Niyogi, P. (2002). Laplacian Eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14, pages 585–591, Cambridge, MA. The MIT Press. Cited on pages 17 and 57.
- Belkin, M. and Niyogi, P. (2004). Semi-supervised learning on Riemannian manifolds. *Machine Learning*, 56(1–3):209–239. Cited on page 18.
- Bell, A. and Sejnowski, T. (1995). An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159. Cited on page 9.

- Belongie, S., Malik, J., and Puzicha, J. (2001). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522. Cited on pages 124, 158, and 162.
- Bengio, Y. (2007). Learning deep architectures for AI. Technical Report 1312, Université de Montréal. Cited on pages 37, 57, and 58.
- Bengio, Y., Delalleau, O., Roux, N. L., Vincent, J.-F. P. P., and Ouimet, M. (2004a). Learning eigenfunctions links spectral embedding and Kernel PCA. *Neural Computation*, 16(10):2197–2219. Cited on page 24.
- Bengio, Y. and LeCun, Y. (2007). Scaling learning algorithms towards AI. In Bottou, L., Chapelle, O., DeCoste, D., and Weston, J., editors, *Large-Scale Kernel Machines*, pages 321–360. The MIT Press. Cited on pages 35 and 37.
- Bengio, Y. and Monperrus, M. (2004). Non-local manifold tangent learning. In *Advances in Neural Information Processing Systems*, volume 17, pages 129–136, Cambridge, MA. The MIT Press. Cited on pages 35 and 38.
- Bengio, Y., Paiement, J.-F., Vincent, P., Delalleau, O., Roux, N. L., and Ouimet, M. (2004b). Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems*, volume 16, pages 177–184, Cambridge, MA. The MIT Press. Cited on pages 24 and 27.
- Berezhnoy, I., Postma, E., and van den Herik, J. (2007). Computer analysis of Van Gogh's complementary colours. *Pattern Recognition Letters*, 28(6):703–709. Cited on pages 118 and 121.
- Betechuoh, B., Marwala, T., and Tettey, T. (2006). Autoencoder networks for HIV classification. *Current Science*, 91(11):1467–1473. Cited on page 21.
- Bezdidko, S. (1974). The use of Zernike polynomials in optics. *Sovjet Journal on Optical Technology*, 41:425. Cited on page 159.
- Bianconi, F. and Fernández, A. (2007). Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern Recognition*, 40:3325–3335. Cited on page 89.
- Biggs, N. (1974). Algebraic graph theory. In *Cambridge Tracts in Mathematics*, volume 67. Cambridge University Press. Cited on page 169.
- Bishop, C. (2006). *Pattern recognition and machine learning*. Springer, New York, NY. Cited on pages 2, 5, and 99.
- Bishop, C., Svensen, M., and Williams, C. (1998). GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234. Cited on page 8.
- Blei, D., Griffiths, T., Jordan, M., and Tenenbaum, J. (2004). Hierarchical topic models and the nested Chinese restaurant process. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems*, volume 16, pages 17–24, Cambridge, MA. The MIT Press. Cited on page 79.
- Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022. Cited on page 79.

- Blunsden, S. (2004). Texture classification using non-parametric Markov Random Fields. Master's thesis, School of Informatics, University of Edinburgh. Cited on page 85.
- Bookstein, F. (1989). Principal warps: Thin-plate splines and decomposition of transformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585. Cited on page 163.
- Borchers, B. and Young, J. (2007). Implementation of a primaldual method for SDP on a shared memory parallel architecture. *Computational Optimization and Applications*, 37(3):355–369. Cited on page 26.
- Bovik, A. (1991). Analysis of multichannel narrow-band filters for image texture segmentation. *IEEE Transactions on Signal Processing*, 39(9):2025–2034. Cited on page 88.
- Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press, New York, NY. Cited on page 10.
- Brand, M. (2002). Charting a manifold. In *Advances in Neural Information Processing Systems*, volume 15, pages 985–992, Cambridge, MA. The MIT Press. Cited on pages 23 and 36.
- Brand, M. (2004). From subspaces to submanifolds. In *Proceedings of the 15th British Machine Vision Conference*, London, UK. British Machine Vision Association. Cited on pages 9 and 35.
- Bridle, J. (1989). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Fogelman-Soulie, F. and Héroult, J., editors, *Neurocomputing: Algorithms, Architectures, and Applications*. Springer-Verlag, New York, NY. Cited on page 70.
- Brosnan, T. and Sun, D.-W. (2004). Improving quality inspection of food products by computer vision – a review. *Journal of Food Engineering*, 61(1):3–16. Cited on page 2.
- Brun, A., Park, H.-J., Knutsson, H., and Westin, C.-F. (2003). Coloring of dt-mri fiber traces using laplacian eigenmaps. In *Proceedings of the Eurocast 2003, Neuro Image Workshop*. Cited on page 18.
- Bulacu, M. and Schomaker, L. (2007). Text-independent writer identification and verification using textural and allographic features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):701–717. Cited on page 177.
- Bulacu, M., Schomaker, L., and Vuurpijl, L. (2003). Writer identification using edge-based directional features. In *Proceedings of ICDAR 2003*, pages 937–941. Cited on page 164.
- Burges, C. (2005). *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, chapter Geometric Methods for Feature Selection and Dimensional Reduction: A Guided Tour. Kluwer Academic Publishers, Dordrecht, The Netherlands. Cited on page 8.
- Burghouts, G. and Geusebroek, J. (2006). Color textons for texture recognition. In *Proceedings of the British Machine Vision Conference*, volume 3, pages 1099–1108. Cited on page 133.
- Burghouts, G. and Geusebroek, J. (2008). Performance evaluation of local color invariants. *Computer Vision and Image Understanding*, (in press). Cited on pages 115 and 133.

- Caelli, T. and Julesz, B. (1978). Experiments in the visual perception of texture. *Biological Cybernetics*, 28:167–175. Cited on page 87.
- Cappers, R., Bekker, R., and Jans, J. (2006). *Digital Seed Atlas of The Netherlands*. Barkhuis Publishing, Groningen, The Netherlands. Cited on page 123.
- Caputo, B., Hayman, E., and Mallikarjuna, P. (2005). Class-specific material categorisation. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1597–1604. Cited on pages 94 and 98.
- Caruana, R., Lawrence, S., and Giles, L. (2001). Overtting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances of Neural Information Processing Systems*, volume 13, pages 402–408. Cited on page 67.
- Chang, H., Yeung, D.-Y., and Xiong, Y. (2004). Super-resolution through neighbor embedding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 275–282. Cited on page 17.
- Chang, K.-Y. and Ghosh, J. (1998). Principal curves for nonlinear feature extraction and classification. In *Applications of Artificial Neural Networks in Image Processing III*, pages 120–129, Bellingham, WA. SPIE. Cited on pages 9 and 36.
- Chatfield, C. and Collins, A. (1980). *Introduction to Multivariate Analysis*. Chapman and Hall. Cited on pages 12 and 13.
- Chin, T.-J. and Suter, D. (2008). Out-of-sample extrapolation of learned manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1547–1556. Cited on page 28.
- Choi, H. and Choi, S. (2007). Robust kernel Isomap. *Pattern Recognition*, 40(3):853–862. Cited on pages 12 and 27.
- Chui, C. (1992). *An Introduction to Wavelets*. Elsevier, Amsterdam, The Netherlands. Cited on page 92.
- Clifford, P. (1990). Markov Random Fields in statistics. In Grimmett, G. and Welsh, D., editors, *Disorder in Physical Systems. A Volume in Honour of John M. Hammersley*, pages 19–32. Oxford Press. Cited on page 86.
- Cohen, P. (1995). *Empirical Methods for Artificial Intelligence*. The MIT Press, Cambridge, MA. Cited on page 5.
- Collins, A. and Loftus, E. (1975). A spreading activation theory of semantic processing. *Psychological Review*, 82:407–428. Cited on page 78.
- Cook, J., Sutskever, I., Mnih, A., and Hinton, G. (2007). Visualizing similarity data with a mixture of maps. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, volume 2, pages 67–74. Cited on pages 43, 44, 45, 70, and 132.
- Costa, J. and Hero, A. (2005). Classification constrained dimensionality reduction. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 1077–1080. Cited on page 18.

- Cox, T. and Cox, M. (1994). *Multidimensional scaling*. Chapman & Hall, London, UK. Cited on page 21.
- Cross, G. (1980). *Markov Random Field texture models*. PhD thesis, Michigan State University, East Lansing, MI. Cited on page 85.
- Cula, O. and Dana, K. (2004). 3D texture recognition using bidirectional feature histograms. *International Journal of Computer Vision*, 59(1):33–60. Cited on pages 94, 98, and 100.
- Dana, K., van Ginneken, B., Nayar, S., and Koenderink, J. (1999). Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34. Cited on page 101.
- Daubechies, I. (1992). *Ten lectures on wavelets*. SIAM CBMS-61. Cited on page 92.
- Daugman, G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169. Cited on page 88.
- de Ridder, D. and Franc, V. (2003a). Robust manifold learning. Technical Report CTU-CMP-2003-08, Department of Cybernetics, Czech Technical University, Prague, Czech Republic. Cited on page 36.
- de Ridder, D. and Franc, V. (2003b). Robust subspace mixture models using t-distributions. In *Proceedings of the British Machine Vision Conference 2003*, pages 319–328. Cited on page 36.
- de Silva, V. and Tenenbaum, J. (2003). Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems*, volume 15, pages 721–728, Cambridge, MA. The MIT Press. Cited on pages 27 and 58.
- Demartines, P. and Héroult, J. (1997). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154. Cited on pages 9, 20, and 57.
- DeMers, D. and Cottrell, G. (1993). Non-linear dimensionality reduction. In *Advances in Neural Information Processing Systems*, volume 5, pages 580–587, San Mateo, CA. Morgan Kaufmann. Cited on page 21.
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38. Cited on page 22.
- Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271. Cited on page 12.
- Donoho, D. and Grimes, C. (2005). Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431. Cited on pages 18 and 19.
- Doyle, P. and Snell, L. (1984). Random walks and electric networks. In *Carus mathematical monographs*, volume 22. Mathematical Association of America. Cited on page 169.
- Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Cited on page 163.

- Dunn, D. and Higgins, W. (1995). Optimal Gabor filters for texture segmentation. *IEEE Transaction on Image Processing*, 4(7):947–964. Cited on page 89.
- Duraiswami, R. and Raykar, V. (2005). The manifolds of spatial hearing. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 285–288. Cited on page 17.
- Edelman, S. and Duvdevani-Bar, S. (1997). Similarity, connectionism, and the problem of representation in vision. *Neural Computation*. Cited on page 77.
- Efros, A. and Leung, T. (1999). Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1033–1038. Cited on page 86.
- Faichney, J. and Gonzalez, R. (2002). Combined colour and contour representation using anti-aliased histograms. In *Proceedings of the 6th International Conference on Signal Processing*, pages 735–739. Cited on page 111.
- Faloutsos, C. and Lin, K.-I. (1995). FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174, New York, NY. ACM Press. Cited on page 9.
- Fang, J. and Qiu, G. (2003). Human face detection using angular radial transform and support vector machines. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 669–672. Cited on page 161.
- Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531. Cited on pages 126 and 132.
- Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188. Cited on page 9.
- Floyd, R. (1962). Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345. Cited on page 12.
- Fokkema, D., Sleijpen, G., and van der Vorst, H. (1999). Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM Journal on Scientific Computing*, 20(1):94–125. Cited on page 27.
- Forsyth, D. and Ponce, J. (2003). *Computer vision: A modern approach*. Prentice Hall. Cited on pages 3 and 163.
- Freeman, W. and Adelson, E. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906. Cited on page 91.
- Frey, B. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315:972–976. Cited on pages 99 and 177.
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press Professional, Inc., San Diego, CA. Cited on page 8.

- Gelman, A., Carlin, J., Stern, H., and Rubin, D. (1995). *Bayesian data analysis*. Chapman & Hall, New York, NY. Cited on page 79.
- Geusebroek, J., Boomgaard, R., Smeulders, A., and Geerts, H. (2001). Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350. Cited on page 133.
- Ghahramani, Z. and Hinton, G. (1996a). The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto. Cited on page 22.
- Ghahramani, Z. and Hinton, G. (1996b). Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, Department of Computer Science, University of Toronto. Cited on page 85.
- Gibbs, J. (1898). Fourier series. *Nature*, 59. Cited on page 92.
- Globerson, A., Chechik, G., Pereira, F., and Tishby, N. (2007). Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295. Cited on page 132.
- Goldberg, Y., Zakai, A., Kushnir, D., and Ritov, Y. (2008). Manifold learning: The price of normalization. *Journal of Machine Learning Research*, 9:1909–1939. Cited on page 35.
- Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. (2005). Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, volume 17, pages 513–520, Cambridge, MA. The MIT Press. Cited on page 9.
- Golub, G., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–224. Cited on page 34.
- Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783. Cited on pages 53 and 169.
- Graepel, T. (2002). Kernel matrix completion by semidefinite programming. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 694–699, Berlin, Germany. Springer-Verlag. Cited on page 34.
- Graf, A. and Wichmann, F. (2002). Gender classification of human faces. In *Biologically Motivated Computer Vision 2002, LNCS 2525*, pages 491–501. Cited on page 37.
- Griffiths, T., Steyvers, M., and Tenenbaum, J. (2007). Topics in semantic representation. *Psychological Review*, 114(2):211–244. Cited on pages 77 and 79.
- Grigorescu, C. and Petkov, N. (2003). Distance sets for shape filters and shape recognition. *IEEE Transactions on Image Processing*, 12(10):1274–1286. Cited on page 158.
- Hamm, J., Lee, D., Mika, S., and Schölkopf, B. (2003). A kernel view of the dimensionality reduction of manifolds. Technical Report TR-110, Max Planck Institute for Biological Cybernetics, Germany. Cited on page 24.
- Haralick, R. and Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3. Cited on page 85.

- He, X., Cai, D., Yan, S., and Zhang, H.-J. (2005). Neighborhood preserving embedding. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, pages 1208–1213. Cited on pages 9, 17, and 18.
- He, X. and Niyogi, P. (2004). Locality preserving projections. In *Advances in Neural Information Processing Systems*, volume 16, page 37, Cambridge, MA. The MIT Press. Cited on pages 9 and 18.
- He, X. and Zemel, R. (2008). Latent topic random fields: Learning using a taxonomy of labels. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8. Cited on pages 126 and 132.
- Hilbert, D. (1953). *Grundzüge einer allgemeinen Theorie der linearen Integralgleichungen*. Chelsea Pub. Co. Cited on page 93.
- Hinton, G. (1981). Implementing semantic networks in parallel hardware. In Hinton, G. and Anderson, J., editors, *Parallel Models of Associative Memory*. Erlbaum, Hillsdale, NJ. Cited on page 78.
- Hinton, G. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800. Cited on pages 21, 87, 171, and 172.
- Hinton, G., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554. Cited on page 21.
- Hinton, G. and Roweis, S. (2002). Stochastic Neighbor Embedding. In *Advances in Neural Information Processing Systems*, volume 15, pages 833–840, Cambridge, MA. The MIT Press. Cited on pages 20, 40, and 43.
- Hinton, G. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507. Cited on pages 21, 58, 60, 63, and 64.
- Hoffmann, H. (2007). Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874. Cited on page 14.
- Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22th Annual International SIGIR Conference*, pages 50–57, New York, NY. ACM Press. Cited on page 77.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441. Cited on pages 8, 11, 40, and 54.
- Huber, R., Ramoser, H., Mayer, K., Penz, H., and Rubik, M. (2005). Classification of coins using an eigenspace approach. *Pattern Recognition Letters*, 26(1):61–75. Cited on pages 2 and 12.
- Hughes, N. and Tarassenko, L. (2003). Novel signal shape descriptors through wavelet transforms and dimensionality reduction. In *Wavelet Applications in Signal and Image Processing X*, pages 763–773. Cited on page 37.
- Huttenlocher, D., Klanderman, D., and Rucklidge, A. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863. Cited on page 158.

- Hyvärinen, A., Hurri, J., and Hoyer, P. (2008). *Natural Image Statistics: A probabilistic approach to computational early vision*. Springer, New York, NY. Cited on page 87.
- Jacobs, R. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1:295–307. Cited on pages 47, 48, and 73.
- Jain, A. and Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244. Cited on page 164.
- Jenkins, O. and Mataric, M. (2002). Deriving action and behavior primitives from human motion data. In *International Conference on Intelligent Robots and Systems*, volume 3, pages 2551–2556. Cited on pages 17 and 37.
- Jimenez, L. and Landgrebe, D. (1997). Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man and Cybernetics*, 28(1):39–54. Cited on pages 2 and 8.
- Johnson, A. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449. Cited on page 103.
- Johnson, C., Hendriks, E., Bereznoy, I., Brevdo, E., Hughes, S., Daubechies, I., Li, J., Postma, E., and Wang, J. (2008). Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4):37–48. Cited on pages 118 and 120.
- Jones, J. and Palmer, L. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258. Cited on pages 88 and 101.
- Julesz, B. (1962). Visual pattern discrimination. *IRE Transactions on Information Theory*, 8:84–92. Cited on page 87.
- Julesz, B. (1981). Textons, the element of texture perception and their interactions. *Nature*, 290:91–97. Cited on pages 85 and 94.
- Julesz, B., Gilbert, E., and Victor, J. (1978). Visual discrimination of textures with identical third-order statistics. *Biological Cybernetics*, 31:137–140. Cited on page 87.
- Kadir, T. and Brady, M. (2001). Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105. Cited on page 114.
- Kakutani, S. (1945). Markov processes and the Dirichlet problem. *Proceedings of the Japan Academy*, 21:227–233. Cited on page 169.
- Kalman, R. (1963). Mathematical description of linear dynamical systems. *SIAM Journal on Control and Optimization*, 1(2):152–192. Cited on page 85.
- Kam, M., Fielding, G., and Conn, R. (1997). Writer identification by professional document examiners. *Journal of Forensic Sciences*, 42:778–785. Cited on page 177.
- Kambhatla, N. and Leen, T. (1997). Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516. Cited on page 22.

- Kang, Y., Morooka, K., and Nagahashi, H. (2005). Scale invariant texture analysis using multi-scale local autocorrelation features. In *Lecture Notes in Computer Science*, volume 3459, pages 363–373, New York, NY: Springer-Verlag. Cited on page 85.
- Kashyap, R. and Khotanzed, A. (1986). A model-based method for rotation invariant texture classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(7):472–481. Cited on page 85.
- Kawamoto, A. (1993). Nonlinear dynamics in the resolution of lexical ambiguity: A parallel distributed processing account. *Journal of Memory and Language*, 32:474–516. Cited on page 78.
- Kharal, R. (2006). Semidefinite embedding for the dimensionality reduction of DNA microarray data. Master's thesis, University of Waterloo, Canada. Cited on page 15.
- Kim, H.-K., Kim, J.-D., Sim, D.-G., and Oh, D.-I. (2000). A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume 1, pages 307–310. Cited on pages 158, 159, and 160.
- Kim, K., Jung, K., and Kim, H. (2002). Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, 9(2):40–42. Cited on page 14.
- Kim, W. and Kim, Y. (1999). A new region-based shape descriptor: The ART (Angular Radial Transform) descriptor. Technical Report MPEG99/M5472, ISO/IEC JTC1/SC29/WG11. Cited on page 160.
- Kingsbury, N. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Applied and Computational Harmonic Analysis*, 10(3):234–253. Cited on page 93.
- Koenderink, J. (1984). The structure of images. *Biological Cybernetics*, 50:363–396. Cited on page 161.
- Kohonen, T. (1989). *Self-organization and associative memory: 3rd edition*. Springer-Verlag New York, Inc., New York, NY. Cited on pages 8, 99, and 177.
- Kokiopoulou, E. and Saad, Y. (2007). Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2134–2156. Cited on pages 9 and 17.
- Koldehoff, S. (2002). The Wacker forgeries: A catalogue. *Van Gogh Museum Journal*, pages 138–149. Cited on page 120.
- Kuhn, H. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97. Cited on page 163.
- Kung, S., Diamantaras, K., and Taur, J. (1994). Adaptive Principal component EXtraction (APEX) and applications. *IEEE Transactions on Signal Processing*, 42(5):1202–1217. Cited on page 24.

- Lafon, S. and Lee, A. (2006). Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403. Cited on pages 15, 16, 53, and 56.
- Lanckriet, G., Cristianini, N., Bartlett, P., and Jordan, L. G. M. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72. Cited on page 34.
- Landauer, T. and Dumais, S. (1997). A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240. Cited on page 77.
- Laub, J., Macke, J., Müller, K.-R., and Wichmann, F. (2007). Inducing metric violations in human similarity judgements. In *Advances in Neural Information Processing Systems*, volume 19, pages 777–784. Cited on page 77.
- Laub, J. and Müller, K.-R. (2004). Feature discovery in non-metric pairwise data. *Journal of Machine Learning Research*, 5:801–818. Cited on page 77.
- Law, M. and Jain, A. (2006). Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 28(3):377–391. Cited on page 27.
- Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278. Cited on pages 4, 107, 114, and 157.
- Lee, J., Lendasse, A., Donckers, N., and Verleysen, M. (2000). A robust nonlinear projection method. In *Proceedings of the 8th European Symposium on Artificial Neural Networks*, pages 13–20. Cited on pages 20 and 57.
- Lee, J. and Verleysen, M. (2005). Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing*, 67:29–53. Cited on pages 12, 13, and 53.
- Lee, J. and Verleysen, M. (2007). *Nonlinear dimensionality reduction*. Springer, New York, NY. Cited on pages 4 and 8.
- Leung, T. and Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44. Cited on pages 94, 98, and 100.
- Levina, E. and Bickel, P. (2004). Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems*, volume 17, Cambridge, MA. The MIT Press. Cited on page 29.
- Li, H., Teng, L., Chen, W., and Shen, I.-F. (2005). Supervised learning on local tangent space. In *Lecture Notes on Computer Science*, volume 3496, pages 546–551, Berlin, Germany. Springer Verlag. Cited on page 28.
- Li, J. and Allinson, N. (2007). A comprehensive review of current local features for computer vision. Cited on page 157.

- Lim, I., Ciechomski, P., Sarni, S., and Thalmann, D. (2003). Planar arrangement of high-dimensional biomedical data sets by Isomap coordinates. In *Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems*, pages 50–55. Cited on pages 13, 17, and 37.
- Lima, A., Zen, H., Nankaku, Y., Miyajima, C., Tokuda, K., and Kitamura, T. (2004). On the use of Kernel PCA for feature extraction in speech recognition. *IEICE Transactions on Information Systems*, E87-D(12):2802–2811. Cited on page 14.
- Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116. Cited on page 114.
- Lindman, H. (1974). *Analysis of variance in complex experimental designs*. W.H. Freeman & Co., San Francisco, CA. Cited on page 5.
- Liu, H. and Motoda, H. (1998). *Feature Extraction, Construction and Selection: A Datamining Perspective*. Springer, New York, NY. Cited on page 3.
- Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. Cited on pages 4 and 157.
- Luce, R. (1963). Detection and recognition. In Luce, R., Bush, R., and Galanter, E., editors, *Handbook of Mathematical Psychology*, pages 103–190, New York, NY. Wiley. Cited on page 68.
- McCallum, A. (1999). Multi-label text classification with a mixture model trained by em. In *AAAI Workshop on Text Learning*. Cited on page 79.
- McCallum, A., Corrada-Emmanuel, A., and Wang, X. (2004). The author-recipient-topic model for topic and role discovery in social networks: Experiments with Enron and academic email. Technical Report UM-CS-2004-096, Department of Computer Science, University of Massachusetts, Amherst, MA. Cited on page 79.
- McClelland, J. and Rumelhart, D. (1981). An interactive activation model of context effects in letter perception: Part 1. an account of basic findings. *Psychological Review*, 88(5):375–407. Cited on page 78.
- Mekuz, N. and Tsotsos, J. (2006). Parameterless Isomap with adaptive neighborhood selection. In *Proceedings of the 28th DAGM Symposium*, pages 364–373, Berlin, Germany. Springer. Cited on page 36.
- Mellor, M., Hong, B.-W., and Brady, M. (2008). Locally rotation, contrast, and scale invariant descriptors for texture analysis. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 30(1):52–61. Cited on pages 4 and 101.
- Meytlis, M. and Sirovich, L. (2007). On the dimensionality of face space. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 29(7):1262–1267. Cited on pages 35 and 58.
- Mikolajczyk, K. and Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86. Cited on page 105.
- Mnih, A. and Hinton, G. (2007). Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648. Cited on page 48.

- Mokhtarian, F., Abbasi, S., and Kittler, J. (1996). Efficient and robust retrieval by shape content through curvature scale space. In *Proceedings of the International Workshop on Image Databases and Multimedia Search*, pages 35–42. Cited on pages 158, 161, and 162.
- Mokhtarian, F. and Suomela, R. (1998). Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1376–1381. Cited on page 162.
- Mori, G., Belongie, S., and Malik, J. (2005). Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837. Cited on pages 158 and 163.
- Mori, S., Nishida, H., and Yamada, H. (1999). *Optical character recognition*. John Wiley & Sons, Hoboken, NJ. Cited on page 2.
- Motoyoshi, I., Nishida, S., Sharan, L., and Adelson, E. (2007). Visual perception: A gloss on surface properties. *Nature*, 447:206–209. Cited on page 88.
- Munkres, J. (2000). *Topology: A First Course, 2nd edition*. Prentice-Hall, Upper Saddle River, NJ. Cited on page 68.
- Nadler, B., Lafon, S., Coifman, R., and Kevrekidis, I. (2006). Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis: Special Issue on Diffusion Maps and Wavelets*, 21:113–127. Cited on pages 15 and 53.
- Nam, K., Je, H., and Choi, S. (2004). Fast Stochastic Neighbor Embedding: A trust-region algorithm. In *Proceedings of the IEEE International Joint Conference on Neural Networks 2004*, volume 1, pages 123–128, Budapest, Hungary. Cited on page 20.
- Nelson, D., McEvoy, C., and Schreiber, T. (1998). The University of South Florida word association, rhyme, and word fragment norms. Cited on page 73.
- Nene, S., Nayar, S., and Murase, H. (1996). Columbia Object Image Library (COIL-20). Technical Report CUCS-005-96, Columbia University. Cited on page 48.
- Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, Cambridge, MA. The MIT Press. Cited on page 18.
- Niskanen, M. and Silvén, O. (2003). Comparison of dimensionality reduction methods for wood surface inspection. In *Proceedings of the 6th International Conference on Quality Control by Artificial Vision*, pages 178–188, Gatlinburg, TN. International Society for Optical Engineering. Cited on pages 8, 13, and 37.
- Olshausen, B. and Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609. Cited on page 87.
- Parisi-Baradad, V., Lombarte, A., Garca-Ladona, E., Cabestany, J., Piera, J., and Chic, O. (2005). Otolith shape contour analysis using affine transformation invariant wavelet transforms and curvature scale space representation. *Marine and Freshwater Research*, 56:795–804. Cited on page 162.

- Park, J.-H., Zhang, Z., Zha, H., and Kasturi, R. (2004). Local smoothing for manifold learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 452–459. Cited on page 36.
- Partridge, M. and Calvo, R. (1997). Fast dimensionality reduction and Simple PCA. *Intelligent Data Analysis*, 2(3):292–298. Cited on page 12.
- Passeraub, P., Besse, P.-A., de Raad, C., Dezuari, O., Quinet, F., and Popovic, R. (1997). Coin recognition using an inductive proximity sensor microsystem. In *International Conference on Solid State Sensors and Actuators*, volume 1, pages 389–392. Cited on page 178.
- Patwari, N. and Hero, A. (2004). Manifold learning algorithms for localization in wireless sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 857–860. Cited on page 19.
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572. Cited on pages 8, 11, and 54.
- Platt, J. (2005). FastMap, MetricMap, and Landmark MDS are all Nyström algorithms. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 261–268. Cited on pages 12, 24, and 27.
- Plaut, D. (1997). Structure and function in the lexical system: Insights from distributed models of word reading and lexical decision. *Language and Cognitive Processes*, 12:765–805. Cited on page 78.
- Portilla, J. and Simoncelli, E. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71. Cited on pages 87, 88, and 91.
- Posadas, A., Vidal, F., de Miguel, F., Alguacil, G., Pena, J., Ibanez, J., and Morales, J. (1993). Spatial-temporal analysis of a seismic series using the principal components method. *Journal of Geophysical Research*, 98(B2):1923–1932. Cited on page 12.
- Rabiner, L. and Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1). Cited on page 85.
- Rajpoot, N., Arif, M., and Bhalerao, A. (2007). Unsupervised learning of shape manifolds. In *Proceedings of the British Machine Vision Conference*. Cited on page 16.
- Randen, T. (1997). *Filter and Filter Bank Design for Image Texture Recognition*. PhD thesis, Norwegian University of Science and Technology. Cited on page 85.
- Randen, T. and Husoy, J. (1999). Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310. Cited on page 88.
- Raytchev, B., Yoda, I., and Sakaue, K. (2004). Head pose estimation by nonlinear manifold learning. In *Proceedings of the 17th ICPR*, pages 462–466. Cited on page 13.
- Reed, T. and du Buf, J. H. (1993). A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding*, 57(3):359–372. Cited on page 3.

- Ricard, J., Coeurjolly, D., and Baskurt, A. (2005). Generalizations of angular radial transform for 2d and 3d shape retrieval. *Pattern Recognition Letters*, 26(14):2174–2186. Cited on pages 158 and 161.
- Rodd, J., Gaskell, M., and Marslen-Wilson, W. (2004). Modelling the effects of semantic ambiguity in word recognition. *Cognitive Science*, 28:89–104. Cited on page 78.
- Rosen-Zvi, M., Griffiths, T., and Smyth, M. S. P. (2004). The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, Arlington, VA. AUAI Press. Cited on page 79.
- Roth, S. and Black, M. (2005). Fields of experts: A framework for learning image priors. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 860–867. Cited on pages 87 and 132.
- Roweis, S. (1997). EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems*, volume 10, pages 626–632. Cited on page 12.
- Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326. Cited on pages 8, 16, 17, and 30.
- Roweis, S., Saul, L., and Hinton, G. (2001). Global coordination of local linear models. In *Advances in Neural Information Processing Systems*, volume 14, pages 889–896, Cambridge, MA. The MIT Press. Cited on pages 9 and 35.
- Sablatnig, R., Kammerer, P., and Zolda, E. (1998). Hierarchical classification of paintings using face- and brush stroke models. In *Proceedings of the International Conference on Pattern Recognition*, pages 172–174. Cited on page 118.
- Salakhutdinov, R. and Hinton, G. (2007). Learning a non-linear embedding by preserving class neighbourhood structure. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, volume 2, pages 412–419. Cited on pages 9, 61, 63, 67, and 132.
- Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted Boltzmann Machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, pages 791–798. Cited on page 48.
- Samko, O., Marshall, A., and Rosin, P. (2006). Selection of the optimal parameter value for the Isomap algorithm. *Pattern Recognition Letters*, 27(9):968–979. Cited on page 36.
- Sammon, J. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409. Cited on pages 20, 68, and 77.
- Sanguinetti, G. (2008). Dimensionality reduction of clustered datasets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):535–540. Cited on pages 9 and 28.
- Saul, L., Weinberger, K., Ham, J., Sha, F., and Lee, D. (2006). Spectral methods for dimensionality reduction. In *Semisupervised Learning*, Cambridge, MA. The MIT Press. Cited on page 8.
- Saxena, A., Gupta, A., and Mukerjee, A. (2004). Non-linear dimensionality reduction by locally linear isomaps. *Lecture Notes in Computer Science*, 3316:1038–1043. Cited on page 13.

- Schmid, C. (2001). Constructing models for content-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 39–45. Cited on page 90.
- Schmid, C., Lazebnik, S., and Ponce, J. (2004). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278. Cited on page 103.
- Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319. Cited on pages 13, 24, and 27.
- Schomaker, L., Franke, K., and Bulacu, M. (2007). Using codebooks of fragmented connected-component contours in forensic and historic writer identification. *Pattern Recognition Letters*, 28(6):719–727. Cited on pages 2, 99, and 177.
- Sha, F. and Saul, L. (2005). Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 785–792. Cited on page 9.
- Shastri, L. and Ajjanagadde, V. (1993). From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, 16(3):417–494. Cited on page 78.
- Shawe-Taylor, J. and Christianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK. Cited on pages 13, 14, 20, and 24.
- Shepard, R. (1957). Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22:325–345. Cited on page 68.
- Shepard, R. (1968). Cognitive psychology: A review of the book by U. Neisser. *American Journal of Psychology*, 81:285–289. Cited on page 77.
- Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905. Cited on page 18.
- Shoa, T., Thomas, G., Shafai, C., and Shoa, A. (2004). Extracting a focused image from several out of focus micromechanical structure images. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 505–508. Cited on page 114.
- Simoncelli, E. and Freeman, W. (1995). The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proceedings of the IEEE 2nd International Conference on Image Processing*, pages 444–447. Cited on page 91.
- Song, L., Smola, A., Borgwardt, K., and Gretton, A. (2007). Colored Maximum Variance Unfolding. In *Advances in Neural Information Processing Systems*, volume 21 (in press). Cited on page 40.
- Spearman, C. (1904). General intelligence objectively determined and measured. *American Journal of Psychology*, 15:206–221. Cited on pages 8 and 9.

- Steyvers, M. and Tenenbaum, J. (2005). The large-scale structure of semantic networks: statistical analyses and a model of semantic growth. *Cognitive Science*, 29(1):41–78. Cited on pages 69 and 73.
- Strand, J. and Taxt, T. (1994). Local frequency features for texture classification. *Pattern Recognition*, 27(10). Cited on page 85.
- Sudderth, E., Torralba, A., Freeman, W., and Willsky, A. (2005). Learning hierarchical models of scenes, objects, and parts. In *Proceedings of the 10th International Conference on Computer Vision*, volume 2, pages 1331–1338. Cited on pages 126 and 132.
- Suykens, J. (2007). Data visualization and dimensionality reduction using kernel maps with a reference point. Technical Report 07-22, ESAT-SISTA, K.U. Leuven, Belgium. Cited on page 9.
- Szoplík, T. and Arsenault, H. (1985). Rotation-variant optical data processing using the 2D nonsymmetric Fourier transform. *Applied Optics*, 24(2):168–172. Cited on pages 104 and 165.
- Szummer, M. and Jaakkola, T. (2001). Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems*, volume 14, pages 945–952. Cited on page 53.
- Tanase, M., Veltkamp, R., and Haverkort, H. (2007). Multiple polyline to polygon matching. In *Proceedings 16th Annual Symposium on Algorithms and Computation*, pages 60–70. Cited on page 158.
- Teague, M. (1979). Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8):920–930. Cited on page 159.
- Teh, Y., Jordan, M., Beal, M., and Blei, D. (2004). Hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems*, volume 17, pages 1385–1392, Cambridge, MA. The MIT Press. Cited on page 79.
- Teh, Y. and Roweis, S. (2002). Automatic alignment of hidden representations. In *Advances in Neural Information Processing Systems*, volume 15, pages 841–848, Cambridge, MA. The MIT Press. Cited on pages 21 and 22.
- Tenenbaum, J. (1998). Mapping a manifold of perceptual observations. In *Advances in Neural Information Processing Systems*, volume 10, pages 682–688, Cambridge, MA. The MIT Press. Cited on page 13.
- Tenenbaum, J., de Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323. Cited on pages 8, 12, and 30.
- Teng, L., Li, H., Fu, X., Chen, W., and Shen, I.-F. (2005). Dimension reduction of microarray data based on local tangent space alignment. In *Proceedings of the 4th IEEE International Conference on Cognitive Informatics*, pages 154–159. Cited on pages 20 and 37.
- Tieleman, T. (2008). Training Restricted Boltzmann Machines using approximations to the likelihood gradient. In *Proceedings of the International Conference on Machine Learning*, volume 25, pages 1064–1071. Cited on page 172.

- Tipping, M. (2000). Sparse kernel principal component analysis. In *Advances in Neural Information Processing Systems*, volume 13, pages 633–639, Cambridge, MA. The MIT Press. Cited on pages 14 and 27.
- Tipping, M. and Bishop, C. (1999). Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482. Cited on page 22.
- Torgerson, W. (1952). Multidimensional scaling I: Theory and method. *Psychometrika*, 17:401–419. Cited on pages 11, 20, 40, 54, 68, and 77.
- Torralba, A., Fergus, R., and Freeman, W. (2007). Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Boston, MA. Cited on page 3.
- Tuceryan, M. and Jain, A. (1998). Texture analysis. In Chen, C., Pau, L., and Wang, P., editors, *The Handbook of Pattern Recognition and Computer Vision*, pages 207–248. World Scientific Publishing Co., Singapore. Cited on page 85.
- Turk, M. and Pentland, A. (1991). Face recognition using eigenfaces. In *Proceedings of the Computer Vision and Pattern Recognition 1991*, pages 586–591. Cited on page 12.
- Tuzel, O., Yang, L., Meer, P., and Foran, D. (2007). Classification of hematologic malignancies using texton signatures. *Pattern Analysis and Applications*, 10(4):277–290. Cited on page 94.
- Tversky, A. and Hutchinson, J. (1986). Nearest neighbor analysis of psychological spaces. *Psychological Review*, 93(11):3–22. Cited on pages 69 and 77.
- van der Maaten, L. (2009). A new benchmark dataset for handwritten character recognition. Technical Report TiCC 2009-02, TiCC, Tilburg University, Tilburg, The Netherlands. Cited on page 63.
- van der Maaten, L. and Boon, P. (2006). COIN-O-MATIC: A fast and reliable system for coin classification. In Hanbury, A. and Nölle, M., editors, *Proceedings of the MUSCLE Coin Workshop 2006*, pages 7–17. Cited on page 165.
- van der Maaten, L., Boon, P., Pajmans, J., Lange, A., and Postma, E. (2008). Computer vision and machine learning for archaeology. In Clark, J. and Hagemester, E., editors, *Proceedings of the CAA-2006*, pages 361–367. Cited on page 2.
- van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2431–2456. Cited on page 51.
- van der Maaten, L. and Postma, E. (2005). Improving automatic writer identification. In Verbeecq, K., Tuyls, K., Nowé, A., Manderick, B., and Kuijpers, B., editors, *Proceedings of the 17th Belgian-Dutch Conference on Artificial Intelligence*, pages 260–266. Cited on page 164.
- van der Maaten, L. and Postma, E. (2006). Towards automatic coin classification. In Sablatnig, R., Hemsley, J., Kammerer, P., Zolda, E., and Stockinger, J., editors, *Proceedings of the EVA-Vienna 2006*, pages 19–26. Cited on page 165.
- van der Maaten, L. and Postma, E. (2007). Texton-based texture analysis. In Dastani, M. and de Jong, E., editors, *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence*, pages 213–220. Cited on page 179.

- van der Maaten, L., Postma, E., and van den Herik, H. (2009). Dimensionality reduction: A comparative review. Submitted to *Journal of Machine Learning Research*. Cited on page 57.
- Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Review*, 38(1):49–95. Cited on page 14.
- Varma, M. and Zisserman, A. (2002). Classifying images of materials: Achieving viewpoint and illumination independence. In *Proceedings of the 7th European Conference on Computer Vision*, volume 3, pages 255–271. Cited on pages 94, 98, and 100.
- Varma, M. and Zisserman, A. (2003). Texture classification: Are filter banks necessary? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 691–698. Cited on pages 94, 98, 100, 101, 102, 103, and 113.
- Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81. Cited on pages 89 and 94.
- Varma, M. and Zisserman, A. (2007). A statistical approach to material classification using image patch exemplars. *Preprint*. Cited on page 94.
- Veltkamp, R. and Latecki, L. (2006). Properties and performances of shape similarity measures. In *Data Science and Classification*, pages 47–56. Cited on page 158.
- Venna, J. (2007). *Dimensionality reduction for visual exploration of similarity structures*. PhD thesis, Helsinki University of Technology, Helsinki, Finland. Cited on page 8.
- Venna, J. and Kaski, S. (2006). Visualizing gene interaction graphs with local multidimensional scaling. In *Proceedings of the 14th European Symposium on Artificial Neural Networks*, pages 557–562. Cited on pages 28 and 63.
- Verbeek, J. (2006). Learning nonlinear image manifolds by global alignment of local linear models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1236–1250. Cited on page 9.
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518. Cited on page 2.
- Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269. Cited on page 85.
- Wang, J., Zhang, Z., and Zha, H. (2005). Adaptive manifold learning. In *Advances in Neural Information Processing Systems*, volume 17, pages 1473–1480, Cambridge, MA. The MIT Press. Cited on page 36.
- Weinberger, K., Packer, B., and Saul, L. (2005). Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *Proceedings of the 10th International Workshop on AI and Statistics*, Barbados, WI. Society for Artificial Intelligence and Statistics. Cited on pages 27, 28, and 35.
- Weinberger, K., Sha, F., and Saul, L. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the 21st International Conference on Machine Learning*, pages 839–846. Cited on pages 14 and 57.

- Weinberger, K., Sha, F., Zhu, Q., and Saul, L. (2007). Graph Laplacian regularization for large-scale semidefinite programming. In *Advances in Neural Information Processing Systems*, volume 19. Cited on pages 15, 27, 28, and 58.
- Weiss, Y. (1999). Segmentation using eigenvectors: a unifying view. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 975–982, Los Alamitos, CA. IEEE Computer Society Press. Cited on page 18.
- Weldon, T. and Higgins, W. (1996a). Design of multiple Gabor filters for texture segmentation. In *Proceedings of the International Conference on Acoustic Speech and Signal Processing*, volume 2, pages 243–246. Cited on page 89.
- Weldon, T. and Higgins, W. (1996b). Integrated approach to texture segmentation using multiple Gabor filters. In *Proceedings of the International Conference on Image Processing*, pages 955–958. Cited on page 89.
- Welling, M., Rosen-Zvi, M., and Hinton, G. (2004). Exponential family harmoniums with an application to information retrieval. In *Advances in Neural Information Processing Systems*, volume 17, pages 1481–1488. Cited on pages 21 and 171.
- Wilbraham, H. (1848). On a certain periodic function. *Cambridge and Dublin Mathematical Journal*, 3:198–201. Cited on page 92.
- Williams, C. (2002). On a connection between Kernel PCA and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19. Cited on pages 11, 12, 24, and 66.
- Wolberg, G. and Zokai, S. (2000). Robust image registration using log-polar transform. In *Proceedings of the IEEE International Conference on Image Processing*, pages 493–496. Cited on page 104.
- Wolf, W., Ozer, B., and Lv, T. (2002). Smart cameras as embedded systems. *Computer*, 35(9):48–53. Cited on page 2.
- Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82. Cited on page 127.
- Xiao, L., Sun, J., and Boyd, S. (2006). A duality view of spectral methods for dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1041–1048. Cited on page 25.
- Xie, X. and Mirmehdi, M. (2007). TEXEMS: Texture exemplars for defect detection on random textured surfaces. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(8):1454–1464. Cited on page 94.
- Xu, Q. and Chen, Y. (2006). Multiscale blob features for grayscale, rotation and spatial scale invariant texture classification. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 29–32. Cited on page 85.
- Xu, R., Damelin, S., and Wunsch, D. (2007). Applications of diffusion maps in gene expression data-based cancer diagnosis analysis. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4613–4616. Cited on page 16.

- Yang, L. (2004). Sammon's nonlinear mapping using geodesic distances. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 303–306. Cited on page 34.
- Yuille, A., Hallinan, P., and Cohen, D. (1992). Feature extraction from faces using deformable techniques. *International Journal of Computer Vision*, 8(2):99–111. Cited on page 3.
- Zalesny, A. and van Gool, L. (2001). A compact model for viewpoint dependent texture synthesis. In *Lecture Notes in Computer Science*, volume 2018, pages 124–143. Cited on page 86.
- Zang, J. and Tan, T. (2002). Brief review of invariant texture analysis methods. *Pattern Recognition*, 35(3):735–747. Cited on page 85.
- Zernike, F. (1934). Beugungstheorie des Schneidverfahrens und seiner verbesserten Form, der Phasenkontrastmethode. *Physica*, 1:689–704. Cited on page 159.
- Zhang, D. and Lu, G. (2003). Evaluation of MPEG-7 shape descriptors against other shape descriptors. *Multimedia Systems*, 9(1):15–30. Cited on page 158.
- Zhang, T., Yang, J., Zhao, D., and Ge, X. (2007). Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70:1547–1533. Cited on pages 9 and 20.
- Zhang, Z. and Zha, H. (2003). Local linear smoothing for nonlinear manifold learning. Technical Report CSE-03-003, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA. Cited on page 36.
- Zhang, Z. and Zha, H. (2004). Principal manifolds and nonlinear dimensionality reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, 26(1):313–338. Cited on pages 19 and 20.
- Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919. Cited on page 53.

A Image features

In this appendix, we give an overview of alternative techniques for feature extraction that are not covered in the thesis. In particular, we focus on three types of features: (1) local image features, (2) shape features, and (3) edge-based statistical features. We discuss the three types of image features separately in Section A.1 to A.3.

A.1 Local image features

This section only gives a brief overview of local image features. Local image features aim to model a small region of an image. Local image features are usually employed in object detection tasks, in which first keypoints are obtained using a keypoint detector such as the Harris detector or the SIFT detector. Subsequently, the small image regions around the keypoints are represented using local image features for matching purposes.

In this section, we discuss two local image features: (1) SIFT features and (2) RIFT features, discussed separately in subsection A.1.1 and A.1.2. For a comprehensive review of such features, we refer to [Li and Allinson, 2007].

A.1.1 SIFT features

Scale-Invariant Feature Transform (SIFT) features construct a histogram of the magnitude and orientation of the image gradient in a small image patch, which is typically a small image region around a keypoint identified by a keypoint detector [Lowe, 2004]. The histogram consists of 16 orientation sub-histograms, each of which has 8 bins, leading to a 128-dimensional feature. The construction of the SIFT feature consists of three main steps: (1) the gradient magnitude and orientation at each pixel in the image patch are computed, (2) the gradient magnitudes are weighted using a Gaussian window that is centered onto the image patch, and (3) the weighted gradient magnitudes are then accumulated into orientation histograms measured over subregions of size 4×4 pixels. The construction is illustrated in Figure A.1, in which the length of the arrows correspond to the magnitude of the gradient, and the direction of the arrows to the orientation of the gradient.

A.1.2 RIFT features

Rotation-Invariant Feature Transform (RIFT) features are a generalization of SIFT features that are invariant to rotations of the small image patch that is represented by the features [Lazebnik *et al.*, 2005]. The feature divides the local image patch into concentric rings with equal ring widths. For each of the concentric rings, a gradient orientation histogram is computed in the same way as in the SIFT features. The orientations are measured with respect to the gradient orientation at the center of the image patch. The resulting feature vectors are invariant to rotations of the image

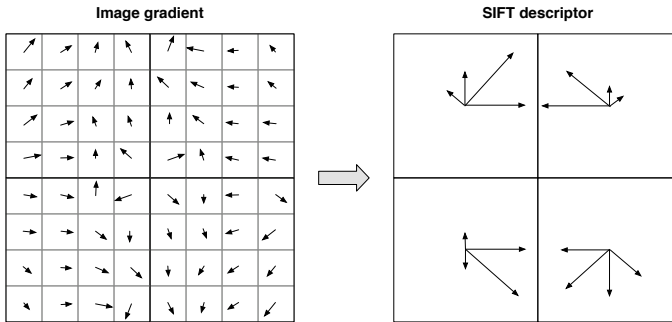


Figure A.1 Illustration of the construction of SIFT features.

patches, however, like the SIFT features, RIFT features are not invariant to flipping of the image patches.

A.2 Shape features

Shape features aim at modeling the outer shape of an object by means of descriptors that are invariant to changes in scale and orientation of the objects. In addition, shape features aim to have a certain degree of invariance to distortions of the shape caused by scale changes, 3D rotations, and non-rigidness of the depicted object. For instance, humans perceive the three shapes in Figure A.2 as depicting similar chickens, despite the non-rigid motions that the depicted chickens exhibit.



Figure A.2 Visual appearance of three chickens under non-rigid motions.

A large number of shape features and shape similarity measures has been presented throughout the years, including techniques based on Zernike moments [Kim *et al.*, 2000], the angular radial transform [Ricard *et al.*, 2005], Hausdorff distances [Huttenlocher *et al.*, 1993], Fourier descriptors [Zhang and Lu, 2003], curvature scale spaces [Mokhtarian *et al.*, 1996], shape contexts [Belongie *et al.*, 2001; Mori *et al.*, 2005], distance sets [Grigorescu and Petkov, 2003], and turning functions [Tanase *et al.*, 2007]. An overview and comparison of a variety of shape features is given by Veltkamp and Latecki [2006]. Roughly, shape features can be subdivided into

two main types: region-based features and contour-based features. Region-based shape features take into account the entire region that is filled by the shape when representing a shape, whereas contour-based shape features represent solely the outer contour of the object. The main advantage of region-based descriptors over contour-based descriptors is that they are more robust to small changes in complex shape contours. This robustness is due to taking into account not only contour pixels but all pixels that constitute the shapes. Figure A.3 shows an example in which region-based shape features would perform better than contour-based shape features, because disconnecting the star from the circle has a great influence on the shape contours, whereas the shape region hardly changes. In contrast, contour-based shape features are better at capturing small details in shape contours.

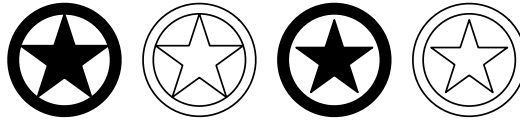


Figure A.3 Example of two perceptually similar shapes with a very different contour.

This section describes four important shape features, two of which are region-based features, and two of which are contour-based features. The two region-based shape features that are discussed are Zernike moments (subsection A.2.1) and the angular radial transform features (subsection A.2.2). The contour-based shape features that are addressed are curvature scale space features (subsection A.2.3) and shape contexts (subsection A.2.4).

A.2.1 Zernike moments

Zernike moments are statistical moments that are computed from the product of a shape image with a collection of Zernike polynomials [Teague, 1979; Kim *et al.*, 2000]. Statistical moments are measurements that provide a characterization for an underlying probability distribution. For instance, the mean and the variance of a distribution correspond to the first and second central moments. Zernike moments measure statistics of the product of the shape image with a collection of so-called Zernike polynomials. Mathematically, the (n, m) -order moment of an image $f(x, y)$ is given by

$$F_n^m = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V_n^m(x) f(x, y) dx dy, \quad m \neq n. \quad (\text{A.1})$$

In the equation, $V_n^m(x)$ is a function that is based on the Zernike polynomial, whereas the function $f(x)$ is given by the shape image. The Zernike polynomial arises in the expansion of the wavefront function of an optical system with circular pupils, and is commonly applied in optics [Zernike, 1934; Bezdikdo, 1974]. The Zernike polynomial R_n^m with order (n, m) (in polar coordinates (ρ, θ)) is given by

$$R_n^m(\rho) = \begin{cases} \sum_{i=0}^{(n-|m|)/2} \frac{(-1)^i (n-i)!}{i! (\frac{1}{2}(n+|m|)-i)! (\frac{1}{2}(n-|m|)-i)!} \rho^{n-2i} & , \text{ if } (n - |m|) \text{ even,} \\ 0 & , \text{ if } (n - |m|) \text{ odd,} \end{cases} \quad (\text{A.2})$$

where n is a positive integer, m is a non-zero positive integer, and $|m| \leq n$, and $0 \leq \rho \leq 1$ (i.e., we assume a unit circle). It can be shown that the set of Zernike polynomials is completely

orthogonal. The (n, m) order of the Zernike polynomial is given by

$$V_n^m(\rho, \theta) = R_n^m(\rho)e^{im\theta}. \quad (\text{A.3})$$

As an example, the $(4, 4)$ -order Zernike polynomial is shown in Figure A.4. The Zernike moment F_n^m of the shape image is defined as

$$F_n^m = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 (V_n^m)^* f(\rho, \theta) d\rho d\theta. \quad (\text{A.4})$$

In the equation, $(V_n^m)^*$ is the complex conjugate of V_n^m , and $f(\rho, \theta)$ is the polar version of the shape image (scaled to the unit disk). Because of the orthogonality of the Zernike basis functions, the moment values are independent. The absolute value of Zernike moments can be proven to be rotation-invariant. In addition, Zernike moments are robust to small variations in the shape. In [Kim *et al.*, 2000], a shape is characterized by all possible Zernike moments up to $n = 10$, which leads to a total of 36 moments. Due to the limited number of Zernike moments that has to be computed and stored, Zernike moments provide an efficient feature representation of shape images.

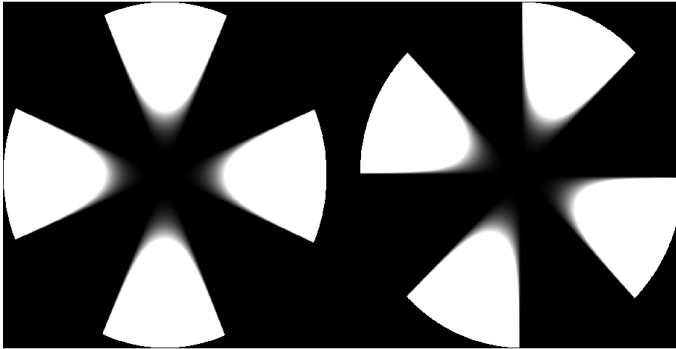


Figure A.4 Real and imaginary part of the $(4, 4)$ order of a Zernike polynomial (in Cartesian coordinates).

A.2.2 Angular radial transform features

The angular radial transform [Kim and Kim, 1999] is similar to the Zernike moments described in the previous subsection in that it computes moments from a collection of basis functions applied on the shape image. The angular radial transform differs from Zernike moments in the basis functions it employs. The angular radial transform employs the basis function $R_n(\rho)$ that is given (in polar coordinates (ρ, θ)) by

$$R_n(\rho) = \begin{cases} 2 \cos(\pi n \rho) & , \text{ if } n \neq 0, \\ 0 & , \text{ if } n = 0. \end{cases} \quad (\text{A.5})$$

Similar to the Zernike basis functions, the set of basis functions is orthogonal, which leads to completely independent moment values. The (n, m) order of the polynomial is then given by

Equation A.3. The corresponding moment is computed using Equation A.4. Similar to Zernike moments, the angular radial transform is rotation-invariant and robust to small shape variations (e.g., due to non-rigidity of the depicted object).

The angular radial transform was selected as the region-based image descriptor in the MPEG-7 standard. In addition, the angular radial transform may be generalized to grayscale images [Ricard *et al.*, 2005], which makes it applicable to other vision tasks than shape matching as well. For instance, a successful application of the generalized angular radial transform features to face detection is presented by Fang and Qiu [2003].

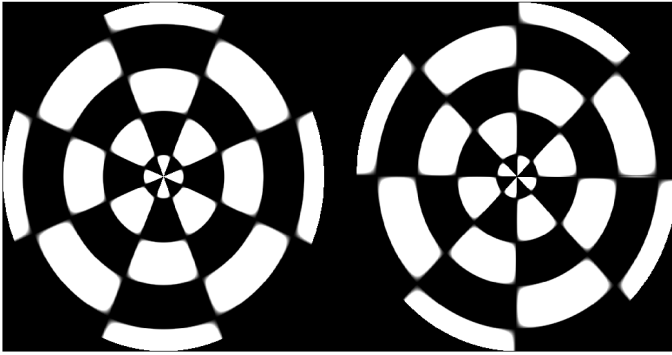


Figure A.5 Real and imaginary part of the (4, 4) order of the angular radial transform (in Cartesian coordinates).

A.2.3 Curvature scale-space features

Curvature scale-space (CSS) features are based on a multi-scale analysis of the curvature of the shape contour [Mokhtarian *et al.*, 1996]. Curvature is a measure for the curviness of a line. The curvature of a circle with radius r is $\frac{1}{r}$, and the curvature of a straight line is 0. The curvature function C of a contour function (x, y) is given by

$$C(x, y) = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}, \quad (\text{A.6})$$

where x' and x'' represent the first and second derivative of x respectively. A zero-crossing of the curvature function of a contour corresponds to an inflection point on the contour. The locations of inflection points are important shape features, because they do not change under affine transformations of the shape contour. CSS features measure the position of the inflection points of the contour through the scale space¹. The extraction of CSS features consists of three main steps that are performed iteratively. First, the contour is convolved with a Gaussian kernel with increasing variance σ in order to obtain the contour representation at a coarser scale. Second, the curvature function of the contour at this scale is computed. Third, the zero-crossings of the curvature function are computed and plotted in an image that depicts the locations of the zero-crossings of the curvature function (i.e., the locations of the inflection points) of the contour as a

¹For an extensive overview on scale space theory, we refer to [Koenderink, 1984].

function of the scale parameter σ . The process is iterated until all inflection points in the shape contour have vanished (due to the blurring of the contour function), and results in a so-called curvature scale space image. An example of such a CSS image is shown in Figure A.6.

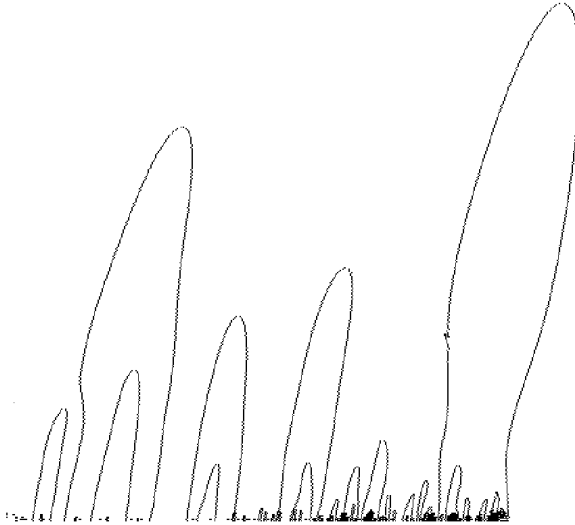


Figure A.6 Example of a CSS image.

The CSS image represents information on all inflection points of the original contour, however, only the main peaks in the CSS image are of interest, because they correspond to the most important inflection points in the contour. The small peaks reflect information that is under large influence of small changes in the shape, and therefore, they are generally considered as noise and thus ignored. The positions of the main peaks are stored and form the final CSS features. A simple matching procedure that aligns two CSS images and sums the Euclidean distances between the main peaks is described by Mokhtarian *et al.* [1996], and allows for the computation of the similarity between shapes.

Curvature scale space features have been selected as the MPEG-7 standard contour descriptor. Successful applications of CSS features have been presented to, e.g., fish classification [Parisi-Baradad *et al.*, 2005] and corner detection [Mokhtarian and Suomela, 1998].

A.2.4 Shape contexts

Shape contexts are shape features that represent a shape by means of a collection of points that are sampled from the shape contour [Belongie *et al.*, 2001]. The sampling of points from the shape contour may be performed using a method that selects points as uniformly as possible over the shape contour, or by means of a method that selects the boundary points in such a way that the number of sampled points is proportional to the curvature of the shape contour. The points that are sampled from the contour are represented by means as shape context descriptors. Shape context descriptors represent a point on the shape boundary by measuring its distance and relative

angle to all other points. The distances and angles to all other points are coarsely quantized² and a joint angle-distance histogram is constructed from the quantized values. An example of a shape context descriptor is shown in Figure A.7.

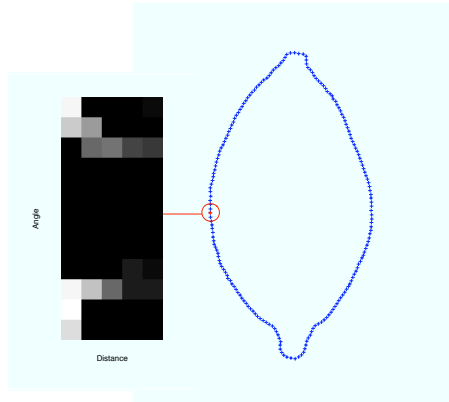


Figure A.7 Example of a shape context descriptor. The shape descriptor for the highlighted point is shown.

A complete set of shape context descriptors (a so-called shape context) contains global information about the shape contour. In order to compute the dissimilarity between two shape contexts, the Hungarian algorithm [Kuhn, 1955] is applied on the pairwise Euclidean distances between the shape context descriptors in the two shape contexts. The Hungarian algorithm finds the optimal assignment between the points sampled from the first shape and the points sampled from the second shape (based on the Euclidean distances between the shape context descriptors) in $O(n^3)$. The costs of the assignment form the dissimilarity between the two shapes. The bending energy of the thin plate spline warping³ that describes the warping between both shapes indicates to what extent the first shape contour has to be warped in order to match the second shape contour [Bookstein, 1989], and may be added to the dissimilarity measure in order to enhance it. The main advantage of shape contexts are its intuitiveness and straightforward implementation. The main disadvantage of the use of shape contexts is the computationally expensive matching that is necessary in order to compute the similarity between two shapes represented by shape contexts. An approach to partially resolve this weakness by performing vector quantization on the shape context descriptors is presented by Mori *et al.* [2005].

A.3 Edge-based statistical features

Edge-based statistical features aim at representing the contours of objects. Edge-based statistical features compute statistics of edge-detected versions of images. The edge detection is generally performed by means of an edge detector, such as the Sobel edge detector or the Harris edge detector [Forsyth and Ponce, 2003]. Contour-based shape features (such as shape contexts and CSS features) may also be considered edge-based statistical features. However, we assume that

²In the quantization of the distance values, a logarithmic scale is usually employed.

³The thin plate spline is a two-dimensional generalization of B-splines [Duchon, 1977].

edge-based statistical features represent not only the outer shape of an object, but attempt to capture all edge information obtained from the edge-detected image. For instance, such features may be used to model the layout of the stamp on a coin or a stroke of handwriting.

In this section, we present two edge-based statistical features, viz. edge-hinge features and edge angle-distance features. In Appendix F, we present the results of experiments in which these features are applied in writer identification and coin classification. Edge-hinge features are discussed in subsection A.3.1. In subsection A.3.2, we present edge angle-distance features.

A.3.1 Edge-hinge features

Edge-hinge features characterize the changes in the direction of a connected line, such as a stroke of handwritten text. This makes them very well applicable to, e.g., writer identification [Bulacu *et al.*, 2003; van der Maaten and Postma, 2005]. Edge-hinge features are extracted from handwriting images by means of a window that is slid over an edge-detected handwriting image. Whenever the central pixel of the window is *on*, the two edge fragments (i.e., connected sequences of pixels) emerging from this central pixel are considered. Their directions are measured and stored as pairs. A joint probability distribution $P(\varphi_1, \varphi_2)$ is estimated from a large sample of such pairs. An example of an angle pair is shown in Figure A.8. The reader should note that edge-hinge features are closely related to the edge orientation histograms that are used as shape features by Jain and Vailaya [1996].

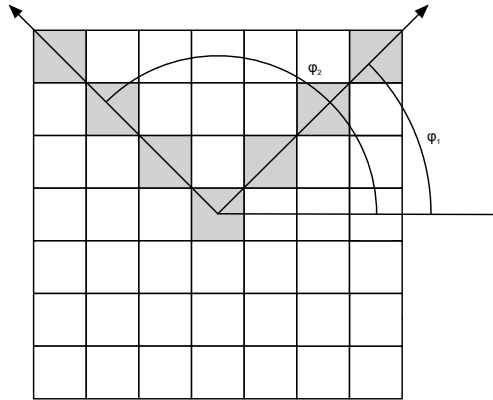


Figure A.8 Angle pair (φ_1, φ_2) .

A.3.2 Edge angle-distance features

Edge angle-distance features attempt to represent edge information on a circular surface in a manner that is invariant to changes in scale and rotation of the surface. This makes edge angle-distance features well suitable for the representation of, e.g., stamps on coins or pills. Below, we introduce edge angle-distance histograms as a combination of edge distance histograms and edge angle histograms.

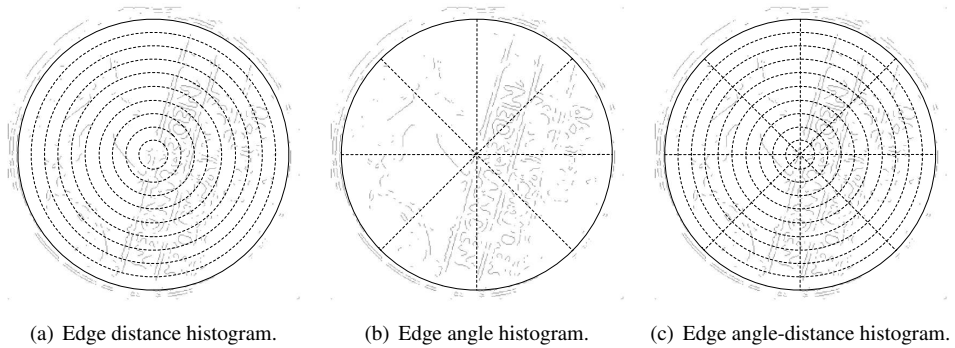


Figure A.9 Edge-based statistical histograms.

Edge distance histograms estimate the distribution of the distances of edge pixels to the center of the surface [van der Maaten and Postma, 2006]. The histograms are computed by dividing the surface into a fixed number of circular concentric parts, as is illustrated in Figure A.9(a). The number of edge pixels in each part is accumulated, and the resulting histograms are normalized. Edge distance histograms are rotation invariant by definition.

Although edge distance histograms were shown to be good features for, e.g., coin classification [van der Maaten and Postma, 2006], they do not incorporate relative angular information in the edge images. The relative angular distribution of edge pixels can be described using edge angle histograms. Edge angle histograms are computed by dividing the circular surface by pie-shaped parts [van der Maaten and Boon, 2006], as is illustrated in Figure A.9(b). The number of edge pixels in the parts is accumulated, and the resulting histogram is normalized. In contrast to edge distance histograms, edge angle histograms are not rotation invariant by definition. Rotation invariance of the edge angle feature can be obtained by computing the magnitude of the Fourier transform of the obtained histogram [Szoplik and Arsenault, 1985]. This step makes the histogram invariant under circular shifts (which correspond to rotations of the surface). Using the magnitude of the Fourier transform requires a large number of bins in the histogram, since a rotation of the surface should imply a circular shift of the histogram instead of a change in the values of the histogram bins.

In order to give a good characterization of the distribution of edge pixels over the edge image, angular and distance information can be combined by the estimation of a joint angle-distance distribution. We refer to the estimations of the joint distributions as edge angle-distance histograms. Edge angle-distance histograms incorporate both distance and relative angular information of the edge pixels in the edge image. The histograms are computed by dividing the edge image into parts as illustrated in Figure A.9(c). The number of edge pixels is binned for each part, and subsequently the resulting histogram is normalized. The feature is made rotation invariant by computing the magnitudes of the Fourier transforms of all distance bands in the distribution. The reader should note that edge angle-distance features have a large resemblance to shape context descriptors.

B Derivation of the t-SNE gradient

In effect, t-SNE minimizes the Kullback-Leibler divergence between the joint probabilities p_{ij} in the high-dimensional space and the joint probabilities q_{ij} in the low-dimensional space. The values of p_{ij} are defined to be the symmetrized conditional probabilities, whereas the values of q_{ij} are obtained by means of a Student-t distribution with one degree of freedom

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \quad (\text{B.1})$$

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}, \quad (\text{B.2})$$

where $p_{j|i}$ and $p_{i|j}$ are either obtained from Equation 3.1 or from the random walk procedure described in Section 3.4. The values of p_{ii} and q_{ii} are set to zero. The Kullback-Leibler divergence between the two joint probability distributions P and Q is given by

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (\text{B.3})$$

$$= \sum_i \sum_j p_{ij} \log p_{ij} - p_{ij} \log q_{ij}. \quad (\text{B.4})$$

In order to make the derivation less cluttered, we define two auxiliary variables d_{ij} and Z as follows

$$d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|, \quad (\text{B.5})$$

$$Z = \sum_{k \neq l} (1 + d_{kl}^2)^{-1}. \quad (\text{B.6})$$

Note that if \mathbf{y}_i changes, the only pairwise distances that change are d_{ij} and d_{ji} for $\forall j$. Hence, the gradient of the cost function C with respect to \mathbf{y}_i is given by

$$\frac{\delta C}{\delta \mathbf{y}_i} = \sum_j \left(\frac{\delta C}{\delta d_{ij}} + \frac{\delta C}{\delta d_{ji}} \right) (\mathbf{y}_i - \mathbf{y}_j) \quad (\text{B.7})$$

$$= 2 \sum_j \frac{\delta C}{\delta d_{ij}} (\mathbf{y}_i - \mathbf{y}_j). \quad (\text{B.8})$$

The gradient $\frac{\delta C}{\delta d_{ij}}$ is computed from the definition of the Kullback-Leibler divergence in Equation B.4 (note that the first part of this equation is a constant).

$$\frac{\delta C}{\delta d_{ij}} = - \sum_{k \neq l} p_{kl} \frac{\delta(\log q_{kl})}{\delta d_{ij}} \quad (\text{B.9})$$

$$= - \sum_{k \neq l} p_{kl} \frac{\delta(\log q_{kl} Z - \log Z)}{\delta d_{ij}} \quad (\text{B.10})$$

$$= - \sum_{k \neq l} p_{kl} \left(\frac{1}{q_{kl} Z} \frac{\delta((1 + d_{kl}^2)^{-1})}{\delta d_{ij}} - \frac{1}{Z} \frac{\delta Z}{\delta d_{ij}} \right) \quad (\text{B.11})$$

The gradient $\frac{\delta((1+d_{kl}^2)^{-1})}{\delta d_{ij}}$ is only nonzero when $k = i$ and $l = j$. Hence, the gradient $\frac{\delta C}{\delta d_{ij}}$ is given by

$$\frac{\delta C}{\delta d_{ij}} = 2 \frac{p_{ij}}{q_{ij} Z} (1 + d_{ij}^2)^{-2} - 2 \sum_{k \neq l} p_{kl} \frac{(1 + d_{ij}^2)^{-2}}{Z}. \quad (\text{B.12})$$

Noting that $\sum_{k \neq l} p_{kl} = 1$, we see that the gradient simplifies to

$$\frac{\delta C}{\delta d_{ij}} = 2p_{ij}(1 + d_{ij}^2)^{-1} - 2q_{ij}(1 + d_{ij}^2)^{-1} \quad (\text{B.13})$$

$$= 2(p_{ij} - q_{ij})(1 + d_{ij}^2)^{-1}. \quad (\text{B.14})$$

Substituting this term into Equation B.8, we obtain the gradient

$$\frac{\delta C}{\delta \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(1 + d_{ij}^2)^{-1} (\mathbf{y}_i - \mathbf{y}_j). \quad (\text{B.15})$$

C Analytical solution to random walk probabilities

In this appendix, we briefly describe the analytical solution to the random walk probabilities that are employed in the random walk version of t-SNE (see Section 3.4). The solution is described in more detail by Grady [2006].

It can be shown that computing the probability that a random walk initiated from a non-landmark point (on a graph that is specified by adjacency matrix \mathbf{W}) first reaches a specific landmark point is equal to computing the solution to the combinatorial Dirichlet problem in which (1) the boundary conditions are at the locations of the landmark points, (2) the considered landmark point is fixed to unity, and (3) the other landmarks points are set to zero [Kakutani, 1945; Doyle and Snell, 1984]. In practice, the solution can thus be obtained by minimizing the combinatorial formulation of the Dirichlet integral

$$D[\mathbf{x}] = \frac{1}{2} \mathbf{x}^T \mathbf{L} \mathbf{x}, \quad (\text{C.1})$$

where \mathbf{L} represents the graph Laplacian. Mathematically, the graph Laplacian is given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} = \text{diag} \left(\sum_j w_{1j}, \sum_j w_{2j}, \dots, \sum_j w_{nj} \right)$. Without loss of generality, we may reorder the landmark points in such a way that the landmark points come first. As a result, the combinatorial Dirichlet integral decomposes into

$$D[\mathbf{x}_N] = \frac{1}{2} \begin{bmatrix} \mathbf{x}_L^T & \mathbf{x}_N^T \end{bmatrix} \begin{bmatrix} \mathbf{L}_L & \mathbf{B} \\ \mathbf{B}^T & \mathbf{L}_N \end{bmatrix} \begin{bmatrix} \mathbf{x}_L \\ \mathbf{x}_N \end{bmatrix} \quad (\text{C.2})$$

$$= \frac{1}{2} \left(\mathbf{x}_L^T \mathbf{L}_L \mathbf{x}_L + 2 \mathbf{x}_N^T \mathbf{B}^T \mathbf{x}_L + \mathbf{x}_N^T \mathbf{L}_N \mathbf{x}_N \right), \quad (\text{C.3})$$

where we use the subscript \cdot_L to indicate the landmark points, and the subscript \cdot_N to indicate the non-landmark points. Differentiating $D[\mathbf{x}_N]$ with respect to \mathbf{x}_N and finding its critical points amounts to solving the linear systems

$$\mathbf{L}_N \mathbf{x}_N = -\mathbf{B}^T. \quad (\text{C.4})$$

Please note that in the equation, \mathbf{B}^T is a matrix containing the columns from the graph Laplacian \mathbf{L} that correspond to the landmark points (excluding the rows that correspond to landmark points). After normalization of the solutions to the systems \mathbf{X}_N , the column vectors of \mathbf{X}_N contain the probability that a random walk initiated from a non-landmark point terminates in a landmark point. One should note that the linear systems specified in Equation C.4 are only nonsingular if the graph is completely connected, or if each connected component in the graph contains at least one landmark point [Biggs, 1974].

Because we are interested in the probability of a random walk initiated from a *landmark point* terminating at another landmark point, we duplicate all landmark points in the neighborhood graph, and initiate the random walks from the duplicate landmarks. Because of memory

constraints, it is not possible to store the entire matrix \mathbf{X}_N into memory (note that we are only interested in a small number of rows from this matrix, viz. in the rows corresponding to the duplicate landmark points). Hence, we solve the linear systems defined by the columns of $-\mathbf{B}^T$ one-by-one, and store only the parts of the solutions that correspond to the duplicate landmark points. For computational reasons, we first perform a Cholesky factorization of \mathbf{L}_N , such that $\mathbf{L}_N = \mathbf{C}\mathbf{C}^T$, where \mathbf{C} is an upper-triangular matrix. Subsequently, the solution to the linear system in Equation C.4 is obtained by solving the linear systems $\mathbf{C}\mathbf{y} = -\mathbf{B}^T$ and $\mathbf{C}\mathbf{x}_N = \mathbf{y}$ using a fast backsubstitution method.

D Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) [Ackley *et al.*, 1985; Hinton, 2002] is an undirected probabilistic graphical model, i.e., a special kind of Markov Random Field. Its structure is a fully connected bipartite graph, in which one group of nodes (the visual nodes \mathbf{v}) models the data, and one group of nodes (the hidden nodes \mathbf{h}) models the latent structure of the data. The nodes in the RBM may follow any exponential family distribution [Welling *et al.*, 2004], but often, they are assumed to be binary stochastic, i.e., to follow a Bernoulli distribution. The structure of an RBM is illustrated in Figure D.1.

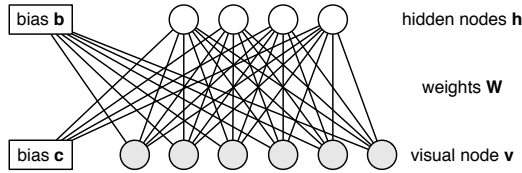


Figure D.1 Schematic layout of a Restricted Boltzmann Machine.

Since an RBM is a special case of a Markov Random Field, the joint distribution over all nodes is given by a Boltzmann distribution that is specified by the energy function $E(\mathbf{v}, \mathbf{h})$. Mathematically, the joint distribution over all nodes is thus given by

$$P(\mathbf{v}, \mathbf{h}) = \exp(-E(\mathbf{v}, \mathbf{h})). \quad (\text{D.1})$$

The most common choice for the energy function is a linear function

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i,j} W_{i,j} v_i h_j - \sum_i b_i v_i - \sum_j c_j h_j, \quad (\text{D.2})$$

in which $W_{i,j}$ represents the weight of the connection between node v_i and h_j , b_i represents the bias on node v_i , and c_j represents the bias on node h_j . Noting that the states of the visual nodes are conditionally independent given the states of the hidden nodes, and the hidden nodes are conditionally independent given the visual nodes, it can easily be shown¹ that this energy function gives rise to conditional distributions $P(v_i = 1 | \mathbf{h})$ and $P(h_j = 1 | \mathbf{v})$ that are given by the sigmoid function of the input into a node

$$P(v_i = 1 | \mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{i,j} h_j - b_i)} = \sigma \left(\sum_j W_{i,j} h_j + b_i \right), \quad (\text{D.3})$$

¹Note that $p(\mathbf{v} | \mathbf{h}) = \frac{p(\mathbf{v}, \mathbf{h})}{\sum_{\mathbf{v}'} p(\mathbf{v}', \mathbf{h})}$, that $p(\mathbf{v}, \mathbf{h}) \propto \exp(\mathbf{h}^T \mathbf{W} \mathbf{v})$ if we omit the biases, and that \mathbf{v} may have the value 0 or 1. As a result, $p(\mathbf{v} = 0 | \mathbf{h}) = \frac{\exp(0)}{\exp(0) + \exp(\mathbf{h}^T \mathbf{W})} = \frac{1}{1 + \exp(\mathbf{h}^T \mathbf{W})}$ and $p(\mathbf{v} = 1 | \mathbf{h}) = \frac{1}{1 + \exp(-\mathbf{h}^T \mathbf{W})}$.

$$P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{i,j}v_i - c_j)} = \sigma\left(\sum_i W_{i,j}v_i - c_j\right). \quad (\text{D.4})$$

Now that we fully defined the RBM, we turn to the problem of learning the weights \mathbf{W} and biases \mathbf{b} and \mathbf{c} such that the marginal distribution over the visual nodes under the model, $P_{model}(\mathbf{v})$, is identical to the observed data distribution $P_{data}(\mathbf{v})$. The RBM is trained as to minimize the natural distance between the data distribution $P_{data}(\mathbf{v})$ and the model distribution $P_{model}(\mathbf{v})$. Mathematically, it is trained to minimize² the Kullback-Leibler divergence $KL(P_{data}||P_{model})$. The gradient of the Kullback-Leibler divergence with respect to the weights $W_{i,j}$ is given by

$$\frac{\delta KL(P_{data}||P_{model})}{\delta W_{i,j}} = \langle v_i h_j \rangle_{P_{data}} - \langle v_i h_j \rangle_{P_{model}}, \quad (\text{D.5})$$

where $\langle \cdot \rangle_{P_{model}}$ represents an expected value under the model distribution, and $\langle \cdot \rangle_{P_{data}}$ represents an expected value under the data distribution.

Although the form of the gradient is fairly simple, it is impossible to actually compute the gradient, in particular, because the term $\langle v_i h_j \rangle_{P_{model}}$ cannot be computed analytically. Sampling from the model distribution, for instance, using Gibbs sampling (note that the required conditionals are given by Equation D.3 and D.4.), is also infeasible because it would require the Markov chain to be run infinitely long. In order to alleviate this problem, an alternative gradient has been proposed that minimizes a slightly different objective function that is called the *contrastive divergence* [Hinton, 2002]. The contrastive divergence measures the tendency of the model distribution to *walk away* from the data distribution by $KL(P_{data}||P_{model}) - KL(P_1||P_{model})$, where $P_1(\mathbf{v})$ represents the distribution over the visual nodes as the RBM is allowed to perform one complete Gibbs sweep away from the data distribution. The contrastive divergence can be minimized efficiently using standard gradient descent techniques, using an approximate gradient that is given by

$$\frac{\delta KL(P_{data}||P_{model}) - KL(P_1||P_{model})}{\delta W_{i,j}} \approx \langle v_i h_j \rangle_{P_{data}} - \langle v_i h_j \rangle_{P_1}. \quad (\text{D.6})$$

The term $\langle v_i h_j \rangle_{P_1}$ is now estimated from samples that are obtained by clamping a data vector onto the visual nodes, and performing one complete Gibbs sweep (CD-1). Alternatively, more than one Gibbs sweep may be employed (CD- n), or we may use a single Markov chain as shown by Tieleman [2008].

²In this case, minimizing the Kullback-Leibler divergence is identical to maximizing the log-likelihood of the data.

E Derivation of the multiple maps t-SNE gradient

Multiple maps t-SNE minimizes the sum of Kullback-Leibler divergences between the pairwise similarities $p_{j|i}$ and the conditional probabilities $q_{j|i}$ in the low-dimensional space. The values of $p_{j|i}$ are given and assumed to obey $\sum_j p_{j|i} = 1$, whereas the values of $q_{j|i}$ are obtained by combining Student-t distributions with one degree of freedom over all maps

$$q_{j|i} = \frac{\sum_m \pi_i^{(m)} \pi_j^{(m)} \left(1 + \|\mathbf{y}_i^{(m)} - \mathbf{y}_j^{(m)}\|^2\right)^{-1}}{\sum_{m'} \sum_{k \neq i} \pi_i^{(m')} \pi_k^{(m')} \left(1 + \|\mathbf{y}_i^{(m')} - \mathbf{y}_k^{(m')}\|^2\right)^{-1}}, \quad (\text{E.1})$$

where we defined the mixing proportions $\pi_i^{(m)}$ by means of mixing weights $w_i^{(m)}$ through

$$\pi_i^{(m)} = \frac{\exp\left(-w_i^{(m)}\right)}{\sum_{m'} \exp\left(-w_i^{(m')}\right)}. \quad (\text{E.2})$$

The values of $p_{i|i}$ and $q_{i|i}$ are set to zero. The sum of Kullback-Leibler divergences between the probability distributions P_i and Q_i is given by

$$C = \sum_i KL(P_i || Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \quad (\text{E.3})$$

$$= \sum_i \sum_j p_{j|i} \log p_{j|i} - p_{j|i} \log q_{j|i}. \quad (\text{E.4})$$

In order to make the derivation less cluttered, we define two auxiliary variables $d_{ij}^{(m)}$ and Z_i as follows

$$d_{ij}^{(m)} = \|\mathbf{y}_i^{(m)} - \mathbf{y}_j^{(m)}\|^2, \quad (\text{E.5})$$

$$Z_i = \sum_m \sum_{k \neq i} \pi_i^{(m)} \pi_j^{(m)} \left(1 + d_{kl}^{(m)}\right)^{-1}. \quad (\text{E.6})$$

Note that if $\mathbf{y}_i^{(m)}$ changes, the only pairwise distances that change are $d_{ij}^{(m)}$ and $d_{ji}^{(m)}$ for $\forall j$. Hence, the gradient of the cost function C with respect to $\mathbf{y}_i^{(m)}$ is given by

$$\frac{\delta C}{\delta \mathbf{y}_i^{(m)}} = 2 \sum_j \left(\frac{\delta C}{\delta d_{ij}^{(m)}} + \frac{\delta C}{\delta d_{ji}^{(m)}} \right) \left(\mathbf{y}_i^{(m)} - \mathbf{y}_j^{(m)} \right). \quad (\text{E.7})$$

The gradient of the cost function C with respect to the pairwise distance $d_{ij}^{(m)}$ is given by

$$\frac{\delta C}{\delta d_{ij}^{(m)}} = \sum_k \sum_l p_{l|k} \frac{\delta(-\log q_{l|k})}{\delta d_{ij}^{(m)}} \quad (\text{E.8})$$

$$= - \sum_k \sum_l p_{l|k} \frac{\delta(\log q_{l|k} Z_k - \log Z_k)}{\delta d_{ij}^{(m)}} \quad (\text{E.9})$$

$$= - \sum_k \sum_l p_{l|k} \left(\frac{1}{q_{l|k} Z_k} \frac{\delta \left(\sum_{m'} \pi_k^{(m')} \pi_l^{(m')} (1 + d_{kl}^{(m')})^{-1} \right)}{\delta d_{ij}^{(m)}} - \frac{1}{Z_k} \frac{\delta Z_k}{\delta d_{ij}^{(m)}} \right) \quad (\text{E.10})$$

$$= \frac{p_{j|i}}{q_{j|i} Z_i} \pi_i^{(m)} \pi_j^{(m)} (1 + d_{ij}^{(m')})^{-2} - \sum_l p_{l|i} \frac{1}{Z_i} \pi_i^{(m)} \pi_j^{(m)} (1 + d_{ij}^{(m)})^{-2} \quad (\text{E.11})$$

$$= \frac{p_{j|i}}{q_{j|i} Z_i} \pi_i^{(m)} \pi_j^{(m)} (1 + d_{ij}^{(m')})^{-2} - \frac{1}{Z_i} \pi_i^{(m)} \pi_j^{(m)} (1 + d_{ij}^{(m)})^{-2} \quad (\text{E.12})$$

$$= \frac{\pi_i^{(m)} \pi_j^{(m)} (1 + d_{ij}^{(m)})^{-2}}{q_{j|i} Z_i} (p_{j|i} - q_{j|i}). \quad (\text{E.13})$$

The gradient of the cost function C with respect to the mixing weights $w_i^{(m)}$ is given by

$$\frac{\delta C}{\delta w_i^{(m)}} = \pi_i^{(m)} \left(\left(\sum_{m'} \pi_i^{(m')} \frac{\delta C}{\delta \pi_i^{(m')}} \right) - \frac{\delta C}{\delta \pi_i^{(m)}} \right). \quad (\text{E.14})$$

The gradient of the cost function C with respect to the mixing proportions $\pi_i^{(m)}$ is given by

$$\frac{\delta C}{\delta \pi_i^{(m)}} = \sum_k \sum_l p_{l|k} \frac{\delta(-\log q_{l|k})}{\delta \pi_i^{(m)}} \quad (\text{E.15})$$

$$= - \sum_k \sum_l p_{l|k} \frac{\delta(\log q_{l|k} Z_k - \log Z_k)}{\delta \pi_i^{(m)}} \quad (\text{E.16})$$

$$= - \sum_k \sum_l p_{l|k} \left(\frac{1}{q_{l|k} Z_k} \frac{\delta \left(\sum_{m'} \pi_k^{(m')} \pi_l^{(m')} (1 + d_{kl}^{(m')})^{-1} \right)}{\delta \pi_i^{(m)}} - \frac{1}{Z_k} \frac{\delta Z_k}{\delta \pi_i^{(m)}} \right) \quad (\text{E.17})$$

$$= \left(- \sum_j \left(\frac{p_{j|i}}{q_{j|i} Z_i} + \frac{p_{i|j}}{q_{i|j} Z_j} \right) \pi_j^{(m)} (1 + d_{ij}^{(m)})^{-1} \right) + \left(\sum_k \sum_l \frac{p_{l|k}}{Z_k} \frac{\delta Z_k}{\delta \pi_i^{(m)}} \right) \quad (\text{E.18})$$

$$= \left(- \sum_j \left(\frac{p_{j|i}}{q_{j|i} Z_i} + \frac{p_{i|j}}{q_{i|j} Z_j} \right) \pi_j^{(m)} (1 + d_{ij}^{(m)})^{-1} \right) + \left(\sum_k \frac{1}{Z_k} \frac{\delta Z_k}{\delta \pi_i^{(m)}} \right) \quad (\text{E.19})$$

$$= - \sum_j \left(\frac{p_{j|i}}{q_{j|i} Z_i} + \frac{p_{i|j}}{q_{i|j} Z_j} \right) \pi_j^{(m)} (1 + d_{ij}^{(m)})^{-1} + \sum_j \left(\frac{1}{Z_i} + \frac{1}{Z_j} \right) \pi_j^{(m)} (1 + d_{ij}^{(m)})^{-1} \quad (\text{E.20})$$

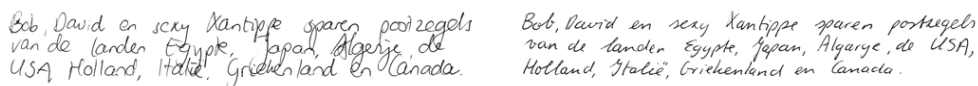
$$= - \sum_j \left(\frac{1}{q_{j|i} Z_i} (p_{j|i} - q_{j|i}) + \frac{1}{q_{i|j} Z_j} (p_{i|j} - q_{i|j}) \right) \pi_j^{(m)} (1 + d_{ij}^{(m)})^{-1}. \quad (\text{E.21})$$

F Applications of edge-based statistical features

In Appendix F, we present two applications of edge-based statistical features on challenging computer vision task. The first application identifies the writer of a piece of text based on his handwriting, and is presented in Section F.1. The second application classifies a coin based on a digital photograph of the coin, and is presented in Section F.2.

F.1 Writer identification

Writer identification is a subfield of forensic handwriting analysis that aims at identifying the writer of a piece of handwritten text. Figure F.1 shows an example of handwritten text written by two different writers. Currently, writer identification is performed by forensic handwriting experts, however, there exists evidence that the judgments of these experts lack reliability [Kam *et al.*, 1997]. The important, sometimes even decisive, role that these judgments play in criminal courts, prompts for a more objective way of handwriting analysis.



Bob, David en scay kanttype sparen postzegels van de landen Egypte, Japan, Algerije, de USA, Holland, Italië, Griekenland en Canada.

Bob, David en scay kanttype sparen postzegels van de landen Egypte, Japan, Algerije, de USA, Holland, Italië, Griekenland en Canada.

Figure F.1 Handwritten text by two different writers.

We performed experiments with edge-hinge features on the Firemaker dataset. The dataset contains the handwritings of 250 writers, who all wrote two pages of text. We measured the generalization performance of 1-nearest neighbor classifiers that are trained on the set of first pages. The generalization performances are measured on the set of second pages, and are reported in Table F.1. The table presents the results for edge-hinge features that were extracted using various fragment lengths (i.e., window sizes), and in addition, using combinations of fragment lengths (leading to multi-scale edge-hinge features).

In order to increase the performance on the identification task, we combined the multiscale edge-hinge features with grapheme features. Graphemes are small strokes of handwriting, and are obtained by segmenting the handwriting. They can be viewed upon as the building blocks of handwriting (i.e., graphemes are to handwriting what textons are to texture). Grapheme features characterize handwriting by means of a grapheme codebook, which is constructed by performing vector quantization (e.g., using k -means clustering, Kohonen maps [Kohonen, 1989], or affinity propagation [Frey and Dueck, 2007]) on a set of graphemes. An example of a grapheme codebook is shown in Figure F.2. In our experiments, combining the multiscale edge-hinge features with (multiscale) grapheme features [Schomaker *et al.*, 2007] increases the generalization performance to 97%. This performance comes up to that of current state-of-the-art writer identification systems [Bulacu and Schomaker, 2007].

<i>Fragment lengths</i>	<i>Generalization perf.</i>	<i>Fragment lengths</i>	<i>Generalization perf.</i>
{3}	68%	{5, 7}	74%
{5}	70%	{5, 9}	77%
{7}	70%	{7, 9}	72%
{9}	69%	{3, 5, 7}	80%
{3, 5}	77%	{3, 7, 9}	78%
{3, 7}	77%	{5, 7, 9}	76%
{3, 9}	79%	{3, 5, 7, 9}	81%

Table F.1 Generalization performances of multiscale edge-hinge features.

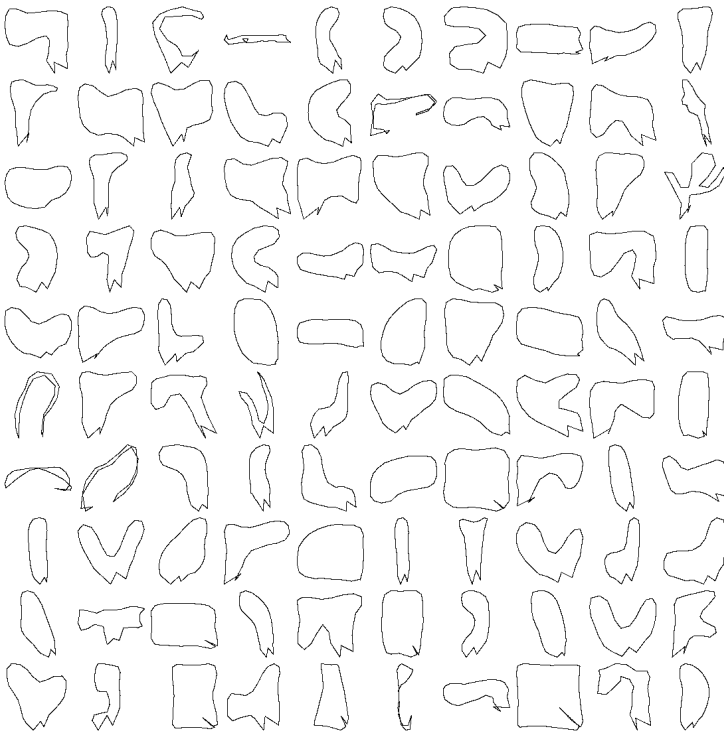


Figure F.2 Grapheme codebook.

F.2 Coin classification

During the introduction of the Euro, various charity organizations collected large numbers of pre-Euro coins in order to raise extra funds for their work. Traditional coin sorting machines [Passerub *et al.*, 1997] are not capable of sorting these coins, due to the large number of coin types and currencies that is present in the obtained coin collection. Image-based coin classification systems make digital photographs of the coins next to traditional thickness and weight measurements

and, thereby, they may alleviate the weaknesses of traditional coin sorting systems. In addition, image-based coin classification systems may be of interest to institutions that are interested in the preservation of the numismatic heritage.

The left part of Figure F.3 shows two examples of photographs made by an image-based coin classification system. Below, we present a system that performs fast and reliable classification of heterogeneous coin collections based on such photographs. The system is described in more detail in [van der Maaten and Postma, 2007]. The workflow of our system consists of three main stages: (1) segmentation, (2) feature extraction, and (3) classification. Segmentation of coins may be performed by means of applying an edge detection with an adaptive threshold and some additional morphological operations or by means of an Hough circle detector. However, it falls outside the scope of this thesis. In the feature extraction stage, we extract edge angle-distance distributions from the segmented coin images. In addition, we extract polar gradient orientation images from the coin images. In the classification stage, we first preselect a number of possible coin classes based on the edge angle-distance distributions. Subsequently, the classification is performed by means of nearest neighbor classifier that was trained on the gradient orientation images. The gradient orientation images are aligned during the nearest-neighbor search, i.e., we apply a standard template matching approach on the gradient orientation images. Since a coin has two coin sides, the classification has to be performed twice. In order to ensure reliability of the system, a coin is solely classified if the classifications of the two coin sides correspond.



Figure F.3 Two coin photographs and two coin prototypes.

We performed experiments on a large dataset of modern coins that were collected during the introduction of the Euro, called the MUSCLE CIS dataset. The dataset is divided into a fixed training set of 20,000 coins, and a fixed test set of 5,000 coins. The training set contains 2,268 different coin faces, corresponding to 692 coin classes. In addition, the training set contains a prototype for each coin face. The prototypes were obtained by registering all coin images belonging to the same coin face, and averaging over the registered coin images. A few examples of coin images and prototypes are shown in Figure F.3. In our experiments with the modern coin dataset, we only use the coin prototypes as training data. In the testset, approximately 400 of the coin classes appear. In addition, the test set contains 3 to 4% coins that are not in the training set, and that should be classified as unknown. In order to evaluate the performance of our system, we performed experiments in which we trained the system on the 2,268 coin prototypes in the trainingset. We evaluated the performance of our approach by measuring the number of correct and incorrect classifications on the testset of 5,000 modern coins.

The results of this experiment are presented in Table F.2. In Table F.2, we present the percentage of correct classifications, the percentage of classifications as unknown, the percentage of incorrect classifications, and the computation time that was consumed for the classification of

Settings		Results			
<i>Preselection</i>	<i>Templ. match.</i>	<i>Correct</i>	<i>Unknown</i>	<i>Incorrect</i>	<i>Comp. time</i>
MEADH	None	53.78%	44.46%	1.76%	1,769 sec.
None	Orientation	92.92%	5.98%	1.10%	17,780 sec.
MEADH	Orientation	88.56%	10.58%	0.86%	10,180 sec.

Table F.2 Performance of our system on the modern coin dataset.

5,000 coins (= 10,000 coin images) on a standard laptop computer. We report the performance of our system both with and without the preselection. In addition, we present the results when the classification is performed based on the edge angle-distance histograms. The results in Table F.2 reveal that our approach is capable of correctly classifying a large percentage of the coins, while only making a low percentage of misclassifications (taking into account that 3 to 4% of the coins in the testset was not in the trainingset). Furthermore, we observe that the preselection based on multiscale edge distance-angle distributions allows for a significant reduction in the computation time, without severely decreasing the generalization performance of the system. In fact, the preselection based on multiscale edge distance-angle distributions even seems to reduce the number of incorrect classifications. Classification based on solely the edge angle-distance histograms already allows for correct classification of over 50% of the coins (note that probability level is $\sim 0.05\%$). The computation time that is needed to process a single coin image is only 1 second on a standard laptop computer (this includes reading, segmentation, feature extraction, and classification of the coin image).

List of Figures

2.1	Taxonomy of dimensionality reduction techniques.	10
2.2	Schematic structure of an autoencoder.	22
2.3	Two low-dimensional data representations.	29
2.4	Four of the artificial datasets.	31
3.1	Gradients of three types of SNE.	46
3.2	Visualizations of 6,000 handwritten digits from the MNIST dataset.	50
3.3	Visualizations of the ORL dataset.	51
3.4	Visualizations of the COIL-20 dataset.	52
3.5	Illustration of the random walk version of t-SNE.	53
3.6	Visualization of 6,000 digits from the MNIST dataset.	55
4.1	Overview of the three-stage training procedure of a parametric t-SNE network.	61
4.2	Visualizations of the MNIST dataset.	65
4.3	Visualizations of the characters dataset.	65
4.4	Illustrations of multiple-maps t-SNE.	71
4.5	Maps of the word association dataset constructed by multiple-maps t-SNE (a-c).	75
4.5	Maps of the word association dataset constructed by multiple-maps t-SNE (d-f).	76
4.6	Generative process of Latent Dirichlet Allocation.	79
5.1	Examples of textures.	84
5.2	Visual appearance of a texture photographed under different lighting conditions.	84
5.3	Graphical model of a Markov Random Field with a grid structure.	87
5.4	Real and imaginary parts of a Gabor filter.	89
5.5	The basis of the Maximum Response (MR) filter banks.	90
5.6	The Schmid filter bank.	91
5.7	A steerable pyramid filter bank with three levels and three orientations ($k = 3$).	92
5.8	Illustration of the Gibbs phenomenon.	92
5.9	Complex wavelet transform filter tree.	93
5.10	Wavelets corresponding to the complex wavelet transform.	93
5.11	Example of a (pixel-based) texton codebook with pixel-based textons.	95
6.1	The 61 texture classes in the CURET texture dataset.	102

6.2	Illustration of the construction of a spin image.	104
6.3	Illustration of the construction of polar Fourier features.	105
6.4	Illustration of the second-order ellipse.	106
6.5	Second-order ellipses drawn onto a grayscale image.	107
6.6	Illustration of affine-covariant image regions.	108
6.7	The 25 texture classes in the UIUCTex texture dataset.	109
6.8	Map of the UIUCTex dataset by t-SNE on affine-invariant texture features.	112
7.1	Example of one of the texton codebooks.	120
7.2	Visualization of the paintings dataset.	121
7.3	Visualization of authentic Van Gogh paintings.	122
7.4	Four examples of seeds.	123
7.5	A seed map constructed by t-SNE based on texton and shape context features.	125
7.6	Backprojection of texton-based texture features.	128
A.1	Illustration of the construction of SIFT features.	158
A.2	Visual appearance of three chickens under non-rigid motions.	158
A.3	Example of two perceptually similar shapes with a very different contour.	159
A.4	Real and imaginary part of the $(4, 4)$ order of a Zernike polynomial.	160
A.5	Real and imaginary part of the $(4, 4)$ order of the angular radial transform.	161
A.6	Example of a CSS image.	162
A.7	Example of a shape context descriptor.	163
A.8	Angle pair (φ_1, φ_2)	164
A.9	Edge-based statistical histograms.	165
D.1	Schematic layout of a Restricted Boltzmann Machine.	171
F.1	Handwritten text by two different writers.	177
F.2	Grapheme codebook.	178
F.3	Two coin photographs and two coin prototypes.	179

List of Tables

2.1	Properties of techniques for dimensionality reduction.	25
2.2	Parameter settings for the experiments.	30
2.3	Generalization errors of 1-NN classifiers trained on artificial datasets.	31
2.4	Trustworthinesses $T(12)$ on the artificial datasets.	31
2.5	Generalization errors of 1-NN classifiers trained on natural datasets.	33
2.6	Trustworthinesses $T(12)$ on the natural datasets.	33
3.1	Cost function parameter settings for the experiments.	49
3.2	Trustworthinesses $T(12)$ of the visualizations of the five datasets.	51
4.1	Generalization errors on the MNIST and characters dataset.	66
4.2	Trustworthiness $T(12)$ of embeddings of the MNIST and characters dataset.	66
6.1	Generalization errors of texon-based texture classifiers.	103
6.2	Generalization errors of invariant texon-based texture classifiers (i).	110
6.3	Generalization errors of invariant texon-based texture classifiers (ii).	110
6.4	Generalization errors of invariant texon-based texture classifiers (iii).	111
6.5	Generalization errors of invariant texon-based texture classifiers (iv).	113
F.1	Generalization performances of multiscale edge-hinge features.	178
F.2	Performance of our system on the modern coin dataset.	180

List of Abbreviations

CD-n	Contrastive Divergence- n
CFA	Coordinated Factor Analysis
CWT	Complex Wavelet Transform
FA	Factor Analysis
GDA	Generalized Discriminant Analysis
HLLE	Hessian Locally Linear Embedding
k-NN	k -Nearest Neighbor
KL	Kullback-Leibler divergence
KPCA	Kernel Principal Components Analysis
LDA	Latent Dirichlet Allocation
LDA	Linear Discriminant Analysis
LEM	Laplacian Eigenmaps
LLC	Locally Linear Coordination
LLE	Locally Linear Embedding
LLTSA	Linear Local Tangent Space Analysis
LPP	Locality Preserving Projection
LSA	Latent Semantic Analysis
LTSA	Local Tangent Space Analysis
MDS	Multidimensional scaling
MR8	Maximum Response-8 filter bank
MRF	Markov Random Field
NCA	Neighborhood Components Analysis
NPE	Neighborhood Preserving Embedding
PCA	Principal Components Analysis
pLSI	Probabilistic Latent Semantic Indexing
pPCA	Probabilistic Principal Components Analysis
RBM	Restricted Boltzmann Machine
RQ	Research question
SDP	Semidefinite program
SNE	Stochastic Neighbor Embedding
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbor Embedding

Summary

The extraction of informative features from visual data is one of the most important problems in the development of computer vision systems. Feature extraction is necessary in order to address the two main problems of image-space representations: (1) the dimensionality problem, i.e., the high dimensionality of image-space representations and (2) the variance problem, i.e., the susceptibility of image-space representations to variations in natural images. The dimensionality problem is due to the large number of pixels that constitute an image. The variance problem is due to the drastic changes individual pixel values may undergo under the presence of variations such as rotations, changes in viewpoint, and scale changes. Feature extraction aims to resolve the two weaknesses of image-space representations by extracting invariant informative features from the visual data. Over the last few decades, a large number of studies have resulted in the development of a variety of features, some of which we aim to improve in this thesis. The problem statement of the thesis reads:

How can we mitigate the dimensionality and variance problems in computer vision systems?

The thesis investigates two types of features that address the two weaknesses of image-space representations: (1) dimensionality reduction features and (2) texture features. Dimensionality reduction features mitigate the dimensionality of image-space representations by building a representation that exploits the (non)linear relations between the values of individual pixels. Texture features are an important example of image features that aim to construct invariant representations for the texture of surfaces. The problem statement of the thesis is translated into the following two research questions.

- **Research question 1:** *How can we improve existing dimensionality reduction features?*
- **Research question 2:** *How can existing texture features be adapted to be invariant to variations that occur in uncontrolled environments, such as rotations, rescalings, and lighting changes?*

We start our research in Chapter 2, in which we focus on the first research question: *How can we improve existing dimensionality reduction features?* The chapter presents an extensive comparative review of state-of-the-art dimensionality reduction features with experiments on a variety of datasets, and identifies the most important weaknesses and limitations of the underlying techniques. In particular, we conclude that existing dimensionality reduction techniques that focus on

retaining the local structure of a data manifold are hampered by flaws in their objective functions that are the result of the convex nature of these functions.

Chapter 3 presents and investigates a new dimensionality reduction technique, called t-Distributed Stochastic Neighbor Embedding (t-SNE), that addresses some of the weaknesses that were identified in Chapter 2. The experiments in Chapter 3 reveal the strong performance of t-SNE in a number of visualization experiments.

In Chapter 4, we present two extensions of t-SNE that aim to extend the technique to two new learning settings in which: (1) a parametric mapping from the high-dimensional to the low-dimensional space is required, and (2) the extracted features need to be non-metric. The chapter presents illustrative experiments for both extensions of t-SNE. We argue that the second extension gives rise to a computational model for semantic representation.

In Chapter 5, we shift our focus to the second research question: *How can existing texture features be adapted to be invariant to variations that occur in uncontrolled environments, such as rotations, rescalings, and lighting changes?* The chapter presents a literature overview of state-of-the-art texture features, which can be subdivided into four main types. The chapter concludes that one of these types, the so-called texton-based texture features, is an interesting type of features that have not yet been investigated in sufficient detail. In particular, two important issues need to be addressed: (1) the performance of image-based textons compared to filter-based textons in terms of performance in classification experiments is unclear and (2) current texton-based texture features are hampered by the second weakness of image-space representations, which is their susceptibility to variations in natural images.

Chapter 6 attempts to resolve the two issues that were raised in Chapter 5. The first issue is addressed by performing a range of experiments in which we compare image-based textons with a variety of filter-based textons. From results of these experiments, we conclude image-based textons may slightly outperform filter-based textons, and offer important computational advantages. The second issue is addressed by the development of three new texton-based texture features, two of which are invariant under rotations, and one of which is invariant under local affine transformations. The chapter presents experiments with the new invariant texture features that reveal the merits of the new texture features.

In Chapter 7, we investigate the new features developed in the previous chapters in two real-world computer vision tasks. First, we investigate the application of a combination of t-SNE and texton-based texture features in the assessment of paintings by Van Gogh and his contemporaries. The results of our experiments reveal that our approach is capable of identifying forged Van Gogh-paintings and paintings by his contemporaries in a collection of Van Gogh and non-Van Gogh paintings. Second, we investigate the application of a combination of t-SNE and texton-based texture features in the recognition of seeds based on photographic reproductions. The results of this study reveal that our approach is promising, in particular, when it is combined with relevant domain knowledge.

Chapter 8 concludes the thesis by answering the two research questions and the problem statement. We conclude the dimensionality problem can successfully be addressed using novel dimensionality reduction techniques such as t-SNE, and that the variance problem may be addressed by identifying affine-covariant image regions and using the Fourier coefficients of polar image representations. In addition to the conclusions, Chapter 8 presents guidelines for future work.

Samenvatting

De extractie van informatieve kenmerken uit visuele data is één van de belangrijkste problemen in de ontwikkeling van automatische beeldverwerkingssystemen. Het extraheren van zulke kenmerken is noodzakelijk om de twee belangrijkste problemen van beeldruimte-representaties aan te pakken: (1) het dimensionaliteitsprobleem: de hoge dimensionaliteit van beeldruimte-representaties en (2) het variantieprobleem: de gevoeligheid van beeldruimte-representaties voor variaties in natuurlijke beelden. De hoge dimensionaliteit van beeldruimte-representaties wordt veroorzaakt door het grote aantal pixels waaruit een afbeelding is opgebouwd. De gevoeligheid voor variaties wordt veroorzaakt door de drastische veranderingen die individuele pixel-waarden kunnen ondergaan als gevolg van variaties in het beeld zoals rotaties, veranderen van gezichtspunt en schaalveranderingen. De laatste tientallen jaren is er veel onderzoek gedaan naar de extractie van kenmerken uit visuele data. Dit onderzoek heeft geleid tot de ontwikkeling van een groot aantal technieken die zulke kenmerken extraheren. Het doel van deze thesis is om sommige van de ontwikkelde technieken voor kenmerk-extractie te verbeteren. De probleemstelling van het proefschrift luidt als volgt:

Hoe kunnen we het dimensionaliteits- en het variantieprobleem van beeldverwerkingssystemen verminderen?

Het proefschrift beschrijft onderzoek naar twee typen kenmerken die de twee nadelen van beeldruimte-representaties aanpakken: (1) dimensiereductie-kenmerken en (2) textuur-kenmerken. Dimensiereductie-kenmerken verlagen de dimensionaliteit van beeldruimte representaties door een representatie te construeren die de (niet-)lineaire relaties tussen de waarden van individuele pixels exploiteert. Textuur-kenmerken zijn een belangrijk voorbeeld van beeld-kenmerken die invariant zijn onder bepaalde transformaties van de beelden. De probleemstelling van dit proefschrift wordt vertaald in de volgende twee onderzoeksvragen:

- **Onderzoeksvraag 1:** *Hoe kunnen we bestaande dimensiereductie-kenmerken verbeteren?*
- **Onderzoeksvraag 2:** *Hoe kunnen bestaande textuur-kenmerken aangepast worden zodat ze invariant zijn onder variaties die voorkomen in ongecontroleerde omgevingen, zoals rotaties, schalingen, en veranderingen in belichting?*

Ons onderzoek begint in hoofdstuk 2, waarin we ons richten op de eerste onderzoeksvraag: *Hoe kunnen we bestaande dimensiereductie-kenmerken verbeteren?* Het hoofdstuk presenteert een uitgebreide vergelijkende review van moderne dimensiereductie-kenmerken met behulp van experimenten op een groot aantal datasets, en identificeert de belangrijkste nadelen en limitaties

van de onderliggende technieken. De belangrijkste conclusie van het hoofdstuk is dat bestaande dimensiereductie-technieken, die zich richten op het behoud van de lokale structuur van data, tegenvallend presteren door fouten in hun kostenfuncties. Deze fouten hangen vaak samen met de convexiteit van de kostenfuncties.

Hoofdstuk 3 presenteert en onderzoekt een nieuwe dimensiereductie-techniek genaamd t-Distributed Stochastic Neighbor Embedding (t-SNE). De nieuwe techniek tracht sommige van de in hoofdstuk 2 geïdentificeerde nadelen van bestaande dimensiereductie-technieken op te lossen. Hoofdstuk 3 toont de uitzonderlijk goede prestaties van t-SNE in een aantal visualisatie-experimenten.

In hoofdstuk 4 presenteren we twee nieuwe varianten van t-SNE die de techniek uitbreiden naar twee nieuwe leeromgevingen, waarin: (1) een parametrische functie van de hoogdimensionale naar de laag-dimensionale ruimte geleerd dient te worden en (2) de geëxtraheerde kenmerken bij voorkeur niet voldoen aan de metrische axioma's. Het hoofdstuk presenteert illustratieve experimenten voor beide nieuwe varianten van t-SNE. We beargumenteren dat de tweede uitbreiding (het niet-metrische model) leidt tot een geschikt computationeel cognitief model voor semantische representatie.

In hoofdstuk 5 richten we ons op de tweede onderzoeksvraag: *Hoe kunnen bestaande textuurkenmerken aangepast worden zodat ze invariant zijn tegen variaties die voorkomen in ongecontroleerde omgevingen, zoals rotaties, schalingen, en veranderingen in belichting?* Het hoofdstuk presenteert een literatuuroverzicht van moderne textuur-kenmerken, die kunnen worden onderverdeeld in vier belangrijke typen. Het hoofdstuk concludeert dat één van deze typen, de zogenaamde texton-gebaseerde textuur-kenmerken, een interessant type kenmerken is dat nog in onvoldoende mate onderzocht is. Twee specifieke problemen met betrekking tot texton-gebaseerde textuur-kenmerken dienen aangepakt te worden: (1) het is onduidelijk hoe beeld-gebaseerde textons presteren in vergelijking met filter-gebaseerde textons in classificatie-experimenten en (2) huidige texton-gebaseerde textuur-kenmerken worden gehinderd door het variantieprobleem van beeldruimte-representaties.

In hoofdstuk 6 trachten we de twee problemen die beschreven zijn in hoofdstuk 5 aan te pakken. Het eerste probleem wordt aangepakt door een groot aantal experimenten uit te voeren, waarin beeld-gebaseerde textons vergeleken worden met verschillende filter-gebaseerde textons. Uit het resultaat van deze experimenten concluderen we dat beeld-gebaseerde textons licht beter presteren dan filter-gebaseerde textons, en bovendien belangrijke computationele voordelen bieden. Het tweede probleem wordt aangepakt door de ontwikkeling van drie nieuwe texton-representaties, waarvan er twee invariant zijn onder rotaties, en één invariant is onder lokale affine transformaties. Het hoofdstuk presenteert experimenten met de nieuwe invariante kenmerken die de voordelen van onze textuur-representaties aantonen.

In hoofdstuk 7 onderzoeken we de prestaties van de kenmerken die we in de vorige hoofdstukken ontwikkeld hebben in twee 'echte' beeldverwerkingstaken. Als eerste onderzoeken we de toepassing van een combinatie van t-SNE en texton-gebaseerde textuur-kenmerken in de evaluatie van schilderijen van Van Gogh en zijn tijdgenoten. De resultaten van dit onderzoek laten zien dat onze aanpak in staat is vervalste Van Gogh-schilderijen en schilderijen van zijn tijdgenoten te onderscheiden van echte Van Gogh-schilderijen. Als tweede onderzoeken we de toepassing van een combinatie van t-SNE en texton-gebaseerde textuur-kenmerken in de automatische herkenning van zaden, gebaseerd op foto's van deze zaden. De resultaten van dit onder-

zoek laten zien dat onze aanpak veelbelovend is, vooral wanneer deze gecombineerd wordt met relevante domeinkennis.

Hoofdstuk 8 besluit de thesis met de antwoorden op de twee onderzoeksvragen, en presenteert onze ideeën over toekomstig onderzoek.

Curriculum Vitae

Laurentius Johannes Paulus van der Maaten was born in Epe, The Netherlands, on the 20th of April 1984. He attended secondary school at the Gymnasium Apeldoorn in Apeldoorn from 1995 to 2001 and obtained the ‘Gymnasium’ diploma. Subsequently, he started his studies in Knowledge Engineering at Maastricht University. In 2003, he took summer classes at Baylor University in Waco, TX, USA. He obtained his M.Sc. degree with a major in Artificial Intelligence from Maastricht University in 2005. His M.Sc. thesis was a study on the automatic identification of writers using biometric features extracted from handwriting.

After obtaining his M.Sc. degree, he accepted a position as a Ph.D. student at Maastricht University under supervision of Eric Postma and Jaap van den Herik. Funded by the Netherlands Organization for Scientific Research (NWO) in the context of the CATCH program (project RICH, grant 640.000.002), he performed research on computer vision and machine learning with applications in the cultural heritage domain in collaboration with the Dutch State Service for Archaeology. The focus of his work was on dimensionality reduction and texture modeling, and on applications such as coin classification and automatic analysis of paintings. In 2008, he worked for six months in Geoffrey Hinton’s machine learning lab at the University of Toronto, performing research on data visualization and models for semantic representation. Also in 2008, he moved to Tilburg University to finish his Ph.D. thesis under supervision of Eric Postma and Jaap van den Herik. In 2009, he accepted a position as a post-doctoral researcher at Delft University of Technology.

In 2007, he won the MUSCLE CIS benchmark competition on coin classification (together with Paul Boon). In 2008, his video (together with Eric Postma) entitled ‘Digital Analysis of Van Gogh Paintings’ won the AAI-08 Most Innovative Video Award. He developed and maintains the Matlab Toolbox for Dimensionality Reduction, which is currently used by thousands of researchers worldwide.

Publications

The investigations performed during my Ph.D. research resulted in the following publications.

- L.J.P. van der Maaten and E.O. Postma. Texton-Based Texture Features with Local Affine Invariance. Submitted to *British Machine Vision Conference*.
- L.J.P. van der Maaten, E.O. Postma and H.J. van den Herik. Dimensionality Reduction: A Comparative Review. Submitted to *Journal of Machine Learning Research*.
- L.J.P. van der Maaten. Preserving Local Structure in Gaussian Process Latent Variable Models. To appear in *Proceedings of Benelearn-09*, 2009.
- L.J.P. van der Maaten. Learning a Parametric Embedding by Preserving Local Structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR W&CP 5:384-391*, 2009.
- L.J.P. van der Maaten. A New Benchmark Dataset for Handwritten Character Recognition. Tilburg University Technical Report, TiCC 2009-02, 2009.
- L.J.P. van der Maaten and E.O. Postma. Identifying the Real Van Gogh with Brushstroke Textons. Tilburg University Technical Report, TiCC 2009-01, 2009.
- L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9(Nov):2431-2456, 2008.
- L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. University of Toronto Technical Report, UTML TR 2008-001, 2008.
- L.J.P. van der Maaten and E.O. Postma. Texton-Based Texture Classification. In Dastani, M. and de Jong, E., editors, *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence*, pages 213-220, 2007.
- L.J.P. van der Maaten. An Introduction to Dimensionality Reduction Using Matlab. Technical Report MICC 07-07. Maastricht University, Maastricht, The Netherlands, 2007.
- S. Vanderlooy, L.J.P. van der Maaten, and I. Sprinkhuizen-Kuyper. Off-line Learning with Transductive Confidence Machines: An Empirical Evaluation. In Carbonell, J.G. and Siekman, J., editors, *Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 310-323, 2007.

- S. Vanderlooy, L.J.P. van der Maaten, and I. Sprinkhuizen-Kuyper. Off-line learning with Transductive Confidence Machines: An Empirical Evaluation. Technical Report MICC-IKAT 07-03. Maastricht University, Maastricht, The Netherlands, 2007.
- L.J.P. van der Maaten and P.J. Boon. COIN-O-MATIC: A Fast and Reliable System for Coin Classification. In Hanbury, A., and Nölle, M., *Proceedings of the MUSCLE Coin Workshop 2006*, pages 7–17, 2006.
- L.J.P. van der Maaten and E.O. Postma. Towards Automatic Coin Classification. In Sablatnig, R., Hemsley, J., Kammerer, P., Zolda, E., and Stockinger, J., *Proceedings of the 1st EVA 2006 Vienna Conference*, pages 19–26, 2006.
- L.J.P. van der Maaten, P.J. Boon, J.J. Paijmans, A.G. Lange, and E.O. Postma. Computer Vision and Machine Learning for Archaeology. In Clark, J.T. and Hagemester, E.M., editors, *Proceedings of Computer Applications and Quantative Methods in Archaeology 2006*, pages 361–367, 2008.
- L.J.P. van der Maaten and E.O. Postma. Improving Automatic Writer Identification. In Verbeeck, K., Tuyls, K., Nowé, A., Manderick, B., and Kuijpers, B., editors, *Proceedings of the 17th Belgian-Dutch Conference on Artificial Intelligence*, pages 260–266, 2005.

SIKS Dissertation Series

1998¹

- 1 Johan van den Akker (CWI) *DEGAS - An Active, Temporal Database of Autonomous Objects*
- 2 Floris Wiesman (UM) *Information Retrieval by Graphically Browsing Meta-Information*
- 3 Ans Steuten (TUD) *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*
- 4 Dennis Breuker (UM) *Memory versus Search in Games*
- 5 Eduard Oskamp (RUL) *Computerondersteuning bij Straftoemeting*

1999

- 1 Mark Sloof (VU) *Physiology of Quality Change Modelling; Automated Modelling of Quality Change of Agricultural Products*
- 2 Rob Potharst (EUR) *Classification using Decision Trees and Neural Nets*
- 3 Don Beal (UM) *The Nature of Minimax Search*
- 4 Jacques Penders (UM) *The Practical Art of Moving Physical Objects*
- 5 Aldo de Moor (KUB) *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*
- 6 Niek Wijngaards (VU) *Re-Design of Compositional Systems*
- 7 David Spelt (UT) *Verification Support for Object Database Design*
- 8 Jacques Lenting (UM) *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation*

2000

- 1 Frank Niessink (VU) *Perspectives on Improving Software Maintenance*
- 2 Koen Holtman (TU/e) *Prototyping of CMS Storage Management*
- 3 Carolien Metselaar (UvA) *Sociaal-organisatorische Gevolgen van Kennistechnologie; een Procesbenadering en Actorperspectief*
- 4 Geert de Haan (VU) *ETAG, A Formal Model of Competence Knowledge for User Interface Design*
- 5 Ruud van der Pol (UM) *Knowledge-Based Query Formulation in Information Retrieval*
- 6 Rogier van Eijk (UU) *Programming Languages for Agent Communication*
- 7 Niels Peek (UU) *Decision-Theoretic Planning of Clinical Patient Management*
- 8 Veerle Coupé (EUR) *Sensitivity Analysis of Decision-Theoretic Networks*

¹Abbreviations: SIKS – Dutch Research School for Information and Knowledge Systems; CWI – Centrum voor Wiskunde en Informatica, Amsterdam; EUR – Erasmus Universiteit, Rotterdam; KUB – Katholieke Universiteit Brabant, Tilburg; KUN – Katholieke Universiteit Nijmegen; RUG – Rijksuniversiteit Groningen; RUL – Rijksuniversiteit Leiden; FONS – Ferrologisch Onderzoeksinstituut Nederland/Sweden; RUN – Radboud Universiteit Nijmegen; TUD – Technische Universiteit Delft; TU/e – Technische Universiteit Eindhoven; UL – Universiteit Leiden; UM – Universiteit Maastricht; UT – Universiteit Twente, Enschede; UU – Universiteit Utrecht; UvA – Universiteit van Amsterdam; UvT – Universiteit van Tilburg; VU – Vrije Universiteit, Amsterdam.

- 9 Florian Waas (CWI) *Principles of Probabilistic Query Optimization*
- 10 Niels Nes (CWI) *Image Database Management System Design Considerations, Algorithms and Architecture*
- 11 Jonas Karlsson (CWI) *Scalable Distributed Data Structures for Database Management*

2001

- 1 Silja Renooij (UU) *Qualitative Approaches to Quantifying Probabilistic Networks*
- 2 Koen Hindriks (UU) *Agent Programming Languages: Programming with Mental Models*
- 3 Maarten van Someren (UvA) *Learning as Problem Solving*
- 4 Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*
- 5 Jacco van Ossenbruggen (VU) *Processing Structured Hypermedia: A Matter of Style*
- 6 Martijn van Welie (VU) *Task-Based User Interface Design*
- 7 Bastiaan Schonhage (VU) *Diva: Architectural Perspectives on Information Visualization*
- 8 Pascal van Eck (VU) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*
- 9 Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*
- 10 Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice BRAHMS: a Multiagent Modeling and Simulation Language for Work Practice Analysis and Design*
- 11 Tom van Engers (VU) *Knowledge Management: The Role of Mental Models in Business Systems Design*

2002

- 1 Nico Lassing (VU) *Architecture-Level Modifiability Analysis*
- 2 Roelof van Zwol (UT) *Modelling and Searching Web-based Document Collections*
- 3 Henk Ernst Blok (UT) *Database Optimization Aspects for Information Retrieval*
- 4 Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining*
- 5 Radu Serban (VU) *The Private Cyberspace Modeling Electronic Environments Inhabited by Privacy-Concerned Agents*
- 6 Laurens Mommers (UL) *Applied Legal Epistemology; Building a Knowledge-based Ontology of the Legal Domain*
- 7 Peter Boncz (CWI) *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*
- 8 Jaap Gordijn (VU) *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*
- 9 Willem-Jan van den Heuvel (KUB) *Integrating Modern Business Applications with Objectified Legacy Systems*
- 10 Brian Sheppard (UM) *Towards Perfect Play of Scrabble*
- 11 Wouter Wijngaards (VU) *Agent Based Modelling of Dynamics: Biological and Organisational Applications*
- 12 Albrecht Schmidt (UvA) *Processing XML in Database Systems*
- 13 Hongjing Wu (TU/e) *A Reference Architecture for Adaptive Hypermedia Applications*
- 14 Wieke de Vries (UU) *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*
- 15 Rik Eshuis (UT) *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*
- 16 Pieter van Langen (VU) *The Anatomy of Design: Foundations, Models and Applications*
- 17 Stefan Manegold (UvA) *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003

- 1 Heiner Stuckenschmidt (VU) *Ontology-Based Information Sharing in Weakly Structured Environments*
- 2 Jan Broersen (VU) *Modal Action Logics for Reasoning About Reactive Systems*
- 3 Martijn Schuemie (TUD) *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*
- 4 Milan Petković (UT) *Content-Based Video Retrieval Supported by Database Technology*
- 5 Jos Lehmann (UvA) *Causation in Artificial Intelligence and Law – A Modelling Approach*
- 6 Boris van Schooten (UT) *Development and Specification of Virtual Environments*
- 7 Machiel Jansen (UvA) *Formal Explorations of Knowledge Intensive Tasks*
- 8 Yong-Ping Ran (UM) *Repair-Based Scheduling*
- 9 Rens Kortmann (UM) *The Resolution of Visually Guided Behaviour*
- 10 Andreas Lincke (UT) *Electronic Business Negotiation: Some Experimental Studies on the Interaction between Medium, Innovation Context and Cult*
- 11 Simon Keizer (UT) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*
- 12 Roeland Ordelman (UT) *Dutch Speech Recognition in Multimedia Information Retrieval*
- 13 Jeroen Donkers (UM) *Nosce Hostem – Searching with Opponent Models*
- 14 Stijn Hoppenbrouwers (KUN) *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*
- 15 Mathijs de Weerd (TUD) *Plan Merging in Multi-Agent Systems*
- 16 Menzo Windhouwer (CWI) *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouse*
- 17 David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*
- 18 Levente Kocsis (UM) *Learning Search Decisions*

2004

- 1 Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*
- 2 Lai Xu (UvT) *Monitoring Multi-party Contracts for E-business*
- 3 Perry Groot (VU) *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*
- 4 Chris van Aart (UvA) *Organizational Principles for Multi-Agent Architectures*
- 5 Viara Popova (EUR) *Knowledge Discovery and Monotonicity*
- 6 Bart-Jan Hommes (TUD) *The Evaluation of Business Process Modeling Techniques*
- 7 Elise Boltjes (UM) *Voorbeeld_{IG} Onderwijs; Voorbeeldgestuurd Onderwijs, een Opstap naar Abstract Denken, vooral voor Meisjes*
- 8 Joop Verbeek (UM) *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale Politie Gegevensuitwisseling en Digitale Expertise*
- 9 Martin Caminada (VU) *For the Sake of the Argument; Explorations into Argument-based Reasoning*
- 10 Suzanne Kabel (UvA) *Knowledge-rich Indexing of Learning-objects*
- 11 Michel Klein (VU) *Change Management for Distributed Ontologies*
- 12 The Duy Bui (UT) *Creating Emotions and Facial Expressions for Embodied Agents*
- 13 Wojciech Jamroga (UT) *Using Multiple Models of Reality: On Agents who Know how to Play*
- 14 Paul Harrenstein (UU) *Logic in Conflict. Logical Explorations in Strategic Equilibrium*
- 15 Arno Knobbe (UU) *Multi-Relational Data Mining*
- 16 Federico Divina (VU) *Hybrid Genetic Relational Search for Inductive Learning*

- 17 Mark Winands (UM) *Informed Search in Complex Games*
- 18 Vania Bessa Machado (UvA) *Supporting the Construction of Qualitative Knowledge Models*
- 19 Thijs Westerveld (UT) *Using generative probabilistic models for multimedia retrieval*
- 20 Madelon Evers (Nyenrode) *Learning from Design: facilitating multidisciplinary design teams*

2005

- 1 Floor Verdenius (UvA) *Methodological Aspects of Designing Induction-Based Applications*
- 2 Erik van der Werf (UM) *AI techniques for the game of Go*
- 3 Franc Grootjen (RUN) *A Pragmatic Approach to the Conceptualisation of Language*
- 4 Nirvana Meratnia (UT) *Towards Database Support for Moving Object data*
- 5 Gabriel Infante-Lopez (UvA) *Two-Level Probabilistic Grammars for Natural Language Parsing*
- 6 Pieter Spronck (UM) *Adaptive Game AI*
- 7 Flavius Frasincaar (TU/e) *Hypermedia Presentation Generation for Semantic Web Information Systems*
- 8 Richard Vdovjak (TU/e) *A Model-driven Approach for Building Distributed Ontology-based Web Applications*
- 9 Jeen Broekstra (VU) *Storage, Querying and Inferencing for Semantic Web Languages*
- 10 Anders Bouwer (UvA) *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*
- 11 Elth Ogston (VU) *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*
- 12 Csaba Boer (EUR) *Distributed Simulation in Industry*
- 13 Fred Hamburg (UL) *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*
- 14 Borys Omelayenko (VU) *Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics*
- 15 Tibor Bosse (VU) *Analysis of the Dynamics of Cognitive Processes*
- 16 Joris Graaumans (UU) *Usability of XML Query Languages*
- 17 Boris Shishkov (TUD) *Software Specification Based on Re-usable Business Components*
- 18 Danielle Sent (UU) *Test-selection strategies for probabilistic networks*
- 19 Michel van Dartel (UM) *Situated Representation*
- 20 Cristina Coteanu (UL) *Cyber Consumer Law, State of the Art and Perspectives*
- 21 Wijnand Derks (UT) *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*

2006

- 1 Samuil Angelov (TU/e) *Foundations of B2B Electronic Contracting*
- 2 Cristina Chisalita (VU) *Contextual issues in the design and use of information technology in organizations*
- 3 Noor Christoph (UvA) *The role of metacognitive skills in learning to solve problems*
- 4 Marta Sabou (VU) *Building Web Service Ontologies*
- 5 Cees Pierik (UU) *Validation Techniques for Object-Oriented Proof Outlines*
- 6 Ziv Baida (VU) *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*
- 7 Marko Smiljanic (UT) *XML schema matching – balancing efficiency and effectiveness by means of clustering*
- 8 Eelco Herder (UT) *Forward, Back and Home Again - Analyzing User Behavior on the Web*
- 9 Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*
- 10 Ronny Siebes (VU) *Semantic Routing in Peer-to-Peer Systems*
- 11 Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*

- 12 Bert Bongers (VU) *Interactivation - Towards an e-cology of people, our technological environment, and the arts*
- 13 Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*
- 14 Johan Hoorn (VU) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*
- 15 Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*
- 16 Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*
- 17 Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*
- 18 Valentin Zhizhkun (UvA) *Graph transformation for Natural Language Processing*
- 19 Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*
- 20 Marina Velikova (UvT) *Monotone models for prediction in data mining*
- 21 Bas van Gils (RUN) *Aptness on the Web*
- 22 Paul de Vrieze (RUN) *Fundamentals of Adaptive Personalisation*
- 23 Ion Juvina (UU) *Development of a Cognitive Model for Navigating on the Web*
- 24 Laura Hollink (VU) *Semantic Annotation for Retrieval of Visual Resources*
- 25 Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*
- 26 Vojkan Mihajlovic (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
- 27 Stefano Bocconi (CWI) *Vox Populi: generating video documentaries from semantically annotated media repositories*
- 28 Borkur Sigurbjornsson (UvA) *Focused Information Access using XML Element Retrieval*

2007

- 1 Kees Leune (UvT) *Access Control and Service-Oriented Architectures*
- 2 Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: A Formal Approach*
- 3 Peter Mika (VU) *Social Networks and the Semantic Web*
- 4 Jurriaan van Diggelen (UU) *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*
- 5 Bart Schermer (UL) *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*
- 6 Gilad Mishne (UvA) *Applied Text Analytics for Blogs*
- 7 Natasa Jovanovic' (UT) *To Whom It May Concern - Addressee Identification in Face-to-Face Meetings*
- 8 Mark Hoogendoorn (VU) *Modeling of Change in Multi-Agent Organizations*
- 9 David Mobach (VU) *Agent-Based Mediated Service Negotiation*
- 10 Huib Aldewereld (UU) *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*
- 11 Natalia Stash (TU/e) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*
- 12 Marcel van Gerven (RUN) *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*
- 13 Rutger Rienks (UT) *Meetings in Smart Environments; Implications of Progressing Technology*
- 14 Niek Bergboer (UM) *Context-Based Image Analysis*
- 15 Joyca Lacroix (UM) *NIM: a Situated Computational Memory Model*
- 16 Davide Grossi (UU) *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*
- 17 Theodore Charitos (UU) *Reasoning with Dynamic Networks in Practice*
- 18 Bart Orriens (UvT) *On the development and management of adaptive business collaborations*

- 19 David Levy (UM) *Intimate relationships with artificial partners*
- 20 Slinger Jansen (UU) *Customer Configuration Updating in a Software Supply Network*
- 21 Karianne Vermaas (UU) *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*
- 22 Zlatko Zlatev (UT) *Goal-oriented design of value and process models from patterns*
- 23 Peter Barna (TU/e) *Specification of Application Logic in Web Information Systems*
- 24 Georgina Ramírez Camps (CWI) *Structural Features in XML Retrieval*
- 25 Joost Schalken (VU) *Empirical Investigations in Software Process Improvement*

2008

- 1 Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*
- 2 Alexei Sharpanskykh (VU) *On Computer-Aided Methods for Modeling and Analysis of Organizations*
- 3 Vera Hollink (UvA) *Optimizing hierarchical menus: a usage-based approach*
- 4 Ander de Keijzer (UT) *Management of Uncertain Data - towards unattended integration*
- 5 Bela Mutschler (UT) *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*
- 6 Arjen Hommersom (RUN) *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*
- 7 Peter van Rosmalen (OU) *Supporting the tutor in the design and support of adaptive e-learning*
- 8 Janneke Bolt (UU) *Bayesian Networks: Aspects of Approximate Inference*
- 9 Christof van Nimwegen (UU) *The paradox of the guided user: assistance can be counter-effective*
- 10 Wauter Bosma (UT) *Discourse oriented Summarization*
- 11 Vera Kartseva (VU) *Designing Controls for Network Organizations: a Value-Based Approach*
- 12 Jozsef Farkas (RUN) *A Semiotically oriented Cognitive Model of Knowledge Representation*
- 13 Caterina Carraciolo (UvA) *Topic Driven Access to Scientific Handbooks*
- 14 Arthur van Bunningen (UT) *Context-Aware Querying; Better Answers with Less Effort*
- 15 Martijn van Otterlo (UT) *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains*
- 16 Henriette van Vugt (VU) *Embodied Agents from a User's Perspective*
- 17 Martin Op't Land (TUD) *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*
- 18 Guido de Croon (UM) *Adaptive Active Vision*
- 19 Henning Rode (UT) *From document to entity retrieval: improving precision and performance of focused text search*
- 20 Rex Arendsen (UvA) *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met een overheid op de administratieve lasten van bedrijven*
- 21 Krisztian Balog (UvA) *People search in the enterprise*
- 22 Henk Koning (UU) *Communication of IT-architecture*
- 23 Stefan Visscher (UU) *Bayesian network models for the management of ventilator-associated pneumonia*
- 24 Zharko Aleksovski (VU) *Using background knowledge in ontology matching*
- 25 Geert Jonker (UU) *Efficient and Equitable exchange in air traffic management plan repair using spender-signed currency*
- 26 Marijn Huijbregts (UT) *Segmentation, diarization and speech transcription: surprise data unraveled*
- 27 Hubert Vogten (OU) *Design and implementation strategies for IMS learning design*
- 28 Ildikó Flesh (RUN) *On the use of independence relations in Bayesian networks*

- 29 Dennis Reidsma (UT) *Annotations and subjective machines - Of annotators, embodied agents, users, and other humans*
- 30 Wouter van Atteveldt (VU) *Semantic network analysis: techniques for extracting, representing and querying media content*
- 31 Loes Braun (UM) *Pro-active medical information retrieval*
- 32 Trung Hui (UT) *Toward affective dialogue management using partially observable Markov decision processes*
- 33 Frank Terpstra (UvA) *Scientific workflow design; theoretical and practical issues*
- 34 Jeroen De Knijf (UU) *Studies in Frequent Tree Mining*
- 35 Benjamin Torben-Nielsen (UvT) *Dendritic morphology: function shapes structure*

2009

- 1 Rasa Jurgenelaite (RUN) *Symmetric Causal Independence Models*
- 2 Willem Robert van Hage (VU) *Evaluating Ontology-Alignment Techniques*
- 3 Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*
- 4 Josephine Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*
- 5 Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*
- 6 Muhammad Subianto (UU) *Understanding Classification*
- 7 Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*
- 8 Volker Nannen (VU) *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*
- 9 Benjamin Kanagwa (RUN) *Design, Discovery and Construction of Service-oriented Systems*
- 10 Jan Wielemaker (UvA) *Logic programming for knowledge-intensive interactive applications*
- 11 Alexander Boer (UvA) *Legal Theory, Sources of Law and the Semantic Web*
- 12 Peter Massuthe (TU/e) *Operating Guidelines for Services*
- 13 Steven de Jong (UM) *Fairness in Multi-Agent Systems*
- 14 Maksym Korotkiy (VU) *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*
- 15 Rinke Hoekstra (UvA) *Ontology Representation - Design Patterns and Ontologies that Make Sense*
- 16 Fritz Reul (UvT) *New Architectures in Computer Chess*
- 17 Laurens van der Maaten (UvT) *Feature Extraction from Visual Data*

TiCC Ph.D. Series

1. Pashiera Barkhuysen

Audiovisual prosody in interaction

Promotor: M.G.J. Swerts, E.J. Krahmer

Tilburg, 3 October 2008

2. Ben Torben-Nielsen

Dendritic morphology: function shapes structure

Promotores: H.J. van den Herik, E.O. Postma

Co-promotor: K.P. Tuyls

Tilburg, 3 December 2008

3. Hans Stol

A framework for evidence-based policy making using IT

Promotor: H.J. van den Herik

Tilburg, 21 January 2009

4. Jeroen Geertzen

Act recognition and prediction. Explorations in computational dialogue modelling

Promotor: H.C. Bunt

Co-promotor: J.M.B. Terken

Tilburg, 11 February 2009

5. Sander Canisius

Structural prediction for natural language processing: a constraint satisfaction approach

Promotores: A.P.J. van den Bosch, W.M.P. Daelemans

Tilburg, 13 February 2009

6. Laurens van der Maaten

Feature Extraction from Visual Data

Promotores: E.O. Postma, H.J. van den Herik

Co-promotor: A.G. Lange

Tilburg, 23 June 2009

Index

- Affine-covariant region, 105
- Angular radial transform, 160
- Archaeobotany, 122
- Autoencoder, 21, 66

- Boltzmann distribution, 86, 171

- Centering, 11, 13, 20
- Centrality, 69, 71
- Classical scaling, 11, 54
- Coin classification, 178
- Complex wavelet transform, 93
- Convexity, 10
- Crowding problem, 44
- CUReT dataset, 101
- Curse of dimensionality, 8, 35
- Curse of intrinsic dimensionality, 57
- Curvature scale space, 161

- Dataset
 - Broken Swiss roll, 30
 - Characters, 63
 - COIL-20, 31, 48
 - CUReT, 101
 - Firemaker, 177
 - FSU word association, 73
 - Helix, 30
 - HIVA, 31
 - MNIST, 31, 48, 63
 - MUSCLE CIS, 179
 - Netflix, 48
 - NiSIS, 31
 - ORL, 31
 - Swiss roll, 30
 - Twin peaks, 30

 - UIUCTex, 107
 - Word-features, 48
- Degree matrix, 18
- Diffusion distance, 15
- Diffusion maps, 15, 56
- Dimensionality problem, 2
- Dimensionality reduction, 3, 9
- Dirichlet integral, 169
- Distance
 - Diffusion, 15
 - Geodesic, 12
- Distributed representation, 78

- Early compression, 47
- Early exaggeration, 47
- Edge angle-distance features, 164
- Edge-hinge features, 164

- Filter bank
 - Gabor, 88
 - Maximum Response, 89
 - Schmid, 90
 - Steerable pyramids, 91
- Filters, 87
- Fourier transform, 91

- Gabor filter, 88
- Gaussian kernel, 15
- Geodesic distance, 12
- Gibbs phenomenon, 92
- Grapheme, 177
- Graylevel co-occurrence features, 85

- Hammersley-Clifford theorem, 86
- Haralick's texture features, 85

- Hessian LLE, 18
- Hilbert transform, 93
- Image features, 4
- Isomap, 12, 56
- Julesz conjecture, 87
- Kernel trick, 13
- Kullback-Leibler divergence, 41
- Laplacian (graph), 18
- Laplacian Eigenmaps, 17
- Latent Dirichlet Allocation, 79
- Latent Semantic Analysis, 77
- LLE, 56
- Local tangent space, 19
- Local Tangent Space Analysis, 19
- Locally Linear Coordination, 21
- Locally Linear Embedding, 16
- Machine learning, 2
- Manifold charting, 23
- Markov Random Field, 21, 85, 171
- Maximum Variance Unfolding, 14
- Metric axioms, 68
- Mixture of Factor Analyzers, 22, 23
- Multiple maps t-SNE, 70
- Nyström approximation, 24, 27
- Out-of-sample extension, 27
- Parametric mapping, 25, 27, 60
- Parametric t-SNE, 60
- Perplexity, 41
- Principal Components Analysis, 11, 54, 66
 - Kernel, 13
- Rayleigh quotient, 10, 23
- Restricted Boltzmann Machine, 21, 60, 171
- RIFT features, 157
- Sammon mapping, 20
- Second-order matrix, 105
- Seeds, 122
- Semantic network, 78
- Semantic space, 77
- Semidefinite program, 26
- Shannon entropy, 42
- Shape contexts, 123, 162
- Shape features, 158
- SIFT features, 157
- Spin image, 103
- Steerability, 91
- Stochastic Neighbor Embedding, 40
 - Multiple Maps t-Distributed, 70
 - Parametric t-Distributed, 60
 - t-Distributed, 43
 - UNI-, 44
- Student t-distribution, 62
- t-Distributed Stochastic Neighbor Embedding, 43
- Texton
 - Affine-invariant, 105
 - Filter-based, 100
 - Image-based, 100
 - Invariant, 103
 - Polar Fourier, 104
 - Spin image, 103
- Texton codebook, 99
- Texton features, 94
- Texton frequency histogram, 99
- Topic model, 79, 126
- Triangle inequality, 68, 71
- Trustworthiness, 28
- UIUCTex dataset, 107
- Van Gogh, 119
- Variance problem, 3
- Wacker forgery, 120
- Wavelet transform
 - Complex, 91
- Writer identification, 177
- Zernike moments, 159