Sparse Grouping and Invariant Representations for Estimation and Recognition

Guoshen Yu

Sparse Grouping and Invariant Representations for Estimation and Recognition

A dissertation presented

by

Guoshen Yu

in fulfillment of the requirements

for the degree of Doctor of Philosophy

in the subject of Applied Mathematics

Ecole Polytechnique

Palaiseau, France

June 30, 2009

Jury

Emmanuel	Bacry	Examiner
Michael	Elad	Reviewer
Henri	Maître	Examiner
Stéphane	Mallat	Advisor
Jean-Michel	Morel	Examiner
Jean	Ponce	Reviewer
Guillermo	Sapiro	Examiner
Jean-Jacques	Slotine	Examiner

Thesis advisor

Stéphane Mallat

Author

Guoshen Yu

Sparse Grouping and Invariant Representations for Estimation and Recognition

Abstract

This thesis develops several contributions for signal and image processing as well as for computer vision. The first part includes a new audio denoising algorithm and a super-resolution algorithm for image zooming. These algorithms are based on some new sparse representations by blocks. A time-frequency block thresholding procedure is introduced for the audio denoising, which enables noise reduction without introducing artifacts, with the results superior to the state-of-the-art. This first part also develops a general approach to solve inverse problems with some piecewise linear sparser representations over the blocks. The application to the image super-resolution allows obtaining a fast algorithm, which clearly improves the PSNR relatively to the existing algorithms.

The second part of the thesis introduces an algorithm (ASIFT) of establishing correspondences between images, which is invariant to affine transforms. It is demonstrated that this algorithm satisfies the invariance constraints and it is able to make correspondences between objects observed under arbitrary angles. Its numeric complexity is of the same order as the most efficient algorithms, with a significantly higher robustness thanks to its affine invariance.

The third part of the thesis introduces a biologically plausible implementation of visual grouping. Inspired by the mechanism of neural synchronization in perceptual grouping, a general algorithm based on neural oscillators is proposed to make visual grouping. The same algorithm is shown to achieve promising results on several classical visual grouping problems, including point clustering, contour integration, and image segmentation.

Contents

Title Page i
Abstract ii
Table of Contents iii
Acknowledgments vi
Dedication viii

1 Introduction **1**
1.1 Sparse Image and Audio Processing with Blocks 3
1.1.1 Block Thresholding Denoising 4
1.1.2 Audio Denoising by Time-Frequency Block Thresholding 6
1.1.3 Image Denoising by Block Pursuit Thresholding 9
1.1.4 Sparse Super-Resolution with Block Pursuits 10
1.2 Affine Invariant Image Comparison 14
1.3 Visual Grouping by Neural Oscillators 19
1.4 Thesis Organization 24

I Sparse Image and Audio Processing with Blocks **25**

2 Audio Denoising by Time-Frequency Block Thresholding **26**
2.1 Introduction 26
2.2 State of the Art 28
2.2.1 Time-frequency Audio Denoising 28
2.2.2 Diagonal Estimation 30
2.2.3 Non-diagonal Estimation 33
2.3 Time-Frequency Block Thresholding 35
2.3.1 Block Thresholding Algorithm 35
2.3.2 Block Thresholding Risk and Choice of λ 37
2.3.3 Adaptive Block Thresholding 39
2.3.4 Non-Diagonal Wiener Post-Processing and Masking Noise 42
2.4 Experiments and Results 45

3 Image Denoising by Block Pursuit Thresholding 51
3.1 Introduction . . . 51
3.2 Wavelet Representations . . . 53
3.3 Block Thresholding . . . 55
3.4 Block Pursuits . . . 57
3.4.1 Block Pursuit Algorithm . . . 57
3.4.2 Fast Implementation . . . 60
3.5 Image Denoising by Block Pursuit Thresholding . . . 63

4 Sparse Super-Resolution by Block Pursuits 68
4.1 Introduction . . . 68
4.2 Directional Interpolation and Sparsity . . . 70
4.2.1 Directional Interpolation . . . 70
4.2.2 Directional Interpolator . . . 73
4.2.3 From Directional Interpolation to Sparsity . . . 75
4.3 Structured Super-Resolution with Block Pursuits . . . 78
4.3.1 Interpolations in Wavelet Domain . . . 78
4.3.2 Structured Sparsity and Directional Interpolation . . . 80
4.3.3 Window Fourier and Wavelet Block Pursuits . . . 81
4.3.4 Directional Regularity . . . 84

II Affine Invariant Image Comparison 87

5 On the Consistency of the SIFT Method 88
5.1 Introduction . . . 89
5.2 Image Operators Formalizing SIFT . . . 93
5.3 The Right Gaussian Blur to Achieve Well-sampling . . . 97
5.4 Scale and SIFT: Consistency of the Method . . . 100

6 ASIFT: A Fully Affine Invariant Image Comparison Method 109
6.1 Introduction . . . 110
6.2 Affine Camera Model and Tilts . . . 114
6.2.1 The Affine Camera Model . . . 115
6.2.2 Transition Tilts . . . 118
6.3 State-of-the-art . . . 119
6.3.1 Scale-Invariant Feature Transform (SIFT) . . . 121
6.3.2 Maximally Stable Extremal Regions (MSER) . . . 122
6.3.3 Harris-Affine and Hessian-Affine . . . 123
6.4 Affine-SIFT (ASIFT) . . . 125
6.4.1 ASIFT Algorithm . . . 125
6.4.2 Latitude and Longitude Sampling . . . 127
6.4.3 Acceleration with Two Resolutions . . . 130
6.4.4 ASIFT Complexity . . . 131
6.5 The Mathematical Justification . . . 133

6.5.1 Inverting Tilts . . . 134
6.5.2 Proof that ASIFT works . . . 136
6.5.3 Algorithmic Sampling Issues . . . 138
6.6 Experiments . . . 138
6.6.1 Standard Test Database . . . 139
6.6.2 Absolute Tilt Tests . . . 141
6.6.3 Transition Tilt Tests . . . 146
6.6.4 Other Test Images . . . 149

III Visual Grouping by Neural Oscillators 154

7 Visual Grouping by Neural Oscillators 155
7.1 Introduction . . . 155
7.2 Model and Algorithm . . . 157
7.2.1 Neural Oscillators . . . 157
7.2.2 Diffusive Connections . . . 158
7.2.3 Concurrent Synchronization and Stability . . . 160
7.2.4 Visual Grouping Algorithm . . . 161
7.2.5 Relation to Previous Work . . . 162
7.3 Point Clustering . . . 163
7.4 Contour Integration . . . 165
7.5 Image Segmentation . . . 169

8 Conclusion 173

Publication Lists 175

A Appendix 177

Bibliography 179

Acknowledgments

I couldn't help being astonished, and sometimes feeling sad, that the three-year doctoral training has gone so rapidly and is now meeting its end.

My deep gratitude first goes to Stéphane Mallat, Professor at Ecole Polytechnique. His guidance has been constantly ample (on average 2 to 3 long discussions per week, plus frequent telephone conversations and emails of pages long) in each phase in preparing this thesis, from conceptual initialization, mathematical analysis, algorithm design to numerical result analysis, and extends to paper writing and talk preparation. Besides his profound scientific vision, his extraordinary research passion and good methodologies in all aspects are things that I have learned from and shall cherish for a life-time. In addition, his kind thoughtfulness has made the whole process extremely enjoyable. It is my great fortune to have enjoyed the opportunity to work with him.

I am equally indebted to Jean-Michel Morel, Professor at ENS Cachan, who has provided me with an extremely precious opportunity to work on a related but different topic in parallel, and has been sacrificing a tremendous amount of time guiding me all the way long. His deep understanding and great experience in image processing and computer vision have opened my eyes, and his kindness and generosity are exceptional. Again I feel that I am extremely lucky to have enjoyed this collaboration.

Of course, my deep gratefulness goes to Jean-Jacques Slotine, Professor at MIT, as well. Not only did he host me for a four-month stay in his lab, which was an extraordinary experience for me, but he constantly sent me brilliant new ideas that have broadened my view.

I am very grateful to Emmanuel Bacry, Research Fellow at CNRS. His great experience on audio processing and extraordinary kindness have made the work fruitful and full of pleasure.

I would like to thank very much the thesis reviewers Michael Elad, Associate Professor

at Technion, and Jean Ponce, Professor at ENS, for their time devoted to carefully reading the manuscript and for the inspiring discussions. The same gratitude goes to the examiners Emmanuel Bacry, Henri Maître, Professor at Telecom ParisTech, Guillermo Sapiro, Professor at University of Minnesota, and Jean-Michel Morel who presided the jury.

Among all my former teachers, I am especially grateful to Yuanyuan Wang, Professor at Fudan University, and Henri Maître. They have led me to this signal and image processing path during my bachelor and master phases, and have been continuously encouraging me during my doctoral study.

I would like to thank Anne Roth and Sandra Schnakenbourg for their proofreading parts of the thesis manuscript (remaining English errors are mine). I am grateful to the kind secretaries in the CMAP laboratory Nasséra Nacer, Wallis Filippi, Nathalie Hurel, Anna Johnsson and Alexandra Noiret, not only for their constant help during the these three years, but also for their great effort in preparing the cocktail after the thesis defense.

Let me take this opportunity to thank Kamel Hamdache and Antonin Chambolle, the two successive laboratory directors, for their support during the preparation of this thesis and for their kindness.

My deepest gratitude goes to my parents Jiageng Yu and Weiping Zhang. They will not read this thesis, but without their love and support I can't imagine how I could have accomplished this journey.

Dedicated to my parents Jiageng Yu and Weiping Zhang.

Chapter 1

Introduction

Image, audio processing and computer vision have myriad applications, from digital cameras, video surveillance, phones, image web searching to medical and satellite imaging analysis. In many applications, signal quality improvement is desired: One wishes to estimate better quality signals, with less noise or higher resolution, from observed ones. In other applications, automatic image recognition is required.

In the last two decades or so, science and technology in image and signal processing have achieved revolutionary progress. Among the fundamental contributions, sparse signal representation and processing have an important position. Wavelets [141, 126] and nonlinear estimation with wavelets [49] opened the door to sparse signal representation and processing. To improve the wavelet representations for images including edges that are geometrically regular, a number of multiscale directional transforms such as curvelet [20, 19] and bandlet [99, 98, 129] dictionaries have been introduced. Given these redundant dictionaries, computationally efficient algorithms including matching pursuit [131] and l^1 pursuit [28] have been developed to calculate good sparse signal approximations. Connections between approximation, sparsity and dictionary coherence properties were investigated and

became more apparent [189, 70]. While sparse signal representation and processing have direct applications in image and signal processing problems such as compression, noise reduction and inverse problems, it deeply influences computer vision as well. Recent pattern recognition algorithms extract salient features in sparse representations [199, 177, 222, 221]. Sparse features are further incorporated with invariance to achieve viewpoint invariant image recognition [119, 89, 21].

An ideal dictionary should provide all types of signals with sparse representations that contain as few as possible coefficients completely decorrelated. Due to the high complexity of images and sounds, however, constructing such a dictionary is impossible. Redundancy is not completely removed from sparse representations for typical images and sounds. For example, geometrical structures such as contours and harmonic lines are presented in image wavelet representations and audio time-frequency representations. Taking advantage of this geometric prior information improves image and signal processing. The first part of this thesis investigates coefficient processing by block in sparse representations. Grouping coefficients in blocks adapted to signal geometry improves non-linear estimations, which leads to better audio and image noise reduction results. Calculating oriented blocks adapted to image geometry amounts to identifying a geometric image model that can be used as prior to solve inverse problems and, in particular, image super-resolution zooming.

In computer vision, sparse features are important for pattern recognition since robust recognition of one object against the others requires a small number of salient features that capture the characteristics of the object. Sparse salient features should at the same time be invariant to variation of pattern observation conditions, for example viewpoint changes in image recognition, so that the recognition is independent to observation conditions. The SIFT method (scale-invariant feature transform) [118, 119] successfully incorporates scale, translation and rotation invariance in sparse features and has achieved

unprecedented success in image recognition applications. While SIFT is fully invariant to image zoom, translation and rotation [154], its robustness to view angle changes is modest. Its performance drops quickly when the view angle change between two images under comparison increases. A number of algorithms have been proposed to improve the view angle invariance [135, 145, 147, 190, 191, 192, 159, 160, 21, 149], however, improvements seem marginal and SIFT continues to be the leader by far. In the second part of the thesis we attack this viewpoint invariance challenge and propose a new fully affine invariant image comparison algorithm Affine-SIFT (ASIFT).

While coefficient grouping allows to improve signal estimation in sparse representations, visual grouping is an important tool in computer vision applications as well. Gestalt psychology [210, 140, 92, 65] formalizes the laws of visual perception and addresses some grouping principles including proximity, good continuation and color constancy. Biological evidences have shown that in the brain the neural synchronization provides a general functional mechanism for perceptual grouping [15, 209]. In computer vision, visual grouping problems have been studied in various frameworks [157, 153, 179, 47, 43, 41, 127], however, comparatively little attention has been devoted to exploiting neural-like mechanism in visual grouping. The last part of the thesis introduces a biologically plausible visual grouping implementation with neural oscillators and shows applications on point clustering, contour integration and image segmentation.

1.1 Sparse Image and Audio Processing with Blocks

In sparse image and audio representations, the presence of geometrical regularity such as contours and harmonic lines motivates treating the representation coefficients by block rather than individually. Based on the work of Cai and Silverman in mathematical statis-

tics [17, 18, 16], Chapter 2 introduces a new non-diagonal audio denoising algorithm through time-frequency block thresholding [219], with rectangular blocks whose sizes are adjusted automatically by the Stein risk estimator [185]. The block thresholding is generalized in Chapter 3 with oriented blocks that adaptively follow geometrical regularity. A block pursuit algorithm is introduced to decompose sparse representation coefficients into blocks selected from a block dictionary. Applications on image denoising are shown. The block pursuit procedure identifies geometric image model and calculates structured sparse representation. An image super-resolution zooming algorithm is derived in Chapter 4 by directional interpolation along the block directions in which the image is directionally regular.

1.1.1 Block Thresholding Denoising

Image and audio signals are often corrupted by noise during the signal acquisition. Signal denoising aims at attenuating the noise while retaining the underlying signals.

Let y be a noisy signal that is the sum of a clean signal f and a noise w of zero mean:

$$y[n] = f[n] + w[n], \;\; n = 0, \ldots, N-1.$$

The noisy signal y is decomposed over a dictionary of vectors $\mathcal{D} = \{g_p\}_{p\in\Gamma}$ that is supposed to be a tight frame [126] with frame bound A:

$$Y[p] = \langle y, g_p \rangle, \quad p \in \Gamma.$$

A denoising algorithm modifies transform coefficients by multiplying each of them by an attenuation factor $a[p]$ to attenuate the noise component. The resulting "denoised" signal estimator is

$$\hat{f}[n] = \frac{1}{A} \sum_{p\in\Gamma} \hat{F}[p]\, g_p[n] = \frac{1}{A} \sum_{p\in\Gamma} a[p]\, Y[p]\, g_p[n]. \tag{1.1}$$

The dictionary $\mathcal{D}$ is said to provide a sparse representation for f if the transform coefficients $F[p] = \langle f, g_p \rangle, \;\; p \in \Gamma$, are mostly zero with a few large coefficients. Signal

denoising is more efficient in sparse representations, in which most coefficients $Y[p]$ contain purely noise and can be filtered without degrading the signal.

Donoho and Johnstone have introduced thresholding operators whose risk is close to the lower bound for Gaussian white noise [49]. For example, a soft thresholding operator attenuates each empirical coefficient $Y[p]$ by a factor

$$a[p] = \left(1 - \frac{\lambda\sigma[p]}{|Y[p]|}\right)_+ \tag{1.2}$$

where $\sigma[p]$ is the standard deviation of the noise coefficient indexed p, λ is the thresholding parameters and $(z)_+ = \max(z, 0)$. Since the attenuation factor $a[p]$ depends only on the empirical coefficient $Y[p]$ and the noise amplitude $\sigma[p]$ of the same index, these operators are called diagonal denoising estimators.

Diagonal denoising estimators are simple and do not take into account the coefficient redundancy in sparse signal representations. As shown in Figures 1.1 and 1.3, geometrical structures such as harmonic lines and contours are presented in audio time-frequency representations and image wavelet representations. Combining this geometric information in denoising estimation helps to improve the denoising performance.

Block thresholding operators are non-diagonal and were introduced by Cai and Silverman [17, 18] in mathematical statistics to improve the asymptotic decay of diagonal thresholding estimators. Block thresholding provides a nice tool to incorporate signal geometric redundancy in thresholding estimation. The coefficients $Y[p]$ are partitioned in I disjoint blocks B_i in which indices are grouped together. A block thresholding estimator multiplies all coefficients within B_i with a same attenuation factor a_i

$$\hat{f} = \frac{1}{A}\sum_{i=1}^{I}\sum_{p\in B_i} a_i\, Y[p]\, g_p, \tag{1.3}$$

where each a_i depends on all coefficients $Y[p]$ for $p \in B_i$. For example soft block thresholding

has the attenuation factors

$$a_i = \left(1 - \frac{\lambda \|\sigma\|^2_{B_i}}{\|Y\|^2_{B_i}}\right)_+, \tag{1.4}$$

where

$$\|Y\|^2_{B_i} = \sum_{p \in B_i} |Y[p]|^2 \quad \text{and} \quad \|\sigma\|^2_{B_i} = \sum_{p \in B_i} |\sigma[p]|^2,$$

and λ is the thresholding parameter. Adjusting the block size can be interpreted as trading-off bias and variance of block thresholding risk. Block size can be adjusted automatically by minimizing a Stein estimator [185] of the block thresholding risk [16].

1.1.2 Audio Denoising by Time-Frequency Block Thresholding

Audio signals are often contaminated by background environment noise and buzzing or humming noise from audio equipments. Time-frequency audio-denoising procedures compute sparse audio time-frequency representations and processes the resulting coefficients to attenuate the noise.

Diagonal denoising estimators such as power subtraction [10, 11, 110], which is a variant of the soft thresholding (1.2), make scalar decision without time-frequency regularization. The resulting attenuated coefficients thus lack of time-frequency regularity. It produces isolated time-frequency coefficients which restore isolated time-frequency structures that are perceived as a musical noise [22, 217].

To reduce musical noise as well as the estimation risk, several authors have proposed *non-diagonal* denoising algorithms that regularize the estimation with time-frequency filtering [54, 55, 35, 137, 77, 121]. Non-diagonal estimators clearly outperform diagonal estimators but depend upon regularization filtering parameters. Large regularization filters reduce the noise energy but introduce more signal distortion [22, 36, 54, 52]. It is desirable that filter parameters are adjusted depending upon the nature of audio signals. In practice, however, they are selected empirically [22, 35, 36, 54, 55]. Moreover, the estimators that

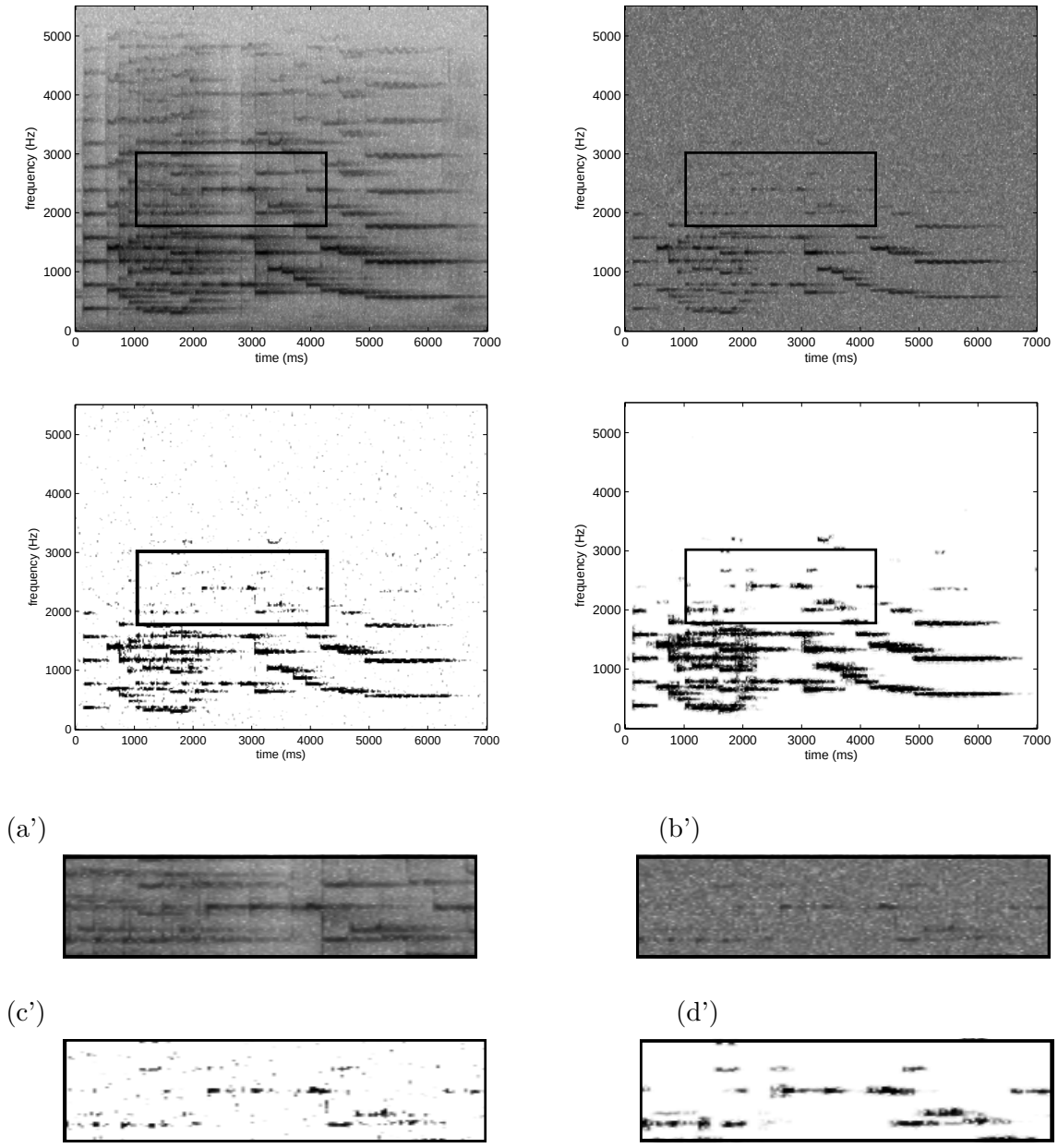


Figure 1.1: (a),(b): Log-spectrograms of the original and noisy "Mozart" signals. (c),(d): attenuation coefficients calculated with a power subtraction and a block thresholding. Black pixels correspond to 1 and white to 0. (a')(b')(c')(d'): zooms over rectangular regions indicated in (a)(b)(c)(d).

are derived with a Bayesian approach [35, 36, 54, 55] model audio signals with Gaussian, Gamma or Laplacian processes. Although such models are often appropriate for speech, they do not take into account the complexity of other audio signals such as music, that include strong attacks.

Chapter 2 introduces a new non-diagonal audio denoising algorithm through adaptive time-frequency block thresholding [219], based on the work of Cai and Silverman in mathematical statistics [17, 18, 16]. For audio time-frequency denoising, we show that block thresholding regularizes the estimate and is thus effective in musical noise reduction. Block parameters are automatically adjusted by minimizing a Stein estimator of the block thresholding risk [185], which is calculated analytically from the noisy signal values with no prior stochastic audio signal model. Numerical experiments show that this new adaptive estimator is robust to signal type variations and improves the SNR and the perceived quality with respect to state-of-the-art audio denoising algorithms.

Figure 1.1 compares the power subtraction and time-frequency block thresholding denoising of a short recording of the Mozart oboe concerto with an additive Gaussian white noise. Figure 1.1(a) and 1.1(b) show respectively the log spectrograms $\log|F[l,k]|$ and $\log|Y[l,k]|$ of the original signal f and its noisy version y, with l and k respectively the time and frequency indexes. Figure 1.1(c) displays a diagonal power subtraction attenuation map $a[l,k]$, with black points corresponding to values close to 1. The zoom in Figure 1.1(c') shows that this attenuation map contains many isolated coefficients close to 1 (black points). These isolated coefficients restore isolated windowed Fourier vectors $g_{l,k}[n]$ that produce a musical noise. Figure 1.1(d) displays a block thresholding attenuation map a_i. The zoom in Figure 1.1(d') shows that non-diagonal block thresholding attenuation factors are much more regular than the diagonal power subtraction attenuation factors in Figure 1.1(c'). Isolated points responsible for musical noise are not kept and signal structure is better

restored.

1.1.3 Image Denoising by Block Pursuit Thresholding

Wavelet representations are sparse for most natural images [126]. As shown in Figure 1.3, wavelet coefficients are quasi-zero where image is uniform and large coefficients are concentrated along the contours. Diagonal denoising estimation is efficient in sparse wavelet representations, however, improvement can be achieved by grouping coefficients in blocks to take better advantage of the geometrical regularity in the representations.

Block thresholding techniques have been investigated [126, 25, 30] to remove noise from images. Similar to audio time-frequency block denoising, image wavelet coefficients are partitioned in square blocks in which a block attenuation rule is applied. As sparse image representations contain geometrical structures more complex than those in audio time-frequency representations that are mainly horizontal and vertical lines, square or rectangular blocks are inadequate to fit image geometry, which increases the risk of the resulting block thresholding estimators. Oriented blocks adapted to image geometry are required.

Chapter 3 introduces a block pursuit thresholding algorithm that generalizes block thresholding method by computing oriented blocks adapted to image geometry. Calculating adaptive oriented blocks amounts to a set covering problem [45] that seeks to cover the image geometry with oriented blocks. A greedy block pursuit procedure is proposed to calculate the oriented blocks selected from a dictionary of blocks. Applications on image denoising are shown.

Geometry in sparse image representations are represented by large coefficients that locally present some oriented geometric structures. To cover these coefficients on the prior structures, a dictionary $\mathcal{D}_B$ of oriented blocks are constructed. As illustrated in Figure 1.2, each block is set of points whose shape may fit some part of the prior structures. The blocks

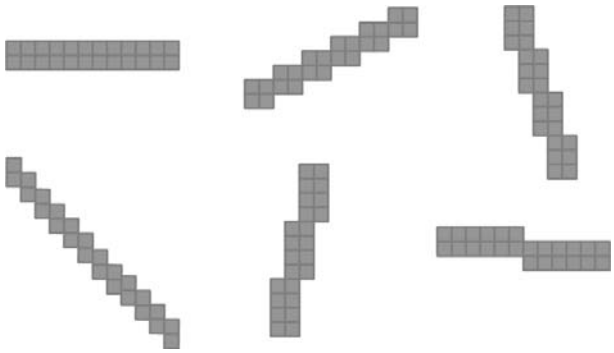


Figure 1.2: Examples of oriented blocks.

are translated to cover the image plane. A greedy block pursuit procedure computes the coefficient energy in all the blocks in $\mathcal{D}_B$ and selects the blocks iteratively one by one in a decreasing order according to the block energy. The block pursuit algorithm has complexity $\mathcal{O}\left((\log_2 K)B^{\#}K\right)$, where K is the size of the block dictionary $\mathcal{D}_B$ and $B^{\#}$ is the block size.

Figure 1.3 compares block pursuit thresholding image denoising with block thresholding and hard thresholding. The block thresholding attenuation is piecewise constant. The attenuation in the blocks that go across the contour degrades the contour and protects the noise. The block pursuit thresholding has oriented block accurately follow the contour and therefore improves the denoising with respect to block thresholding. Compared with hard thresholding that removes some contour coefficients, block pursuit thresholding groups the coefficients along the contour and protects them better, which results to a better contour restoration.

1.1.4 Sparse Super-Resolution with Block Pursuits

The block pursuit procedure allows to identify geometric image model that can be used as prior to solve inverse problems and, in particular, image super-resolution zooming.

Zooming operators that increase the size of images are often needed for digital display

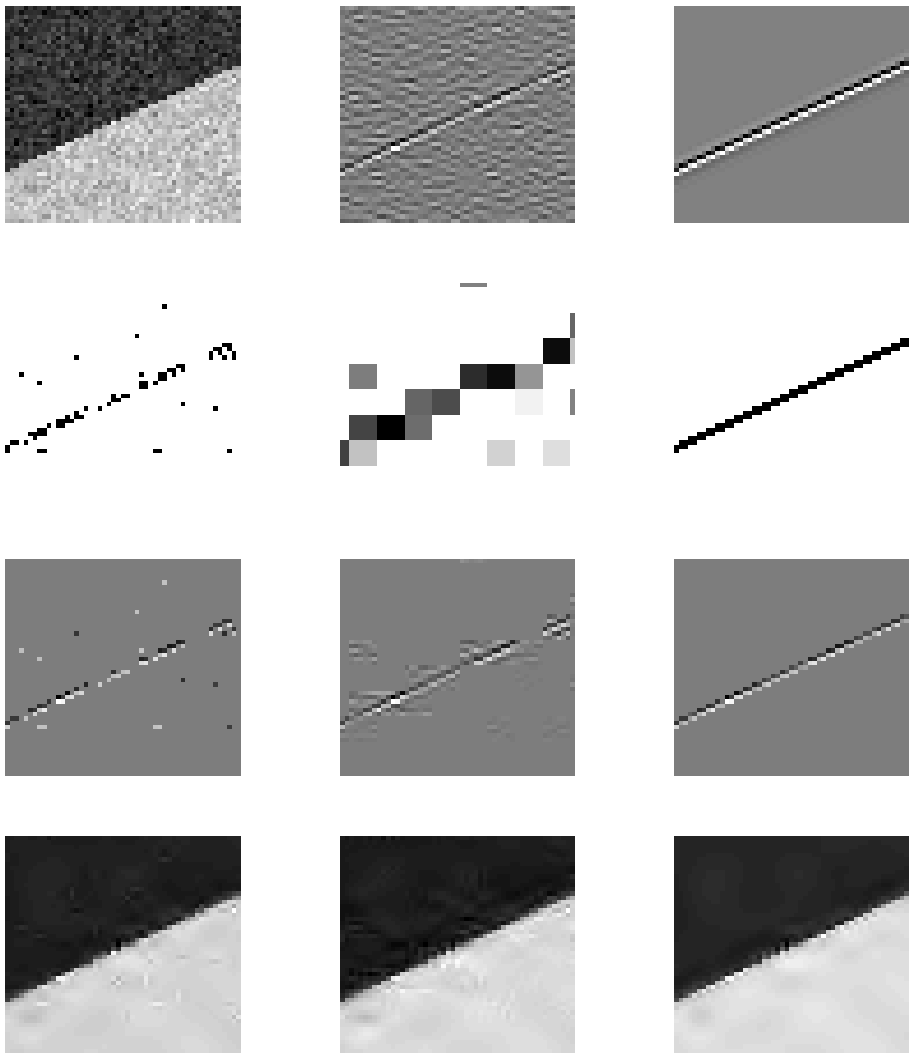


Figure 1.3: From left to right. First row: noisy image y, translation-invariant wavelet coefficients (1st scale, horizontal band) of noisy image and of clean image. Second row: attenuation factors of hard thresholding (HT), block thresholding (BT) and block pursuit thresholding (BPT). Gray-level from white to black: value from 0 to 1. Third row: denoised wavelet coefficients (1st scale, horizontal band) by HT, BT and BPT. Fourth row: denoised image with HT (39.50 dB), BT (39.39 dB) and BPT (40.76 dB).

of images or videos. Linear interpolations [187] are simple and fast, but they introduce artifacts such as zigzag patterns when images are aliased. Super-resolution estimations that take advantage of image prior information are required to restore better images [125, 134]. A large body of super-resolution literature relies on a sequence of low-resolution images or a training process to reconstruct a high-resolution image (see for example [61, 68, 50]). Applications of these methods are restricted when the only relevant data available is a single low-resolution image of interest, or if the memory resource is limited.

Single image super-resolution zooming is more difficult but is possible by interpolating the image along directions for which it is geometrically regular. Directional interpolations, usually known as edge-directed or content-adaptive interpolation, interpolate along directions that are computed with ad-hoc directional regularity estimations [105, 206, 31].

Sparse super-resolution algorithms rely on a sparsity prior. If a signal has a sparse representation in a dictionary then a super-resolution estimation may be computed from lower-resolution measurements [80, 196, 156], and reliable recovery requires that the dictionary is sufficiently *incoherent* [126]. This approach has been used successfully for seismic sparse spike inversion or image inpainting [51, 58, 126]. Sparse prior is related to image geometric regularity. Geometrically regular images have a sparse representation in curvelet [20, 19] or bandlet [99, 98, 129] dictionaries. However, subsampling a curvelet or a bandlet dictionary does not define a sufficiently incoherent dictionary to recover sparse super-resolution estimations for image zooming. Recovering these vectors individually from a subsampled signal requires a full search in a large dictionary which leads to errors. It is necessary to further constrain the sparse representation.

Directional interpolations are much more constraint since locally all pixels are recovered by performing an interpolation with a single direction. Taking advantage of this prior information, Chapter 4 introduces a super-resolution algorithm that computes structured

sparse representation by projecting image wavelet coefficients over vector spaces instead of individual vectors [218, 130]. Selecting the vector spaces amounts to identifying a geometric image model and is calculated with the block pursuit procedures in the wavelet domain. A hierarchical cascade of block pursuit procedures, which factorizes the vector space selection in angle estimation and location assignment, regularizes the geometry estimation and reduces the computational load. Super-resolution estimation is obtained by directionally interpolating the image wavelet coefficients in the vector spaces where there is directionally regularity. This can be interpreted as selecting the linear approximation vectors in the chosen spaces.

Figure 1.4 compares a separable cubic spline interpolation with a super-resolution interpolation computed with the proposed algorithm. The super-resolution achieves a significant PSNR improvement and improves the visual image quality where the image is geometrically regular. High-frequencies are restored along the direction of regularity.

Figure 1.4: Left: High-resolution image. Middle: Cubic spline interpolation 21.37 dB. Right: Super-resolution 23.82 dB

1.2 Affine Invariant Image Comparison

The notion of sparsity is not only applied in signal denoising and super-resolution, but in computer vision as well. In pattern recognition, an object is often sparsely represented by a small set of salient features and recognition is performed by classifying or matching these sparse features. We will focus on an important problem in computer vision — image comparison — for which feature invariance is essential in addition to sparsity and saliency.

Image comparison aims at establishing correspondences between same objects that appear in different images. This is a fundamental step in many computer vision and image processing applications such as image recognition, image retrieval, 3D reconstruction, object tracking, robot localization and image registration [62].

Following the Gestalt psychophysical invariance laws proposed by Wertheimer, Attneave and Kanizsa [210, 3, 91], a good image matching algorithm should satisfy the following requirements [21]:

1. Robust to partial occlusions.

2. Invariant to illumination changes.

3. Insensitive to the noise inherent to any image acquisition device.

4. Independent of the viewpoint changes.

A typical image matching consists in, for each of the concerned image, first detecting points of interest and selecting a region around each point, and then associating each region with a descriptor. Correspondences may thus be established by matching the descriptors [148]. Detectors and descriptors should have some invariance according to the "good matching" requirements above.

The first three "good matching" requirements are relatively easy to satisfy. Local descriptors induce robustness to partial occlusion. Features based on directions of gradients or level sets have certain invariance to illumination changes. Noise can be reduced by prefiltering the images before comparison.

The fourth requirement viewpoint invariance is however much more challenging. Image deformation due to viewpoint changes can be locally approximated by affine transforms, which can be decomposed to rotation, translation, zoom, camera axis latitude and longitude angles, as illustrated in Figure 1.5. The latitude angle θ introduces an important concept tilt $t = 1/\cos\theta$ that is a directional subsampling factor of the frontal image in the direction given by the longitude ϕ. As shown in Figure 1.6, while the *absolute tilt* that is the tilt between a frontal view and a slanted view is relatively small (at maximum about 6), the *transition tilt* that is the tilt from one slanted view to another can be as high as the product of the absolute tilts of the two images ($6^2 = 36$). Since images under comparison are usually both of slanted view, affine invariant image comparison algorithms should be invariant to a very high transition tilt.

State-of-the-art local image detectors can be classified by their incremental invariance properties. All of them are translation invariant. The Harris point detector [85] is also rotation invariant. The Harris-Laplace, Hessian-Laplace and the DoG (Difference-of-Gaussian) region detectors [144, 147, 119, 64] are invariant to rotations and changes of scale. Some moment-based region detectors [112, 6] including the Harris-Affine and Hessian-Affine region detectors [145, 147], an edge-based region detector [190, 192], an intensity-based region detector [191, 192], an entropy-based region detector [90], and two level line-based region detectors MSER ("maximally stable extremal region") [135] and LLD ("level line descriptor") [159, 160, 21] are designed to be invariant to affine transforms.

In his milestone paper [119], Lowe has proposed a scale-invariant feature transform

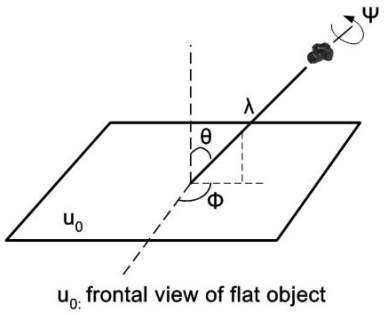


Figure 1.5: Geometric interpretation affine deformation. The camera is looking at u_0 that is a flat physical object. The angles ϕ and θ are respectively the camera optical axis *longitude* and *latitude*. A third angle ψ parameterizes the camera *rotation*, and λ is a *zoom* parameter. The camera can have in addition *translation* in parallel to u_0, which is not shown in the figure.

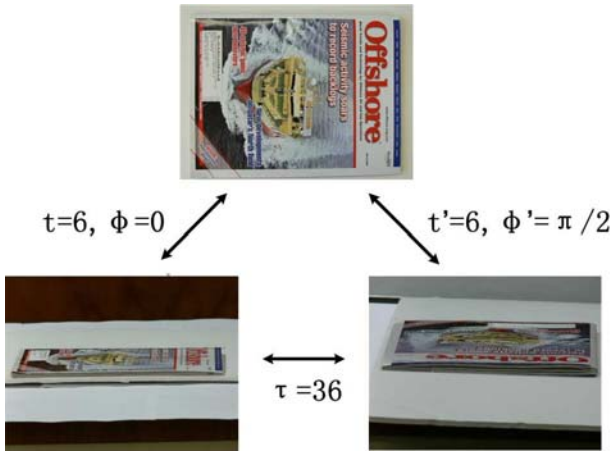


Figure 1.6: The frontal image (above) is squeezed in one direction on the left image by a slanted view, and squeezed in an orthogonal direction by another slanted view. The compression factor or *absolute tilt* is about 6 in each view. The resulting compression factor, or *transition tilt* from left to right is actually 36.

(SIFT) that is invariant to image scaling and rotation and partially invariant to illumination and viewpoint changes. The SIFT method combines the DoG region detector that is fully rotation, translation and scale invariant [154] with a descriptor based on the gradient orientation distribution in the region, which is partially illumination and viewpoint invariant [119]. The SIFT descriptor has been shown to be superior to other many descriptors [146, 148] such as the distribution-based shape context [8], the geometric histogram [2] descriptors, the derivative-based complex filters [6, 173], and the moment invariants [195]. A number of SIFT descriptor variants and extensions, including PCA-SIFT [93], GLOH (gradient location-orientation histogram) [148] and SURF (speeded up robust features) [7] have been developed ever since [67, 103]. They claim more robustness and distinctiveness with scaled-down complexity.

The mentioned state-of-the-art methods have achieved brilliant success. However, none of them is fully affine invariant. MSER, LLD, Harris-Affine and Hessian-Affine attempt to achieve affine invariance by normalizing all the affine parameters. However, as we point out, zoom cannot be normalized without simulating blur because a camera zoom-out consists in essentially blurring the image before subsampling. As a tilt is a zoom operation along one dimension, this applies to tilt as well. In consequence, these normalization methods are not scale invariant and have limited affine invariance. It is verified experimentally that Harris-Affine and Hessian-Affine are robust to transition tilts of maximum value $\tau_{\max} \approx 2.5$ and MSER and LLD $\tau_{\max} \approx 10$ under best conditions. SIFT is actually the only method that is fully scale invariant [154]. However, since it does not treat the camera latitude and longitude angles, its performance drops quickly under substantial viewpoint changes. SIFT is robust to transition tilt smaller than $\tau_{\max} \approx 2$.

Chapter 6 introduce a new affine invariant image comparison framework Affine-SIFT (ASIFT) [155, 220]. Unlike MSER, LLD, Harris-Affine and Hessian-Affine which normalize

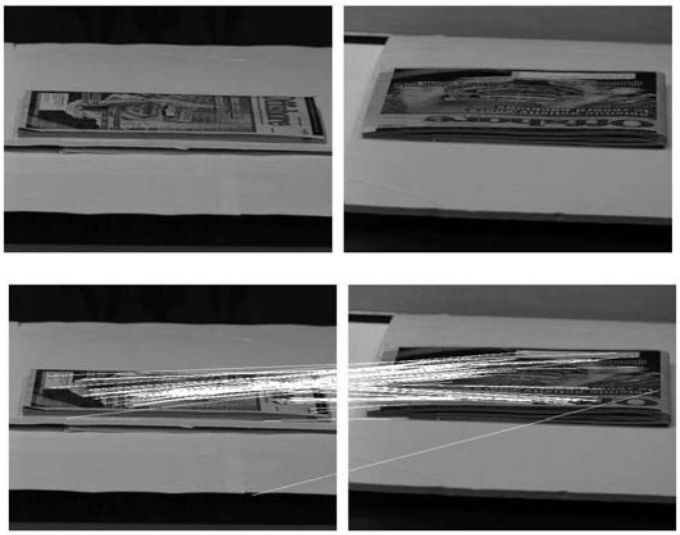


Figure 1.7: Top: Image pair with transition tilt $t \approx 36$. (SIFT, Harris-Affine, Hessian-Affine and MSER fail completely.) Bottom: ASIFT finds 120 matches out which 4 are false.

all the six affine parameters, ASIFT simulates three parameters and normalizes the rest. The scale and the changes of the camera axis orientation are the three simulated parameters. The other three, rotation and translation, are normalized. More specifically, ASIFT simulates the two camera axis angles, and then applies SIFT which simulates the scale and normalizes the rotation and the translation. Mathematically, ASIFT is proved fully affine invariant, up to sampling errors. Against any prognosis, simulating all views depending on the two camera orientation parameters is feasible with no dramatic computational load. A sparse sampling of the simulated parameters is proposed. A coarse-to-fine two-resolution implementation of ASIFT is described, which has about twice the complexity of a single SIFT routine. As shown by the example in Figure 1.7, ASIFT permits to reliably identify features that have undergone transition tilts of large magnitude, up to 36 and higher, and outperforms significantly the state-of-the-art, including SIFT, MSER, Harris-Affine and Harris-Affine.

1.3 Visual Grouping by Neural Oscillators

While coefficient grouping helps to improve signal estimation, visual grouping is an important tool for image recognition as well. Consider Figure 1.8. Why do we perceive in these visual stimuli a cluster of points, a straight contour and a hurricane? How is the identification achieved between a subgroup of stimuli and the perceived objects? These classical questions can be addressed from various points of view, physiological, mathematical and biological.

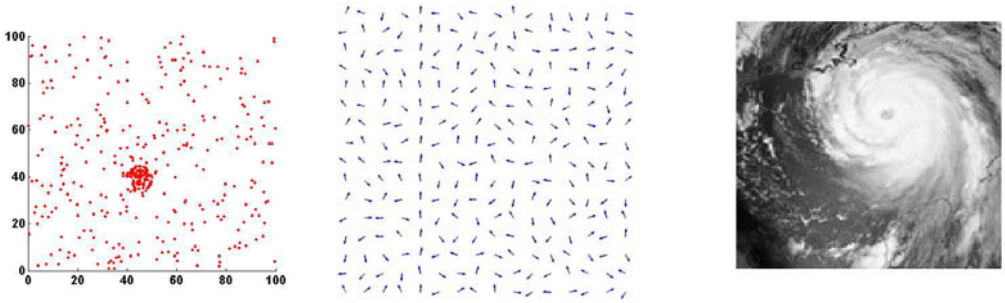


Figure 1.8: Left: a cloud of points in which a dense cluster is embedded. Middle: a random direction grid in which a vertical contour is embedded. Right: an image in which a hurricane is embedded.

Many physiological studies, e.g. [65, 86, 101, 216], have shown evidence of grouping in visual cortex. Gestalt psychology [210, 140, 92, 48] formalizes the laws of visual perception and addresses some grouping principles such as proximity, good continuation and color constancy, in order to describe the construction of larger groups from atomic local information in the stimuli.

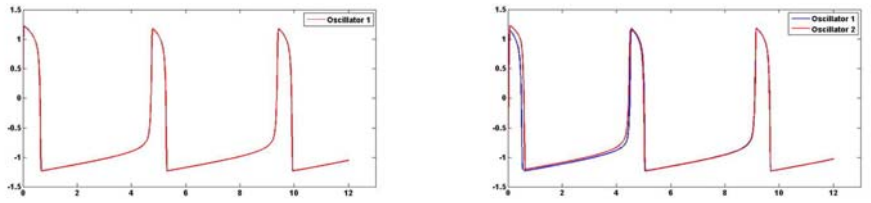
In computer vision, visual grouping has been studied in various mathematical frameworks, including graph-based methods [164, 79] and in particular normalized cuts [179, 43], harmonic analysis [127], probabilistic approaches [57, 47, 46, 48], variational formulations [157, 153, 4], Markov Random Fields [74], statistical techniques [71, 29, 41], among

others [193, 165].

In the brain, at a finer level of functional detail, the distributed *synchronization* known to occur at different scales has been proposed as a general functional mechanism for perceptual grouping [15, 180, 209]. In computer vision, however, comparatively little attention has been devoted to exploiting neural-like oscillators in visual grouping. Wang and his colleagues have performed very innovative work using oscillators for image segmentation [186, 204, 205, 116, 26] and have extended the scheme to auditory segregation [12, 202, 203]. They constructed oscillator networks with local excitatory lateral connections and a global inhibitory connection. Adopting similar ideas, Yen and Finkel have simulated facilitatory and inhibitory interactions among oscillators to conduct contour integration [216]. Li has proposed elaborate visual cortex models with oscillators and applied them on lattice drawings segmentation [106, 107, 108, 109]. Kuzmina and his colleagues [96, 97] have constructed a simple self-organized oscillator coupling model, and applied it on synthetic lattice images as well. Faugeras et al. have started studying oscillatory neural mass models in the contexts of natural and machine vision [63].

Chapter 7 introduces a simple and general biologically plausible visual grouping implementation with neural oscillators [223, 224], based on diffusive connections and concurrent synchronization [166]. A neural oscillator network is constructed, with each oscillator associated to an atomic element in the stimuli, for example a point, an orientation or a pixel. Without coupling, the oscillators are desynchronized and oscillate in random phases. Under diffusive coupling with the coupling strength appropriately tuned, they may converge to multiple groups of synchronized elements, namely concurrent synchronization. The synchronization of oscillators within each group indicates the perceptual grouping of the underlying stimulative atoms, while the desynchronization between groups suggests group segregation.

As illustrated in Figure 1.9(a), a neural oscillator is a dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$



a **b**

Figure 1.9: **a.** the oscillation trace of a single oscillator. **b.** synchronization of two oscillators coupled through diffusive connections. The two oscillators start to be fully synchronized at about $t = 5$.

that oscillates when it receives appropriate input, where $\mathbf{x}$ is the state vector and t is the time index. The oscillators are connected with diffusive coupling [207]

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i, t) + \sum_{i \neq j} k_{ij}(\mathbf{x}_j - \mathbf{x}_i), \qquad i = 1, \ldots, N \tag{1.5}$$

where k_{ij} is the coupling strength. Oscillators i and j are said to be *synchronized* if $\mathbf{x}_i$ remains equal to $\mathbf{x}_j$. Once the elements are synchronized, the coupling terms in (1.5) disappear, so that each individual element exhibits its natural and uncoupled behavior, as illustrated in Figure 1.9(b). A larger value of k_{ij} tends to reduce the state difference $\mathbf{x}_i - \mathbf{x}_j$ and thus to reinforce the synchronization between oscillators i and j.

The key to using diffusively-coupled neural oscillators for visual grouping is to tune the couplings so that the oscillators synchronize if their underlying atoms belong to the same visual group, and desynchronize otherwise. According to Gestalt psychology [210, 92, 140], visual stimulative atoms having similarity (e.g. gray-level, color, orientation) or proximity tend to be grouped perceptually. This suggests making strong coupling between neural oscillators whose underlying stimuli are similar. Such coupling is implemented by a Gaussian tuning

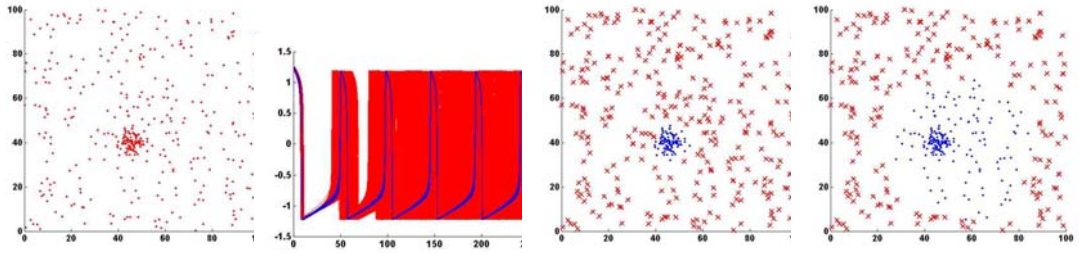
$$k_{ij} = e^{\frac{-|s_i - s_j|^2}{\beta^2}}. \tag{1.6}$$

where s_i and s_j are stimuli of the two oscillators, for example position for point clustering,

orientation for contour integration and gray-level for image segmentation, and β is a tuning parameter. The coupling strength falls off as a Gaussian function of the distance between the stimuli.

Figure 1.10 illustrates an example in which a dense cluster of 100 Gaussian distributed points is embedded in a cloud of 300 points uniformly randomly distributed. The neural oscillator system converges to one synchronized group that corresponds to the cluster with all the "outliers" totally desynchronized in the background, as shown in Figure 1.10(b). The resulting identification of the underlying cluster is shown in Figure 1.10(c). Normalized cuts [179] confuses a large number of outliers around the cluster of interest, as illustrated in Figure 1.10(d).

The same algorithm is applied on contour integration and image segmentation. As shown in Figures 1.11 and 1.12, it achieves promising results compared with state-of-the-art computer vision approaches normalized-cuts [179, 43].



a **b** **c** **d**

Figure 1.10: **a.** A cloud of points made of 300 points uniformly randomly distributed in a space of size 100×100, in addition to a cluster of 100 Gaussian distributed points with standard deviation equal to 3×3. **b.** The neural oscillator system converges to one synchronized group that corresponds to the cluster with all the "outliers" totally desynchronized in the background. **c.** and **d.** Clustering results by respectively neural oscillators and normalized cuts: blue dots represent the cluster detected by the algorithm and red crosses are the "outliers". In the latter many outliers are confused with the cluster of interest.

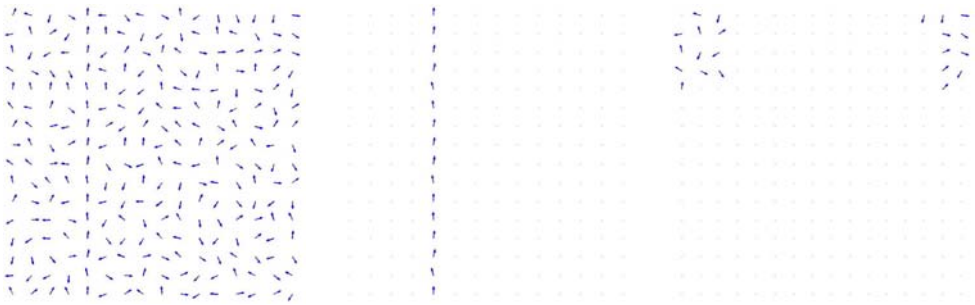


Figure 1.11: From left to right: A vertical contour is embedded in a uniformly distributed orientation grid, contour integration by neural oscillators and normalized cuts.

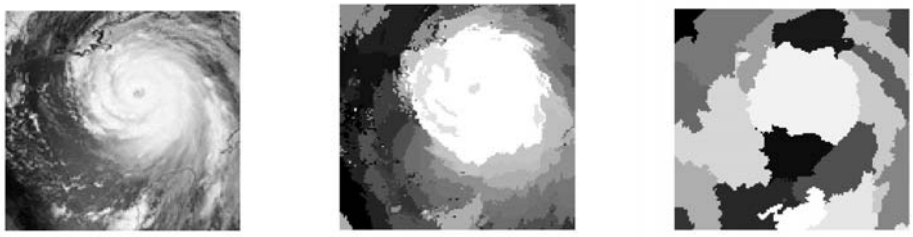


Figure 1.12: Real image segmentation. From left to right: a radar image (128×128); segmentation in 20 classes by neural oscillators and multiscale normalized cuts.

1.4 Thesis Organization

This thesis consists of three parts. The first part of the thesis is devoted to image and audio estimation with grouping in sparse representations. Chapter 2 introduces a time-frequency block thresholding procedure for audio denoising. Chapter 3 generalizes block thresholding and introduces a block pursuit algorithm. Applications in image denoising are shown. An image super-resolution zooming algorithm is derived in Chapter 4 with the block pursuit procedures that identify geometric image model and calculate structured sparse representations.

The second part of the thesis addresses invariant image comparison. A short Chapter 5 is devoted to the mathematical arguments proving that SIFT is similarity invariant. Chapter 6 introduces a new ASIFT algorithm that is fully affine invariant.

In the third part of the thesis, Chapter 7 introduces a visual grouping implementation with networks of neural oscillators and shows applications on point clustering, contour integration and image segmentation.

Part I

Sparse Image and Audio Processing with Blocks

Chapter 2

Audio Denoising by Time-Frequency Block Thresholding

This Chapter introduces a new time-frequency block audio denoising algorithm based on the block thresholding estimation [17, 18, 16]. Block sizes are automatically adjusted by minimizing a Stein estimator of the block thresholding risk [185]. The numerical experiments show that the proposed method removes efficiently the "musical noise" artifact and improves the SNR and the perceived quality with respect to state-of-the-art audio denoising algorithms.

2.1 Introduction

Audio signals, whether music or speech, are often corrupted by noise during recording and transmission. Audio denoising procedures are designed to attenuate the noise and retain the signal of interest.

Diagonal time-frequency audio denoising algorithms attenuate the noise by processing each window Fourier or wavelet coefficient independently, with empirical Wiener [139], power subtraction [10, 11, 110] or thresholding operators [49]. These algorithms create isolated time-frequency structures that are perceived as a "musical noise" [22, 217]. Ephraim and Malah [54, 55] showed that this musical noise is strongly attenuated with non-diagonal time-frequency estimators that regularize the estimation by recursively aggregating time-frequency coefficients. This approach has further been improved by optimizing the SNR estimation with parameterized filters [35] that rely on stochastic audio models. However, these parameters should be adjusted to the nature of the audio signal, which often varies and is unknown. In practice, they are empirically fixed [22, 35, 54, 55].

This Chapter introduces a new non-diagonal audio denoising algorithm through adaptive time-frequency block thresholding [217]. Block thresholding has been introduced by Cai and Silverman in mathematical statistics [17, 18, 16] to improve the asymptotic decay of diagonal thresholding estimators. For audio time-frequency denoising, we show that block thresholding regularizes the estimate and is thus effective in musical noise reduction. Block parameters are automatically adjusted by minimizing a Stein estimator of the risk [185], which is calculated analytically from the noisy signal values. Numerical experiments show that this new adaptive estimator is robust to signal type variations and improves the SNR and the perceived quality with respect to state of the art audio denoising algorithms.

The Chapter first reviews the state of the art time-frequency audio denoising algorithms by emphasizing the difference between diagonal and non-diagonal methods. Section 2.3 introduces time-frequency block thresholding and computes a Stein unbiased estimate of the resulting risk to adjust automatically the block parameters. Numerical experiments and comparisons are presented in Section 2.4, with objective and subjective measures.

2.2 State of the Art

2.2.1 Time-frequency Audio Denoising

Time-frequency audio-denoising procedures compute a short-time Fourier transform or a wavelet transform or a wavelet packet transform of the noisy signal, and processes the resulting coefficients to attenuate the noise. These representations reveal the time-frequency signal structures that can be discriminated from the noise. We concentrate on the coefficient processing as opposed to the choice of representations. Numerical experiments are performed with short-time Fourier transforms that are most commonly used in audio processing.

The audio signal f is contaminated by a noise w that is often modeled as a zero mean Gaussian process independent of f:

$$y[n] = f[n] + w[n], \qquad n = 0, 1, \ldots, N-1. \tag{2.1}$$

A time-frequency transform decomposes the audio signal y over a family of time-frequency atoms $\{g_{l,k}\}_{l,k}$ where l and k are the time and frequency (or scale) localization indices. The resulting coefficients shall be written:

$$Y[l,k] = \langle y, g_{l,k} \rangle = \sum_{n=0}^{N-1} y[n]\, g^*_{l,k}[n] \ .$$

where * denotes the conjugate. These transforms define a complete and often redundant signal representation. In this Chapter we shall suppose that these time-frequency atoms define a tight frame [40, 126], which means that there exists $A > 0$ such that

$$\|y\|^2 = \frac{1}{A} \sum_{l,k} |\langle y, g_{l,k} \rangle|^2.$$

This implies a simple reconstruction formula

$$y[n] = \frac{1}{A} \sum_{l,k} Y[l,k]\, g_{l,k}[n].$$

The constant A is a redundancy factor and if $A = 1$ then a tight frame is an orthogonal basis. A tight frame behaves like a union of A orthogonal bases.

A frame representation provides an energy control. The redundancy implies that a signal f has a non-unique way to be reconstructed from a tight frame representation: $f[n] = \frac{1}{A}\sum_{l,k} C[l,k]\, g_{l,k}[n]$, but all such reconstructions satisfy

$$\|f\|^2 \leq \frac{1}{A}\sum_{l,k} |C[l,k]|^2 \ , \tag{2.2}$$

with an equality if $C[l,k] = \langle f, g_{l,k}\rangle, \forall l,k$.

Short-time Fourier atoms can be written: $g_{l,k}[n] = s[n-lu]\exp\left(\frac{i2\pi kn}{K}\right)$, where $s[n]$ is a time window of support size K, which is shifted with a step $u \leq K$. l and k are respectively the integer time and frequency indices with $0 \leq l < N/u$ and $0 \leq k < K$. In this Chapter, $s[n]$ is the square root of a Hanning window and $u = K/2$ so one can verify that the resulting window Fourier atoms $\{g_{l,k}\}_{l,k}$ define a tight frame with $A = 2$.

A denoising algorithm modifies time-frequency coefficients by multiplying each of them by an attenuation factor $a[l,k]$ to attenuate the noise component. The resulting "denoised" signal estimator is:

$$\hat{f}[n] = \frac{1}{A}\sum_{l,k} \hat{F}[l,k]\, g_{l,k}[n] = \frac{1}{A}\sum_{l,k} a[l,k]\, Y[l,k]\, g_{l,k}[n] \ . \tag{2.3}$$

Time-frequency denoising algorithms differ through the calculation of the attenuation factors $a[l,k]$. The noise coefficient variance

$$\sigma^2[l,k] = E\{|\langle w, g_{l,k}\rangle|^2\}$$

is supposed to be known or estimated with methods such as [34, 49, 132]. If the noise is stationary, which is often the case, then the noise variance does not depend upon time: $\sigma^2[l,k] = \sigma^2[k]$.

2.2.2 Diagonal Estimation

Simple time-frequency denoising algorithms compute each attenuation factor $a[l,k]$ only from the corresponding noisy coefficient $Y[l,k]$ and are thus called *diagonal estimators*. These algorithms have a limited performance and produce a musical noise.

To minimize an upper bound of the quadratic estimation risk

$$r = E\{\|f - \hat{f}\|^2\} \leq \frac{1}{A} \sum_{l,k} E\{|F[l,k] - \hat{F}[l,k]|^2\}, \tag{2.4}$$

(2.4) being a consequence of (2.2), one can verify [49] that the optimal attenuation factor is

$$a[l,k] = 1 - \frac{1}{\xi[l,k]+1} \tag{2.5}$$

where $\xi[l,k] = F^2[l,k]/\sigma^2[l,k]$ is the *a priori* SNR. The resulting risk lower bound, also called oracle risk r_o, is

$$r_0 \leq \frac{1}{A} R_o \quad \text{where} \quad R_o = \sum_{l,k} \frac{|F[l,k]|^2 \sigma^2[l,k]}{|F[l,k]|^2 + \sigma^2[l,k]} \ . \tag{2.6}$$

This lower bound cannot be reached because the "oracle" attenuation factor (2.5) depends upon the *a priori* SNR $\xi[l,k]$ which is unknown. It is thus necessary to estimate this SNR.

Diagonal estimators of the SNR $\xi[l,k]$ are computed from the *a posteriori* SNR defined by $\gamma[l,k] = |Y[l,k]|^2/\sigma^2[l,k]$. One can verify that

$$\hat{\xi}[l,k] = \gamma[l,k] - 1 \tag{2.7}$$

is an unbiased estimator. Inserting this estimator in the oracle formula (2.5) defines the empirical Wiener estimator [110, 139]

$$a[l,k] = \left(1 - \frac{1}{\hat{\xi}[l,k]+1}\right)_+ \tag{2.8}$$

with the notation $(z)_+ = \max(z,0)$. Variants of this empirical Wiener are obtained by minimizing a sum of signal distortion and residual noise energy [56, 53, 87, 122] or by computing a maximum likelihood estimate [110, 139, 214].

Power subtraction estimators [10, 11, 110, 178, 183] generalize the empirical Wiener attenuation rule:

$$a[l,k] = \left(1 - \lambda \left[\frac{1}{\hat{\xi}[l,k]+1}\right]^{\beta_1}\right)_+^{\beta_2} \tag{2.9}$$

where $\beta_1, \beta_2 \geq 0$ and $\lambda \geq 1$ is an over-subtraction factor to compensate variation of noise amplitude.

Following the statistical work of Donoho and Johnstone [49], thresholding estimators have also been studied for audio noise removal. A hard thresholding [60, 95, 102, 200] either retains or sets to zero each noisy coefficient with

$$a[l,k] = 1_{\{\hat{\xi}[l,k]+1 > \lambda^2\}}. \tag{2.10}$$

Soft-thresholding estimator [24, 88, 104, 176] is a special case of power subtraction (2.9) with $\beta_1 = 1/2$, $\beta_2 = 1$. Donoho and Johnstone have proved that for Gaussian white noises, the quadratic risk of thresholding estimators is close to the oracle lower bound [49].

The attenuation factor $a[l,k]$ of these diagonal estimators only depends upon $Y[l,k]$ with no time-frequency regularization. The resulting attenuated coefficients $a[l,k]Y[l,k]$ thus lack of time-frequency regularity. It produces isolated time-frequency coefficients which restore isolated time-frequency structures that are perceived as a musical noise. Figure 2.1 shows the denoising of a short recording of the Mozart oboe concerto with an additive Gaussian white noise. Figure 2.1(a) and 2.1(b) show respectively the log spectrograms $\log|F[l,k]|$ and $\log|Y[l,k]|$ of the original signal f and its noisy version y. Figure 2.1(c) displays a power subtraction attenuation map $a[l,k]$, with black points corresponding to values close to 1. The zoom in Figure 2.1(c') shows that this attenuation map contains many isolated coefficients close to 1 (black points). These isolated coefficients restore isolated windowed Fourier vectors $g_{l,k}[n]$ that produce a musical noise.

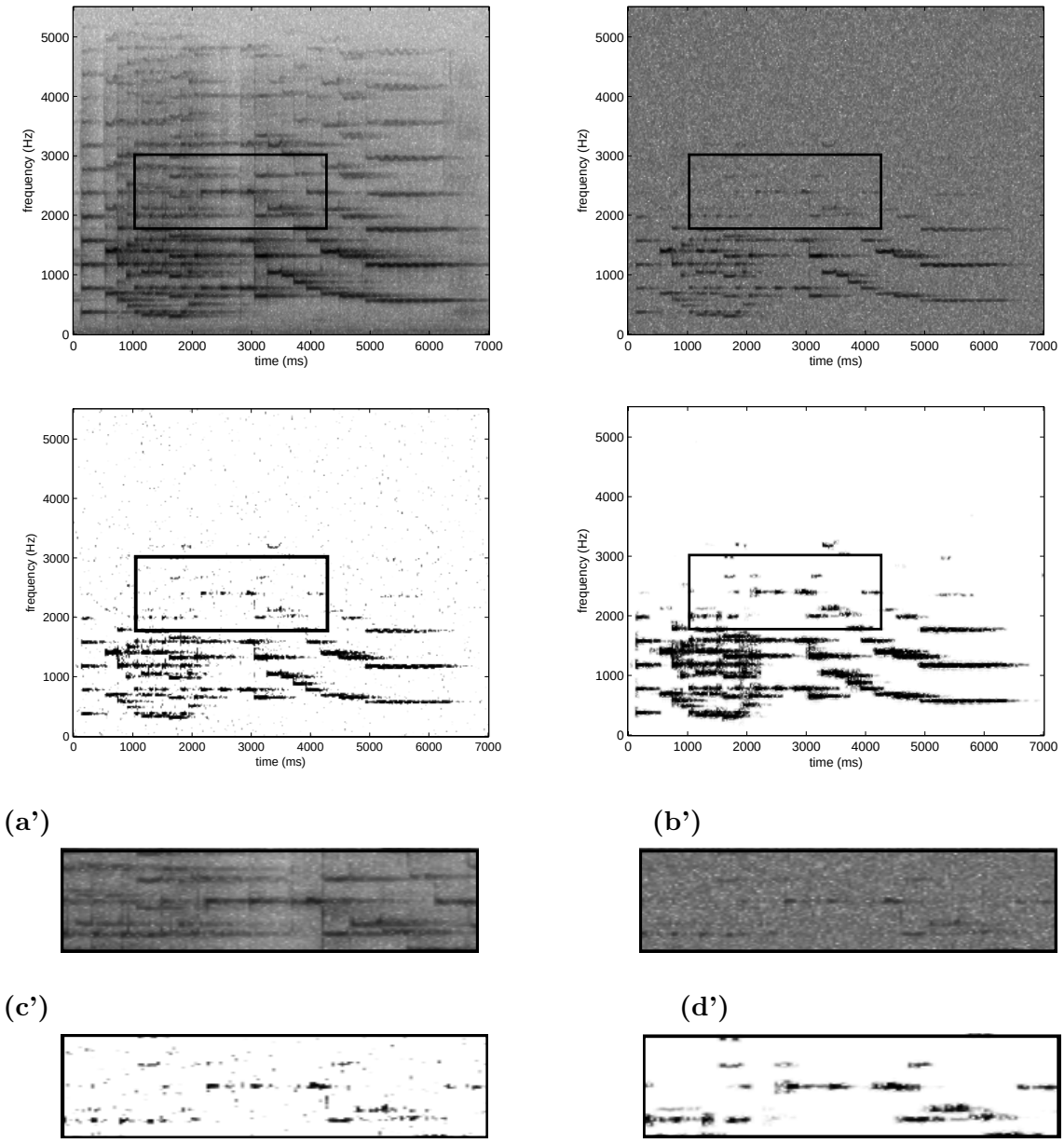


Figure 2.1: (a),(b): Log-spectrograms of the original and noisy "Mozart" signals. (c),(d): attenuation coefficients calculated with a power subtraction and a block thresholding. Black pixels correspond to 1 and white to 0. (a')(b')(c')(d'): zooms over rectangular regions indicated in (a)(b)(c)(d).

2.2.3 Non-diagonal Estimation

To reduce musical noise as well as the estimation risk, several authors have proposed to estimate the *a priori* SNR $\xi[l,k]$ with a time-frequency regularization of the *a posteriori* SNR $\gamma[l,k]$. The resulting attenuation factors $a[l,k]$ thus depend upon the data values $Y[l',k']$ for (l',k') in a whole neighborhood of (l,k) and the resulting estimator $\hat{f}[n] = \frac{1}{A}\sum_{l,k} a[l,k]\, Y[l,k]\, g_{l,k}[n]$ is said to be *non-diagonal*.

In their pioneer paper Ephraim and Malah [54] have introduced a *decision-directed* SNR estimator obtained with a first order recursive time filtering:

$$\hat{\xi}[l,k] = \alpha\, \tilde{\xi}[l-1,k] + (1-\alpha)\,(\gamma[l,k]-1)_+, \tag{2.11}$$

where $\alpha \in [0,1]$ is a recursive filter parameter and $\tilde{\xi}[l-1,k] = |\hat{F}[l-1,k]|^2/\sigma^2[l,k]$ is an empirical SNR estimate of $F[l-1,k]$ based on the previously computed estimate. This decision-directed SNR estimator has been applied with various attenuation rules such as empirical Wiener estimator (2.8) [32], Ephraim and Malah's minimum mean-square error spectral amplitude (MMSE-SA) [54], log spectral amplitude estimator (MMSE-LSA) [55] and Wolfe and Godsill's minimum mean-square error spectral power estimator (MMSE-SP) [211] that are derived from a Bayesian formulation using a Gaussian speech model [36, 33, 39, 54, 55, 95, 124], as well as Martin's MMSE estimators using a Gamma speech model [133]. These work clearly showed that the regularization of the SNR estimation reduces musical noise as well as the estimation risk $r = E\{\|\hat{f} - f\|^2\}$.

Cohen [35] improved the decision-directed SNR estimator by combining a causal recursive temporal filter with a noncausal compactly supported time-frequency filter to get a first SNR estimation. He then refines this estimation in a Bayesian formulation by computing a new SNR estimation using the MMSE-SP attenuation rule [211] from the first SNR estimate. This noncausal *a priori* SNR estimator has been combined with attenuation rules

derived from Gaussian [35, 36], Gamma and Laplacian speech models [37]. Other SNR estimators have been proposed by Cohen [38] with generalized autoregressive conditional heteroscedasticity (GARCH), applied with MMSE-LSA attenuation rules of Gamma and Laplacian speech models [38].

Matz and Hlawatsch have also proposed to estimate the SNR with a rectangular time-frequency filter and to use it together with the empirical Wiener estimator (2.8) [137]. In one example, they showed a noticeable performance gain with respect to a diagonal SNR estimation. The same non-diagonal SNR estimation has been applied in [138] where the authors automatically adapted the size of the short-time Fourier windows to the signal properties.

Thresholding estimators [49] have also been studied with time-regularized thresholds [77, 121], which are indirectly based on non-diagonal SNR estimations $\hat{\xi}[l,k]$. Such thresholds can further be adapted to a detection of speech presence [5, 27, 188].

Non-diagonal estimators clearly outperform diagonal estimators but depend upon regularization filtering parameters. Large regularization filters reduce the noise energy but introduce more signal distortion [22, 36, 54, 52]. It is desirable that filter parameters are adjusted depending upon the nature of audio signals. In practice, however, they are selected empirically [22, 35, 36, 54, 55]. Moreover, the attenuation rules and the *a priori* SNR estimators that are derived with a Bayesian approach [35, 37, 36, 38, 33, 39, 54, 55, 95, 124] model audio signals with Gaussian, Gamma or Laplacian processes. Although such models are often appropriate for speech, they do not take into account the complexity of other audio signals such as music, that include strong attacks.

2.3 Time-Frequency Block Thresholding

Block thresholding was introduced in statistics by Cai and Silverman [17, 18, 16] and studied by Hall et al. [82, 81, 83] to obtain nearly minimax signal estimators. The "p-point uncertainty model" proposed by Matz and Hlawatsch [137] also led to a block thresholding estimator with fixed parameters that are chosen empirically. For audio signal denoising, we describe an adaptive block thresholding non-diagonal estimator that automatically adjusts all parameters. It relies on the ability to compute an estimate of the risk, with no prior stochastic audio signal model, which makes this approach particularly robust.

2.3.1 Block Thresholding Algorithm

A time-frequency block thresholding estimator regularizes power subtraction estimation (2.9) by calculating a single attenuation factor over time-frequency blocks. The time-frequency plane $\{l, k\}$ is segmented in I blocks B_i whose shape may be chosen arbitrarily. The signal estimator $\hat{f}$ is calculated from the noisy data y with a constant attenuation factor a_i over each block B_i

$$\hat{f}[n] = \frac{1}{A} \sum_{i=1}^{I} \sum_{(l,k) \in B_i} a_i \, Y[l,k] \, g_{l,k}[n]. \tag{2.12}$$

To understand how to compute each a_i, one relates the risk $r = E\{\|f - \hat{f}\|^2\}$ to the frame energy conservation (2.2) and obtains

$$r = E\{\|f - \hat{f}\|^2\} \leq \frac{1}{A} \sum_{i=1}^{I} \sum_{(l,k) \in B_K} E\{|a_i Y[l,k] - F[l,k]|^2\}. \tag{2.13}$$

Since $Y[l,k] = F[l,k] + W[l,k]$ one can verify that the upper bound of (2.13) is minimized by choosing

$$a_i = 1 - \frac{1}{\xi_i + 1} \tag{2.14}$$

where $\xi_i = \overline{F_i^2}/\overline{\sigma_i^2}$ is the average *a priori* SNR in B_i. It is calculated from

$$\overline{F_i^2} = \frac{1}{B_i^\#} \sum_{(l,k)\in B_i} |F[l,k]|^2 \text{ and } \overline{\sigma_i^2} = \frac{1}{B_i^\#} \sum_{(l,k)\in B_i} \sigma^2[l,k] \ ,$$

which are the average signal energy and noise energy in B_i, and $B_i^\#$ is the number of coefficients $(l,k) \in B_i$. The resulting oracle block risk r_{bo} satisfies

$$r_{bo} \leq \frac{1}{A} R_{bo} \text{ where } R_{bo} = \sum_{i=1}^{I} B_i^\# \frac{\overline{F_i^2}\,\overline{\sigma_i^2}}{\overline{F_i^2} + \overline{\sigma_i^2}}. \tag{2.15}$$

The oracle block attenuation coefficients a_i in (2.14) can not be calculated because the *a priori* SNR ξ_i is unknown. Cai and Silverman [17] introduced block thresholding estimators that estimate the SNR over each B_i by averaging the noisy signal energy:

$$\hat{\xi}_i = \frac{\overline{Y_i^2}}{\overline{\sigma_i^2}} - 1 \tag{2.16}$$

where

$$\overline{Y_i^2} = \frac{1}{B_i^\#} \sum_{(l,k)\in B_i} |Y[l,k]|^2 \ .$$

Observe that if $\sigma[l,k] = \overline{\sigma}_i$ for all $(l,k) \in B_i$ then $\hat{\xi}_i$ is an unbiased estimator of ξ_i. The resulting attenuation factor a_i is computed with a power subtraction estimator (2.9)

$$a_i = \left(1 - \frac{\lambda}{\hat{\xi}_i + 1}\right)_+ . \tag{2.17}$$

A block thresholding estimator can thus be interpreted as a non-diagonal estimator derived from averaged SNR estimations over blocks. Each attenuation factor is calculated from all coefficients in each block, which regularizes the time-frequency coefficient estimation. Fig 2.1(d) displays a block thresholding attenuation map a_i with black points corresponding to values close to 1. The zoom in Fig 2.1(d') shows that non-diagonal block thresholding attenuation factors are much more regular than the diagonal power subtraction attenuation factors in Fig 2.1(c') and they do not keep isolated points responsible for musical noise.

2.3.2 Block Thresholding Risk and Choice of λ

An upper bound of the risk of the block thresholding estimator is computed by analyzing separately the bias and variance terms. Observe that the upper bound of the oracle risk r_{bo} in (2.15) with blocks is always larger than that of the oracle risk r_o in (2.6) without blocks, because the former is obtained through the same minimization but with less parameters as attenuation factors remain constant over each block. A direct calculation shows that

$$R_{bo} - R_o = \sum_{i=1}^{I} \sum_{(l,k)\in B_i} \frac{\overline{\xi_i}\xi[l,k](\overline{\sigma_i^2} - \sigma^2[l,k]) + (\overline{F_i^2} - |F[l,k]|^2)}{(\overline{\xi_i}+1)(\xi[l,k]+1)} \geq 0. \tag{2.18}$$

R_{bo} is close to R_o if both the noise and the signal coefficients have little variation in each block. This bias term is thus reduced by choosing the blocks so that in each block B_i either (i) $F[l,k]$ and $\sigma[l,k]$ vary little; or (ii) $\xi[l,k] \gg 1$ and $\sigma[l,k]$ varies little; or (iii) $\xi[l,k] \ll 1$ and $F[l,k]$ varies little.

Block thresholding (2.17) approximates the oracle block attenuation (2.14) by replacing ξ_i with an estimate $\hat{\xi}_i$ in (2.16) and by setting an over-subtraction factor $\lambda \geq 1$ to control the variance term of risk due to the noise variation. If the noise w is a Gaussian white noise, then the resulting risk $r = E\{\|f - \hat{f}\|^2\}$ can be shown to be close to the oracle risk (2.15). The average noise energy over a block B_i is

$$\overline{W_i^2} = \frac{1}{B_i^{\#}} \sum_{(l,k)\in B_i} |W[l,k]|^2. \tag{2.19}$$

If the frame is an orthogonal basis, in the particular case where all blocks B_i have the same size $B^{\#}$ and the noise is Gaussian white noise with variance σ^2 (hence $\overline{W_i^2} = \overline{W^2}$) then Cai [17] proved that

$$r = E\{\|\hat{f} - f\|^2\} \leq 2\lambda R_{bo} + 4N\sigma^2 Prob\{\overline{W^2} > \lambda\sigma^2\}, \tag{2.20}$$

where $Prob\{\}$ is the probability measure. We have mentioned that a tight frame behaves very similarly to a union of A orthogonal bases. Therefore the oracle inequality with a

frame representation holds as well:

$$r = E\{\|\hat{f} - f\|^2\} \leq \frac{2\lambda}{A} R_{bo} + \frac{4M}{A}\sigma^2 Prob\{\overline{W^2} > \lambda\sigma^2\}, \tag{2.21}$$

where $M \geq N$ is the number of vectors $g_{l,k}$ in the frame. For the window Fourier frame used in this Chapter, $M = 2N$ and $A = 2$.

The second term $4M\sigma^2 Prob\{\overline{W^2} > \lambda\sigma^2\}$ is a variance term corresponding to a probability of keeping pure noise coefficients, i.e., f is zero ($y = w$) and $a_i \neq 0$ (c.f. (2.17)). $Prob\{\overline{W^2} > \lambda\sigma^2\}$ is the probability to keep a residual noise. The oracle risk and the variance terms in (2.21) are competing. When λ increases the first term increases and the variance term decreases. Similarly, when the block size $B^{\#}$ increases the oracle risk R_{bo} increases whereas the variance decreases. Adjusting λ and the block sizes $B^{\#}$ can be interpreted as an optimization between the bias and the variance of our block thresholding estimator. The parameter λ is set depending upon $B^{\#}$ by adjusting the residual noise probability

$$Prob\{\overline{W^2} > \lambda\sigma^2\} = \delta. \tag{2.22}$$

The probability δ is a perceptual parameter. We set $\delta = 0.1\%$ in (2.22) as our psychoacoustic experiments show that with a residual noise probability $\delta \approx 0.1\%$, musical noise is hardly perceptible.

Let $B_i^{\#} = \mathsf{L}_i \times \mathsf{W}_i$ be a rectangular block size, where $\mathsf{L}_i \geq 2$ and $\mathsf{W}_i \geq 2$ are respectively the block length in time and the block width in frequency (the unit being the time-frequency index in the window Fourier transform). One can verify that with half overlapping Hanning windows the average noise energy $\overline{W^2}$ follows approximatively a χ^2 distribution degrees with $B_i^{\#}$ degree of freedom. Thus solving λ in (2.22) amounts to looking up a χ^2 table. Table 2.1 gives values for a frequency width $\mathsf{W}_i \geq 2$. Due to discretization effects, λ takes nearly the same values for $\mathsf{W}_i = 1$ and $\mathsf{W}_i = 2$. We thus compute λ for $\mathsf{W}_i = 1$ by multiplying $B_i^{\#}$ by

2 and looking at Table 2.1. That (2.22) holds with λ shown in Table 2.1 can also be verified by Monte Carlo simulation.

$B_i^{\#}$	4	8	16	32	64	128
λ	4.7	3.5	2.5	2.0	1.8	1.5

Table 2.1: Thresholding level λ calculated for different block size $B^{\#}$ with $\delta = 0.1\%$.

2.3.3 Adaptive Block Thresholding

A block thresholding segments the time-frequency plane in disjoint rectangular blocks of length L_i in time and width W_i in frequency. In the following by "block size" we mean a choice of block shapes and sizes among a collection of possibilities. The adaptive block thresholding chooses the sizes by minimizing an estimate of the risk.

The risk $E\{\|f-\hat{f}\|^2\}$ cannot be calculated since f is unknown, but it can be estimated with a Stein risk estimate [185]. Best block sizes are computed by minimizing this estimated risk. We saw in (2.13) that the block thresholding risk satisfies

$$r = E\{\|f-\hat{f}\|^2\} \leq \frac{1}{A}\sum_{i=1}^{I}\sum_{(l,k)\in B_i} E\{|a_i Y[l,k]-F[l,k]|^2\}. \tag{2.23}$$

Since $Y[l,k] = F[l,k] + W[l,k]$ and $W[l,k]$ has a zero mean, $F[l,k]$ is the mean of $Y[l,k]$. To estimate the block thresholding risk Cai [16] uses the Stein estimator of the risk when computing the mean of a random vector, which is given by Stein theorem [185].

Theorem (Stein Unbiased Risk Estimate SURE)**.** *Let* $\mathbf{Y} = (Y_1,\ldots,Y_p)$ *be a normal random vector with the identity as covariance matrix and mean* $\mathbf{F} = (F_1,\ldots,F_p)$*. Let* $\mathbf{Y}+\mathbf{h}(\mathbf{Y})$ *be an estimator of* $\mathbf{F}$*, where* $\mathbf{h} = (h_1,\ldots,h_p) : \mathrm{R}^p \to \mathrm{R}^p$ *almost differentiable (*$h_j : \mathrm{R}^p \to \mathrm{R}^1, \forall j$*). Define* $\triangledown \cdot \mathbf{h} = \sum_{j=1}^{p} \frac{\partial}{\partial Y_j} h_j$*. If* $E\left\{\sum_{j=1}^{p} |\frac{\partial}{\partial Y_j} h_j(\mathbf{Y})|\right\} < \infty$*, then*

$$R = E\|\mathbf{Y}+\mathbf{h}(\mathbf{Y})-\mathbf{F}\|^2 = p + E\left\{\|\mathbf{h}(\mathbf{Y})\|^2 + 2\triangledown\cdot\mathbf{h}(\mathbf{Y})\right\}. \tag{2.24}$$

So

$$\hat{R} = p + \|\mathbf{h}(\mathbf{Y})\|_2^2 + 2\nabla \cdot \mathbf{h}(\mathbf{Y}) \tag{2.25}$$

is an unbiased estimator of the risk R of $\mathbf{Y} + \mathbf{h}(\mathbf{Y})$, called Stein Unbiased Risk Estimator [185].

An estimation of the risk $E\{\|\hat{f} - f\|^2\}$ upper bound (2.23) is derived from this theorem by computing an estimator $\hat{R}_i$ of the risk in each block B_i: $R_i = \sum_{(l,k)\in B_i} E\{|F[l,k] - a_i Y[l,k]|^2\}$. Over a block B_i, the mean vector $\mathbf{F}_i = (F[l,k])_{(l,k)\in B_i}$ of $\mathbf{Y}_i = (Y[l,k])_{(l,k)\in B_i}$ is estimated by $\hat{\mathbf{F}}_i = (\hat{F}[l,k])_{(l,k)\in B_i}$ with $\hat{\mathbf{F}}_i = a_i \mathbf{Y}_i = \mathbf{Y}_i + \mathbf{h}(\mathbf{Y}_i)$. From the expression (2.17) of a_i we derive that

$$\mathbf{h}(\mathbf{Y}_i) = -\mathbf{Y}_i \left(\lambda \frac{\overline{\sigma}_i^2}{\overline{Y}_i^2} 1_{\overline{Y}_i^2 \geq \lambda \overline{\sigma}_i^2} + 1_{\overline{Y}_i^2 < \lambda \overline{\sigma}_i^2} \right).$$

Under the hypothesis that the noise variance remains constant on each block, $\sigma^2[l,k] = \overline{\sigma}_i^2$ for $(l,k) \in B_i$, the resulting Stein estimator of the risk $R_i = \sum_{l,k\in B_i} E\{|F[l,k] - a_i Y[l,k]|^2\}$ is

$$\hat{R}_i = \overline{\sigma}_i^2 \left(B_i^{\#} + E\left\{ \|\mathbf{h}(\mathbf{Y}_i/\overline{\sigma}_i)\|^2 + 2\nabla \cdot \mathbf{h}(\mathbf{Y}_i/\overline{\sigma}_i) \right\} \right) \tag{2.26}$$

and a direct calculation shows that

$$\hat{R}_i = \overline{\sigma}_i^2 \left(B_i^{\#} + \frac{\lambda^2 B_i^{\#} - 2\lambda(B_i^{\#} - 2)}{\overline{Y_i^2}/\overline{\sigma}_i^2} \mathbf{1}_{\overline{Y_i^2} \geq \lambda \overline{\sigma_i^2}} + B_i^{\#} \, (\overline{Y_i^2}/\overline{\sigma}_i^2 - 2) \mathbf{1}_{\overline{Y_i^2} < \lambda \overline{\sigma_i^2}} \right). \tag{2.27}$$

If the noise is Gaussian white and the frame is an orthogonal basis then the noise coefficients are uncorrelated with same variance and Stein theorem proves that $\hat{R}_i$ is an unbiased risk estimator of the risk R_i. If the noise is not white but stationary then the noise variance does not change in time. If the blocks B_i are sufficiently narrow in frequency then the noise variance still remains constant over each block so the risk estimator remains unbiased. We mentioned that a tight frame behaves very similarly to a union of A orthogonal bases. As

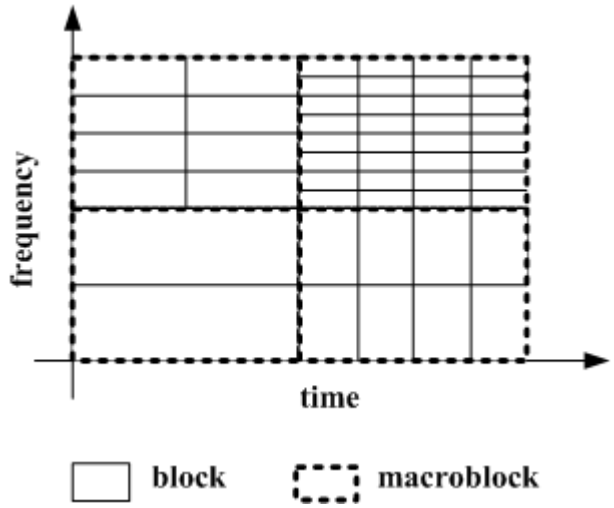


Figure 2.2: Partition of macroblocks into blocks of different sizes.

a consequence, the theorem result applies approximately and the resulting estimator mains nearly unbiased.

The adaptive block thresholding groups coefficients in blocks whose sizes are adjusted to minimize the Stein risk estimate and it attenuates coefficients in those blocks. To regularize the adaptive segmentation in blocks, the time-frequency plane is first decomposed in macroblocks M_j, $j = 1, 2 \ldots, J$, as illustrated in Figure 2.2. Each macroblock M_j is segmented in blocks B_i of same size which means that $B_i^{\#} = P_j$ is constant over a macroblock M_j. The Stein risk estimation over M_j is $\frac{1}{A} \sum_{i \in M_j} \hat{R}_i$. Several such segmentations are possible and we want to choose the one that leads to the smallest risk estimation. The optimal block size and hence P_j is calculated by choosing the block shape that minimizes $\sum_{i \in M_j} \hat{R}_i$. Once the block sizes are computed, coefficients in each B_i are attenuated with (2.17), where λ is calculated with (2.22).

In numerical experiments, each macroblock is segmented with 15 possible block sizes $L \times W$ with a combination of block length $\mathsf{L} = 8, 4, 2$ and block width $\mathsf{W} = 16, 8, 4, 2, 1$.

The size of macroblocks is set to be equal to the maximum block size 8×16. Figure 2.2 illustrates different segmentations of these macroblocks into time-frequency blocks of same size. Minimizing the estimated risk adapts the blocks to the signal time-frequency properties. In particular, it eliminates "pre-echo" artifacts on signal onsets and results in less distortion on signal transients.

Figure2.3(a) zooms on the onset of "Mozart" signal whose log-spectrogram is illustrated in Fig 2.1(b). The attenuation factors of block thresholding with a fixed block size $\mathsf{L} = 8$ and $\mathsf{W} = 1$ are displayed in Figure2.3(b). At the beginning of the harmonics, blocks of large attenuation factors spread beyond the onset of the signal. Fig2.3(b') illustrates the horizontal blocks at the onsets marked in Figs 2.3(a) and (b). In the time interval where the blocks exceed the signal onset, moderate attenuation is performed, and since the noise is not eliminated a transient noise component is heard before the signal beginning. This can be called as a "pre-echo" artifact. On the other hand, this moderate attenuation in the blocks that exceeds signal onsets muffles the onsets as well.

In Figs 2.3(c)(c'), the adaptive block method chooses blocks of shorter length L in the first part of "Mozart", which hardly exceed the onset of the signal. This reduces considerably the "pre-echo" artifact. After the onset, the adaptive block method chooses narrow horizontal blocks, to better capture the harmonic signal structures.

2.3.4 Non-Diagonal Wiener Post-Processing and Masking Noise

Similarly to the bootstrapping algorithm of Cohen [35] which performs a second SNR estimation from the signal obtained after a first denoising, the block thresholding estimation is improved by applying a second thresholding estimation. A block-thresholding algorithm regularizes the time-frequency estimation as compared to a diagonal thresholding, but it outputs a time-frequency estimation with some block structures as shown in Figure 2.4(b).

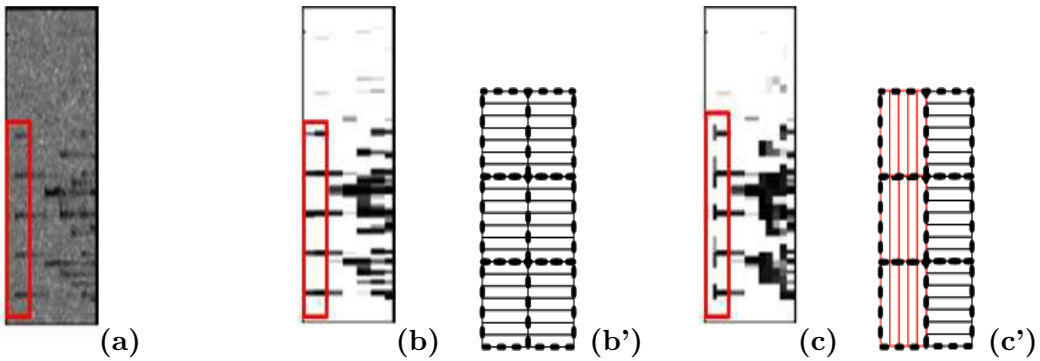


Figure 2.3: Zoom on the onset of "Mozart". (a): Log-spectrogram. (b): Attenuation coefficients of a fixed block thresholding. (b'): Block sizes in the time-frequency rectangle at the signal onset. (c): Attenuation coefficients of an adaptive block thresholding. (c'): Adapted block sizes at the signal onset.

This first estimation is used as an input to compute a Wiener time-frequency estimation that takes advantage of the time-frequency regularization provided by the block thresholding estimation.

Let $\hat{f}$ be the block thresholding estimation from the noisy data y. Similarly to the post-processing proposed by Baraniuk for images denoising [76], this first estimation is post-processed by computing a new attenuation factor using the oracle formula (2.5) calculated from its time-frequency coefficients $\hat{F}[l,k] = \langle \hat{f}, g_{l,k} \rangle$:

$$\tilde{a}[l,k] = \frac{|\hat{F}[l,k]|^2}{|\hat{F}[l,k]|^2 + \sigma^2[l,k]}. \tag{2.28}$$

This new attenuation factor is applied on the noisy time-frequency coefficients to reconstruct a second estimator.

$$\tilde{f}[n] = \frac{1}{A} \sum_{l,k} \tilde{a}[l,k]\, Y[l,k]\, g_{l,k}[n] \ .$$

This Wiener estimator is non-diagonal since the attenuation coefficients $\tilde{a}[l,k]$ depend upon values of $Y[l',k']$ in a time-frequency neighborhood of (l,k). Comparing with Fig 2.4(b), Fig 2.4(c) shows that the amplitude of the non-diagonal Wiener attenuation factors $\tilde{a}[l,k]$ is more regular then the block thresholding attenuation factors and is closer to the oracle

attenuation (2.5) displayed in Figure 2.4(d). Experiments show that this post-processing increases the SNR on average by about 0.2 dB and improves the audio quality of denoised signals.

Retaining a low-amplitude noise is sometimes desirable to mask artifices generated by an estimation procedure [10, 178]. Following [10], one can retain a masking noise by setting a floor value to the attenuation factor:

$$\tilde{a}_M[l,k] = \max(\tilde{a}[l,k], a_0) \tag{2.29}$$

where $0 < a_0 \ll 1$ is the minimum attenuation factor of the noise.

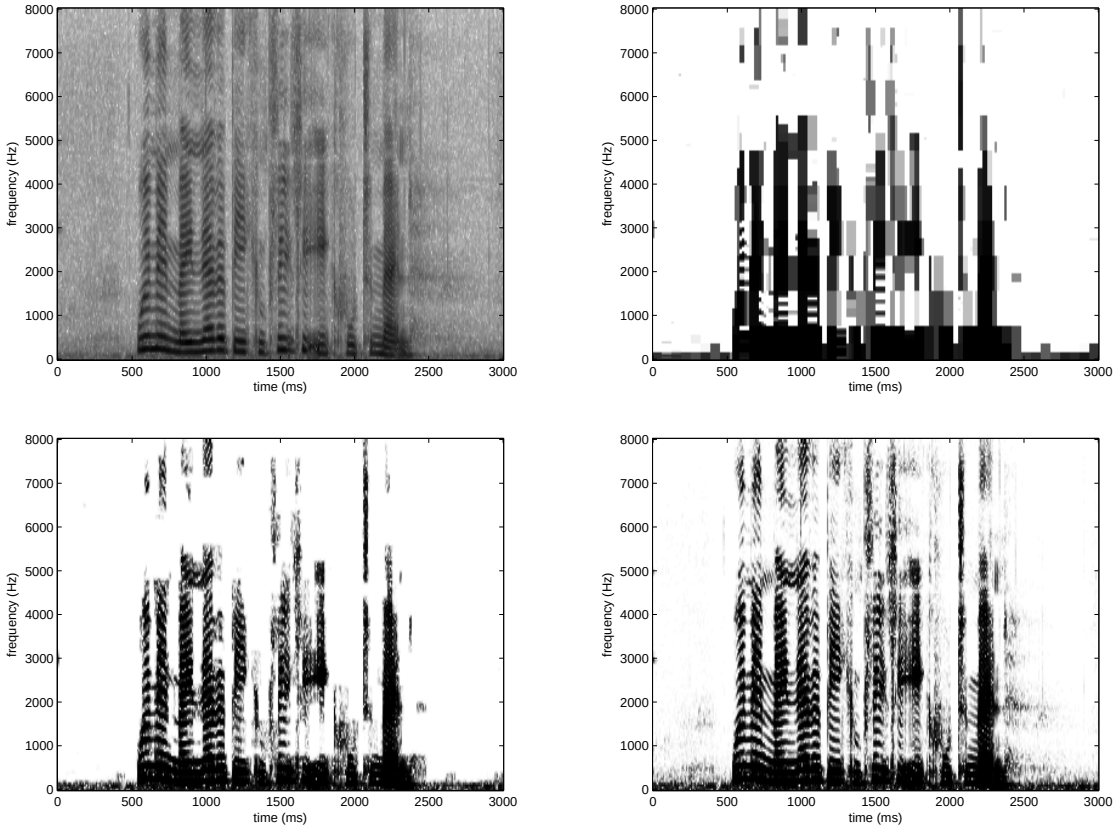


Figure 2.4: (a): log-spectrogram of "TIMIT-F". (b),(c),(d): attenuation coefficients respectively of a block thresholding, of a non-diagonal Wiener estimator, and of an oracle estimator.

2.4 Experiments and Results

The experiments presented below have been performed on various types of audio signals: "Piano" is a simple example that contains a single clear clavier stroke; "Mozart" is a musical excerpt that contains relatively quick notes played by a solo oboe; "TIMIT-M" and "TIMIT-F" are respectively male and female utterances taken from the TIMIT database [72]. "TIMIT-M" and "TIMIT-F" are sampled at 16 kHz whereas all the other signals are sampled at 11 kHz. They were corrupted by Gaussian white noise of different amplitude. Short-time Fourier transform with half-overlapping windows were used in the experiments. These windows are square root of Hanning windows of size 50 ms for "Piano" and "Mozart" and 20 ms for "TIMIT-M" and "TIMIT-F".

For each sound, denoising with "partial noise removal" and "maximum noise removal" were applied: the former retains some low-amplitude residual noise; the latter removes almost all the original noise.

Block thresholding was configured as described in Sections 2.3.3 and 2.3.4. For partial noise removal and maximum noise removal, we respectively set $a_0 \approx 0.05$ (the residual noise was calibrated to have similar energy for all methods under comparison) and $a_0 = 0$ in (2.29).

MMSE-LSA attenuation rule [55] of Ephraim and Malah was also used in our evaluation. Combined with the decision-directed *a priori* SNR estimator (2.11) with $\alpha = 0.98$ as proposed in [54, 55], this algorithm (referred to as LSA-DD) led to satisfactory results for partial noise removal. However, it resulted in too much signal distortion for maximum noise removal as a larger α was configured. Consequently, for this case, we substituted the decision-directed SNR estimator by the noncausal SNR estimator recommended in [35] which has been shown more effective in noise reduction. The so-obtained algorithm is

referred to as LSA-NC.

Power subtraction (2.9) was configured with $\lambda = 5$, $\beta_1 = \beta_2 = 1$ as recommended in [10]. The floor value a_0 in (2.29) has the same values as the ones chosen for block thresholding ($a_0 \approx 0.05$ for partial noise removal and $a_0 = 0$ for maximum noise removal).

Both objective and subjective evaluations have been performed. The objective measures are respectively the SNR and the segmental SNR [169] defined as

$$SNR = 10\log_{10}\frac{\sum_{n=0}^{N-1} f^2[n]}{\sum_{n=0}^{N-1}(f[n]-\hat{f}[n])^2} \tag{2.30}$$

$$SegSNR = \frac{1}{H}\sum_{l=0}^{H-1}\mathcal{T}\left(10\log_{10}\frac{\sum_{n=0}^{S-1} f^2[n+lS/2]}{\sum_{n=0}^{S-1}(f[n+lS/2]-\hat{f}[n+lS/2])^2}\right) \tag{2.31}$$

where H represents the number of frames in the signal, S is the number of samples per frame that corresponds to 32 ms, and $\mathcal{T}(x) = \min[\max(x,-10),35]$ confines the SNR in each frame to a perceptually meaningful range between 35 dB and -10 dB. Segmental SNR has been shown to have a higher correlation with perceived quality than SNR does [169].

Table 2.2 compares the SNR and the segmental SNR of the three denoising algorithms : block thresholding (BT), MMSE-LSA based algorithms (LSA-DD or LSA-NC) and power substraction (PS). One can observe that the MMSE-LSA based algorithms achieved systematically a better SNR than the power subtraction method, the average gain being 0.3 dB for partial noise removal and 1.3 dB for maximum noise removal. Yet another systematic SNR improvement was achieved by block thresholding over MMSE-LSA, with an average gain of 0.9 dB for partial noise removal and 0.8 dB for maximum noise removal. With respect to segmental SNR, though the average gains are smaller, these results are confirmed: block thresholding outperformed MMSE-LSA based algorithms which performed better than power substraction.

The subjective evaluation was performed by a large group of 200 adult listeners. All subjects claimed to have normal hearing, 151 claimed to listen to music regularly, 58 claimed

Signal & SNR	Partial Noise Removal			Maximum Noise Removal		
	PS	LSA-DD	BT	PS	LSA-NC	BT
Mozart -2.73 dB	8.68	8.91	**11.12**	8.75	10.15	**11.90**
Mozart 3.46 dB	13.01	13.21	**14.46**	12.92	14.01	**14.45**
Mozart 9.23 dB	17.17	17.93	**18.40**	16.98	18.10	**18.45**
Mozart 14.73 dB	21.11	21.12	**22.49**	20.87	21.99	**22.43**
Piano 4.75 dB	17.70	18.24	**19.95**	18.30	19.45	**20.47**
TIMIT-M 10.76 dB	18.65	18.84	**19.46**	18.55	19.16	**19.70**
TIMIT-F 20.63 dB	25.15	25.21	**26.46**	24.95	25.88	**26.38**

Signal & SSNR	Partial Noise Removal			Maximum Noise Removal		
	PS	LSA-DD	BT	PS	LSA-NC	BT
Mozart -5 dB	6.32	7.17	**8.53**	6.80	8.23	**9.77**
Mozart 0 dB	10.56	11.61	**12.12**	10.76	12.14	**12.24**
Mozart 5 dB	14.79	15.87	**15.92**	14.79	16.01	**16.14**
Mozart 10 dB	18.68	19.31	**19.96**	18.52	19.78	**19.90**
Piano -5 dB	5.74	6.70	**7.53**	6.77	8.42	**8.94**
TIMIT-M 0 dB	9.16	9.97	**9.98**	9.61	10.85	**11.02**
TIMIT-F 10 dB	15.04	15.70	**16.51**	14.88	15.67	**16.45**

Table 2.2: Comparison of Power Subtraction (PS), Ephraim and Malah (LSA-DD or LSA-NC) and Block Thresholding (BT) algorithms, on 4 types of noisy signals with different noise levels. The top table gives the SNR values for partial noise removal and maximum noise removal, and the bottom table gives the segmental SNR values.

to have some general knowledge on signal processing and 26 claimed to have had experience using audio processing softwares. The authors were obviously excluded from this test.

Each subject participated in an evaluation of successively the 7 sounds mentioned above. The evaluation of each sound consisted in 3 consecutive steps: partial noise removal, maximum noise removal and a comparison between these two noise removals. For the first two steps, each subject had to rank the 3 denoising results (block thresholding, MMSE-LSA and power subtraction) according to their global appreciation of the sounds. Let us note that they had the possibility to give a same rank to several methods each time. In the third step, each subject had to select between the 2 previously top ranked denoising results (i.e., the top ranked partial denoising result and the top ranked maximum denoising result) the

one they appreciated the most. In all cases, the subjects could listen to the denoising results as well as to the noisy sounds as many times as they wished. The order of the sounds and of the denoising results were randomized in order to minimize any bias. The overall test for a single subject lasted for about 15 minutes.

The subjective evaluation showed clearly that the power subtraction algorithm is by far the least favored as it obtained less than 4% top ranking votes for each of the sounds. The major complaint the subjects had about it was the strong musical noise artifact.

Signal & SSNR	Partial Noise Removal			Maximum Noise Removal		
	BT	LSA-DD	*EQU.*	BT	LSA-NC	*EQU.*
Mozart -5 dB	**47.0**	26.0	*27.0*	**80.1**	10.5	*9.4*
Mozart 0 dB	**47.3**	21.6	*31.1*	**44.1**	37.5	*18.4*
Mozart 5 dB	**53.2**	22.8	*24.0*	**40.4**	38.7	*20.9*
Mozart 10 dB	**54.7**	12.0	*33.3*	**41.3**	24.7	*34.0*
Piano -5 dB	**54.7**	29.3	*16.0*	**70.0**	12.1	*17.9*
TIMIT-M 0 dB	**61.9**	10.7	*27.4*	**39.4**	38.5	*22.1*
TIMIT-F 10 dB	**34.5**	30.9	*34.5*	**37.0**	26.0	*37.0*
Music	**51.4**	22.3	*26.3*	**55.2**	24.7	*20.1*
95% CI	(48.2, 54.5)	(19.8, 25.0)	*(23.6, 29.1)*	(52.1, 58.3)	(22.1, 27.5)	*(17.7, 22.7)*
Speech	**48.2**	20.8	*31.0*	**38.2**	32.3	*29.5*
95% CI	(43.2, 53.2)	(16.9, 25.1)	*(26.5, 35.8)*	(33.4, 43.1)	(27.7, 37.1)	*(25.1, 34.2)*

Table 2.3: Subjective comparison between Block Thresholding (BT) and Ephraim and Malah (LSA-DD and LSA-NC), for partial noise removal and maximum noise removal. The columns BT and LSA give the percentage of listeners that preferred the corresponding algorithm over the other one, for each noisy signal. The column EQU. gives the percentage of listeners for whom the quality of both algorithms is equal. The last two table rows aggregate the results for all Music signals (Mozart and Piano) and all Speech signals (TIMIT-M and TIMIT-F), and they give the 95% confidence interval (CI) derived from the number of listeners.

Table 2.3 concentrates on the comparison between block thresholding and the MMSE-LSA algorithms. Confirming the previous (segmented) SNR results, in the case of musical sounds, the subjects showed a clear preference for block thresholding over MMSE-LSA for both partial noise removal and maximum noise removal. Again, for the male speech sound

TIMIT-M, block thresholding is very clearly preferred over the MMSE-LSA algorithm in the case of partial noise removal. Besides, a slight preference for block thresholding is shown for the female sound TIMIT-F in the case of maximum noise removal. On the other speech sounds (TIMIT-M with maximum noise removal and TIMIT-M with partial noise removal), the results do not show any significant difference. Table 2.3 also displays the 95% confidence intervals of the overall votes on music and speech signals. For example, the statistics show that one is 95% confident that between 48.2% and 54.5% of subjects favor block thresholding for music signals in the case of partial noise removal. These small confidence intervals, nonoverlapping in most cases, demonstrate the high reliability of this subjective evaluation and confirm the preference for block thresholding.

For musical sounds, one can explain the improvement of block thresholding over MMSE-LSA based algorithms as follow. For partial noise removal, the residual noise is more uniform, closer to a white noise and less "metallic" than the one obtained by LSA-DD. For maximum noise removal, block thresholding produces less musical noise than LSA-NC, and it results in less distortion on signal transients. With the Piano sound for instance, which corresponds to one of the highest vote in favor of block thresholding, the clavier stroke is much less muffled by block thresholding than by LSA-NC, due to its adaptive block size adjustment as explained in Section 2.3.3. These improvements are not significant enough for speech sounds (except for the partial noise removal of the male voice TIMIT-M for which the vote is clearly in favor of block thresholding) to lead to a clear distinction between the two algorithms.

Finally, the third step of the evaluation showed that maximum noise removal was most of the time preferred to partial noise removal. A little musical noise does not seem to be as annoying as a small residual noise. However, such preference is much stronger for musical sounds (99.2% v.s. 9.8%) than for speech sounds (71.7% v.s. 29.3%) for which intelligibility

and a clear articulation (i.e., clear transients) appear to be one of the main criteria.

Mozart	$\mathsf{W} = 16$	$\mathsf{W} = 8$	$\mathsf{W} = 4$	$\mathsf{W} = 2$	$\mathsf{W} = 1$
$\mathsf{L} = 8$	25.3	10.4	5.2	4.0	11.5
$\mathsf{L} = 4$	10.7	4.2	3.0	1.9	3.6
$\mathsf{L} = 2$	5.1	2.5	2.2	3.0	7.3
TIMIT-M	$\mathsf{W} = 16$	$\mathsf{W} = 8$	$\mathsf{W} = 4$	$\mathsf{W} = 2$	$\mathsf{W} = 1$
$\mathsf{L} = 8$	26.4	9.1	6.3	1.7	3.9
$\mathsf{L} = 4$	12.3	7.8	1.5	1.3	2.4
$\mathsf{L} = 2$	11.9	6.7	3.0	1.7	3.9

Table 2.4: Percentage of the different block size selected by the block thresholding algorithm for Mozart (top) and TIMIT-M (bottom).

The block size distribution presented in Table 2.4 shows the adaptivity of the block thresholding algorithm. The largest block size $\mathsf{L} \times \mathsf{W} = 8 \times 16$ is most frequently selected because it is optimal for large time-frequency regions where the signal energy is uniformly dominated by the noise energy. The blocks of size 8×1 having a narrow frequency width occur relatively often for musical signals such as Mozart recording because it matches their narrow frequency harmonics. On the contrary, the speech signal TIMIT-M privileges 2×16 blocks having a narrow time width because speech signals contain many short transients. As expected, the adaptive window size adjustment follows the signal time-frequency energy distribution properties.

Chapter 3

Image Denoising by Block Pursuit Thresholding

In Chapter 2, time-frequency block thresholding with rectangular blocks is introduced for audio noise removal. Block size is adjusted by the Stein risk estimator [185] to better fit the audio time-frequency properties. However, as block orientation is not adaptive, blocks cannot precisely adapt to image geometry.

This Chapter generalizes block thresholding by introducing a block pursuit procedure that calculates a covering of the image sparse representation coefficients with oriented blocks selected from a block dictionary appropriately designed. The selected blocks follow the image geometry. Block thresholding with these blocks reduces the block thresholding risk.

3.1 Introduction

Numerical images are always contaminated by noise, CCD noise for example. Image denoising aims at removing the noise while retaining the underlying image content.

Similar to audio time-frequency block denoising, block thresholding techniques have

been investigated for image denoising [126, 25, 30]. The authors proposed to group image wavelet coefficients in square blocks and apply block attenuation rules, and showed improvements over diagonal estimations [49]. Sparse image representations such as wavelet representations, on the other hand, contain geometrical structures much more complex than those in audio time-frequency representations that are mainly horizontal and vertical lines. In consequence, square or rectangular blocks are inadequate to fit image geometry, which increases the risk of the resulting block thresholding estimators. Oriented blocks adapted to image geometry are required for improvement. Calculating adaptive oriented blocks that fit image geometry is more difficult than adjusting the rectangular block size, which has been solved by using the Stein risk estimator [185] in audio time-frequency block denoising [219].

This Chapter generalizes the block thresholding by introducing a block pursuit procedure that calculates a covering of the image sparse representation coefficients with blocks selected from a block dictionary appropriately designed. For image denoising applications, the block pursuit algorithm calculates blocks that follow the image geometry. The resulting block pursuit thresholding improves the PSNR with respect to block thresholding.

We concentrate on coefficient processing as opposed to the choice of signal representations. Numerical experiments are performed with translation-invariant wavelets most commonly used in image denoising. Translation-invariant wavelet transform is recalled in Section 3.2. Section 3.3 reviews briefly the block thresholding estimators. The block pursuit algorithm and a fast implementation are described in Section 3.4. Image denoising by block pursuit thresholding is presented in Section 3.5.

3.2 Wavelet Representations

Wavelet dictionaries provide sparse representations for most natural images. A two-dimensional separable translation dyadic wavelet tight frames [126]

$$\{\psi^d_{j,u}, \phi_{J,u}\}_{1\le d\le 3,\ 1\le j\le J,\ 0\le u\le N-1}$$

is obtained by translating and dilating wavelet functions of three directions $\{\psi^d\}_{1\le d\le 3}$ and a scaling function ϕ, whose Fourier transform is shown in Figure 3.1:

$$\psi^d_{j,u}[n] = \frac{1}{2^j}\psi^d\left(\frac{n-u}{2^j}\right) \quad \text{and} \quad \phi_{J,u}[n] = \frac{1}{2^J}\phi\left(\frac{n-u}{2^J}\right).$$

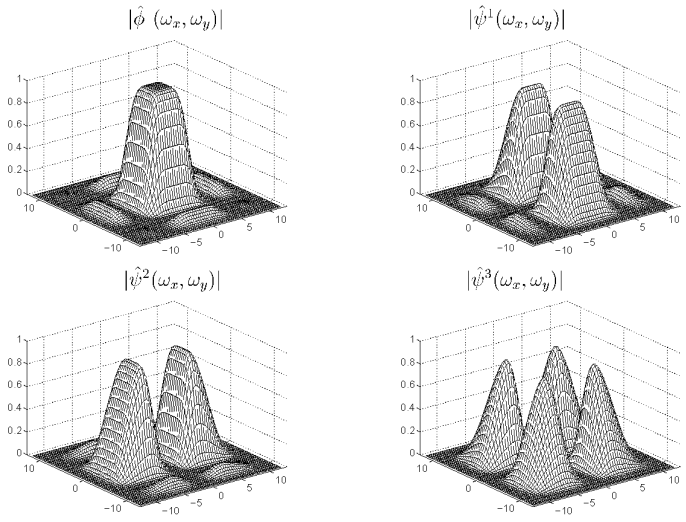


Figure 3.1: Fourier transform of a scaling function and 3 wavelet functions.

Figure 3.2 displays the wavelet coefficients F_d and the approximation coefficients F of an image f

$$F_{d,j}[u] = \langle f, \psi^d_{j,u}\rangle \quad \text{and} \quad F_J[u] = \langle f, \phi_{J,u}\rangle.$$

The wavelet image representation is sparse, as most wavelet coefficients are almost zero (in gray). A few large coefficients are concentrated along the contours.

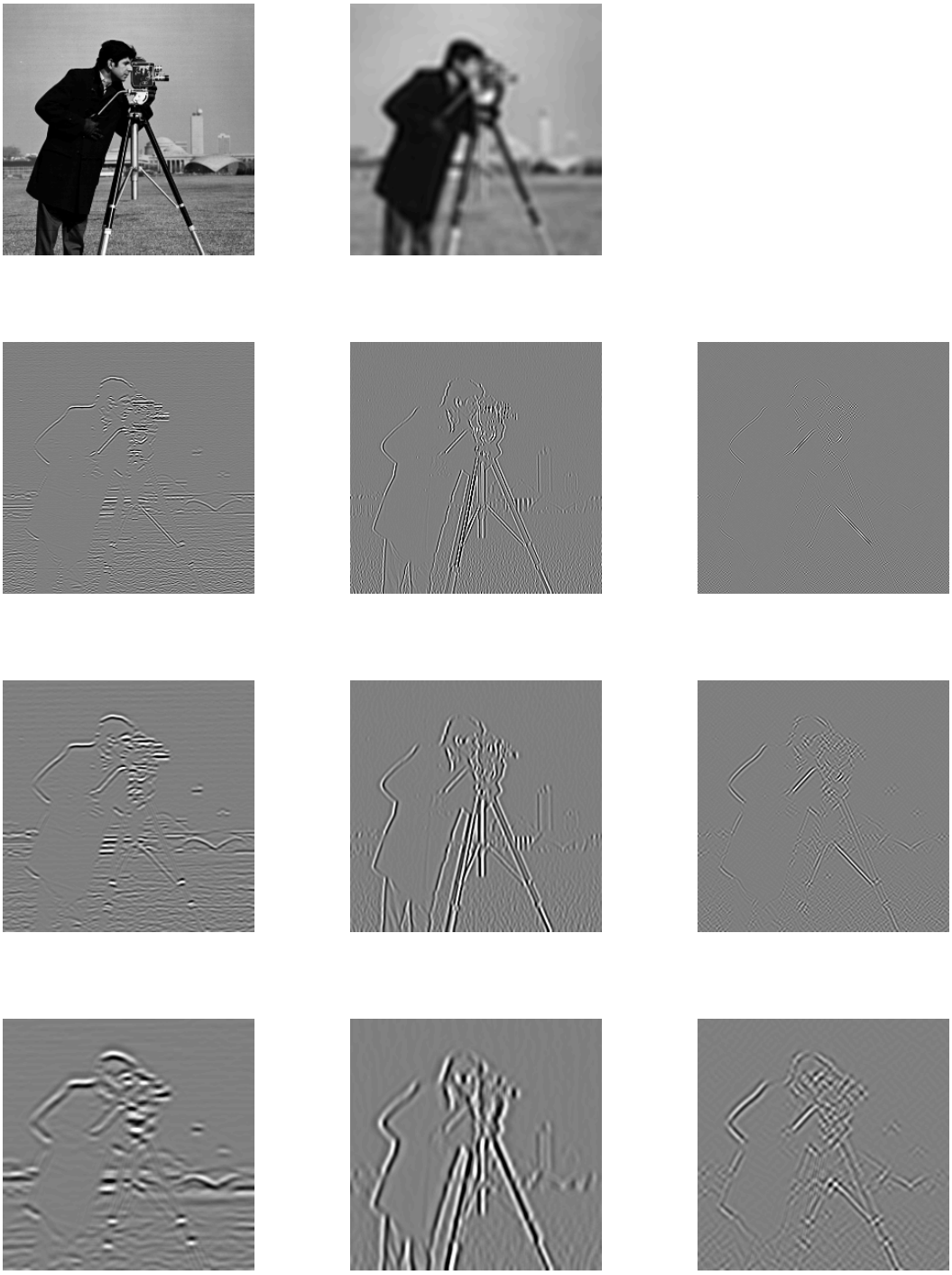


Figure 3.2: First row, from left to right: Image Cameraman and its third-scale translation-invariant approximation coefficients. From second to fourth row, from left to right. 1st, 2nd and 3rd translation-invariant wavelet coefficients at horizontal, vertical and diagonal directions. White and black gray-levels represent positive and negative values. Coefficients in gray are near zero.

The wavelet tight frame representation satisfies an energy conservation

$$\|f\|^2 = \frac{1}{A}\left(\sum_{d=1}^{3}\sum_{j=1}^{J}\sum_{u=0}^{N-1}|F_{d,j}[u]|^2 + \sum_{u=0}^{N-1}|F_J[u]|^2\right),$$

where $A \geq 1$ is the frame bound of the wavelet dictionary. This implies a simple reconstruction

$$f = \frac{1}{A}\left(\sum_{d=1}^{3}\sum_{j=1}^{J}\sum_{u=0}^{N-1}F_{d,j}[u]\psi^d_{j,u} + \sum_{u=0}^{N-1}F_J[u]\phi_{J,u}\right).$$

3.3 Block Thresholding

We recall the main formulas of the block thresholding estimators that have been described in Section 2.3. Let y be a noisy image that is the sum of a clean image f and a Gaussian noise w of zero mean: $y[n] = f[n] + w[n],\ n = 0, \ldots, N-1$. y is decomposed over a dictionary of vectors $\mathcal{D} = \{g_p\}_{p\in\Gamma}$ that is supposed to be a tight frame with frame bound A:

$$Y[p] = \langle y, g_p\rangle,$$

which induces the energy conservation

$$\|y\|^2 = \frac{1}{A}\|Y\|^2,$$

and the signal reconstruction

$$y = \frac{1}{A}\sum_{p\in\Gamma}Y[p]g_p.$$

The noise coefficient variance is denoted as

$$|\sigma[p]|^2 = E\{|\langle w, g_p\rangle|^2\}.$$

A block thresholding estimator [17, 18, 16] partitions the coefficients $Y[p]$ in I disjoint blocks B_i in which indices are grouped together, and multiplies all coefficients within each

B_i with a same attenuation factor a_i

$$\hat{f} = \frac{1}{A} \sum_{i=1}^{I} \sum_{p \in B_i} a_i \, Y[p] \, g_p, \tag{3.1}$$

where each a_i depends on all coefficients $Y[p]$ for $p \in B_i$.

Cai [17, 18] has introduced a soft block thresholding estimator

$$a_i = \left(1 - \frac{\lambda_s \, \|\sigma\|^2_{B_i}}{\|Y\|^2_{B_i}}\right)_+, \tag{3.2}$$

and a hard block thresholding estimator

$$a_i = \begin{cases} 1 & \text{if } \|Y\|^2_{B_i} \geq \lambda_h \|\sigma\|^2_{B_i} \\ 0 & \text{if } \|Y\|^2_{B_i} < \lambda_h \|\sigma\|^2_{B_i} \end{cases}. \tag{3.3}$$

where

$$\|Y\|^2_{B_i} = \sum_{p \in B_i} |Y[p]|^2 \quad \text{and} \quad \|\sigma\|^2_{B_i} = \sum_{p \in B_i} |\sigma[p]|^2.$$

are the empirical signal energy and the noise energy in the block B_i, λ_s and λ_h are soft and hard thresholding parameters. For Gaussian white noises, Cai has proved that the quadratic risk of block thresholding estimators is close to the oracle block thresholding risk lower bound [17].

As explained in Chapter 2, the block thresholding estimators have small risk if in each block B_i the coefficients are either far above or far below the noise amplitude

$$\forall p \in B_i, \ |F[p]| \gg \sigma[p] \quad \text{or} \quad \forall p \in B_i, \ |F[p]| \ll \sigma[p]. \tag{3.4}$$

This implies that to reduce the block thresholding risk, a block should not mix the coefficients that have signal-to-noise ratio (SNR) $\xi[p] = |F[p]|^2/\sigma^2[p] \gg 1$ and $\ll 1$. In other words, it requires that image structures and pure noise coefficients are separated in different blocks. The blocks should therefore fit signal geometry. While rectangular blocks approximate sufficiently well geometrical structures in audio signal spectrograms that are

mainly horizontal and vertical lines as shown in Chapter 2, they are not adequate to fit richer geometry in images. Figure 3.3 illustrates a simple image example with a slanted contour. Square and rectangular blocks necessarily go across the contour and therefore mixes coefficients of very different SNR. The block thresholding on these blocks degrades the contour and/or retains the noise along the contour. Blocks with orientation adapted to the image geometry are required for improvement.

3.4 Block Pursuits

Images have geometrical regularity. In sparse wavelet representations, large coefficients concentrate along the contours. As illustrated in Figure 3.4, oriented blocks of different orientations are required to fit the image geometry, so that (3.4) can be satisfied which leads to a small block thresholding risk.

More generally, although we work in sparse image representations, coefficients are not completely decorrelated and present some prior structures. To reduce the block thresholding risk, a dictionary $\mathcal{D}_B$ of blocks are constructed to cover the large coefficients on the prior structures: Each block is set of points whose shape may fit some part of the prior structures. Covering the large coefficients with a dictionary of blocks is a set covering problem [45]. A greedy block pursuit algorithm is introduced to calculate this set covering with blocks of arbitrary shapes.

3.4.1 Block Pursuit Algorithm

The block pursuit algorithm is a greedy procedure that iteratively selects the blocks from a dictionary $\mathcal{D}_B = \{B_k\}_k$ one by one in a decreasing order according to the block energy. All blocks in the dictionary have the same size $B^{\#}$ so that the they generate comparable energy.

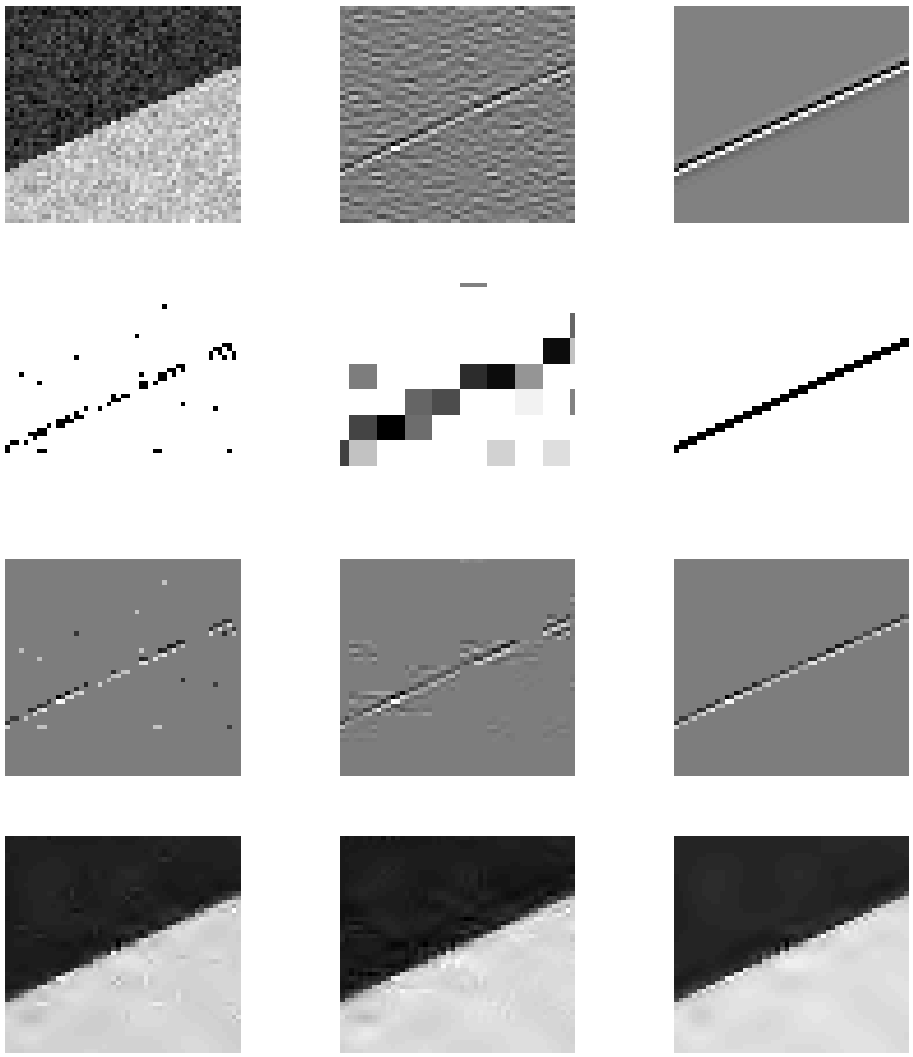


Figure 3.3: From left to right. First row: noisy image y, translation-invariant wavelet coefficients (1st scale, horizontal band) of noisy image Y and of clean image F. Second row: attenuation factors of hard thresholding (HT), block thresholding (BT) and block pursuit thresholding (BPT). Gray-level from white to black: value from 0 to 1. Third row: denoised wavelet coefficients (1st scale, horizontal band) by HT, BT and BPT. Fourth row: denoised image $\hat{f}$ with HT (39.50 dB), BT (39.39 dB) and BPT (40.76 dB). BPT adapts the image geometry and therefore restores better the contour and removes more efficiently the noise along the contour.

Figure 3.4: Adaptive oriented blocks are required to fit the image geometry.

The block pursuit algorithm is initialized with $R^0Y = Y$. At the iteration m, it selects the block that yields the largest energy

$$\|R^m Y\|^2_{B_{k_m}} = \max_k \|R^m Y\|^2_{B_k}, \tag{3.5}$$

where

$$\|R^m Y\|^2_{B_k} = \sum_{p \in B_k} |R^m Y[p]|^2, \quad \forall k \tag{3.6}$$

is the energy of the coefficients in the block B_k. The non-zero coefficients in the selected block B_{k_m} are set to zero:

$$R^{m+1}Y[p] = \begin{cases} 0 & \text{if } p \in B_{k_m} \text{ and } R^m Y[p] \neq 0 \\ R^m Y[p] & \text{otherwise} \end{cases}. \tag{3.7}$$

Inserting (3.7) in (3.6) yields

$$\|R^{m+1}Y\|^2_{B_k} = \|R^m Y\|^2_{B_k} - \sum_{p \in B_k \cap B_{k_m}} |R^m Y[p]|^2, \quad \forall k. \tag{3.8}$$

Block pursuit converges in $M \leq N$ iterations as in each iteration at least one coefficient is set to 0.

For denoising applications, the block pursuit procedure stops when the residual coefficients $R^{M-1}Y$ have energy comparable to noise. The coefficients covered by the selected blocks $\{B_{k_m}\}_{0\leq m<M}$ correspond to image geometry and are thus retained, which amounts to a hard block thresholding (3.3). Assume that the noise is Gaussian white with variance σ^2. The hard block thresholding (3.3) implies that the block pursuit procedure should stop at the M-th iteration when

$$\|R^{M-1}Y\|^2_{B_{k_{M-1}}} \leq \lambda_h B^{\#}_{k_{M-1}} \sigma^2, \tag{3.9}$$

where $B^{\#}_{k_{M-1}}$ is the number of non-zero coefficients $R^{M-1}Y$ in $B_{k_{M-1}}$ and λ_h is the hard block thresholding parameter. The threshold λ_h trades off between noise removal and signal restoration. Indeed when the underlying signal coefficients are zero, the block energy $\|R^{M-1}Y\|^2_{B_{k_{M-1}}}$ follows approximately a χ^2 distribution with $B^{\#}_{k_{M-1}}$ degrees of freedom. Increasing λ_h reduces the probability to keep the residual noise, but may result in the removal of the coefficients due to the image geometry. The coefficients that are not covered by the selected blocks have energy comparable to noise and are attenuated with the more conservative standard soft block thresholding (3.2).

Observe that if the transform coefficients $Y[p] = \langle y, \phi_p \rangle$ are calculated with an orthogonal basis $\mathcal{D} = \{g_p\}_{p=1,\dots,N}$, then the block pursuit can be interpreted as an orthogonal space matching pursuit that decomposes y with a dictionary of vector spaces $\mathcal{D}_{\mathbf{W}} = \{\mathbf{W}_k\}_k$, where the vector space $\mathbf{W}_k$ is generated by a family of vectors $\{g_p\}_{p\in B_k}$.

3.4.2 Fast Implementation

To calculate the complexity of the block pursuit algorithm, let us define $L[p]$ the number of blocks that cover the coefficient $Y[p]$. As each block covers $B^{\#}$ coefficients and in total $\sum_{p=1}^{N} L[p]$ coefficient-times are covered, the number of blocks in the dictionary $\mathcal{D}_B$

is $K = \sum_{p=1}^{N} L[p]/B^{\#}$. Observe that the block energy update (3.8) that has complexity $\mathcal{O}\left(\sum_{p\in B_{k_m}, R^m Y[p]\neq 0} L[p]\right)$ is more computationally efficient than recalculating the block energy with (3.6) in $\mathcal{O}\left(\sum_{p=1}^{N} L[p]\right)$. The block pursuit procedure is thus initialized with (3.6) and calculates at each iteration with two types of operations: one maximum operation (3.5) and $\sum_{p\in B_{k_m}, R^m Y[p]\neq 0} L[p]$ times data access for the energy update (3.8).

The block pursuit procedure can be implemented with different data structures. We will see that a heap data structure provides faster implementation than conventional unordered data array.

With an unordered data array of size K, calculating the maximum element requires complexity $\mathcal{O}(K)$ and reading and writing an element has complexity $\mathcal{O}(1)$. Therefore for one iteration, (3.5) and (3.8) have respectively complexity $\mathcal{O}(K)$ and $\mathcal{O}\left(\sum_{p\in B_{k_m}, R^m Y[p]\neq 0} L[p]\right)$. The part with $\mathcal{O}(K)$ due to the maximum operation (3.5) typically dominates the complexity. To see this, let us take a typical example where all the points are covered by a constant number of $L[p] = L$ blocks. As the image size N is typically bigger than the square of block size $B^{\#}$, $\mathcal{O}(K) = \mathcal{O}(NL/B^{\#}) > \mathcal{O}\left(\sum_{p\in B_{k_m}, R^m Y[p]\neq 0} L[p]\right) = \mathcal{O}(LB^{\#})$. The complexity per iteration of the unordered array implementation is thus $\mathcal{O}(K)$.

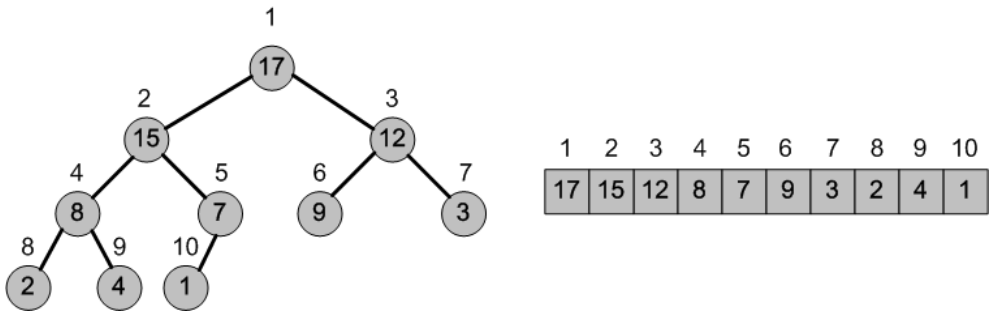


Figure 3.5: A heap viewed as a binary tree (left) and an array (right). The number within the circle at each node in the tree is the value stored in the node. The number next to the node is the corresponding index in the array. This figure is adapted from [42].

A heap is a binary tree data structure that satisfies the *heap property*: For each node

other than the root, its value is at most the value of its parent [42]. Figure 3.5 shows a heap example. A heap allows to manage information during the execution of the algorithm [42]. A heap of size K is transferred from an unordered array with complexity $\mathcal{O}(K)$. It returns the maximum element fast in $\mathcal{O}(1)$, but requires longer time $\mathcal{O}(\log_2 K)$ to insert and delete an element in order to maintain the heap structure [42]. Using the heap structure, the complexity of block pursuit is dominated by the energy update (3.8) which is in $\mathcal{O}\left((\log_2 K)\sum_{p\in B_{k_m}, R^m Y[p]\neq 0} L[p]\right)$ per iteration. With $L[p] = L$, the complexity for the first iteration is $\mathcal{O}\left((\log_2 K)LB^{\#}\right)$. Since more and more residual coefficients $R^m Y[p]$ become zero, the complexity per iteration decreases when the number of iteration increases.

Let us compare the complexity of the unordered array and the heap implementations. The initialization (3.6) is common for the two and is calculated with complexity $\mathcal{O}\left(\sum_{p=1}^{N} L[p]\right) = \mathcal{O}\left(B^{\#}K\right)$. The unordered array implementation requires $\mathcal{O}(MK)$ for M iterations. As the block size $B^{\#}$ is typically much smaller than the number M of iterations, the overall complexity of the unordered array implementation for M iterations is $\mathcal{O}(MK)$. In worse case when $M = N$, its complexity is $\mathcal{O}(NK)$. On the other hand, the heap implementation requires $\mathcal{O}(K)$ to build the heap and its complexity for M iterations is upper bounded by $\mathcal{O}\left((\log_2 K)MLB^{\#}\right)$. Since $K = NL/B^{\#}$ and typically $N > (B^{\#})^2$, the heap implementation is typically faster than the unordered array implementation. As $\mathcal{O}\left((\log_2 K)\sum_{m=0}^{M-1}\sum_{p\in B_{k_m}, R^m Y[p]\neq 0} L[p]\right)$ is upper bounded by $\mathcal{O}\left((\log_2 K)\sum_{p=1}^{N} L[p]\right) = \mathcal{O}\left((\log_2 K)B^{\#}K\right)$, which dominates the heap building and is in the same order as the initialization operations up to a $\log_2 K$ factor, the worse-case overall complexity of the heap implementation is $\mathcal{O}\left((\log_2 K)B^{\#}K\right)$.

3.5 Image Denoising by Block Pursuit Thresholding

The block pursuit thresholding denoising is applied in translation-invariant wavelet representations described in Section 3.2. As illustrated in Figure 3.2, sparse wavelet image representations contain some geometry: Large coefficients concentrate along the contours. In order to identify the geometry, we construct a dictionary $\mathcal{D}_B = \{B_k\}_k$ that contains elongated blocks with orientations uniformly sampled in $[0, \pi)$, some examples being illustrated in Figure 3.6. Each block in the dictionary may locally fit the image geometry. The blocks are translated to cover the whole image plane. A same size is imposed on each block so that no one is privileged over the others.

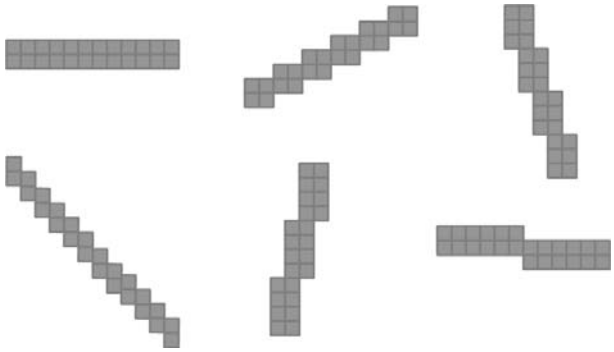


Figure 3.6: Examples of oriented blocks.

As the wavelet coefficients along the contour dilate by a factor of 2^j in the direction orthogonal to the contour when the wavelet scale j goes higher, the width of the oriented blocks are set proportional to 2^j. On a contour, first-scale translation-invariant wavelet coefficients have at least a width of 2 pixels. The block width is thus $W_2 = 2$ in the first scale. In the experiments blocks of size $W_1 \times W_2 = 12 \times 2^j$ are used.

The hard block pursuit threshold in (3.9) is set equal to $\lambda_h = 6$ which results in general the highest PSNR. Comparing to the threshold value $3^2 = 9$ that is often used in diagonal

hard thresholding, a smaller threshold allows to restore better the image geometry. Blocks calculated with a smaller λ_h tends to fit the noise and thus decreases the PSNR. The block pursuit residual coefficients have comparable energy to noise and are partitioned into square blocks of size $W_s \times W_s$ with $W_s = 3 \times 2^j$, where j is the wavelet scale, and are attenuated with the soft block thresholding (3.2) with the threshold $\lambda_s = 1.5$, which has been shown good for image block thresholding denoising [126].

In the experiments images are contaminated by Gaussian white noise of different amplitude. The hard thresholding used the threshold 3σ and the (soft) block thresholding are configured with block size $W_s \times W_s$ where $W_s = 3 \times 2^j$, where j is the wavelet scale, and the threshold $\lambda_s = 1.5$.

Figure 3.7 illustrates a denoising example on the image Cameraman. Block thresholding improves 0.2 dB with respect to hard thresholding while the block pursuit thresholding gains 0.4 dB over block thresholding. The block pursuit thresholding fits the image geometry much more precisely than the block thresholding that has block-wise constant attenuation factors, and therefore restores sharper contours and removes more efficiently the noise. Compared with hard thresholding, block pursuit thresholding is more regular and restores better image details.

Table 3.1 compares the block pursuit thresholding with hard thresholding and block thresholding on various images. Block pursuit thresholding improves on average 0.6 dB PSNR with respect to hard thresholding. The gain of block pursuit thresholding over block thresholding, about 0.3 dB on average, is of the same order as that of block thresholding over hard thresholding.

Let us notice that rather than optimizing the image representations [44, 123, 127, 184, 128, 14] we concentrate on improving the thresholding techniques. Donoho and Johnstone [49] have shown that hard thresholding risk is upper bounded by $2 \log_e N$ times the

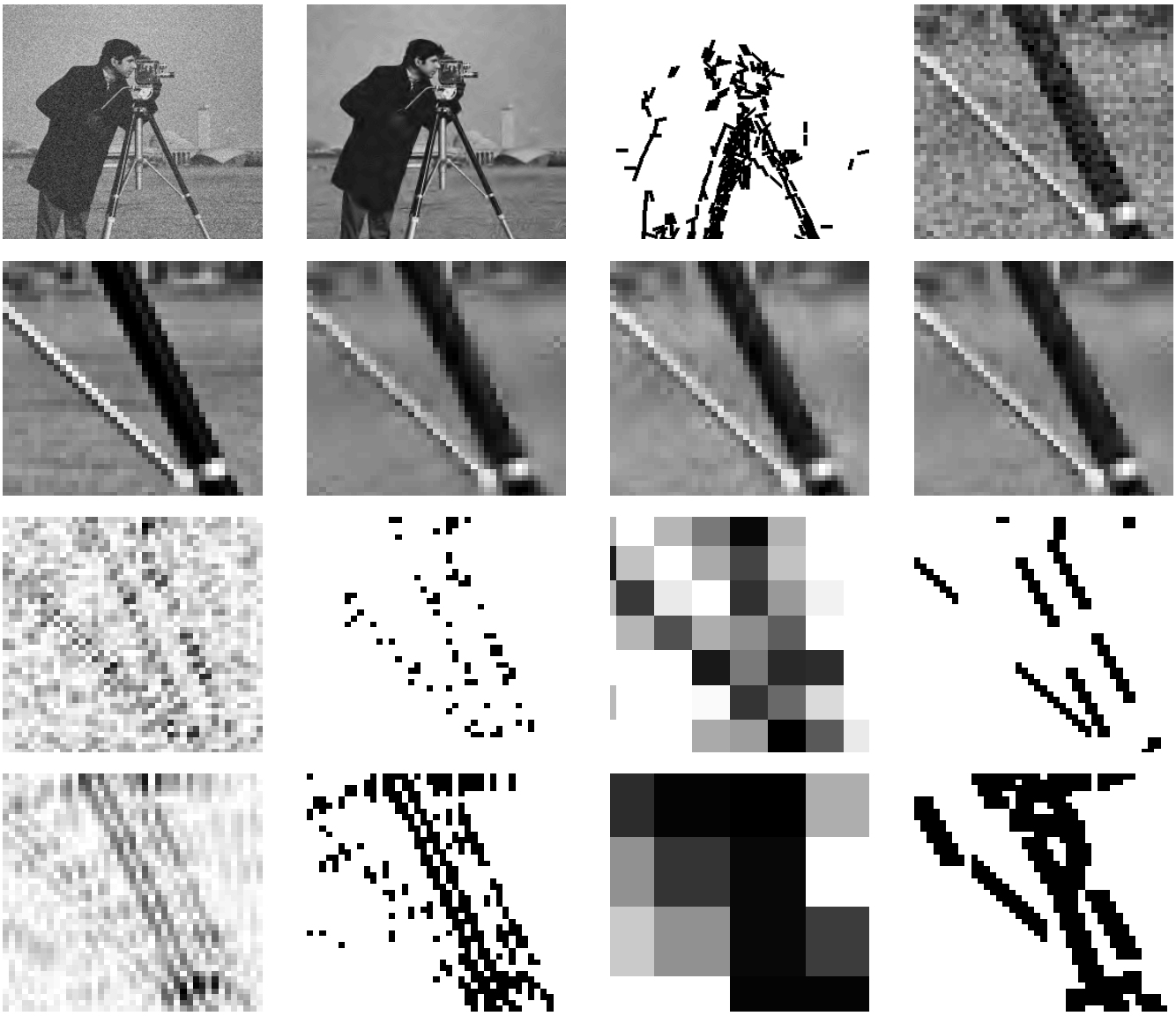


Figure 3.7: From left to right. First row: noisy image Cameraman with $\sigma = 20$, denoised image by block pursuit thresholding (BPT) 29.15 dB, adaptive oriented blocks on the second-scale wavelet vertical band, zoom of noisy image. Second row: zoom of clean image, of image denoised by hard thresholding (HT) 28.53 dB (on the whole image), block thresholding (BT) 28.75 dB, and BPT 29.15 dB. Third row: zoom of noisy wavelet coefficient modules in the first-scale wavelet vertical directions, corresponding HT attenuation factors, BT attenuation factors, BPT adaptive oriented blocks. Fourth row: the same as the third row in the second wavelet scale.

oracle diagonal estimation risk. As block thresholding risk is upper bounded by the oracle block thresholding risk, which is itself upper bounded by the oracle diagonal risk, its gain over hard thresholding can be no larger than a limited factor $2\log_e N$. With the same translation-invariant wavelet representation, the about 0.6 dB block pursuit thresholding with respect to hard thresholding is pretty important. Further PSNR improvement is possible through optimizing the image representations by retransforming the coefficients in the blocks once the blocks are calculated.

Another factor that prevents the block pursuit thresholding from achieving a more substantial PSNR improvement is the limitation of the block shapes in the dictionary. In order to approach the oracle diagonal estimation risk, the blocks should always group the coefficients that have either large or small SNR. This requires that the dictionary contains blocks with rich enough shapes. In numerical computation, however, the number of block shapes is limited for computational efficiency consideration.

In the next Chapter, block pursuit is applied in image super-resolution, where capturing directional geometry information is essential for restoring sharper images and reducing artifacts. Super-resolution with block pursuits will be shown to achieve significant improvements with respect to cubic-spline interpolation.

Cameraman 256 × 256

σ	5	10	15	20	25	30	35	40	Avg. Impr.
PSNR	34.16	28.13	24.61	22.11	20.16	18.59	17.28	16.10	w.r.t. HT
HT	36.94	32.51	30.10	28.53	27.36	26.40	25.64	25.02	
BT	37.45	32.89	30.38	28.75	27.59	26.58	25.87	25.21	**+ 0.28**
BPT	37.62	33.19	30.77	29.15	27.98	26.97	26.20	25.57	**+ 0.62**

House 256 × 256

σ	5	10	15	20	25	30	35	40	Avg. Impr.
PSNR	34.16	28.13	24.61	22.11	20.16	18.59	17.28	16.10	w.r.t. HT
HT	37.43	34.26	32.45	31.09	30.07	28.92	28.19	27.57	
BT	38.33	34.63	32.65	31.32	30.27	29.17	28.48	27.88	**+ 0.34**
BPT	38.36	34.89	33.00	31.70	30.60	29.52	28.72	28.11	**+ 0.61**

Peppers 512 × 512

σ	5	10	15	20	25	30	35	40	Avg. Impr.
PSNR	34.16	28.13	24.61	22.11	20.16	18.59	17.28	16.10	w.r.t. HT
HT	36.57	33.99	32.45	31.16	30.09	29.17	28.32	27.67	
BT	37.16	34.17	32.51	31.25	30.20	29.30	28.51	27.91	**+0.20**
BPT	37.19	34.43	32.86	31.66	30.60	29.71	28.90	28.25	**+0.52**

Lena 512 × 512

σ	5	10	15	20	25	30	35	40	Avg. Impr.
PSNR	34.16	28.13	24.61	22.11	20.16	18.59	17.28	16.10	w.r.t. HT
HT	37.61	34.55	32.64	31.33	30.20	29.30	28.51	27.75	
BT	38.16	34.98	33.01	31.70	30.57	29.73	28.93	28.24	**+0.43**
BPT	38.20	35.16	33.24	31.96	30.81	29.94	29.13	28.38	**+0.61**

Boat 512 × 512

σ	5	10	15	20	25	30	35	40	Avg. Impr.
PSNR	34.16	28.13	24.61	22.11	20.16	18.59	17.28	16.10	w.r.t. HT
HT	35.95	32.56	30.64	29.24	28.13	27.26	26.58	25.95	
BT	36.70	33.01	31.04	29.62	28.50	27.60	26.94	26.29	**+0.42**
BPT	36.73	33.18	31.27	29.86	28.75	27.84	27.16	26.48	**+0.62**

Table 3.1: Comparison of PSNR over hard thresholding (HT), block thresholding (BT) and block pursuit thresholding (BPT). The far right column in each table shows the average PSNR improvement of BT and BPT with respect to HT.

Chapter 4

Sparse Super-Resolution by Block Pursuits

Super-resolution image interpolation requires to identify image geometric regularity. Block pursuit procedure identifies geometric image model in sparse representations. The block pursuit algorithm projects sparse transform coefficients over structured vector spaces instead of individual vectors and regularizes the sparse decomposition. A super-resolution image zooming is derived. Numerical experiments illustrate the efficiency of the proposed super-resolution procedure compared to cubic spline interpolations.

4.1 Introduction

Zooming operators that increase the size of images are often needed for digital display of images or videos. When images are aliased, linear interpolations [187] introduce artifacts such as Gibbs oscillations or zigzag along edges, and restore a blurred image. Better images can be estimated with super-resolution procedures which take advantage of this aliasing together with some geometric image properties.

A super-resolution algorithm computes a signal estimation in a space of dimension larger than the input signal size [126]. Super-resolution algorithms are necessarily non-linear and can recover high frequency information by taking advantage of prior signal information. A large body of super-resolution literature relies on a sequence of low-resolution images or a training process to reconstruct a high-resolution image (see for example [61, 68, 50]). Applications of these methods are restricted when the only relevant data available is a single low-resolution of interest, or if the memory resource is limited.

Single image super-resolution zooming is more difficult but is possible by interpolating the image along directions for which it is geometrically regular. Directional interpolations, usually known as edge-directed or content-adaptive interpolation, interpolate along directions that are computed with ad-hoc directional regularity estimations [105, 206, 31]. These algorithms are used in industry with good numerical results.

If a signal has a sparse representation in a dictionary then a super-resolution estimation may be computed from lower-resolution measurements [80, 196, 156], and reliable recovery requires that the dictionary is sufficiently *incoherent*. This approach has been used successfully for seismic sparse spike inversion or image inpainting [51, 58, 126]. Geometrically regular images have a sparse representation in curvelet [20, 19] or bandlet [99, 98, 129] dictionaries. However, subsampling a curvelet or a bandlet dictionary does not define a sufficiently incoherent dictionary to recover sparse super-resolution estimations for image zooming. Recovering these vectors individually without constraint from a subsampled signal requires a full search in a large dictionaries which leads to errors.

This Chapter introduces a super-resolution algorithm which computes structured sparse representations by projecting the image wavelet coefficients over selected subspaces and then making linear approximation in the selected subspaces. For super-resolution interpolation, this algorithm takes better advantage of prior image information by selecting vector spaces

as opposed to individual vectors, which also reduces the computational complexity. These vector spaces are selected with a cascade of block pursuit procedures described in Chapter 3. Directional interpolations derived from this sparse representation yields a super-resolution image estimation.

The Chapter first relates directional interpolations to sparse super-resolution image zooming and reviews sparse super-resolution approaches. A novel directional interpolator is introduced in Section 4.2.2. Section 4.3 introduces structured sparse representations by selecting vector spaces with block pursuit algorithms. A super-resolution interpolation is derived, and numerical experiments provide comparisons with cubic spline interpolations.

4.2 Directional Interpolation and Sparsity

Image super-resolution zooming is possible by interpolating the image along directions for which it is geometrically regular. Sparse super-resolution algorithms identifies geometrical regularity by decomposing the image with a curvelet or bandlet dictionary and interpolates the image along the directions of selected vectors.

4.2.1 Directional Interpolation

Let $f[n]$ be a high-resolution image whose frequency support is in $[-\pi, \pi]^2$. The measured low-resolution image y is obtained from a subsampling of f

$$y[n_1, n_2] = f[Kn_1, Kn_2] + w[n_1, n_2], \tag{4.1}$$

where w models the noise and $K > 1$ is the subsampling factor. In the following we assume $K = 2$ for simplicity. An image zooming computes an estimate $\tilde{f}$ of f from y.

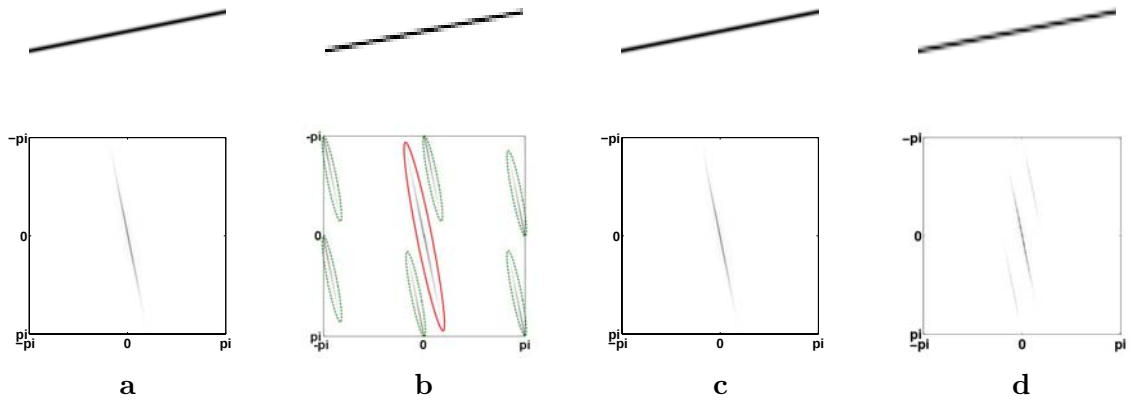


Figure 4.1: (a): high-resolution edge image $f[n]$ and its Fourier transform. (b): a low-resolution image $y[n] = f[2n]$ and its aliased Fourier transform. The spectrum component circled by the ellipse in the center corresponds to the edge. The other components circled by the dashed ellipse are due to the aliasing. (c): super-resolution estimation with directional interpolation along the contour direction (62.42 dB). (d): 2D cubic-spline interpolation (43.37 dB).

If $w = 0$, the Fourier transform $\hat{y}$ of y is related to the Fourier transform $\hat{f}$ of f by

$$\hat{y}(\omega_1, \omega_2) = \sum_{k_1=0,1} \sum_{k_2=0,1} \hat{f}(\omega_1 + k_1\pi, \omega_2 + k_2\pi). \tag{4.2}$$

If y is free of aliasing, i.e., if the frequency support of f is included in $[-\pi/2, \pi/2]^2$ then a perfect reconstruction $\tilde{f} = f$ is obtained with a 2D linear sinc interpolation, which is a low-pass filtering:

$$\hat{\tilde{f}}(\omega_1, \omega_2) = \hat{y}(\omega_1, \omega_2)\, \hat{h}(\omega_1, \omega_2) = \hat{f}(\omega_1, \omega_2), \tag{4.3}$$

with

$$\hat{h}(\omega_1, \omega_2) = \begin{cases} 1 & (\omega_1, \omega_2) \in [-\pi/2, \pi/2]^2 \\ 0 & \text{otherwise} \end{cases}. \tag{4.4}$$

If the frequency support of $\hat{f}$ exceeds $[-\pi/2, \pi/2]^2$, which is most often the case then y is aliased and (4.3) does no hold anymore. Any 2D linear interpolation introduces errors that result in artifacts such as Gibbs oscillations, blur and zigzag patterns along contours,

as shown in Figure 4.1(d). The ideal low-pass filter (4.4) is generally replaced by a cubic spline interpolation filter which reduces artifacts.

If the image f has some directional regularity, it is possible to improve this estimation and recover signal frequencies higher than $\pi/2$ by interpolating the image in the appropriate direction. Indeed, if the image is locally regular in a direction θ then its local Fourier transform has a narrow low-pass frequency support along this direction θ. Figure 4.1(a) gives a simple illustration with a straight edge of direction θ whose Fourier transform is elongated in the direction $\theta + \pi/2$. As shown in Figure 4.1(b), the spectrum of the subsampled image y is aliased, but the aliased parts circled by the dashed ellipses do not overlap the main component due to the spectrum of f circled by the big ellipse in the center. Observe that if the low-pass filter h is replaced by a directional interpolator h_θ whose Fourier transform has a support that includes the main component and vanishes at the aliased parts, then it recovers an estimation $\tilde{f}$ whose Fourier transform satisfies

$$\hat{\tilde{f}}(\omega_1, \omega_2) = \hat{y}(\omega_1, \omega_2)\, \hat{h}_\theta(\omega_1, \omega_2) \approx \hat{f}(\omega_1, \omega_2), \tag{4.5}$$

and thus achieves an almost perfect reconstruction by eliminating the aliased components, as shown in Figure 4.1(c). If $\theta = 0$ or $\theta = \pi/2$ then the support of $\hat{f}(\omega_1, \omega_2)$ overlaps its aliased components and it is therefore impossible to separate them with a filter. In this case, no super-resolution is possible and the interpolation is implemented with a low-pass filter h.

Adaptive interpolation algorithms finds locally if there exists a direction θ along which the image variations are more regular than other directions, in which case it performs the interpolation in this direction with the interpolator I_θ. For complex images, measuring the "best direction of regularity" is difficult. Ad-hoc algorithms have been developed to do so and are used in industry for non-linear image zooming.

4.2.2 Directional Interpolator

As illustrated in Figure 4.1(b), to restore the directional structure and remove the aliasing from a low-resolution image requires a directional interpolator I_θ whose Fourier transform support includes the main component and vanishes at the aliased parts. An ideal directional low-pass filter whose Fourier transform is an indicator function with an elongated support oriented along $\theta + \pi/2$ (which corresponds to the central ellipse in Figure 4.1(b)) satisfies this condition, but it creates Gibbs oscillation artifact due to its infinite support in space. On the other hand, the directional interpolator should have high enough order, cubic spline as opposed to the first order linear interpolation for example, to achieve more precise recovery. In addition, for fast implementation, the direction interpolator should be separable. [206] describes a directional bilinear interpolator that generalizes the standard bilinear interpolation, but extending it to higher order interpolation is indirect.

The proposed separable directional interpolator is factorized in three steps with each step a one-directional interpolation and it fully takes advantage of directional regularity along θ. Interpolating along θ implies that the underlying filter is low-pass in θ and elongated along $\theta + \pi/2$ in Fourier. The order of the resulting directional interpolator can be easily adjusted by changing one dimensional interpolation kernel.

Multiplying by 2 the image rows and columns requires computing interpolations along two directions. A one-dimensional interpolation along θ provides accurate coefficient estimations if the image is regular in the direction of θ. The mid-point between any two points having an angle θ is calculated with such an interpolation. This will oversample by a factor two either the image rows, or the image columns or the diagonals of angle $\pm\pi/4$. Along these oversampled rows, columns or diagonals, one can now compute a cubic spline interpolation with little aliasing error. These directions are interpolated to further increase the number of samples. The position of these new samples are chosen so that any missing

upscaled image coefficient is in the middle of two new samples having a relative position of angle θ. These missing image coefficients are then computed with an interpolation along θ from new samples. This last interpolation along θ is precise as the image is regular in the direction of θ.

Figure 4.2 illustrates this interpolation procedure with an example for $\theta = \arctan 1/2$. The subsampled grid shown with $\times$. The algorithm is decomposed in three steps.

1. Computation of the mid-point $\bigcirc$ between any two pair of samples having an angle θ, which are located along image columns for $\theta = \arctan 1/2$.

2. Interpolation along the oversampled image columns to compute new sample values $\bullet$.

3. Computation of the up-scaled image values at positions $\square$ by interpolating the new samples $\bullet$ along θ.

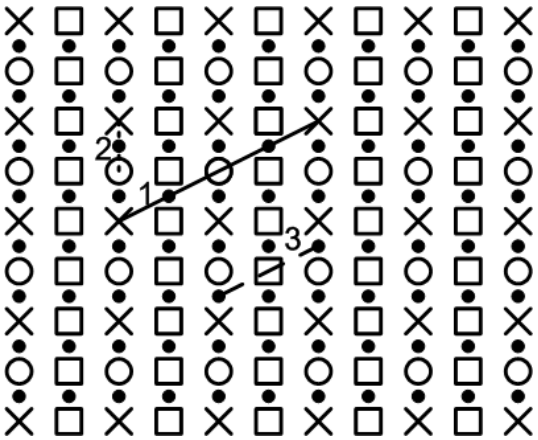


Figure 4.2: Directional interpolation in three steps:a one-dimensional interpolations along angle θ, a vertical interpolation and another interpolation along θ.

Figure 4.1(c) shows a directional interpolation example. With the cubic spline kernel, the proposed directional interpolator achieves almost perfect reconstruction with 62.42 dB PSNR, which is about 20 dB higher than the standard cubic spline interpolation. High

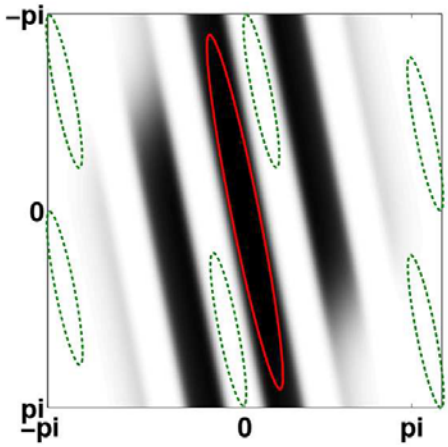


Figure 4.3: Spectrum of the directional interpolator I_θ. Gray-level from black to white: value from 1 to 0. Compare with Figure 4.1(b): The ellipse in the center corresponds the spectrum component of the edge that will be retained; the other dashed ellipse correspond to the spectrum components due to the aliasing that will be eliminated.

frequency information is restored along the contour and the result is free of zigzag artifact. Changing the directional interpolation kernel to the first order linear interpolation degrades about 6 dB PSNR with respect to that with the cubic spline, which shows the importance of using a higher order kernel.

A comparison between the spectrum of the directional interpolator I_θ shown in Figure 4.3 and Figure 4.1(b) confirms that I_θ allows to cancel the aliasing and restore the contour that is regular along the direction θ.

4.2.3 From Directional Interpolation to Sparsity

If a signal f has some directional geometrical regularity then it has a sparse representation in a dictionary $\mathcal{D} = \{g_p\}_{p\in\Gamma}$ of curvelets [20, 19] or bandlets [99, 98, 129]. Finding an appropriate direction of interpolation can be connected to sparse super-resolution estimation, although we shall see that this sparse super-resolution estimation may not perform well for image interpolation. In the following we shall concentrate on a dictionary of curvelets

but the same conclusions apply to a bandlet dictionary. A curvelet is an elongated oscillatory waveform whose Fourier transform is concentrated along a particular direction in the Fourier plane, as illustrated in Figure 4.4. If f is an image with contours that are geometrically regular then it has a sparse representation in a curvelet dictionary. The signal f can thus be approximated by a small number $|\Lambda|$ of curvelets in a set Λ which specifies the directions, scales and positions of these curvelets

$$f = f_\Lambda + w_\Lambda \quad \text{with} \quad f_\Lambda = \sum_{p \in \Lambda} a[p]\, g_p. \tag{4.6}$$

The approximation error w_Λ has a relatively small norm.

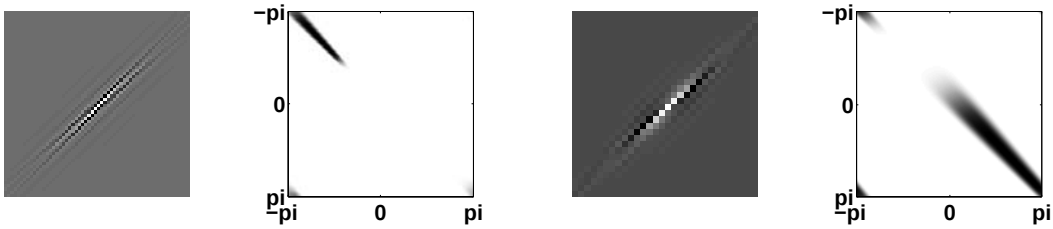


Figure 4.4: (a): curvelet $g_p[n]$. (b): Fourier transform $\hat{g}_p(\omega)$. (c): subsampled curvelet Ug_p. (d): Aliased fourier transform $\widehat{Ug}_p(\omega)$.

Let us denote $y = Uf + w$ the low-resolution measurements where U is a subsampling operator. An estimation of f from y is computed by calculating an estimation of the sparse approximation support Λ of f in $\mathcal{D}$ [126] [51]. This is done by observing that y has a sparse representation with this same approximation support

$$y = \sum_{p \in \Lambda} a[p]\, U g_p + w' \quad \text{with} \quad w' = U w_\Lambda + w, \tag{4.7}$$

in the subsampled dictionary

$$\mathcal{D}_U = \{U g_p\}_{p \in \Gamma}. \tag{4.8}$$

The approximation support Λ is estimated by computing a sparse approximation $\tilde{y}$ of y in

$\mathcal{D}_U$

$$\tilde{y} = \sum_{p \in \tilde{\Lambda}} \tilde{a}[p]\, U g_p \ , \tag{4.9}$$

with an $\mathbf{l}^1$ pursuit [28] or a matching pursuit [126]. A super-resolution estimation $\tilde{f}$ of f is derived by inverting U on the decomposition (4.9) of $\tilde{y}$. This is equivalent to applying an interpolation operator I_p to each subsampled Ug_p to recover g_p:

$$\tilde{f} = \sum_{p \in \tilde{\Lambda}} \tilde{a}[p]\, I_p(U g_p) = \sum_{p \in \tilde{\Lambda}} \tilde{a}[p]\, g_p \ . \tag{4.10}$$

This estimation can thus be interpreted as an adaptive interpolation of each Ug_p, and the adaptive interpolation is performed along the curvelet direction to recover the curvelet g_p. The estimated curvelet support $\tilde{\Lambda}$ defines the directions and supports of the interpolators that are used to compute $\tilde{f}$ from y.

The directional interpolation $\tilde{f}$ is an accurate estimation of f if the estimated support $\tilde{\Lambda}$ providing the regularity directions is an appropriate estimation of the approximation support Λ of f. The work of Tropp [189] shows that such a recovery is possible if the vectors in the transformed dictionary $\mathcal{D}_U = \{Ug_p\}_{p \in \Gamma}$ are highly incoherent. However this condition does not hold for a dictionary obtained by subsampling curvelets on a uniform subgrid. Indeed, the finest scale curvelets g_p, which are responsible for high-frequency information restoration, when subsampled by the operator U, have no more vanishing moment and a relatively large energy at low frequencies. Therefore they have a large correlation with other subsampled curvelets Ug_p at different scales and orientations. Since the dictionary $\mathcal{D}_U$ is highly redundant and not sufficiently incoherent, the computed support $\tilde{\Lambda}$ may be very different from the sparse approximation support Λ of f [189]. It leads to interpolation along inappropriate directions which introduces errors. As a result, on typical images, this type of super-resolution interpolation in curvelet dictionaries does not provide better results then cubic spline interpolations. The same conclusion applies when using a dictionary of

bandlet vectors.

4.3 Structured Super-Resolution with Block Pursuits

As explained in Section 4.2, sparse super-resolution algorithms are highly flexible but suffer from this flexibility. In a curvelet or bandlet dictionary, a signal is sparse if it is well approximated by a small number of curvelets or bandlets, but there is no constraint on the properties of these curvelets or bandlets. Recovering these curvelets from a subsampled signal requires a full search in a large dictionaries which leads to errors. Directional interpolations are much more constraint since locally all pixels are recovered by performing an interpolation with a single direction.

To better take advantage of this property, instead of decomposing the signal over dictionary vectors that are selected individually, a structured sparse super-resolution interpolation is computed by projecting the signal over vector spaces. The space projection is calculated with the block pursuits described in Section 3.4. The choice of spaces is regularized by a hierarchical cascade of block pursuits that first selects locally the most appropriate angles in square neighborhoods and then finding the locations for these angles. The factorization of the block computation to angle estimation and location assignment reduces the computational complexity. In the selected spaces where the signal is directionally regular, the signal is linearly approximated which amounts to a directional linear interpolation.

4.3.1 Interpolations in Wavelet Domain

Observe in Figure 4.1(b) that there is very little aliasing in the low-frequency square $[-\pi/2, \pi/2]^2$ of $\hat{y}(\omega)$. Errors introduced by linear interpolations are mostly concentrated at high frequencies. To separate low and high frequencies, y is decomposed with the finest

scale wavelets $\{\psi_n^d\}_{1\leq d\leq 3,n}$ and the scaling functions $\{\phi_n\}_n$

$$y = y_H + y_L = \frac{1}{A}\left(\sum_{d=1}^{3}\sum_{u=0}^{N-1} Y_d[u]\psi_u^d + \sum_{u=0}^{N-1} Y[u]\phi_u\right), \tag{4.11}$$

where

$$Y_d[u] = \langle y, \psi_u^d\rangle,\ 1 \leq d \leq 3 \quad \text{and} \quad Y[u] = \langle y, \phi_u\rangle \tag{4.12}$$

are respectively the finest-scale translation-invariant wavelet coefficients along the three directions and the approximation coefficients, A is the wavelet frame bound, y_H and y_L are respectively the high-frequency and low-frequency components of y.

Since y_L is almost aliasing-free, it can be interpolated with a linear cubic-spline interpolator I_l. However, an optimized non-linear directional interpolation operator I_n is required to estimate the signal higher frequencies by interpolating y_H while removing the aliasing:

$$\tilde{f} = I_n(y_H) + I_l(y_L). \tag{4.13}$$

Observe that both interpolation and translation-invariant wavelet transform involve convolution operations that are commutable. Image interpolation can thus be casted as interpolation in the wavelet domain. The approximation coefficients are interpolated with the cubic spline interpolator

$$\tilde{F} = I_l(Y) \tag{4.14}$$

and the wavelet coefficients are interpolated with the non-linear directional interpolator

$$\tilde{F}_d = I_n(Y_d), \quad 1 \leq d \leq 3. \tag{4.15}$$

The convolution operations in interpolations are defined in the low-resolution grid, therefore the zoomed coefficients need to be separated to 4 sub-grids and each sub-grid reconstructs the corresponding sub-image, as illustrated in Figure 4.5. The super-resolution estimation $\tilde{f}$ is obtained by combining the sub-images.

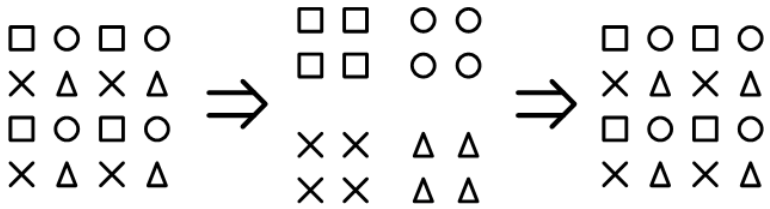


Figure 4.5: The zoomed wavelet and approximation coefficients are separated to four sub-grid represented by the four shapes. Each sub-grid reconstructs a sub-image. The super-resolution estimation is obtained by combining the sub-images.

4.3.2 Structured Sparsity and Directional Interpolation

The structured sparse super-resolution is conducted in the wavelet domain. To structure the sparse representation, the wavelet coefficients Y_d are projected to spaces selected from a dictionary $\{\mathbf{W}_k\}_k$ of vector spaces

$$Y_d = \sum_{m=0}^{M-1} P_{\mathbf{W}_{k_m}} Y_d. \tag{4.16}$$

The choice of the subspaces $\{\mathbf{W}_{k_m}\}_{0 \le m < M}$ is optimized to provide a sparse representation. As illustrated in Figure 4.6, a space $\mathbf{W}_k$ is generated by a block B_k of diracs defined simultaneously in the three wavelet orientations. The blocks are elongated with different directions that may locally fit the geometry. They are of same size so the spaces have constant dimension $\dim(\mathbf{W}_k) = B^{\#}$, $\forall k$. The decomposition (4.16) depends on Y_d and is therefore *non-linear*.

The super-resolution is performed by associating an interpolator I_m to each selected space W_{k_m}. If the coefficients in $\mathbf{W}_{k_m}$ are directionally regular, $P_{\mathbf{W}_{k_m}}$ is well approximated by a *linear* approximation

$$\widetilde{P_{\mathbf{W}_{k_m}} Y_d} = \sum_{p=0}^{P-1} \langle P_{\mathbf{W}_{k_m}} Y_d, g_p^m \rangle g_p^m, \quad P < B^{\#}, \tag{4.17}$$

and I_m is a linear directional interpolation. Otherwise I_m is a separable cubic spline inter-

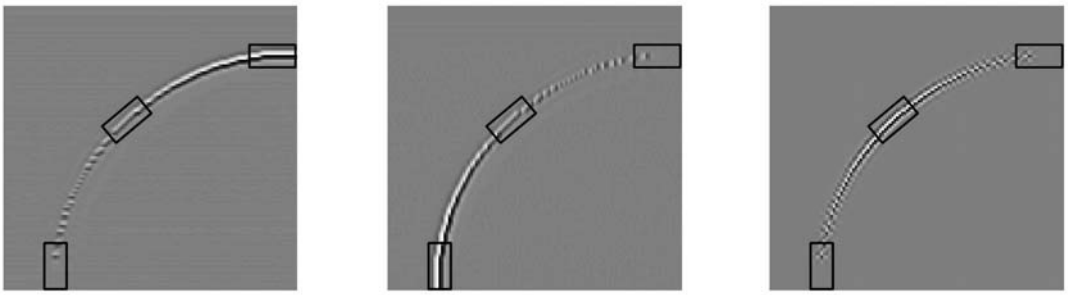


Figure 4.6: A block B_k covers simultaneously the finest scale wavelet image coefficients Y_d in the three orientations $d = 1, 2, 3$ (from left to right). For clarity, only 3 blocks are shown.

polation. The super-resolution estimation (4.15) can thus be written

$$\tilde{F}_d = I_n \left(\sum_{m=0}^{M-1} P_{\mathbf{W}_{k_m}} Y_d \right) = \sum_{m=0}^{M-1} I_m \left(P_{\mathbf{W}_{k_m}} Y_d \right) = \sum_{p=0}^{P-1} \langle P_{\mathbf{W}_{k_m}} Y_d, g_p^m \rangle I_m(g_p^m). \tag{4.18}$$

4.3.3 Window Fourier and Wavelet Block Pursuits

A hierarchical cascade of block pursuits in window Fourier and wavelet domains is proposed to select the vector spaces. The hierarchical procedure that factorizes the space computation to angle selection and location assignment regularizes the estimation and reduces the computational complexity.

Appropriate block directions correspond to orientations along which the image is locally regular. If the image is regular in the direction θ then its subsampling has an energy concentrated in a frequency band of angle $\theta + \pi/2$ plus some aliasing components illustrated in Figure 4.1(b). Directions of regularity in an image window can thus be identified by computing the energy concentration of a local image window Fourier transform along oriented frequency blocks, which is implemented with a block pursuit procedure.

Since a single direction must be estimated for each image pixel for directional interpolation, the window Fourier block pursuit is calculated over the finest scale image component

y_H that contains all the three wavelet orientations. y_H are localized in square windows by a multiplication with translated Hanning windows of 16×16 pixels. Let s be such a high frequency image window. The Fourier transform S of s gives the window Fourier transform of y_H. As the spectrum of an image window that contains some geometrical regularity is concentrated in an oriented band as shown in Figure 4.1(b), a dictionary $\mathcal{D}_b = \{b_k\}_{1 \leq k \leq K}$ of oriented blocks that pass the origin as illustrated in Figure 4.7 is constructed to identify the spectrum support of the directional regularity. The block orientations are uniformly sampled in $[0, \pi)$ over $K = 20$ angles in the numerical computation and all the blocks are adjusted to the have same size.

The direction identification in S is performed with a block pursuit procedure explained in Section 3.4 with the block dictionary $\mathcal{D}_b$. Selecting a family of blocks $\{b_{k_m}\}_{0 \leq m < M}$ amounts to selecting a family of local directions $\{\theta_m\}_{0 \leq m < M}$ along which the image window has some regularity. The block pursuit procedure in window Fourier identifies M directions in an image window, with M typically small since an image contains locally a small number of directions. In other words, the block pursuit procedure stops after M iterations. In the numerical computation, $M = 5$ over an image window of 16^2 pixels, and is thus much smaller than the total number $K = 20$ of possible angles.

The window Fourier block pursuit selects M directions along which the image window has some directional regularity, but it does not tell the location of directional regularity. A block pursuit procedure over the wavelet coefficients Y_d localized in the same window is cascaded to find the location. As shown in Figure 4.8, a dictionary $\mathcal{D}_B$ of blocks that corresponds to the M selected directions is constructed. Blocks of size $W_1 \times W_2 = 12 \times 2$ are used in the experiments. Using blocks with width $W_2 = 2$ instead of pure one-dimensional blocks with $W_2 = 1$ takes advantage of the fact that the translation-invariant wavelet coefficients along the contour have always more than one pixel wide, and reinforces the

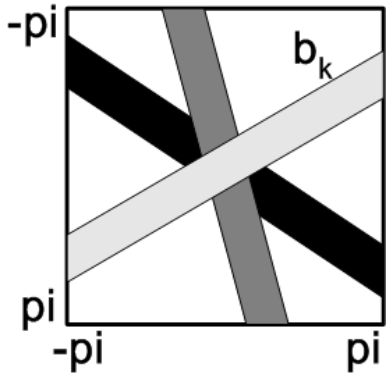


Figure 4.7: Blocks in window Fourier.

block selection decision. The blocks are translated to cover the window. As shown in Figure 4.6, a block B_k is defined simultaneously in the three wavelet orientations. The three wavelet orientations $\{Y_d\}_{1\leq d\leq 3}$ are thus combined in the block pursuit procedure so that locally a single direction is calculated. The block pursuit procedure stops until the residue $R^m Y_d$ in the window goes to zero, when the wavelet coefficients are fully covered by the selected blocks.

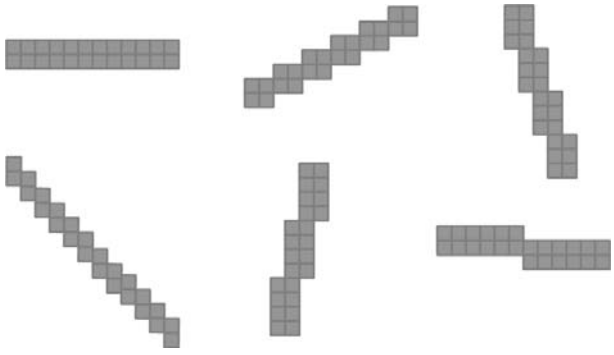


Figure 4.8: Examples of geometric blocks.

In order to avoid the border effect due to the local windows, in the numerical computation, the Hanning window is translated with half-overlapping to cover the image plane in

window Fourier block pursuits. In each 16×16 image window S, only the directions calculated for the central 8×8 pixels are retained. Selecting locally M angles by the window Fourier block pursuit procedure can be interpreted as pruning the wavelet block dictionary $\mathcal{D}_B$ by retaining M directions in each window. The wavelet block pursuit is performed on the whole wavelet coefficients array Y_d with the pruned block dictionary.

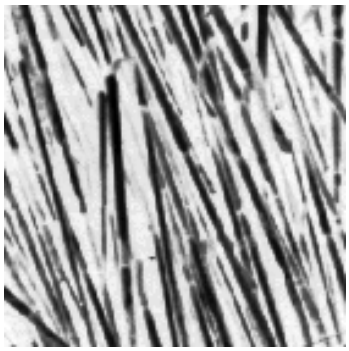
4.3.4 Directional Regularity

As explained in Section 4.2.1, an accurate directional interpolation requires that the image is regular in the interpolation direction. An oriented block B_{k_m} is selected such that wavelet coefficients have a high energy within block given that its angle is selected among local directions of regularity. To verify that wavelet coefficients are indeed regular in the block B_{k_m} of angle θ, we check that their derivatives in the block direction have a small energy

$$\gamma \sum_{d=1}^{3} \sum_{u \in B_{k_m}} \left| \frac{Y_d[u] - Y_d[u-\theta]}{\sqrt{2}} \right|^2 \leq \sum_{k=1}^{3} \sum_{u \in B_{k_m}} |Y_d[u]|^2,$$

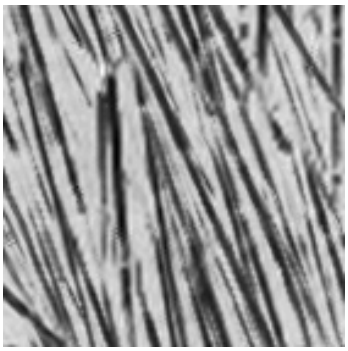
where $\gamma > 1$ is a threshold. If this is indeed the case, then wavelet coefficients do have a regular variation along the direction θ of B_{k_m} and are thus interpolated along this direction θ with the directional interpolator I_θ introduced in Section 4.2.2. Otherwise, if the regularity is not sufficient in the direction θ, a more conservative cubic spline interpolation is applied. A larger γ imposes a higher directional regularity requirement. In the numerical experiments, γ is set equal to 3.

Figure 4.9 compares a separable cubic spline interpolation with a super-resolution interpolation computed with the proposed super-resolution algorithm. The super-resolution achieves a significant PSNR improvement and improves the visual image quality where the image is geometrically regular. High-frequencies are restored along the direction of regularity. This clearly appears in the straws, the hat border and the hairs of various directions.



High-resolution image

Cubic spline 21.37 dB



Super-resolution 23.82 dB

High-resolution image

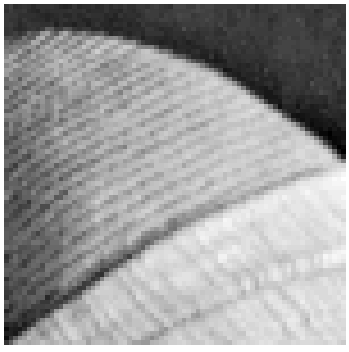
Cubic spline 34.74 dB

Super-resolution 35.81 dB

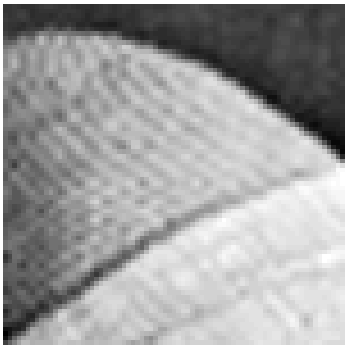
High-resolution image

Cubic spline 29.89 dB

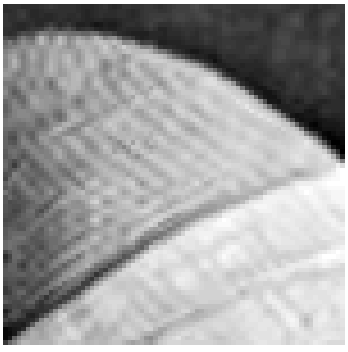
Super-resolution 30.35 dB



High-resolution image



Cubic spline 33.72 dB



Super-resolution 34.00 dB

Figure 4.9: From left to right: high-resolution image, cubic-spline interpolation, proposed block pursuit super-resolution.

However, on the hat image at the bottom, because of Moiré effects, the window Fourier block pursuit does not identify appropriate directions from the lower-resolution hat image, and the super-resolution barely improves the cubic-spline result. When the image is too complex, the aliasing may destroy the directions of image regularity, in which case no super-resolution is achieved.

Part II

Affine Invariant Image Comparison

Chapter 5

On the Consistency of the SIFT Method

The notion of sparsity is not only applied in signal denoising and super-resolution, but in computer vision as well. In computer vision, sparse features are important for pattern recognition since robust recognition of one object against the others requires a small number of salient features that capture the characteristics of the object. Sparse salient features should at the same time be invariant to variation of pattern observation conditions, for example viewpoint changes in image recognition, so that the recognition is independent to observation condition. In his milestone paper [119], Lowe has introduced the SIFT method (scale-invariant feature transform) that successfully incorporates scale, translation and rotation invariance in sparse features and has achieved brilliant success in image recognition applications.

This short Chapter is devoted to the mathematical arguments proving that SIFT is indeed similarity invariant. The mathematical proof is given under the assumption that the Gaussian smoothing performed by SIFT gives aliasing free sampling. The validity of

this main assumption is confirmed by a rigorous experimental procedure. These results explain why SIFT outperforms all other image feature extraction methods when it comes to scale invariance. The SIFT consistency proof contributes to the demonstration in the following Chapter 6 where we show that the new proposed method Affine-SIFT is fully affine invariant.

5.1 Introduction

Image matching aims at establishing correspondences between same objects that appear in different images. This is a fundamental step in many computer vision and image processing applications such as image recognition, 3D reconstruction, object tracking, robot localization and image registration [62].

The general (solid) shape matching problem starts with several photographs of a physical object, possibly taken with different cameras and viewpoints. These digital images are the *query* images. Given other digital images, the *search* images, the question is whether some of them contain, or not, a view of the object taken in the query image. This problem is by far more restrictive than the *categorization* problem, where the question is to recognize a *class* of objects, like chairs or cats. In the shape matching framework several instances of the very *same* object, or of copies of this object, are to be recognized.

A typical image matching method first detects points of interest, then selects a region around each point, and finally associates with each region a descriptor. Correspondences between two images may be established by matching the descriptors of both images.

Many variations exist on the computation of interest points, following the pioneering work of Harris and Stephens [85]. The Harris-Laplace and Hessian-Laplace region detectors [144, 147] are invariant to rotation and scale changes. Some moment-based region detec-

tors [112, 6] including Harris-Affine and Hessian-Affine region detectors [145, 147], an edge-based region detector [192], an intensity-based region detector [192], an entropy-based region detector [90], and two independently developed level line-based region detectors MSER ("maximally stable extremal region") [135] and LLD ("level line descriptor") [159, 160, 21] are designed to be invariant to affine transformations. These two methods stem from the Monasse image registration method [150] that used well contrasted extremal regions to register images. MSER is the most efficient one and has shown better performance than other affine invariant detectors [149]. However, as pointed out in [119], no known detector is actually fully affine invariant: All of them start with initial feature scales and locations selected in a non-affine invariant manner. The difficulty comes from the scale change from an image to another: This change of scale is actually an under-sampling, which means that the images differ by a blur.

In his milestone paper [119], Lowe has addressed this central problem and has proposed the so called scale-invariant feature transform (SIFT) descriptor, that is invariant to image translations and rotations, to scale changes (blur), and robust to illumination changes. It is also surprisingly robust to large enough orientation changes of the viewpoint (up to 60 degrees). Based on the scale-space theory [111], the SIFT procedure simulates all Gaussian blurs and normalizes local patches around scale covariant image key points that are Laplacian extrema. A number of SIFT variants and extensions, including PCA-SIFT [93] and gradient location-orientation histogram (GLOH) [148], that claim to have better robustness and distinctiveness with scaled-down complexity have been developed ever since [67, 103]. Demonstrated to be superior to other descriptors [146, 148], SIFT has been popularly applied for scene recognition [59, 151, 172, 197, 78, 174] and detection [69, 162], robot localization [9, 163, 158], image registration [213], image retrieval [84], motion tracking [194, 94], 3D modeling and reconstruction [171, 198], building panoramas [1, 13], or

photo management [212, 100, 23].

The initial goal of the SIFT method is to compare two images (or two image parts) that can be deduced from each other (or from a common one) by a rotation, a translation, and a zoom. The method turned out to be also robust to large enough changes in view point angle, which explains its success. In this method, following a classical paradigm, stable points of interest are supposed to lie at extrema of the Laplacian of the image in the image scale-space representation. The scale-space representation introduces a smoothing parameter σ. Images $\mathbf{u}_0$ are smoothed at several scales to obtain $\mathbf{w}(\sigma, x, y) =: (G_\sigma * \mathbf{u}_0)(x, y)$, where

$$G_\sigma(x, y) = G(\sigma, x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

is the 2D-Gaussian function with integral 1 and standard deviation σ. The notation $*$ stands for the space 2-D convolution in (x, y). The description of the SIFT method involves sampling issues, which we shall discuss later.

Taking apart all sampling issues and several thresholds whose aim is to eliminate unreliable features, the whole method can be summarized in one single sentence:

One sentence description *The SIFT method computes scale-space extrema (σ_i, x_i, y_i) of the space Laplacian of $w(\sigma, x, y)$, and then samples for each one of these extrema a square image patch whose origin is (x_i, y_i), whose x-direction is one of the dominant gradients around (x_i, y_i), and whose sampling rate is $\sqrt{\sigma_i^2 + \mathbf{c}^2}$.*

The constant $\mathbf{c} = 0.8$ is the tentative standard deviation of the image blur. The resulting samples of the digital patch at scale σ_i are encoded by their gradient direction, which is invariant under nondecreasing contrast changes. This accounts for the robustness of the method to illumination changes. In addition, only local histograms of the direction of the gradient are kept, which accounts for the robustness of the final descriptor to changes of view angle (see Figure 5.2).

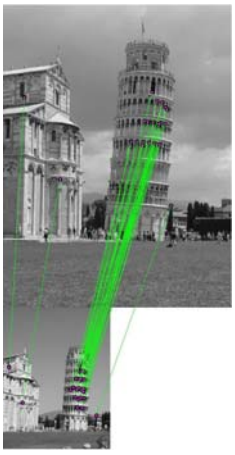


Figure 5.1: A result of the SIFT method, using an outliers elimination method [170]. Pairs of matching points are connected by segments.

The goal of this short Chapter is to give the mathematical arguments proving that the SIFT method indeed is scale invariant, and that its main assumption, that images are well-sampled under Gaussian blur, is right. Thus, this Chapters does not intend to propose a new variant or extension of the SIFT method; on the contrary it intends to demonstrate that no other method will ever improve more than marginally the SIFT scale invariance (see Figures 5.1 and 5.7 for striking examples). To the best of our knowledge, and in spite of the more than thousand papers quoting and using SIFT, the analysis presented here does not seem to have been done previously.

The Chapter is organized as follows. A simple formalism (Section 5.2) is introduced to obtain a condensed description of the SIFT shape encoding method. Using this formalism Section 5.4 proves mathematically that the SIFT method indeed computes translation, rotation and scale invariants. This proof is correct under the main assumption that image blur can be assumed to be Gaussian, and that images with a Gaussian blur larger than 0.8 are approximately (but accurately) well-sampled and can therefore be interpolated. Section 5.3 gives a procedure and checks the validity of this crucial Gaussian blur assumption.

5.2 Image Operators Formalizing SIFT

All continuous *image* operators including the sampling will be written in bold capital letters **A**, **B** and their composition as a mere juxtaposition **AB**. For any affine map A of the plane consider the affine transform of $\mathbf{u}$ defined by $\mathbf{A}\mathbf{u}(\mathbf{x}) =: \mathbf{u}(A\mathbf{x})$. For instance $\mathbf{H}_\lambda\mathbf{u}(\mathbf{x}) =: \mathbf{u}(\lambda\mathbf{x})$ denotes an expansion of $\mathbf{u}$ by a factor λ^{-1}. In the same way if $\mathbf{R}$ is a rotation, $\mathbf{R}\mathbf{u} =: \mathbf{u} \circ R$ is the image rotation by R^{-1}.

Sampling and Interpolation

Let us denote by $\mathbf{u}(\mathbf{x})$ a continuous and bounded image defined for every $\mathbf{x} = (x, y) \in \mathbb{R}^2$, and by u a digital image, only defined for $(n_1, n_2) \in \mathbb{Z}^2$. The δ-sampled image $u = \mathbf{S}_\delta\mathbf{u}$ is defined on $\mathbb{Z}^2$ by

$$\mathbf{S}_\delta\mathbf{u}(n_1, n_2) = \mathbf{u}(n_1\delta, n_2\delta); \tag{5.1}$$

Conversely, the Shannon interpolate of a digital image is defined as follows [73]. Let u be a digital image, defined on $\mathbb{Z}^2$ and such that $\sum_{n\in\mathbb{Z}^2} |u(n)|^2 < \infty$ and $\sum_{n\in\mathbb{Z}^2} |u(n)| < \infty$. (Of course, these conditions are automatically satisfied if the digital has a finite number of non-zero samples, which is the case here.) We call Shannon interpolation Iu of u the only $L^2(\mathbb{R}^2)$ function having u as samples and with spectrum support contained in $(-\pi, \pi)^2$. Iu is defined by the Shannon-Whittaker formula

$$Iu(x, y) =: \sum_{(n_1,n_2)\in\mathbb{Z}^2} u(n_1, n_2)\mathrm{sinc}(x - n_1)\mathrm{sinc}(y - n_2),$$

where $\mathrm{sinc}\ x =: \frac{\sin \pi x}{\pi x}$. The Shannon interpolation has the fundamental property $\mathbf{S}_1 Iu = u$. Conversely, if $\mathbf{u}$ is L^2 and band-limited in $(-\pi, \pi)^2$, then

$$I\mathbf{S}_1\mathbf{u} = \mathbf{u}. \tag{5.2}$$

In that case we simply say that $\mathbf{u}$ is *band-limited*. We shall also say that a digital image $u = \mathbf{S}_1\mathbf{u}$ is *well-sampled* if it was obtained from a band-limited image $\mathbf{u}$.

The Gaussian Semigroup

$\mathbf{G}$ denotes the convolution operator on $\mathbb{R}^2$ with the Gaussian kernel $\mathbf{G}_\sigma(x_1, x_2) = \frac{1}{2\pi(\mathbf{c}\sigma)^2} e^{-\frac{x_1^2+x_2^2}{2(\mathbf{c}\sigma)^2}}$, namely $\mathbf{G}\mathbf{u}(x, y) =: (\mathbf{G} * \mathbf{u})(x, y)$. $\mathbf{G}_\sigma$ satisfies the semigroup property

$$\mathbf{G}_\sigma\mathbf{G}_\beta = \mathbf{G}_{\sqrt{\sigma^2+\beta^2}}. \tag{5.3}$$

The proof of the next formula is a mere change of variables in the integral defining the convolution.

$$\mathbf{G}_\sigma\mathbf{H}_\gamma\mathbf{u} = \mathbf{H}_\gamma\mathbf{G}_{\sigma\gamma}\mathbf{u}. \tag{5.4}$$

Using the above notation, the next paragraph formalizes the SIFT method.

Formalized SIFT Scale Invariant Features Transform

The SIFT method is easily formalized in the continuous setting, while in practice images are always digital. The main assumption of the SIFT method being that all blurs can be assumed Gaussian, it will be crucial to prove that Gaussian blur gives in practice well-sampled images.

1. **Geometry:** there is an underlying infinite resolution bounded planar image $\mathbf{u}_0(\mathbf{x})$ that has undergone a similarity $\mathbf{A}\mathbf{u}_0$ (modeling a rotation, translation, and homothety) before sampling.

2. **Sampling and blur:** the camera blur is assimilated to a Gaussian with standard deviation $\mathbf{c}$. The typical value of $\mathbf{c}$ will be fixed thereafter. In Lowe's paper, $\mathbf{c}$ belongs to $[0.5, 0.8]$. The initial digital image is therefore $u = \mathbf{S}_1\mathbf{G}_\mathbf{c}\mathbf{A}\mathbf{u}_0$;

3. **Sampled scale space:** at all scales $\sigma > 0$, the SIFT method computes a good sampling of $\mathbf{u}(\sigma, \cdot) = \mathbf{G}_\sigma \mathbf{G}_\mathbf{c} \mathbf{A} \mathbf{u}_0$ and "key points" $(\sigma, \mathbf{x})$, namely scale and space extrema of $\Delta \mathbf{u}(\sigma, \cdot)$;

4. **Covariant resampling:** the blurred $\mathbf{u}(\sigma, \cdot)$ image is sampled around each key point at a rate proportional to $\sqrt{\mathbf{c}^2 + \sigma^2}$. The directions of the sampling axes are fixed by a dominant direction of $\nabla \mathbf{u}(\sigma, \cdot)$ in a σ-neighborhood of the key point. This yields rotation, translation and scale invariant samples in which the 4 parameters of $\mathbf{A}$ have been eliminated (see Figure 5.3);

5. **Illumination invariance:** the final SIFT descriptors keep only the orientation of the samples gradient to gain invariance with respect to light conditions.

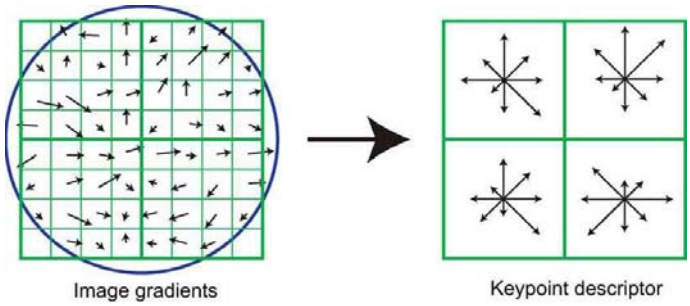


Figure 5.2: Each key-point is associated *a square image patch whose size is proportional to the scale and whose side direction is given by the assigned direction.* Example of a 2×2 descriptor array of orientation histograms (right) computed from an 8×8 set of samples (left). The orientation histograms are quantized into 8 directions and the length of each arrow corresponds to the magnitude of the histogram entry.

Steps 1 to 5 are the main steps of the method. We have omitted all details that are not relevant in the discussion to follow. Let them be mentioned briefly. The Laplacian extrema are kept only if they are larger than a fixed threshold that eliminates small features mainly due to noise. This threshold is not scale invariant. The ratio of the eigenvalues of the Hessian of the Laplacian must be close enough to 1 to ensure a good key point localization.

Figure 5.3: SIFT key points. The arrow starting point, length and the orientation signify respectively the key point position, scale, and dominant orientation. These features are covariant to any image similarity change.

(Typically, straight edge points have only one large Hessian eigenvalue, are poorly localized, and are therefore ruled out by this second threshold, which is scale invariant.)

Two more features, however, must be commented upon. Lowe assumes that the initial image has a $\mathbf{c} = 0.5$ Gaussian blur. (We call $\mathbf{c}$ Gaussian blur a convolution with a Gaussian with standard deviation $\mathbf{c}$). This implies a slight under-sampling that is compensated by a complementary Gaussian blur applied to the image, that puts the actual initial blur to 0.8. In accordance with this choice, a 2-sub-sampling in the SIFT scale-space computations is always preceded by a $2 \times 0.8 = 1.6$ Gaussian blur.

Of course, the Gaussian convolution cannot be applied to the continuous image but only to the samples. This is valid if and only if a discrete convolution can give an account of the underlying continuous one, that is, if the image is well-sampled.

The **discrete Gaussian convolution** applied to a digital image is defined as a digital operator by

$$G_\delta u =: S_1 \mathbf{G}_\delta I u. \tag{5.5}$$

This definition maintains the Gaussian semi-group used repeatedly in SIFT,

$$G_\delta G_\beta = G_{\sqrt{\delta^2+\beta^2}}. \tag{5.6}$$

Indeed, using twice (5.5) and once (5.3) and (5.2),

$$G_\delta G_\beta u = \mathbf{S}_1\mathbf{G}_\delta I\mathbf{S}_1\mathbf{G}_\beta Iu = \mathbf{S}_1\mathbf{G}_\delta\mathbf{G}_\beta Iu = \mathbf{S}_1\mathbf{G}_{\sqrt{\delta^2+\beta^2}}Iu = G_{\sqrt{\delta^2+\beta^2}}u.$$

The SIFT method uses repeatedly this formula and a 2-sub-sampling of images with Gaussian blur larger than 1.6. To summarize, the SIFT sampling manoeuvres are valid if and only if:

Claim 1. *For every σ larger than 0.8 and every continuous and bounded image $\mathbf{u}_0$, the Gaussian blurred image $\mathbf{G}_\sigma\mathbf{u}_0$ is well sampled, namely $I\mathbf{S}_1\mathbf{G}_\sigma\mathbf{u}_0 = \mathbf{G}_\sigma\mathbf{u}_0$.*

This claim is not a mathematical statement, but it will be checked experimentally in the next Section.

5.3 The Right Gaussian Blur to Achieve Well-sampling

Images need to be blurred before they are sampled. In principle Gaussian blur cannot lead to a good sampling because it is not *stricto sensu* band limited. Therefore the Shannon-Whittaker formula does not apply. However, in practice it does. The aim here is to define a procedure that checks that a Gaussian blur works and to fix the minimal variance of the blur ensuring well-sampling (up to a minor mean square and visual error).

One must distinguish two types of blur: The *absolute* blur with standard deviation $\mathbf{c}_a$ is the one that must be applied to an ideal infinite resolution (blur free) image to create an approximately band-limited image before 1-sampling; The *relative* blur $\sigma = \mathbf{c}_r(t)$ is the one that must be applied to a well-sampled image before a sub-sampling by a factor of t.

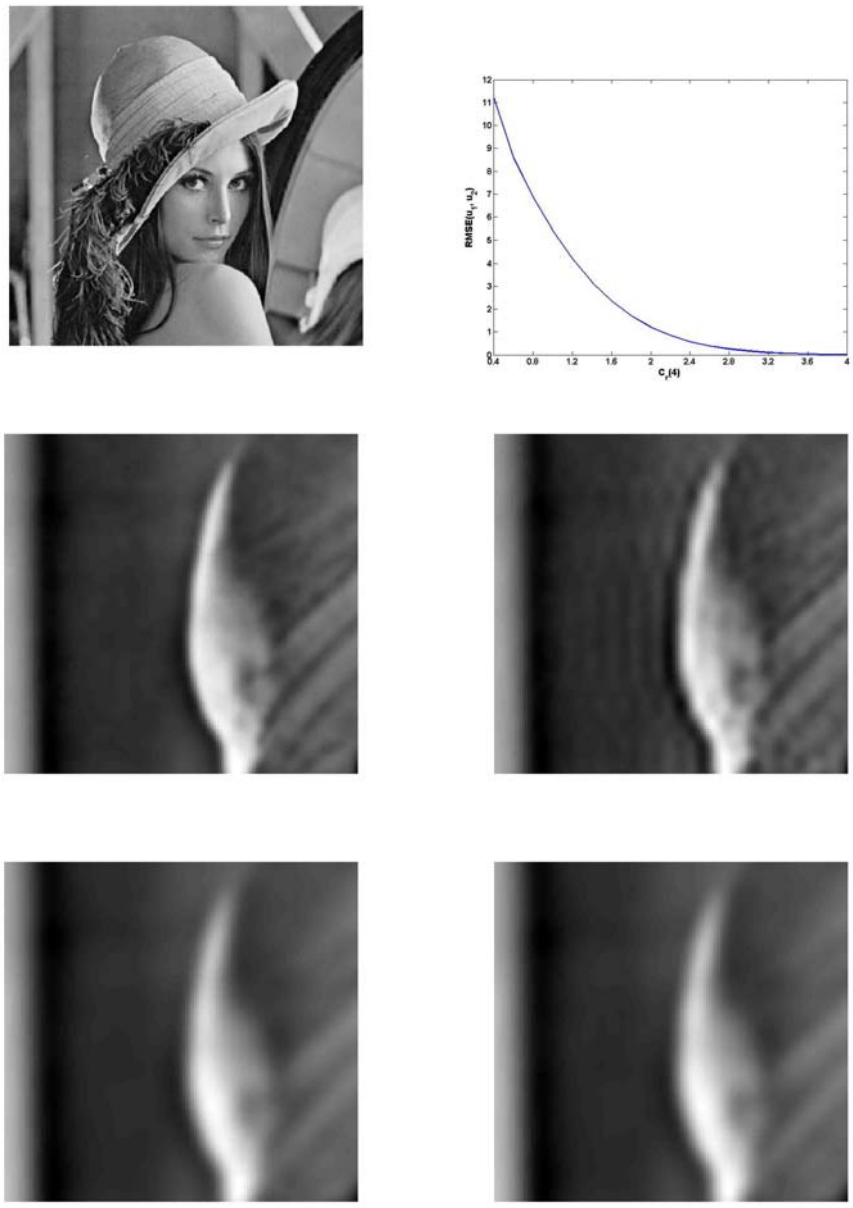


Figure 5.4: Top left: $\mathbf{u}$ Lena. Top right: RMSE(u_1, u_2) vs $\mathbf{c}_r(4)$. Middle (from left to right): u_1 and u_2 (zoomed) with $\mathbf{c}_r(4) = 1.6$. RMSE$(\mathbf{u}_1, \mathbf{u}_2)$=2.3. Bottom (from left to right): u_1 and u_2 (zoomed) with $\mathbf{c}_r(4) = 3.2$. RMSE(u_1, u_2)=0.1.

In the case of Gaussian blur, because of the semi-group formula (5.3), the relation between the absolute and relative blur is

$$t^2\mathbf{c}_a^2 = \mathbf{c}_r^2(t) + \mathbf{c}_a^2,$$

which yields

$$\mathbf{c}_r(t) = \mathbf{c}_a\sqrt{t^2 - 1}. \tag{5.7}$$

In consequence, if $t \gg 1$, then $\mathbf{c}_r(t) \approx \mathbf{c}_a t$.

Two experiments have been designed to calculate the anti-aliasing absolute Gaussian blur $\mathbf{c}_a$ ensuring that an image is approximately well-sampled. The first experiment compares for several values of $\mathbf{c}_r(t)$ the digital images

$$u_1 =: G_{\mathbf{c}_r(t)}u = \mathbf{S}_1\mathbf{G}_{\mathbf{c}_r(t)}Iu \quad \text{and} \quad u_2 =: (\mathbf{S}_{1/t}I)\mathbf{S}_t G_{\mathbf{c}_r(t)}u = (\mathbf{S}_{1/t}I)\mathbf{S}_t\mathbf{G}_{\mathbf{c}_r(t)}Iu,$$

where u is an initial digital image that is well-sampled, $\mathbf{S}_t$ is a t sub-sampling operator, $\mathbf{S}_{1/t}$ a t over-sampling operator, and I a Shannon-Whitakker interpolation operator. The discrete convolution by a Gaussian is defined in (5.5). Since t is an integer, the t sub-sampling is trivial. The Shannon over-sampling $\mathbf{S}_{1/t}I$ with an integer zoom factor t is obtained by the classic zero-padding method. This method is exactly Shannon interpolation if the initial image is both band-limited and periodic [73].

If the anti-aliasing filter size $\mathbf{c}_r(t)$ is too small, u_1 and u_2 can be very different. The right value of $\mathbf{c}_r(t)$ should be the smallest value permitting $u_1 \approx u_2$. Figure 5.4 shows u_1 and u_2 with $t = 4$ and plots their root mean square error $\mathrm{RMSE}(u_1, u_2)$. An anti-aliasing filter with $\mathbf{c}_r(4) = 1.6$ is clearly not broad enough: u_2 presents strong ringing artifacts. The ringing artifact is instead hardly noticeable with $\mathbf{c}_r(4) = 3.2$. The value $\mathbf{c}_r(4) \simeq 3.2$ is a good visual candidate, and this choice is confirmed by the curve showing that $\mathrm{RMSE}(u_1, u_2)$ decays rapidly until $\mathbf{c}_r(4)$ gets close to 3.2, and is stable and small thereafter. By (5.7), this value of $\mathbf{c}_r$ yields $\mathbf{c}_a = 0.83$. This value has been confirmed by experiments on ten

digital images. A doubt can be cast on this experiment, however, because its result slightly depends on the assumption that the initial blur on u is equal to $\mathbf{c}_a$. A second experiment is performed to verify this assumption.

In the second experiment, $\mathbf{c}_a$ has been evaluated directly by using a binary image u_0 that does not contain any blur. As illustrated in Figure 5.5, u_0 is obtained by binarizing Lena (Figure 5.4) with its median value as the threshold. Since u_0 is now blur-free, we can compare for several values of $\mathbf{c}_a$ and for $t = 4$, which is large enough, the digital images

$$u_1 =: G_{t\mathbf{c}_a} u = \mathbf{S}_1 \mathbf{G}_{t\mathbf{c}_a} I u \quad \text{and} \quad u_2 =: (\mathbf{S}_{1/t} I)\mathbf{S}_t G_{t\mathbf{c}_a} u = (\mathbf{S}_{1/t} I)\mathbf{S}_t \mathbf{G}_{t\mathbf{c}_a} I u,$$

As shown in Figure 5.5, $\mathbf{c}_a = 0.8$ is the smallest value ensuring no visual ringing in u_2. Under this value, for example for $\mathbf{c}_a = 0.4$, clear ringing artifacts are present in u_2. That $\mathbf{c}_a = 0.8$ is the correct value is confirmed by the RMSE(u_1, u_2) curve showing that the mean square error decays rapidly until $\mathbf{c}_a$ goes down to 0.8, and is stable and small thereafter. The result, confirmed in ten experiments with different initial images, is consistent with the value obtained in the first experimental setting.

Figure 5.6 illustrates the same experiment on a Gaussian white noise image that has constant spectrum energy over all frequencies. This example reflects the texture image sampling issue. The critical value $\mathbf{c}_a = 0.8$ is reconfirmed. Under this value, for example for $\mathbf{c}_a = 0.4$, u_2 looks clearly different than u_1, while the two are almost identical with $\mathbf{c}_a = 0.8$.

5.4 Scale and SIFT: Consistency of the Method

We denote by $\mathcal{T}$ an arbitrary image translation, by $\mathbf{R}$ an arbitrary image rotation, by $\mathbf{H}$ an arbitrary image homothety, and by $\mathbf{G}$ an arbitrary Gaussian convolution, all applied to continuous images. We say that there is strong commutation if we can exchange the

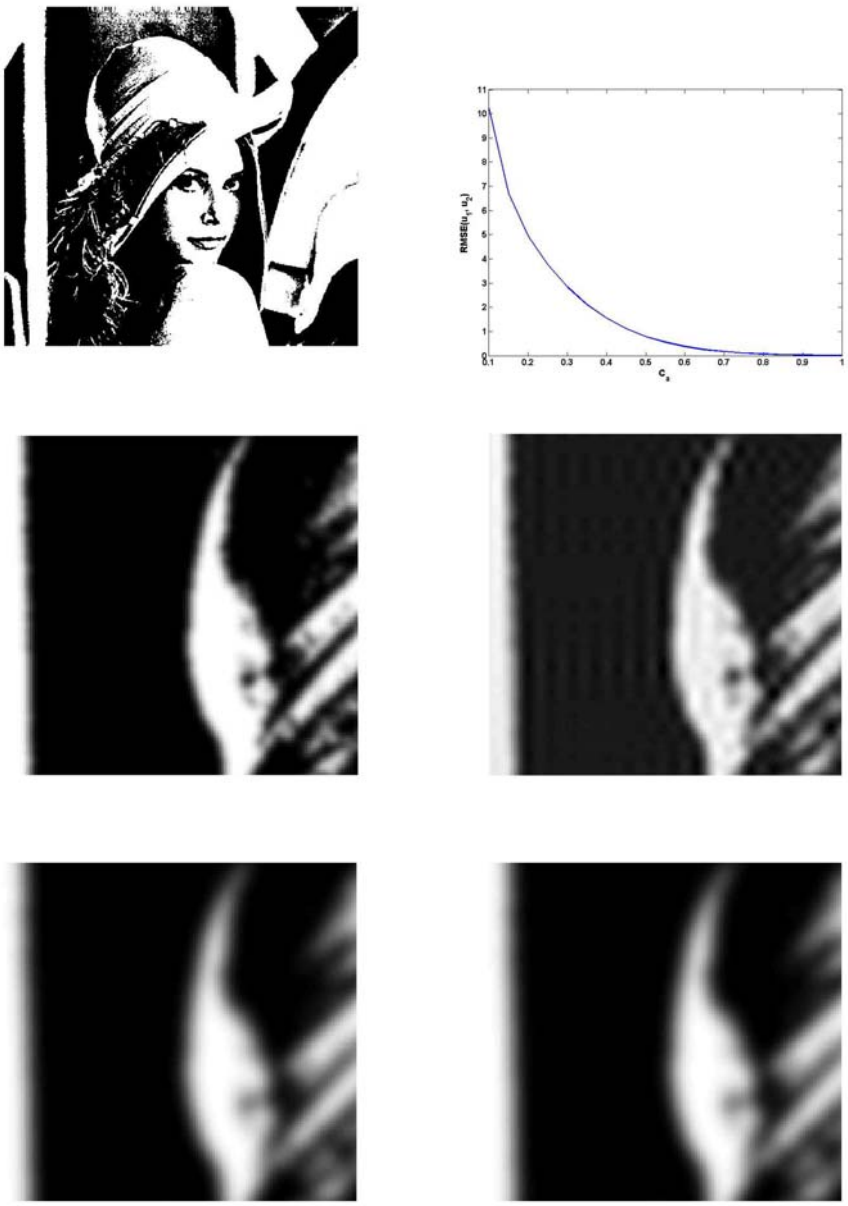


Figure 5.5: Top left: $\mathbf{u}$ Binarized Lena (gray-levels 50 and 0). Top right: RMSE(u_1, u_2) vs $\mathbf{c}_a$. Middle (from left to right): u_1 and u_2 (zoomed) with $\mathbf{c}_a = 0.4$. RMSE(u_1, u_2)=1.5. Bottom (from left to right): u_1 and u_2 (zoomed) with $\mathbf{c}_a = 0.8$. RMSE(u_1, u_2)=0.07.

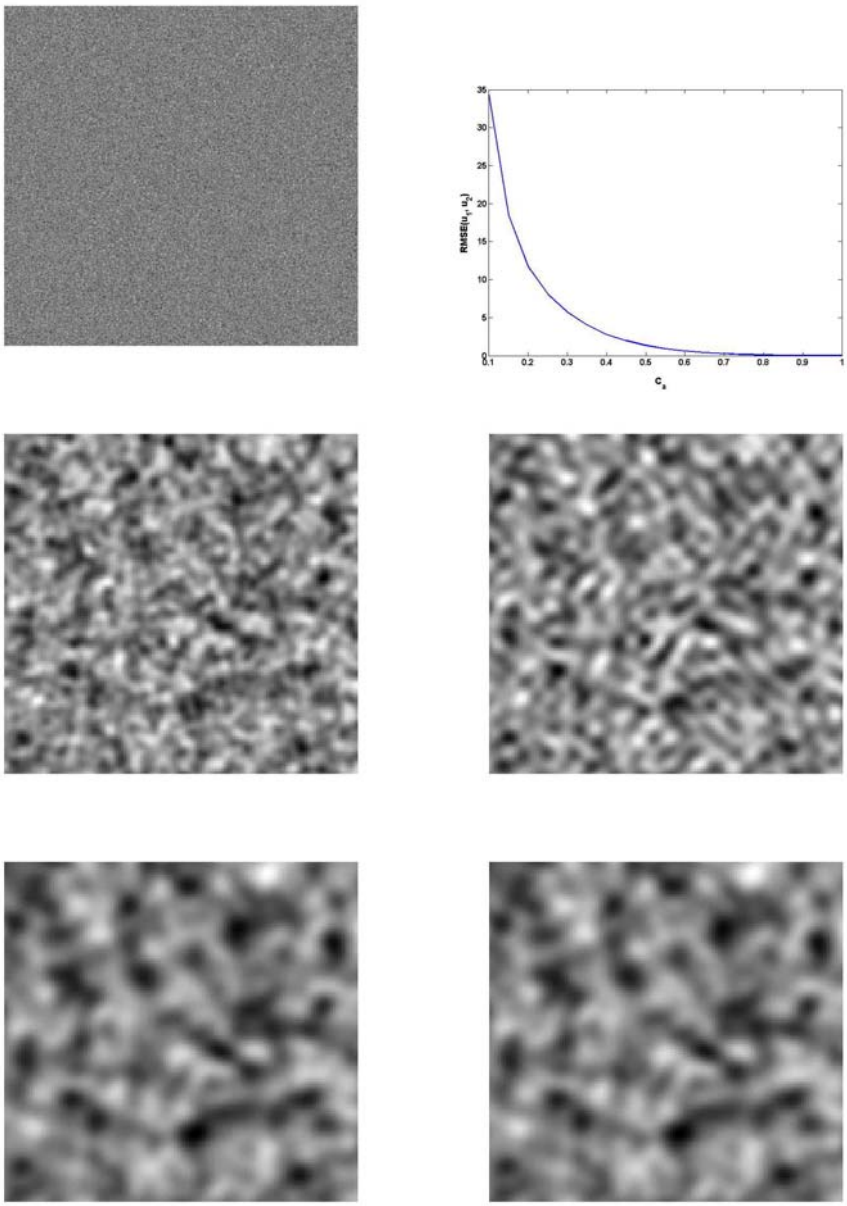


Figure 5.6: Top left: $\mathbf{u}$ Gaussian white noise ($\sigma = 30$). Top right: RMSE(u_1, u_2) vs $\mathbf{c}_a$. Middle (from left to right): u_1 and u_2 (zoomed) with $\mathbf{c}_a = 0.4$. RMSE(u_1, u_2)=2.8. Bottom (from left to right): u_1 and u_2 (zoomed) with $\mathbf{c}_a = 0.8$. RMSE(u_1, u_2)=0.1.

order of application of two of these operators. We say that there is weak commutation between two of these operators if we have (e.g.) $\mathbf{R}\mathcal{T} = \mathcal{T}'\mathbf{R}$, meaning that given $\mathbf{R}$ and $\mathcal{T}$ there is $\mathcal{T}'$ such that the former relation occurs. The next lemma is straightforward.

Lemma 1. *All of the aforementioned operators weakly commute. In addition,* $\mathbf{R}$ *and* $\mathbf{G}$ *commute strongly.*

In this Section, in conformity with the SIFT model of Section 5.2, the digital image is a frontal view of an infinite resolution ideal image $\mathbf{u}_0$. In that case, $\mathbf{A} = \mathbf{H}\mathcal{T}\mathbf{R}$ is the composition of a homethety $\mathbf{H}$, a translation $\mathcal{T}$ and a rotation $\mathbf{R}$. Thus the digital image is $u = \mathbf{S}_1\mathbf{G}_\delta\mathbf{H}\mathcal{T}\mathbf{R}\mathbf{u}_0$, for some $\mathbf{H}$, $\mathcal{T}$, $\mathbf{R}$ as above. Assuming that the image is not aliased boils down, by the experimental results of Section 5.3, to assuming $\delta \geq 0.8$.

Lemma 2. *For any rotation* $\mathbf{R}$ *and any translation* $\mathcal{T}$*, the SIFT descriptors of* $\mathbf{S}_1\mathbf{G}_\delta\mathbf{H}\mathcal{T}\mathbf{R}\mathbf{u}_0$ *are identical to those of* $\mathbf{S}_1\mathbf{G}_\delta\mathbf{H}\mathbf{u}_0$.

Proof. Using the weak commutation of translations and rotations with all other operators (Lemma 1), it is easily checked that the SIFT method is rotation and translation invariant: The SIFT descriptors of a rotated or translated image are identical to those of the original. Indeed, the set of scale space Laplacian extrema is covariant to translations and rotations. Then the normalization process for each SIFT descriptor situates the origin at each extremum in turn, thus canceling the translation, and the local sampling grid defining the SIFT patch has axes given by peaks in its gradient direction histogram. Such peaks are translation invariant and rotation covariant. Thus, the normalization of the direction also cancels the rotation. □

Lemma 3. *Let* u *and* v *be two digital images that are frontal snapshots of the same continuous flat image* $\mathbf{u}_0$*,* $u = \mathbf{S}_1\mathbf{G}_\beta\mathbf{H}_\lambda\mathbf{u}_0$ *and* $v =: \mathbf{S}_1\mathbf{G}_\delta\mathbf{H}_\mu\mathbf{u}_0$*, taken at different distances,*

with different Gaussian blurs and possibly different sampling rates. Let $\mathbf{w}(\sigma, \mathbf{x}) = (\mathbf{G}_\sigma \mathbf{u})(\mathbf{x})$ *denote the scale space of* $\mathbf{u}$*. Then the scale spaces of* u *and* v *are*

$$\mathbf{u}(\sigma, \mathbf{x}) = \mathbf{w}(\lambda\sqrt{\sigma^2 + \beta^2}, \lambda\mathbf{x}) \quad \textit{and} \quad \mathbf{v}(\sigma, \mathbf{x}) = \mathbf{w}(\mu\sqrt{\sigma^2 + \delta^2}, \mu\mathbf{x}).$$

If $(s_0, \mathbf{x}_0)$ *is a key point of* $\mathbf{w}$ *satisfying* $s_0 \geq \max(\lambda\beta, \mu\delta)$*, then it corresponds to a key point of* $\mathbf{u}$ *at the scale* σ_1 *such that* $\lambda\sqrt{\sigma_1^2 + \beta^2} = s_0$*, whose SIFT descriptor is sampled with mesh* $\sqrt{\sigma_1^2 + \mathbf{c}^2}$*, where the constant* $\mathbf{c} = 0.8$ *is the tentative standard deviation of the initial image blur. In the same way* $(s_0, \mathbf{x}_0)$ *corresponds to a key point of* $\mathbf{v}$ *at scale* σ_2 *such that* $s_0 = \mu\sqrt{\sigma_2^2 + \delta^2}$*, whose SIFT descriptor is sampled with mesh* $\sqrt{\sigma_2^2 + \mathbf{c}^2}$*.*

Proof. The interpolated initial images are by (5.2)

$$\mathbf{u} =: I\mathbf{S}_1\mathbf{G}_\beta\mathbf{H}_\lambda\mathbf{u}_0 = \mathbf{G}_\beta\mathbf{H}_\lambda\mathbf{u}_0 \text{ and } \mathbf{v} =: I\mathbf{S}_1\mathbf{G}_\delta\mathbf{H}_\mu\mathbf{u}_0 = \mathbf{G}_\delta\mathbf{H}_\mu\mathbf{u}_0.$$

Computing the scale-space of these images amounts to convolving these images for every $\sigma > 0$ with $\mathbf{G}_\sigma$, which yields, using the commutation relation (5.4) and the semigroup property (5.3):

$$\mathbf{u}(\sigma, \cdot) = \mathbf{G}_\sigma\mathbf{G}_\beta\mathbf{H}_\lambda\mathbf{u}_0 = \mathbf{G}_{\sqrt{\sigma^2+\beta^2}}\mathbf{H}_\lambda\mathbf{u}_0 = \mathbf{H}_\lambda\mathbf{G}_{\lambda\sqrt{\sigma^2+\beta^2}}\mathbf{u}_0.$$

By the same calculation, this function is compared by SIFT with

$$\mathbf{v}(\sigma, \cdot) = \mathbf{H}_\mu\mathbf{G}_{\mu\sqrt{\sigma^2+\delta^2}}\mathbf{u}_0.$$

Let us set $\mathbf{w}(s, \mathbf{x}) =: \mathbf{G}_s\mathbf{u}_0$. Then the scale spaces compared by SIFT are

$$\mathbf{u}(\sigma, \mathbf{x}) = \mathbf{w}(\lambda\sqrt{\sigma^2 + \beta^2}, \lambda\mathbf{x}) \quad \text{and} \quad \mathbf{v}(\sigma, \mathbf{x}) = \mathbf{w}(\mu\sqrt{\sigma^2 + \delta^2}, \mu\mathbf{x}).$$

Let us consider an extremal point $(s_0, \mathbf{x}_0)$ of the Laplacian of the scale space function $\mathbf{w}$. If $s_0 \geq \max(\lambda\beta, \mu\delta)$, an extremal point occurs at scales σ_1 for (the Laplacian of) $\mathbf{u}(\sigma, \mathbf{x})$ and σ_2 for (the Laplacian of) $\mathbf{v}(\sigma, \mathbf{x})$ satisfying

$$s_0 = \lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}. \tag{5.8}$$

We recall that each SIFT descriptor at a key point $(\sigma_1, \mathbf{x}_1)$ is computed from space samples of $\mathbf{x} \to \mathbf{u}(\sigma, \mathbf{x})$. The origin of the local grid is $\mathbf{x}_1$, the intrinsic axes are fixed by one of the dominant directions of the gradient of $\mathbf{u}(\sigma_1, \cdot)$ around $\mathbf{x}_1$, in a circular neighborhood whose size is proportional to σ_1. The SIFT descriptor sampling rate around the key point is also proportional to $\sqrt{\sigma_1^2 + \mathbf{c}^2}$ in $\mathbf{u}(\sigma_1, \mathbf{x})$, and to $\sqrt{\sigma_2^2 + \mathbf{c}^2}$ in $\mathbf{u}(\sigma_2, \mathbf{x})$. □

Theorem 1. *Let u and v be two digital images that are frontal snapshots of the same continuous flat image $\mathbf{u}_0$, $u = \mathbf{S}_1\mathbf{G}_\beta\mathbf{H}_\lambda\mathcal{T}\mathbf{R}\mathbf{u}_0$ and $v =: \mathbf{S}_1\mathbf{G}_\delta\mathbf{H}_\mu\mathbf{u}_0$, taken at different distances, with different Gaussian blurs and possibly different sampling rates, and up to a camera translation and rotation around its optical axe. Without loss of generality, assume $\lambda \leq \mu$. Then if the blurs are identical ($\beta = \delta = \mathbf{c}$, where the constant $\mathbf{c} = 0.8$ is the tentative standard deviation of the initial image blur), all SIFT descriptors of u are identical to SIFT descriptors of v. If $\beta \neq \delta$ (or $\beta = \delta \neq \mathbf{c}$), the SIFT descriptors of u and v become (quickly) similar when their scales grow, namely as soon as $\frac{\sigma_1}{\max(\mathbf{c},\beta)} \gg 1$ and $\frac{\sigma_2}{\max(\mathbf{c},\delta)} \gg 1$.*

Proof. By the result of Lemma 2, we can neglect the effect of translations and rotations. Therefore assume without loss of generality that the images under comparison are as in Lemma 3. Assume a key point $(s_0, \mathbf{x}_0)$ of $\mathbf{w}$ has scale $s_0 \geq \max(\lambda\beta, \mu\delta)$. This key point has a sampling rate proportional to s_0. There is a corresponding key point $(\sigma_1, \frac{\mathbf{x}_0}{\lambda})$ for $\mathbf{u}$ with sampling rate $\sqrt{\sigma_1^2 + \mathbf{c}^2}$ and a corresponding key point $(\sigma_2, \frac{\mathbf{x}_0}{\mu})$ with sampling rate $\sqrt{\sigma_2^2 + \mathbf{c}^2}$ for $\mathbf{v}$. To have a common reference for these sampling rates, it is convenient to refer to the corresponding sampling rates for $\mathbf{w}(s_0, \mathbf{x}_0)$, which are $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2}$ for the SIFT descriptors of $\mathbf{u}$ at scale σ_1, and $\mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$ for the descriptors of $\mathbf{v}$ at scale σ_2. Thus the SIFT descriptors of $\mathbf{u}$ and $\mathbf{v}$ for $\mathbf{x}_0$ will be identical if and only if $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} = \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$. Now, we have $\lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}$, which implies $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} = \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$ if and only

if

$$\lambda^2\beta^2 - \mu^2\delta^2 = (\lambda^2 - \mu^2)\mathbf{c}^2. \tag{5.9}$$

Since λ and μ correspond to camera distances to the observed object $\mathbf{u}_0$, they are pretty arbitrary. Thus in general the only way to get (5.9) is to have $\beta = \delta = \mathbf{c}$, which means that the blurs of both images have been guessed correctly. In any case, $\beta = \delta = \mathbf{c}$ does imply that the SIFT descriptors of both images are identical.

The second statement is straighforward: if σ_1 and σ_2 are large enough with respect to β, δ and $\mathbf{c}$, the relation $\lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}$, implies $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} \simeq \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$. □

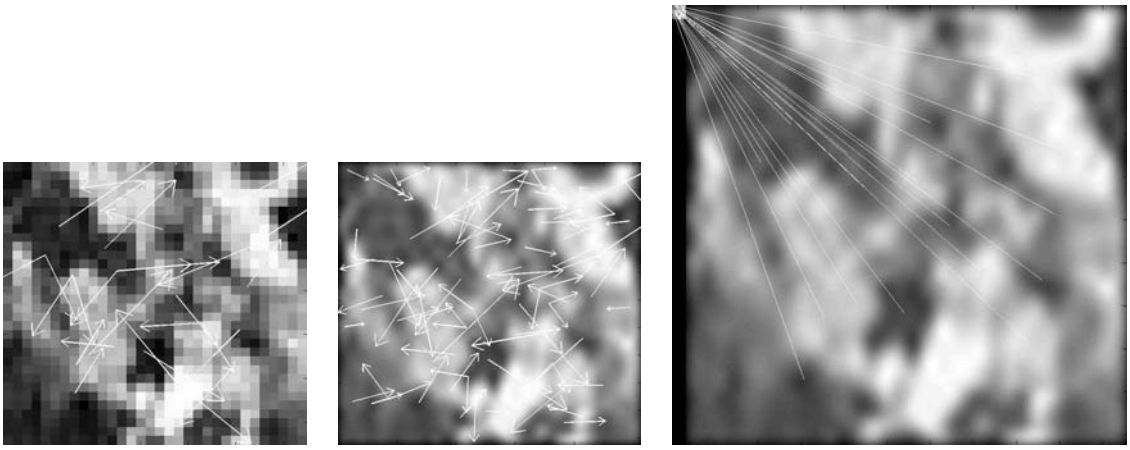


Figure 5.7: Scale invariance of SIFT, an illustration of Theorem 1. Left: a very small digital image u with its 28 key points. For the conventions to represent key points and matches, see the comments in Figure 5.3. Middle: this image is over sampled by a 32 factor to $\mathbf{S}_{1/32}Iu$. It has 86 key points. Right: 22 matches found between $\mathbf{u}$ and $\mathbf{H}_{1/32}\mathbf{u}$.

The almost perfect scale invariance of SIFT stated in Theorem 1 is illustrated by the striking example of Figure 5.7. The 28 SIFT key points of a very small image u are compared to the 86 key points obtained by zooming in u by a 32 factor: The resulting digital image is $v = \mathbf{S}_{1/32}Iu$, again obtained by zero-padding. For better observability, both images are displayed with the same size by enlarging the pixels of u. Almost each key point (22 out of

28) of u finds its counterpart in v. 22 matches are detected between the descriptors as shown on the right. If we trust Theorem 1, all descriptors of u should have been retrieved in v. This does not fully happen for two reasons. First, the SIFT method thresholds (not taken into account in the theorem) eliminate many potential key points. Second, the zero-padding interpolation giving v is imperfect near the image boundaries.

By the second part of Theorem 1, the reliability of the SIFT matching increases with scale. This fact is illustrated in Figure 5.8. Starting from a high resolution image $\mathbf{u}_0$, two images u and v are obtained by simulated zoom out, $u = \mathbf{S}_1\mathbf{G}_\beta\mathbf{H}_\lambda\mathbf{u}_0 = \mathbf{S}_\lambda\mathbf{G}_{\lambda\beta}\mathbf{u}_0$ and $v = \mathbf{S}_\mu\mathbf{G}_{\mu\delta}\mathbf{u}_0$, with $\lambda = 2$, $\mu = 4$, $\beta = \delta = 0.8$. Pairs of SIFT descriptors of u and v in correspondence, established by a SIFT matching, are compared using an Euclidean distance $\mathbf{d}$. The scale rate σ_1/σ_2 as well as the distance d between the matched key points are plotted against σ_2 in Figure 5.8. That $\sigma_1/\sigma_2 \approx 2$ for all key points confirms that the SIFT matching process is reliable. As stated by the theorem, the rate σ_1/σ_2 goes to $\mu/\lambda = 2$ when σ_2 increases, and the distance $\mathbf{d}$ goes down. However, when the scale is small ($\sigma_2 < 1$), σ_1/σ_2 is very different from 2 and $\mathbf{d}$ is large.

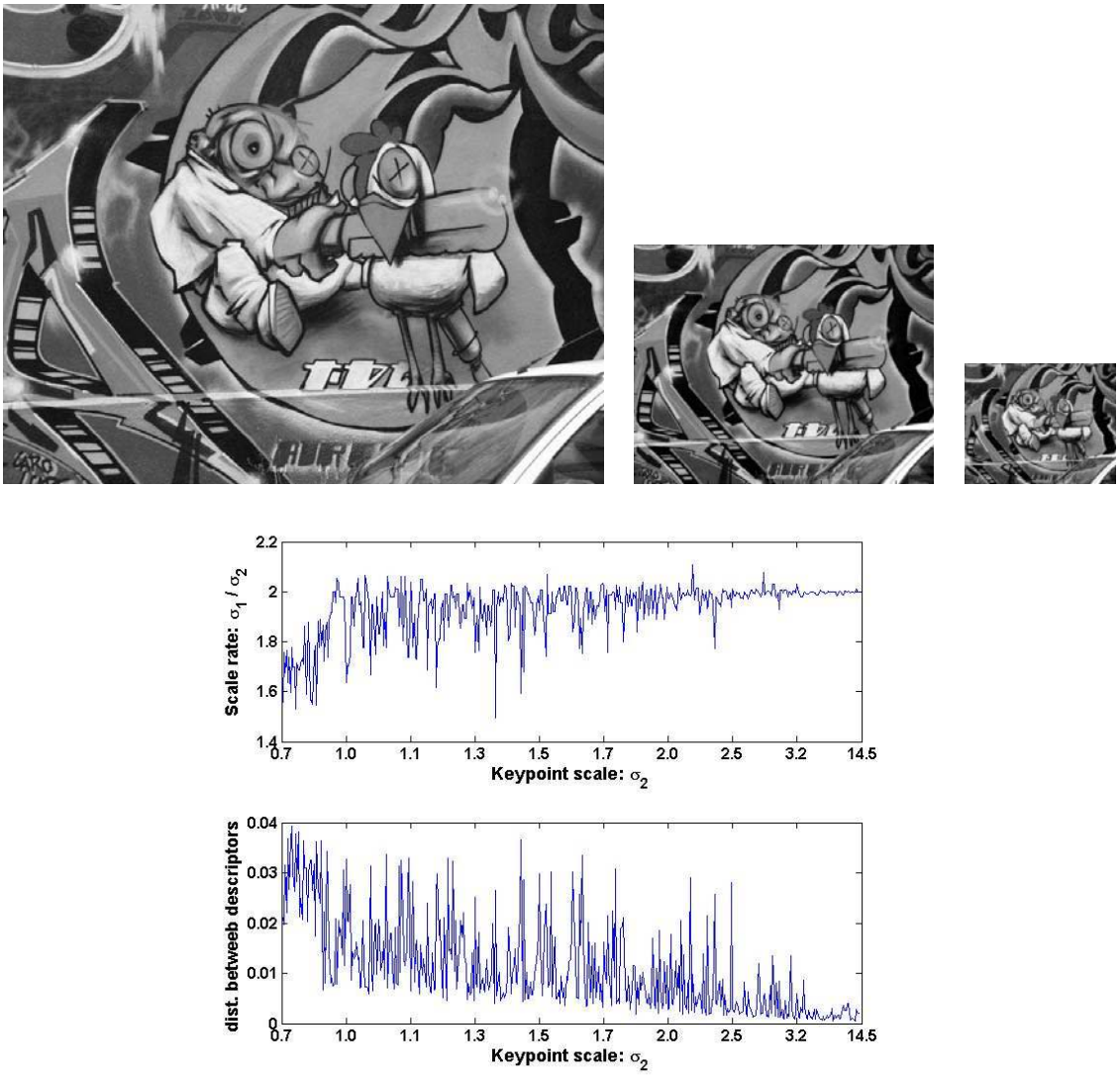


Figure 5.8: Top (from left to right): $\mathbf{u}_0$, u, v. Middle: Rate of scales σ_1/σ_2 of matched keypoints in u and v against σ_2. Bottom: Distance between matched descriptors of u and v against σ_2.

Chapter 6

ASIFT: A Fully Affine Invariant Image Comparison Method

In Chapter 5, the SIFT method [119] has been shown fully invariant with respect zoom, rotation and translation. SIFT simulates the zoom in the scale space and normalizes the rotation and translation. However, as SIFT does not treat the camera axis orientation parameters, its performance drops considerably when view angle change increases.

The method proposed in this Chapter, Affine-SIFT (ASIFT), simulates all image views obtainable by varying the two camera axis orientation parameters, namely the latitude and the longitude angles, left over by the SIFT method. Then it covers the other four parameters by using the SIFT method itself. The resulting method will be mathematically proved to be fully affine invariant. Against any prognosis, simulating all views depending on the two camera orientation parameters is feasible with no dramatic computational load. A coarse-to-fine two-resolution scheme further reduces the ASIFT complexity to about twice that of SIFT. Many experiments show that ASIFT outperforms significantly the state-of-the-art methods, including SIFT, MSER, Harris-Affine, and Hessian-Affine.

6.1 Introduction

Image matching aims at establishing correspondences between same objects that appear in different images and is a fundamental step in many computer vision and image processing applications. A major difficulty of image matching is viewpoint change. The change of camera position induces an apparent deformation of the object image. Thus, recognition must be invariant to such deformations.

The state-of-the-art image matching algorithms usually consist of two parts: *detector* and *descriptor*. They first detect points of interest in the images under comparison and select a region around each point of interest, and then associate an invariant descriptor or feature to each region. Correspondences may thus be established by matching the descriptors. Detectors and descriptors should be as invariant as possible.

In recent years local image detectors have bloomed. They can be classified by their incremental invariance properties. All of them are translation invariant. The Harris point detector [85] is also rotation invariant. The Harris-Laplace, Hessian-Laplace and the DoG (Difference-of-Gaussian) region detectors [144, 147, 119, 64] are invariant to rotations and changes of scale. Some moment-based region detectors [112, 6] including the Harris-Affine and Hessian-Affine region detectors [145, 147], an edge-based region detector [190, 192], an intensity-based region detector [191, 192], an entropy-based region detector [90], and two level line-based region detectors MSER ("maximally stable extremal region") [135] and LLD ("level line descriptor") [159, 160, 21] are designed to be invariant to affine transforms. MSER, in particular, has been demonstrated to have often better performance than other affine invariant detectors, followed by Hessian-Affine and Harris-Affine [149].

In his milestone paper [119], Lowe has proposed a scale-invariant feature transform (SIFT) that is invariant to image scaling, rotation and translation, and partially invariant

to illumination and viewpoint changes. The SIFT method combines the DoG region detector that is rotation, translation and scale invariant (a mathematical proof of its scale invariance is given in [154]) with a descriptor based on the gradient orientation distribution in the region, which is partially illumination and viewpoint invariant [119]. These two stages of the SIFT method will be called respectively *SIFT detector* and *SIFT descriptor*. The SIFT detector is *a priori* less invariant to affine transforms than the Hessian-Affine and the Harris-Affine detectors [144, 147]. However, when combined with the SIFT descriptor [149], its overall affine invariance turns out to be comparable, as we shall see in many experiments.

The SIFT descriptor has been shown to be superior to other many descriptors [146, 148] such as the distribution-based shape context [8], the geometric histogram [2] descriptors, the derivative-based complex filters [6, 173], and the moment invariants [195]. A number of SIFT descriptor variants and extensions, including PCA-SIFT [93], GLOH (gradient location-orientation histogram) [148] and SURF (speeded up robust features) [7] have been developed ever since [67, 103]. They claim more robustness and distinctiveness with scaled-down complexity. The SIFT method and its variants have been popularly applied for scene recognition [59, 151, 172, 197, 78, 174, 215, 152] and detection [69, 162], robot localization [9, 175, 163, 158], image registration [213], image retrieval [84], motion tracking [194, 94], 3D modeling and reconstruction [171, 198], building panoramas [1, 13], photo management [212, 100, 182, 23], as well as symmetry detection [120].

The mentioned state-of-the-art methods have achieved brilliant success. However, none of them is fully affine invariant. As pointed out in [119], Harris-Affine and Hessian-Affine start with initial feature scales and locations selected in a non-affine invariant manner. The non-commutation between optical blur and affine transforms shown in Section 6.3 also explains the limited affine invariance performance of the normalization methods MSER, LLD, Harris-Affine and Hessian-Affine. As shown in [21], MSER and LLD are not even

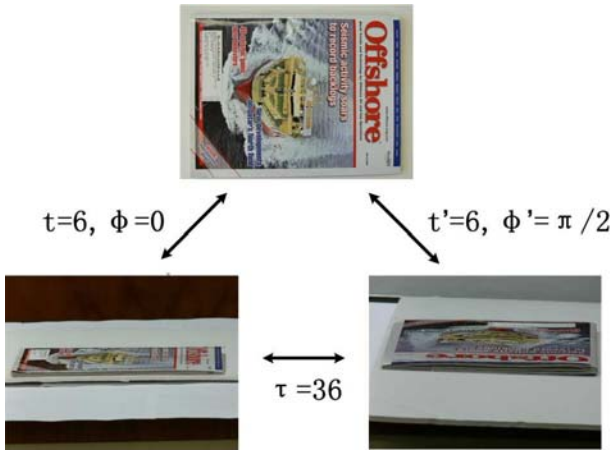


Figure 6.1: The frontal image (above) is squeezed in one direction on the left image by a slanted view, and squeezed in an orthogonal direction by another slanted view. The compression factor or *absolute tilt* is about 6 in each view. The resulting compression factor, or *transition tilt* from left to right is actually 36. See Section 6.2 for the formal definition of these tilts. Transition tilts quantify the affine distortion. The aim is to detect image similarity under transition tilts as large as this one.

fully scale invariant: they do not cope with the drastic changes of the level line geometry due to blur. SIFT is actually the only method that is fully scale invariant. However, since it is not designed to cover the whole affine space, its performance drops quickly under substantial viewpoint changes.

The present Chapter proposes a fully affine-invariant image comparison method Affine-SIFT (ASIFT). Unlike MSER, LLD, Harris-Affine and Hessian-Affine which normalize all the six affine parameters, ASIFT simulates three parameters and normalizes the rest. The scale and the camera axis orientation angles are the three simulated parameters. The other three, rotation and translation, are normalized. More specifically, ASIFT simulates the two camera axis parameters, and then applies SIFT which simulates the scale and normalizes the rotation and the translation. A coarse-to-fine two-resolution implementation of ASIFT is proposed, that has about twice the complexity of a single SIFT routine. To the best of our knowledge the first work suggesting to simulate affine parameters appeared in [168]

where the authors proposed to simulate four tilt deformations in a cloth motion capture application.

The Chapter introduces a crucial parameter for evaluating the performance of affine recognition, the *transition tilt*. The transition tilt measures the degree of viewpoint change from one view to another. Figures 6.1 and 6.2 give a first intuitive approach to *absolute tilt* and *transition tilt*. They illustrate why simulating large tilts on both compared images proves necessary to obtain a fully affine invariant recognition. Indeed, transition tilts can be much larger than absolute tilts. In fact they can behave like the square of absolute tilts. The affine invariance performance of the state-of-the-art methods will be evaluated by their attainable transition tilts.

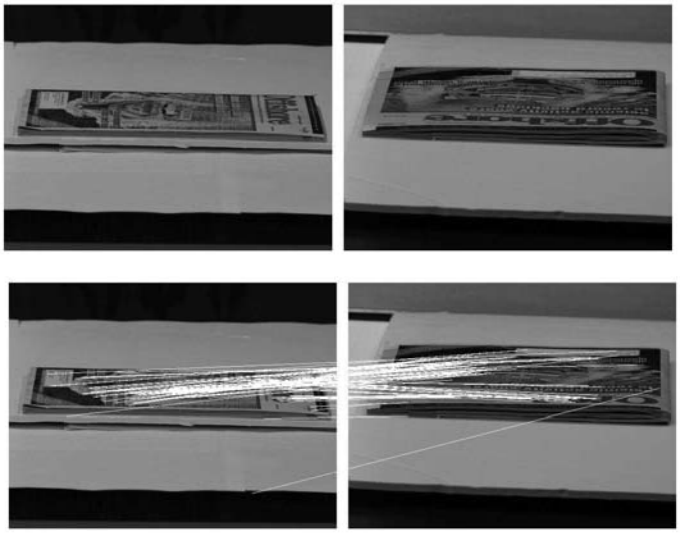


Figure 6.2: Top: Image pair with transition tilt $t \approx 36$. (SIFT, Harris-Affine, Hessian-Affine and MSER fail completely.) Bottom: ASIFT finds 120 matches out which 4 are false. See comments in text.

The Chapter is organized as follows. Section 6.2 describes the affine camera model and introduces the transition tilt. Section 6.3 reviews the state-of-the-art image matching method SIFT, MSER, Harris-Affine and Hessian-Affine and explains why they are not fully

affine invariant. The ASIFT algorithm is described in Section 6.4. Section 6.5 gives a mathematical proof that ASIFT is fully affine invariant, up to sampling approximations. Section 6.6 is devoted to extensive experiments where ASIFT is compared with the state-of-the art algorithms.

A website with an online demo is available.

http://www.cmap.polytechnique.fr/∼yu/research/ASIFT/demo.html. It allows the users to test ASIFT with their own images. It also contains an image dataset (for systematic evaluation of robustness to absolute and transition tilts), and more examples.

6.2 Affine Camera Model and Tilts

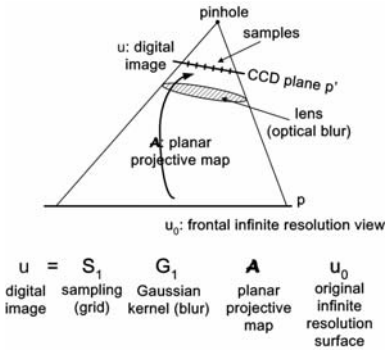


Figure 6.3: The projective camera model $\mathbf{u} = \mathbf{S}_1 \mathrm{G}_1 \mathcal{A} u_0$. $\mathcal{A}$ is a planar projective transform (a homography). G_1 is an anti-aliasing Gaussian filtering. $\mathbf{S}_1$ is the CCD sampling.

As illustrated by the camera model in Figure 6.3, digital image acquisition of a flat object can be described as

$$\mathbf{u} = \mathbf{S}_1 \mathrm{G}_1 A \mathcal{T} u_0 \tag{6.1}$$

where $\mathbf{u}$ is a digital image and u_0 is an (ideal) infinite resolution frontal view of the flat object. $\mathcal{T}$ and A are respectively a plane translation and a planar projective map due to the camera motion. G_1 is a Gaussian convolution modeling the optical blur, and $\mathbf{S}_1$ is the stan-

dard sampling operator on a regular grid with mesh 1. The Gaussian kernel is assumed to be broad enough to ensure no aliasing by the 1-sampling, namely $I\mathbf{S}_1\mathrm{G}_1 A\mathcal{T} u_0 = \mathrm{G}_1 A\mathcal{T} u_0$, where I denotes the Shannon-Whittaker interpolation operator. A major difficulty of the recognition problem is that the Gaussian convolution G_1, which becomes a broad convolution kernel when the image is zoomed out, does not commute with the planar projective map A.

6.2.1 The Affine Camera Model

Figure 6.4: The global deformation of the ground is strongly projective (a rectangle becomes a trapezoid), but the local deformation is affine: each tile on the pavement is almost a parallelogram.

We shall proceed to a further simplification of the above model, by reducing A to an affine map. Figure 6.4 shows one of the first perspectively correct Renaissance paintings by Paolo Uccello. The perspective on the ground is strongly projective: the rectangular pavement of the room becomes a trapezoid. However, each tile on the pavement is almost a parallelogram. This illustrates the local tangency of perspective deformations to affine maps. Indeed, by the first order Taylor formula, any planar smooth deformation can be approximated around each point by an affine map. The apparent deformation of a plane object induced by a camera motion is a planar homographic transform, which is smooth, and therefore locally tangent to affine transforms. More generally, a solid object's apparent

deformation arising from a change in the camera position can be locally modeled by affine planar transforms, provided the object's facets are smooth. In short, all local perspective effects can be modeled by local affine transforms $u(x, y) \to u(ax + by + e, cx + dy + f)$ in each image region.

Figure 6.5 illustrates the same fact by interpreting the local behavior of a camera as equivalent to multiple cameras at infinity. These cameras at infinity generate affine deformations. In fact, a camera position change can generate any affine map with positive determinant. The next theorem formalizes this fact and gives a camera motion interpretation to affine deformations.

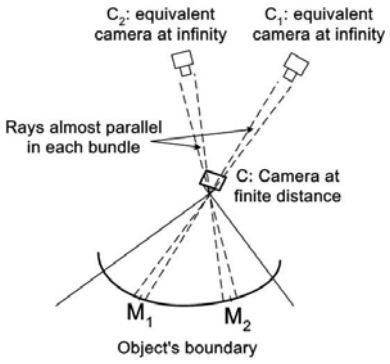


Figure 6.5: A camera at finite distance looking at a smooth object is equivalent to multiple local cameras at infinity. These cameras at infinity generate affine deformations.

Theorem 2. *Any affine map* $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ *with strictly positive determinant which is not a similarity has a unique decomposition*

$$A = H_\lambda R_1(\psi) T_t R_2(\phi) = \lambda \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \tag{6.2}$$

where $\lambda > 0$, $\lambda^2 t$ *is the determinant of* A, R_i *are rotations,* $\phi \in [0, \pi)$, *and* T_t *is a tilt, namely a diagonal matrix with first eigenvalue* $t > 1$ *and the second one equal to* 1.

The theorem follows the Singular Value Decomposition (SVD) principle. The proof is given in the Appendix A.

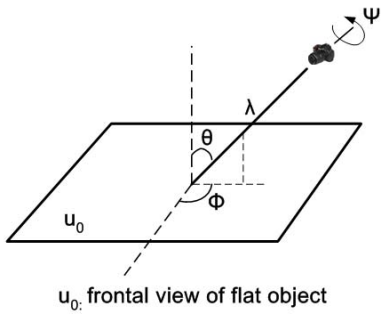


Figure 6.6: Geometric interpretation of the decomposition (6.2). The image u is a flat physical object. The angles ϕ and θ are respectively the camera optical axis *longitude* and *latitude*. A third angle ψ parameterizes the camera spin, and λ is a zoom parameter.

Figure 6.6 shows a camera motion interpretation of the affine decomposition (6.2): ϕ and $\theta = \arccos 1/t$ are the viewpoint angles, ψ parameterizes the camera spin and λ corresponds to the zoom. The camera is assumed to stay far away from the image and starts from a frontal view u, i.e., $\lambda = 1$, $t = 1$, $\phi = \psi = 0$. The camera can first move parallel to the object's plane: this motion induces a translation $\mathcal{T}$ that is eliminated by assuming (without loss of generality) that the camera axis meets the image plane at a fixed point. The plane containing the normal and the optical axis makes an angle ϕ with a fixed vertical plane. This angle is called *longitude*. Its optical axis then makes a θ angle with the normal to the image plane u. This parameter is called *latitude*. Both parameters are classical coordinates on the *observation hemisphere*. The camera can rotate around its optical axis (rotation parameter ψ). Last but not least, the camera can move forward or backward, as measured by the zoom parameter λ.

In (6.2) the tilt parameter, which has a one-to-one relation to the latitude angle $t =$

$1/\cos\theta$, entails a strong image deformation. It causes a directional subsampling of the frontal image in the direction given by the longitude ϕ.

6.2.2 Transition Tilts

The parameter t in (6.2) is called *absolute tilt*, since it measures the tilt between the *frontal* view and a *slanted* view. In real applications, both compared images are usually slanted views. The *transition tilt* is designed to quantify the amount of tilt between two such images.

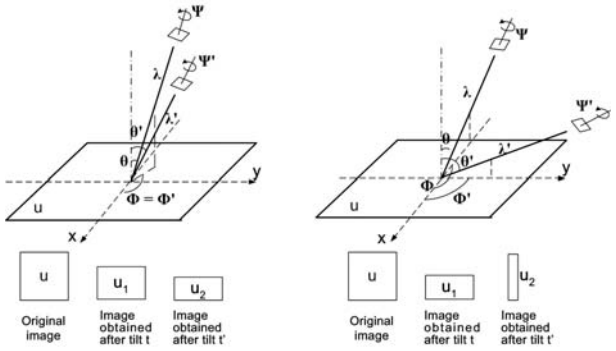


Figure 6.7: Illustration of the difference between absolute tilt and transition tilt. The small parallelograms on the top represent the cameras looking at u. Left: longitudes $\phi = \phi'$, latitudes $\theta = 30°$, $\theta' = 60°$, absolute tilts $t = 1/\cos\theta = 2/\sqrt{3}$, $t' = 1/\cos\theta' = 2$, transition tilts $\tau(u_1, u_2) = t'/t = \sqrt{3}$. Right: longitudes $\phi = \phi' + 90°$, latitudes $\theta = 60°$, $\theta' = 75.3°$, absolute tilts $t = 1/\cos\theta = 2$, $t' = 1/\cos\theta' = 4$, transition tilts $\tau(u_1, u_2) = t't = 8$.

Definition 1. *Consider two views of a planar image, $u_1(x, y) = u(A(x, y))$ and $u_2(x, y) = u(B(x, y))$ where A and B are two affine maps such that BA^{-1} is not a similarity. With the notation of* (6.2)*, we call respectively transition tilt $\tau(u_1, u_2)$ and transition rotation $\phi(u_1, u_2)$ the unique parameters such that*

$$BA^{-1} = H_\lambda R_1(\psi) T_\tau R_2(\phi). \tag{6.3}$$

One can easily check the following structure properties for the transition tilt:

- The transition tilt is symmetric, i.e., $\tau(u_1, u_2) = \tau(u_2, u_1)$;
- The transition tilt only depends on the absolute tilts and on the longitude angle difference: $\tau(u_1, u_2) = \tau(t, t', \phi - \phi')$;
- One has $t'/t \leq \tau \leq t't$, assuming $t' = \max(t', t)$;
- The transition tilt is equal to the absolute tilt: $\tau = t'$, if the other image is in frontal view ($t = 1$).

Figure 6.7 illustrates the affine transition between two images taken from different viewpoints, and in particular the difference between absolute tilt and transition tilt. On the left, the camera is first put in two positions corresponding to absolute tilts t and t' with the longitude angles $\phi = \phi'$. The transition tilt between the resulting images u_1 and u_2 is $\tau(u_1, u_2) = t'/t$. On the right the tilts are made in two orthogonal directions: $\phi = \phi' + \pi/2$. A simple calculation shows that the transition tilt between u_1 and u_2 is the product $\tau(u_1, u_2) = tt'$. Thus, *two moderate absolute tilts can lead to a large transition tilt!* Since in realistic cases the absolute tilt can go up to 6, which corresponds to a latitude angle $\theta \approx 80.5°$, the *transition tilt* can easily go up to 36. The necessity of considering high transition tilts is illustrated in Figure 6.8.

6.3 State-of-the-art

Since an affine transform depends upon six parameters, it is prohibitive to simply *simulate* all of them and compare the simulated images. An alternative way that has been tried by many authors is *normalization*. As illustrated in Fig. 6.9, normalization is a magic method that, given a patch that has undergone an unknown affine transform, transforms the patch into a standardized one that is independent of the affine transform.

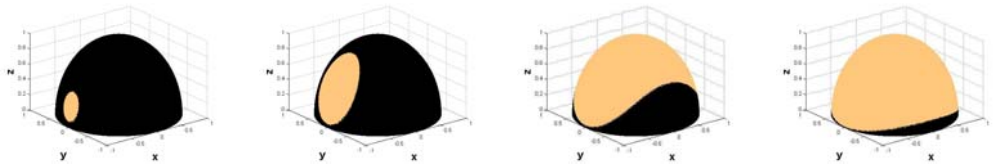


Figure 6.8: This figure illustrates the necessity of considering high transition tilts to match images under all possible views of a flat object. Two cameras look at a flat object lying in the center of the hemisphere. Their optical axes point towards the center of the hemisphere. The first camera is positioned at the center of the bright region drawn on the first hemisphere. Its latitude is $\theta = 80°$ (absolute tilt $t = 5.8$). The black regions on the four hemispheres represent the positions of the second camera for which the transition tilt between the two cameras are respectively higher than 2.5, 5, 10 and 40. Only the fourth hemisphere is almost bright, but it needs a transition tilt as large as 40 to cover it well.

Translation normalization can be easily achieved: a patch around (x_0, y_0) is translated back to a patch around $(0, 0)$. A rotation normalization requires a circular patch. In this patch, a principal direction is found, and the patch is rotated so that this principal direction coincides with a fixed direction. Thus, out of the six parameters in the affine transform, three are easily eliminated by normalization. Most state-of-the-art image matching algorithms adopt this normalization.

For the other three parameters, namely the scale and the camera axis angles, things get more difficult. This Section describes how the state-of-the-art image matching algorithms SIFT [119], MSER [135] and LLD [159, 160, 21], Harris-Affine and Hessian-Affine [145, 147] deal with these parameters.

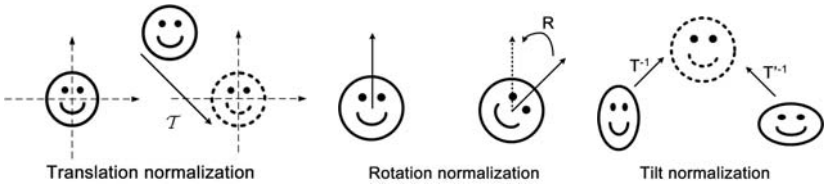


Figure 6.9: Normalization methods seek to eliminate the effect of a class of affine transforms by associating the same standard patch to all transformed patches.

6.3.1 Scale-Invariant Feature Transform (SIFT)

The initial goal of the SIFT method [119] is to compare two images (or two image parts) that can be deduced from each other (or from a common one) by a rotation, a translation and a scale change. The method turned out to be also robust to rather large changes in viewpoint angle, which explains its success.

SIFT achieves the scale invariance by *simulating* the zoom in the scale-space. Following a classical paradigm, SIFT detects stable points of interest at extrema of the Laplacian of the image in the image scale-space representation. The scale-space representation introduces a smoothing parameter σ. Images u_0 are smoothed at several scales to obtain $w(\sigma, x, y) := (G_\sigma * u_0)(x, y)$, where

$$G_\sigma(x, y) = G(\sigma, x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

is the 2D-Gaussian function with integral 1 and standard deviation σ. The notation $*$ stands for the space 2-D convolution.

Taking apart all sampling issues and several thresholds eliminating unreliable features, the SIFT detector can be summarized in one single sentence:
The SIFT method computes scale-space extrema (σ_i, x_i, y_i) of the spatial Laplacian of $w(\sigma, x, y)$, and then samples for each one of these extrema a square image patch whose origin is (x_i, y_i), whose x-direction is one of the dominant gradients around (x_i, y_i), and whose sampling rate is $\sqrt{\sigma_i^2 + \mathbf{c}^2}$, where the constant $\mathbf{c} = 0.8$ is the tentative standard deviation of the initial image blur.

The resulting samples of the digital patch at scale σ_i are encoded by the SIFT descriptor based on the gradient direction, which is invariant to nondecreasing contrast changes. This accounts for the robustness of the method to illumination changes. The fact that only local histograms of the direction of the gradient are kept explains the robustness of the descriptor

to moderate tilts. The following theorem proved in Chapter 5 confirms the experimental evidence that SIFT is almost perfectly similarity invariant.

Theorem 1. Let u and v be two digital images that are frontal snapshots of the same continuous flat image $\mathbf{u}_0$, $u = \mathbf{S}_1\mathbf{G}_\beta\mathbf{H}_\lambda\mathcal{T}\mathbf{R}\mathbf{u}_0$ and $v =: \mathbf{S}_1\mathbf{G}_\delta\mathbf{H}_\mu\mathbf{u}_0$, taken at different distances, with different Gaussian blurs and possibly different sampling rates, and up to a camera translation and rotation around its optical axe. Without loss of generality, assume $\lambda \leq \mu$. Then if the blurs are identical ($\beta = \delta = \mathbf{c}$, where the constant $\mathbf{c} = 0.8$ is the tentative standard deviation of the initial image blur), all SIFT descriptors of u are identical to SIFT descriptors of v. If $\beta \neq \delta$ (or $\beta = \delta \neq \mathbf{c}$), the SIFT descriptors of u and v become (quickly) similar when their scales grow, namely as soon as $\frac{\sigma_1}{\max(\mathbf{c},\beta)} \gg 1$ and $\frac{\sigma_2}{\max(\mathbf{c},\delta)} \gg 1$.

The extensive experiments in Section 6.6 will show that SIFT is robust to transition tilts smaller than $\tau_{\max} \approx 2$, but fails completely for larger tilts.

6.3.2 Maximally Stable Extremal Regions (MSER)

MSER [135] and LLD [159, 160, 21] try to be affine invariant by an affine normalization of the most robust image level sets and level lines. Both methods *normalize* all of the six parameters in the affine transform. We shall focus on MSER, but the discussion applies to LLD as well.

Extremal regions is the name given by the authors to the connected components of upper or lower level sets. Maximally stable extremal regions, or MSERs, are defined as maximally contrasted regions in the following way. Let $Q_1, ..., Q_{i-1}, Q_i, ...$ be a sequence of nested extremal regions $Q_i \subset Q_{i+1}$, where Q_i is defined by a threshold at level i. In other terms, Q_i is a connected component of an upper (resp. lower) level set at level i. An extremal region in the list Q_{i_0} is said to be maximally stable if the area variation $q(i) := |Q_{i+1} \setminus Q_{i-1}|/|Q_i|$ has a local minimum at i_0, where $|Q|$ denotes the area of a region

$|Q|$. Once MSERs are computed, an affine normalization is performed on the MSERs before they can be compared. Affine normalization up to a rotation is achieved by diagonalizing each MSER's second order moment matrix, and by applying the linear transform that performs this diagonalization to the MSER. Rotational invariants are then computed over the normalized region.

As pointed out in [21] MSER is not fully scale invariant. This fact is illustrated in Figure 6.10. In MSER the scale normalization is based on the size (area) of the detected extremal regions. However, scale change is not just a homothety: it involves a blur followed by subsampling. The blur merges the regions and changes their shape and size. In other terms, the limitation of the method is the non-commutation between the optical blur and the affine transform. As shown in the image formation model (6.1), the image is blurred *after* the affine transform A. The normalization procedure does not eliminate exactly the affine deformation, because $A^{-1}\mathbf{G}_1 A u_0 \neq \mathbf{G}_1 u_0$. Their difference can be considerable when the blur kernel is broad, i.e., when the image is taken with a big zoom-out or with a large tilt. This non-commutation issue is actually a limitation of all the normalization methods.

The feature sparsity is another weakness of MSER. MSER uses only highly contrasted level sets. Many natural images contain few such features. However, the experiments in Section 6.6 show that MSER is robust to transition tilts $\tau_{\max}$ between 5 and 10, a performance much higher than SIFT. But this performance is only verified when there is no substantial scale change between the images, and if the images contain highly contrasted objects.

6.3.3 Harris-Affine and Hessian-Affine

Like MSER, Harris-Affine and Hessian-Affine *normalize* all the six parameters in the affine transform. Harris-Affine [145, 147] first detects Harris key points in the scale-space

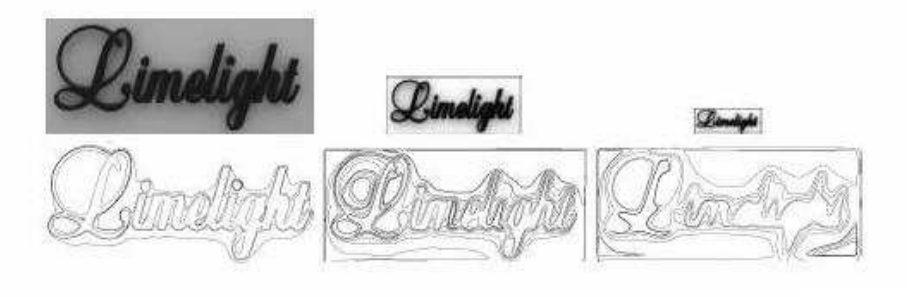


Figure 6.10: Top: the same shape at different scales. Bottom: Their level lines (shown at the same size). The level line shape changes with scale (in other terms, it changes with the camera distance to the object).

using the approach proposed by Lindeberg [111]. Then affine normalization is realized by an iterative procedure that estimates the parameters of elliptical regions and normalizes them to circular ones: at each iteration the parameters of the elliptical regions are estimated by minimizing the difference between the eigenvalues of the second order moment matrix of the selected region; the elliptical region is normalized to a circular one; the position of the key point and its scale in scale space are estimated. This iterative procedure due to [113, 6] finds an isotropic region, which is covariant under affine transforms. The eigenvalues of the second moment matrix are used to measure the affine shape of the point neighborhood. The affine deformation is determined up to a rotation factor. This factor can be recovered by other methods, for example by a normalization based on the dominant gradient orientation like in the SIFT method.

The Hessian-Affine is similar to the Harris-Affine, but the detected regions are blobs instead of corners. Local maximums of the determinant of the Hessian matrix are used as base points, and the remainder of the procedure is the same as for Harris-Affine.

As pointed out in [119], in both methods the first step, namely the multiscale Harris or Hessian detector, is clearly not affine covariant. The features resulting from the iterative procedure should instead be fully affine invariant. The experiments in Section 6.6 show that

Harris-Affine and Hessian-Affine are robust to transition tilts of maximal value $\tau_{\max} \approx 2.5$. This disappointing result may be explained by the failure of the iterative procedure to capture large transition tilts.

6.4 Affine-SIFT (ASIFT)

The idea of combining simulation and normalization is the main ingredient of the SIFT method. The SIFT detector normalizes rotations and translations, and simulates all zooms out of the query and of the search images. Because of this feature, it is the only fully scale invariant method.

As described in Figure 6.11, ASIFT simulates with enough precision all distortions caused by a variation of the camera optical axis direction. Then it applies the SIFT method. In other words, ASIFT simulates three parameters: the scale, the camera longitude angle and the latitude angle (which is equivalent to the tilt) and normalizes the other three (translation and rotation). The mathematical proof that ASIFT is *fully* affine invariance will be given in Section 6.5. The key observation is that, although a tilt distortion is irreversible due to its non-commutation with the blur, it can be compensated up to a scale change by digitally simulating a tilt of same amount in the orthogonal direction. As opposed to the *normalization* methods that suffer from this non-commutation, ASIFT *simulates* and thus achieves the full affine invariance.

Against any prognosis, simulating the whole affine space is not prohibitive at all with the proposed affine space sampling. A two-resolution scheme will further reduce the ASIFT complexity to about twice that of SIFT.

6.4.1 ASIFT Algorithm

ASIFT proceeds by the following steps.

1. Each image is transformed by simulating all possible affine distortions caused by the change of camera optical axis orientation from a frontal position. These distortions depend upon two parameters: the longitude ϕ and the latitude θ. The images undergo ϕ-rotations followed by tilts with parameter $t = \left|\frac{1}{\cos\theta}\right|$ (a tilt by t in the direction of x is the operation $u(x, y) \rightarrow u(tx, y)$). For digital images, the tilt is performed by a directional t-subsampling. It requires the previous application of an antialiasing filter in the direction of x, namely the convolution by a Gaussian with standard deviation $\mathbf{c}\sqrt{t^2 - 1}$. The value $\mathbf{c} = 0.8$ is the value chosen by Lowe for the SIFT method [119]. As shown in Chapter 5, it ensures a very small aliasing error.

2. These rotations and tilts are performed for a finite and small number of latitude and longitude angles, the sampling steps of these parameters ensuring that the simulated images keep close to any other possible view generated by other values of ϕ and θ.

3. All simulated images are compared by a similarity invariant matching algorithm (SIFT).

The sampling of the latitude and longitude angles is specified below and will be explained in detail in Section 6.4.2.

- The latitudes θ are sampled so that the associated tilts follow a geometric series 1, a, a^2, , . . . , a^n, with $a > 1$. The choice $a = \sqrt{2}$ is a good compromise between accuracy and sparsity. The value n can go up to 5 or more. In consequence transition tilts going up to 32 and more can be explored.

- The longitudes ϕ are for each tilt an arithmetic series 0, b/t, . . . , kb/t, where $b \simeq 72^\circ$ seems again a good compromise, and k is the last integer such that $kb/t < 180^\circ$.

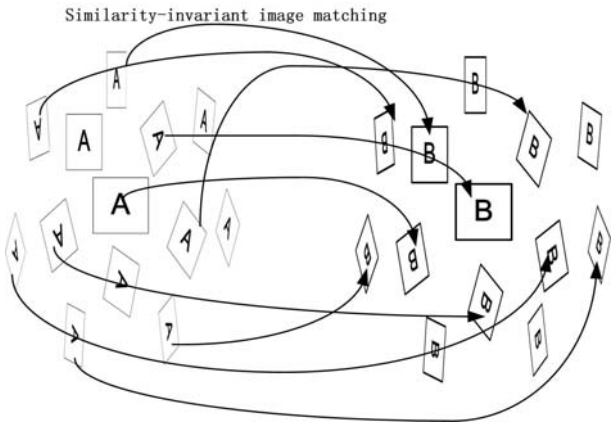


Figure 6.11: Overview of the ASIFT algorithm. The square images A and B represent the compared images **u** and **v**. ASIFT simulates all distortions caused by a variation of the camera optical axis direction. The simulated images, represented by the parallelograms, are then compared by SIFT, which is invariant to scale change, rotation and translation.

6.4.2 Latitude and Longitude Sampling

The ASIFT latitude and the longitude sampling will be determined experimentally.

Sampling Ranges

The camera motion illustrated in Figure 6.6 shows ϕ varying from 0 to 2π. But, by Theorem 2, simulating $\phi \in [0, \pi)$ is enough to cover all possible affine transforms.

The sampling range of the tilt parameter t is more critical. Object recognition under any slanted view is possible only if the object is perfectly planar and Lambertian. Since this is never the case, a practical physical upper bound $t_{\max}$ must be experimentally obtained by using image pairs taken from indoor and outdoor scenes, each image pair being composed of a frontal view and a slanted view. Two case studies were performed. The first one was a magazine placed on a table with the artificial illumination coming from the ceiling as shown in Figure 6.12. The outdoor scene was a building façade with some graffiti as illustrated in Figure 6.13. The images have 600×450 resolution. For each image pair, the

true tilt parameter t was obtained by on site measurements. ASIFT was applied with very large parameter sampling ranges and small sampling steps, thus ensuring that the actual affine distortion was accurately approximated. The ASIFT matching results of Figures 6.12 and 6.13 show that the physical limit is $t_{\max} \approx 4\sqrt{2}$ corresponding to a view angle $\theta_{\max} = \arccos 1/t_{\max} \approx 80°$. The sampling range $t_{\max} = 4\sqrt{2}$ allows ASIFT to be invariant to transition tilt as large as $(4\sqrt{2})^2 = 32$. (With higher resolution images, larger transition tilts would definitely be attainable.)

Figure 6.12: Finding the maximal attainable absolute tilt. From left to right, the tilt t between the two images is respectively $t \approx 3, 5.2, 8.5$. The number of correct ASIFT matches is respectively 151, 12, and 0.

Sampling Steps

In order to have ASIFT invariant to any affine transform, one needs to sample the tilt t and angle ϕ with a high enough precision. The sampling steps $\triangle t$ and $\triangle \phi$ must be fixed experimentally by testing several natural images.

The camera motion model illustrated in Fig. 6.6 indicates that the sampling precision of the latitude angle $\theta = \arccos 1/t$ should increase with θ: the image distortion caused by

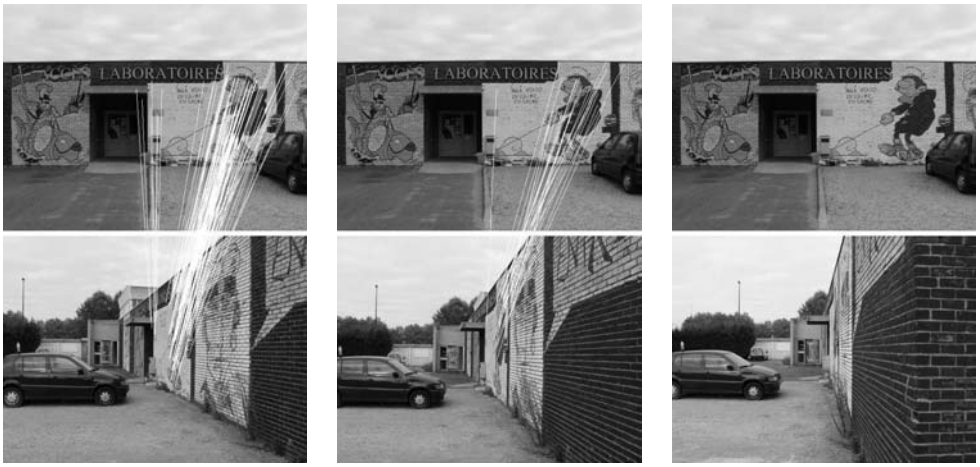


Figure 6.13: Finding the maximal attainable absolute tilt. From left to right, the absolute tilt t between the two images is respectively $t \approx 3.8, 5.6, 8$; the number of correct ASIFT matches is respectively 116, 26 and 0.

a fixed latitude angle displacement $\triangle\theta$ is more drastic at larger θ. A geometric sampling for t satisfies this requirement. Naturally, the sampling ratio $\triangle t = t_{k+1}/t_k$ should be independent of the angle ϕ. In the sequel, the tilt sampling step is experimentally fixed to $\triangle t = \sqrt{2}$.

Similarly to the latitude sampling, one needs a finer longitude ϕ sampling when $\theta = \arccos 1/t$ increases: the image distortion caused by a fixed longitude angle displacement $\triangle\phi$ is more drastic at larger latitude angle θ. The longitude sampling step in the sequel will be $\triangle\phi = \frac{72^\circ}{t}$.

The sampling steps $\triangle t = \sqrt{2}$ and $\triangle\phi = \frac{72^\circ}{t}$ were validated by applying successfully SIFT between images with simulated tilt and longitude variations equal to the sampling step values. The extensive experiments in Section 6.6 justify the choice as well. Figure 6.14 illustrates the resulting irregular sampling of the parameters $\theta = \arccos 1/t$ and ϕ on the observation hemisphere: the samples accumulate near the equator.

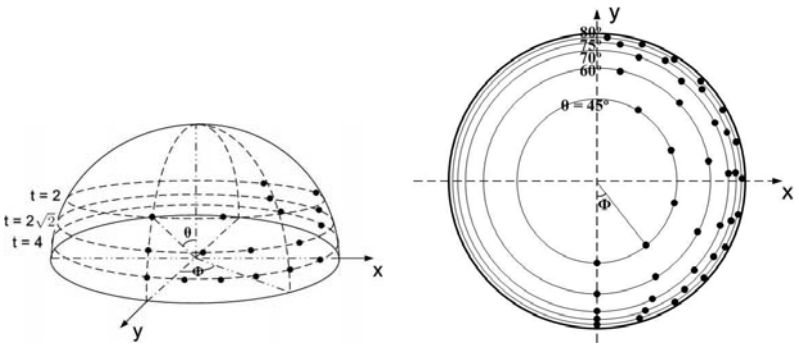


Figure 6.14: Sampling of the parameters $\theta = \arccos 1/t$ and ϕ. The samples are the black dots. Left: perspective illustration of the observation hemisphere (only $t = 2, 2\sqrt{2}, 4$ are shown). Right: zenith view of the observation hemisphere. The values of θ are indicated on the figure.

6.4.3 Acceleration with Two Resolutions

The coarse-to-fine two-resolution procedure accelerates ASIFT by applying the ASIFT method described in Section 6.4.1 on a low-resolution version of the query and the search images. In case of success, the procedure selects the affine transforms that yielded matches in the low-resolution process, then simulates the selected affine transforms on the original query and search images, and finally compares the simulated images by SIFT. The two-resolution method is summarized as follows.

1. Subsample the query and the search images $\mathbf{u}$ and $\mathbf{v}$ by a $K \times K$ factor: $\mathbf{u}' = \mathbf{S}_K \mathbf{G}_K \mathbf{u}$ and $\mathbf{v}' = \mathbf{S}_K \mathbf{G}_K \mathbf{v}$, where $\mathbf{G}_K$ is an anti-aliasing Gaussian discrete filter and $\mathbf{S}_K$ is the $K \times K$ subsampling operator.

2. Low-resolution ASIFT: apply ASIFT as described in Section 6.4.1 to $\mathbf{u}'$ and $\mathbf{v}'$.

3. Identify the M affine transforms yielding the biggest numbers of matches between $\mathbf{u}'$ and $\mathbf{v}'$.

4. High-resolution ASIFT: apply ASIFT to $\mathbf{u}$ and $\mathbf{v}$, but simulate only the M affine

transforms.

Fig. 6.15 shows an example. The low-resolution ASIFT that is applied on the $K \times K = 3 \times 3$ subsampled images finds 19 correspondences and identifies the $M = 5$ best affine transforms. The high-resolution ASIFT finds 51 correct matches.

Figure 6.15: Two-resolution ASIFT. Left: low-resolution ASIFT applied on the 3×3 subsampled images finds 19 correct matches. Right: high-resolution ASIFT finds 51 matches.

6.4.4 ASIFT Complexity

The complexity of the ASIFT method will be estimated under the recommended configuration: the tilt and angle ranges are $[t_{\min}, t_{\max}] = [1, 4\sqrt{2}]$ and $[\phi_{\min}, \phi_{\max}] = [0°, 180°]$, and the sampling steps are $\triangle t = \sqrt{2}$, $\triangle \phi = \frac{72°}{t}$. A t tilt is simulated by t times subsampling in one direction. The query and the search images are subsampled by a $K \times K = 3 \times 3$ factor for the low-resolution ASIFT. Finally, the high-resolution ASIFT simulates the M

best affine transforms that are identified, but only in case they lead to enough matches. In real applications where a query image is compared with a large database, the likely result for the low-resolution step is failure. The final high-resolution step counts only when the images matched at low resolution.

Estimating the ASIFT complexity boils down to calculate the image area simulated by the low-resolution ASIFT. Indeed the complexity of the image matching *feature computation* is proportional to the input image area. One can verify that the total image area simulated by ASIFT is proportional to the number of simulated tilts t: the number of ϕ simulations is proportional to t for each t, but the t subsampling for each tilt simulation divides the area by t. More precisely, the image area input to low-resolution ASIFT is

$$\frac{1 + (|\Gamma_t| - 1)\frac{180^\circ}{72^\circ}}{K \times K} = \frac{1 + 5 \times 2.5}{3 \times 3} = 1.5$$

times as large as that of the original images, where $|\Gamma_t| = |\{1, \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}\}| = 6$ is the number of simulated tilts and $K \times K = 3 \times 3$ is the subsampling factor. Thus the complexity of the low-resolution ASIFT feature calculation is 1.5 times as much as that of a single SIFT routine. The ASIFT algorithm in this configuration is invariant to transition tilts up to 32. Higher transition tilt invariance is attainable with larger $t_{\max}$. The complexity growth is *linear* and thus marginal with respect to the *exponential* growth of transition tilt invariance.

Low-resolution ASIFT simulates 1.5 times the area of the original images and generates in consequence about 1.5 times more features on both the query and the search images. The complexity of low-resolution ASIFT *feature comparison* is therefore $1.5^2 = 2.25$ times as much as that of SIFT.

If the image comparisons involve a large database where most comparisons will be failures, ASIFT stops essentially at the end of the low-resolution procedure, and the overall complexity is about twice the SIFT complexity, as argued above.

If the comparisons involve a set of images with high matching likeliness, then the high resolution step is no more negligible. The overall complexity of ASIFT depends on the number M of the identified good affine transforms simulated in the high-resolution procedure as well as on the simulated tilt values t. However, in that case, ASIFT ensures many more detections than SIFT, because it explores many more viewpoint angles. In that case the *complexity rate per match detection* is in practice equal to or smaller than the per match detection complexity of a SIFT routine.

The SIFT subroutines can be implemented in parallel in ASIFT (for both the low-resolution and the high-resolution ASIFT). Recently many authors have investigated SIFT accelerations [93, 67, 103]. A realtime SIFT implementation has been proposed in [181]. Obviously all the SIFT acceleration techniques directly apply to ASIFT.

6.5 The Mathematical Justification

This Section proves mathematically that ASIFT is fully affine invariant, up to sampling errors. The key observation is that a tilt can be compensated up to a scale change by another tilt of the same amount in the orthogonal direction.

The proof is given in a continuous setting which is by far simpler, because the image sampling does not interfere. Since the digital images are assumed to be well-sampled, the Shannon interpolation (obtained by zero-padding) paves the way from discrete to continuous.

To lighten the notation, G_σ will also denote the convolution operator on $\mathbb{R}^2$ with the Gauss kernel $\mathrm{G}_\sigma(x,y) = \frac{1}{2\pi(\mathbf{c}\sigma)^2} e^{-\frac{x^2+y^2}{2(\mathbf{c}\sigma)^2}}$, namely $\mathrm{G}u(x,y) := (\mathrm{G} * u)(x,y)$, where the constant $\mathbf{c} = 0.8$ is chosen for good anti-aliasing [119, 154]. The one-dimensional Gaussians will be denoted by $\mathrm{G}^x_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\mathbf{c}\sigma} e^{-\frac{x^2}{2(\mathbf{c}\sigma)^2}}$ and $\mathrm{G}^y_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\mathbf{c}\sigma} e^{-\frac{y^2}{2(\mathbf{c}\sigma)^2}}$. G_σ satisfies the

semigroup property

$$\mathrm{G}_\sigma \mathrm{G}_\beta = \mathrm{G}_{\sqrt{\sigma^2+\beta^2}} \tag{6.4}$$

and it commutes with rotations:

$$\mathrm{G}_\sigma R = R\mathrm{G}_\sigma. \tag{6.5}$$

We shall denote by $*_y$ the 1-D convolution operator in the y-direction. In the notation $\mathrm{G}*_y$, G is a one-dimensional Gaussian depending on y and

$$\mathrm{G} *_y u(x,y) := \int \mathrm{G}^y(z) u(x, y-z) dz.$$

6.5.1 Inverting Tilts

Let us distinguish two tilting procedures:

Definition 2. *Given $t > 1$, the tilt factor, define*

- *the geometric tilt : $T_t^x u_0(x,y) := u_0(tx, y)$. In case this tilt is made in the y direction, it will be denoted by $T_t^y u_0(x,y) := u_0(x, ty)$;*
- *the simulated tilt (taking into account camera blur): $\mathbb{T}_t^x v := T_t^x \mathrm{G}^x_{\sqrt{t^2-1}} *_x v$. In case the simulated tilt is done in the y direction, it is denoted $\mathbb{T}_t^y v := T_t^y \mathrm{G}^y_{\sqrt{t^2-1}} *_y v$.*

As described by the image formation model (6.1), an infinite resolution scene u_0 observed from a slanted view in the x direction is distorted by a *geometric* tilt before it is blurred by the optical lens, i.e., $u = \mathrm{G}_1 T_t^x u_0$. Reversing this operation is in principle impossible, because of the tilt and blur non-commutation. However, the next lemma shows that a *simulated* tilt $\mathbb{T}_t^y$ in the orthogonal direction provides actually a pseudo inverse to the *geometric* tilt T_t^x.

Lemma 4. $\mathbb{T}_t^y = H_t \mathrm{G}^y_{\sqrt{t^2-1}} *_y (T_t^x)^{-1}$.

Proof. Since $(T_t^x)^{-1}u(x,y) = u(\frac{x}{t}, y)$,

$$\left(\mathrm{G}_{\sqrt{t^2-1}} *_y (T_t^x)^{-1}u\right)(x,y) = \int \mathrm{G}_{\sqrt{t^2-1}}(z)u(\frac{x}{t}, y-z)dz.$$

Thus

$$H_t\left(\mathrm{G}_{\sqrt{t^2-1}} *_y (T_t^x)^{-1}u\right)(x,y) = \int \mathrm{G}_{\sqrt{t^2-1}}(z)u(x, ty-z)dz =$$

$$\left(\mathrm{G}^y_{\sqrt{t^2-1}} *_y u\right)(x,ty) = \left(T_t^y \mathrm{G}^y_{\sqrt{t^2-1}} *_y u\right)(x,y).$$

□

By the next Lemma, a tilted image $\mathrm{G}_1 T_t^x u$ can be tilted back by tilting in the orthogonal direction. The price to pay is a t zoom out. The second relation in the lemma means that the application of the *simulated* tilt to a well-sampled image yields an image that keeps the well-sampling property. This fact is crucial to simulate tilts on digital images.

Lemma 5. *Let $t \geq 1$. Then*

$$\mathbb{T}_t^y(\mathrm{G}_1 T_t^x) = \mathrm{G}_1 H_t; \tag{6.6}$$

$$\mathbb{T}_t^y \mathrm{G}_1 = \mathrm{G}_1 T_t^y. \tag{6.7}$$

Proof. By Lemma 4, $\mathbb{T}_t^y = H_t \mathrm{G}^y_{\sqrt{t^2-1}} *_y (T_t^x)^{-1}$. Thus,

$$\mathbb{T}_t^y(\mathrm{G}_1 T_t^x) = H_t \mathrm{G}^y_{\sqrt{t^2-1}} *_y ((T_t^x)^{-1}\mathrm{G}_1 T_t^x). \tag{6.8}$$

By a variable change in the integral defining the convolution, it is an easy check that

$$(T_t^x)^{-1}\mathrm{G}_1 T_t^x u = \left(\frac{1}{t}\mathrm{G}_1(\frac{x}{t}, y)\right) * u, \tag{6.9}$$

and by the separability of the 2D Gaussian in two 1D Gaussians,

$$\frac{1}{t}\mathrm{G}_1(\frac{x}{t}, y) = \mathrm{G}_t(x)\mathrm{G}_1(y). \tag{6.10}$$

¿From (6.9) and (6.10) one obtains

$$(T^x)^{-1}\mathrm{G}_1 T_t^x u = ((\mathrm{G}_t^x(x)\mathrm{G}_1^y(y)) * u = \mathrm{G}_t^x(x) *_x \mathrm{G}_1^y(y) *_y u,$$

which implies

$$\mathrm{G}^y_{\sqrt{t^2-1}} *_y (T^x)^{-1}\mathrm{G}_1 T_t^x u = \mathrm{G}^y_{\sqrt{t^2-1}} *_y (\mathrm{G}_t^x(x) *_x \mathrm{G}_1^y(y) *_y u) = \mathrm{G}_t u.$$

Indeed, the 1D convolutions in x and y commute and $\mathrm{G}^y_{\sqrt{t^2-1}} * \mathrm{G}_1^y = \mathrm{G}_t^y$ by the Gaussian semigroup property (6.4). Substituting the last proven relation in (6.8) yields

$$\mathbb{T}_t^y \mathrm{G}_1 T_t^x u = H_t \mathrm{G}_t u = \mathrm{G}_1 H_t u.$$

The second relation (6.7) follows immediately by noting that $H_t = T_t^y T_t^x$. □

6.5.2 Proof that ASIFT works

The meaning of Lemma 5 is that we can design an exact algorithm that simulates all inverse tilts, up to scale changes.

Theorem 3. *Let $u = \mathrm{G}_1 A \mathcal{T}_1 u_0$ and $v = \mathrm{G}_1 B \mathcal{T}_2 u_0$ be two images obtained from an infinite resolution image u_0 by cameras at infinity with arbitrary position and focal lengths. (A and B are arbitrary affine maps with positive determinants and $\mathcal{T}_1$ and $\mathcal{T}_2$ arbitrary planar translations.) Then ASIFT, applied with a dense set of tilts and longitudes, simulates two views of u and v that are obtained from each other by a translation, a rotation, and a camera zoom. As a consequence, these images match by the SIFT algorithm.*

Proof. We start by giving a formalized version of ASIFT using the above notation.

(Dense) ASIFT

1. Apply a dense set of rotations to both images u and v.

2. Apply in continuation a dense set of *simulated* tilts $\mathbb{T}_t^x$ to all rotated images.

3. Perform a SIFT comparison of all pairs of resulting images.

Notice that by the relation

$$\mathbb{T}_t^x R(\frac{\pi}{2}) = R(\frac{\pi}{2})\mathbb{T}_t^y, \tag{6.11}$$

the algorithm also simulates tilts in the y direction, up to a $R(\frac{\pi}{2})$ rotation.

By the affine decomposition (6.2),

$$BA^{-1} = H_\lambda R_1 T_t^x R_2. \tag{6.12}$$

The *dense* ASIFT applies in particular:

1. $\mathbb{T}^x_{\sqrt{t}} R_2$ to $\mathrm{G}_1 A\mathcal{T}_1 u_0$, which by (6.5) and (6.7) yields $\tilde{u} = \mathrm{G}_1 T^x_{\sqrt{t}} R_2 A\mathcal{T}_1 u_0 := \mathrm{G}_1 \tilde{A}\mathcal{T}_1 u_0$.

2. $R(\frac{\pi}{2})\mathbb{T}^y_{\sqrt{t}} R_1^{-1}$ to $\mathrm{G}_1 B\mathcal{T}_2 u_0$, which by (6.5) and (6.7) yields $\mathrm{G}_1 R(\frac{\pi}{2}) T^y_{\sqrt{t}} R_1^{-1} B\mathcal{T}_2 u_0 := \mathrm{G}_1 \tilde{B}\mathcal{T}_2 u_0$.

Let us show that $\tilde{A}$ and $\tilde{B}$ only differ by a similarity. Indeed,

$$\tilde{B}^{-1} R(\frac{\pi}{2}) H_{\sqrt{t}} \tilde{A} = B^{-1} R_1 T^y_{\sqrt{t}^{-1}} T^x_{\sqrt{t}} H_{\sqrt{t}} R_2 A = B^{-1} R_1 T_t^x R_2 A = B^{-1}(H_{\frac{1}{\lambda}} BA^{-1})A = H_{\frac{1}{\lambda}}.$$

It follows that $\tilde{B} = R(\frac{\pi}{2}) H_{\lambda\sqrt{t}} \tilde{A}$. Thus,

$$\tilde{u} = \mathrm{G}_1 \tilde{A}\mathcal{T}_1 u_0 \ \text{ and } \ \tilde{v} = \mathrm{G}_1 R(\frac{\pi}{2}) H_{\lambda\sqrt{t}} \tilde{A}\mathcal{T}_2 u_0$$

are two of the images simulated by ASIFT, and are deduced from each other by a rotation and a $\lambda\sqrt{t}$ zoom. It follows from Theorem 1 that their descriptors are identical as soon as the scale of the descriptors exceeds $\lambda\sqrt{t}$. □

Remark 1. *The above proof gives the value of the simulated tilts achieving success: if the transition tilt between u and v is t, then it is enough to simulate a $\sqrt{t}$ tilt on both images.*

6.5.3 Algorithmic Sampling Issues

Although the above proof deals with asymptotic statements when the sampling steps tend to zero or when the SIFT scales tend to infinity, the approximation rate is quick, a fact that can only be checked experimentally. This fact is actually extensively verified by the huge amount of experimental evidence on SIFT, that shows first that the recognition of scale invariant features is robust to a rather large latitude and longitude variation, and second that the scale invariance is quite robust to moderate errors on scale. Section 6.4.2 has evaluated the adequate sampling rates and ranges for tilts and longitudes.

The above algorithmic description has neglected the image sampling issues, but care was taken that input images and output images be always written in the $\mathrm{G}_1 u$ form. For the digital input images, which always have the form $\mathbf{u} = \mathbf{S}_1 \mathrm{G}_1 u_0$, the Shannon interpolation algorithm I is first applied, to give back $I\mathbf{S}_1 \mathrm{G}_1 u_0 = \mathrm{G}_1 u_0$. For the output images, which always have the form $\mathrm{G}_1 u$, the sampling $\mathbf{S}_1$ gives back a digital image.

6.6 Experiments

ASIFT image matching performance will be compared with the state-of-the-art approaches using the detectors SIFT [119], MSER [135], Harris-Affine, and Hessian-Affine [144, 147], all combined with the most popular SIFT descriptor [119]. The MSER detector combined with the correlation descriptor as proposed in the original work [135] was initially included in the comparison, but its performance was found to be slightly inferior to that of the MSER detector combined by the SIFT descriptor, as indicated in [146]. Thus only the latter will be shown. In the following, the methods will be named after their detectors, namely ASIFT, SIFT, MSER, Harris-Affine and Hessian-Affine.

The experiments include extensive tests with the standard Mikolajczyk database [143],

a systematic evaluation of methods' invariance to absolute and transition tilts and other images of various types (resolution 600×450).

In the experiments the Lowe [117] reference software was used for SIFT. For all the other methods we used the binaries of the MSER, the Harris-Affine and the Hessian-Affine detectors and the SIFT descriptor provided by the authors, all downloadable from [143].

The low-resolution ASIFT applied a 3×3 image subsampling. ASIFT may detect repeated matches from the image pairs simulated with different affine transforms. All the redundant matches have been removed. (A match between two points p_1 and p_2 was considered redundant with a match between p_3 and p_4 if $\mathbf{d}^2(p_1, p_3) < 3$ and $\mathbf{d}^2(p_2, p_4) < 3$, where $\mathbf{d}(p_i, p_j)$ denotes the Euclidean distance between p_i and p_j.)

6.6.1 Standard Test Database

The standard Mikolajczyk database [143] was used to evaluate the methods' robustness to four types of distortions, namely blur, similarity, viewpoint change, and jpeg compression. Five image pairs (image 1 vs images 2 to 6) with increasing amount of distortion were used for each test. Figure 6.16 illustrates the number of correct matches achieved by each method. For each method, the number of image pairs m on which more than 20 correct matches are detected and the average number of matches n over these m pairs are shown for each test. Among the methods under comparison, ASIFT is the only one that works well for the entire database. It also systematically finds more correct matches. More precisely:

- **Blur.** ASIFT and SIFT are very robust to blur, followed by Harris-Affine and Hessian-Affine. MSER are not robust to blur.
- **Zoom plus rotation**. ASIFT and SIFT are very robust to zoom plus rotation, while MSER, Harris-Affine and Hessian-Affine have limited robustness, as explained in Section 6.3.

- **Viewpoint change.** ASIFT is very robust to viewpoint change, followed by MSER. On average ASIFT find 20 times more matches than MSER. SIFT, Harris-Affine and Hessian-Affine have comparable performance: they fail when the viewpoint change is substantial.

  The test images (see Figure 6.17) provided optimal conditions for MSER: the camera-object distances are similar, and well contrasted shapes are always present.

- **Compression.** All considered methods are very robust to JPEG compression.

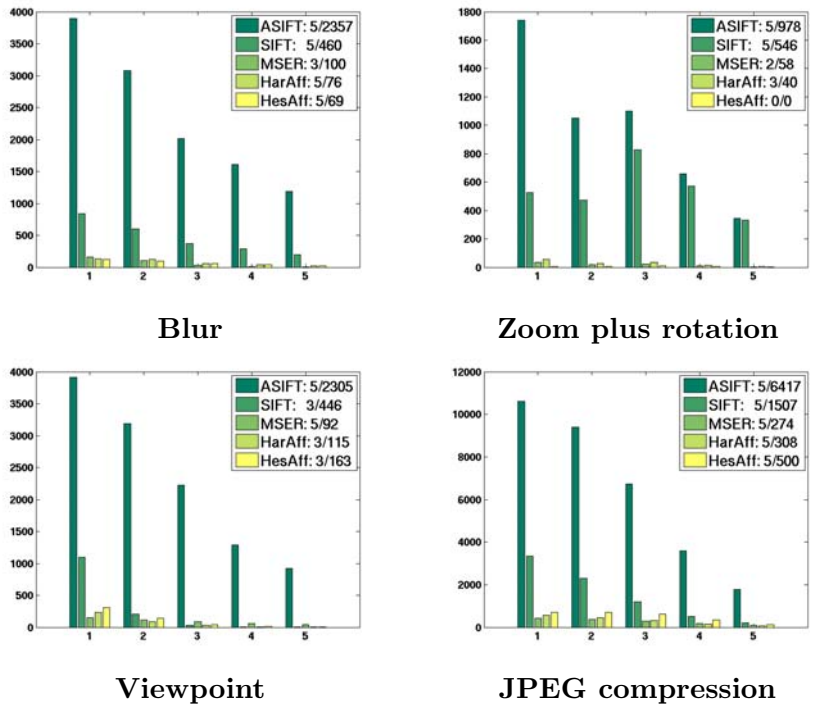


Figure 6.16: Number of correct matches achieved by ASIFT, SIFT, MSER, Harris-Affine, and Hessian-Affine under four types of distortions, namely blur, zoom plus rotation, viewpoint change and jpeg compression, in the standard Mikolajczyk database. On the top-right corner of each graph m/n gives for each method the number of image pairs m on which more than 20 correct matches were detected, and the average number of matches n over these m pairs.

Fig. 6.17 shows the classic image pair Graffiti 1 and 6. ASIFT finds 925 correct matches. SIFT, Harris-Affine and Hessian-Affine find respectively 0, 3 and 1 correct matches: the

$\tau \approx 3.2$ transition tilt is just a bit too large for these methods. MSER finds 42 correct correspondences.

Figure 6.17: Two Graffiti images with transition tilt $\tau \approx 3.2$. ASIFT (shown), SIFT (shown), Harris-Affine, Hessian-Affine and MSER(shown) find 925, 2, 3, 1 and 42 correct matches.

The next sections describe more systematic evaluations of the robustness to absolute and transition tilts of the compared methods. The normalization methods MSER, Harris-Affine, and Hessian-Affine have been shown to fail under large scale changes (see another example in Fig. 6.18). To focus on tilt invariance, the experiments will therefore take image pairs with similar scales.

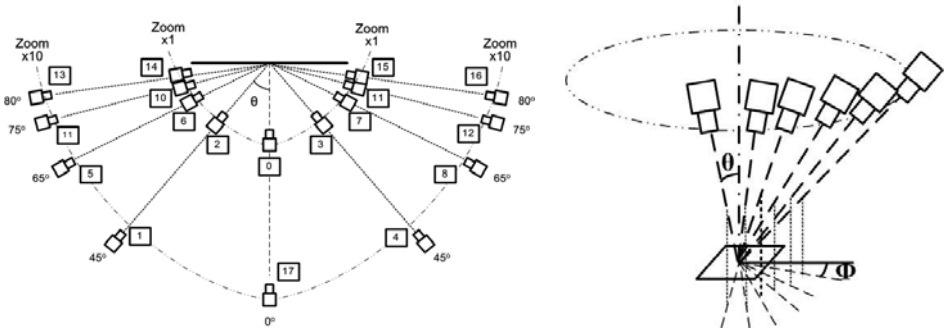
6.6.2 Absolute Tilt Tests

Figure 6.19-a illustrates the experimental setting. The painting illustrated in Figure 6.20 was photographed with an optical zoom varying between $\times 1$ and $\times 10$ and with viewpoint angles between the camera axis and the normal to the painting varying from 0° (frontal view) to 80°. It is clear that beyond 80°, to establish a correspondence between

Figure 6.18: Robustness to scale change. ASIFT (shown), SIFT (shown), Harris-Affine (shown), Hessian-Affine, and MSER find respectively 221, 86, 4, 3 and 4 correct matches. Harris-Affine, Hessian-Affine and MSER are not robust to scale change.

the frontal image and the extreme viewpoint becomes haphazard. With such a big change of view angle on a reflective surface, the image in the slanted view can be totally different from the frontal view.

Table 6.1 summarizes the performance of each algorithm in terms of number of correct matches. Some matching results are illustrated in Figures 6.22 to 6.23. MSER, which uses maximally stable level sets as features, obtains most of the time many less correspondences than the methods whose features are based on local maxima in the scale-space. As depicted in Fig. 6.21, for images taken at a short distance (zoom $\times 1$) the tilt varies on the same flat object because of the perspective effect, an example being illustrated in Fig. 6.22. The number of SIFT correspondences drops dramatically when the angle is larger than 65° (tilt $t \approx 2.3$) and it fails completely when the angle exceeds 75° (tilt $t \approx 3.8$). At 75°, as shown in Figure 6.22, most SIFT matches are located on the side closer to the camera where the actual tilt is actually smaller. The performance of Harris-Affine and Hessian-Affine decays



a b

Figure 6.19: The settings adopted for systematic comparison. Left: absolute tilt test. An object is photographed with a latitude angle varying from 0° (frontal view) to 80°, from distances varying between 1 and 10, which is the maximum focus distance change. Right: transition tilt test. An object is photographed with a longitude angle ϕ that varies from 0° to 90°, from a fixed distance.

Figure 6.20: The painting (left) and the magazine cover (right) that were photographed in the absolute and transition tilt tests.

considerably when the angle goes over 75° (tilt $t \approx 3.8$). The MSER correspondences are always fewer and show a noticeable decline over 65° (tilt $t \approx 2.4$). ASIFT works until 80° (tilt $t \approx 5.8$).

Consider now images taken at a camera-object distance multiplied by 10, as shown in Figure 6.23. For these images the SIFT performance drops considerably: recognition is possible only with angles smaller than 45°. The performance of Harris-Affine and Hessian-Affine declines steeply when the angle goes from 45° to 65°. Beyond 65° they fail completely. MSER struggles at the angle of 45° and fails at 65°. ASIFT functions perfectly until 80°.

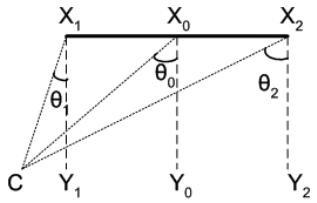


Figure 6.21: When the camera focus distance is small, the absolute tilt of a plane object can vary considerably in the same image due to the strong perspective effect.

Rich in highly contrasted regions, the magazine shown in Figure 6.20 is more favorable to MSER. Table 6.2 shows the result of a similar experiment performed with the magazine, with the latitude angles from 50 to 80° on one side and with the camera focus distance $\times 4$. Figure 6.24 shows the result with 80° angle. The performance of SIFT, Harris-Affine and Hessian-Affine drops steeply with the angle going from 50 to 60° (tilt t from 1.6 to 2). Beyond 60° (tilt $t = 2$) they fail completely. MSER finds many correspondences until 70° (tilt $t \approx 2.9$). The number of correspondences drops when the angle exceeds 70° and becomes too small at 80° (tilt $t \approx 5.8$) for robust recognition. ASIFT works until 80°.

The above experiments suggest an estimate of the maximal absolute tilts for the method under comparison. For SIFT, this limit is hardly above 2. The limit is about 2.5 for Harris-Affine and Hessian-Affine. The performance of MSER depends on the type of image. For

Figure 6.22: Correspondences between the painting images taken from short distance (zoom $\times 1$) at frontal view and at 75° angle. The local absolute tilt varies: $t \approx 4$ (middle), $t < 4$ (right part), $t > 4$ (left part). ASIFT (shown), SIFT (shown), Harris-Affine, Hessian-Affine, and MSER (shown) find respectively 202, 15, 3, 1, and 5 correct matches.

Figure 6.23: Correspondences between long distance views (zoom $\times 10$), frontal view and 80° angle, absolute tilt $t \approx 5.8$. ASIFT (shown), SIFT, Harris-Affine (shown), Hessian-Affine, and MSER (shown) find respectively 116, 1, 1, 0, and 2 correct matches.

Z × 1					
θ/t	SIFT	HarAff	HesAff	MSER	ASIFT
$-80°/5.8$	1	16	1	4	110
$-75°/3.9$	24	36	7	3	281
$-65°/2.3$	117	43	36	5	483
$-45°/1.4$	245	83	51	13	559
$45°/1.4$	195	86	26	12	428
$65°/2.4$	92	58	32	11	444
$75°/3.9$	15	3	1	5	202
$80°/5.8$	2	6	6	5	204
Z × 10					
θ/t	SIFT	HarAff	HesAff	MSER	ASIFT
$-80°/5.8$	1	1	0	2	116
$-75°/3.9$	0	3	0	6	265
$-65°/2.3$	10	22	16	10	542
$-45°/1.4$	182	68	45	19	722
$45°/1.4$	171	54	26	15	707
$65°/2.4$	5	12	5	6	468
$75°/3.9$	2	1	0	4	152
$80°/5.8$	3	0	0	2	110

Table 6.1: Absolute tilt invariance comparison with photographs of the painting in Figure 6.20. Number of correct matches of ASIFT, SIFT, Harris-Affine (HarAff), Hessian-Affine (HesAff), and MSER for viewpoint angles between 45° and 80°. Top: images taken with zoom ×1. Bottom: images taken with zoom ×10. The latitude angles and the absolute tilts are listed in the left column. For the ×1 zoom, strong perspective effect is present and the tilts shown are average values.

images with highly contrasted regions, MSER reaches a 5 absolute tilt. However, if the images do not contain highly contrasted regions, the performance of MSER can drop under small tilts. For ASIFT, a 5.8 absolute tilt that corresponds to an extreme viewpoint angle of 80° is easily attainable.

6.6.3 Transition Tilt Tests

The magazine shown in Figure 6.20 was placed face-up and photographed to obtain two sets of images. As illustrated in Figure 6.19-b, for each image set the camera with a

θ/t	SIFT	HarAff	HesAff	MSER	ASIFT
50°/1.6	267	131	144	150	1692
60°/2.0	20	29	39	117	1012
70°/2.9	1	2	2	69	754
80°/5.8	0	0	0	17	349

Table 6.2: Absolute tilt invariance comparison with photographs of the magazine cover (Figure 6.20). Number of correct matches of ASIFT, SIFT, Harris-Affine (HarAff), Hessian-Affine (HesAff), and MSER for viewpoint angles between 50 and 80°. The latitude angles and the absolute tilts are listed in the left column.

Figure 6.24: Correspondences between magazine images taken with zoom ×4, frontal view and 80° angle, absolute tilt $t \approx 5.8$. ASIFT (shown), SIFT (shown), Harris-Affine, Hessian-Affine, and MSER (shown) find respectively 349, 0, 0, 0, and 17 correct matches.

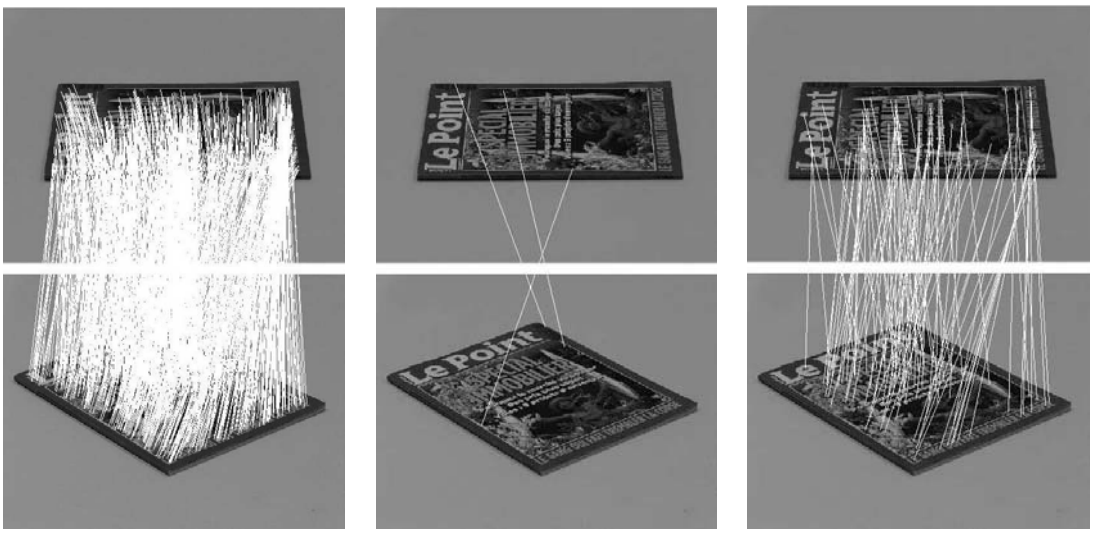


Figure 6.25: Correspondences between the magazine images taken with absolute tilts $t_1 = t_2 = 2$ with longitude angles $\phi_1 = 0°$ and $\phi_2 = 50°$, transition tilt $\tau \approx 3$. ASIFT (shown), SIFT (shown), Harris-Affine, Hessian-Affine and MSER (shown) find respectively 745, 3, 1, 3, 87 correct matches.

fixed latitude angle θ corresponding to $t = 2$ and 4 circled around, the longitude angle ϕ growing from 0 to 90°. The camera focus distance and the optimal zoom was $\times 4$. In each set the resulting images have the same absolute tilt $t = 2$ or 4, while the transition tilt τ (with respect to the image taken at $\phi = 0°$) goes from 1 to $t^2 = 4$ or 16 when ϕ goes from 0 to 90°. To evaluate the maximum invariance to transition tilt, the images taken at $\phi \neq 0$ were matched against the one taken at $\phi = 0$.

Table 6.3 compares the performance of the algorithms. When the absolute tilt is $t = 2$, the SIFT performance drops dramatically when the transition tilt goes from 1.3 to 1.7. With a transition tilt over 2.1, SIFT fails completely. Similarly a considerable performance decline is observed for Harris-Affine and Hessian-Affine when the transition tilt goes from 1.3 to 2.1. Hessian-Affine slightly outperforms Harris-Affine, but both methods fail completely when the transition tilt goes above 3. Figure 6.25 shows an example that SIFT, Harris-Affine and Hessian-Affine fail completely under a moderate transition tilt $\tau \approx 3$. MSER and ASIFT

work stably up to a 4 transition tilt. ASIFT finds ten times as many correspondences as MSER covering a much larger area.

Under an absolute tilt $t = 4$, SIFT, Harris-Affine and Hessian-Affine struggle at a 1.9 transition tilt. They fail completely when the transition tilt gets bigger. MSER works stably until a 7.7 transition tilt. Over this value, the number of correspondences is too small for reliable recognition. ASIFT works perfectly up to the 16 transition tilt. The above experiments show that the maximum transition tilt, about 2 for SIFT and 2.5 for Harris-Affine and Hessian-Affine, is by far insufficient. This experiment and others confirm that MSER ensures a reliable recognition until a transition tilt of about 10, but this is only true when the images under comparison are free of scale change and contain highly contrasted regions. The experimental limit transition tilt of ASIFT goes easily up to 36 (see Figure 6.2).

6.6.4 Other Test Images

ASIFT, SIFT, MSER, Harris-Affine and Hessian-Affine will be now tried with various classic test images and some new ones. Proposed by Matas et al. in their online demo [136] as a standard image to test MSER [135], the images in Figure 6.26 show a number of containers placed on a desktop [1]. ASIFT, SIFT, Harris-Affine, Hessian-Affine and MSER find respectively 255, 10, 23, 11 and 22 correct correspondences. Figure 6.27 contains two orthogonal road signs taken under a view change that makes a transition tilt $\tau \approx 2.6$. ASIFT successfully matches the two signs finding 50 correspondences while all the other methods totally fail. The pair of aerial images of Pentagon shown in Figure 6.28 shows a moderate transition tilt $\tau \approx 2.5$. ASIFT works perfectly by finding 378 correct matches, followed by MSER that finds 17. Harris-Affine, Hessian-Affine and SIFT fail by finding respectively

[1] We thank Tinne Tuytelaars for having kindly provided us with the images [192].

$\mathbf{t_1 = t_2 = 2}$					
ϕ_2/τ	SIFT	HarAff	HesAff	MSER	ASIFT
10°/1.3	408	233	176	124	1213
20°/1.7	49	75	84	122	1173
30°/2.1	5	24	32	103	1048
40°/2.5	3	13	29	88	809
50°/3.0	3	1	3	87	745
60°/3.4	2	0	1	62	744
70°/3.7	0	0	0	51	557
80°/3.9	0	0	0	51	589
90°/4.0	0	0	1	56	615
$\mathbf{t_1 = t_2 = 4}$					
ϕ_2/τ	SIFT	HarAff	HesAff	MSER	ASIFT
10°/1.9	22	32	14	49	1054
20°/3.3	4	5	1	39	842
30°/5.3	3	2	1	32	564
40°/7.7	0	0	0	28	351
50°/10.2	0	0	0	19	293
60°/12.4	1	0	0	17	145
70°/14.3	0	0	0	13	90
80°/15.6	0	0	0	12	106
90°/16.0	0	0	0	9	88

Table 6.3: Transition tilt invariance comparison (object photographed: the magazine cover shown in Figure 6.20). Number of correct matches of ASIFT, SIFT, Harris-Affine (HarAff), Hessian-Affine (HesAff), and MSER for viewpoint angles between 50 and 80°. The affine parameters of the two images are $\phi_1 = 0°$, $t_1 = t_2 = 2$ (above), $t_1 = t_2 = 4$ (below). ϕ_2 and the transition tilts τ are in the left column.

6, 2 and 8 matches. The Statue of Liberty shown in Figure 6.29 presents a strong relief effect. ASIFT finds 22 good matches. The other methods fail completely. Figure 6.30 shows some deformed cloth (images from [114, 115]). ASIFT outperforms significantly the other methods by finding respectively 141 and 370 correct matches, followed by SIFT that finds 31 and 75 matches. Harris-affine, Hessian-affine and MSER do not get a significant number of matches.

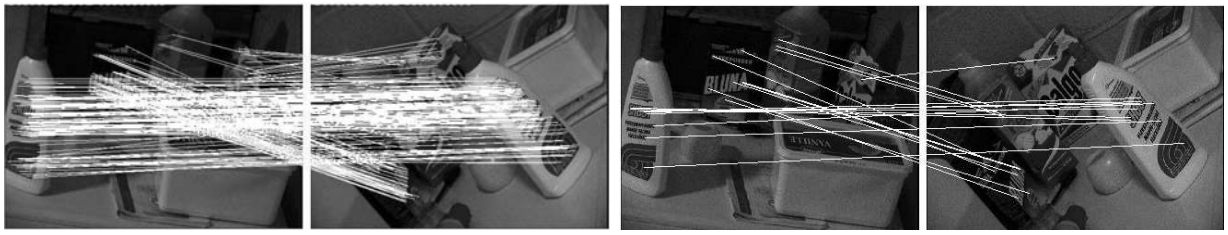


Figure 6.26: Image matching (images used by Matas et al [136]). Transition tilt: $\tau \in [1.6, 3.0]$. From top to bottom, left to right: ASIFT (shown), SIFT, Harris-Affine, Hessian-Affine and MSER (shown) find respectively 254, 10, 23, 11 and 22 correct matches.

Figure 6.27: Image matching: road signs. Transition tilt $\tau \approx 2.6$. ASIFT (shown), SIFT, Harris-Affine, Hessian-Affine and MSER (shown) find respectively 50, 0, 0, 0 and 1 correct matches.

Figure 6.28: Pentagon, with transition tilt $\tau \approx 2.5$. ASIFT (shown), SIFT (shown), Harris-Affine, Hessian-Affine and MSER(shown) find respectively 378, 6, 2, 8 and 17 correct matches.

Figure 6.29: Statue of Liberty, with transition tilt $\tau \in [1.3, \infty)$. ASIFT (shown), SIFT (shown), Harris-Affine, Hessian-Affine and MSER find respectively 22, 1, 0, 0 and 0 correct matches.

Figure 6.30: Image matching with object deformation. Left: flag. ASIFT (shown), SIFT, Harris-Affine, Hessian-Affine and MSER find respectively 141, 31, 15, 10 and 2 correct matches. Right: SpongeBob. ASIFT (shown), SIFT, Harris-Affine, Hessian-Affine and MSER find respectively 370, 75, 8, 6 and 4 correct matches.

Part III

Visual Grouping by Neural Oscillators

Chapter 7

Visual Grouping by Neural Oscillators

In the first part of the thesis, coefficient grouping is applied to improve signal estimation in sparse representations. In computer vision, visual grouping is an important tool as well. This Chapter introduces a biologically plausible visual grouping implementation with neural oscillators and shows its applications on point clustering, contour integration and image segmentation. Based on the framework of concurrent synchronization of dynamical systems, simple networks of neural oscillators are constructed. The oscillators are connected with diffusive coupling appropriately tune so that synchronization of oscillators within each group indicates perceptual grouping of the underlying stimulative atoms, while desynchronization between groups corresponds to group segregation.

7.1 Introduction

Let us consider Figure 7.1. Why do we perceive in these visual stimuli a cluster of points, a straight contour and a hurricane? How is the identification achieved between

atomic stimuli and the perceived objects?

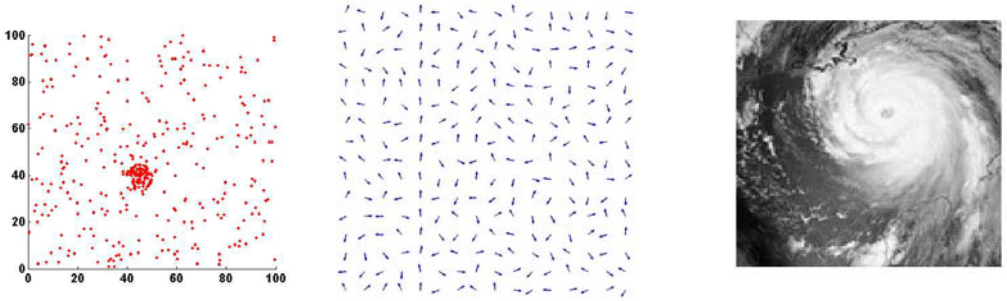


Figure 7.1: Left: a cloud of points in which a dense cluster is embedded. Middle: a random direction grid in which a vertical contour is embedded. Right: an image in which a hurricane is embedded.

Gestalt psychology [210, 140, 92, 48] proposes some visual grouping principles such as proximity, good continuation and color constancy that describe the construction of larger groups from atomic local information in the stimuli. In computer vision, various mathematical frameworks have been investigated to implement visual grouping [179, 43, 127, 57, 47, 157, 153, 74, 193, 165].

In the brain the distributed neural *synchronization* has been proposed as a general functional mechanism for perceptual grouping [15, 180, 209]. Among a relative small body of work that exploit neural-like oscillators in visual grouping [216, 106, 107, 108, 109, 96, 97, 63], Wang and his colleagues have performed pioneering and very innovative work using oscillators for image segmentation [186, 204, 205, 116, 26] and have extended the scheme to auditory segregation [12, 202, 203]. They constructed oscillator networks with local excitatory lateral connections and a global inhibitory connection.

This Chapter proposes a simple biologically plausible visual grouping implementation with networks of neural oscillators, based on diffusive connections and concurrent synchronization [166]. The key idea is to embed the desired grouping properties in the choice of *diffusive couplings* between oscillators, so that the oscillators synchronize if their underlying

visual stimulative atoms belong to the same visual group and desynchronize otherwise. The same algorithm is applied to point clustering, contour integration and image segmentation.

The Chapter is organized as follows. Section 7.2 introduces a basic model of neural oscillators with diffusive coupling connections, and proposes a general visual grouping algorithm. Sections 7.3, 7.4 and 7.5 describe in detail the neural oscillator solutions for point clustering, contour integration and image segmentation and show a number of examples. The results are compared with normalized cuts, a popular computer vision method [179, 43]. As detailed in Sec 7.2.5, the method differs from the work of Wang et al [186, 204, 205, 116, 26] in several fundamental aspects, including the synchronization/desynchronization mechanism and the coupling structure.

7.2 Model and Algorithm

The model is a network of neural oscillators coupled with diffusive connections. Each oscillator is associated to an atomic element in the stimuli, for example a point, an orientation or a pixel. Without coupling, the oscillators are desynchronized and oscillate in random phases. Under diffusive coupling with the coupling strength appropriately tuned, they may converge to multiple groups of synchronized elements. The synchronization of oscillators within each group indicates the perceptual grouping of the underlying stimulative atoms, while the desynchronization between groups suggests group segregation.

7.2.1 Neural Oscillators

A neural oscillator is an elementary unit associated to an atomic element in the stimuli, and it models the perception of the stimulative atom. We use a modified form of FitzHugh-

Nagumo neural oscillators [66, 161], similar to [116, 26],

$$\dot{v}_i = 3v_i - v_i^3 - v_i^7 + 2 - w_i + I_i \tag{7.1}$$

$$\dot{w}_i = c[\alpha(1 + \tanh(\rho v_i)) - w_i] \tag{7.2}$$

where v_i is the membrane potential of the oscillator, w_i is an internal state variable representing gate voltage, I_i represents the external current input and α, ρ and c are strictly positive constants. This elementary unit oscillates if I_i exceeds a certain threshold and α, ρ and c are in an appropriate range (within this range, the grouping results have low sensitivities to the values of α, ρ and c — in all experiments we use $\alpha = 12$, $c = 0.04$, $\rho = 4$). Figure 7.2(a). plots the oscillation trace of membrane potential v_i. Other spiking oscillator models can be used similarly.

7.2.2 Diffusive Connections

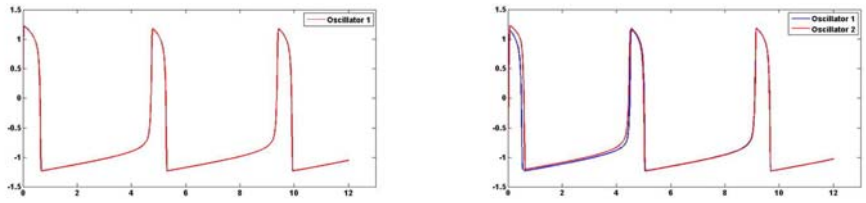
Oscillators are coupled to form a network which aggregates the perception of individual atoms in the visual stimulus. The oscillators are coupled through diffusive connections.

Let us denote by $\mathbf{x_i} = [v_i, w_i]^T$ the state vectors of the oscillators introduced in Section 7.2.1, each with dynamics $\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i, t)$. A neural oscillator network is composed of N oscillators, connected with diffusive coupling [207]

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i, t) + \sum_{i \neq j} k_{ij}(\mathbf{x}_j - \mathbf{x}_i), \quad i = 1, \ldots, N \tag{7.3}$$

where k_{ij} is the coupling strength.

Oscillators i and j are said to be *synchronized* if $\mathbf{x}_i$ remains equal to $\mathbf{x}_j$. Once the elements are synchronized, the coupling terms in (7.3) disappear, so that each individual element exhibits its natural and uncoupled behavior, as illustrated in Figure 7.2. A larger value of k_{ij} tends to reduce the state difference $\mathbf{x}_i - \mathbf{x}_j$ and thus to reinforce the synchronization between oscillators i and j (see the Appendix A for more details).



a **b**

Figure 7.2: **a.** the oscillation trace of a single oscillator. **b.** synchronization of two oscillators coupled through diffusive connections. The two oscillators start to be fully synchronized at about $t = 5$.

The key to using diffusively-coupled neural oscillators for visual grouping is to tune the couplings so that the oscillators synchronize if their underlying atoms belong to the same visual group, and desynchronize otherwise. According to Gestalt psychology [210, 92, 140], visual stimulative atoms having similarity (e.g. gray-level, color, orientation) or proximity tend to be grouped perceptually. This suggests making strong coupling between neural oscillators whose underlying stimuli are similar. Such coupling is implemented by a Gaussian tuning

$$k_{ij} = e^{\frac{-|s_i - s_j|^2}{\beta^2}}. \tag{7.4}$$

where s_i and s_j are stimuli of the two oscillators, for example position for point clustering, orientation for contour integration and gray-level for image segmentation; and β is a tuning parameter: the coupling strength falls off as a Gaussian function of the distance between the stimuli. Psychophysical evidence of Gaussian tuning in vision has been observed [167]. In computer vision, Gaussian tuning has been applied in various applications such as image denoising [14], segmentation [179, 165], recognition [177], mean shift clustering [71, 29, 41] and contour integration [216].

7.2.3 Concurrent Synchronization and Stability

In perception, fully synchronized elements in each group are bound, while different groups are segregated [201]. Concurrent synchronization analysis provides a mathematical tool to study stability and convergence properties of the corresponding neural oscillator networks.

In an ensemble of dynamical elements, *concurrent synchronization* is defined as a regime where the whole system is divided into multiple groups of fully synchronized elements, but elements from different groups are not necessarily synchronized [166]. Networks of oscillators coupled by diffusive connections defined in Sections 7.2.2 are specific cases of this general framework [166, 207].

A subset of the global state space is called *invariant* if trajectories that start in that subset remain in that subset. In our synchronization context, the invariant subsets of interest are linear subspaces, corresponding to some components of the overall state being equal ($\mathbf{x}_i = \mathbf{x}_j$) in (7.3). Concurrent synchronization analysis quantifies stability and convergence to invariant linear subspaces. Furthermore, a property of concurrent synchronization analysis, which turns out to be particularly convenient in the context of grouping, is that the actual invariant subset itself needs not be known *a priori* to guarantee stable convergence to it.

Concurrent synchronization may first be studied in an idealized setting, e.g., with exactly equal inputs to groups of oscillators, and noise-free conditions. This allows one to compute minimum coupling gains to guarantee global exponential convergence to the invariant synchronization subspace. *Robustness* of concurrent synchronization, a consequence of its exponential converge properties, allows the qualitative behavior of the nominal model to be preserved even in non-ideal conditions. In particular, it can be shown and quantified that for high convergence rates, actual trajectories differ little from trajectories based on

an idealized model [207].

A more specific discussion of stability and convergence is given in the Appendix A. The reader is referred to [166] for more details on the analysis tools.

7.2.4 Visual Grouping Algorithm

A basic and general visual grouping algorithm is obtained by constructing a neural oscillator network according to the following steps.

1. Construct a neural oscillator network. Each oscillator (7.1, 7.2) is associated to one atom in the stimuli. Oscillators are connected with diffusive connections (7.3) using the Gaussian-tuned gains (7.4). The coupling tuning for point clustering, contour integration and image segmentation will be specified in the following Sections.

2. Simulate the so-constructed network. The oscillators converge to concurrently synchronized groups.

3. Identify the synchronized oscillators and equivalently the visual groups. A group of synchronized oscillators indicates that the underlying visual stimulative atoms are perceptually grouped. Desynchronization between groups suggests that the underlying stimulative atoms in the two groups are segregated.

The differential equations (7.1) and (7.2) are solved using a Runge-Kutta method (with the Matlab ODE solver). Once the system converges to the invariant synchronization subspace (typically within 2 or 3 cycles), synchronization can be identified. As shown in Figures 7.3-b and 7.4-b, traces of synchronized oscillators coincide in time, while those of desynchronized groups are separated [201]. For point clustering and contour integration, the identification of synchronization in the oscillation traces is implemented by thresholding the correlation among the traces as proposed in [216]. For image segmentation, a k-means

algorithm (with random initialization and correlation distance measure) is applied on the oscillation traces to identify the synchronization. A complete description of the image segmentation algorithm is given in Figure 7.9, and the algorithm applies to point clustering and contour integration as well.

7.2.5 Relation to Previous Work

Among the relatively small body of previous work based on neural oscillators mentioned in Section 7.1, the pioneering work of Wang et al. [186, 204, 205, 26] on image segmentation called LEGION (locally excitatory globally inhibitory oscillator networks) is most related to the present Chapter. Yen and Finkel [216] have studied similar ideas for contour integration. While its starting point is the same, namely achieving grouping through oscillator synchronization and tight coupling between oscillators associated to similar stimuli, the method presented in this Chapter differs fundamentally from LEGION in the following aspects:

- **Synchronization mechanism.** A global inhibitor that allows active oscillators to inhibit others is a key element in LEGION in order to achieve synchronization within one group and distinction between different groups: At one time only one group of oscillators is active and the others are inhibited. The present work relies on *concurrent synchronization* [166] and does not contain any inhibitor: All oscillators are active simultaneously; multiple groups of oscillators, synchronized within each group and desynchronized between groups, coexist.

- **Coupling mechanism.** The oscillators coupling in LEGION is obtained through *stimulation*: Local excitation coupling is implemented through positive stimulation which tends to activate the oscillators in the neighborhood; global inhibition is obtained by negative stimulation which tends to deactivate other oscillators. The cou-

pling in the present work is based on *diffusive* connections (7.3): The coupling term is a state difference which tends to zero as synchronization occurs [207].

- **Robustness.** The original LEGION was shown to be sensitive to noise [186, 204]. A concept of lateral potential for each oscillator was introduced in a later version of LEGION [205, 26] to improve its robustness and reduce over-segmentation in real image applications. The proposed method based on non-local diffusive connections has inherent robustness to noise.

In turn, these fundamental differences make the proposed neural oscillator framework fairly simple and general. It provides solutions not only for image segmentation, but also for point clustering and contour integration, as detailed in the following Sections.

7.3 Point Clustering

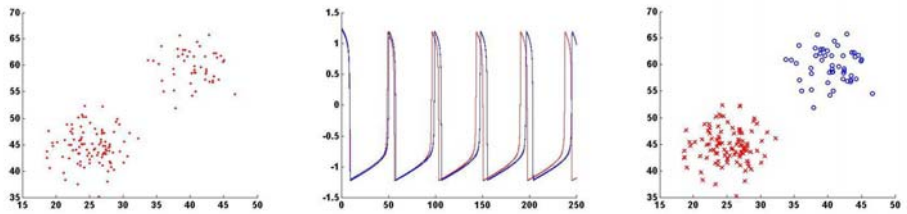
Point clustering with neural oscillators is based on diffusive connections (7.3) and follows directly the general algorithm in Section 7.2.4. Let us denote $\mathbf{c}_i = (i_x, i_y)$ the coordinates of a point $\mathbf{p}_i$. Each point $\mathbf{p}_i$ is associated to an oscillator $\mathbf{x}_i$. The proximity Gestalt principle [210, 92, 140] suggests strong coupling between oscillators corresponding to proximate points. More precisely, the coupling strength between $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$k_{ij} = \begin{cases} e^{\frac{-|\mathbf{c}_i - \mathbf{c}_j|^2}{\beta^2}} & \text{if } \; j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}, \tag{7.5}$$

where $\mathcal{N}_i$ is the set of M points closest to $\mathbf{p}_i$: an oscillator $\mathbf{x}_i$ is coupled with its M nearest neighbors. Oscillators may be indirectly coupled over larger distances through synchronization propagation (e.g., if $\mathbf{x}_1$ is coupled with $\mathbf{x}_2$ and $\mathbf{x}_2$ is coupled with $\mathbf{x}_3$, then $\mathbf{x}_1$ is indirectly coupled with $\mathbf{x}_3$.) M and β tune the coupling and thus adjust the size of the clusters one expects to detect. In the experiments, $M = 10$ and $\beta = 3$. The external

inputs I_i of the oscillators in (7.1) are chosen as uniformly distributed random variables in the appropriate range.

Figure 7.3 illustrates an example in which the points make clearly two clusters (Gaussian distributed, centered at (25,40) and (40,60) with standard deviation equal to 3×3). As shown in Figure 7.3(b), the oscillator system converges to two concurrently synchronized groups separated in the time dimension, each corresponding to one cluster. The identification of the two groups induces the clustering of the underlying points, as shown in Figure 7.3(c).



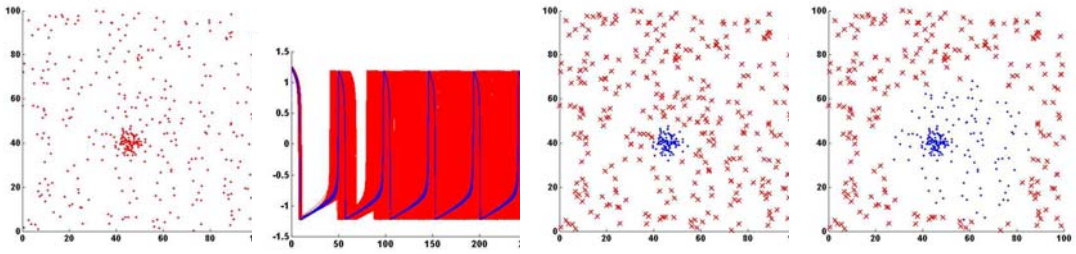
a **b** **c**

Figure 7.3: **a.** Points to cluster. **b.** The oscillators converge to two concurrently synchronized groups. **c.** Clustering results. The blue circles and the red crosses represent the two clusters.

Figure 7.4 presents a more challenging setting where one seeks to identify a dense cluster in a cloud of points. The cloud is made of 300 points uniformly randomly distributed in a space of size 100×100, in addition to a dense cluster of 100 Gaussian distributed points with standard deviation equal to 3×3. The neural oscillator system converges to one synchronized group that corresponds to the cluster with all the "outliers" totally desynchronized in the background, as shown in Figure 7.4(b). As in [216], the synchronized traces are segregated from the background by thresholding the correlation among all the traces (threshold = 0.99), which results in the identification of the underlying cluster as shown in Figure 7.4(c). The result of normalized cuts [179] is shown in Figure 7.4(d): a large number of outliers

around the cluster of interest are confused with the cluster.

In this experiment, the neural oscillator solution implemented with Matlab on a Pentium 4 PC takes about 10 seconds.



a **b** **c** **d**

Figure 7.4: **a.** A cloud of points made of 300 points uniformly randomly distributed in a space of size 100×100, in addition to a cluster of 100 Gaussian distributed points with standard deviation equal to 3×3. **b.** The neural oscillator system converges to one synchronized group that corresponds to the cluster with all the "outliers" totally desynchronized in the background. **c.** and **d.** Clustering results by respectively neural oscillators and normalized cuts: blue dots represent the cluster detected by the algorithm and red crosses are the "outliers". In the latter many outliers are confused with the cluster of interest.

7.4 Contour Integration

Field et al. [65] have performed interesting experiments to test human capacity of contour integration, i.e., of identifying a path within a field of randomly-oriented Gabor elements. They made some quantitative observations in accordance with the Gestalt "good continuation" law [210, 216]:

- Contour integration can be made when the successive elements in the path, i.e., the element-to-element angle β (see Figure 7.5), differ by 60° or less.
- There is a constraint between the element-to-element angle β and the element-to-path angle α (see Figure 7.5). The visual system can integrate large differences in element-to-element orientation only when those differences lie along a smooth path, i.e., only

when the element-to-path angle α is small enough. For example, with $\alpha = 15°$ and $\beta = 0°$ the contour integration is difficult, even though the observers can easily track a $15°$ orientation difference when there is no variation ($\alpha = 0°$ and $\beta = 15°$).

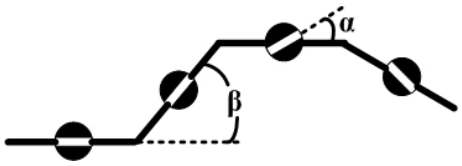


Figure 7.5: The element-to-element angle β is the difference in angle of orientation of each successive path segment. The element-to-path angle α is the angle of orientation of the element with respect to the path. This figure is adapted from [65].

Figures 7.6-7.8 show the setting of our contour integration experiments, similar to that in [65]. An orientation value $\mathbf{o}_i \in [0, 2\pi)$ is defined for each point $i = (i_x, i_y)$ in a grid, as illustrated by the arrows. Smooth contours potentially imbedded in the grid are to be detected.

Following the general visual grouping algorithm described in Section 7.2.4, neural oscillators with diffusive connections (7.3) are used to perform contour integration. Each orientation in the grid is associated to one oscillator. The coupling of the oscillators i and j follows the Gestalt law of "good continuation" and, in particular, the results of the psychovisual experiments of [65]:

$$k_{ij} = \begin{cases} \exp\left(-\frac{|\mathbf{o}_i - \mathbf{o}_j|^2}{\delta^2} - \frac{|\frac{\mathbf{o}_i + \mathbf{o}_j}{2} - \mathbf{o}_{ij}|^2}{\gamma^2}\right) & \text{if } |i-j| \leq w \\ 0 & \text{otherwise} \end{cases} . \tag{7.6}$$

where

$$\mathbf{o}_{ij} = \begin{cases} \theta_{ij} & \text{if } |\theta_{ij} - \frac{|\mathbf{o}_i + \mathbf{o}_j|}{2}| < |\theta_{ij} + \pi - \frac{|\mathbf{o}_i + \mathbf{o}_j|}{2}| \\ \theta_{ij} + \pi & \text{otherwise} \end{cases}$$

is the undirectional orientation of the path ij (the closest to the average element-to-element orientation $\frac{|\mathbf{o_i} + \mathbf{o_j}|}{2}$ modulo π), with $\theta_{ij} = \arctan\left(\frac{i_y - j_y}{i_x - j_x}\right)$. Oscillators with a small element-

to-element angle β (the first term in (7.6)) and a small element-to-path angle α (the second term in (7.6)) are coupled more tightly. The neural oscillator system makes therefore smooth contour integration. δ and γ tune the smoothness of the detected contour. As contour integration is known to be rather local [65], the coupling (7.6) is effective within a neighborhood of size $(2w+1) \times (2w+1)$. In the experiments the parameters are configured as $\delta = 20°$, $\gamma = 10°$ and $w = 1$, in line with the results of the psychovisual experiments of Field et al [65]. The external inputs I_i of the oscillators in (7.1) are set proportional to the underlying orientations $\mathbf{o}_i$.

Figure 7.6(a) presents a grid in which orientations are uniformly distributed in space, except for one vertical contour. The orientation of the elements on the vertical contour undertakes furthermore a Gaussian perturbation of standard deviation $\sigma = 10°$. The neural oscillator system converges to one synchronized group that corresponds to the contour with all the other oscillators desynchronized (the traces are similar to Figure 7.4(b)). As in [216], the synchronized group is segregated from the background by thresholding the correlation among the traces (threshold = 0.99). This results in the "pop-out" of the contour shown in Figure 7.6(b). As shown in Figure 7.6(c), the multiscale normalized cut [43] does not succeed to segregate the contour from the background. (Normalized cuts fail in the following contour integration experiments as well and are not shown.) Figure 7.7 illustrates a similar example with two intersecting straight contours.

Figure 7.8(a) illustrates a smooth curve embedded in the uniformly randomly distributed orientation background. With some minor effort, subjects are able to identify the curve due to its "good continuation". Similarly the neural system segregates the curve from the background with the oscillators lying on the curve fully synchronized, as illustrated in Figure 7.8(b).

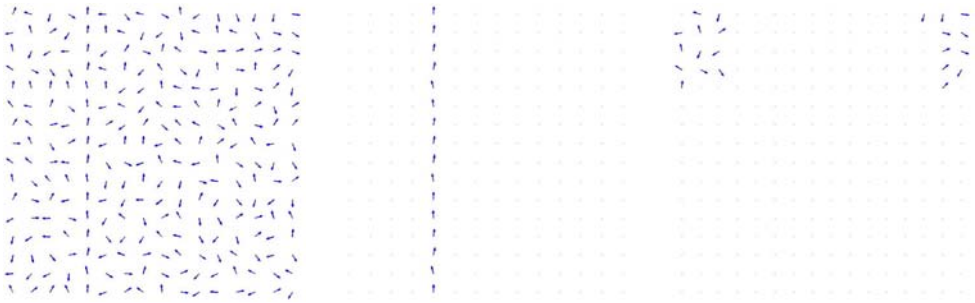


Figure 7.6: **a.** A vertical contour is embedded in a uniformly distributed orientation grid. **b.** and **c.** Contour integration by respectively neural oscillators and normalized cuts.

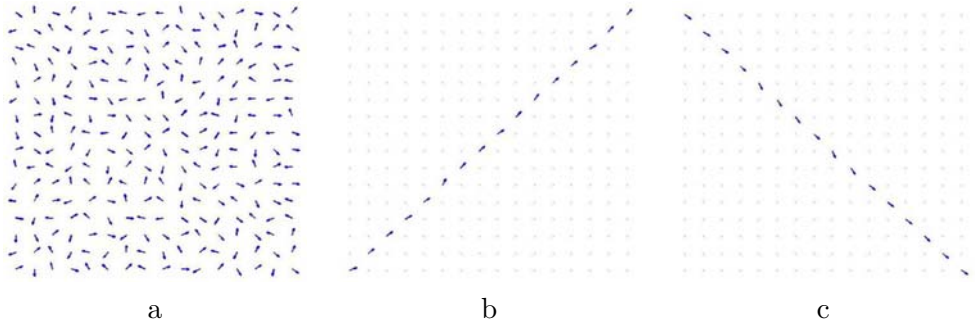
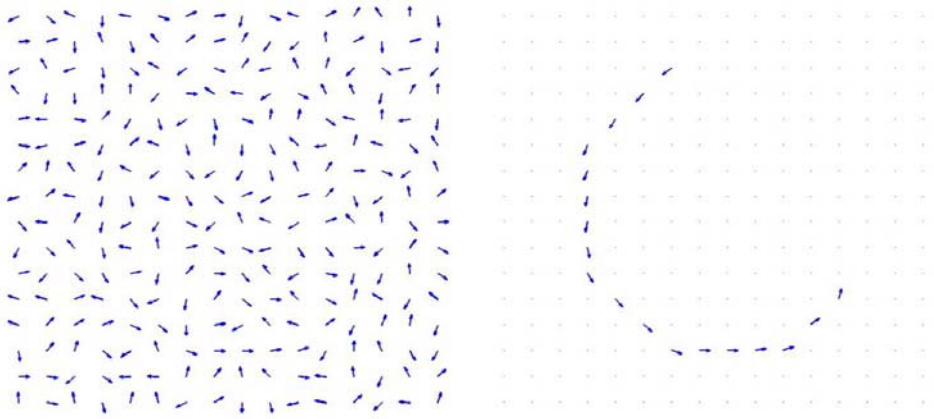


Figure 7.7: **a.** Two contours are embedded in a uniformly distributed orientation grid. **b.** and **c.** the two identified contours identified by neural oscillators.

In these experiment, the neural oscillator solution implemented with Matlab on a Pentium 4 PC takes about 10 seconds.



a b

Figure 7.8: **a.** A smooth curve is embedded in a uniformly distributed orientation grid. **b.** The curve identified by neural oscillators.

7.5 Image Segmentation

The proposed image segmentation scheme follows the general visual grouping algorithm described in Section 7.2.4. One oscillator is associated to each pixel in the image. Within a neighborhood the oscillators are non-locally coupled with a coupling strength

$$k_{ij} = \begin{cases} e^{\frac{-|u_i - u_j|^2}{\beta^2}} & \text{if } |i-j| < w \\ 0 & \text{otherwise} \end{cases} . \tag{7.7}$$

where $u_{\mathbf{i}}$ is the pixel gray-level at coordinates $i = (i_x, i_y)$ and w adjusts the size of the neighborhood. Pixels with similar gray-levels are coupled more tightly, as suggested by the color constancy Gestalt law [210, 92, 140]. As in [216, 14], a Gaussian function of gray-level difference is used to tuned to the coupling strength. Non-local coupling plays an important role in regularizing the image segmentation, with a larger w resulting in more regularized

segmentation and higher robustness to noise. A complete image segmentation algorithm description is given in Figure 7.9.

Figure 7.10(a) illustrates a synthetic image (the gray-levels of the black, gray and white parts are 0, 128, and 255) contaminated by white Gaussian noise of moderate standard deviation $\sigma = 10$. The segmentation algorithm was configured with $\beta = \sigma$ and $w = 3$. The external inputs I_i of the oscillators in (7.1) are set proportional to the underlying orientation u_i. The oscillators converge into three concurrently synchronized groups as plotted in Figure 7.10(b), which results in a perfect segmentation as shown in Figure 7.10(c). K-means with a random initialization and a correlation distance measure is applied to detect oscillator synchronization in all the image segmentation experiments.

Figure 7.11 shows some real image segmentation examples in comparison with the multiscale normalized cuts [43]. Both methods obtain rather regular segmentation with hardly any "salt and pepper" holes. Using the same number of classes, the segmentation obtained by neural oscillators seems more subtle and closer to human perception: in the sagittal MRI (Magnetic Resonance Imaging), salient regions such as cortex, cerebellum and lateral ventricle are segregated with good accuracy; in the radar image, the cloud boundaries and eye of the hurricane are more precisely segregated.

Image segmentation.

1. Construct the neural oscillator network.

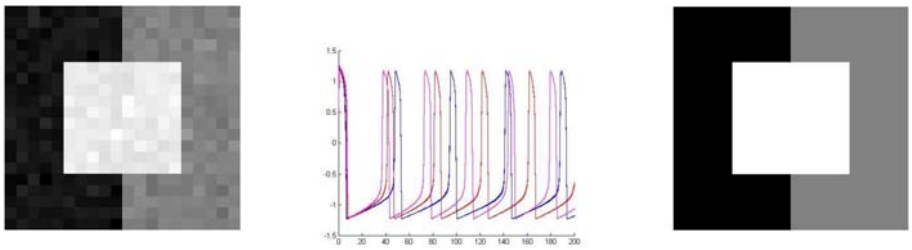
   (a) Make N neural oscillators $\mathbf{x}_i = [v_i, w_i]^T$ following (7.1,7.2): $\alpha = 12$, $c = 0.04$, $\rho = 4$, $I_i = a(u_i)$, where u_i is the gray level (0-255) of pixel i, $i = 1, \ldots, N$, with N the number of pixels in the image, and $a(u_i) = (u_i - u_{\min})\frac{I_{\max} - I_{\min}}{u_{\max} - u_{\min}} + I_{\min}$ is an affine mapping that maps the gray level the appropriate external input range $[I_{\min}, I_{\max}]$, with $u_{\min} = \min_i u_i$ and $u_{\min} = \max_i u_i$, $I_{\min} = 0.8$ and $I_{\max} = 2$.

   (b) Couple the N oscillators with diffusive connections (7.3), where the coupling strength k_{ij} follows the non-local Gaussian tuning (7.7). The coupling is stronger with bigger β and w. Typical parameter configuration is: $\beta = 20$ for real images, $\beta = \sigma$ for synthetic images contaminated by white Gaussian noise of standard deviation σ; $w = 3$ for images with little or moderate noise, $w = 5$ for very noisy images.

2. Simulate the neural network by solving the differential equation system constructed in Step 1. In the experiments, the Matlab ODE solver is used to solve the differential equations. Typically the simulation is performed for 6 oscillation cycles and the oscillators converge within 2 or 3 cycles.

3. Apply a k-means algorithm (with random initialization and correlation distance measure) on the oscillation traces starting from the 3rd cycle after which the oscillators have typically converged. The oscillation traces are clustered in M classes, the corresponding oscillators and underlying pixels follow the same classification.

Figure 7.9: Complete algorithm description: image segmentation.



a **b** **c**

Figure 7.10: **a.** A synthetic image (the gray-levels of the black, gray and white parts are respectively 0, 128, 255) contaminated by white Gaussian noise of standard deviation $\sigma = 10$. **b.** The traces of the neural oscillation ($\beta = \sigma$, $w = 3$). The oscillators converge into three concurrently synchronized groups. **c.** Segmentation result.

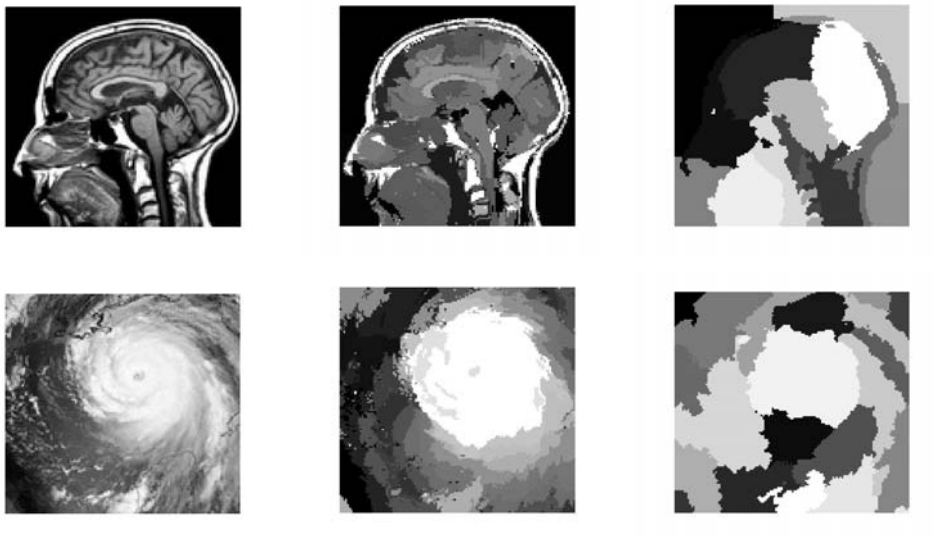


Figure 7.11: Real image segmentation. From top to bottom, left to right: a sagittal MRI image (128×128); segmentation in 15 classes by neural oscillators and multiscale normalized cuts; a radar image (128×128); segmentation in 20 classes by neural oscillators and multiscale normalized cuts. Neural oscillator network is configured with $\beta = 20$ and $w = 3$.

Chapter 8

Conclusion

This thesis contributes to three aspects in image, signal processing and computer vision. The first part is devoted to sparse geometric image and signal processing with blocks. Sparse representation coefficients are grouped in blocks that adapt to signal geometrical regularity and coefficient processing is conducted by block. We introduce a time-frequency block thresholding audio denoising algorithm that thresholds the time-frequency coefficients by block, and adapts the block size to the time-frequency regularity of the audio signal. The adaptation is performed by minimizing a Stein unbiased risk estimator calculated from the data. The proposed algorithm removes efficiently the musical noise artifact and improves the SNR and the perceived quality with respect to state-of-the-art audio denoising algorithms. Furthermore, we generalize block thresholding with oriented blocks that adapt to image geometrical regularity. Oriented blocks are calculated with a block pursuit procedure that decomposes sparse representation coefficients to blocks selected from a dictionary of blocks. The resulting block pursuit thresholding improves PSNR with respect to block thresholding. The block pursuit procedure identifies geometrical image model and calculates structured sparse representations. A super-resolution zooming algorithm is derived by directional

interpolation along the block directions in which image is directionally regular. Numerical experiments illustrate the efficiency of this super-resolution procedure compared to cubic spline interpolations.

In the second part of the thesis, we propose a new affine invariant image comparison algorithm ASIFT. While the state-of-the-art image comparison method SIFT is fully similarity invariant by simulating the zoom in the scale space and normalizing the translation and the rotation, the new method simulates in addition the two camera axis orientation angles. More specifically, ASIFT simulates the camera axis latitude and longitude angles, and then applies SIFT which simulates the scale and normalizes the rotation and the translation. Mathematically, ASIFT is proved fully affine invariant, up to sampling errors. A sparse sampling of the simulated parameters is proposed. A coarse-to-fine implementation of ASIFT is described, that has about twice the complexity of a single SIFT routine. Many numerical experiments show that ASIFT outperforms significantly the state-of-the-art, including SIFT, MSER, Harris-Affine and Harris-Affine.

In the third part we introduce a biologically inspired visual grouping implementation based on dynamical systems. Simple networks of neural oscillators coupled with diffusive connections are proposed to solve visual grouping problems. The key idea is to embed the desired grouping properties in the choice of the diffusive couplings, so that synchronization of oscillators within each group indicates perceptual grouping of the underlying stimulative atoms, while desynchronization between groups corresponds to group segregation. Compared with state-of-the-art approaches, the same algorithm is shown to achieve promising results on several classical visual grouping problems, including point clustering, contour integration, and image segmentation.

Publication List

Journal Papers

1. J.M. Morel and G.Yu, ASIFT: A New Framework for Fully Affine Invariant Image Comparison, *SIAM Journal on Imaging Sciences*, vol 2, no 2, pp. 438-469, 2009.
2. G. Yu, S. Mallat, E. Bacry, Audio Denoising by Time-Frequency Block Thresholding, *IEEE Trans. on Signal Processing*, vol 56, no. 5, pp. 1830-1839, May 2008.
3. G. Yu and J.J. Slotine, Visual Grouping by Neural Oscillators, accepted to *IEEE Trans. Neural Networks*, 2009.
4. J.M. Morel and G. Yu, On the consistency of the SIFT Method, accepted to *Inverse Problems and Imaging*, 2009.
5. S. Mallat and G. Yu, Super-Resolution with Piecewise Linear Sparse Representations, to be sumitted, 2009.

Conference Papers

1. S. Mallat and G. Yu, Structured Space Pursuits for Geometric Super-Resolution, invited paper, *Proc. IEEE International Conference on Image Processing (ICIP)*, 2009.
2. G. Yu and S. Mallat, Sparse Super-Resolution with Space Matching Pursuits, *Proc. Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)*, Saint-Malo, 2009.
3. G. Yu and J.M. Morel, A Fully Affine Invariant Image Comparison Method, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, 2009.
4. G. Yu and J.J. Slotine, Audio Classification from Time-Frequency Texture, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, 2009.

5. G. Yu and J.J.Slotine, Visual Grouping by Neural Oscillators, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, 2009.

6. G. Yu and J.J. Slotine, Fast Wavelet-based Visual Classification, *Proc. IEEE International Conference on Pattern Recognition (ICPR)*, Tampa, 2008.

7. G. Yu, E. Bacry and S. Mallat, Audio Signal Denoising with Complex Wavelets and Adaptive Block Attenuation, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hawaii, pp. 869-872, 2007.

Patents

1. S. Mallat and G. Yu, Video enhancement using recursive bandlets, PCT FR2008/050179, PCT IB2008/051770, filed Feb. 6, 2008.

2. G. Yu and J.M. Morel, Proédé et dispositif de reconnaissance invariante-affine de formes (Process and system of affine-invariant shape recognition), FR 08/53244, filed May 15, 2008.

3. Fast pattern classification based on a sparse transform, US/PCT 13191, filed June 2, 2008.

Appendix A

Appendix

Proof of Theorem 2

Proof. Consider the real symmetric positive semi-definite matrix A^tA, where A^t denotes the transposed matrix of A. By classic spectral theory there is an orthogonal transform O such that $A^tA = ODO^t$ where D a diagonal matrix with ordered eigenvalues $\lambda_1 \geq \lambda_2$. Set $O_1 = AOD^{-\frac{1}{2}}$. Then $O_1O_1^t = AOD^{-\frac{1}{2}}D^{-\frac{1}{2}}O^tA^t = AOD^{-1}O^tA^t = A(A^tA)^{-1}A^t = I$. Thus, there are orthogonal matrices O_1 and O such that

$$A = O_1D^{\frac{1}{2}}O^t. \tag{A.1}$$

Since the determinant of A is positive, the product of the determinants of O and O_1 is positive. If both determinants are positive, then O and O_1 are rotations and we can write $A = R(\psi)DR(\phi)$. If ϕ is not in $[0, \pi)$, changing ϕ into $\phi - \pi$ and ψ into $\psi + \pi$ ensures that $\phi \in [0, \pi)$. If the determinants of O and O_1 are both negative, replacing O and O_1 respectively by $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} O$ and $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} O_1$ makes them into rotations without altering (A.1), and we can as above ensure $\phi \in [0, \pi)$ by adapting ϕ and ψ. The final decomposition is obtained by taking for λ the smaller eigenvalue of $D^{\frac{1}{2}}$. □

Convergence and Stability

One can verify ([166], to which the reader is referred for more details on the analysis tools) that a sufficient condition for global exponential concurrent synchronization of an oscillator network is

$$\lambda_{\min}(\mathbf{VLV}^T) > \sup_{\mathbf{a},t} \lambda_{\max}\left(\frac{\partial}{\partial \mathbf{x}}(\mathbf{a}, t)\right), \tag{A.2}$$

where $\lambda_{\min}(\mathbf{A})$ and $\lambda_{\max}(\mathbf{A})$ are respectively the smallest and largest eigenvalues of the symmetric matrix $\mathbf{A}_s = (\mathbf{A} + \mathbf{A}^T)/2$, $\mathbf{L}$ is the Laplacian matrix of the network ($\mathbf{L}_{ii} = \sum_{j \neq i} k_{ij}$, $\mathbf{L}_{ij} = -k_{ij}$ for $j \neq i$) and $\mathbf{V}$ is a projection matrix on $\mathcal{M}^\perp$. Here $\mathcal{M}^\perp$ is the subspace orthogonal to the subspace $\mathcal{M}$ in which *all* the oscillators are in synchrony − or, more generally in the case of a hierarchy, where all oscillators at each level of the hierarchy are in

synchrony (i.e., $\mathbf{x}_1 = \cdots = \mathbf{x}_N$ at each level). Note that $\mathcal{M}$ itself need not be invariant (i.e. all oscillators synchronized at each level of the hierarchy need not be a particular solution of the system), but only needs to be a *subspace* of the actual invariant synchronization subspace ([75], [166] section 3.3.i.), which may consist of synchronized subgroups according to the input image. Indeed, the space where all $\mathbf{x}_i$ are equal (or, in the case of a hierarchy, where at each level all $\mathbf{x}_i$ are equal), while in general not invariant, is always a subspace of the actual invariant subspace corresponding to synchronized subgroups.

These results can be applied e.g. to individual oscillator dynamics of the type (7.1). Let $\mathbf{J}_i$ denote the Jacobian matrix of the individual oscillator dynamics (7.1), $\mathbf{J}_i = \partial[\dot{v}_i\ \dot{w}_i]^T/\partial[v_i, w_i]^T$. Using a diagonal metric transformation $\mathbf{\Theta} = \mathrm{diag}(\sqrt{c\alpha\beta}, 1)$, one easily shows, similarly to [207, 208], that the transformed Jacobian matrix $\mathbf{\Theta}\mathbf{J}_i\mathbf{\Theta}^{-1} - \mathrm{diag}(k, 0)$ is negative definite for $k > 3 + \frac{\alpha\beta}{4}$. More general forms of oscillators can also be used. For instance, other second-order models can be created based on a smooth function $f(.)$ and an arbitrary sigmoid-like function $\sigma(.) \geq 0$ such that $0 \leq \sigma'(.) \leq 1$, in the form

$$\dot{v}_i = f(v_i) - w_i + I_i \tag{A.3}$$

$$\dot{w}_i = c[\alpha\sigma(\beta v_i) - w_i] \tag{A.4}$$

with the transformed Jacobian matrix negative definite for $k > f'(v_i) + \frac{\alpha\beta}{4}$.

From a stability analysis point of view, the coupling matrix $\mathbf{K}$ of the Gaussian-tuned coupling, composed of coupling coefficients k_{ij} in (7.4), presents desirable properties. It is symmetric, and the classical theorem of Schoenberg (see [142]) shows that it is positive definite. Also, although it may actually be state-dependent, the coupling matrix $\mathbf{K}$ can always be treated simply as a time-varying external variable for stability analysis and Jacobian computation purposes, as detailed in ([207], section 4.4, [166], section 2.3).

Bibliography

[1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *International Conference on Computer Graphics and Interactive Techniques*, pages 853–861, 2006.

[2] AP Ashbrook, NA Thacker, PI Rockett, and CI Brown. Robust recognition of scaled shapes using pairwise geometric histograms. *Proc. BMVC*, pages 503–512, 1995.

[3] F. Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183–193, 1954.

[4] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer; 2nd ed. edition, 2006.

[5] M. Bahoura and J. Rouat. Wavelet speech enhancement based on time–scale adaptation. *Speech Communication*, 48(12):1620–1637, 2006.

[6] A. Baumberg. Reliable feature matching across widely separated views. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 1:774–781, 2000.

[7] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *European Conference on Computer Vision*, 1:404–417, 2006.

[8] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002.

[9] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke. Metric Localization with Scale-Invariant Visual Features Using a Single Perspective Camera. *European Robotics Symposium*, 2006.

[10] M. Berouti, R. Schwartz, J. Makhoul, B. Beranek, I. Newman, and MA Cambridge. Enhancement of speech corrupted by acoustic noise. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'79.*, volume 4, 1979.

[11] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113–120, 1979.

[12] G.J. Brown and D.L. Wang. Modelling the perceptual segregation of double vowels with a network of neural oscillators. *Neural Networks*, 10(9):1547–1558, 1997.

[13] M. Brown and D. Lowe. Recognising panorama. In *Proc. the 9th Int. Conf. Computer Vision, October*, pages 1218–1225, 2003.

[14] A. Buades, B. Coll, and J.M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530, 2006.

[15] G. Buzsaki. *Rhythms of the Brain*. Oxford University Press, USA, 2006.

[16] T. Cai and H. Zhou. A data-driven block thresholding approach to wavelet estimation. *Statistics Dept., Univ. of Pennsylvania, Tech. Rep*, 2005.

[17] T.T. Cai. Adaptive wavelet estimation: a block thresholding and oracle inequality approach. *Annals of Statistics*, pages 898–924, 1999.

[18] T.T. Cai. Bemard W. Silverman Incorporating Information on Neighboring Coefficients into Wavelet Estimation. *Sankhya*, 63:127–148, 2001.

[19] E. Candes, L. Demanet, D. Donoho, and L. Ying. Fast Discrete Curvelet Transforms. *SIAM Journal of Multiscale Modeling and Simulation*, 5(3):861–899, 2006.

[20] E.J. Candes and D.L. Donoho. Curvelets: a surprisingly effective nonadaptive representation for objects with edges. Curves and Surfaces, 105–120. Edited by C. Rabut, A. Cohen, and LL Schumaker, 2000.

[21] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur. *A Theory of Shape Identification*. Springer Verlag, 2008.

[22] O. Cappe. Elimination of the musical noise phenomenon with the Ephraim andMalah noise suppressor. *IEEE Transactions on Speech and Audio Processing*, 2(2):345–349, 1994.

[23] E.Y. Chang. EXTENT: fusing context, content, and semantic ontology for photo annotation. *Proceedings of the 2nd international workshop on Computer vision meets databases*, pages 5–11, 2005.

[24] S. Chang, Y. Kwon, S. Yang, and I. Kim. Speech enhancement for non-stationary noise environment by adaptivewavelet packet. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002. Proceedings.(ICASSP'02)*, volume 1, 2002.

[25] C. Chaux, A. Benazza-Benyahia, and J.C. Pesquet. A block-thresholding method for multispectral image denoising. In *Wavelets XI. Edited by Papadakis, Manos; Laine, Andrew F.; Unser, Michael A. Proceedings of the SPIE*, volume 5914, pages 495–507, 2005.

[26] K. Chen and D.L. Wang. A dynamically coupled neural oscillator network for image segmentation. *Neural Networks*, 15(3):423–439, 2002.

[27] S.H. Chen and J.F. Wang. Speech enhancement using perceptual wavelet packet decomposition and Teager energy operator. *The Journal of VLSI Signal Processing*, 36(2):125–139, 2004.

[28] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1999.

[29] Y. Cheng. Mean Shift, Mode Seeking, and Clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, page 790799, 1995.

[30] C. Chesneau, J. Fadili, and J.L. Starck. Stein block thresholding for image denoising. *Arxiv preprint arXiv:0809.3486*, 2008.

[31] M.A. Chughtai and N. Khattak. An Edge Preserving Locally Adaptive Anti-aliasing Zooming Algorithm with Diffused Interpolation. In *Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision*. IEEE Computer Society Washington, DC, USA, 2006.

[32] I. Cohen. Enhancement of speech using bark-scaled wavelet packet decomposition. In *Seventh European Conference on Speech Communication and Technology*. ISCA, 2001.

[33] I. Cohen. Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator. *IEEE Signal processing letters*, 9(4):113–116, 2002.

[34] I. Cohen. Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging. *IEEE Transactions on Speech and Audio Processing*, 11(5):466–475, 2003.

[35] I. Cohen. Speech enhancement using a noncausal a priori SNR estimator. *IEEE Signal Processing Letters*, 11(9):725–728, 2004.

[36] I. Cohen. Relaxed statistical model for speech enhancement and a priori SNR estimation. *IEEE Transactions on Speech and Audio Processing*, 13(5 Part 2):870–881, 2005.

[37] I. Cohen. Speech enhancement using super-Gaussian speech models and noncausal a priori SNR estimation. *Speech Communication*, 47(3):336–350, 2005.

[38] I. Cohen. Speech spectral modeling and enhancement based on autoregressive conditional heteroscedasticity models. *Signal processing*, 86(4):698–709, 2006.

[39] I. Cohen and B. Berdugo. Speech enhancement for non-stationary noise environments. *Signal Processing*, 81(11):2403–2418, 2001.

[40] R.R. Coifman and D.L. Donoho. Translation-invariant de-noising. *Wavelet and Statistics, Lecture Notes in Statistics, A. Antoniadis and G. Oppenheim, ed*, pages 125–150, 1995.

[41] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 603–619, 2002.

[42] T.H. Corman, C.E. Leiserson, R.L. Rivest, and C. Stein. Introduction to algorithms, 1990.

[43] T.B. Cour, F. Benezit, and J. Shi. Spectral Segmentation with Multiscale Graph Decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, 2005.

[44] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, et al. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080, 2007.

[45] G. Davis, Mallat S., and Avellaneda M. Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98, 1997.

[46] A. Desolneux, L. Moisan, and J.-M. Morel. Computational gestalts and perception thresholds. *Journal of Physiology - Paris*, 97(2-3):311–322, 2003.

[47] A. Desolneux, L. Moisan, and J.-M. Morel. A grouping principle and four applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):508–513, 2003.

[48] A. Desolneux, L. Moisan, and J.-M. Morel. *Gestalt Theory and Image Analysis, a Probabilistic Approach.* Interdisciplinary Applied Mathematics series, Springer Verlag, 2007. Preprint available at http://www.cmla.ens-cachan.fr/Utilisateurs/morel/lecturenote.pdf.

[49] D.L. Donoho and J.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[50] M. Elad and D. Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. *The Computer Journal*, 2007.

[51] M. Elad, J.L. Starck, P. Querre, and D.L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Appl. Comput. Harmon. Anal*, 19:340–358, 2005.

[52] Y. Ephraim and I. Cohen. Recent advancements in speech enhancement. *The Electrical Engineering Handbook*, pages 15–12.

[53] Y. Ephraim, H. Lev-Ari, and W.J.J. Roberts. A brief survey of speech enhancement. *The Electrical Engineering Handbook, CRC Press*, 2005.

[54] Y. Ephraim and D. Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(6):1109–1121, 1984.

[55] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean square error log-spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-33:443–445, 1985.

[56] Y. Ephraim and HL Van Trees. A signal subspace approach for speech enhancement. *IEEE Transactions on Speech and Audio Processing*, 3(4):251–266, 1995.

[57] A. Desolneux P. Mus F. Cao, J. Delon and F. Sur. A unified framework for detecting groups and application to shape recognition. *Technical Report 1746, IRISA*, September 2005.

[58] M.J. Fadili, J.L. Starck, and F. Murtagh. Inpainting and Zooming Using Sparse Representations. *The Computer Journal*, 2007.

[59] Q. Fan, K. Barnard, A. Amir, A. Efrat, and M. Lin. Matching slides to presentation videos using SIFT and scene background matching. *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 239–248, 2006.

[60] O. Farooq and S. Datta. Wavelet-based denoising for robust feature extraction for speech recognition. *Electronics Letters*, 39(1):163–165, 2003.

[61] S. Farsiu, D. Robinson, M. Elad, P. Milanfar, and CALIFORNIA UNIV SANTA CRUZ ELECTRICAL ENGINEERING DEPT. Advances and Challenges in Super-Resolution. *International Journal of Imaging Systems and Technology*, 14(2):47–57, 2004.

[62] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[63] O. Faugeras, F. Grimbert, and J.J. Slotine. Stability and synchronization in neural fields. *S.I.A.M. Journal on Applied Mathematics*, 68(8), 2008.

[64] L. Février. A wide-baseline matching library for Zeno. *Internship report, www.di.ens.fr/~fevrier/papers/2007-InternsipReportILM.pdf*, 2007.

[65] D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system: evidence for a local "association field". *Vision Res*, 33(2):173–193, January 1993.

[66] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1:445–466., 1961.

[67] J.J. Foo and R. Sinha. Pruning SIFT for scalable near-duplicate image matching. *Proceedings of the Eighteenth Conference on Australasian Database*, 63:63–71, 2007.

[68] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-Based Super-Resolution. *IEEE Computer Graphics and Applications*, pages 56–65, 2002.

[69] G. Fritz, C. Seifert, M. Kumar, and L. Paletta. Building detection from mobile imagery using informative SIFT descriptors. *Lecture Notes in Computer Science*, pages 629–638.

[70] J.J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, 2004.

[71] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.

[72] J.S. Garofolo et al. Getting started with the DARPA TIMIT CD-ROM: An acoustic phonetic continuous speech database, 1988.

[73] C. Gasquet and P. Witomski. *Fourier Analysis and Applications: Filtering, Numerical Computation, Wavelets*. Springer Verlag, 1999.

[74] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, pages 564–584, 1987.

[75] L. Gerard and J.J.E. Slotine. Neuronal networks and controlled symmetries. *arXiv:q-bio/0612049v4*.

[76] E.P. Ghael, A.M. Sayeed, and R.G. Baraniuk. Improved wavelet denoising via empirical Wiener filtering. In *Proceedings of SPIE*, 1997.

[77] Y. Ghanbari and M.R. Karami-Mollaei. A new approach for speech enhancement based on the adaptive thresholding of the wavelet packets. *Speech Communication*, 48(8):927–940, 2006.

[78] I. Gordon and D.G. Lowe. What and Where: 3D Object Recognition with Accurate Pose. *Lecture Notes in Computer Science*, 4170:67, 2006.

[79] L. Grady. Space-Variant Computer Vision: A Graph-Theoretic Approach. *PhD thesis, Boston University Graduate School of Arts and Sciences, 2004*.

[80] O. G. Guleryuz. Nonlinear Approximation Based Image Recovery Using Adaptive Sparse Reconstructions and Iterated DenoisingPart I: Theory. *Image Processing, IEEE Transactions on*, 15(3):539–554, 2006.

[81] P. Hall, G. Kerkyacharian, and D. Picard. Block threshold rules for curve estimation using kernel and wavelet methods. *Annals of Statistics*, pages 922–942, 1998.

[82] P. Hall, G. Kerkyacharian, and D. Picard. Note on the wavelet oracle. *Statist. Probab. Lett*, 43:415–420, 1999.

[83] P. Hall, G. Kerkyacharian, and D. Picard. On the minimax optimality of block thresholded wavelet estimators. *Statistica Sinica*, 9:33–50, 1999.

[84] J.S. Hare and P.H. Lewis. Salient regions for query by image content. *Image and Video Retrieval: Third International Conference, CIVR*, pages 317–325, 2004.

[85] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 15:50, 1988.

[86] R.F. Hess, A. Hayes, and DJ Field. Contour integration and cortical processing. *Journal of Physiology-Paris*, 97(2-3):105–119, 2003.

[87] Y. Hu and PC Loizou. Incorporating a psychoacoustical model in frequency domain speech enhancement. *IEEE Signal Processing Letters*, 11(2 Part 2):270–273, 2004.

[88] M.T. Johnson, X. Yuan, and Y. Ren. Speech signal enhancement through adaptive wavelet thresholding. *Speech Communication*, 49(2):123–133, 2007.

[89] Mikolajczyk K. and Schmid C. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[90] T. Kadir, A. Zisserman, and M. Brady. An Affine Invariant Salient Region Detector. In *European Conference on Computer Vision*, pages 228–241, 2004.

[91] G. Kanizsa. *Organization in Vision: Essays on Gestalt Perception*. Praeger, 1979.

[92] G. Kanizsa. Grammatica del Vedere, Il Mulino, Bologna, 1980. *Traduction française: La grammaire du voir, Diderot Editeur, Arts et Sciences*, 1996.

[93] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2:506–513, 2004.

[94] J. Kim, S.M. Seitz, and M. Agrawala. Video-based document tracking: unifying your physical and electronic desktops. *Proc. of the 17th Annual ACM Symposium on User interface Software and Technology*, 24(27):99–107, 2004.

[95] N.S. Kim and J.H. Chang. Spectral enhancement based on global soft decision. *IEEE Signal processing letters*, 7(5):108–110, 2000.

[96] M. Kuzmina, E. Manykin, and I. Surina. Tunable Oscillatory Network for Visual Image Segmentation. *Proc. of ICANN*, pages 1013–1019, 2001.

[97] M. Kuzmina, E. Manykin, and I. Surina. Oscillatory network with self-organized dynamical connections for synchronization-based image segmentation. *BioSystems*, 76(1-3):43–53, 2004.

[98] E. Le Pennec and S. Mallat. Bandelet Image Approximation and Compression. *SIAM Journal of Multiscale Modeling and Simulation*, 4(3):992–1039, 2005.

[99] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *Image Processing, IEEE Transactions on*, 14(4):423–438, 2005.

[100] B.N. Lee, W.Y. Chen, and E.Y. Chang. Fotofiti: web service for photo management. *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 485–486, 2006.

[101] T.S. Lee. Computations in the early visual cortex. *Journal of Physiology-Paris*, 97(2-3):121–139, 2003.

[102] S.F. Lei and Y.K. Tung. Speech enhancement for nonstationary noises by wavelet packet transform and adaptive noise estimation. In *Intelligent Signal Processing and Communication Systems, 2005. ISPACS 2005. Proceedings of 2005 International Symposium on*, pages 41–44, 2005.

[103] H. Lejsek, F.H. Ásmundsson, B.T. Jónsson, and L. Amsaleg. Scalability of local image descriptors: a comparative study. *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 589–598, 2006.

[104] M. Li, H. G. McAllister, N. D. Black, and T. A. De Perez. Perceptual time-frequency subtraction algorithm for noise reductionin hearing aids. *IEEE Transactions on Biomedical Engineering*, 48(9):979–988, 2001.

[105] X. Li and M. T. Orchard. New edge-directed interpolation. *Image Processing, IEEE Transactions on*, 10(10):1521–1527, 2001.

[106] Z. Li. A Neural Model of Contour Integration in the Primary Visual Cortex. *Neural Computation*, 10(4):903–940, 1998.

[107] Z. Li. Visual segmentation by contextual influences via intra-cortical interactions in the primary visual cortex. *Network: Computation in Neural Systems*, 10(2):187–212, 1999.

[108] Z. Li. Pre-attentive segmentation in the primary visual cortex. *Spatial Vision*, 13(1):25–50, 2000.

[109] Z. Li. Computational Design and Nonlinear Dynamics of a Recurrent Network Model of the Primary Visual Cortex*. *Neural Computation*, 13(8):1749–1780, 2001.

[110] JS Lim and AV Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, 67(12):1586–1604, 1979.

[111] T. Lindeberg. Scale-space theory: a basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1):225–270, 1994.

[112] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-D depth cues from affine distortions of local 2-D brightness structure. *Proc. ECCV*, pages 389–400, 1994.

[113] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *Image and Vision Computing*, 15(6):415–434, 1997.

[114] H. Ling and D.W. Jacobs. Deformation invariant image matching. In *Proc. ICCV*, pages 1466–1473, 2005.

[115] H. Ling and K. Okada. Diffusion Distance for Histogram Comparison. In *Proc. CVPR*, pages 246–253, 2006.

[116] X. Liu and D.L. Wang. Range image segmentation using a relaxation oscillator network. *Neural Networks, IEEE Transactions on*, 10(3):564–573, 1999.

[117] D.G. Lowe. SIFT Keypoint Detector: online demo http://www.cs.ubc.ca/∼lowe/keypoints/.

[118] D.G Lowe. Object recognition from local scale-invariant features. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, 1999.

[119] D.G Lowe. Distinctive image features from scale-invariant key points. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[120] G. Loy and J.O. Eklundh. Detecting symmetry and symmetric constellations of features. *Proceedings of ECCV*, 2:508–521, 2006.

[121] C. T. Lu and H. C. Wang. Enhancement of single channel speech based on masking property and wavelet transform. *Speech Communication*, 39(1):409–427, 2003.

[122] C.T. Lu and H.C. Wang. Speech enhancement using perceptually-constrained gain factors in critical-band-wavelet-packet transform. *Electronics Letters*, 40(6):394–396, 2004.

[123] J. Mairal, M. Elad, G. Sapiro, et al. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53, 2008.

[124] D. Malah, RV Cox, and AJ Accardi. Tracking speech-presence uncertainty to improve speech enhancementin non-stationary noise environments. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1999*, volume 2, 1999.

[125] F. Malgouyres and F. Guichard. Edge direction preserving image zooming: a mathematical and numerical analysis. *SIAM Journal on Numerical Analysis*, pages 1–37, 2002.

[126] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way, 3rd edition*. Academic Press, 2008.

[127] S. Mallat. Geometrical grouplets. *Applied and Computational Harmonic Analysis*, 2008.

[128] S. Mallat and G. Peyré. A review of Bandlet methods for geometrical image representation. *Numerical Algorithms*, 44(3):205–234, 2007.

[129] S. Mallat and G. Peyre. Orthogonal bandelet bases for geometric image approximation. *Comm. Pure Appl. Math.*, 61(9):1173–1212, 2008.

[130] S. Mallat and G. Yu. Structured space pursuits for geometric super-Resolution. In *Invited paper, IEEE ICIP, Egypt*, 2009.

[131] SG Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[132] R. Martin. Noise power spectral density estimation based on optimal smoothingand minimum statistics. *IEEE Transactions on Speech and Audio Processing*, 9(5):504–512, 2001.

[133] R. Martin. Speech enhancement using MMSE short time spectral estimation with gamma distributed speech priors. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2002*, volume 1, 2002.

[134] S. Masnou and J.M. Morel. Level lines based disocclusion. In *Proc. International Conference on Image Processing (ICIP)*, pages 259–263, 1998.

[135] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[136] J. Matas, O. Chum, M. Urban, and T.g Pajdla. Wbs image matcher: online demo http://cmp.felk.cvut.cz/∼wbsdemo/demo/.

[137] G. Matz and F. Hlawatsch. Minimax robust nonstationary signal estimation based on a p-point uncertainty model. *Journal of the Franklin Institute*, 337(4):403–419, 2000.

[138] G. Matz, F. Hlawatsch, and A. Raidl. Signal-adaptive robust time-varying Wiener filters: best subspaceselection and statistical analysis. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01)*, volume 6, 2001.

[139] R. McAulay and M. Malpass. Speech enhancement using a soft-decision noise suppression filter. *IEEE Transactions on acoustics, speech and signal processing*, 28(2):137–145, 1980.

[140] W. Metzger. Laws of seeing. 2006.

[141] Y. Meyer. *Wavelets and operators*. Cambridge University Press, 1992.

[142] C.A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, (2):11–22, 1986.

[143] K Mikolajczyk. http://www.robots.ox.ac.uk/∼vgg/research/affine/.

[144] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. *Proc. ICCV*, 1:525–531, 2001.

[145] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *Proc. ECCV*, 1:128–142, 2002.

[146] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 257–263, June 2003.

[147] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[148] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Trans. PAMI*, pages 1615–1630, 2005.

[149] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1):43–72, 2005.

[150] P. Monasse. Contrast invariant image registration. *Proc. of the International Conf. on Acoustics, Speech and Signal Processing, Phoenix, Arizona*, 6:3221–3224, 1999.

[151] P. Moreels and P. Perona. Common-frame model for object recognition. *Neural Information Processing Systems*, 2004.

[152] P. Moreels and P. Perona. Evaluation of Features Detectors and Descriptors based on 3D Objects. *International Journal of Computer Vision*, 73(3):263–284, 2007.

[153] J.M. Morel and S. Solimini. Variational methods in image segmentation. *Progress in Nonlinear Differential Equations and their Applications*, 14, 1995.

[154] J.M. Morel and G. Yu. On the consistency of the SIFT method. Technical Report Prepublication, to appear in Inverse Problems and Imaging (IPI), CMLA, ENS Cachan, 2008.

[155] J.M. Morel and G. Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[156] N. Mueller, Y. Lu, and M.N. Do. Image interpolation using multiscale geometric representations. In *Computational Imaging V. Edited by Bouman, Charles A.; Miller, Eric L.; Pollak, Ilya. Proceedings of the SPIE*, volume 6498, page 64980A, 2007.

[157] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math*, 42(5):577–685, 1989.

[158] A. Murarka, J. Modayil, and B. Kuipers. Building Local Safety Maps for a Wheelchair Robot using Vision and Lasers. In *Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision*. IEEE Computer Society Washington, DC, USA, 2006.

[159] P. Musé, F. Sur, F. Cao, and Y. Gousseau. Unsupervised thresholds for shape matching. *Proc. of the International Conference on Image Processing*, 2:647–650.

[160] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J.M. Morel. An A Contrario Decision Method for Shape Element Recognition. *International Journal of Computer Vision*, 69(3):295–315, 2006.

[161] J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, Oct. 1962.

[162] A. Negre, H. Tran, N. Gourier, D. Hall, A. Lux, and JL Crowley. Comparative study of People Detection in Surveillance Scenes. *Structural, Syntactic and Statistical Pattern Recognition, Proceedings Lecture Notes in Computer Science*, 4109:100–108, 2006.

[163] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[164] Edwin Olson, Matthew Walter, John Leonard, and Seth Teller. Single cluster graph partitioning for robotics applications. In *Proceedings of Robotics Science and Systems*, pages 265–272, 2005.

[165] P. Perona and W. Freeman. A factorization approach to grouping. *European Conference on Computer Vision*, 1:655–670, 1998.

[166] Q.C. Pham and J.J. Slotine. Stable concurrent synchronization in dynamic system networks. *Neural Netw.*, 20(1):62–77, 2007.

[167] U. Polat and D. Sagi. Lateral Interactions between Spatial Channels: Suppression and Facilitation Revealed by Lateral Masking Experiments. *Vision Research*, 33:993–999, 1992.

[168] D. Pritchard and W. Heidrich. Cloth Motion Capture. *Computer Graphics Forum*, 22(3):263–271, 2003.

[169] S.R. Quackenbush. Objective measures of speech quality. 1995.

[170] J. Rabin, Y. Gousseau, and J. Delon. A contrario matching of local descriptors. Technical Report hal-00168285, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2007.

[171] F. Riggi, M. Toews, and T. Arbel. Fundamental Matrix Estimation via TIP-Transfer of Invariant Parameters. *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 02*, pages 21–24, 2006.

[172] J. Ruiz-del Solar, P. Loncomilla, and C. Devia. A New Approach for Fingerprint Verification Based on Wide Baseline Matching Using Local Interest Points and Descriptors. *Lecture Notes in Computer Science*, 4872:586, 2007.

[173] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or How do I organize my holiday snaps?. *Proc. ECCV*, 1:414–431, 2002.

[174] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. *Proceedings of the 15th international conference on Multimedia*, pages 357–360, 2007.

[175] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2, 2001.

[176] J.W. Seok and K.S. Bae. Speech enhancement with reduction of noise components in the wavelet domain. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97.*, volume 2, 1997.

[177] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411, 2007.

[178] Y. Shao and C.H. Chang. A Generalized Time–Frequency Subtraction Method for Robust Speech Enhancement Based on Wavelet Filter Banks Modeling of Human Auditory System. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(4):877–889, 2007.

[179] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transaction On Pattern Analysis and Machine Intelligence*, pages 888–905, 2000.

[180] W. Singer and C.M. Gray. Visual Feature Integration and the Temporal Correlation Hypothesis. *Annual Reviews in Neuroscience*, 18(1):555–586, 1995.

[181] S. Sinha, J.M. Frahm, M. Pollefeys, et al. GPU-based Video Feature Tracking and Matching. *EDGE 2006, workshop on Edge Computing Using New Commodity Architectures*, 2006.

[182] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Transactions on Graphics (TOG)*, 25(3):835–846, 2006.

[183] K.V. Sørensen. Speech enhancement with natural sounding residual noise based on connected time-frequency speech presence regions. *EURASIP Journal on Applied Signal Processing*, 2005(18):2954–2964, 2005.

[184] J.L. Starck, E.J. Candès, and D.L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, 2002.

[185] C.M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, pages 1135–1151, 1981.

[186] D. Terman and D.L. Wang. Global competition and local cooperation in a network of neural oscillators. *Physica D: Nonlinear Phenomena*, 81(1-2):148–176, 1995.

[187] P. Thevenaz, T. Blu, and M. Unser. Interpolation revisited [medical images application]. *Medical Imaging, IEEE Transactions on*, 19(7):739–758, 2000.

[188] H. Tolba. A time-space adapted wavelet de-noising algorithm for robust automatic speech recognition in low-SNR environments. In *Circuits and Systems, 2003. MWSCAS'03. Proceedings of the 46th IEEE International Midwest Symposium on*, volume 1, 2003.

[189] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242, 2004.

[190] T. Tuytelaars and L. Van Gool. Content-based image retrieval based on local affinely invariant regions. *Int. Conf. on Visual Information Systems*, pages 493–500, 1999.

[191] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. *British Machine Vision Conference*, pages 412–425, 2000.

[192] T. Tuytelaars and L. Van Gool. Matching Widely Separated Views Based on Affine Invariant Regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[193] S. Ullman and A. Shashua. Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network. *ICCV, Clearwater, FL*, 1988.

[194] L. Vacchetti, V. Lepetit, and P. Fua. Stable Real-Time 3D Tracking Using Online and Offline Information. *IEEE Trans PAMI*, pages 1385–1391, 2004.

[195] L.J. Van Gool, T. Moons, and D. Ungureanu. Affine/Photometric Invariants for Planar Intensity Patterns. *Proceedings of the 4th European Conference on Computer Vision-Volume I-Volume I*, pages 642–651, 1996.

[196] V. Velisavljevic and R. Coquoz. Image interpolation with directionlets. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 837–840, 2008.

[197] M. Veloso, F. von Hundelshausen, and PE Rybski. Learning visual object definitions by observing human activities. In *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots,*, pages 148–153, 2005.

[198] M. Vergauwen and L. Van Gool. Web-based 3D Reconstruction Service. *Machine Vision and Applications*, 17(6):411–426, 2005.

[199] P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[200] J.S. Walker and Y.J. Chen. Denoising Gabor Transforms.

[201] D.L. Wang. The time dimension for scene analysis. *Neural Networks, IEEE Transactions on*, 16(6):1401–1426, 2005.

[202] D.L. Wang and G.J. Brown. Separation of speech from interfering sounds based on oscillatorycorrelation. *Neural Networks, IEEE Transactions on*, 10(3):684–697, 1999.

[203] D.L. Wang and P. Chang. An oscillatory correlation model of auditory streaming. *Cognitive Neurodynamics*, 2(1):7–19, 2008.

[204] D.L. Wang and D. Terman. Locally excitatory globally inhibitory oscillator networks. *Neural Networks, IEEE Transactions on*, 6(1):283–286, 1995.

[205] D.L. Wang and D. Terman. Image Segmentation Based on Oscillatory Correlation. *Neural Computation*, 9(4):805–836, 1997.

[206] Q. Wang and RK Ward. A New Orientation-Adaptive Interpolation Method. *Image Processing, IEEE Transactions on*, 16(4):889–900, 2007.

[207] W. Wang and J.J.E. Slotine. On partial contraction analysis for coupled nonlinear oscillators. *Biological Cybernetics*, 92(1):38–53, 2005.

[208] W. Wang and J.J.E. Slotine. Contraction analysis of time-delayed communications and group cooperation. *IEEE Transactions on Automatic Control,*, 51(4):712– 717, 2006.

[209] R.J. Watt and W.A. Phillips. The function of dynamic grouping in vision. *Trends in Cognitive Sciences*, 4(12):447–454, 2000.

[210] M. Wertheimer. Untersuchungen zur Lehre der Gestalt, II. *Psychologische Forschung*, 4:301–350, 1923. Translation published as Laws of Organization in Perceptual Forms, in Ellis, W. (1938). A source book of Gestalt psychology (pp. 71-88). Routledge & Kegan Paul.

[211] PJ Wolfe and SJ Godsill. Simple alternatives to the Ephraim and Malah suppression rule forspeech enhancement. In *Statistical Signal Processing, 2001. Proceedings of the 11th IEEE Signal Processing Workshop on*, pages 496–499, 2001.

[212] K. Yanai. Image collector III: a web image-gathering system with bag-of-keypoints. *Proc. of the 16th Int. Conf. on World Wide Web*, pages 1295–1296, 2007.

[213] G. Yang, CV Stewart, M. Sofka, and CL Tsai. Alignment of challenging image pairs: Refinement and region growing starting from a single keypoint correspondence. *IEEE Trans. Pattern Anal. Machine Intell.*, 2007.

[214] J. Yang. Frequency domain noise suppression approaches in mobile telephonesystems. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993. ICASSP-93.*, volume 2, 1993.

[215] J. Yao and W.K. Cham. Robust multi-view feature matching from multiple unordered views. *Pattern Recognition*, 40(11):3081–3099, 2007.

[216] S.C. Yen and L.H. Finkel. Extraction of perceptually salient contours by striate cortical networks. *Vision Research*, 38(5):719–741, 1998.

[217] G. Yu, E. Bacry, and S. Mallat. Audio signal denoising with complex wavelets and adaptive block attenuation. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, volume 3, 2007.

[218] G. Yu and S. Mallat. Sparse Super-Resolution with Space Matching Pursuits. In *Proc. SPARS'09, Saint-Malo*, 2009.

[219] G. Yu, S. Mallat, and E. Bacry. Audio denoising by time-frequency block thresholding. *IEEE Transactions on Signal Processing*, 56(5):1830–1839, 2008.

[220] G. Yu and J.M. Morel. A fully affine invariant image comparison method. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Taipei*, 2009.

[221] G. Yu and J.J. Slotine. Fast Wavelet-Based Visual Classification. In *Proc. IEEE Int. Conference on Pattern Recognition (ICPR), Tampa*, 2008.

[222] G. Yu and J.J. Slotine. Audio classification from time-frequency texture. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Taipei*, 2009.

[223] G. Yu and J.J. Slotine. Visual grouping by neural oscillators. *accepted to IEEE Trans. Neural Networks*, 2009.

[224] G. Yu and J.J. Slotine. Visual grouping by neural oscillators. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Taipei*, 2009.