Some Geometric Methods for the Analysis of Images and Textures
Gui-Song Xia

To cite this version:

HAL Id: pastel-00682590
https://pastel.archives-ouvertes.fr/pastel-00682590
Submitted on 26 Mar 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Doctorat ParisTech

T H è S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal et Images »

présentée et soutenue publiquement par

Gui-Song Xia
le 18 mars 2011

Méthodes géométriques pour l’analyse
d’images et de textures

Directeur de thèse : Yann Gousseau
Co-encadrement de la thèse : Julie Delon

Jury
M. Jean-François Aujol, Professeur, IMB, Université Bordeaux 1
Mme Isabelle Bloch, Professeur, LTCI, Telecom ParisTech
Mme Julie Delon, CR1 Chercheur, LTCI, Telecom ParisTech
M. Yann Gousseau, Professeur, LTCI, Telecom ParisTech
M. Michael Lindenbaum, Professeur, Technion, Israel Institute of Technology
M. Lionel Moisan, Professeur, MAP5, Université Paris Descartes
M. Pascal Monasse, Chercheur, IMAGINE, Ecole des Ponts ParisTech

TELECOM ParisTech
ecole de l’Institut Télécom - membre de ParisTech
I first express my sincere gratitude to Prof. Hong Sun and Prof. Henri Maître who provided me this great opportunity to work towards my Ph.D degree. I appreciate very much their constant support and attention during my stay in Telecom ParisTech. Their help and encouragement were crucial for my thesis and stay in France.

It is difficult to overstate my gratitude to my Ph.D. supervisors, Dr. Julie Delon and Prof. Yann Gousseau. They have enlightened me through their deep knowledge of image analysis and applied mathematics. Throughout my Ph.D period, they provided encouragement, sound advice, good teaching, good company, and lots of good ideas. I also thank them a lot for their invaluable help during my stay in France. I would have been lost without them.

I would like to thank the jury members of my Ph.D thesis. I thank Prof. Lionel Moisan and Prof. Jean-François Aujol for their detail reports and suggestion on my manuscript. I thank Prof. Isabelle Bloch, Prof. Michael Lindenbaum and Dr. Pascal Monasse for accepting to participate in the jury of my thesis.

I convey my thanks to all the members in TII group for their constant help. I am very lucky to work with them in the large family. Special thanks to Bin Luo, Thomas Hurtut, Julien Rabin, Marie Lienou, Vincent Duval, Adrian Marques, Payam Birjandi, Aymen Shabou, Gabrielle Leheureux, Charles-Alban Deledalle, Thibault Napoléon, Jean-Baptiste Bordes, Baptiste Mazin, Thierry Guillemot, Cecilia Aguerrebere. Many thanks to my friends who also worked toward a Ph.D in France: Haijian Zhang, Lei Yu, Jiayu Chen, Xu Li, Yinghua Wang, Yunlong Cai, Hao Cai, Fei Yuan.

I want to take this opportunity to thank Department of Signal and Image Processing, Telecom ParisTech for providing me all the necessary facilities during my stay and work in France. I also would like to thank the Fondation Telecom and CNRS for funding my PhD work.

I am very grateful to my family for their constant support. I thank them for all their care and encouragement during my Ph.D studies.

I dedicate this work to my wife Tiantian.
Abstract

This thesis focuses on the studies of the extraction and characterization of local image structures, in the context of images and texture analysis. Relying on the level lines of images or on the somehow dual and less structured notion of gradient orientation, the contributions of the thesis concentrate on following three themes:

The first part of this thesis presents a new method for texture analysis that in spirit is similar to morphological granulometries, while allowing a high degree of geometrical and radiometric invariances. The shapes (that is, the interiors of the connected components of level lines) are the basic elements or textons on which the proposed texture analysis is performed. Based on these textons, we exhibited a set of simple statistics, obtained using classical shape moments, invariant to either similarity or affine transformations. Therefore, and because each shape is individually normalized, the proposed texture indexing is invariant to local geometrical transforms, allowing for the recognition of non-planar or even non-rigid textures.

Also using the topographic map representation, the second part of this thesis develops a general approach for the abstraction of images, the aim of which is to automatically generate abstract images from realistic photographs. We suggest a way to abstract images by replacing all geometrical shapes by one or a few reference shapes. As a result, we obtain images that evoke the abstraction painting schools of the beginning of the 20th century while giving a faithful account of an example image structure. In order to achieve this goal, a structured representation of the image, accounting for relationships between objects, is needed. We make use of a hierarchical and morphological representation of images. As a byproduct, we show that the same framework enables the creation of other non-photorealistic renderings.

The subject of the last part of this thesis is the detection of junctions in natural images. The approach relies on the local directions of level lines through the orientation of image gradient. We introduce a generic junction analysis scheme. The first asset of the proposed procedure is an automatic criterion for the detection of junctions, permitting to deal with textured parts in which no detection is expected. Second, the method yields a characterization of L-, Y- and X- junctions, including a precise computation of their type, localization and scale. Contrary to classical approaches, scale characterization does not rely on the linear scale-space, therefore enabling geometric accuracy. First, an a-contrario approach is used to compute the meaningfulness of a junction. This approach relies on a statistical modeling of suitably normalized gray level gradients. Then, exclusion principles between junctions permit their precise characterization. We give implementation details for this procedure and evaluate its efficiency through various experiments.
Résumé court

Cette thèse se concentre sur l’étude de l’extraction et de la caractérisation des structures locales, dans le contexte de l’analyse d’images et des textures. S’appuyant sur les lignes de niveau des images ou sur la notion duale et moins structurée d’orientation du gradient, les contributions de cette thèse se concentrent sur trois thèmes suivants:

La première partie présente une nouvelle méthode pour l’analyse de texture qui dans l’esprit est similaire à la granulométrie morphologique, tout en permettant un haut degré d’invariance géométrique et radioradiométrique. Les formes (défini à l’intérieur des composantes connexes de lignes de niveau) sont les éléments de base ou textons sur lesquelles l’analyse de texture proposée est effectuée. Sur la base de ces textons, nous avons exposé un ensemble de statistiques simples, obtenues en utilisant les classiques moments de forme, invariants soit par similitude soit par transformations. Par conséquent, et parce que chaque forme est individuellement normalisée, l’indexation de texture proposée est invariante à des transformations géométriques locales, permettant la reconnaissance des textures non-planaires et non-rigides.

Avec l’aide de la représentation par carte topographique, la deuxième partie de cette thèse développe une approche générale pour l’abstraction d’images, dont le but est de générer automatiquement des images abstraites à partir de photographies réalistes. Nous proposons un moyen d’abstraire une image en remplaçant toutes les formes géométriques par une ou plusieurs formes de référence. En conséquence, nous obtenons des images qui évoquent les écoles de peinture abstraites du début du 20ème siècle tout en rendant compte fidèlement des structures de l’image. Afin d’atteindre cet objectif, une représentation structurée de l’image, c’est-à-dire qui représente les relations entre les objets, est nécessaire. Nous faisons usage d’une représentation hiérarchique et morphologique des images. À partir de ce résultat, nous montrons que ce cadre permet la création de rendus non-photoréalistes.

Le sujet de la dernière partie de cette thèse est la détection des jonctions dans les images naturelles. L’approche s’appuie sur les directions locales de lignes de niveau à travers l’orientation du gradient de l’image. Nous introduisons un système générique d’analyse de jonction. Le premier avantage de la procédure proposée est un critère pour la détection automatique de jonctions. Celui-ci permet de traiter des parties texturées de l’image dans lesquelles aucune détection n’est attendue. Deuxièmement, la méthode donne une caractérisation des jonction en L-, Y et en X, y compris un calcul précis de leur type, de leur localisation et de leur échelle. Contrairement aux approches classiques, la caractérisation de l’échelle ne repose pas sur un espace-échelle linéaire, et permet donc d’obtenir une bonne précision géométrique. En premier lieu, une approche a contrario est utilisée pour calculer
la significativité d’une jonction. Cette approche repose sur une modélisation statistique des gradients de niveau de gris convenablement normalisés. Dans un second temps, les principes d’exclusion entre jonctions permettent leur caractérisation précise. Nous donnons les détails de la mise en œuvre de cette procédure et évaluons son efficacité au travers de diverses expériences.
## Contents

1 Introduction
   1.1 Motivations of the Thesis ........................................... 23
   1.1.1 Encoding the image structures ................................... 23
   1.1.2 Invariance in the analysis of images and textures .......... 24
   1.2 Topics of the Thesis .................................................. 25
   1.2.1 Handling the geometrical aspects of textures ............... 25
   1.2.2 Automatic abstraction of photographs ....................... 26
   1.2.3 Detecting meaningful junctions in images ................... 27
   1.3 Main Contributions of the Thesis ................................ 28
   1.4 Detailed Outline of the Thesis .................................... 30
   1.5 Publications Related to the Thesis ................................ 32
   1.6 Other publications during the Ph.D ................................ 33

I Shape-based Texture Analysis ........................................... 35

2 A Generic Shape-based Texture Analysis Method ................. 37
   2.1 Introduction .......................................................... 37
   2.1.1 A brief view on texture analysis ............................... 37
   2.1.2 Motivations for shape-based invariant texture analysis .... 40
   2.1.3 Invariant texture analysis: previous and related work ...... 41
   2.1.4 Contributions ..................................................... 43
   2.2 Topographic Map ...................................................... 44
   2.2.1 Definition and properties ....................................... 44
   2.2.2 Topographic map and textures .................................. 46
   2.3 Invariant Texture Descriptors ..................................... 47
   2.3.1 Marginals of invariant moments ................................ 48
   2.3.1.1 Invariant moments reminder ................................. 49
   2.3.1.2 Texture features from second order moments ............. 50
   2.3.2 Dependencies in the topographic map .......................... 51
   2.3.3 Contrast information ............................................. 53

3 Experimental Analysis of the Indexing Scheme .................. 55
   3.1 Descriptors Comparison ............................................. 55
   3.2 Comparative Evaluations ............................................ 56
<table>
<thead>
<tr>
<th>Section Number</th>
<th>Section Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1</td>
<td>Experimental protocols</td>
<td>56</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Databases</td>
<td>57</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Performances on Brodatz</td>
<td>57</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Comparisons on UIUC Database</td>
<td>60</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Comparisons on UMD Database</td>
<td>62</td>
</tr>
<tr>
<td>3.3</td>
<td>On Invariance and Discriminative Power</td>
<td>64</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Invariance to resolution changes</td>
<td>64</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Local invariance vs discriminative power</td>
<td>66</td>
</tr>
<tr>
<td>3.4</td>
<td>Scaling Behavior of the Analysis Scheme</td>
<td>69</td>
</tr>
<tr>
<td>4</td>
<td>Three Applications of the Theory</td>
<td>71</td>
</tr>
<tr>
<td>4.1</td>
<td>Texture Segmentation from Shape Ensembles</td>
<td>71</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Partitioning the tree of shapes is not enough for segmentation</td>
<td>71</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Segmentation with active contours</td>
<td>72</td>
</tr>
<tr>
<td>4.2</td>
<td>Structured Satellite Image Indexing</td>
<td>76</td>
</tr>
<tr>
<td>4.3</td>
<td>Image Retrieval based on Statistical Manifolds</td>
<td>83</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Fisher information distance for image comparison</td>
<td>84</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Adaptive image retrieval based on statistical manifolds</td>
<td>84</td>
</tr>
<tr>
<td>5</td>
<td>A Generic Structured Image Manipulation Framework</td>
<td>93</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>93</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Related work</td>
<td>95</td>
</tr>
<tr>
<td>5.2</td>
<td>Organizing Shapes by Topographic Maps</td>
<td>96</td>
</tr>
<tr>
<td>5.3</td>
<td>A Flexible Image Manipulation Framework</td>
<td>97</td>
</tr>
<tr>
<td>5.3.1</td>
<td>General outline</td>
<td>97</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Image abstraction</td>
<td>100</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Painting-like effects</td>
<td>101</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparison with Arty Shapes</td>
<td>107</td>
</tr>
<tr>
<td>5.5</td>
<td>Image Filtering</td>
<td>110</td>
</tr>
<tr>
<td>5.6</td>
<td>Discussion</td>
<td>120</td>
</tr>
<tr>
<td>III</td>
<td>Meaningful Junction Detection</td>
<td>135</td>
</tr>
<tr>
<td>6</td>
<td>Definition and Detection of Junctions</td>
<td>137</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>137</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Previous and related works (part 1)</td>
<td>137</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Motivations</td>
<td>140</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Previous and related works (part 2 : invariances and scale selection)</td>
<td>140</td>
</tr>
<tr>
<td>6.1.4</td>
<td>Contributions and outline of the chapter</td>
<td>141</td>
</tr>
<tr>
<td>6.2</td>
<td>Contrast Invariant Junctions</td>
<td>141</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Local normalization of the gradient</td>
<td>141</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Discrete junctions</td>
<td>142</td>
</tr>
</tbody>
</table>
CONTENTS

6.2.3 Junction strength ........................................... 143
6.3 An a contrario Approach for Junction Detection .................. 146
   6.3.1 Null hypothesis .......................................... 146
   6.3.2 Distribution of $t(j)$ under $H_0$ .......................... 147
   6.3.3 Meaningful junctions .................................... 149
   6.3.4 Minimum scale of detection .............................. 151
   6.3.5 Maximality ............................................. 151
   6.3.6 The final algorithms for junction detection ............... 153
6.4 Implementation .................................................. 155
   6.4.1 Computing NFA in practice .............................. 155
   6.4.2 Choice of $K(r)$ and $\Delta(r)$ .......................... 156
   6.4.3 Direction refinement ..................................... 157
   6.4.4 Speed up ................................................. 158
   6.4.5 Algorithm pipeline ....................................... 160

7 Detecting Junctions in Natural Images ............................. 163
   7.1 Controlling the Number of False Detections .................. 163
      7.1.1 False alarms when $H_0$ is “almost” satisfied ......... 163
      7.1.2 Detections in Gaussian noise ........................... 165
   7.2 Scale Invariance ............................................ 166
      7.2.1 Scale and resolution in the proposed approach ......... 166
      7.2.2 Comparison with classical methods ..................... 171
   7.3 Contrast Invariance .......................................... 174
      7.3.1 Total contrast invariance ............................... 174
      7.3.2 Comparison with classical approaches .................. 176
   7.4 Comparison Evaluation on Junction Benchmark ................. 176
   7.5 More Experiments ........................................... 180

8 Conclusion and Perspectives ...................................... 193

A Appendix ......................................................... 199
   A.1 Distribution of $\max(|\cos \theta| - |\sin \theta|, 0)$ .............. 199
   A.2 Distribution of $|\nabla I| \cdot \max(|\cos \theta| - |\sin \theta|, 0)$ ...... 200

Bibliography ........................................................ 216
Résumé

Cette thèse s’intéresse à l’utilisation de méthodes géométriques dans l’analyse d’images et des textures.

Les approches proposées dans ce manuscrit soit s’appuient sur les lignes de niveau des images ou sur la notion en quelque sorte duale et moins structurée d’orientation du gradient. La première partie de cette thèse présente un schéma de forme basée sur l’analyse de texture fondée sur la carte topographique, qui est une organisation arborescente des lignes de niveau des images. Aussi avec l’aide de la représentation de la carte topographique, la deuxième partie de cette thèse développe une approche générale pour le captage d’images, dont le but est de générer automatiquement des images abstraites à partir de photographies réalistes. Le sujet de la dernière partie de cette thèse est la détection des jonctions dans les images naturelles. L’approche s’appuie sur les directions locales de lignes de niveau grâce à l’orientation du gradient de l’image.

Avant de présenter les contributions de cette thèse en détail, nous rappelons brièvement quelques faits de base sur l’analyse d’images et de textures et de présenter les motivations de la thèse.

Introduction

Encodage des structures d’images

Dans ce manuscrit, par “structures”, nous nous référerons à des structures géométriques. Les objets du monde réel peuvent apparaître de multiples façons à la caméra. Les structures sont présentes dans les images avec un large éventail d’échelles spatiales, sauf quand elles sont acquises dans des conditions contrôlées. Koenderink a souligné au début de son travail [108] que le réel défi de l’analyse d’image est de comprendre simultanément les structures d’image à tous les niveaux de résolution.

Dans la communauté de vision par ordinateur, l’espace échelle linéaire [121] a été largement appliqué pour manipuler la nature multi-échelles des images. Il représente une image comme une famille d’image avec différents niveau de lissage, par filtrage de l’image originale avec des noyaux gaussiens de taille croissante. Il est connu qu’une telle approche amène une forte perte d’information, et qu’elle implique une perte de précision dans la description des structures et aussi qu’elle peut introduire des structures parasites. Les représentations pyramidales [29, 50] et l’analyse multirésolution [130] ont également été largement utilisées avec succès pour la détermination des structures de l’image à différents niveaux de résolutions. Une limitation principale de ces méthodes est leur manque d’efficacité à représenter...
des morceaux très structurées d’images.

L’école de morphologie mathématique a depuis longtemps élaboré d’autres représentations que les représentations non-linéaires et multi-échelle. L’une des plus élégantes est la carte topographique [37], qui est élaborée à partir des lignes de niveau, c’est-à-dire des composantes connexes des limites topologiques des ensembles de niveau. Cette représentation hérite d’une structure d’arbre à partir des propriétés de nidification des ensembles de niveau et nous permet de complètement représenter l’information géométrique et radiométrique d’une image en même temps, tout en permettant la manutention séparée de ces deux aspects. Cette décomposition a été appliquée avec succès à une variété de problèmes en traitement d’image et en vision par ordinateur [144, 123, 30, 124, 128]. Comparé avec d’autres représentations multi-échelles, elle possède de bonnes propriétés qui lui permettent de faire face à des images très structurées. Une première motivation de cette thèse est d’étudier l’utilisation de lignes de niveau pour le codage de textures structurées, en faisant usage de deux des aspects structurés et multi-échelle de cette représentation morphologique. Cette partie du travail est fortement inspirée par la notion de granulométrie de l’école morphologie mathématique, comme nous le verrons plus en détail.

**Invéniance dans l’analyse d’images et de textures**

Lorsque l’on traite avec des images naturelles, l’une des difficultés est que les images de la même scène peuvent souffrir de perturbations géométriques inconnues ainsi que de variations d’illumination.

Par exemple, la figure 1 affiche trois images de la même scène prises à partir de différents points de vue. On observe que les structures de l’image appartenant à ces mêmes objets peuvent avoir des appariences très différentes. En outre, il y a des cas où il peut y avoir des déformations non-rigides entre les images. Un exemple typique est une texture situé sur une surface non plane ou non-rigide: ce type de texture subit des déformations complexes, voir Figure 2 pour une représentation graphique.

![Figure 1: Trois images de la même scène sous différents points de vue.](image)

Ces phénomènes exigent certaines propriétés d’invariances dans les méthodes d’analyse d’images. Les plus élémentaires sont l’invariance par similitude et par transformations...
affines, ou moins souvent par homographies, qui correspondent à des déformations induites par des changements particuliers de point de vue entre scènes planaires. Pour manipuler ces variations non-planaires ou non-rigide, l’invariance locale est préférable [169, 113, 14].

Dans la dernière décennie, la méthode la plus populaire pour atteindre ces invariances pour l’analyse d’image a été le descripteur invariant par changement d’échelle (SIFT) [126], qui s’appuie sur l’espace échelle linéaire. Cet algorithme sélectionne tout d’abord la taille des structures comme des extrema dans l’espace échelle et normalise ensuite les variations de description d’image. Comme mentionné précédemment, en raison de l’utilisation du filtre gaussien, la localisation des structures d’intérêt devient imprécise et les échelles choisies sont souvent difficiles à interpréter et peuvent ne pas correspondre à l’ampleur réelle des structures. En outre, il n’est pas trivial de faire face aux changements de contraste plus complexes que des changements affines en utilisant l’espace échelle linéaire.

Une autre motivation de cette thèse est de démontrer l’utilité de lignes de niveau pour atteindre invariances locales dans le cas des champs homogènes (textures) qui subissent des déformations très forts, au lieu d’utiliser l’espace échelle linéaire comme dans [113, 229]. Dans l’analyse de texture particulière invariante sera réalisée directement sur les formes, de l’intérieur des lignes de niveau, sans utiliser de représentation à l’échelle linéaire de l’espace. Toujours en relation avec des invariances, nous allons expérimentalement montrer qu’il devrait y avoir un compromis entre les propriétés d’invariance et le pouvoir discriminant des méthodes d’analyse d’image.

Dans la troisième partie de la thèse, les jonctions seront extraites en s’appuyant sur les directions locales le long des lignes de niveau et de leur importance sera automatiquement calculé. En particulier, l’une des principales contributions de la thèse est de montrer que l’échelle d’une structure (dans ce cas une jonction) peut être déterminée en utilisant un principe de maximalité, menant au choix de l’échelle à laquelle la structure est la plus significatif, dans le sens de Desolneux-Moisan-Morel [58]. Même si cette partie de la thèse ne traite que de la détection des jonctions, les informations d’échelle peuvent être utilisées comme une alternative à l’espace échelle linéaire pour le codage de structures invariantes.
Problématique

Manipulation des aspects géométriques de textures

La texture est un ingrédient fondamental de la structure des images naturelles. Au cours des 4 dernières décennies, de nombreuses recherches ont été réalisées sur l’analyse de texture, comme par exemple [87, 75, 114, 208, 232, 191, 159]. Cependant, la plupart de ces approches ont des difficultés à représenter les aspects géométriques des textures, comme les transitions et les contours allongés. Afin de surmonter cette difficulté, des approches alternatives ont été développées pour fournir des représentations plus efficaces des textures structurées, telles que grouplets [159], des approches comme les ondelettes, et la décomposition structure-texture [10, 8], une méthode pour décomposer les images aux structures et textures, puis traitement chaque morceau séparément.

D’autres solutions au problème sont les méthodes géométriques menées par le théorie des Textons [100], qui remarque que seuls les statistiques du premier ordre de quelques éléments localement remarquables, appelées textons, comme par exemple les extrémités de la ligne et des blobs, sont significatifs pour la discrimination. Toutefois, extraire efficacement les textons est loin d’être trivial. Certains algorithmes ont été proposés pour détecter des structures explicitement répétées dans des textures [111, 112, 113] et ont démontré de bonnes performances sur un ensemble d’images très structurées.

Dans l’école de morphologie mathématique, la granulométrie [131, 187] a été proposée pour caractériser les textures en invoquant les distributions de taille de certains éléments de structure prédéfinis. L’idée est proche de celle de la théorie des textons, avec comme textons prédéfinis de petits cercles, des rectangles et des diamants, éléments dont la nature répétitive est transmise par les distributions.

La partie de la thèse sur l’analyse de texture essaie de combler l’espace entre la granulométrie et la théorie des Textons en introduisant des formes géométriques.

Abstraction automatique des photos

Le rendu non-photoréaliste (NPR) [83, 215, 125, 92] est un domaine de l’infographie qui se concentre sur la production ou l’imitation d’une grande variété de styles artistiques expressifs, généralement à partir les images naturelles.

Une tendance constante dans le domaine du NPR a été d’utiliser des outils d’analyse d’images pour guider la phase de rendu [83, 48, 16, 15, 154, 195]. La principale difficulté d’une telle tâche est de rendre compte des structures de l’image à différentes échelles et de tenir compte de leurs interactions.

La partie de la thèse sur le NPR est inspirée par les peintures abstraites géométriques, en particulier des peintures, telles que les œuvres exposées dans la Figure 3.

Remarquez que ces œuvres sont souvent composées d’un petit ensemble de formes, telles que des rectangles, des cercles ou autres formes simples. De façon analogique à la peinture abstraite géométrique, nous allons présenter un moyen de créer automatiquement des images abstraites. Ce travail est étroitement lié à des expériences menées par Alvarez-Gousseau-Morel [4], qui ont systématiquement étudié le domaine des images abstraites qui peuvent être créées en combinant les principes de base tels que l’occlusion, l’exclusion ou l’inclusion de simple formes géométrique. En contraste avec ce travail, nous étudions la
Figure 3: Quelques exemples de peintures abstraites. De gauche à droite: le *Composition VIII* et *Plusieurs cercles* par Wassily Kandinsky, et le *Composition A* de Piet Mondrian.

création d'images abstraites en manipulant toutes les formes d’une image naturelle donnée ou en les remplaçant par des formes géométriques. La structure de l’image créée est donc héritée de la structure de l’image naturelle et non créée ; et ce par placement stochastique et en utilisant les règles d’interaction.

![Figure 4](image)

Figure 4: Illustrations de la détection de jonction. **gauche:** choix de l’emplacement et l’échelle utilisant de Harris-Laplace. Localisations et échelles sont choisis comme des maxima locaux d’un opérateur différentiel à l’échelle et l’espace. **droite:** jonctions détectées par Harris sur une image de bruit gaussien. La couleur (du bleu au rouge) de chaque jonction indique son importance (de bas en haut). Pas de jonction doit être détectée dans cette image.

**Détection jonctions significatives dans les images**

Les jonctions, sont des points dans les images où les bords se rejoignent ou se croisent, sont fréquemment utilisé comme fonctions de base de la vision par ordinateur pour la
la compréhension de l’image. La détection des jonctions pose deux problèmes : trouver les intersections de bords et l’identifier l’échelle des structures. Le détecteur de Harris [88] est l’une des méthodes populaires pour localiser les jonctions, mais se traduit souvent par de nombreuses fausses détections, voir la figure 4b, car il n’y a pas de critère d’adaptation déterminer l’intérêt d’une jonction. C’est cela qui motive la détection de jonction significative dans les images. Le modèle de jonctions que nous proposons utilise les événements visuels constitués de la réunion non-accidentelle de plusieurs bords dans une certaine configuration. Plus précisément, nous faisons usage de l’a contrario, méthodologie proposée par [58] qui prête une attention particulière aux statistiques de l’image et à certains principes comme la maximalité. Dans cette partie, l’utilisation de l’orientation du gradient est préférée à l’utilisation directe des lignes de niveau pour des raisons de robustesse, en particulier pour être robuste aux changements d’illumination. Notons que la sensibilité relative des lignes de niveau à l’illumination n’est pas un problème lors de l’analyse des textures, car dans ce cas nous prêtons attention à la répartition des structures.

Pour la sélection de l’échelle, la plupart des approches classiques s’appuient sur l’espace échelle linéaire [120, 185, 126]. Comme mentionné dans la section 1.1.1, les lacunes de ce genre d’approches sont qu’elles perdent à la fois la précision dans la localisation et dans l’échelle, voir la figure 4a pour une illustration graphique. Sinon, nous allons montrer qu’en vérifiant les directions locales de lignes de niveau à différentes tailles de patch et en comparant leur significativité, nous pouvons détecter les jonctions avec une invariance d’échelle. Par ailleurs, nous sommes en mesure d’identifier les branches de jonction (à la fois leur nombre et leurs emplacements) et l’emplacement de la jonction avec le même principe.

Méthodologies et résultats

L’analyse de textures invariante basée sur formes

L’analyse des texture est un long et difficile problème du traitement d’image et de la vision par ordinateur. Yves Meyer a récemment défini la texture comme étant “un subtil équilibre entre la répétition et l’innovation” [137]. Bien qu’il existe d’énormes contributions à la littérature, beaucoup de travail reste à faire pour mieux comprendre la nature stochastique et la répétition des textures, en particulier, des textures haute résolution, où apparaissent de grandes structures (par exemple les contours allongés).

J’ai présenté une nouvelle méthode pour l’analyse de texture qui dans l’esprit est similaire à la granulométrie morphologique, tout en autorisant un haut degré d’invariance géométrique et radiométrique. L’idée était de décomposer la texture en formes en s’appuyant sur une méthode de décomposition en lignes de niveau, à savoir la carte topographique [37]. Ensuite, nous calculons et construisons des descriptor invariants de texture et de à partir de ces représentations basées sur les formes.

Les formes (l’intérieur des composantes connexes de lignes de niveau) sont les éléments de base ou textons sur lesquelles l’analyse de texture proposée a été réalisée. Comme la carte topographique fournit une représentation complète de l’image, la modélisation de la texture t a été remplacée par la modélisation de l’arbre de formes (S,T), tel que p(t) = p(T,S). Les caractéristiques des texture ont été obtenus à partir de (S,T) et il
était donc assez naturel de considérer les moments classiques des formes invariantes de $S$. Nous avons proposé d’utiliser les moments invariant du deuxième ordre [96]. Afin d’enrichir l’analyse proposée, nous avons pris en compte les dépendances de forme multi-échelles sur la structure de $t$ de la carte topographique, en tant que

$$\alpha(s) = \frac{\mu_{00}(s)}{\langle \mu_{00}(s') \rangle}_{s' \in N^M},$$

où $\mu_{00}(s)$ est le domaine des formes $s$ et $\langle \cdot \rangle _{s' \in N^M}$ est l’opérateur moyenneur sur la famille ancêtre, $N^M$, de $s$. Comme chaque forme a été normalisée individuellement, l’indexation proposée des textures est invariante par transformations géométriques locales, ce qui permet la reconnaissance des textures non-planaires ou même non-rigides. Différentes expériences relatives à la classification et à l’extraction des textures ont démontré l’efficacité de la méthode d’analyse proposée sur plusieurs bases de données difficiles, voir Figure 5.

Figure 5: Les meilleurs résultats de récupération sur la base de données UIUC [113], obtenu sur la classe T15 texture en utilisant le descripteur proposé (Similarity Invariant + histogramme de contraste). L’image requête est en première position et les 39 échantillons les plus similaires suivent, classés en fonction de leurs scores. Il est intéressant de remarquer qu’aucun apprentissage n’est impliqué dans ces expériences. Les résultats de récupération pour tous les échantillons de texture sont disponibles à l’adresse [220].

Basé sur la représentation de texture basée sur des formes, j’ai proposé de représenter une image de texture par un arbre d’ellipses, puis d’examiner les propriétés statistiques sur
l’arbre d’ellipses et enfin d’obtenir une segmentation de l’image de texture en regroupant toutes les ellipses dans plusieurs sous-ensembles selon certaines mesures statistiques. Le processus de regroupement a été formulé comme un problème de minimisation énergétique et la solution a été obtenue par évolution d’un contour actif basé sur la divergence de Kullback-Leibler (KL), grâce à une méthode de minimisation globale rapide. Grâce aux caractéristiques à base d’ellipse, la méthode de segmentation peut intégrer l’information locale et globale dans l’image, voir Figure 5 pour des exemples.
Rendu non-photoréaliste fondé sur des structures

Cette partie concerne l’abstraction géométriques de photos numériques en utilisant la carte topographique. En particulier, nous avons proposé un moyen d’abstraire les images en remplaçant toutes les formes géométriques par une ou plusieurs formes de référence seulement. Ainsi, nous obtenons des images qui évoquent les écoles de peinture abstraites du début du 20ème siècle tout en donnant un compte rendu fidèle d’une structure d’image.

Figure 7: Forme de transfert à partir d’un dictionnaire. **Haut (de gauche à droite):** image originale, premier dictionnaire et second dictionnaire. **Bas (de gauche à droite):** résultat du transfert de formes en utilisant les formes à partir du premier dictionnaire, puis celles du second. Des images haute résolution peut être consultée à l’adresse [221].

Je décris brièvement l’approche comme suit: Premièrement, la structure géométrique de l’image est analysée à travers l’utilisation de la carte topographique. Puis, la géométrie de ces formes est modifiée. Finalement, une nouvelle image est créée à partir de l’arbre résultant. Les modifications géométriques proposées des formes allant de simples déplacements aléatoires des formes, à la création d’effets simulant une peinture, en passant par le remplacement de toutes les formes de l’image par des formes provenant d’un dictionnaire, en suivant certaines règles de remplacement.

Les points forts de l’approche proposée sont les suivants. Tout d’abord, l’image est
représentée par une collection de formes, ce qui donne une prise directe sur les manipulations géométriques. Deuxièmement, la carte topographique pour toutes les échelles de l’image, permet la modélisations des relations entre les textures et les structures macroscopiques de l’image. Enfin, la structure hiérarchique de la représentation choisie permet aux propriétés d’inclusions entre les formes d’être partiellement conservés dans l’image synthétisée, ce qui donne un aspect fortement structurés aux images abstraites. Certains résultats sont présentés dans la figure 7.

Précise détection de jonctions

Les jonctions qui sont des structures locales spécifiques des images, sont de première importance pour la perception visuelle et la compréhension d’une. Ils font partie du croquis primitif de la représentation schématique des images apportées par D. Marr [132]. Les approches récentes de calcul de ce croquis montrent le rôle clé joué par les jonctions. En fonction du nombre de bords qu’elles relient, les jonctions sont souvent classées en L, Y (ou T-) et X. En particulier, le rôle des jonctions en T comme indicateur de la perception des occlusions a été bien étudié par G. Kanisza [104]. Plus tard, il a été montré que les jonctions sont d’essentiel indices locaux et que leur configuration spécifique (par exemple en T ou en Y) doit être pris en compte dans ce processus. Les rôles distincts des jonctions L et en T pour la perception du mouvement, en particulier par le phénomène d’ouverture, est connu depuis longtemps. Les types de jonction et leurs positions ont également montrés qu’ils ont un fort impact sur la perception de la luminosité et de la transparence. Les jonctions sont donc naturellement utilisées comme des indices importants pour différentes tâches de vision par ordinateur. De plus, ils révèlent des relations importantes sur les occlusions entre les objets, ils sont impliqués dans la segmentations d’images et la reconnaissance d’objets.

Nous avons introduit un régime d’analyse des jonctions, en s’appuyant sur la cadre de a-contrario [58]. Nous avons modélisé les jonctions comme des configurations particulières et locales des orientations de gradient qui sont peu susceptibles d’arriver par hasard. Plus précisément, une jonction est définie comme une structure d’image discrète

\[ j : \{p, r, \{\theta_m\}_{m=1}^{M}\} \]

caractérisée par son centre \( p \), son échelle \( r \in \mathbb{N} \) et un ensemble de directions de branches \( \{\theta_1, \ldots, \theta_M\} \) autour de \( p \). Soit I est une image discrète. Pour \( \epsilon > 0 \), une jonction \( j \) de l’ordre \( M \) et à l’échelle \( r \) est dit \( \epsilon \)-significative, si

\[ \text{NFA}(j) := \#J(M) \cdot F_j(t(j)) \leq \epsilon, \]

avec \( \#J(M) \) le nombre de toutes les détections possibles, \( t(j) \) la force de jonction basé sur des gradients locaux et \( F_j(t(j)) \) la probabilité de la jonction \( j \). Nous avons montré que le calcul de \( F_j(t(j)) \) peut être mis en œuvre sous une forme fermée fondée sur les statistiques locales de gradients d’image, sous une hypothèse, appelée hypothèse nulle. La quantité \( \text{NFA}(j) \) est en fait une mesure de l’importance perceptive de la jonction ; plus petit il est, plus la jonction \( j \) est significative. Cette significativité perceptive \( \text{NFA}(j) \) est alors utilisée pour élaborer un ensemble de principes d’exclusion pour la détection et l’estimation de
l’emplacement, l’échelle, ainsi que le type de jonction.

Figure 8: Sélection de l’échelle. (a) montre l’échelle caractéristique des jonctions en L, Y et X données par notre approche, et (b) montre l’échelle caractéristique des jonctions donnée par Harris-Laplace sur une image synthétique. (c) et (d) montrent la même comparaison sur l’image House. L’emplacement des jonctions est noté par une croix rouge et l’échelle est illustrée par un cercle jaune. Voir le texte pour plus d’explications.

La première contribution de la procédure proposée est un critère pour la détection automatique des jonctions, permettant de traiter des régions texturées dans lesquelles aucune détection n’est prévue. Deuxièmement, la méthode fournie une caractérisation des jonctions en L, Y et en X, y compris un calcul précis de leur type, de leur localisation et de leur d’échelle. Contrairement aux approches classiques, la caractérisation de l’échelle ne repose pas sur l’espace-échelle linéaire, et permet ainsi une bonne précision géométrique.

Aussi grâce au modèle a contrario, un seul paramètre est nécessaire pour contrôler la détection.

Conclusion

La première partie de ce manuscrit propose un système fondé sur les formes pour l’analyse de texture invariantes en s’appuyant sur la carte topographique, une représentation morphologique multi-échelle des des images. Les formes contenues dans la carte topographique peuvent être considérées comme des textons dans le contexte de l’analyse de texture. Les caractéristiques locales des textures sont calculés à partir des statistiques de ces formes.
Nous avons montré que les méthodes morphologiques d’indexation peuvent traiter des images de texture très structurées et constituer l’état-de-l’art des résultats dans l’analyse de texture invariantes, même dans le cas de transformations non-rigides. En outre, nous avons appliqué avec succès les descripteurs de texture proposées, combinées un modèle de contour actif [95], au problème de la segmentation de texture. L’approche proposée a également démontré une bonne performance sur la reconnaissance des images satellites haute résolution. Enfin, nous avons suggéré que la comparaison de textures avec les des distances géodésiques sur la variété statistique formée par PDFs améliore significativement les performances de reconnaissance. Basé sur de telles distances géodésiques, une méthode a-contrario a été introduite pour définir le seuil correspondant dans le processus de recherche des images similaires dans une base de données avec un exemple donné.

Dans la deuxième partie de cette thèse, nous nous sommes concentrés sur un problème de NPR et avons montré que la carte topographique permet la création d’abstractions géométriques à partir de photographies numériques. La méthode proposée peut être utilisée pour produire de nombreux effet d’abstraction. En contraste avec les méthodes correspondantes dans le domaine de la NPR, la procédure de synthèse présentée dans cette partie permet des interactions complexes entre les formes. De plus, la carte topographique consiste en une structure hiérarchique, conduite par l’inclusion de formes pour toutes les échelles de l’image. En outre, nous avons proposé le filtre scale-ratio, qui permet de supprimer les structures d’image selon le rapport d’échelle entre les formes et leurs parents le long de la carte topographique. Enfin, la dernière partie de cette thèse a proposé une approche pour la détection de jonctions significatives dans des images naturelles. Nous nous occupons de la détection de la jonction comme étant un problème de groupement. En particulier, les jonctions ont été prises comme étant des endroits dans des images où se croisent différents bords.

La détection des jonctions dans une image est ensuite dirigée par un modèle a contrario, établi à partir des propriétés statistiques de l’image. La méthode proposée permet d’identifier l’emplacement, une échelle, et les branches de chaque jonction détectées simultanément. Il est intéressant de remarquer que tous ceux-ci est réalisé sans l’aide d’un espace échelle linéaire, qui est largement utilisé dans les approches classiques pour l’intérêt des points de détection, tels que SIFT [126]. En contraste avec les œuvres précédentes traitant de détections de jonction, la méthode proposée trouve des jonctions qui ne sont pas seulement localement saillantes, mais aussi significative par rapport à la perception globale de l’image. Une significativité perceptive est en fait associée à chaque jonction et peut être utilisé pour contrôler la détection. En outre, la méthode proposée est robuste aux changements d’échelle et de contraste. Et de plus, par rapport aux méthodes classiques de détection de jonction, notre méthode nécessite moins de paramètres.
Chapter 1

Introduction

This thesis focuses on the use of geometric methods in the analysis of images and textures. Investigations in the manuscript either rely on the level lines of images or on the somehow dual and less structured notion of gradient orientation. The first part of this thesis presents a shape-based texture analysis scheme grounded on the topographic map [37], a tree organization of the level lines of images. Also using the topographic map representation, the second part of this thesis develops a general approach for the abstraction of images, the aim of which is to automatically generate abstract images from realistic photographs. The subject of the last part of this thesis is the detection of junctions in natural images. The approach relies on the local directions of level lines through the orientation of image gradient.

Before presenting the contributions of this thesis in detail, we briefly recall some basic facts about the analysis of images and textures and present the motivations of the thesis.

1.1 Motivations of the Thesis

1.1.1 Encoding the image structures

By "structures", we refer to "geometric structures" in this manuscript. Since real-world objects may appear in unknown ways to the camera, structures are present within images at a wide range of spatial scales, except when dealing with a controlled acquisition protocol. Koenderink pointed out in his early work [108] that the actual challenge of image analysis is to understand image structures at all levels of resolution simultaneously.

In the computer vision community, the linear scale space [121] has been widely applied to handle the multi-scale nature of images. It represents an image as a one-parameter family of smoothed images, by blurring the original one with Gaussian kernels with increasing size. It is known that such an approach yields a strong loss of information, that it implies a loss of precision in the description of structures and also that it may produce spurious structures. Pyramidal representations [29, 50] and multiresolution analysis [130] have also been extensively and successfully used for computing the image structures at different resolutions. One main limitation of these methods is their lack of efficiency at representing highly structured parts of images.

The mathematical morphology school has long ago developed alternative, non-linear
and multi-scale representations. One of the most elegant is the topographic map [37], which is made of the level lines, that is the connected components of the topological boundaries of the level sets. This representation inherits a tree structure from the nesting properties of level sets and allows us to completely represent the geometric and radiometric information of an image simultaneously, while allowing the separate handling of both these aspects. This decomposition has been successfully applied to a variety of problems in image processing and computer vision [144, 123, 30, 124, 128]. Compared with other multi-scale representations, it demonstrates good properties to deal with highly structured images. A first motivation of this thesis is to investigate the use of level lines for the encoding of structured textures, making use of both the structured and multi-scale aspects of this morphological representation. This part of the work is strongly inspired by the notion of granulometry from the Mathematical Morphology school, as we will see in detail further.

1.1.2 Invariance in the analysis of images and textures

When dealing with natural images, one challenge is that images of the same scene may suffer from unknown geometric perturbations as well as illumination variations. For instance, Figure 1.1 displays three images taken from different viewpoints of the same scene. Observe that the image structures belonging to the same objects may have very different appearances. Moreover, there are cases where there may be non-rigid deformations between images. A typical example is the one of textures lying on non-planar or non-rigid surfaces: these textures undergo complex deformations, see Figure 1.2 for a graphic illustration.

![Figure 1.1: Three images of the same scene under different viewpoints.](image)

These phenomena demand some invariant properties from the image analysis methods. The most basic ones are invariances to similarity and affine transformations, or less often to homography, which correspond to deformations induced by specific viewpoint changes on planar scenes. For handling non-planar or non-rigid variations, one prefers local invariances [169, 113, 14]. In the past decade, the most popular method to achieve such invariances for image analysis has been the scale-invariant feature transform (SIFT) [126], which relies on the linear scale space. This algorithm first selects the size of structures
1.2. Topics of the Thesis

Figure 1.2: Samples from the UIUC database [113]. This texture lies on a non-rigid surface resulting in complex deformations between the samples.

as extrema in the scale space and then normalizes the variations for image description. As mentioned before, because of the use of Gaussian blur, the localization of the interest structures becomes imprecise and the selected scales are often difficult to interpret and may not correspond to the true scale of the structures. Furthermore, it is not trivial to deal with contrast changes beyond affine changes by using the linear scale space.

Another motivation of this thesis is to demonstrate the usefulness of level lines for achieving local invariances in the case of homogeneous fields (textures) undergoing very strong deformations, instead of using the linear scale space as in [113, 229]. In particular invariant texture analysis will be performed directly on shapes, the interior of level lines, without using any linear scale space representation. Still in relation with invariances, we will experimentally show that there should be a trade-off between the invariant properties and the discriminative power of image analysis methods.

In the third part of the thesis, junctions will be extracted by relying on the local directions along level lines and their scale will be automatically computed. In particular, one of the main contribution of the thesis is to show that the scale of a structure (in this case a junction) may be determined by using a maximality principle, boiling down to the choice of the scale at which the structure is the most meaningful, in the sense of Desolneux-Moisan-Morel [58]. Even though this part of the thesis only deal with the detection of junctions, the scale information can be used as an alternative to the linear scale space for the invariant coding of structures.

1.2 Topics of the Thesis

1.2.1 Handling the geometrical aspects of textures

Texture is a fundamental ingredient of the structure of natural images. Over the past 4 decades, numerous research have been done on texture analysis, see e.g. [87, 75, 114, 208, 232, 191, 159]. However, most of these approaches have difficulties in representing the geometrical aspects of textures, such as sharp transitions and elongated contours. In order to overcome this difficulty, alternative approaches have been developed to provide more efficient representations of structured textures, such as grouplets [159], a wavelet-like approach, and structure-texture decomposition [10, 8], a method to decompose images into structures and textures and then dealing with each part separately.
1. Introduction

Other solutions to the problem are the geometric methods led by the texton theory [100], which states that only the first-order statistics of a few local "conspicuous" features, called textons, e.g. line ends and blobs, are significant for texture discrimination. However, to efficiently extract the textons is far from trivial. Some algorithms have been suggested to explicitly detect repeated structures in textures [111, 112, 113] and demonstrated good performances on a set of highly structured images.

In the mathematical morphology school, granulometry [131, 187] has been proposed to characterize textures by relying on the size distributions of some predefined structure elements. The idea is close to that of texton theory, with the textons as predefined small circles, rectangles and diamonds and the repetitive nature conveyed by distributions.

The part of the thesis on texture analysis tries to fill the gap between granulometry and texton theory by introducing geometric shapes, which are adaptive to the textures, as textons.

1.2.2 Automatic abstraction of photographs

Non-photorealistic rendering (NPR) [83, 215, 125, 92] is an area of computer graphics which concentrates on producing or mimicking a wide variety of expressive artistic styles, usually starting from natural images. A constant trend in the field of NPR has been to use image analysis tools to guide the rendering stage [83, 48, 16, 15, 154, 195]. The main difficulty of such a task is to account for image structures at different scales and to reflect their interactions.

![Figure 1.3: Some examples of abstract paintings. From left to right: the Composition VIII and Several Circles by Wassily Kandinsky, and the Composition A by Piet Mondrian.](image)

The part of the thesis on NPR is inspired by abstract paintings, particularly geometric abstract paintings, such as the works displayed in Figure 1.3. Observe that these works are often composed of a small set of shapes, such as rectangles, circles or other simple forms. Analogically to the geometric abstract painting, we will present a way to automatically create abstract images. This work is closely related to the experiments by Alvarez-Gousseau-Morel [4], which have systematically studied the range of abstract images that can be created by combining basic principles such as occlusion, addition, exclusion or inclusion with simple geometrical shapes. In contrast with this work, we investigate the creation of abstract images by manipulating all shapes from a given natural image or by replacing them with some geometric shapes. The structure of the created image is
1.2. Topics of the Thesis

therefore inherited from the structure of the natural image and not created by prescribed
stochastic placement and interaction rules.

Figure 1.4: Illustrations of the junction detection. **Left**: location and scale selection
using Harris-Laplace. Locations and scales are selected as local maxima of a differential
operator in scale and space. **Right**: detected junctions by Harris detector on a Gaussian
noise image. The color (from blue to red) of each junction indicates its significance (from
low to high). No junction should be detected in this image.

1.2.3 Detecting meaningful junctions in images

Like textures, junctions, i.e. points in images where edges join or intersect, are also fre-
quently used as basic features in computer vision and image understanding. The detection
of junctions is two-fold: finding the intersections of edges as well as identifying the scale of
the structures. The Harris detector [88] is one of the popular methods for locating junc-
tions, but often results in many false detections, see Figure 1.4b, since there is no adaptiv-
criterion for thresholding the junction significance. This motivates the definition of *meaningful junction detection* in images. We first model junctions as visual events consisting
of the non-accidental meeting of several edges at some position. More precisely, we make
use of the *a contrario* methodology [58] and pay a special attention to the statistics of
the image and to some maximality principles. In this part, the use of gradient orientation
is preferred to the direct use of level lines for robustness reasons, in particular to to be
robust to illumination changes. Observe that the relative sensitivity of level lines to illu-
mination is not an issue when analyzing textures because in this case we pay attention to
the distribution of structures.

For scale selection, most of the classical approaches rely on the linear scale space [120,
185, 126]. As mentioned in Section 1.1.1, the shortcomings of this kind of approaches is
that they lose the precision both in location and in scale, see Figure 1.4a for a graphic
illustration. Alternatively, we will show that by checking the local directions of level lines
at different sizes of patches and comparing their meaningfulness we can detect junctions with scale invariance. Besides, we are able to identify the junction branches (both their number and their locations) and the junction location with the same principle.

1.3 Main Contributions of the Thesis

The main contributions of this thesis are summarized and listed as follows:

(1) In Chapter 2, a shape-based texture indexing scheme is suggested by relying on the complete set of level lines of the image, the so-called topographic map [37]. The shapes, i.e. the interiors of the connected components of level lines, are taken as textons to perform the proposed texture analysis. The resulted analysis scheme enables a direct and adaptive handling of structured aspects of textures. Based on these textons, we then apply the scheme to the problem of invariant texture recognition, by exhibiting a set of simple statistics from the shapes. Since shapes are individually normalized, the proposed texture descriptors are invariant to local geometrical deformations and local contrast changes. It enables us to recognize non-planar or even non-rigid textures. In Chapter 3, various experiments of texture recognition on several challenging databases demonstrate the efficiency of the proposed method. Although numerous investigations have been made to the analysis and synthesis of textures in the mathematical morphology school [44, 6, 12, 196, 49], few work on invariances [84, 205] has been reported. Our work demonstrates that granulometry-like analysis enables a direct and very efficient way to achieve a high level of invariance, both to geometric and radiometric changes. Observe that this work somehow fills the gap between granulometry and texton theory, by taking shapes as textons.

(2) Chapter 4 is devoted to three applications of the shape-based texture indexing scheme detailed in Chapter 2. First, the proposed texture descriptors have been demonstrated to be valid for segmenting texture images, being combined with a recent active contour model [26]. In comparison with classical approaches which segment homogeneous textures, our method enables the partition of texture images with geometric deformations. Second, the indexing scheme is applied to the retrieval and classification of high-resolution satellite images, where both textures and geometric parts are important. The last contribution of this chapter is to introduce an adaptive image retrieval method based on statistical manifolds. It first shows that the image recognition performance can be improved significantly through using the geodesic distances on statistical manifolds and then introduces a method for setting the matching threshold for image retrieval automatically.

(3) In Chapter 5, a general framework for the structured abstraction of images is presented. We first show that the topographic map enables the creation of geometric abstractions from digital photographs. The results evoke the abstraction painting schools of the beginning of the 20th century. We show the versatility of the approach through numerous experiments. As an extension of this framework, we then suggest the scale-ratio filter, a filter which suppresses image structures according to the scale
ratio between shapes and their parents along the topographic map. The filter show interesting results for both denoising and image simplification.

(4) Chapter 6 addresses the problem of finding meaningful junctions in natural images by relying on a perception grouping principle, called *a contrario* detection theory. Junctions are detected as peculiar local configurations of the gradient orientations that are unlikely to be due to chance. It assigns a perceptual significance to each junction and uses it to control the detection and estimate the location, scale, as well as the type of junction. In contrast with previous works dealing with junction detections, the proposed method detects junctions with scale and contrast invariance. Thanks to the *a contrario* model, only one parameter is involved to control the detection. Comparison experiments in Chapter 7 demonstrate that the proposed method outperforms classical approaches on junction detection, especially in terms of precision and invariance.
1.4 Detailed Outline of the Thesis

The thesis is organized in three parts. This section provides a detailed outline of each part. Note that the content in this section is redundant with the previous sections and will be largely used for the introduction of corresponding chapters, and it is only aimed at serving as a reading guide.

Part I: Shape-based Texture Analysis

The first part of this manuscript concentrates on the problem of shape-based invariant texture analysis and its applications.

- Chapter 2 (A generic shape-based texture analysis method) This chapter introduces the theoretical part of the shape-based invariant texture analysis method. Section 2.1 provides a brief view on general texture analysis methods and presents the motivation and the state-of-the-art on the problem of invariant texture analysis. Section 2.2 briefly recalls the definition and properties of the topographic map, which is the basic tool used in the following parts and briefly comments on its relationships with textures. Section 2.3 details the invariant texture descriptors, which are computed both from the marginals of invariant shape moments and from shape dependencies in the topographic map.

- Chapter 3 (Experimental analysis of the indexing scheme) The aim of this chapter is to experimentally analyze the shape-based invariant texture analysis scheme developed in Chapter 2. First, it is explained in Section 3.1 how to compare texture images using the descriptors introduced in Chapter 2. Then, in Section 3.2, the performances of the resulting comparison scheme are investigated by confronting it with state-of-the-art texture descriptors. In order to meet the standards of the current literature in texture indexing, these experiments are performed on three different databases, namely the classical Brodatz database, the UIUC database [113] and the more recent UMD database [226]. Next, Section 3.3 is devoted to a discussion on invariance to resolution changes as well as to some comments on the trade-off between invariance and discriminative power. Eventually, Section 3.4 investigates the scaling properties of the proposed scheme (how it behaves when the size of the database is increased) on the union of the three aforementioned databases.

- Chapter 4 (Three applications of the theory) This Chapter proposes three applications of the shape-based texture indexing scheme introduced in Chapter 2. Section 4.1 studies the use of the scheme for the segmentation of images according to texture cues. Section 4.2 investigates the structured characterization of high-resolution satellite images. Section 4.3 first suggests an unsupervised learning algorithm to improve the performance of texture analysis and then proposes an adaptive image retrieval framework based on statistical manifolds.

Part II: Structured Image Manipulation

This part of the thesis investigates the use of topographic map in the context of NPR.
1.4. Detailed Outline of the Thesis

- **Chapter 5 (A generic structured image manipulation framework)** In this chapter, we show that the topographic map enables the creation of geometrical abstractions from digital images. In Section 5.1, we describe the motivation of this work and briefly review related works on this topic. Section 5.2 provides some further comments on topographic maps and details its adaptation for color images. Section 5.3 first presents the general framework for the geometrical manipulation of images and then lists a set of algorithms to achieve different abstract effects. In Section 5.4, the proposed framework is compared with *Arty Shapes* [195]. Finally Section 5.5 is devoted to the *scale-ratio filter*.

**Part III: Meaningful Junctions Detection**

This chapter addresses the problem of finding junctions in natural images by relying on a perception grouping principle, called *a contrario* detection theory [58].

- **Chapter 6 (Definition and Detection of Junctions)** In this chapter, Section 6.1 introduces the motivations and related works on junction detection. Section 6.2 presents how to make contrast invariant junction proposals based on local image structures. Section 6.3 proposes the *a contrario* junction model and Section 6.4 give some implementation details.

- **Chapter 7 (Detecting Junctions in Natural Images)** This chapter concentrates on the experimental analysis of the junction detection algorithm proposed in Chapter 6. Section 7.1 illustrates the control of the number of false detections in a random noise image. The invariance properties of the method are then investigated, with respect to scale changes in Section 7.2 and with respect to contrast changes in Section 7.3. In Section 7.4, the proposed method is finally compared with some state-of-the-art junction detection approaches.
1.5 Publications Related to the Thesis

The works in this manuscript have lead to the following publications:

- The results in Chapter 2 and Chapter 3 have been published in the *International Journal of Computer Vision (IJCV)* [223]:

  (1) Gui-Song Xia, Julie Delon and Yann Gousseau,
  *Shape-based invariant texture indexing*,

  and a short version of which was presented at ICPR [222]:

  (2) Gui-Song Xia, Julie Delon and Yann Gousseau,
  *Locally invariant texture analysis from topographical map*,
  The 19th International Conference on Pattern Recognition (ICPR), 2008.

- The results on satellite images in Chapter 4 have been presented on the *Symposium: 100 Years ISPRS - Advancing Remote Sensing Science* [224]:

  (3) Gui-Song Xia, Wen Yang, Julie Delon, Yann Gousseau, Hong Sun and Henri Maître,
  *Structural high-resolution satellite image indexing*,

  and an extension of the texture segmentation has been published by BMVC 2011:

  (4) Gui-Song Xia and Fei Yuan,
  *Texture segmentation by grouping ellipse ensembles via active contours*,

- The work on adaptive image retrieval in Chapter 4 is under preparation for a pattern recognition journal:

  (5) Gui-Song Xia, Julie Delon and Yann Gousseau,
  *Adaptive image retrieval based on statistical manifolds*,
  Under preparation

- The result on NPR in Chapter 5 is under preparation for an image processing journal

  (6) Gui-Song Xia, Julie Delon and Yann Gousseau,
  *A general framework for structured image manipulations*,
  Under preparation

- The result on junction detection in Chapter 6 and Chapter 7 has been submitted to *International Journal of Computer Vision (IJCV)*:

  (7) Gui-Song Xia, Julie Delon and Yann Gousseau,
  *Accurate junction detection and characterization in natural images*,
1.6 Other publications during the Ph.D

During the Ph.D thesis, I have also collaborated with other researchers on satellite images understanding and activity recognition, which lead to following publications:

- collaborated researches on satellite image understanding:
  
  (8) Wen Yang, Dengxin Dai, Bill Triggs and Gui-Song Xia,  

  (9) Wen Yang, Dengxin Dai and Gui-Song Xia,  
  Semantic labeling of SAR images with CRFs on region adjacency graph,  

  (10) Wen Yang, Bill Triggs, Dengxin Dai and Gui-Song Xia,  
  \textit{Scene segmentation via low-dimensional semantic representation and CRFs},  

  (11) Chu He, Xin-Ping Deng, Gui-Song Xia, Wen Yang and Hong Sun,  
  \textit{Topographic gray-level multiscale analysis and its application to histogram modification},  

  (12) Wen Yang, D. Dai, B. Triggs, Gui-Song Xia and Chu He,  
  \textit{Fast semantic scene segmentation with CRFs},  

- collaborated researches on activity recognition:

  (13) Fei Yuan, Gui-Song Xia, Hichem Sahbi and Véronique Prinet,  
  \textit{Spatio-temporal interest points chain for activity recognition},  
  Asian Conference on Pattern Recognition (ACPR) : Beijing, China, 2011.

  (14) Fei Yuan, Gui-Song Xia, Hichem Sahbi, Véronique Prinet,  
  \textit{Spatio-temporal context for mid-level features for activity recognition},  
  Pattern Recognition, 2011 (under minor revision).
Part I

Shape-based Texture Analysis
Chapter 2

A Generic Shape-based Texture Analysis Method

2.1 Introduction

2.1.1 A brief view on texture analysis

Texture is a fundamental ingredient of the structure of natural images. It both plays an important role in human visual perception and offers crucial cues for solving a wide range of computer vision problems, such as image segmentation or scene analysis. Yet, the analysis of texture is a long standing and challenging problem in computer vision and image understanding.

Given some pictures, we can easily figure out whether they are textures or not and we are also able to classify them into different categories. However, as observed in [70], "texture is a stuff that is easy to recognize but difficult to define". Over the years, although a number of different texture definitions have been proposed by vision researchers [87, 203], there is no universally agreed formal definition. A common view on texture representation is that a texture is an ensemble of elementary patterns organized by some rules of repetition. For instance, Tamura et al. took textures as "macroscopic regions" whose structures are "simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule" [198], Haralick described image texture by "the number and types of its (tonal) primitives and the spatial organization or layout of its (tonal) primitives" [86] and recently Meyer coined texture as a subtle balance between repetition and innovation [137]. Starting from this agreement, texture analysis often consists in exploring texture elements and then revealing the compositional rules or repetitive nature of these basic elements. This chapter proposes a formal definition of this view.

Over the course of the past 40 years, numerous studies have been done on texture analysis and it is a challenging task to summarize them all in several clear categories. However, we can find three main paradigms among them: the pixel-statistics paradigm, filter-bank paradigm and texton theory paradigm.
Pixel-statistics Paradigm The pixel-statistics paradigm of texture analysis mostly originated from the Julesz Conjecture [99]. It first suggests that humans cannot distinguish textures with identical second-order statistics. Although the conjecture was then proven to be false by Julesz himself years later [101], the idea that image textures could be modeled by relying on low-order statistics of pixels has inspired a great number of studies on texture analysis, which characterize image textures with a set of sufficient statistics.

Among others, co-occurrence matrix [87, 54] is a widely used approach to compute pixel-based statistics. The preliminary use of co-occurrence matrices involved in statistics of pairwise pixel relationships [87, 86] in several predefined fashions. Local Binary Pattern (LBP) [160, 153] extended the concept of co-occurrence by developing a framework for studying the statistics of co-occurrence binary patterns. In a different manner, the use of Markov random field (MRF) models for texture modeling attempted to handle the repetition nature of textures by relying on probabilistic models of a small set of pixels [75, 45].

Filter-bank Paradigm The repetitive nature of textures has oriented some of the very early research on automatic texture discrimination toward frequency or autocorrelation analysis, see e.g. [102] and [42]. Next, in order to deal with local transitions as well as with the “innovation” part of textures, one has favored localized frequency analysis. The filter-bank paradigm is motivated by the research in understanding brain’s visual cortex, that human visual system performs local spatial frequency analysis on retinal images which could be simulated by a computational model using a filter bank, e.g. Gabor filters [110, 52].

Significant progress of such method on texture was made during the 1990s [114, 208, 232, 191], relying on statistical representation of filter-bank responses. In such methods, the distributions, either marginal or joint ones, of filter responses are learnt from training images and represented by clusters or histograms [114, 208]. Finally, texture analysis and synthesis are based on this representation. It is noteworthy that Zhu et al. [232] have proposed a theory, named FRAME, to unify the filter bank and random fields for texture modeling. It can efficiently handle the analysis and synthesis of random textures.

One limitation filter-bank approaches, however, lies in their difficulty in efficiently representing the geometrical aspects of textures, such as sharp transitions and elongated contours. In order to overcome this difficulty, several alternative wavelet-like approaches, such as grouplets, have been proposed to enable more efficient representations of structured textures, see e.g. [159].

Texton-based Paradigm This paradigm is actually inspired by the texton theory of Julesz [100], which states that textons are "the putative units of pre-attentive human texture perception", related to texture’s local features, such as edges, line ends, blobs, etc. Julesz found that the first-order statistics of "a few local conspicuous features" are significant for texture discrimination. The texton theory in fact led to a structural approach to texture description, which first extracts texture primitives as local features and then investigates their organization.

Granulometry [131, 187] can be viewed as a classical example of structured texture analysis. It characterizes textures by relying on responses to morphological filtering with structuring elements of increasing sizes. Instead of using predefined preliminary elements,
2.1. Introduction

![Figure 2.1: The defined geometric texture elements and some examples of the representation of textures by the method of Lafarge [111]. (taken from [111])](image1)

![Figure 2.2: Texton dictionaries by small patches and interest points. (a) Original texture image. (b) Top 20 textons found by clustering all 13 x 13 patches of the image. (c) A sparse set of regions found by the Laplacian detector. Each region is normalized to yield a 13 x 13 patch. (d) Textons obtained by clustering the normalized patches. (taken from [112])](image2)

A more explicit way is to detect such texture elements beforehand. For instance, Zhu et al. [233] detect textons by using a number of image bases with deformable spatial configurations, which are learned from static texture images. Lafarge et al. [111] first define a set of geometric objects, e.g. segment, line, circle, band etc., as texture elements and then detect those elements and study their organization through Markov Point Process (MPP). The defined geometric textons and some examples of the representation of textures are shown in Figure 2.1. As observed in [233], a more general manner is to extract small image patches from texture, cluster them into textons and finally investigate the statistics of such
2. A Generic Shape-based Texture Analysis Method

textons, see Figure 2.2 (a) and (b) for an instance. Lazebnik et al. [112, 113] extended this idea by removing the redundancy between patch-based textons through utilizing interest regions in images, as illustrated in Figure 2.2 (c) and (d).

Compared with the pixel-statistics paradigm and filter-bank paradigm, the texton-based one can more easily handle structured parts in texture, such as edges and bars, which emerge in high-resolution image textures. However, the computation or detection of such textons is not trivial. It is also worth noticing that modeling the interactions between textons may involve heavy computation.

2.1.2 Motivations for shape-based invariant texture analysis

Shape-based Texture Analysis As mentioned before, the Mathematical Morphology school has long ago [82, 187] proposed a non-linear multi-scale analysis tool for texture, the so-called granulometry. Granulometries are obtained from an image by applying elementary morphological operations with structuring elements of increasing sizes. Because such basic morphological operations operate on the level sets of images, the resulting analysis enables a direct handling of edges and shapes contained in textures. Actually, even better accounting of these structured parts may be achieved by directly using the level sets of textures, as will be done in this chapter. Indeed, this would enable to describe directly what may be thought about as textons, in particular using geometrical tools. The first motivation of this work is therefore to extend granulometry-type texture analysis in order to keep the ability to characterize structured textures while achieving better precision. This is closely related to the theory of connected filters in mathematical morphology, already shown to yield better classification results for textures than plain granulometry, see [205]. As we shall see, the analysis proposed in this chapter will rely on a morphological and multi-scale decomposition of images, the topographic map as introduced by Caselles et al. [37].

Invariant Texture Recognition A challenging issue when analyzing texture is that texture surfaces are usually perceived under unknown viewing conditions. Except when dealing with a controlled image acquisition protocol, for instance in specific industrial applications, texture analysis methods should comply with some invariance requirements. The most basic ones are translation, scale and orientation invariances. For instance, such invariances are needed to identify the three textures in Figure 2.3. It is also desirable

![Figure 2.3: Three samples of the same texture class from the UIUC database [113]. There are scale and orientation changes between the samples.](image-url)
to achieve invariance to some contrast changes, in order to deal with variable lighting conditions. Next, the requirement of invariance with respect to viewpoint changes for flat texture yields analysis that are invariant with respect to affine or projective transforms. Moreover, textures can live on non-flat surfaces, as it is the case for bark on a tree or for folded textiles. Such an example is shown in Figure 2.4, where three different samples of the same texture class (plaid) from the UIUC database [113] are displayed. Several recent approaches to the analysis of such textures rely on the extraction of local features that are individually invariant to some geometric transforms, such as similarity or affine transforms [112, 135]. In contrast with previous works dealing with invariant 3D texture analysis, such locally invariant methods do not need any learning of the deformations [208, 114] or explicit modeling [216] of the 3D surfaces.

![Figure 2.4: Three samples of the same texture class from the UIUC database [113]. This texture lies on non-rigid surfaces implying complex deformations between the samples.](image)

The second motivation for this work is to show that such invariances can be achieved by working directly on the morphological textons mentioned in the previous paragraph. For this, the basic idea is to encode each of them using classical invariant shape descriptors that have been well studied since the 70s [96]. Let us mention at this point that few works have explored such a path, with the notable exception of [205], where rotation-invariant texture analysis is proposed, and [84], where similar ideas as ours are briefly explored to perform scale-invariant texture classification. We will see in what follows how the basic strategy described above in fact enables invariances to rather extreme geometric and radiometric deformations. Then, Chapter 3 will show that the proposed indexing scheme enables retrieval and classification of textures that equal or outperform the existing state of the art on locally invariant approaches on several databases.

### 2.1.3 Invariant texture analysis: previous and related work

This section briefly summarizes different directions that have been explored for the invariant analysis of texture images. Texture analysis has been a very active research field over the last four decades, and an exhaustive study of this field is beyond the scope of this thesis. Some surveys and comparative studies of existing methods can be found in [86, 203, 166, 164, 228], the last one being devoted to invariant texture analysis. In what follows, we first focus on classical approaches and the type of global invariances they allow. By global invariances, we mean invariances to global transforms of the image. We then summarize recent approaches to the analysis of texture that are invariant under
local transforms of images. We focus on methods that are invariant \textit{by design} and do not include in this short discussion methods that are invariant as the result of a learning process [208, 114] or an explicit modeling of 3D textures surfaces [216].

The use of co-occurrence matrices [87, 54] is still a popular approach, relying on non-parametric statistics at the pixel level. It is also worth noticing that this path along non-parametric statistics has been very fruitful for the purpose of texture synthesis [63]. Rotation invariance can be achieved for such indexing methods by using polar coordinate systems, as detailed in [53]. In a related direction, Pietikäinen \textit{et al.} [160, 153] propose a rotation invariant local binary pattern (joint distribution of gray values on circular local neighborhoods) to describe texture images. Still at the pixel level, Kashyap and Khotanzad [105] developed rotation invariant autoregressive models. Cohen \textit{et al.} [45], among others, have introduced rotation invariant Gaussian Markov random fields to model textures. However, the design of scale invariant Markov random field rapidly implies very involved computations, see e.g. [75]. Of course, pixel statistics can be averaged over different neighborhoods and make use of multi-resolution schemes, but these statistics are certainly not the easiest way to achieve scale or affine invariant analysis of textures.

A second popular and efficient way to analyze textures relies on filtering. Many works have focused on different filter-bank (a set of orientation and spatial-frequency selective linear filters) families, different sub-band decompositions, and on optimization of the filters for texture feature separation, see e.g. [191, 164, 182, 72]. Many of these approaches enable translation invariance (by using over-complete representations), rotation and scale invariance, by using effective filter designs, see e.g. [43, 62, 162, 135]. Some contrast invariance can also be achieved by normalizing responses to filters.

As already mentioned, an alternative approach to the analysis of textures has been proposed by the mathematical morphology school in the framework of granulometry. The idea is to characterize an image by the way it evolves under morphological operations such as opening or closing when the size of the structuring elements is increased [187, 131]. These ideas have been successfully applied to the classification of textures, see e.g. [44, 6], as well as the related approach [12], making use of stochastic geometry. Several works rely on the theory of connected operators [180] to compute granulometry without the need for structuring elements, see [117, 66], thus potentially enabling greater geometrical invariances. However, as mentioned before, there are few works showing the benefit of the geometrical nature of morphological operators to achieve similarity or affine invariant texture classification, with the notable exception of [205], where a shape-size pattern spectra is proposed as a way to classify images. Their shapes are extracted by Max- and Min-tree [181], which is similar to us but with more redundancy [144]. Images are then characterized by using the moments of joint 2-D shape-size pattern spectra. In particular, it is shown that this spectra enables rotation-invariant classification of texture images. Similar ideas as ours are briefly explored in [84], where it is proposed to globally use the Earth Mover’s Distance between topographic maps to perform scale invariant texture classification. To the best of our knowledge, no work has proposed the use of morphological attributes to achieve viewpoint invariant description of textures. Concerning radiometric invariant analysis of texture, the benefit of using contrast invariant morphological operators to recognize texture under various illumination conditions has not yet been demonstrated. Authors of [85] have developed an illumination invariant morphological scheme to index textures,
but they achieve invariance thanks to histogram modification techniques and not by using
the contrast invariant properties of morphological analysis.

The strength of applying fractal theory to texture analysis lies in the multi-resolution
nature of texture, which is the basis of fractals. Early work applying fractal geometry
to texture analysis was done by Peleg et al [157], where a fractal signature, the slope
of the area of the gray-level surface at varying resolutions, was used to describe texture.
Later, Espinal et al [65] used wavelet-based fractal dimension for texture description. Such
approaches have been shown to enable globally invariant texture analysis. Recently, Xu et
al [225] proposed the use of the multi-fractal spectrum vector to describe textures while
achieving globally invariance under bi-Lipschitz transforms, a general class of transforms
which includes perspective transforms and smooth texture surface deformations. Herwig
et al. [214] proposed a 2D wavelet Leader based multifractal formalism for invariant texture
classification.

Recently, several works proposed to use individually normalized local features in order
to represent textures while being locally invariant to geometric or radiometric transforms,
see [113, 229, 206, 135]. In [113] and [229], a set of interest local affine regions are selected
to build a sparse representation of textures relying on affine invariant descriptors. Textures
are represented thanks to bag-of-features, a method that has been proved very efficient to
recognize object categories, see e.g. [116]. In [206], textures are characterized statistically
by the full joint PDF of their local fractal dimension and local fractal length, and this
approach is shown to be discriminative and affine invariant. Very recently, Mellor et
al. [135] have shown that similar local invariances can be obtained using a filter bank
approach. These authors develop a new family of filters, enabling a texture analysis that
is locally invariant to contrast changes and to similarities.

2.1.4 Contributions

As explained earlier in the introduction, the goal of this chapter is to introduce a new
method for texture analysis that in spirit is similar to morphological granulometries, while
allowing a high degree of geometrical and radiometric invariances. The approach we pro-
pose relies on the complete set of level lines of the image, the so-called topographic map,
introduced by Caselles et al. [37]. The shapes (that is, the interiors of the connected com-
ponents of level lines) are the basic elements or textons on which the proposed texture
analysis is performed.

Then, based on these textons, we exhibit a set of simple statistics, obtained using clas-
sical shape moments, invariant to either similarity or affine transformations. Therefore,
and because each shape is individually normalized, the proposed texture indexing is in-
variant to local geometrical transforms, allowing for the recognition of non-planar or even
non-rigid textures. Various experiments on texture classification and retrieval demonstrate
the efficiency of the proposed analysis method on several challenging databases.

Although, there are plenty of work on texture analysis and synthesis in the mathemati-
cal morphology school [44, 6, 12, 196, 49], few of them have focused on invariances [84, 205].
Our work demonstrates that granulometry-like analysis enables a direct and very efficient
way to achieve a high level of invariance, both to geometric and radiometric changes. Ob-
serve that this work somehow fills the gap between granulometry and texton theory, by
taking shapes as textons.

The rest of this chapter is organized as follows. First, in Section 2.2, we briefly recall the definition of topographic map. Next, Section 2.2.2 shows the representation of textures by topographic map. Section 2.3 develops locally invariant features based on the statistics of the shapes of textures.

2.2 Topographic Map

2.2.1 Definition and properties

In this section, we recall the definition of the topographic map and its main properties. The topographic map has been suggested as an efficient way to represent images by Caselles et al. [36, 37]. It is made of the level lines, defined as the connected components of the topological boundaries of the level sets of the image. As we shall see, this map inherits a tree structure from the nesting properties of level sets and is an elegant way to completely represent the geometric information of an image while remaining independent of the contrast.

The upper level sets of an image \( u : \Omega \rightarrow \mathbb{R} \) are defined as the sets

\[
\chi_\lambda(u) = \{ x \in \Omega ; u(x) \geq \lambda \},
\]

where \( \lambda \in \mathbb{R} \). We can define in the same way the lower level sets \( \chi^\lambda(u) \) of \( u \) by inverting the inequality. Remark that if \( \varphi \) is a strictly increasing contrast change, then

\[
\chi_{\varphi(\lambda)}(\varphi(u)) = \chi_\lambda(u),
\]

which means that the set of all upper level sets remains the same under increasing contrast changes. Moreover, the image is completely described by its upper level sets. Indeed, \( u \) can be reconstructed thanks to the following formula

\[
u(x) = \sup \{ \lambda \in \mathbb{R} ; x \in \chi_\lambda(u) \}.
\]

Of course, the same property holds for lower level sets. Now, observe that these upper (lower) level sets constitute a decreasing (increasing) family. Indeed, if \( \lambda \) is greater than \( \mu \), then \( \chi_\lambda(u) \) is included in \( \chi_\mu(u) \) (and conversely \( \chi^\lambda(u) \) contains \( \chi^\mu(u) \)). It follows that the connected components of upper level sets (respectively of the lower level sets) are naturally embedded in a tree structure. Several authors [180, 36, 91], have proposed to use these trees of connected components (one for the upper level sets, one for the lower level sets) as an efficient way to represent and manipulate images, thanks to their hierarchical structure and their robustness to local contrast changes. Observe that the maximally stable extremal regions (MSER) [133] detector in images also relies on connected component of level sets.

Now, the notion of level lines (topological boundaries of level sets) enables to merge both trees, which motivates further the use of the topographic map to represent images. Monasse and Guichard fully exploited this fact and, drawing on the notion of shape, developed an efficient way to compute this hierarchical representation of images [145], called Fast Level Set Transform (FLST). A shape is defined as a connected component of an upper or lower level set, whose holes have been filled. A hole of a set \( A \) in an image is defined as a
connected component of the complementary set of $A$ that does not intersect the border of the image. It is shown in [145] that the set of shapes of an image has a tree structure. Under some regularity assumption on the image, this tree is equivalent to the topographic map (that is the set of all level lines). For discrete images, the only technicality needed in order to define the shapes is that two different notions of connectivity should be adopted for level sets: 8-connectivity for upper level sets and 4-connectivity for lower sets (the opposite convention could of course be adopted). For more precision and results on the topographic map, we refer to the recent monograph [34]. For the experiments performed in this paper, we compute the topographic maps using the FLST code available in the free processing environment Megawave. For a recent alternative to the computation of the topographic map, see [194]. An example of the representation of a synthetic image by its topographic map is shown in Fig. 2.5.

![Diagram](http://www.cmla.ens-cachan.fr/Cmla/Megawave/)

**Figure 2.5:** Representation of an image by its topographic map (this example is taken from [144]). Left: an original digital image, with gray levels from 0 to 5; Right: representation of the image by its tree of shapes, where $(A, B, \ldots, I)$ denote the corresponding shapes.

The topographic map has a natural scale-space structure, where the notion of scale corresponds to the areas of the shapes [146]. This is of course a first motivation to investigate its use for texture analysis. Moreover, because it is made of the level lines of the image, the topographic map permits to study textures at several scales without geometric degradation when going from fine to coarse scales. This is actually a very strong property of this scale-space representation. Contrarily to approaches using the linear scale space or linear filtering, it allows a faithful account of the geometry at all scales. Figure 2.6 illustrates this ability. This figure shows a needlework texture, in which the smallest scales represent the fine net of the needlework, while the large scales capture the boundaries of the flowers that are represented.

Next, the topographic map is invariant to any increasing contrast change. In fact, it is even invariant to any local contrast change as defined in [38]. This property is of primary
interest to define texture analysis schemes that are robust to illumination changes. At this point, it is important to add that while individual lines may not be strictly invariant to illumination changes, *marginals* of the geometrical attributes of lines are, as will be demonstrated by the experimental section.

Last, the basic elements of the topographic map are shapes obtained from connected components of the level sets. Therefore, it provides a local representation of the image. As we shall see, this locality, combined with the fact that the topographic map is by nature a geometric representation of images, enables us to develop analysis schemes that are invariant to local geometrical distortions.

### 2.2.2 Topographic map and textures

We will see precisely in the experimental section that the set of level lines contains pertinent information about the structure of textures. This fact is suggested in the original paper on the topographic map of images [37], where it is stated that "no matter how complicated the patterns of the level lines may be, they reflect the structure of the texture". A first attempt at using the topographic map to classify texture images has been proposed in [84]. In the context of satellite imaging, scales computed from contrasted level lines have proven useful to discriminate between different textured areas [128]. The use of level lines in the context of texture synthesis has also been investigated in [79].

As already said, the idea to be detailed is to encode texture through shape invariant moments, and more precisely as we will see, through second order moments, for robustness reasons. The same choice is made in [143] for image registration and in [84] for texture
2.3. Invariant Texture Descriptors

The goal of this section is to define texture features that are both invariant to some geometric changes and discriminative enough. According to the shape-based texture indexing recognition. Before proceeding, we show an experiment suggesting that, indeed, level lines carry some interesting information about textures. In Figure 2.7 and Figure 2.8, we show examples of a representation of textures where each shape on the tree is approximated by an ellipse with the same second order moments\(^2\). These textures, including textures with elongated structures, appear to be well described by a tree of ellipses. Another important texture cue is of course the tree structure, \(i.e.\) the relationships between shapes. The scheme of shape-based texture analysis to be detailed is summarized in Figure 2.9.

\[\text{Remark that after replacing each shape in the tree by an approximating ellipse, the inclusion relationships between shapes may be destroyed. The displayed images are obtained using the synthesis algorithm to be presented in Chapter 5, relying on sequential occlusions of shapes.}\]
2. A Generic Shape-based Texture Analysis Method

![Example textures and their shape-based representations](image)

Figure 2.8: Examples of textures represented by second-order shapes, *i.e.* ellipses. (a): three original textures taken from the UIUC dataset; (b): texture representation by keeping the hierarchy of the tree but replacing each shape by an ellipse with the same second-order statistics. Observe that the elongated structures are well represented.

In this section, we first give a short reminder on the invariant moments that can be extracted from the inertia matrix of a shape, focusing on invariances to similarity and affine transforms. More information on this classical subject can be found e.g. in [96, 68, 118, ...]
2.3. **Invariant Texture Descriptors**

Figure 2.9: The shape-based texture indexing scheme.

Then, we show how this moments can be applied to shapes of the topographic map in order to perform locally invariant texture analysis.

### 2.3.1.1 Invariant moments reminder

For $p, q$ integer values, the two-dimensional $(p+q)$th order central moment $\mu_{pq}(s)$ of a shape $s \subset \mathbb{R}^2$ is defined as

$$
\mu_{pq}(s) = \int \int_s (x - \bar{x})^p (y - \bar{y})^q \, dx \, dy,
$$

where $(\bar{x}, \bar{y})$ is the center of mass of the shape, i.e.

$$
\bar{x} = \frac{1}{\mu_{00}(s)} \int \int_s x \, dx \, dy, \quad \text{and} \quad \bar{y} = \frac{1}{\mu_{00}(s)} \int \int_s y \, dx \, dy.
$$

For the sake of simplicity, we will omit the variable $s$ in the following and write $\mu_{pq}$ instead of $\mu_{pq}(s)$. Note that $\mu_{00}$ is the area of the shape and that all central moments $\mu_{pq}$ are invariant to translations.
In order to achieve invariance to scale changes, it is well known and easily shown that moments have to be normalized in the following way

$$\eta_{pq} = \mu_{pq}/\mu_{00}^{(p+q+2)/2}. \quad (2.3.3)$$

As a consequence, any function of the normalized moments $\eta_{pq}$ is invariant to both scale changes and translations of the shape $s$. Now, the sensitivity to noise of these moments quickly increases as their order increases. We observed experimentally that moments of order bigger than two are not robust enough to faithfully account for texture characteristics, and we therefore limit the analysis to moments of order smaller than 2. Since $\eta_{00} = 1$ and $\eta_{01} = \eta_{10} = 0$, invariant features are all obtained from the normalized inertia matrix

$$C = \begin{pmatrix} \eta_{20} & \eta_{11} \\ \eta_{11} & \eta_{02} \end{pmatrix}. \quad (2.3.4)$$

In order to achieve rotation invariance, only two features remain, namely $\lambda_1$ and $\lambda_2$, the two eigenvalues of $C$, with $\lambda_1 \geq \lambda_2$. Observe that using these values boils down to fit the shape an ellipse with semi-major axis $2\sqrt{\lambda_1}$ and semi-minor axis $2\sqrt{\lambda_2}$. Note also that from the seven similarity invariants proposed in the seminal work by Hu [96], the only ones of order two are $\lambda_1 + \lambda_2$ and $(\lambda_1 - \lambda_2)^2$. Now, any function of $\lambda_1$ and $\lambda_2$ would also be invariant to similarity. We chose to use

$$\epsilon = \lambda_2/\lambda_1, \quad (2.3.5)$$

and

$$\kappa = \frac{1}{4\pi\sqrt{\lambda_1\lambda_2}}, \quad (2.3.6)$$

because these invariants have a clearer intuitive meaning and a simpler range than Hu’s moments. The first one lies between 0 and 1 and describes the elongation or the flatness of the shape. It can be shown that the second one also lies between 0 and 1. This invariant can be seen as a measure of the compactness of the shape, which reaches its maximum at ellipses. Indeed, $\kappa$ is a dimensionless ratio between the area of the shape (1 for a normalized shape) and the area of the best ellipse fitting the shape. Note that this invariant is more robust than a measure relying on the boundary of the shape, such as the isoperimetric ratio $\frac{4\pi a}{p^2}$ (where $p$ and $a$ are respectively the perimeter and the area of the shape), see Figure 2.10 for illustration. Next, observe that $\kappa$ (but not $\epsilon$) is further invariant to affine transforms. In fact, $\kappa^{-2}$ is the first affine invariant of Flusser et al., defined in [68].

### 2.3.1.2 Texture features from second order moments

As a first feature to represent textures, we simply compute the marginals over all shapes of the two features $\kappa$ and $\epsilon$. More precisely, for each of these two features, we compute a 1D-histogram by scanning all the shapes of the topographic map. The resulting 1D-histograms are invariant to any local contrast change, even decreasing ones. Now, it is well known that contrast inversion strongly affects the visual perception. For this reason, we restrict the invariance to any local increasing contrast change [38] by splitting each of the previous 1D-histograms in two histograms, one for shapes originating from upper level...
2.3. Invariant Texture Descriptors

Figure 2.10: Comparisons between the isoperimetric ratio \( c = \frac{4\pi a}{p^2} \) and the new defined compactness \( \kappa \): (a)-(c) are three different affine transformed versions of a texture (D101, taken from Brodatz [27]). (d) shows the histogram of \( c \) and (e) shows the histogram of \( \kappa \) on them.

sets (bright shapes) and one for shapes originating from lower level sets (dark shapes). The concatenations of the bright and dark histograms are called respectively elongation histogram (EH) and compactness histogram (CpH).

Observe that since moments are individually normalized for each shape, the resulting features are invariant to local geometrical changes (similarity for EH and affinity for CpH). More precisely, applying a different geometrical transform on each shape does not affect the overall marginals of \( \kappa \) and \( \epsilon \). In particular, this should allow to recognize texture that have undergone non-rigid transforms.

2.3.2 Dependencies in the topographic map

As explained in the previous section, requiring geometrical invariances and robustness restricts the number of possible invariant moments to two. In order to define new features from the topographic map without going into complex geometrical descriptors relying e.g. on the boundary of shapes, it is natural to take shape dependencies into account. Indeed, invariant moment marginals as defined in the previous section do not reflect the relative
positions or inclusions between shapes. Let us illustrate this point by a toy-example. Figure 2.11 shows two simple synthetic textures and their corresponding topographic maps. These two images share the same histograms EH and CpH, in spite of their structural differences.

![Toy example: two synthetic textures and their corresponding topographic maps. Both images have the same shape marginals but different tree structures, as shown in (c) and (d).](image)

We claim that the topographic map, because of its hierarchical structure, enables the extraction of shape dependency in an easy and intuitive way. In this work, we focus on children-parents relationships within the tree, although other relationships could be interesting.

**Definition (Ancestor family $\mathcal{N}^M$)** Let $s$ be a shape of the image. Let $s^m$ be the $m$-th cascaded ancestor of $s$, where $m$ is an integer. That is, $s^1$ is the parent shape of $s$, $s^2$ the parent shape of $s^1$, etc. For $M \geq 1$, the $M$th ancestor family of $s$ is defined as $\mathcal{N}^M = \{s^m, 1 \leq m \leq M\}$.

Now, it is quite simple to extract affine invariant information from these ancestor families. Recall that $\mu_{00}(s)$ is the area of the shape $s$. An affine transformation $AX + b$
2.3. Invariant Texture Descriptors

On $s$ changes $\mu_{00}(s)$ into $\text{det}(A)\mu_{00}(s)$. As a consequence, if we define for any shape $s$

$$\alpha(s) = \frac{\mu_{00}(s)}{(\mu_{00}(s'))_{s' \in \mathcal{N}^M}},$$

(2.3.7)

where $(\cdot)_{s' \in \mathcal{N}^M}$ is the mean operator on $\mathcal{N}^M$, then $\alpha$ is locally affine invariant, in the sense that for each shape $s$, $\alpha(s)$ is only sensitive to transformations applied to its $M$ direct ancestors. Remark also that $0 < \alpha < 1$. Again, the distribution of $\alpha$ is represented by a 1D-histogram, split into dark and bright shapes. The corresponding feature is called scale ratio histogram (SRH).

**Remark** Other features could be extracted from the ancestor family, built e.g. from elongation or compactness as defined in the previous section. However for the purpose of texture indexing, and in particular for the classification and retrieval tasks to be considered in the experimental section, we did not find them to be overly discriminative. These could however be useful for different tasks.

In what follows, we use two sets of texture features. The first one, called SI, is made of the features that are invariant to (local) similarity transforms, while the second one, called AI, is made of the (locally) affine invariant features. That is,

- $SI = \text{CpH} + \text{SRH} + \text{EH},$
- $AI = \text{CpH} + \text{SRH},$

where, as defined before, EH stands for elongation histogram, CpH for compactness histogram and SRH for scale ratio histogram. These are geometric features, in the sense that they are invariant to any (local) increasing contrast change. We believe that these descriptors illustrate the usefulness of the topographic map to analyze texture images, in particular allowing for relatively easy handling of invariances.

2.3.3 Contrast information

The previous geometric features are invariant to *any local increasing contrast change*, as defined in [37]. This is a very strong invariance and we are not aware of any texture analysis scheme having this property. Now, we observed that this invariance is too strong to efficiently recognize many texture classes. In this section, we define contrast features that are invariant to *local affine contrast changes*. This is coherent with the contrast invariances considered in recent works to which we will compare our results, such as [113, 135, 225].

We choose to compute intensity histograms after local normalization by mean and variance on a neighborhood. Such photometric normalization approaches are relatively standard and have been used in local descriptors, see [152, 184]. Schaffalitzky *et. al* [184] enable their texture descriptors to be invariant to local affine illumination changes by normalizing the intensity of each point by the mean and standard deviation over a local adaptive neighborhood (a support region with detected adaptive scale). We follow a similar path, except that we rely on the topographic map to define local neighborhoods.

More precisely, at each pixel $x$, a normalized grey level value is computed as

$$\gamma(x) = \frac{u(x) - \text{mean}_{s(x)}(u)}{\sqrt{\text{var}_{s(x)}(u)}},$$

(2.3.8)
where $s(x)$ is the smallest shape of the topographic map containing $x$, mean$_{s(x)}(u)$ and var$_{s(x)}(u)$ are respectively the mean and the variance of $u$ over $s(x)$. This results in a contrast histogram (CtH), computed by scanning all pixels of $u$. Thanks to the adopted normalization, the resulting feature is invariant to local affine contrast changes, as the features in [113, 135, 225].

One particularity of the proposed normalization (2.3.8) is that the normalized value $\gamma(x)$ at $x$ will generally be negative for shapes coming from an upper level set, and positive for shapes coming from a lower level set (this property is not systematic but very often satisfied on natural images).

Observe that this last feature, CtH, is not invariant to local similarity (or affine) transforms. Indeed, contrast histograms are computed on a pixel by pixel basis which breaks the geometrical invariances we add preserved so far. Now, we observed that this feature is very robust to geometrical distortions of the textures, even in some extreme cases, as will be demonstrated by the experimental section.

In the next chapter, we show how the invariant analysis schemes we just introduced can be efficiently used to compare texture images and analyze its performances on challenging texture databases.
Chapter 3

Experimental Analysis of the Indexing Scheme

This chapter analyzes the shape-based texture indexing scheme experimentally. First, it is explained in Section 3.1 how to compare texture images using the features introduced in the previous section. Then, in Section 3.2, the performances of the resulting comparison scheme are investigated by confronting it with state-of-the-art texture descriptors. More precisely, we follow the experimental protocols presented in [113] and reproduced in [135]. These protocols consist of retrieval and classification tasks. In order to meet the standards of the current literature in texture indexing, these experiments are performed on three different databases, namely the classical Brodatz database, the UIUC database [113] and the more recent UMD database [226]. On these three databases, the descriptors introduced in this chapter show similar or better results than the descriptors presented in [113, 135, 225]. For the sake of completeness, all the results of the retrieval experiments by the proposed method are available at the Internet address [220]. Next, Section 3.3 is devoted to a discussion on invariance to resolution changes (illustrated by experiments on our own high resolution texture database) as well as to some comments on the trade-off between invariance and discriminative power. Eventually, Section 3.4 investigates the scaling properties of the proposed scheme on the union of the three aforementioned databases.

For all experiments of this section, histograms EH, CpH and SRH are computed over 25 bins for bright shapes and 25 bins for dark shapes. Histogram CtH is computed over 50 bins. The value of $M$ used to compute SRH is set to $M = 3$.

3.1 Descriptors Comparison

Two texture samples $u$ and $v$ are compared through the distribution of features, simply by comparing the corresponding histograms. In this chapter, the histograms are compared through Jeffrey divergence, a symmetric modification of the Kullback-Leibler (K-L) divergence. For two discrete distributions $P = (p_1, \ldots, p_N)$ and $Q = (q_1, \ldots, q_N)$, the Jeffrey
divergence between $P$ and $Q$ is defined as
\[
D_J(P\|Q) = \sum_{i=1}^{N} \left( p_i \log \frac{p_i}{m_i} + q_i \log \frac{q_i}{m_i} \right)
\] (3.1.1)
where $m_i = \frac{p_i + q_i}{2}$.

In our tests, probabilistic measures of similarities such as Jeffrey divergence or $\chi^2$-divergence (used by [135]) yield better results than $L^p$-distances (e.g. Manhattan, $p = 1$, or Euclidean, $p = 2$). Using one-dimensional Earth mover’s distance [175] between histograms yields consequently poorer results, probably due to a larger variability in the relative weight of bins than in their positions.

Denote by $D^k_J(u,v)$ the Jeffrey divergence between the $k^{th}$ histograms of the descriptors of $u$ and $v$ (in this work $k \in \{1, \ldots, 3\}$ if we use the descriptor AI+CtH and $k \in \{1, \ldots, 4\}$ if we use SI+CtH). The final distance between $u$ and $v$ can be computed as a weighted sum of the distances $D^k_J(u,v)$,
\[
D(u, v) = \frac{\sum_{k=1}^{K} \omega_k D^k_J(u\|v)}{\sum_{k=1}^{K} \omega_k}
\] (3.1.2)
where $\omega_k$ is the weight assigned to the $k^{th}$ feature. For the sake of simplicity, in the following experiments the weights $\omega_k$ have been chosen as equal. These weights could have been adapted by learning their respective discriminative power on a training data set (see e.g. [229]).

### 3.2 Comparative Evaluations

#### 3.2.1 Experimental protocols

As explained before, in this chapter, we reproduce exactly the retrieval and classification experiments described in the papers of Lazebnik \textit{et al.} [113], Mellor \textit{et al.} [135] and Xu \textit{et al.} [226].

Recall that the approach of Lazebnik \textit{et al.} [113] relies on local descriptors. These descriptors are computed on a sparse set of affine invariant regions of interest. This kind of approach is popular in computer vision and known to be very efficient for object recognition. In the work of Lazebnik \textit{et al.} [113], the best results are obtained with the combination of two region detectors (Harris and Laplacian) and two local descriptors (spin images and RIFT descriptors). The corresponding texture description, which is denoted by $(H+L)(S+R)$, is locally invariant to affine transformations and locally robust to affine contrast changes. The approach of Mellor \textit{et al.} [135] relies on histograms of several invariant combinations of linear filters. This description is locally invariant to similarities and globally invariant to contrast changes. Finally, the method developed by Xu \textit{et al.} [226] is based on a multifractal description of textures. Their description is invariant under many viewpoint changes and non-rigid deformations, as well as local affine contrast changes.

In order to compare the performances of the descriptors we introduced with the best results provided by these papers, experiments are performed on three different databases: the Brodatz database, the UIUC database [113] and UMD database [226]. It is worth
noticing that the corresponding results should be taken cautiously and not directly compared with other retrieval or classification experiments which do not follow exactly the same experimental protocols.

The retrieval experiment consists in using one sample of the database as a query and retrieving the $N_r$ most similar samples. The average number of correctly retrieved samples (generally called recall) when the query spans the whole database is drawn as a function of $N_r$.

For the classification experiment, $N_t$ samples are extracted from each class and used as a training set. Each remaining sample in the database is then affected to the class which contains the nearest training sample. For each value $N_t$, an average classification rate is computed by using randomly selected training sets, in order to eliminate the dependence of the results on some particular sets.

### 3.2.2 Databases

The three different databases used for the comparison tasks are now briefly described.

- **Brodatz Dataset**: The Brodatz’s photo album [27] is a well known benchmark database used to evaluate texture recognition algorithms. Although it lacks some interclass variations, Lazebnik *et al.* [113] point out that this database is a challenging platform for testing the discriminative power of texture descriptors, thanks to its variety of scales and geometric patterns. This database contains 111 different texture images. Following the protocols of [113, 135], we divide each of these images into 9 non overlapping samples of resolution $215 \times 215$. As a result, the complete dataset is composed of 111 texture classes, each one being represented by 9 samples (all in all, 999 samples). Figure 3.1 illustrates 9 samples for 9 different textures in this database.

- **UIUC Database**: This texture database [113] contains 25 texture classes, each one being composed of 40 samples of size $640 \times 480$ (i.e. 1000 samples altogether). Inside each class, the samples are subject to drastic viewpoint changes, contrast changes or even non-rigid deformations. Figure 3.2 illustrates two samples per each texture.

- **UMD Database**: This database, introduced by Xu *et al.* [226] in order to test globally projective invariant features, is composed of 25 different textures classes, each one being represented by 40 samples (1000 samples altogether). These samples show strong viewpoint and scale changes, and significant contrast differences. A significant proportion of this database is made of textures consisting in the repetition of objects. The resolution of these images is $1280 \times 960$. Figure 3.3 illustrates two samples per each texture.

### 3.2.3 Performances on Brodatz

Figure 3.4 shows the retrieval and classification results obtained with the different indexing schemes on the Brodatz database.
In the retrieval experiment, shown on Figure 3.4 (a), the number of retrieved samples $N_r$ takes values from 8 to 50. Since each class contains 9 samples, a perfect indexing method should reach an average recall of 100% for $N_r = 8$. For this number of retrieved samples, the affine invariant descriptor $\text{AI+CtH}$ reaches 77.33% recall, while the similarity invariant descriptor $\text{SI+CtH}$ reaches 80.44%. These results slightly outperform those of Lazebnik’s affine invariant texture descriptor $(H+L)(R+S)$ (76.97% recall) and Mellor’s similarity invariant texture descriptors (77.65% recall). This trend remains valid when $N_r$ increases. It should be remarked that in order to obtain such results on Brodatz, Lazebnik et al. add a shape channel to their description, and lose thereby their invariance to local affine changes.

Following [113, 135], classification rates are estimated by averaging the results on ran-
3.2. Comparative Evaluations

When the number of training samples is 3 for each class, the average classification rate reaches 88.31% for AI+CtH and 90.66% for SI+CtH. For the same level of invariance, these results are equivalent to those reported by Lazebnik et al. (88.15%) and Mellor et al. (89.71% for their similarity invariant descriptor) with the same protocol.

Now, as observed in [135], some images of the original Brodatz database represent the same texture at different scales. Nevertheless, these images are considered as different textures by the experimental protocol, which penalizes invariant indexing schemes. In the same way, we should keep in mind that texture samples are created by cutting each texture of the Brodatz database into pieces. As a consequence, the resulting dataset lacks of viewpoint and scale changes. Consequently, a well chosen non-invariant indexing scheme
3. EXPERIMENTAL ANALYSIS OF THE INDEXING SCHEME

![Graphs showing retrieval and classification performances.](image)

Figure 3.4: Average retrieval (a) and classification (b) performances of different texture indexing schemes on the Brodatz dataset. The blue curves correspond to the performances of the descriptors SI+CtH and AI+CtH (recall that SI stands for similarity invariant local features, AI stands for affine invariant local features and CtH for locally affine invariant contrast histogram; all these features are described in Section 2.3.2), while the red curves show the performances of [113] and [135]. The performance of a non-invariant indexing scheme is also shown for the sake of completeness.

should naturally provide better results on this database. In order to check this statement and for the sake of completeness, we tried to add some non-invariant features to our invariant descriptors. For this purpose, we added to the SI+CtH descriptor the histogram of shapes areas and the histogram of shapes orientations (the orientation being defined as the direction of the principal eigenvector of the inertia matrix (2.3.4)). The corresponding retrieval and classification results are shown in Figures 3.4 (a) and (b). Observe that, as it could be expected, all the results are clearly improved by adding these features.

3.2.4 Comparisons on UIUC Database

Figures 3.5 (a) and (b) show the retrieval and classification results of the AI+CtH and SI+CtH descriptors on the UIUC database. For the same level of invariance, these results are better than those reported in [113] and [135].

Let us observe that we were able to obtain better results than those reported in Figure 3.5 by weighting the contribution of each shape in the descriptors by a power of its area. This trick allows to give more weight to large shapes than to small ones, and hence to take more into account the geometrical aspect of textures. Now, using this trick on the Brodatz database yields a decrease of performances. Therefore, and since we did not find an automatic way to tune this weighting, we chose not to develop this possibility in the present study.

It is also interesting to note that local similarity invariance is enough to correctly retrieve texture classes with strong viewpoint variations. This property is illustrated by
Figure 3.6, which shows the 39 first samples retrieved by SI+CtH when the query is the sample T15_01. This descriptor retrieves 38 samples of the class perfectly, despite the strong viewpoint changes between different samples. This is due both to the fact that three out of four features of SI+CtH are locally affine invariant, as well as to the fact that, as demonstrated by the experiments in Mellor et al., invariance to local similarity already enables a good handling of viewpoint changes. In fact, local similarity invariance yields better results than local affinity invariance on this database, as will be further discussed in Section 3.3.2.

Figure 3.5: Average retrieval (a) and classification (b) performances of different texture indexing schemes on the UIUC database. The blue curves correspond to the performances of the descriptors SI+CtH and AI+CtH (recall that SI stands for similarity invariant local features, AI stands for affine invariant local features and CtH for locally affine invariant contrast histogram; all these features are described in Section 2.3.2), while the red curves show the performances of [113] and [135].

Another specific retrieval result is shown on Figure 3.7 for the texture class T25 of the UIUC database. This class, which represents a plaid under different viewpoints, contains many distortions and non-rigid deformations. Nevertheless, the SI+CtH descriptor retrieves the samples of this class quite well (the average retrieval rate on the whole class reaches 65.26% for 39 retrieved samples). It is also worth noting that 6 out of the 8 errors (highlighted in red on Figure 3.7) come from the same class T03. The retrieval of these samples is false but consistent. An example of a texture yielding a bad retrieval rate is shown in Figure 3.8. The corresponding texture class exhibits both blur and a very strong variability.

For classification of the UIUC database, the descriptors AI+CtH and SI+CtH also show better performances than the methods of Lazebnik et al. [113] and Mellor et al. [135]. More precisely, the classification rate reached by AI+CtH is 66.56% and the one reached by SI+CtH is 70.69% when only one sample is used. These numbers should be compared to the rates of 62.15% and 67.10% achieved respectively in [113] and [135]. An interesting point is that the performances of our descriptors decrease on texture classes containing blur. The descriptors provided in the work of Lazebnik et al. [113] appear to be more
3.2.5 Comparisons on UMD Database

Using the same strategy as before, Figure 3.9 shows the retrieval and classification performances of the descriptors AI+ChT and SI+ChT, along with the results obtained by the method of Xu [225], as well as those obtained on this database with the method of Lazebnik [113] as reported in [225]. Observe that our indexing scheme is particularly well adapted to this database. Indeed, the curves of Figure 3.9 show that both SI+ChT and AI+ChT descriptors perform significantly better than other methods. This may be due to the fact that this representation relies on geometry and is thereby well adapted to highly resolved and structured textures. Figure 3.10 and Figure 3.11 show two specific retrieval
3.2. Comparative Evaluations

Figure 3.7: Retrieval result obtained on the texture class T25 of the UIUC database with the descriptor SI+CtH (Similarity Invariant + Contrast Histogram). The query image is in first position and the 39 most similar samples follow, ordered according to their matching scores. Retrieval errors are indicated in red. It is worth noticing that no learning is involved in these experiments. Retrieval results for all texture samples are available at the address [220].

results, an almost perfect result on a texture made of apple stacks, as well as a result on a texture made of bamboos, for which the retrieval rate is roughly the one we get on the whole database. The AI+CtH and SI+CtH descriptors deal quite well with large scale and illumination changes on the fruit texture. Concerning the bamboos texture, one observes that textures RT21 and RT20 (corn leaves) are visually very similar and relatively hard to discriminate.

Two conclusions arise after the comparison of the descriptors proposed in this chapter with the approaches of [113, 135, 225] on three different texture databases. First, both AI+CtH and SI+CtH are efficient for texture retrieval and classification. These descriptors show robust and consistent results on all three datasets, outperforming state of the art approaches. Second, similarity invariant descriptors always perform better than affine invariant descriptors on all three databases. This aspect will be discussed in the last part of the section.

It is also worth noting that the texture features that we introduced are relatively compact in size. More precisely, each texture sample is represented by 4 histograms of 50 bins each, i.e. 200 values altogether. This size is comparable to that of Xu’s descriptors [225],
Figure 3.8: A “bad” retrieval result obtained on the UIUC database with the descriptor SI+CtH (Similarity Invariant + Contrast Histogram). The query image is in first position and the 39 most similar samples follow, ordered according to their matching scores. This result corresponds to the class T19. The corresponding texture class exhibits both blur and a very strong variability. Observe also that one half of the retrieval errors (indicated in red) are from the texture class T17, which at some scales looks similar to the class T19.

which use 78 values for each texture sample. In comparison, Lazebnik et al. [113] use between 1200 and 4000 values for each sample (40 clusters of 32 or 100-dimensional descriptors), while Mellor et al. [135] represent each sample by a histogram of 4096 bins.

3.3 On Invariance and Discriminative Power

3.3.1 Invariance to resolution changes

It was shown in section 2.3 that descriptors SI and AI are invariant to, respectively, local similarities and local affine transforms. In particular, the invariance to scale changes was ensured by the use of normalized moments computed on the topographic map, which do not change under a perfect, theoretical scale change. However, in practice, scale changes on images often imply resolution changes. These changes can affect texture indexing methods, as investigated in [127]. Such transformations involve blur, which affects the topographic map of images. In order to check the robustness of the descriptors to such changes, we set up the following experiment. Starting from 20 highly resolved texture images (see Figure
3.3. On Invariance and Discriminative Power

Figure 3.9: Average retrieval (a) and classification (b) performances of different texture indexing schemes on UMD database [226]. The blue curves correspond to the performances of the descriptors SI+CtH and AI+CtH (recall that SI stands for similarity invariant local features, AI stands for affine invariant local features and CtH for locally affine invariant contrast histogram; all these features are described in Section 2.3.2), while the red curves show the performances of [226] on this database, as well as those using the method from [113] as reported by [226].

3.12), we build a database of 20 texture classes. In each class, the samples are generated by zooming each original texture image by a factor $t$, using bilinear interpolation. Here $t$ takes its values among $T$ as follows,

$$T = \{0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.6, 0.7, 0.8, 0.9\}.$$

As a consequence, the whole database contains 20 classes of 16 samples, i.e. 320 texture samples. The size of the original images being $3072 \times 2040$, the smallest image size is $384 \times 255$.

Figure 3.13 shows the histograms SRH, CpH, EH and CtH of the 15-th texture shown in Figure 3.12 (pebble beach) for different zoom factors $t$. Observe that the curves coincide as long as the zoom factor remains larger than 0.5 (blue curves). When this factor decreases, the histograms move away from the original ones (for $t = 1$) but remain close to it. Similar behaviors are observed on other textures. This proves empirically the robustness of these features to real resolution changes with a zoom factor larger than .125.

In order to test the discriminative power of these features within the framework of resolution changes, we perform a simple retrieval experiment on this multiresolution database. For each zoom value $t$ in $T$, and each texture class $i$, let $M_i^t$ be the subset of the class made of the images having a resolution larger than $t$. A sample of resolution $t$ and class $i$ being given, its retrieval rate is defined as the proportion of well retrieved samples in $M_i^t$. As usual, the final retrieval rate $r(t)$ is the mean of the retrieval rates over all samples of resolution $t$. Figure 3.14 shows the curves of $r(t)$ when $t$ varies from 0.125 to 1 and when
Figure 3.10: The retrieval results on class RT9 of UMD database, using the descriptor SI+CtH (Similarity Invariant + Contrast Histogram). The query image is in first position and the 39 most similar samples are ordered according to their matching scores. Both examples correspond to non-planar textures. Retrieval results for all texture samples are available at the address [220].

using different texture descriptors. Observe that up to a scale factor of 4, the retrieval results are perfect for SI+CtH.

### 3.3.2 Local invariance vs discriminative power

Following the experiments of section 3.2, the question of the level of invariance required to index a particular database arises naturally. We saw on Brodatz that removing invariance to scale and orientation greatly improved the results, which seems to be coherent with the fact that this database does not present many geometric distortions. Of course, the best level of invariance depends on the database. On UIUC and UMD databases, all descriptors invariant to local similarity changes show significantly better results than locally affine invariant descriptors, which confirms the results presented in [135] and [229]. Moreover, we observe that the advantage of similarity invariance on affine invariance remains true if we restrict ourselves to textures containing strong distortions. This can be surprising since these two databases contain classes with strong non-rigid deformations. We could theoretically expect that local affine invariance, or even local projective invariance would be needed to index such classes correctly (recall that UMD database, for instance, has been
3.3. On Invariance and Discriminative Power

Figure 3.11: The retrieval results on class RT21 of UMD database, using the descriptor SI+CtH (Similarity Invariant + Contrast Histogram). The query image is in first position and the 39 most similar samples are ordered according to their matching scores. Both examples correspond to non-planar textures. Observe that all errors for the class RT21 (bamboos) come from the class RT20 (corn leaves), which is visually quite similar to RT21. Again, no learning is involved in these experiments. Retrieval results for all texture samples are available at the address [220].

built on purpose to test projective invariant descriptors). The fact that features that are only invariant to local similarities show the best results despite these variations can only be explained by a better discriminative power. In other words, there is a natural trade-off between the level of invariance of a texture description and the discriminative power of this description.

Observe that the question of the best level of invariance needed for indexing is also addressed in [229, 207], where learning is used to estimate the optimal weights of the different descriptors.

These remarks also lead to question the need for further invariance in texture indexing. The previous observations suggest that achieving invariance to local similarities may be enough to account for viewpoint variations or non-rigid deformations. Furthermore, to the best of our knowledge, there exists no texture database in the literature on which complete local affine invariance is needed (in the sense that it yields better results than weaker invariances). Without such a database, it seems vain to try to develop features with more sophisticated invariances.
3. Experimental Analysis of the Indexing Scheme

Figure 3.12: Set of $3072 \times 2040$ texture images used to compute a multiresolution database. For each image, 15 samples are created by sub-sampling the original image with a zoom factor $t$ taking its value in the set $T$.

Figure 3.13: Histograms of the pebble beach texture, the 15th texture image shown in Fig. 3.12: (a) Scale-ratio Histogram (SRH), (b) Elongation Histogram (EH), (c) Compactness Histogram (CpH) and (d) Contrast Histogram (CtH), for different zoom factors.
3.4 Scaling Behavior of the Analysis Scheme

In this section, we briefly investigate how the proposed texture analysis scheme behaves when the numbers of texture classes and samples are increased. For this purpose, we simply build up a single database from the three texture databases considered so far (UIUC, UMD and Brodatz) therefore reaching 161 classes and 2999 samples.

![Diagram](image)

Figure 3.14: Average retrieval performances of the descriptors SI+CtH and AI+CtH (recall that SI stands for similarity invariant local features, AI stands for affine invariant local features and CtH for locally affine invariant contrast histogram; all these features are described in Section 2.3.2) on the multiresolution database presented in section 3.3.1.

Figure 3.15: In red, average retrieval (a) and classification (b) performances of Brodatz samples in the entire Brodatz+UIUC+UMD database, using SI+CtH (recall that SI stands for similarity invariant local features and CtH for locally affine invariant contrast histogram; all these features are described in Section 2.3.2). In blue, for comparison, average retrieval (a) and classification (b) performances of the same samples among the isolated Brodatz database.
We repeat the retrieval and classification experiments described in Section 3.2.1 for each sample of the whole Brodatz+UIUC+UMD database. Observe that in this configuration, classification and retrieval become noticeably more difficult: for example, for each query of the UIUC database, only 39 samples among 2999 belong to the same class, instead of 39 among 1000 in the experiments of Section 3.2.1. Figure 3.15 shows the retrieval and classification rates averaged over all Brodatz samples, while Figure 3.16 shows the retrieval and classification rates averaged respectively over all UIUC samples, or over all UMD samples. Observe that although the performances decrease when the numbers of samples and classes increase, the proposed texture analysis scheme scales very well. In particular, it performs better on this large combined database of 2999 samples than Lazebnik’s descriptors on the single UIUC database (1000 samples), and its performances are comparable to those of Xu’s descriptors on the single UMD database (1000 samples).

Figure 3.16: In red, average retrieval (a) and classification (b) performances of UIUC (resp. UMD) samples in the entire Brodatz+UIUC+UMD database, using SI+CtH (recall that SI stands for similarity invariant local features and CtH for locally affine invariant contrast histogram; all these features are described in Section 2.3.2). In blue, for comparison, average retrieval (a) and classification (b) performances of the same samples among the isolated UIUC (resp. UMD) database.
Chapter 4

Three Applications of the Theory

This chapter gives three applications of the shape-based texture indexing scheme introduced in the previous chapter. Section 4.1 studies the use of the scheme for the segmentation of images according to texture cues. Section 4.2 investigates the structured characterization of high-resolution satellite images. Section 4.3 first suggests an unsupervised learning algorithm to improve the performance of texture analysis and then proposes an adaptive image retrieval framework based on statistic manifold.

4.1 Texture Segmentation from Shape Ensembles

The objective of texture segmentation is to partition an image into several regions that are characterized by different texture attributes. In the literatures, many models have been proposed to solve that problem. Among those, the early work of Beck et al. [18] argued that textural segmentation occurs strongly on the basis of the distribution of simple properties, like brightness, color, size, the slopes of contours and lines of the elemental descriptors, of "texture elements". Julesz [100] proposed to use textons, a set of empirical texture features including elongated blobs, line ending or terminators, for computational texture modeling, and the segmentation of texture was consequently achieved by the differences of the textons. Recently, Todorovic et al. [201] suggested to extract explicit textural elements from hierarchical segmentation tree and applied them to texture segmentation.

Considering that the texture features introduced in Chapter 2 are efficient for classification tasks, it is natural to investigate their ability to segment texture images. Observe that the topographic map has a scale-space structure in which no regularization of the geometry is involved. This property makes it particularly interesting in the context of image segmentation.

4.1.1 Partitioning the tree of shapes is not enough for segmentation

This very brief section is devoted to the description of a try we gave at finding a segmentation method solely relying on the partition of the tree of shapes. We explain why we did not pursue this approach further. By relying on the tree of shapes, texture segmentation was investigated by grouping all these shapes, which provides some potentials for achieving
accurate segmentation boundaries due to the multi-scale texture representation without geometric degradation.

An intuitive idea is to label the shapes in the texture image, with the assumption similar to [47] stating that the segmentation boundaries are contained in the level lines of the image. We consider a texture image $I$, its associated tree of shapes $S$ and a set of texture samples $T = \{T_1, \ldots, T_M\}$, whose associated trees of shapes are also computed. Each shape $s \in S$ on the tree is identified to be one class of the sample set, as

$$L(s) = \arg \min_{T_i \in T} d(s, T_i).$$

(4.1.1)

d($s$, $T_i$) is the dissimilarity between shape $s$ and sample $T_i$, computed as follows

$$d(s, T_i) = \sum_{k=1}^{K} D_J(h^s(\alpha_k)\|h^T_{ik}),$$

(4.1.2)

where $D_J(\|)$ is the Jeffrey divergence described in Section 3.1, $h^s(\alpha_k)$ is a histogram operator on $s$, which computes the histograms, CpH, SRH, EH and CtH (respectively corresponding to $k = 1, 2, 3, 4$) described in Chapter 2, of all the shapes inside $s$, and $h^T_{ik}$ is the corresponding histogram from the sample $T_i$. Each pixel $x \in \Omega$ is then labeled by the most frequent label of shapes as

$$L(x) = \max_{T_i \in T} \left\{N_x(T_i), N_x(T_i) := \sum_{x \in s, s \in S} \delta(L(s), T_i)\right\},$$

(4.1.3)

where $\delta(L(s), T_i) = 1$ when $L(s) = T_i$, and $\delta(L(s), T_i) = 0$ otherwise.

Figure 4.1 illustrates such supervised segmentation examples on a synthetic texture image composed two textures. The training patches are framed in red on the original image. The result is not bad, in this case, but the texture boundary is not located accurately enough. The failure is due, in this case, to the fact that the segmentation boundary is not included in level lines of the image. Since this appeared to be quite common, we decided, in order to perform texture segmentation, to rely on active contours, as explained in the next section.

4.1.2 Segmentation with active contours

Partitioning an image into different regions of homogeneous texture with active contours has been widely studied [9, 179, 95]. In this section, we investigate the combined use of the texture features described in Chapter 2 and active contours. First, each pixel $x$ from an image $I : \Omega \rightarrow R$ is described by the features of $s(x)$, the smallest shape of the topographic map containing $x$. Five features computed from $s(x)$ are used: the contrast information $\gamma$, defined in Section 2.3.3, the scale ratio $\alpha$ defined in Section 2.3.2, the orientation $\theta$, as well as the elongation $\epsilon$ and the compactness $\kappa$, both defined in Section 2.3.1. Thus, there is a vector of length 5: $v = (\gamma, \alpha, \theta, \epsilon, \kappa)$ for each pixel $x$. The segmentation then amounts to partition the resulting vectorial image. We chose to use an active contour segmentation model to make a 2-phase partition of the image into the background and the objects of interest.
4.1. Texture Segmentation from Shape Ensembles

Figure 4.1: The supervised segmentation results (right) of two synthetic texture images (left). The training patches are framed in red on the original image. Note that partitioning the tree of shapes is not enough for segmentation, due to the fact that the segmentation boundary is not included in level lines of the image.

**Active Contour based on the Kullback-Leibler (KL) divergence** A recent active contour model has been proposed in [95] to find, within an image, two regions with two probability density functions (PDFs) of texture features as disjoint as possible. We chose to adapt this scheme to our framework because it is adapted to histogram-based texture characterization and efficient. Suppose for the moment that each pixel of the image is characterized by a texture feature $f$. Let $p_{in}$ be the inside PDF, $p_{out}$ the outside PDF, $C := C_{in}$ be the evolving region and $\Omega/C := C_{out}$ its complementary in $\Omega$. The method propose to maximize the KL-divergence between the PDFs of the regions inside and outside the evolving active contour $C$. The PDF corresponds to the random variable made of the texture feature $f$. The $p_{in}$ and $p_{out}$ associated with a region $C$ are evaluated thanks to a
Parzen window as follows:

\[ p_{in}(f, C) = \frac{1}{|C|} \int_{C} G_{\sigma}(f - f(x))dx, \quad (4.1.4) \]

\[ p_{out}(f, C) = \frac{1}{|\Omega/C|} \int_{\Omega/C} G_{\sigma}(f - f(x))dx, \quad (4.1.5) \]

where \(|\cdot|\) is the area of a region and \(G_{\sigma}(\cdot)\) is a Gaussian kernel with zero-mean and standard deviation \(\sigma\), which controls the smoothness of the approximation. Then, the symmetric KL-divergence between \(p_{in}\) and \(p_{out}\) is defined as

\[ KL(p_{in}(C) || p_{out}(C)) = \int_{-\infty}^{\infty} \left( p_{in}(f, C) \cdot \frac{p_{in}(f, C)}{p_{out}(f, C)} + p_{out}(f, C) \cdot \frac{p_{out}(f, C)}{p_{in}(f, C)} \right) df \quad (4.1.6) \]

The segmentation then consists in maximizing the difference of the PDFs inside and outside a contour \(C\), as

\[ \arg \min_{C} \{ L(C) - \lambda KL(p_{in}(C) || p_{out}(C)) \} , \quad (4.1.7) \]

where \(L(C)\) is the length of the contour. After computing the shape derivative, Houhou et al. [95] showed that the minimization of the energy in Equation 4.1.7 can be solved by a variational model proposed by Bresson et al. [26], enabling the fast computation of a global optimum.

In our case, making an assumption that the components of \(v = (\gamma, \alpha, \theta, \epsilon, \kappa)\) are independent, together with the additive property of KL-divergence on independent variables, we have,

\[ KL(p_{in}(C) || p_{out}(C)) = \sum_{f \in \{c, \alpha, \gamma, \epsilon, \kappa\}} \int_{-\infty}^{\infty} \left( p_{in}(f, C) \cdot \frac{p_{in}(f, C)}{p_{out}(f, C)} + p_{out}(f, C) \cdot \frac{p_{out}(f, C)}{p_{in}(f, C)} \right) df. \quad (4.1.8) \]

**Results** Several examples of the resulting segmentation scheme are displayed. Figure 4.2 shows segmentation results on three synthetic texture images. Each image is composed of two different textures, which have been radiometrically corrected in order to share the same global mean and standard deviation. We also show the segmentation results obtained on these images using the features proposed by Houhou et al [95]. These features heavily rely on contrast information, and therefore may fail in cases where both textures share the same mean and variance. In comparison, we are able to correctly discriminate between both regions. We also experiment on several real images. The best segmentation results obtained with this method are shown on Figure 4.3 ("best" refers to the best choice for the crucial parameter \(\lambda\) in Formula (4.1.7)). Figure 4.4 illustrates the segmentation results of two Julesz textures, which are composed of simple shapes or terminators. The segmentations are satisfying. However, observe that the obtained segmentations are inconsistent with human texture perception, since these two textures are not distinguishable by pre-attentive vision [100].

Although this approach may yield excellent results, it is important to notice that those results highly depend on the different parameters of the method (mostly the regularization
4.1. Texture Segmentation from Shape Ensembles

Figure 4.2: Segmentations of synthetic texture images composed of two textures (taken from the UIUC and UMD database), radiometrically normalized to share the same mean and standard derivation. The segmentation boundaries (in red) overlay the original images. **Left column**: synthetic textures; **Middle column**: the segmentations obtained by the method of Houhou et al. [95]. **Right column**: the segmentations obtained by the proposed method.

parameter $\lambda$ in the energy), as it is usual with active contour models. These results could certainly benefit from recent developments in global minimization for active contour models such as those of [26].
4. Three Applications of the Theory

4.2 Structured Satellite Image Indexing

Motivation of the application  Remote sensed satellite imaging has been widely applied to agriculture, geology, forestry, regional planning, and many other applications for analyzing and managing natural resources and human activities. In the past few years, with the development of imaging techniques, satellites with very high spatial resolution imaging systems have been launched, e.g. IKONOS, QuickBird, World-View-1, GeoEye-1, which enable satellite imagery to provide more accurate earth observation and measure small objects on the surface up to 0.5 m.

Figure 4.3: Segmentations of natural images. The segmentation boundaries overlaying the original images are displayed in red. Top row: real textures; Middle row: the segmentations obtained by the method of Houhou et al. [95]. Bottom row: the segmentations obtained by the proposed method.
Figure 4.3: (continued) Segmentations of real images. The segmentation boundaries overlaying the original images are displayed in red. **Top row:** real textures; **Middle row:** the segmentations obtained by the method of Houhou et al. [95]. **Bottom row:** the segmentations obtained by the proposed method.

However, satellite images of high spatial resolution present many challenging problems in image understanding, information mining, and pattern recognition. First, with the enhancement of spatial resolution, more details on the earth surface emerge in satellite imagery. Unlike the case of low-resolution satellite images, where texture and intensity cues
4. Three Applications of the Theory

Figure 4.4: Experiments on the segmentation of two synthetic Julesz textures. The segmentation boundaries overlaying the original images are displayed in red. The segmentations are satisfying. However, observe that the obtained segmentations in the middle and on the right are inconsistent with human texture perception, since these two textures are not distinguishable by pre-attentive vision [100].

have been proved to be efficient for recognition [115, 168, 177], structures become more important for analyzing high-resolution satellite images. It is of great interest to investigate new image indexes, which can describe both the structure and texture information for high-resolution satellite image recognition. Second, in satellite images of high spatial resolution, objects contained in the same type of scenes might appear at different scales and orientations. For instance, the buildings in urban areas or the bridges on the river always show at various sizes and orientations. Moreover, if satellite images were taken under different weather conditions, there might be lighting changes between images of the same type. For these reasons, image indexing methods should comply with some invariant properties, such as scale invariance, orientation invariance and contrast invariance.

In order to extract structural features from optical satellite images of high-resolution, [204] proposed to use statistics of straight lines and their spatial arrangement over relatively small neighborhoods. [20, 21] suggested to use geometrical information, e.g. edge and Junction density, from the extracted road network and segmented urban regions for structural satellite image indexing. As inspired by the works in computer vision, [151] investigated interest point descriptors, such as Scale-invariant feature transform SIFT, for characterizing remote sensed images. Other structural features are computed from the pre-segmentation of images. One main disadvantage of this kind of approaches is that they rely on some pre-analysis of images, such as edge detection and segmentation, which are in themselves challenging problems. In addition, when these indexing schemes focus on structure information, they ignore the use of texture cues. In this section, We apply the proposed shape-based texture indexing method to characterize satellite images.
4.2. Structured Satellite Image Indexing

High-resolution Database  To test the proposed satellite image indexing method, we collect a set of satellite images exported from Google Earth\(^1\), which provides high-resolution satellite images up to \(0.5\) m. Some samples of the database are displayed in Figure ??.

It contains 12 classes of meaningful scenes in high-resolution satellite imagery, including Airport, Bridge, River, Forest, Meadow, Pond, Parking, Port, Viaduct, Residential area, Industrial area, and Commercial area; For each class, there are 50 samples. It’s worth noticing that the image samples of the same class are collected from different regions in satellite images of different resolutions and then might have different scales, orientations and illuminations.

![Figure 4.5: Some samples of the testing high-resolution satellite image database. For each class, there are 50 samples, and 4 of which are shown here.](image)

Methodology and results  For satellite imagery in panchromatic format, all the structure information is, of course, contained in the gray scale image. However, for a multispectral satellite image \(U = \{u_1, u_2, \ldots, u_L\}\) of \(L\) bands, an \(L\)-dimensional vector is stored for each pixel. In this case, we suppose that the main structure information of \(U\) is included by its \(p\)-energy channel \(\mathcal{L}\), defined as

\[
\mathcal{L} = \left( \sum_{u_i \in U} (u_i)^p \right)^{\frac{1}{p}}, \quad p \geq 1.
\]  

(4.2.1)

Actually, Caselles et al. have proved that the main geometric information of natural color images are contained in their luminance channel \([39]\) (where \(p = 1\)). In the context of this work, as we shall see, we will only deal with natural color satellite images, so the analysis of structure information is based on the luminance channel. The same scheme could be applied to multispectral images using the \(p\)-energy channel \(\mathcal{L}\).

\(^1\)http://earth.google.com/
We apply the proposed analysis scheme to two common satellite image recognition tasks: retrieval and classification, as the experiments on textures.

![Figure 4.6: Average retrieval (a) and classification (b) performance.](image)

Figure 4.6a shows the average retrieval performance on the whole database. It indicates that by using only one sample, averagely 52.84% samples of the same class can be correctly retrieved among the first 49 matches. However, in the same context, if we use the mean and standard deviation of Gabor filter responses, only 21.19% samples can be retrieved averagely. According to the performance curve, when the number of matches is extended to 200, 90.50% samples can be retrieved by using the structural indexes. But the same percentage to Gabor features is 54.13%.

Some illustrations of the retrieval results are displayed in Figure 4.7, 4.9a, 4.9b, where a query image is followed by its first 49 closest samples. The retrieval results for all samples can be found at.

<table>
<thead>
<tr>
<th>Category</th>
<th>1 training sample</th>
<th>25 training sample</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StructInd</td>
<td>GaborF</td>
</tr>
<tr>
<td>Airport</td>
<td>53.39</td>
<td>12.72</td>
</tr>
<tr>
<td>Bridge</td>
<td>40.45</td>
<td>7.73</td>
</tr>
<tr>
<td>Commercial</td>
<td>48.74</td>
<td>23.17</td>
</tr>
<tr>
<td>Forest</td>
<td>80.94</td>
<td>42.54</td>
</tr>
<tr>
<td>Industrial</td>
<td>42.93</td>
<td>20.11</td>
</tr>
<tr>
<td>Meadow</td>
<td>75.06</td>
<td>35.08</td>
</tr>
<tr>
<td>Parking</td>
<td>68.74</td>
<td>10.58</td>
</tr>
<tr>
<td>Pond</td>
<td>63.96</td>
<td>29.91</td>
</tr>
<tr>
<td>Port</td>
<td>50.93</td>
<td>12.79</td>
</tr>
<tr>
<td>Residence</td>
<td>32.17</td>
<td>19.04</td>
</tr>
<tr>
<td>River</td>
<td>60.23</td>
<td>27.28</td>
</tr>
<tr>
<td>Viaduct</td>
<td>64.19</td>
<td>13.24</td>
</tr>
</tbody>
</table>

Table 4.1: Average classification rate (%) of Structural indexes (using StructInd for short) and Gabor Features (using GaborF for short) on each category of the database, with the number of training samples as 1 and 25, respectively.
4.2. Structured Satellite Image Indexing

Figure 4.7: A retrieval result of the bridge category obtained by using the proposed indexing scheme. The query image is in the first position and the 49 most similar samples follow, ordered by their matching scores. The false samples are framed in red.

Figure 4.8: Some false alarms of bridge retrievals. The parts framed in red really contain some bridge-like structures.

Figure 4.7 shows a retrieval result of bridge category, which is very structured. Observe that even though there are large illumination changes between samples and the query image, the method works well, thanks to the contrast invariance of the indexing scheme. It’s also interesting to inspect the false alarms and observe that there often contains some structures similar to bridges, see the parts framed in red inside Figure 4.8. Figure 4.9a and 4.9b illustrate two retrieval examples respectively on river and viaduct class. Even structures in this two classes are complicated, the proposed approach works well. All the retrieval results are available in the address [219].

Figure 4.6b shows the average classification performance by using nearest-neighbor classifier, when the number of training images ranges from 1 to 25. It indicates that the
4. Three Applications of the Theory

(a) A retrieval example of river category

(b) A retrieval example of viaduct category

Figure 4.9: Some retrieval results obtained on the database using the proposed indexing scheme. The query image is in the first position and the 49 most similar samples follow, ordered by their matching scores. The false samples are framed in red.

Structural indexes outperform the Gabor features dramatically. Furthermore, Table 4.1 shows the average classification rate for each class of the database. We can see that Gabor features are efficient only on some texture classes, e.g. forest and meadow. The proposed structural indexes work well on classes with complicated structures such as viaduct and
airport, and also on classes containing more textures.

However, we found that the structural indexes can not distinguish industrial and residential classes well. This is because those two categories share many similar structures, and some semantic information of the scene might be helpful [24].

### 4.3 Image Retrieval based on Statistical Manifolds

In this paragraph, two enhancements are proposed in order to compare texture images using the indexing scheme from Chapter 2. First, it is shown that relying on geodesic distances significantly improve both retrieval and classification performances. Second, we introduce a method for the automatic setting of the matching threshold when querying a database with a texture image.

**Improving recognition performances through geodesic distances** In Section 3.2, we compared the proposed texture indexing scheme with three recent sets of invariant descriptors. We therefore followed a prescribed protocol for classification relying on nearest neighbors. Usually, however, classification performances benefit from some more powerful classifiers such as support vector machine (SVM) or from some classification schemes such as AdaBoost. For example, Zhang et al. [229] confirmed that using an SVM classifier instead of the nearest-neighbor classifier improves the classification performance of a bag-of-features representation. In addition, in recent years, it has been suggested that content-based image retrieval (CBIR) benefits from measuring the similarity between images along some intrinsic manifold, even though there is no convincing evidence that such structure actually exists [231, 90]. However, as we shall see, when applied to histogram-based image descriptors, the assumption that images can locally be compared by an $L^2$ norm may not be optimal. In fact, we will see that locally comparing images thanks to the Jeffrey divergence (as in section 3.2) and then using a geodesic distance is very efficient. This approach was previously advocated, for cases where images are represented through the distribution of some features, in [32], in the framework of information geometry, under the name FINE (Fisher Information Nonparametric Embedding). The approach is also equivalent to the first step of the classical ISOMAP algorithm [200], combined with the use of the Jeffrey divergence.

**Threshold on matching** Once a convenient similarity measure has been defined, most retrieval systems do not touch the problem of deciding which images should be matched with a query. The similarity measures may be truncated by some fixed threshold set by the users, independently of the query. The a contrario framework, originally introduced to automatically set detection thresholds in computer vision [58], can settle this decision problem by selecting images whose proximity to a given query cannot be explained by some simple stochastic model. This approach has been recently shown to provide thresholds that adapt both to the query and the database, for the matching of shapes [149] or local features [163]. In the context of image retrieval, the methodology has been applied to the comparison of color images through the EMD distance [97]. The second motivation
is to introduce a retrieval methodology adapted to histogram-based indexing schemes, addressing both the choice of a similarity measure and the decision step.

### 4.3.1 Fisher information distance for image comparison

Consider an image database \( X = \{x_1, \ldots, x_N\} \), in which each image \( x_i \) is described by features \( \Psi_i \). It is sometimes possible [231, 90] (for instance if the database if composed of different viewpoints of the same object) to view images as points on a manifold, embedded in a Euclidean space. In this case, geodesic distances along this manifold can be used for retrieval. In practice, a \( n \)-nearest neighbor graph \( G \) is built from \( X \) and the geodesic distance between two images is approximated by the length of the shortest path between the images in \( G \).

When images are described by histogram-based features, they can be viewed as points lying on some statistical manifolds, the distance on which is the Fisher information metric [5, 197, 32]. It has been underlined [32] that the Fisher information distance compares favorably with Euclidean-based geodesic distances. If each image \( x_i \) is described by \( K \) distributions \( \{\psi^1_i, \psi^2_i, \ldots, \psi^K_i\} \), such a distance can be approximated through geodesic distances by relying for instance on the Jeffrey divergence \( D_J \) between distributions [32], as

\[
D_F(x_i, x_j) = \min_{l=0}^{m-1} \left( \sum_{k=1}^{K} D_J(\psi^k_i \| \psi^k_j) \right)^{1/2},
\]

where the minimum is taken over all possible lengths \( m \) and all possible paths between \( \Psi_i \) and \( \Psi_j \), as \( \{\Psi_i, \Psi(1), \ldots, \Psi(m-1), \Psi_j\} \), on the graph \( G \). More details are given in Algorithm 1.

Replicating the same experiments as in Chapter 3 and replacing Jeffrey divergence by this geodesic distance yields a very clear improvement of performances, especially for the retrieval task, as can be observed in Figure 4.10. In these experiments, a value of \( k = 10 \) has been used for the number of neighbors.

For comparison, we also compute a geodesic distance \( D_G \) from the \( n \)-nearest neighbor graph relying on the \( L^2 \) distance between histograms. The compared retrieval performances are shown in Figure 4.11 (a-b) for invariant texture retrieval (for a joint database UIUC+UMD) and in Figure 4.11 (c) for a satellite image database. These two experiments show the superiority of \( D_F \), the geodesic distance approximating the Fisher information distance, over both \( D_G \) and the plain use of Jeffrey divergences, for the two different retrieval tasks and sets of features.

The computation of geodesic distances is often followed in the literature by a step of dimension reduction, as it is the case with the ISOMAP [200] algorithm. However, observe that for retrieval tasks, dimension reduction is often useless and tends to make distances less precise and to decrease retrieval performances.

### 4.3.2 Adaptive image retrieval based on statistical manifolds

In this part, we present an adaptive image retrieval approach, relying on the a contrario methodology and the histogram-based geodesic distance described in Section 4.3.1.
Algorithm 1: Calculate $D_F(x_i, x_j)$ on sampled manifolds

Require: An image database $X = \{x_1, \ldots, x_N\}$, where each image $x_i$ is described by $K$ distributions $\{\psi_i^1, \psi_i^2, \ldots, \psi_i^K\}$.

Ensure: Fisher information distance $D_F(x_i, x_j)$.

1. Compute the similarity matrix:
   
   for $1 \leq m, n \leq N$
   
   Calculate the Jeffrey divergence $D_{JS}(x_m, x_n)$ between $x_m$ and $x_n$, as follows:
   
   $D_{JS}(x_m, x_n) = \sum_{k=1}^{K} D_J(\psi_m^k || \psi_n^k)$
   
   $= \sum_{k=1}^{K} \int \psi_m^k(z) \cdot \log \frac{\psi_m^k(z)}{\psi_n^k(z)} + \psi_n^k(z) \cdot \log \frac{\psi_n^k(z)}{\psi_m^k(z)} \, dz$

2. Build the $k$-nearest neighbor graph or $\epsilon$-neighborhood graph $G$ of $X$:
   
   for $1 \leq m, n \leq N$
   
   Set the connection weight $W_{ij}$ between $x_m$ and $x_n$ on $G$ as follows,
   
   $W_{ij} = \begin{cases} \sqrt{D_{JS}(x_m, x_n)}, & \text{if } x_m \text{ and } x_n \text{ are connected}, \\ \infty, & \text{otherwise}. \end{cases}$

3. Calculate $D_F(x_i, x_j)$ by pursuing the geodesic distance on graph $G$, via Floyd [67] or Dijkstra algorithm [61].

An a contrario model The a contrario methodology relies on the idea that structures of interest in images can be detected as those which contradict a simple background model, or null hypothesis. It has first been proposed in [58] for the detection of alignments and has then been successfully applied to a wide variety of computer vision tasks (see the monograph [60]). In particular, it has been recently applied to the problem of image matching [163]. Following the same path for image retrieval, we will see that this methodology permits to define adaptive thresholds on distances. Let us also mention that in the second part of this thesis, we will make an extensive use of the a contrario approach in the context of junction detection.

Assume that each image $x_i$ of the database $X$ can be described by a list of features $V_i = \{v_i^1, \ldots, v_i^M\}$ and consider a measure of similarity $D$ between images which can be written as a sum $D(x_i, x_j) = \sum_{m=1}^{M} d(v_i^m, v_j^m)$, where $d$ is a measure of similarity between the features $v_i^m$ and $v_j^m$. The image $x_i$ being given, and $x$ being a random image, consider the statistical hypothesis $H^0_i : \text{"the random variables } d(v_i^m, v^m) \text{ are mutually independent"}$. Now, let $x$ be a random image, described by $\{v^1, \ldots, v^M\}$. The index $i$ being given, we want to test the hypothesis $H^0_i$ that the random variables $d(v_i^m, v^m)$ are mutually independent. If $H^0_i$ is satisfied, the density of the random variable $D(x_i, x)$ is
4. Three Applications of the Theory

Figure 4.10: Improving performances through geodesic distances Average retrieval (a) and classification (b) performances of the descriptor SI+CtH (Similarity Invariant + Contrast Histogram) on UIUC database, with (red curve) and without (blue curve) geodesic distances. Figures (c) and (d): same layout for the UMD database.

given by \( \bigstar_{m=1}^{M} p_{m}^{i} \), where \( \bigstar \) denotes the convolution and where each \( p_{m}^{i} \) is the density of the random variable \( d(v_{m}, v_{m}^{i}) \). It follows that for all \( t \), the probability that the distance between \( x_{i} \) and a random image \( x \) is smaller than \( t \) is

\[
F_{i}(t) := \mathbb{P}_{\mathcal{H}_{0}}(D(x_{i}, x) \leq t) = \int_{-\infty}^{t} \lim_{m=1}^{M} p_{m}^{i}(y)dy.
\]  

(4.3.3)

Observe that in practice, the densities \( p_{m}^{i} \) can be estimated as histograms of the distances \( d(x_{i}^{m}, x_{j}^{m}) \) when \( i \) is fixed and \( j \) spans the database \( X \).

This result permits to define adaptive thresholds on the distance \( D \). For a given number of false alarm \( \epsilon \), we define

\[
t_{i}(\epsilon) = \arg\max_{t \geq 0} \{ F_{i}(t) \leq \frac{\epsilon}{N} \}.
\]  

(4.3.4)

An image \( x_{j} \) is then considered as similar to \( x_{i} \) (validated for retrieval) if \( D(x_{i}, x_{j}) \leq t_{i}(\epsilon) \), i.e if it is unlikely that a random image \( x \) is closer from \( x_{i} \) than \( x_{j} \) is.
4.3. Image Retrieval based on Statistical Manifolds

Figure 4.11: Comparison of the retrieval performances between Jeffrey divergence $D_J$ (red), Euclidean Geodesic $D_G$ (blue) and Fisher information distance $D_F$ (green). (a) and (b) illustrate the results on the texture database: (a) is with the histograms of shape-based granulometries [223] and (b) is with the phase histograms [135]. (c) displays the results of satellite image retrieval.

Figure 4.12: Adaptive thresholds vs. fixed thresholds. The red and green curves show "average number of good retrievals vs. average number of false alarms" according to different thresholds $t(\epsilon)$ with $\epsilon = \{0.1, 1, 10, 100\}$. The blue curves are obtained using fixed threshold on the similarity measure.

Adaptive image retrieval In order to apply the previous approach to the distance $D_F$ (as defined in Section 4.3.1) we need to write it as a sum of distances. To this aim, we propose to embed the data into an Euclidean space while respecting the similarity measures given by $D_F$. This can be achieved by using a classical Multi-Dimensional Scaling (cMDS), as used in ISOMAP, or by computing eigenmaps, as done in Laplacian Eigenmaps, a method that we will call LE from now on. As observed earlier, multi-Dimensional Scaling suffers from its lack of locality when it is applied to retrieval tasks. For this reason, we rely in our experiments on Laplacian Eigenmaps, LE. This is also the choice made in [32], for the so-called Fisher Information Nonparametric Embedding (FINE). We will keep this acronym for the resulting embedding (Jeffrey divergence, followed by geodesic distance, followed by LE embedding). In order to catch up with the independence requirement of the a contrario approach, we then apply a principal component analysis (PCA) to the features obtained, while keeping the same dimension to prevent information loss.
to the resulting embedding as FINE-PCA. The *a contario* parameter setting detailed in the previous paragraph is then applied to the resulting framework. The whole adaptive image retrieval algorithm can be summarized as in Algorithm 2.

**Algorithm 2**: Adaptive image retrieval on statistical manifolds

1: For each \(x_i, x_j \in X\), compute \(D_F(x_i, x_j)\) by using Equation (4.3.1) (the shortest paths in the graph are computed thanks to the Floyd algorithm);
2: Compute the Euclidean embedding \(V\) of \(D_F\) using LE;
3: Change the coordinates of \(V\): \(V' = PCA(V)\);
4: For each index \(i\), estimate the densities \(p_{im}\) empirically from the histograms of \(d(v_i^m, v_j^m) = \|v_i^m - v_j^m\|_2\), and compute the function \(F_i(t)\) thanks to Equation (4.3.3);
5: The value \(\epsilon\) being fixed, estimate the thresholds \(t_i(\epsilon)\) using Equation (4.3.4);
6: Set of retrieved images for the query \(x_i\): \(X_{re} = \{x_j, D_F(x_j, x_i) \leq t_i(\epsilon)\}\).

**Results** The proposed approach is validated on the image databases previously considered in this thesis. We consider the task of invariant texture retrieval, as well as texture-based satellite image retrieval.

The task of invariant texture retrieval, that is the task of retrieving textures under varying viewpoints and illumination conditions, has already been addressed in the previous chapters. Recall that, in order to index textures while being invariant to complex geometric distortions and local contrast changes, several approaches have proposed to represent images through the distribution of local features [113, 135, 226, 223]. We choose two type of features to illustrate the interest of both the Fisher information distance and the adaptive retrieval scheme: the shape-based approach introduced in Chapter 2 and the wavelet-like approach of Mellor et al. [135]. Retrieval is performed on a database obtained by merging two texture sets commonly used to test invariant texture indexing: the UIUC database [113] and the UMD database [226]. This results in 2000 images organized in 50 classes and 40 samples per class, and some samples are shown in Figure ??.

The considered texture features are two sets of histograms: 4 histograms of shape-based granulometries on the one hand (elongation, compactness, contrast, and scale ratio histograms, as described in Chapter 2), and the 2 four-dimensional phase histograms of similarity invariant filter responses described in [135] on the other hand. The proposed retrieval scheme is applied separately to these two feature sets to illustrate its generality.

For satellite image retrieval, we use the same database as in Section 4.2, made of 500 images, classified in 10 different ground types (50 images for each type). The used image features are the same as in Section 4.2: 6 histograms of shape-based granulometries [223] (the four previous histograms, as well as orientation and elongation histograms, suited to the global similarity transforms encountered in satellite images) together with an histogram of gradient intensity.

\(^2\)The database can be found at [http://www.tsi.enst.fr/~xia/satellite_image_project.html](http://www.tsi.enst.fr/~xia/satellite_image_project.html).
Fisher information distance. As can be seen on the curve of Figures 4.10 and 4.11, using the Fisher information distance, yields a significant increase of performances. Let us insist on the fact that this distance, as described by Algorithm 1, is especially simple to compute.

Adaptive image retrieval. The adaptive image retrieval experiments follow the steps described in Algorithm 2. We compare it with the use of fixed thresholds (that is, the same threshold for each query). Observe that using adaptive or fixed thresholds does not affect the similarity distance, and therefore the order in which images are returned from a query. The goal of the decision step is to automatically set the threshold for retrieval, depending on the query and the database content. Therefore, this step is evaluated by plotting the curve “average number of good retrievals vs. average number of false alarms”. Results are averaged over the database. For non adaptive retrieval, each point of the curves is obtained using a fixed threshold on the similarity measure. When using the adaptive approach (either FINE or FINE-PCA), each point is obtained using a fixed value for \( \epsilon \), yielding variable thresholds on the similarity measure thanks to Equation (4.3.4). Results for invariant texture retrieval are shown in Figure 4.12 (a)-(b), where the ideal case is the point at the bottom-right corner. The equivalent for satellite retrieval is shown in Figure 4.12 (c). In the two experiments, the adaptive thresholds \( t \) corresponds to \( \epsilon = \{0.1, 1, 10, 100\} \). The fixed thresholds are set as \((0.05, 0.10, 0.15, 0.20)\) times the corresponding maximum distance. These curves show the improvement of adaptive thresholding (FINE and FINE-PCA) over fixed thresholding, in that they yields better good versus false ratios. Another observation concerns the difference between FINE and FINE-PCA. The use of PCA enables a better prediction of the number of false detections. Indeed, using uncorrelated features is more compatible with the null hypothesis (independence of features) that directly using FINE features. Observe however that the prediction of false alarms is far from being perfect and that one get more false detections than expected. This may be due to the non-independence of features after PCA. Nevertheless, the embedding provided by FINE provides a reasonable compromise between discriminative power and independence.
4. THREE APPLICATIONS OF THE THEORY
Part II

Structured Image Manipulation
Chapter 5

A Generic Structured Image Manipulation Framework

5.1 Introduction

Figure 5.1: Starting from the image on the left, its geometrical structure is analyzed. Then, each shape of the resulting hierarchical structure is replaced either by a disk (center left) a rectangle (center right) or a shape from a Kandinsky painting (right).

\textit{Je ne peins pas les choses, je ne peins que les rapports entre les choses.}

\textit{Henri Matisse}

High level structural properties of images, such as interactions between objects through inclusion or occlusion, play an important role for the artistic representations of scenes. While, in some artistic movements such as impressionism or pointillism, objects are represented through relatively local effects, other schools such as Cubism or Futurism heavily rely on their structural properties. Matisse went far in this direction, by asserting "I don’t paint things, but only relationships between things".

In the field of automatic generation of artistic or functional styles, known as non-photorealistic rendering (NPR), a recurrent trend over the last ten years has been to use
image analysis tools before the rendering stage. Several such works, making use of image segmentation procedures or of scale spaces, are reviewed in the next section. In particular, the work of Song et al. [195] shows that the use of several fixed scale image segmentations enables the creation of exciting image abstractions that are beyond the reach of local image analysis tools. The intrinsic difficulty of such a task is already visible in the pioneering work of Haeblerli [83], where two examples of abstract versions of images are obtained through computationally involved optimization procedures.

In this chapter, we are concerned with extreme geometrical abstractions of digital photographs. In particular, we suggest a way to abstract images by replacing all geometrical shapes by one or a few reference shapes. As a result, we obtain images that evoke the abstraction painting schools of the beginning of the 20th century while giving a faithful account of an example image structure. In order to achieve this goal, a structured representation of the image, accounting for relationships between objects, is needed. We make use of a hierarchical and morphological representation of images. As a byproduct, we show that the same framework enables the creation of other non-photorealistic renderings.

Let us precise at this point that in the field of NPR, the term image abstraction refers to the process of generating simplified, stylized versions of photographs [55, 154]. This task usually requires the omission of small details that are unimportant for our understanding of a scene. While also grounded in some simplification of the image structure, our goal here is different since we investigate the result of an extreme schematisation of images only relying on a few basic shapes and not on the regularization of existing image regions.

We now briefly describe the approach retained in this chapter. First, the geometrical structure of the image is analyzed through the use of the topographic map [37], already defined and used in Section 2.2. Recall that all connected components of level sets, the so-called shapes of the image, are organized in a hierarchical manner to form a tree accounting for inclusion properties. Recall also that the resulting representation of the image is both complete (that is, contains all the image information). Then, the geometry of these shapes is modified, in either light or extreme ways. Eventually, a new image is created from the resulting tree. The geometrical alterations of shapes proposed in this chapter range from simple random displacements of shapes, creating painting-like effects, to the replacement of every shape of the image by shapes from a dictionary, according to some replacement rules.

The strong points of the proposed approach are as follows. First, the image is represented using a collection of shapes, yielding a direct grip on geometrical manipulations. Second, the topographic map account for all scales in the image, enabling interplays between textures and macroscopic structures of the image. Last, the hierarchical structure of the chosen representation allows inclusions properties between shapes to be partially preserved in the synthesized image, yielding highly structured abstract images.

The rest of this chapter is organized as follows: in Section 5.1.1 related works in the field of NPR are presented; in Section 5.2, motivations for using the topographic map in the context of image manipulation are given; Section 5.3 is the core part of the chapter, where the proposed framework is described in details and examples are given. In Section 5.5, applications of the same framework for image filtering are briefly presented. Finally, in Section 5.6, we provide a discussion.
5.1. Introduction

5.1.1 Related work

The origins of NPR take place in interactive painting and drawing systems [83, 215], which describe techniques for creating numerical paintings by sequentially adding brush strokes on an image. A lot of works have since been dedicated to the automation and refinement of stroke-based rendering, e.g. [125, 92, 93, 77, 48, 28]. Most of these works rely on a preliminary image analysis step which encodes the image structure either completely or partially: edge detection [125], linear scale-space [92], region extraction [77], salience map [48], multi-scale edge detection [89], Laplacian pyramids [28], etc.

More recently, several papers have focused on image simplification or abstraction. These works tend to simplify a photograph by highlighting important edges and smoothing regions which are considered as perceptually negligible. A natural way to achieve this kind of effect is also to use image analysis tools. In order to render the geometrical structure of images at multiple scales, several papers [55, 183, 154] rely on a linear scale space. De Carlo and Santella [55, 183] use a hierarchical segmentation and a pruning of the corresponding structure to create an adaptive segmentation of the initial image. The choice of the level of details in each part of the image relies on the local contrast and on a user-driven measure of visual attention. Alternately, Orzan et al. [154] follow Canny edges through a linear scale space in order to define a multi-scale edge-based structure and to create a simplified image by solving a Poisson equation. An alternative direction, suggested by Bangham et al. [13], is to use non-linear scale spaces to produce artistic effects, a path further explored by Kang et al. [103]. Zhao et al. [230] present an interactive system, named Sisley, for abstract painting. It starts with a hierarchical decomposition of the image with interactive guidance from the user, and then automatically renders different components to generate abstract painting.

There are relatively few works concerned with the abstraction of images obtained through radical geometric modifications of shapes. The simplest approach to such a geometric abstraction is the one to be encountered in popular plug-ins aiming at the creation of cubist-like images. These usually amount at superimposing to the original image simple geometric shapes over which the colors are averaged. A more involved approach was long ago suggested in the inspiring work of Haeblerli [83], where geometric shapes are superimposed to the image in order to minimize some energy. Such approaches involve complex optimization procedures, usually relying on some heuristics. A more structured approach to the same goal has been recently presented in [195], where a segmentation of the image (in this case a normalized cut) is used to replace shapes by some geometric primitives such as disks or rectangles, yielding very interesting abstract images. In the present work, we propose to use the topographic map, a hierarchical structure accounting for the inclusion of shapes, as a preliminary analysis step in order to obtain geometrical abstractions of images. The advantages of the proposed analysis tool over more classical ones are further developed in the discussion section.
5.2 Organizing Shapes by Topographic Maps

The topographic map, introduced by Caselles et al. [37] in computer vision, has been described in Chapter 2. It is a hierarchical structure composed of shapes and relying on connected components of level sets. It is a powerful way to represent and manipulate the geometrical content of a gray level image [59, 39, 31]. In particular, we shall see that this map is a flexible and efficient tool to transform or simplify the geometrical structure of an image, while preserving the inclusion properties of shapes. The goal of this section is to figure out the structural properties of the topographic map in the context of image manipulation.

As mentioned in Chapter 2, we also rely on the Fast Level Set Transform (FLST) [145], available in the free processing environment Megawave1, to compute the topographic map. For more precision and results on the topographic map, we refer to the recent monograph [34]. A schematic representation of the topographic map is displayed in Figure 5.2.

![Figure 5.2: From left to right: a flower image, its simplified topographic map and the structure of the corresponding tree.](http://www.cmla.ens-cachan.fr/Cmla/Megawave/)

Three aspects of this map make it particularly suitable for manipulating the geometrical content of an image. First, it constitutes a complete representation of the image content, without any loss of information, contrarily to edge descriptions. This map is therefore well adapted for fine image modifications, even at very small scales. Second, this representation is multi-scale and gives the possibility to play simultaneously with all scales of an image in different ways. Finally, since the topographic map is a hierarchical structure of shapes, which are simply connected, it is especially well adapted to render the inclusion properties of objects in images. For instance, if an image represents a small white disk over a large black disk, the black disk will appear only once in the topographic map, as the parent of the white one. In comparison, a hierarchical segmentation of the image would code the large disk twice, both at a large scale (as a disk), and at a small scale (as a ring around the small disk), making the interpretation of the image as a stack of different object layers more difficult. This particularity of the topographic map will appear as essential in Section 5.3.

Several approaches have been proposed to compute the topographic map of color images. In [39], Caselles et al. show that most of the geometrical content of color images

---

1http://www.cmla.ens-cachan.fr/Cmla/Megawave/
is contained in topographic map of its gray level channel, and therefore advise to use the shapes of this map, associated with their average saturation and hue, to describe color images. In [46], Coll et al. propose to define a total order in the HSV space with a good correspondence to the human perception of geometry. The topographic map of a color image is then computed on this channel. Finally, the authors of [78] propose a color topographic map representation relying on a dichromatic reflectance model. These last two approaches are time consuming and show minor improvement in most images. In this chapter, we stick to the simplest approach of [39].

Figure 5.3: The proposed framework for image abstraction. The image is first decomposed into a tree of shapes $T$ by using FLST [145]. Then, both the tree structure and the shapes are manipulated or edited by the user in automatic or interactive ways, which results in modified tree of shapes $T'$. The synthesized image is finally reconstructed from $T'$.

### 5.3 A Flexible Image Manipulation Framework

#### 5.3.1 General outline

In this section, we present the general framework in which the image structure can be edited and manipulated. This generic procedure is illustrated in Figure ???. First, the input image $I$ is decomposed into a tree of shapes $T = (S_i)_{i=1,...,N}$, using the image representation presented in the previous section. Then, the tree $T$ can be manipulated or edited by the user in automatic or interactive ways. The proposed modifications include shape selection procedures, as well as modifications of the placements and/or the geometry of shapes. These modifications result in a new and modified tree structure $T'$, from which the resulting image is obtained.
5. A Generic Structured Image Manipulation Framework

Decomposition  The decomposition of the input image is implemented by the \textit{Fast Level Set Transformation} (FLST) \cite{145}. First, for a color image $I$, its luminance channel $I_l$ is computed in HSV color space. Then, the topographic map $T = (S_i)_{i=1,...,N}$ of the luminance image $I_l$ is computed. Eventually, for each shape $S_i$, a color $c_i \in \mathbb{R}^3$ is obtained by averaging in RGB space the color values of all pixels included in $S_i$. We call the resulting structure $(S_i, c_i)_{i=1,...,M}$ a \textit{colored tree}.

Manipulation  As a second step of the image modification procedure, the input colored tree is modified. The main modifications of a colored tree $(S_i, c_i)_{i=1,...,M}$ to be illustrated in this chapter are

- \textbf{Shape selection} Only a subset $(\tilde{S}_i)_{i=1,...,M}$ of the shapes $(S_i)_{i=1,...,N}$ are kept. The resulting tree $T' = (\tilde{S}_i)_{i=1,...,M}$ is obtained by defining the parent of $\tilde{S}_i$ as the first ancestor shape of $S_i$ in $T$ that has not been suppressed. The root of the tree is kept unchanged. Three modalities of selection are used in the examples of this chapter.
  
  - Selection according to geometric properties, such as area, elongation, and compactness.
  - Selection according to the contrast of shapes following the automatic procedure presented in \cite{59}. In the following, we refer to the resulting topographic map as a \textit{contrasted sketch} of the image.
  - Interactive selection of shapes, where only shapes meeting a user defined region are kept. For instance, we can interactively segment the image into background and foreground, and only keep shapes belonging to the foreground for rendering.

- \textbf{Shape displacement} The structure of the tree $T$ is kept unchanged, and for each $i = 1,...,N$, a random translation and/or a random rotation is added to $(S_i)$. The displacements can depend on the properties of $S_i$.

- \textbf{Geometric transformation} The structure of the tree is kept unchanged. Each shape $S_i$ is replaced by a new shape $\tilde{S}_i$ drawn from an auxiliary dictionary of shapes (made of simple shapes such as disks or ellipses, made of hand drawn shapes, made of shapes from another image, etc.). The new shape can be chosen according to the geometrical or color properties of $S_i$.

- \textbf{Color modification} The color $c_i$ of each shape is modified by using some color palette, possibly inherited from another colored tree. The new color can depend on geometric properties of the shape.

Reconstruction  Eventually, we reconstruct an image from the modified tree of shapes. Observe that in a tree resulting from either shape displacements or geometric transformations, the parent-children relationships no longer correspond to the inclusion of shapes, as in the input tree $T$. Therefore, one has to make some choice on how to reconstruct images from the tree of shapes. The rendering order of shapes on the tree is given by Algorithm 3. Shapes are superposed upon the image plane from root to leafs and with an increasing order
of compactness. An illustration of the root-to-leaves and less-compact-shape-first rendering order is given in Figure 5.4.

Notice that the shape manipulation of the previous stage could yield aliasing effects when the new shapes are used for the reconstruction. In order to overcome this issue, we choose a simple matting technique consisting in adding a transparency channel with Gaussian decreasing around the shape. The parameter of the Gaussian will be chosen as 0.4 in all forecoming experiments.

Algorithm 3: root-to-leaves and less-compact-shape-first rendering order

**Input:** Tree of shapes \((S_i)_{i=1,\ldots,N}\);

**Output:** Index order \(Q\);

**begin**

Create a queue \(Q\);
Create a list \(C\);

**foreach** shape \(S_i\), \(i = 1,\ldots,N\) **do**

- Compute the compactness, \(\kappa(S_i)\), of \(S_i\);
- Compute the number of children, \(n(S_i)\), of \(S_i\);
- **if** \(n(S_i) == 0\) **then**
  - Put \(S_i\) into \(C\);

**while** \(C\) is not empty **do**

- Select the most compact shape \(s\) from \(C\): \(s = \max_{S_i \in C} \kappa(S_i)\);
- Enqueue \(s\) into \(Q\) and erase \(s\) from \(C\);
- Update \(n(s^f)\), where \(s^f\) is the parent shape of \(s\), as \(n(s^f) \rightarrow n(s^f) - 1\) **if** \(n(s^f) == 0\) **then**
  - Put \(s^f\) into \(C\);

Invert the queue \(Q\).

---

Figure 5.4: An illustration of the root-to-leaves and less-compact-shape-first rendering order. Shapes are superposed upon the image plane from root to leaves and with an increasing order of compactness.
5.3.2 Image abstraction

We first show how the generic decomposition-modification-reconstruction framework presented above enables the creation of abstract versions of digital photographs. In short, we replace all shapes of a given image by shapes from a dictionary. This way, we produce images that at times can evoke the abstract paintings of the Suprematist school, while retaining the overall composition of a given input image. Observe however that if one were to establish a parallel with art, the proposed framework would more evoke the approach of the Cubist school. Quoting Picasso, “There is no abstract art. You must always start with something. Afterward you can remove all traces of reality, there is no hazard since the idea of the object has left an ineffaceable print” [64].

Image are abstracted following the generic framework presented in the previous section. The first step consists in extracting the topographic map and possibly simplifying it, as will be specified for each example. Then, each shape is replaced by a shape extracted from a prescribed dictionary. More precisely, for each shape $S$, a shape $D$ is chosen in the dictionary so as to match $S$. In all the following examples, the shape is chosen as the one minimizing

$$d(S, D) = |e(S) - e(D)| + |\kappa(S) - \kappa(D)|,$$

where $e$ is the elongation of the shape and $\kappa$ is its compactness, as already used in the previous chapters for texture analysis. Writing $\lambda_1 > \lambda_2$ for the two eigenvalues of the normalized inertia matrix of $S$, recall that $e$ and $\kappa$ are defined as $e = \lambda_2/\lambda_1$ and $\kappa = (4\pi)^{-1}(\lambda_1\lambda_2)^{-1/2}$. Then, the shape $S$ is replaced with $T(D)$, where $T$ is an isometry, chosen so that the area and orientation of $T(D)$ matches the one of $S$. In all examples, the color $c$ associated with $S$ is kept unchanged, except when we perform shape transfer between two images, in which case the color may also be transferred.

**Single shape rendering** All shapes of the image are replaced with a simple geometric primitive. In Figure 5.1 as well as on Figure 5.6, top right, and Figure 5.7, middle, we display examples obtained using a single disk as a dictionary. In the first two examples, the background was eliminated using an interactive segmentation procedure, shapes are rendered using the relief effect presented at the end of Section 5.3.3 and the topographic map has been simplified using the contrasted sketch presented in the previous section. In the last example (the van Eyck painting) all shapes from the original image are kept.

Single shape renderings result in abstract compositions in which the movement, as well as part of the semantic of the original image, are readily visible. In short, only the scales of shapes and their spatial relationships are displayed. From this point of view, these experiments are intriguing perception experiments, in which one may be surprised to perceive the meaning of the depicted scene. In Figure 5.1, a similar experiment is displayed using elongated rectangles with fixed elongation. In Figures 5.6, bottom left, and 5.7 right, the dictionary is made of ellipses. In contrast with the previous experiment, the elongation of ellipses can be set to match the replaced shape. Its is interesting, from a perceptual point of view, to observe how much visual information is gained by switching from disks to ellipses with varying elongation. More results are referred to Figures 5.8, Figures 5.9 and Figures 5.10.
Rendering from a more complex dictionary  Next, we show some examples in which the dictionary is made of more complex shapes. In Figure 5.12, the dictionary is made of hand-drawn shapes. In Figure 5.11 the dictionary is made of shapes obtained using a commercial drawing software. In these two examples, the topographic map has been simplified using the contrasted sketch presented in the previous section. In the second case (the football players), an interactive segmentation procedure has been used to get rid of the background. As explained before, shapes from the dictionary are selected so as to minimize elongation and compactness difference with the replaced shape, following Formula (5.3.1), and then rotated and scaled to match the area and orientation of the replaced shape. The effects of this simple procedure may be observed in the heads and wings of the birds in Figure 5.12.

Exchanging shapes and color between images  In the experiments of Figure 5.1, right, and 5.11, bottom right, the dictionary is made of the shapes from an auxiliary image, in both cases a painting from Kandinsky. That is, the result is obtained by sending the geometry of these auxiliary images onto the structure of the input image. In both experiments, the distance function used to select shapes, Equation 5.3.1, is enriched with the difference of the colors of the shapes, as

\[ d(S, D) = |e(S) - e(D)| + |\kappa(S) - \kappa(D)| + \sum_{c \in \{R,G,B\}} |c(S) - c(D)|. \]  

(5.3.2)

5.3.3 Painting-like effects

In this section, we illustrate the flexibility of the general framework presented above, by showing that it permits to obtain some painting-like effects by modifying the shapes of the colored tree of images. Contrarily to some classical rendering approaches [76, 94], this effect is obtained without using any explicit brush model and only involves shape modifications.

Shape shaking  Camera blur in photographs usually results in many nested level lines along contours [37]. We suggest to use this redundancy to produce oscillating boundaries, as may be encountered in oil-paintings. This effect is achieved by adding a random displacement to each shape of the colored tree \( T = (S_i, c_i) \) of the input image, thus mimicking small oscillations produced by the painting tool. The random displacement is made of a rotation whose angle is a uniform random variable between two angles \(-\Delta \theta \) and \(\Delta \theta\) and a translation according to a vector whose norm is uniform between 0 and \(\Delta r\). The orientation of this vector can either be uniform between 0 and \(2\pi\) (incoherent displacement) or coincide with the orientation of the principal axis of the shape (coherent displacement). This second option allows for better preservation of objects’ boundaries and result in sharper images. The resulting effects can be seen on Figure 5.13, Figure 5.14 and Figure 5.15.

In order to adapt to image content, we shake (rotate and translate) shape according to its dominant direction, as \(\frac{(\Delta x)^2}{\lambda_1^2} + \frac{(\Delta y)^2}{\lambda_2^2} \leq \rho\), where \(\lambda_{1,2}\) are the eigenvalues of the ellipse that share the same second-order moments as the shape and \(\rho\) is the amplitude of the random translation and \(\Delta x, \Delta y\) are random translations. We compare the proposed shape shaking method with an artistic filter, called olify, in the GIMP software [161], where for
Figure 5.5: Different abstract renderings from the same image. **Top:** original image on the left, shape shaking on the right; **Bottom:** single shape rendering with ellipses (on the left) and with rectangles (on the right). High-resolution images can be found at the address [221].

each R/G/B channel at a pixel \((x, y)\), the pixel value is reset to be the most commonly occurring R/G/B value in a \(n \times n\) square centered on \((x, y)\). The comparisons between the proposed method and olify are given in Figure 5.13, Figure 5.14 and Figure 5.15.

**Shape smoothing** The second painting-like effect proposed in the same framework aims at producing images mimicking water-color paintings. A median filter is applied individually to each shape of the color tree of the input image. The resulting effects can be seen on Figure 5.13, Figure 5.14 and Figure 5.15. This results in boundary motion, the motion being more important at points of high curvature. This approach shares some similarity with recent works where it has been suggested that interesting painting-like effects can be obtained by applying some operators from mathematical morphology, such as the sequential application openings and closings [25].

The median filter on a binary shape \(S\) can be implemented quickly as described in Algorithm 4.

**Relief Effects** Relief effects on 2D images are usually achieved by using some 3D models, as in bump mapping [22]. In the framework of this chapter, a different relief effect can be achieved by assigning a shadow to each shape. For each shape \(S_i\) with color \(c_i\), a slightly translated version of the same shape is added in the tree as the parent of \(S_i\), with
5.3. A Flexible Image Manipulation Framework

Figure 5.6: Shape selection and single shape rendering. **Top:** original image on the left, image obtained by replacing selected shapes with disks on the right. **Bottom:** replacing selected shapes with ellipses on the left and with rectangles on the right. High-resolution images can be found at the address [221].

Algorithm 4: median filter on a shape $S$

**Input:** a shape $S = \{(x_k, y_k)\}_{k=1}^K$ and the window size $2t + 1$;  
**Output:** smoothed shape $S' = \{(x_k, y_k)\}_{k=1}^M$;

```
begin
    foreach $1 \leq k \leq K$ do
        $msum \leftarrow 0$;
        foreach $x_k - t \leq x \leq x_k + t$ do
            foreach $y_k - t \leq y \leq y_k + t$ do
                $msum \leftarrow msum + 1$;
            if $msum > 2t^2 + 2t$ then
                $S' \leftarrow (x_k, y_k)$

```

A darker color $k.c_i, 0 \leq k < 1$. This effect is used to render the simple geometric shapes of Figures 5.1, 5.6 and 5.11.
Figure 5.7: Single shape rendering. Top: original images. Middle: all shapes are replaced with disks. Bottom: all shapes are replaced with ellipses. High-resolution images can be found at the address [221].
Figure 5.8: Single shape rendering. **Top:** original image. **Bottom:** all shapes are replaced with a single drop shape. High-resolution images can be found at the address [221].
5. A Generic Structured Image Manipulation Framework

(a) original image

(b) all shapes are replaced with circles

Figure 5.9: Single shape rendering.
5.4 Comparison with Arty Shapes

The image abstraction approach most similar to ours is *arty shapes*, proposed by Song et al. [195]. It starts by segmenting an image into disjoint regions of interest, using an image segmentation algorithm, e.g. normalized cut [189]. Observe that the regions in a segmentation usually correspond to image details at one main scale, see Figure 5.16 (a). However, the authors of Arty Shapes want an abstraction method that “both preserves an appropriate amount of detail while keeping the abstractness” [195]. In order to enrich the scales of regions, several segmentations of the image, obtained by using different parameters, are used to provide different levels of details. For instance, in Figure 5.16 (a), for the bird image, two segmentations with respectively 5 regions and 120 regions are used. It then fits each region with several kinds of shapes, such as circle, rectangle, triangle and convex hull. The shape types are selected by relying on a classifier trained from some human labeled data. It finally renders shapes according to their fitting error as,

\[
\frac{|S \cap R|}{|S \cup R|},
\]

where \( S \) and \( R \) are the shapes fitted and the corresponding region. More precisely, shapes with large fitting errors are rendered before those with smaller errors. Figure 5.16 illustrates the abstract results of fitting circles, convex hulls and ellipse respectively. Observe that according to the rendering order, small shapes with large fitting errors are rendered before...
Figure 5.10: Single shape rendering. **Top:** original image. **Bottom:** all shapes are replaced with circles. High-resolution images can be found at the address [221].

bigger shapes with smaller errors. It happens when rendering the bird with circles, where
5.4. Comparison with Arty Shapes

Figure 5.11: Shape transfer from a dictionary. Top (left to right): original image, first dictionary and second dictionary. Bottom (left to right): result of shape transfer using the shapes from the first dictionary, then from the second dictionary. High-resolution images can be found at the address [221].

the body of the bird is hidden behind the background.

Compared with arty shapes, Figure 5.17 illustrates the abstraction of the bird in Figure 5.16 using our method. Figure 5.17 (a) displays the abstract results of fitting circles, rectangles and ellipses respectively. Figure 5.17 (b) shows the rendering result from a three-shape dictionary where shapes are selected according to Equation (5.3.1). For all the results, shapes are rendered in the order described by Algorithm 3. The advantages of the proposed framework over arty shapes are:

1. It abstracts an image but provides more details at multiple scales without losing image structures, see the shapes on the nest in Figure 5.17 for instance.

2. It organizes all shapes in a tree structure, driven by inclusion of shapes\(^2\). The rendering order based on both the tree structure and the compactness of shapes handles the inclusion and occlusion of shapes better than arty shapes in our experiments.

\(^2\)The simplification of these shapes breaks the inclusion relationship between them and converts it to occlusion.
5. A Generic Structured Image Manipulation Framework

Figure 5.12: Shape transfer from a dictionary. **Top**: original image on the left, dictionary on the right. **Bottom**: shape transfer. High-resolution images can be found at the address [221].

5.5 Image Filtering

This section briefly shows how the proposed framework can be used to simplify images, removing texture parts and noise. The proposed filtering is also based on the fact that the camera blur in images usually results in many nested level lines along contours [37]. For a
Figure 5.13: Examples of painting-like effects on a landscape.

shape $S$ on the tree $T$, let us define the scale ratio to be

$$\alpha(S) = \frac{|S|}{|ST|},$$  \hspace{1cm} (5.5.1)
Figure 5.13: (cont.) Examples of painting-like effects on a landscape. High-resolution images can be found at the address [221].
5.5. Image Filtering

(a) original image

(b) shape smoothing

Figure 5.14: Examples of painting-like effects on a landscape.
Figure 5.14: (cont.) Examples of painting-like effects on a landscape. High-resolution images can be found at the address [221].
where $S^f$ is the father shape of $S$ and $| \cdot |$ is the area of the shape. This is actually the order one scale ratio as already introduced in Section 2.3.2 and used for texture analysis.
Obviously, $0 < \alpha(S) < 1$. Observe that the scale ratio of the shape whose parent is the
5.5. Image Filtering

(a) A bird image and its segmentations

(b) Abstract results of fitting circles, convex hulls and ellipses respectively.

Figure 5.16: An example of the method *arty shapes* [195]. (a) displays a bird image and two segmentations with 5 and 120 regions respectively. (b) illustrates the abstract results of fitting circles, convex hulls and ellipse using the method of [195], from left to right.

background may be small. Thus, the scale ratio of a shape $S$ is revised as

$$\gamma(S) = \max\{\alpha(S), \alpha(S_1^c), \ldots, \alpha(S_K^c)\}, \quad (5.5.2)$$

where $S_1^c, \ldots, S_K^c$ are the children of shape $S$. $\gamma$'s are large on shapes of objects and are small on those of noises. The filtering of an image can be done by suppressing all the shapes with $\gamma$ smaller than a threshold $\delta$. In order to preserve very big shapes and to
5. A Generic Structured Image Manipulation Framework

(a) Abstract results of fitting circles, rectangles and ellipse respectively, using our method.

(b) Rendering from a three-shape dictionary (left).

Figure 5.17: Rendering the bird with framework proposed in this chapter. (a) display the abstract results of fitting circles, rectangles and ellipse respectively. (b) illustrates the rendering results from a three-shape dictionary (left). The shape selection is performed according to Equation (5.3.1).

Avoid artefacts in zone with slow gray level variations, only shapes of size smaller than a value $a_{\text{max}}$ are taken into account for the filtering. The detailed algorithm is given in Algorithm 5.

Notice that the proposed scale-ratio filter is related to connectivity-based attribute filtering in mathematical morphology [180, 155], which removes connected components
Algorithm 5: Scale-ratio filter

**Input:** An image $I$ and a threshold $\delta$

**Output:** Filtered image $\tilde{I}$;

begin

- Decompose the image $I$ into a tree of shapes $T = (S_i)_{i=1,\ldots,N}$, using FLST;
- repeat
  - $numR = 0$;
  - foreach shape $S_i$ on the tree $T$ do
    - Compute the scale-ratio $\gamma(S_i)$ on $T$;
    - if $\gamma(S_i) < \delta$ then
      - Remove the shape $S_i$ from $T$;
      - Update the tree $T$, by connecting the children of $S_i$ to its parent $S'_i$;
      - $numR = numR + 1$;
  - until $numR == 0$;
- Reconstruct an image $\tilde{I}$ from the new tree of shapes $T$.

end

of images according to their attributes, such as area, elongation and compactness. The grain filter [33] is an example of attribute filter, which suppresses shapes with size smaller than a given threshold $a_g$ on the topographic map. In contrast with attribute filter, one advantage of the proposed filter is its independence to local affine transformations, which enables us to filter images with perspective and other geometric deformations. The other advantage is that by relying on the scale ratio, it is able to identify noisy structures as isolated structures in a larger shape.

Figure 5.19 and Figure 5.20 illustrate the grain filtering [33] and the scale-ratio filtering on the bateaux image in Figure 5.18a which contains rich scale information. Both filters are performed with different parameters. In contrast with the scale-ratio filter, which smooths textures but keeps objects, the grain filter removes some objects in the original image, since it depends on the scale of objects while the scale-ratio filter is scale invariant for all shapes smaller than $a_{\text{max}}$.

Figure 5.23 illustrates the performance of the scale-ratio filter on an MRI image in Figure 5.18b, where more structural details are expected to be kept when texture details are suppressed. For comparison, the results of grain filter are displayed in Figure 5.22. Both the two methods are performed with different parameters. The maximal size $a_{\text{max}}$ in the scale-ratio filter is set to 1000. Observe how the scale-ratio filter keeps the structural details of the images, benefiting from its scale invariant property.

For color images, the scale-ratio filter is applied to the R, G, B channels respectively. Figure 5.24 shows an example of the scale-ratio filter on an image with JPEG compression artefacts. Observe that while removing some noise and compression artefacts, it keeps the geometrical structures, for instance the T-junctions, in the image.

Figure 5.26 and Figure 5.25 display and compare the grain filtering and scale-ratio filtering on an image with additive Gaussian noise, shown in Figure 5.21. The scale ratio filter seems rather efficient in this case, although some more comparisons with classical methods is clearly needed.
Figure 5.18: Two images used for simplification. The bateaux image is downloaded from the address http://perso.telecom-paristech.fr/~delon/Master_MVA/ and the MRI image is download from the address: http://wsunews.wsu.edu/Content/Publications/MRI1.jpg.

Figure 5.28, Figure 5.27 and Figure 5.29 compare the grain filtering, scale-ratio filtering and median filtering on an impulse noise image in Figure 5.21. The filtering experiments are performed with different parameters. Observe that the scale-ratio filter can remove most of the impulse noises as well as keeping details in the images. However, to achieve the same goal, the grain filter has to use bigger scale, which suppress some image details. It is worth noticing that both these two methods do not produce new level lines in the image. In contrast, the median filter provides smoother results, but most of the details in the image are removed.

5.6 Discussion

In this chapter, we have first shown that a highly structured representation of images, the topographic map, enables the creation of geometrical abstractions from digital images. Now, other complex image analysis tools have been proposed in the field of NPR, as recalled in Section 5.1.1. In particular, the only work we are aware of showing the possibility to achieve the geometrical abstraction presented in this chapter is the work of Song et al. [195]. As explained earlier this work suggests to use a normalized-cut segmentation to allow for shape replacement. The mosaic structure of such a segmentation aims at producing a partition of the image. The approach of Song et al. also relies on some scale parameter to fix the level of detail that is kept, even though a method is given to merge two different scales. These two aspects yield a loss of structure. In contrast, the topographic map consists in a hierarchical structure, driven by the inclusion of shapes and accounting for all scales in
5.6. Discussion

(a) Results of grain filtering with \( a_g \) as 10.

(b) Results of grain filtering with \( a_g \) as 20.

(c) Results of grain filtering with \( a_g \) as 40.

Figure 5.19: Filtering the image in (a) by grain filter [33]. The filtered results are on the left and the differences between the original image and the filtered one are on the right. Observe how the details are removed from the image. For instance, the arches of the bridge and the bird in the sky are almost removed away at the scale of 40.
(d) Results of grain filtering with $a_g$ as 80.

(e) Results of grain filtering with $a_g$ as 100.

(f) Results of grain filtering with $a_g$ as 200.

Figure 5.19: (cont.) Filtering the image in (a) by grain filter [33]. The filtered results are on the left and the differences between the original image and the filtered one are on the right. Observe how the details are removed from the image. For instance, the arches of the bridge and the bird in the sky are almost removed away at the scale of $40$. High-resolution images can be found at the address [221].
5.6. Discussion

(a) Results of scale-ratio filtering with the threshold $\delta = 0.70$ and $a_{max} = 1000$.

(b) Results of scale-ratio filtering with the threshold $\delta = 0.75$ and $a_{max} = 1000$.

(c) Results of scale-ratio filtering with the threshold $\delta = 0.80$ and $a_{max} = 1000$.

Figure 5.20: Filtering the image (a) by the proposed scale-ratio filter. The filtered results are on the left and the differences between the original image and the filtered one are on the right. Observe how the small objects are kept while the noisy details are removed from the image. For instance, the arches of the bridge, the bird in the sky, and the small objects on the boats are well kept, while the sea surface is smoothed.
Figure 5.20: (cont.) Filtering the image (a) by the proposed scale-ratio filter. The filtered results are on the left and the differences between the original image and the filtered one are on the right. Observe how the small objects are kept while the noisy details are removed from the image. For instance, the arches of the bridge, the bird in the sky, and the small objects on the boats are well kept, while the sea surface is smoothed. High-resolution images can be found at the address [221].
the image. As a result, the synthesis procedure proposed in this chapter enables complex interactions between shapes as may be seen in Figure 5.7 and 5.8. Further hierarchical image representations have been proposed for NPR. For example, the authors of [92] make use of a linear scale-space to drive stroke placements in stroke-based rendering and De Carlo et al. [55] use a multi-scale structure to obtain line-drawing style abstraction of images. The authors of [154] also advocate the importance of image structure for the emulation of various artistic styles, through the use of multi-scale edges and Poisson interpolation. We therefore believe that the generic image representation tool advocated for in this chapter, the topographic map, could be of use for other abstract renderings of digital photographs.

The second contribution of this chapter is to propose the scale-ratio filter, which suppresses image structures according to the scale ratio between shapes and their parents along the topographic map. Compared with existing connected filters, such as grain filter, the proposed scale-ratio filter is affine invariant to geometric changes. A first application of this filter is to remove noises from images with perspective changes. It can also be used to filter images with structures at a large range of scales, where scale invariant is demanded, as one expects both removing noises from the images as well as keeping structural details at different scales.
(a) Results of grain filtering with $a_g$ as 10, 20, 40, 60.

(b) The differences between the original image and the filtered ones in (a)

(c) Results of grain filtering with $a_g$ as 80, 100, 200, 300.

(d) The differences between the original image and the filtered ones in (c)

Figure 5.22: Grain filtering [33] of the MRI image in Figure 5.18b. High-resolution images can be found at the address [221].
5.6. Discussion

(a) Scale-ratio filtering with the threshold $\delta$ as 0.7, 0.75, 0.8, 0.85 and $a_{max} = 1000$.

(b) The differences between the original image and the filtered ones in (a).

(c) Results of scale-ratio filtering with the threshold $\delta$ as 0.9, 0.92, 0.95, 0.97 and $a_{max} = 1000$.

(d) The differences between the original image and the filtered ones in (c).

Figure 5.23: Scale-ratio filtering of the MRI image in Figure 5.18b. Observe how the filter keeps the details of the image. High-resolution images can be found at the address [221].
Figure 5.24: Filtering of a color image. The first row shows an image of dauphin and its subpart. The second row displays the scale-ratio filtered with $\delta = 0.9$ and $a_{\text{max}} = 10000$. The last row displays the difference between the original image and the filtered one. Observe that the textures as well as the JPEG compression effects are removed while the geometries, for instance T-junctions, are well kept. High-resolution images can be found at the address [221].
5.6. Discussion

Figure 5.25: Grain filtering of the Gaussian noise degraded image in Figure 5.21, with different parameters. High-resolution images can be found at the address [221].
Figure 5.26: Scale-ratio filtering of the Gaussian noise degraded image in Figure 5.21, with different parameters. High-resolution images can be found at the address [221].
5.6. Discussion

(a) Grain filtering with $a_g = 10$

(b) Grain filtering with $a_g = 40$

(c) Grain filtering with $a_g = 100$

(d) Grain filtering with $a_g = 200$

Figure 5.27: Grain filtering of the impulse noise degraded image in Figure 5.21, with different parameters. High-resolution images can be found at the address [221].
5. A Generic Structured Image Manipulation Framework

(a) Results of scale-ratio filtering with \( \delta = 0.8 \) and \( a_{\text{max}} = 1000 \).

(b) Results of scale-ratio filtering with \( \delta = 0.9 \) and \( a_{\text{max}} = 1000 \).

(c) Results of scale-ratio filtering with \( \delta = 0.95 \) and \( a_{\text{max}} = 1000 \).

(d) Results of scale-ratio filtering with \( \delta = 0.98 \) and \( a_{\text{max}} = 1000 \).

Figure 5.28: Scale-ratio filtering of the impulse noise degraded image in Figure 5.21, with different parameters. High-resolution images can be found at the address [221].
Figure 5.29: Median filtering of the impulse noise degraded image in Figure 5.21, with different half size window $s$. High-resolution images can be found at the address [221].
Part III

Meaningful Junction Detection
Chapter 6

Definition and Detection of Junctions

6.1 Introduction

Junctions, i.e. points in images where edges join or intersect, play an important role in vision and are frequently used as basic features in various tasks of computer vision and image understanding. However, the detection of junctions is a longstanding and partly open problem.

Let us precise straight away that we define junctions in our work as points in images around which there are at least two dominant and different edge directions. Among these junctions, we can distinguish in particular L-, Y- and X-junctions, depending on the number of directions they contain.¹

Psychological studies [7, 104] have suggested that, as features in low level vision, junctions play an important role in visual perception. They have been widely applied to many vision tasks at different levels. Since junctions reveal important occlusion relationships in images, they have been involved in figure/ground separation [74, 167] as well as surface completion [174]. Junctions have been also used for grouping edges and regions to achieve image segmentations [73, 122, 98], by regarding them as intersections of image boundaries or regions. The importance of junctions in perception of brightness and transparency have been investigated by [136, 1]. Using junctions as critical features for motion analysis have been suggested by [2, 134]. The role of junction in object recognition have been studied in [81, 23, 119].

6.1.1 Previous and related works (part 1)

This section gives a brief review of the related works on junction detection. As junctions are of great interest for many vision tasks spanning from low level to high level, the detection

¹It’s worth noticing that, in the literature, the terminology “corners detection” overlaps with the terminology “interest points detection”, which refers to the detection of interest points in images used for understanding tasks. Observe that both corners and interest points are generally thought as robustly detectable points that are stable under some local and global perturbations. The ambiguity between those two terminologies arises from the fact that early corner detectors [88, 120] are sensitive not only to corners in particular, but also to other kinds of salient location in images. Thus corner detectors are usually employed as interest points detectors. In fact, if only real corners (i.e. L-junctions with our terminology) are to be detected a local analysis is often required [156, 178, 192].
has been a very active research field over the last four decades. An exhaustive study of this field is of course beyond the scope of this chapter. Some reviews on corners detection (in the sense of keypoints) can be found in [193, 185, 173]. In what follows, we concentrate on methods that detect real junctions.

Corners can be taken as points which are not self-similar in an image. The measurement of the self-similarity of a point usually involves several points around it, each of which is associated with a small image patch. Moravec [147] suggested to measure the similarity of two points by using the sum of squared differences (SSD) between their associated image patches. The self-similarity of a point is then defined as the smallest SSD between the point and its neighbors (horizontal, vertical and on the two diagonals). Harris and Stephens [88] then proposed to approximate the SSD by computing the autocorrelation in a patch over the image domain. The resulted cornerness measure is isotropic and has an analytic expansion, called the Harris matrix. A large number of detectors rely on the same idea and the detection of corners boils down in these works to analyze the eigenvalues of the matrix, see e.g. [69, 190, 109, 106]. An alternative measurement of self-similarity is the univalue segment assimilating nucleus (USAN) [193], which is the proportion of pixels in a disk that have intensity values similar to those of the nucleus. Corners are detected as the smallest USAN (SUSAN) points in images. The USAN measure has been also computed by using a circle instead of a disk [202, 172, 173].

A second popular and efficient way to detect corners relies on edges in images. In these kinds of approaches, corners are defined as points in the image where a boundary changes direction rapidly. Thus, a straightforward measurement of the cornerness is the curvature and points with high curvature are taken as corners. Many works have concentrated on different and efficient ways to compute the curvature of curves for detecting corners, see e.g. [171, 199, 165, 212, 142, 140]. In a different way, Beymer [19] proposed a junction detector by extending disconnected edges and filling gaps at junctions by controlling the image gradient. Junction detection can also rely on grouping edges in the neighborhood of a candidate junction [71, 170, 158]. Other approaches use edge detection followed by heuristic grouping of edges to form junctions [217, 129]. In particular, the work of Maire et al. [129] relies on a recent contour detector, named global probability of boundary (gPb), whose parameters are learned on the Berkeley human-annotated database. The junction detection then consists in finding the intersection points of the resulting contours, taking into account both contour salience and geometric configuration. This method yields good performance on the same human annotated benchmark [129]. We will also investigate the behavior of our approach on this database.

Among all existing approaches, the model-based or template-based ones are the most suitable for junction detection (real corners not only interest points). In [56], Deriche and Blasszka present computational approaches for a model-based detection of junctions. A junction model is a 2D intensity function of position with a set of parameters describing the angles of the segments, the grey-level intensities and a blur factor. Some points with poor locations are detected by a preliminarily corner detector (for instance the Hessian determinant [17]), and are used as junction candidates. The model is then constructed with the candidates and a set of initial parameters and is subsequently optimized to pursue precise junctions. Parida et al. [156] suggest a region-based model for simultaneously detecting, classifying and reconstructing junctions. A junction is regarded as an image region
which contains piecewise constant wedges joining at the central point. Their work then relies on a template deformation framework and uses minimum description length principle and dynamic programming to obtain the optimal parameters describing the model. This work also involves junction candidates provided by other preliminarily corner detector, say, SUSAN. Following the model of Parida, Cazorla and Escolano [40, 41] propose a region-based and an edge-based model for junction classification by using Bayesian methods. The region-based one formalizes junction detection as a radial image segmentation problem under region competition framework and the edge-based one detects junctions as radial edges minimizing some Bayesian classification error. Ruzon and Tomasi [178] model junctions as points in images where two or more image regions meet. Regions are described by their color distributions, which allows textured objects with the same mean color to be distinguished. More recently, the work of Sinzinger [192] first detects a set of junction candidates by using a preliminary detector (for instance the Harris detector) and then refines those candidates by relying on some complicated radial edge energy.

Figure 6.1: Are detected junctions meaningful? This figure displays the top 350 of the Harris’ corners (in the top-right) and of the Kovesi’s corners [109] (a contrast invariant detector, shown on the bottom-left) of a natural image (on the top-left). The color from blue to red indicates the significance of the corners from low to high. Observe that both operators have large responses on false corners, such as points over the tree on the mid-right of the image. Bottom-right: result of our a contrario junction detector, which gives large responses on geometric structures but don’t detect many junctions in texture.
6.1.2 Motivations

As mentioned before, there are numerous works on junction detection, but most of them look for “interest points” and few of them detect only structured junctions (see Figure 6.1 for an example). In the present chapter, we aim at developing a junction detection algorithm meeting the following requirements:

- **contrast robustness**: many junction detection algorithms (like for instance the Harris corner detector) heavily rely on contrast. This results in the detection of many false corners along highly contrasted image boundaries (see for instance Figure 6.1). On the contrary, the approach developed in this chapter will ensure that junctions not only appear at very contrasted locations but also emerges in area with low illumination.

- **decision criterion**: junction detectors generally do not provide decision criteria for thresholding automatically corner responses, and detect many spurious junctions if these thresholds are not carefully chosen. In this chapter, this decision problem is handled thanks to the *a contrario* detection theory, introduced by Desolneux *et al.* [58], which enable us to detect “meaningful” junctions in natural images while controlling the number of false detections.

- **automatic scale detection**: in order to identify the characteristic scale of a corner, corner detectors usually make use of the linear scale-space [120, 185, 126]. The shortcomings of this kind of approaches is that they quickly lose precision both in location and scale. Alternatively, we will show that identifying the scale of a junction can be achieved easily in our framework by comparing their meaningfulness at different scales. Additionally, we will be able to identify junction branches (both their number and their locations) and the junction location thanks to the same principle.

6.1.3 Previous and related works (part 2: invariances and scale selection)

Several approaches have been developed in the literature in order to fulfill one or several of the previous requirements.

To achieve contrast invariance, Kovesi [109] uses phase congruency to derive corner-ness measurement. In this work, the image gradient is normalized in different and small wedges. An alternative way to achieve contrast invariant is to rely on some morphological representations. For instance, Alvarez *et. al.* [3] analyze junctions in images using an affine morphological scale space. Caselles *et. al.* [35] propose a method based on level-set representation, which detects junctions as points where two level lines meet. Finally, Cao [30] detects junctions as breaking points under a good continuation criterion on image level lines. This last work shares similarities with the method presented in this chapter, as both are contrast invariant and use an *a contrario* framework.

For scale detection, most of the classical approaches rely on the linear scale space. For instance, the Harris-Laplace [138] detector combines the Harris operator with the idea of linear scale-space, applying the Harris corner detector at multiple scales and then
6.2. Contrast Invariant Junctions

The aim of this section is to define junctions in discrete images and to associate to each candidate junction a strength designed to be robust to local contrast changes. This robustness is obtained through the use of a locally normalized gradient, defined in Section 6.2.1. The strength of junctions will be used in Section 6.3 to decide whether junctions are meaningful or not in a given image.

6.2.1 Local normalization of the gradient

Let us start with some definitions and vocabulary that will be used throughout the chapter. A discrete image is a function $I : \Omega \rightarrow \mathbb{R}$, where $\Omega$ is a rectangular subset of $\mathbb{Z} \times \mathbb{Z}$. Let us denote by $\nabla I = (I_x, I_y)$ the discrete gradient of the image $I$. For a pixel $q$ in $\Omega$, we note $\varphi(q) = \arctan \frac{I_y(q)}{I_x(q)}$ the phase of the gradient ($\varphi(q) = \pi/2$ when $I_x(q) = 0$). We define the direction of the pixel $q$ as $\phi(q) = \left(\frac{\pi}{2} + \varphi(q)\right)$ modulo(2$\pi$).

In order to be robust to local contrast changes, we locally normalize the gradient at
\( \mathbf{q} = (x, y) \) and define \( \hat{\nabla} I = (\hat{I}_x, \hat{I}_y) \) as
\[
\hat{I}_x(\mathbf{q}) = \frac{I_x(\mathbf{q})}{\langle \sqrt{I_x^2 + I_y^2} \rangle_{\mathcal{N}_q}} \quad \text{and} \quad \hat{I}_y(\mathbf{q}) = \frac{I_y(\mathbf{q})}{\langle \sqrt{I_x^2 + I_y^2} \rangle_{\mathcal{N}_q}},
\]
(6.2.1)
where \( \mathcal{N}_q \) is a small neighborhood around \( \mathbf{q} \) and \( \langle \cdot \rangle_{\mathcal{N}_q} \) is the average operator on this neighborhood. The resulting gradient is robust to contrast changes that can be approximated by affine transformations on each neighborhood \( \mathcal{N}_q \). An example of the gradient and modified gradient of an image, obtained with a square neighborhood of size \( 5 \times 5 \) around each pixel, is shown in Figure 6.2. We note \( \| \hat{\nabla} I \| = \sqrt{\hat{I}_x^2 + \hat{I}_y^2} \) the norm of this modified gradient. Observe that its phase is the same as the phase of the usual gradient.

Figure 6.2: Gradient \( \nabla I \) and modified gradient \( \hat{\nabla} I \) of an image patch

Now, let us define the normalized gradient
\[
\tilde{I}_x = \frac{\hat{I}_x - \mu_x}{\sigma_x} \quad \text{and} \quad \tilde{I}_y = \frac{\hat{I}_y - \mu_y}{\sigma_y},
\]
(6.2.2)
where \( \mu_x \) (resp. \( \mu_y \)) and \( \sigma_x \) (resp. \( \sigma_y \)) are the empirical mean and standard deviation of \( \hat{I}_x \) (resp. \( \hat{I}_y \)) on the image. The norm \( \| \tilde{\nabla} I \| = \sqrt{\tilde{I}_x^2 + \tilde{I}_y^2} \) of this normalized gradient will be used in the following paragraphs to define the strength of a junction. As we will see, the distribution of this norm in natural images is well approximated by a standard Rayleigh distribution and we will take advantage of this statistical property in Section 6.3 to detect meaningful junctions.

### 6.2.2 Discrete junctions

A junction is defined as a discrete image structure \( j : \{ \mathbf{p}, r, \{ \theta_m \}_{m=1}^M \} \), characterized by its center \( \mathbf{p} = (x, y) \) in \( \Omega \), its scale \( r \in \mathbb{N} \) and a set of branch directions \( \{ \theta_1, \ldots, \theta_M \} \) around \( \mathbf{p} \). The number \( M \) of branches in the junction is called the order of the junction: when \( M = 1, 2, 3 \) or \( 4 \), we speak respectively of \( T \), \( L \), \( Y \) or \( X \)-junctions. The grid of pixels \( \Omega \)
being discrete, the set $\mathcal{D}(r)$ of possible directions at a given scale $r$ is also considered as discrete in this chapter and is defined as

$$\mathcal{D}(r) = \left\{ \frac{2\pi k}{K(r)} : k \in \{1, \ldots K(r)\} \right\},$$

(6.2.3)

where $K(r)$ is the size of $\mathcal{D}(r)$ \(^2\). Observe that the discrete set $\mathcal{D}(r)$ is periodic, in the sense that its last element is the left neighbor of its first element.

For a given scale $r$ and a given direction $\theta$ in $\mathcal{D}(r)$, we define the branch of direction $\theta$ at $p$ as the disk sector

$$S_p(r, \theta) := \{ q \in \Omega; q \neq p, \| \vec{pq} \| \leq r, d_{2\pi}(\alpha(\vec{pq}), \theta) \leq \Delta(r) \},$$

(6.2.4)

where $\Delta(r)$ is a precision parameter, $\alpha(\vec{pq})$ is the angle of the vector $\vec{pq}$ in $[0, 2\pi]$ and where $d_{2\pi}$ is the distance along the unit circle, defined as $d_{2\pi}(\alpha, \beta) = \min(|\alpha - \beta|, 2\pi - |\alpha - \beta|)$. We note $\mathcal{J}(r, \theta)$ the size in pixels of a sector of direction $\theta$ at scale $r$. This size depends on the scale $r$ but also slightly changes with the direction $\theta$ since this area in pixels is computed on a discrete grid.

Finally, we require that two branches of a given junction do not intersect. It follows that the angle between two directions in a junction $\mathcal{J}$ must be larger than $2\Delta(r)$. As a consequence, the number of possible directions in a junction at scale $r$ is smaller than $\frac{\pi}{\Delta(r)}$. An example of a junction with three branches is shown on Figure 6.3.

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{junction.png}
\caption{A junction with three branches.}
\end{figure}

\subsection*{6.2.3 Junction strength}

Since we want to detect if the branches of a junction correspond to edges in the image $I$, it is reasonable to rely on gradient information to define the strength of a branch.

\footnote{The relation between the number of directions $K(r)$ and the scale $r$ will be made explicit in Section 6.4.}
Definition 1 (Strength of a branch). Let \( p \) be a pixel in \( \Omega \), \( r \) a positive scale in \( \mathbb{N} \setminus \{0\} \) and \( \theta \) a direction in \( D(r) \) such that the sector \( S_p(r, \theta) \) is included in \( \Omega \). The strength of the branch \( S_p(r, \theta) \) is defined as the quantity

\[
\omega_p(r, \theta) = \sum_{q \in S_p(r, \theta)} \gamma_p(q),
\]

where

\[
\gamma_p(q) = \| \nabla I(q) \| \cdot \max \left( | \cos(\phi(q) - \alpha(pq)) - | \sin(\phi(q) - \alpha(pq)) |, 0 \right).
\]

Observe that the quantity \( \| \nabla I(q) \| \cdot | \cos(\phi(q) - \alpha(pq)) | \) measures the adequacy between the direction of the pixel \( q \) and the direction of the vector \( pq \), while \( \| \nabla I(q) \| \cdot | \sin(\phi(q) - \alpha(pq)) | \) measures the adequacy between the direction of the pixel \( q \) and the direction perpendicular to \( pq \). The larger \( \omega_p(r, \theta) \) is, the more likely it is that the branch \( S_p(r, \theta) \) corresponds to an edge. Figure 6.4 (c) shows the values of \( \gamma_p(q) \) for the image patch shown in Figure 6.4 (a).

The strength of a junction is derived from the strength of its branches.

Definition 2 (Strength of a junction). Let \( p \) be a pixel in \( \Omega \), \( r \) a positive scale in \( \mathbb{N} \setminus \{0\} \) and \( \{ \theta_m \}_{m=1}^M \) a set of branch directions in \( D(r) \). Assume that the branches \( S_p(r, \theta_m) \) are pairwise disjoint and are all included in \( \Omega \). Then, the strength of the junction \( j : \{ p, r, \{ \theta_m \}_{m=1}^M \} \) is defined as

\[
t(j) := \min_{m=1}^M \omega_p(r, \theta_m).
\]

Starting from this definition, a first naive algorithm of junction detection can be developed. The idea is to detect, for a fixed scale \( r \) and a given threshold \( t \), all the junctions in \( I \) with a strength greater than \( t \). In practice, testing all possible junctions for every point \( p \) in \( \Omega \) is computationally heavy. Among all the potential branches at a given point \( p \), we restrict ourselves to the directions \( \theta \) in the discrete set \( D(r) \) where the periodic function \( \omega_p(r, \theta) \) reaches a local maximum. Moreover, in order to respect the requirement that two branches of a junction should not intersect, we only keep locations \( \theta \) such that \( \omega_p(r, \theta) \) is greater than all \( \omega_p(r, \theta') \) when \( \theta' \) spans \( D(r) \cap [\theta - \Delta(r), \theta + \Delta(r)] \). The set of these semi-local maxima can be computed quickly, for instance by using a non-maximum suppression (NMS) procedure (see [107, 150]) on the function \( \omega_p(r, \cdot) \) on \( D(r) \). In practice, if two semi-local maxima are equal and located at a distance smaller than \( \Delta(r) \), one of them is discarded.

An additional precaution is required in the case \( M = 2 \): in order to avoid detecting all edge points as \( L \)-junctions, we check that these junctions are not detected with two opposite angles.

The overall detection algorithm is summarized in the Algorithm 6.

The first parts of the procedure are illustrated by Figure 6.4. In this example, the pixel \( p \) is chosen as the center of the image 6.4 (a). Figure 6.4 (d) shows in blue the values of \( \omega_p(r, \theta) \) when the angle \( \theta \) spans the periodic set \( D(r) \) and shows in red the semi-local maxima kept after the NMS procedure (step (2) of the loop in the Algorithm 6).
6.2. Contrast Invariant Junctions

Figure 6.4: Computation of junction candidates at scale $r = 12$, when $p$ is the center of the original image (a). The blue curve in (d) shows the strength $\omega_p(r, \theta)$ in function of the direction $\theta$ on the circle. The directions that remain after the NMS procedure are shown in red. Other parameters are set as follows: $K = 75$, and $2\Delta = 0.289$. 
Algorithm 6: Junction detection in an image $I$ at a given scale $r$.

Require: A discrete image $I : \Omega \mapsto \mathbb{R}$, an order $M$, a scale $r \in \mathbb{N} \setminus \{0\}$, a number $K$ of discrete directions, a precision $\Delta$ and a threshold $t$. 

1. compute $\tilde{\nabla}I$ using Equation (6.2.1);
2. for each pixel $p \in \Omega$ do
   3. (1) compute $\omega_p(r, \theta)$ for each $\theta$ in $D(r)$, using Equations (6.2.5) and (6.2.6);
   4. (2) use a NMS procedure to only keep locations $\theta$ such that $\omega_p(r, \theta)$ is larger than all $\omega_p(r, \theta')$ when $\theta'$ spans $D(r) \cap [\theta - \Delta(r), \theta + \Delta(r)]$; call $\Theta$ the set of all these directions;
   5. (3) find all the subsets $\Theta'$ of size $M$ in $\Theta$ such that $\min_{\theta \in \Theta'} \omega_p(r, \theta) \geq t$; for each of these subsets, mark $\{p, r, \Theta'\}$ as a junction if $M \neq 2$, or if $M = 2$ and $\Theta' = \{\theta_1, \theta_2\}$, with $d_2(\theta_1, \theta_2 + \pi) > 2\Delta(r)$.
3. end for
4. return a set of all detected junctions.

Figures 6.4 (e, f, g) represent respectively a candidate L-junction, a candidate Y-junction and a candidate X-junction at $p$.

The main drawback of this detection algorithm is that the threshold $t$ on the junction strength remains the same whatever the junction scale, order and image size. Setting such a threshold globally is not easy and can lead to over-detect at some scales and under-detect at other scales. The goal of the next section is to explain how one can compute automatically detections thresholds that adapt to the junction scale, order and image size. For this purpose, we resort to an a contrario methodology.

6.3 An a contrario Approach for Junction Detection

The a contrario detection theory has been primarily proposed by Desolneux et. al. [58]. This methodology is inspired by geometric grouping laws governing low-level human vision, also known as Gestalt laws [104], and states that meaningful structures in images can be seen as structure which are unlikely to happen under some hypothesis of randomness. The method has been extensively tested and successfully applied to various problems in image processing and computer vision [31, 141, 149, 209, 80, 210]. A complete overview of these methods can be found in [60]. In this section, this methodology is adapted to the detection of meaningful junctions in images.

6.3.1 Null hypothesis

The goal of the following sections is to set detection thresholds on junction strengths in such a way that no junction will be detected in a "generic random image". Let us precise what "generic random image" stands for here. Let $I$ be a random image. For each pixel $q$ in $\Omega$, we note $X(q)$ the random variable corresponding to the value $\|\nabla I(q)\|$ and $Y(q)$ the random variable corresponding to the direction $\phi(q)$. We say that the random variables $\{X(q), Y(q)\}_{q \in \Omega}$ follow the null hypothesis $H_0$ if
1. \( \forall q \in \Omega, Y(q) \) is uniformly distributed over \([0, 2\pi]\);
2. \( \forall q \in \Omega, X(q) \) follows a Rayleigh distribution with parameter 1;
3. the family \( \{X(q), Y(q)\}_{q \in \Omega} \) is made of independent random variables.

Let us make a comment on the second assumption. In [176], Ruderman et al. observe that if we normalize the logarithm of an image intensity by its local mean and standard deviation in a neighborhood \( N_p \) around each pixel, "the histogram of pixel values has Gaussian tails, and the distribution of gradients in the 'variance-normalized' image is almost exactly the Rayleigh distribution". This result can be extended to our modified derivatives \( \hat{I}_x \) and \( \hat{I}_y \): outside of a small neighborhood around 0, their distribution is well approximated by a gaussian distribution. This is illustrated on a particular image \( I \) by Figure 6.5. In this Figure, the empirical distributions of \( \hat{I}_x \) and \( \hat{I}_y \) are drawn in blue, and the best gaussian fits are drawn in red. If we except a peak around 0, this approximation is particularly good. This approximation is of particular interest for junction detection since the gradients around a junction usually take large values. Observe that this gaussian approximation is not valid for the classical gradient \((I_x, I_y)\), whose distribution is quite heavy tailed. It follows that the distributions of \( \hat{I}_x \) and \( \hat{I}_y \) are usually well approximated by standard gaussian distributions and the norm \( \|\nabla I\| \) in natural images approximately follows a central chi-distribution with 2 degrees of freedom (also known as the Rayleigh distribution of parameter 1). This is confirmed experimentally by the example shown on Figure 6.5. Figure 6.6a shows in blue the empirical distributions of \( \|\nabla I\| \) for 50 different natural images, as well as the density of the Rayleigh distribution of parameter 1 (in red).

### 6.3.2 Distribution of \( t(j) \) under \( H_0 \)

Let \( j : \{p, r, \{\theta_m\}_{m=1}^M\} \) be a junction in \( I \) and assume that the normalized gradients and directions of \( I \) follow the null hypothesis \( H_0 \). Then the strengths of the different branches \( S_p(r, \theta_m) \) are independent random variables. Thus, if we note \( t(j) \) the random variable measuring the strength of \( j \), then

\[
\mathbb{P}_{H_0}[t(j) \geq t] = \mathbb{P}_{H_0}[\forall m, \omega_p(r, \theta_m) \geq t] = \prod_{m=1}^{M} \mathbb{P}_{H_0}[\omega_p(r, \theta_m) \geq t].
\]  

(6.3.1)

Now, recall that each \( \omega_p(r, \theta_m) \) is also a sum over \( S_p(r, \theta_m) \) of i.i.d. random variables \( \gamma_p(q) \). For two given points \( p \) and \( q \) in \( \Omega \), the direction \( \alpha(pq) \) is a constant in \([0, 2\pi]\). This implies that under the hypothesis \( H_0 \), the random angle \( \phi(q) - \alpha(pq) \) is still uniformly distributed on \([0, 2\pi]\). As a consequence, each \( \gamma_p(q) \) can be written as a product \( X \cdot \max(|\cos \theta| - |\sin \theta|, 0) \), where \( X \) and \( \theta \) are independent, \( X \) follows a Rayleigh distribution and \( \theta \) is uniformly distributed on \([0, \pi]\). Finally, the distribution of each \( \gamma_p(q) \) under \( H_0 \) can be written (see Appendix A.2 for a proof)

\[
\mu(z) = \frac{1}{2} \delta_0(z) + H(z) \cdot \frac{1}{\sqrt{\pi}} e^{-\frac{z^2}{4}} \cdot \text{erfc}\left(\frac{z}{2}\right)dz,
\]  

(6.3.2)

where \( \delta_0 \) is a Dirac mass at 0, \( H \) is the Heaviside function (\( H(z) = 1, \) for \( z \geq 0, \) and \( H(z) = 0 \) otherwise) and where \( \text{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_{z}^{\infty} e^{-s^2} ds \).
Figure 6.5: Approximations of the distributions of $\hat{I}_x$, $\hat{I}_y$ and $\|\nabla I\|$ (given by Equations (6.2.1)). The blue curves are the empirical histograms and the red curves are the approximated distributions. (e) and (f) are Gaussian distributions and (g) is a Rayleigh distribution of parameter 1.

The empirical distribution of $\gamma_p(q)$ on 50 natural images is displayed in Figure 6.6b, along with its theoretical approximation $\mu$. Under the hypothesis $\mathcal{H}_0$, the law of the sum $\omega_p(r, \theta_m)$ is obtained by convolving $J(r, \theta_m)$ times with itself the distribution $\mu$, where
6.3. An a contrario Approach for Junction Detection

Figure 6.6: Distributions of \( \|\nabla I\| \) (blue curves in (a)) and \( \gamma_p(q) \) (blue curves in (b)) for 50 different natural images. Rayleigh(1) density (red curve in (a)) and \( \mu(z) \) (red curve in (b)) are displayed for comparison.

\[ J(r, \theta_m) \] is the size of a sector of orientation \( \theta_m \) at scale \( r \).

**Proposition 1.** Let \( J : \{p, r, \{\theta_m\}_{m=1}^M\} \) be a junction in \( I \) and suppose that the hypothesis \( H_0 \) is satisfied, then the probability that the random variable \( t(j) \) is larger than a given threshold \( t \) is

\[
F_j(t) := \mathbb{P}_{H_0}[t(j) \geq t] = \prod_{m=1}^{M} \int_{t}^{+\infty} J(r, \theta_m) \star \mu(dz), \quad (6.3.3)
\]

where \( J(r, \theta_m) \) is the size of a sector of orientation \( \theta_m \) at scale \( r \).

The increasing function \( F_j \) does not depend on the position \( p \) of the junction.

### 6.3.3 Meaningful junctions

Thanks to the previous computations, we are now in a position to fix thresholds automatically on junction strengths to decide which junctions should be considered as valid in an image \( I \). This is made possible by thresholding the probability \( (6.3.3) \). In order to control the number of false detections in a given image, this thresholding will take into account the number of possible discrete junctions in the image.

**Number of tests** In this paragraph, we assume that the order \( M \) is fixed, and we call \( J(M) \) the set of all possible junctions of order \( M \) in the discrete image \( I \). The size of \( J(M) \) depends on several parameters: the minimum and maximum authorized scales \( r_{\text{min}} \) and \( r_{\text{max}} \), the size \( N \) of \( I \), the number of directions \( K(r) \) and the precision \( \Delta(r) \) of branches at each scale \( r \). The practical setting of these parameters will be discussed in Section 6.4.2.

At a given location \( p \), once the first branch is chosen among the \( K(r) \) possible directions, the second direction must be chosen in such a way that the two branches of width...
2\Delta(r) do not intersect, which means that only \( K(r)(1 - 2\frac{\Delta(r)}{\pi}) \) directions are authorized. Therefore, at each location \( p \in \Omega \), and for a given scale \( r \), the number of possible junctions of order \( M \) is always smaller than
\[
N_p(r) = \frac{1}{M!} \prod_{m=1}^{M} K(r) \left(1 - 2(m - 1)\frac{\Delta(r)}{\pi}\right). \tag{6.3.4}
\]
It follows that the size of the set \( \mathcal{J}(M) \) can be written
\[
\#\mathcal{J}(M) = \sum_{p \in \Omega} \sum_{r=r_{\text{min}}}^{r_{\text{max}}} N_p(r) = \frac{N}{M!} \sum_{r=r_{\text{min}}}^{r_{\text{max}}} \prod_{m=1}^{M} K(r) \left(1 - 2(m - 1)\frac{\Delta(r)}{\pi}\right). \tag{6.3.5}
\]

\( \epsilon \)-meaningful junctions

The next definition and the following proposition explain how to fix thresholds on junctions strengths in order to control the average number of false detections.

**Definition 3. (\( \epsilon \)-meaningful junction)** Let \( I \) be a discrete image. A junction \( j \) of order \( M \) and scale \( r \) is said to be \( \epsilon \)-meaningful in \( I \) if
\[
\text{NFA}(j) := \#\mathcal{J}(M) \cdot F_j(t(j)) \leq \epsilon. \tag{6.3.6}
\]

The quantity \( \text{NFA}(j) \) is a measure of the meaningfulness of the junction: the smaller it is, the more meaningful the junction \( j \). A junction of order \( M \) and scale \( r \) is detected as \( \epsilon \)-meaningful in \( I \) if its strength \( t(j) \) is larger than the threshold
\[
t(r, \epsilon) := \min\{t; \ F_j(t) \leq \frac{\epsilon}{\#\mathcal{J}(M)}\}. \tag{6.3.7}
\]

Notice that for a fixed \( \epsilon \), this formula yields a different threshold on \( t(j) \) for each value of the scale \( r \). The value \( \epsilon \) is easy to interpret: it corresponds to an expected number of false detections in \( I \).

**Proposition 2.** Let \( I \) be a discrete random image of size \( N \). Assume that the null hypothesis \( \mathcal{H}_0 \) is satisfied. Let \( M \) be a positive integer. The expectation of the number of \( \epsilon \)-meaningful junctions of order \( M \) in \( \mathcal{J}(M) \) is smaller than \( \epsilon \).

**Proof.** First, observe that if \( X \) is a random variable and if we define \( F(t) = \mathbb{P}[X \geq t] \), then for all \( \beta \) in \([0, 1]\), \( \mathbb{P}[F(X) \leq \beta] \leq \beta \). Thus, if \( j \) is a junction of scale \( r \) in \( \mathcal{J}(M) \), since \( F_j \) plays exactly this role towards the random variable \( t(j) \) under the hypothesis \( \mathcal{H}_0 \),
\[
\mathbb{P}_{\mathcal{H}_0} \left[ \text{NFA}(j) \leq \epsilon \right] = \mathbb{P}_{\mathcal{H}_0} \left[ F_j(t(j)) \leq \frac{\epsilon}{\#\mathcal{J}(M)} \right] \leq \frac{\epsilon}{\#\mathcal{J}(M)}. \tag{6.3.8}
\]

Finally,
\[
\mathbb{E}_{\mathcal{H}_0} \left[ \#\{j \in \mathcal{J}(M); \text{NFA}(j) \leq \epsilon\} \right] = \sum_{j \in \mathcal{J}(M)} \mathbb{P}_{\mathcal{H}_0} \left[ \text{NFA}(j) \leq \epsilon \right] \leq \sum_{j \in \mathcal{J}(M)} \frac{\epsilon}{\#\mathcal{J}(M)} = \epsilon. \tag{6.3.9}
\]

\( \square \)
Observe that in the definition of the NFA, the correcting factor $\#J(M)$ is independent of the scale $r$. We could have used different correcting factors, depending on the junction scale $r$, in order to favor some particular scales in the detection, and still have a result similar to proposition 2.

### 6.3.4 Minimum scale of detection

In images, some structures may appear as junctions at large scales without being detected as such when the scale is small. For instance, when three straight edges have close endpoints and are supported by concurring lines, an $Y$-junction may be detected even if the actual edges do not physically meet, which means that the junction contains a gap at its center. Psychophysical experiments [218] suggest that human may also find junctions although there is a small gap at the center. The question of the detection of such junctions is subtle. In practice, we observed that results of detection were visually more satisfying by removing junctions with large gaps at their center. This restriction is imposed by computing for each $\epsilon$-meaningful junction $j : \{p, r, \{\theta_m\}_{m \in \{1, \ldots, M\}}\}$ a minimum scale of detection, defined as (see Figure 6.7)

$$r_d[j] = \min \{r' \leq r; \forall s \in [r', r], \exists j' : \{p, s, \{\theta_m'\}_{m=1}^M\} \text{ s.t. } j' \text{ is } \epsilon\text{-meaningful.} \}, \quad (6.3.10)$$

and by removing all $\epsilon$-meaningful junctions such that $r_d[j] > 12$. In order to be fully scale invariant, this threshold could be replaced by a value proportional to the junction scale $r$.

![Figure 6.7: Scale of a junction. Left: image $I$. The point $p$ is chosen as the center of $I$. Middle: for each scale $r$ (in abscissa), smallest NFA($j$) observed for a junction $j$ of order 3 centered at $p$. Right: smallest scale of detection $r_d = 8$ in magenta and scale of the maximal $\epsilon$-meaningful junction in red (found for $r = 13$), represented on the strength $\gamma_p(q)$.

### 6.3.5 Maximality

**Scale and location** Let $M$ be a positive integer. If a junction $j : \{p, r, \{\theta_m\}_{m \in \{1, \ldots, M\}}\}$ is detected as $\epsilon$-meaningful in an image $I$, with a very small value NFA($j$), it means that the strengths of the sectors $S_p(r, \theta_m)$ at scale $r$ are all very high. If $r$ increases or decreases, these strengths should remain important before eventually becoming smaller than the detection threshold $t(r, \epsilon)$. As a consequence, one can expect to detect as $\epsilon$-meaningful other junctions $j' : \{p, r', \{\theta_m'\}_{m \in \{1, \ldots, M\}}\}$, with the same center $p$ and a scale
6. Definition and Detection of Junctions

$r'$ in a neighborhood of $r$. In the same way, because the junction model relies on angular sectors, junctions $j': \{p', r, \{\theta_m\}_{m \in \{1, \ldots, M\}}\}$, with the same scale $r$ and a center $p'$ in a neighborhood of $p$ may also be detected, particularly if the images $I$ contains some blur. Finally, these two phenomena may be combined and junctions $j': \{p', r', \{\theta_m\}_{m \in \{1, \ldots, M\}}\}$ with $p'$ and $r'$ close to $p$ and $r$ will also be detected. This is illustrated by Figure 6.8. In this example, the left image displays an ideal Y-junction $j: \{p, r, \{\theta_m\}_{m \in \{1, \ldots, 3\}}\}$, with $r = 20$. The center image shows all the meaningful junctions detected with the same center but different scales, and the right image shows all the junctions detected with the same scale but different positions close to $p$. All of these junctions represent the same underlying structure.

**Figure 6.8:** Redundancy of junction detection. For the sake of clarity, each junction is represented by a circle and its center, the radius of the circle is equal to the scale of the junction, and the color of the circle is function of its NFA value (red corresponds to small values, i.e. very meaningful junctions and blue corresponds to high values). **Left:** a junction $j: \{p, r, \{\theta_m\}_{m \in \{1, \ldots, 3\}}\}$, with $r = 20$. **Mid:** all junctions detected at the same point $p$, with different scales. **Right:** all Y-junctions detected in the neighborhood of $p$, with the same directions and scale.

**Orientations** Another kind of redundancy might appear when two junctions $j$ and $j'$ of the same order $M$ are found at the same location $p$ in $\Omega$ and with the same scale $r$. Since the branches of these junctions correspond to semi-local maxima of the strength function $\omega_p(r, \cdot)$, we can assume that at least one of the branches of $j$ is different from the branches of $j'$, and as a consequence, that the underlying structure in the image has an order larger than $M$.

**Maximal junctions** In order to choose the right representative among all these redundant detections, we use an exclusion principle, called maximality. In a neighborhood $N'_p$ (different from the normalization neighborhood $N_p$) around each point $p$, we only authorize one junction of a given order $M$. This junction is chosen as the one with the smallest NFA value (see Figure 6.9 for an illustration).

**Definition 4. (Maximal $\epsilon$-meaningful junction of order $M$)** A junction $j: \{p, r, \{\theta_m\}_{m = 1}^M\}$ is said to be a maximal $\epsilon$-meaningful junction of order $M$ if $j$ is $\epsilon$-meaningful and if $\text{NFA}(j) \leq \text{NFA}(j')$ for any junction $j': \{p', r', \{\theta_m\}_{m = 1}^M\}$, with $p' \in N'_p$. 
In this definition, the NFA permits to compare structures at different scales. Using the strengths \( t(j) \) to carry out this comparison would require a well chosen normalization taking into account the scales of the junctions.

![Figure 6.9: Maximal meaningful junctions or order 3. Each Y-junction is represented by a circle and its center. The radius of the circle indicates the junction scale \( r \), and the color of the circle corresponds to the NFA of the junction (the cooler the color, the larger the NFA). To illustrate the redundancy of detections both in scale and space, each junction is displayed with a height \( 8r \) in the 3D graph. **Left:** all \( \epsilon \)-meaningful Y-junctions, with \( \epsilon = 1 \). **Mid:** the maximal meaningful Y-junctions. **Right:** the maximal meaningful Y-junctions in the image.](image)

In practice, the spatial neighborhood \( N_p' \) for maximality is chosen as a disk centered at \( p \), with a radius \( r_d(j) \).

Observe that in Definition 4, for a given \( M \), we could have restricted maximality to only scale and position, by imposing on the set of orientations \( \{ \theta_m \}^M_{m=1} \) to be close to \( \{ \theta_m \}^M_{m=1} \). If there is a Y-junction in \( I \) located at \( p \) for instance, we would have found 3 L-junctions instead of one. In this case, however, our goal is to find a Y-junction, so the number of L-junctions found at \( p \) does not really matter. In order to choose the right representative at \( p \), a second exclusion principle is added, where only the more complex junction (the one with the largest order) is kept.

**Definition 5.** (Maximal \( \epsilon \)-meaningful junction) A junction \( j : \{ p, r, \{ \theta_m \}^M_{m=1} \} \) is said to be a maximal \( \epsilon \)-meaningful junction if \( j \) is a maximal \( \epsilon \)-meaningful junction of order \( M \) and if there is no maximal \( \epsilon \)-meaningful junction of order \( M' \) located at \( p' \) with \( M' > M \) and \( p' \in N_p' \).

### 6.3.6 The final algorithms for junction detection

The different steps of the whole *a contrario* junction detection are summarized in Algorithms 7, ?? and ?? . Some important remarks and equations on the best way to compute NFA\((j)\) can be found in Section 6.4.

Notice that line 11 of Algorithm 7 is a first step towards maximality, since only the best junction of a given order \( M \) is tested at each point. This permits to speed-up the algorithm by excluding junctions that will obviously not be maximal. If we wish to compute all \( \epsilon \)-meaningful junctions in an image, and not only maximal junctions, lines 11 to 22 should
Algorithm 7: A *contra rio* junction detection

**Require:** An image \(I : \Omega \to \mathbb{R}\), a maximal order \(M'\) and a parameter \(\epsilon\).

**Ensure:** A list of junctions \(Jlist\) and the corresponding list \(r_d[Jlist]\).

1. compute \(\nabla I\) using Equation (6.2.1).
2. for each value of \(M\) between 1 and \(M'\), compute \(#J(M)\) by using Equation (6.3.5).
3. for \(p \in \Omega\) do
4.   \(RD\) be a \(M' \times r_{max}\) matrix (used to record the values \(r_d\)), and fill it with zeros.
5.   for \(r = 1\) to \(r_{max}\) do
6.     (1) compute \(\omega_p(r, \theta)\) for each \(\theta\) in \(D(r)\), using Equations (6.2.5) and (6.2.6);
7.     (2) use a NMS procedure to only keep semi-local maxima of \(\omega_p(r, .)\);
8.     \(\theta_{\text{index}(1)}\) the set of all these directions;
9.     (3) sort the vector \(\omega_p(r, \Theta)\) in a descending order;
10. \(\text{(index, } v) \leftarrow \text{sort}(\omega_p(r, \Theta));\)
11. \(\theta_{\text{index}(2)} = \theta_{\text{index}(3)};\)
12. if \(M = 2\) and \(d_{2\pi}(\theta_{\text{index}(1)}, \theta_{\text{index}(2)} + \pi) \leq 2\Delta(r)\)
13. \(\theta_{\text{index}(2)} = \theta_{\text{index}(3)};\)
14. end if
15. for \(M = 1\) to \(M'\) do
16.   Propose a junction \(j : \{p, r, \{\theta_m\}_{m=\text{index}(1)}^{\text{index}(M)}\}\), with \(t(j) = v(\text{index}(M));\)
17.   Compute \(\log \text{NFA}(j)\) by using Equation (6.4.1);
18.   if \(\log \text{NFA}(j) \leq \log \epsilon\) then
19.     if \(RD(M, r - 1) \neq 0\) then
20.       \(RD(M, r) = RD(M, r - 1)\) and \(r_d[j] = RD(M, r - 1);\)
21.     else
22.       \(RD(M, r) = r\) and \(r_d[j] = r;\)
23.     end if
24.   if \(r_d[j] < 13\) then
25.     Accept the junction proposal \(j: Jlist \leftarrow \);\)
26.   end if
27. end if
28. end for
29. end for

be replaced by

for each junction \(j : \{p, r, \{\theta_m\}_{m=\text{index}(1)}^{\text{index}(M)}\}\), with \(\text{index}(m_1) < \cdots < \text{index}(m_M)\)
and \(t(j) = v(\text{index}(m_M))\),
Compute \(\log \text{NFA}(j)\) by using Equation (6.4.1)
if \(\log \text{NFA}(j) \leq \log \epsilon\)
Accept the junction proposal \(j: Jlist \leftarrow \).
end if
end for.
## 6.4 Implementation

The goal of this section is to provide all numerical details necessary to the implementation and speed-up of Algorithms 7, ?? and ??.

### 6.4.1 Computing NFA in practice

The NFA of a junction \( j : \{ p, r, \{ \theta_m \}_{m=1}^M \} \) has been defined as \( \text{NFA}(j) := \# J(M) \cdot F_j(t(j)) \), where

\[
F_j(t) = \prod_{m=1}^{M} \int_{t}^{+\infty} J(r, \theta_m) \mu(dz),
\]

\( \mu(dz) \)
with
\[ \mu(z) = \frac{1}{2} \delta_0(z) + H(z) \cdot \frac{1}{\sqrt{\pi}} e^{-\frac{z^2}{2}} \operatorname{erfc}(\frac{z}{\sqrt{2}}) dz. \]

In practice, the numerical values taken by \( F_j(t(j)) \) can become smaller than the precision of the computer when the strength \( t(j) \) is too high. For this reason, we write the distribution \( \mu \) as
\[ \mu(z) = \frac{1}{2} \delta_0(z) + \frac{1}{2} R(z) dz, \]
where
\[ R(z) = 2H(z) \cdot \frac{1}{\sqrt{\pi}} e^{-\frac{z^2}{2}} \operatorname{erfc}(\frac{z}{\sqrt{2}}). \]

When this distribution is convolved \( k \) times with itself, it becomes
\[ \bigotimes_{j=1}^{k} \mu = \left( \frac{1}{2} \right)^k \cdot \left( \delta_0 + \sum_{j=1}^{k} \binom{k}{j} (j^R) \right). \]

By integrating this function between \( t > 0 \) and \( +\infty \), we obtain
\[ \int_{t}^{\infty} \bigotimes_{j=1}^{k} \mu = \left( \frac{1}{2} \right)^k \cdot \sum_{j=1}^{k} \binom{k}{j} \int_{t}^{+\infty} (j^R). \]

Since \( \left( \frac{1}{2} \right)^k \) and the integral are both very small, we compute instead
\[ \log \int_{t}^{\infty} \bigotimes_{j=1}^{k} \mu = -k \log(2) + \log \left( \sum_{j=1}^{k} \binom{k}{j} \int_{t}^{+\infty} (j^R) \right). \]

Finally,
\[ \log \text{NFA}(j) = \log(\#J(M)) - \log 2 \sum_{m=1}^{M} J(r, \theta_m) \]
\[ + \sum_{m=1}^{M} \log \left( \sum_{j=1}^{J(r, \theta_m)} \binom{J(r, \theta_m)}{j} \int_{t(j)}^{+\infty} (j^R) \right). \] (6.4.1)

This formula is used in practice and compared to \( \log(\epsilon) \) in order to select \( \epsilon \)-meaningful junctions.

### 6.4.2 Choice of \( K(r) \) and \( \Delta(r) \)

The precision in discrete images is given by the size of a pixel. For this reason, we consider that the number \( K(r) \) of possible directions at a given scale \( r \) around a point \( p \) should be equal to the average number of pixels in \( \Omega \) meeting the circle of radius \( r \) centered at \( p \). In practice, we approximate this number by the integer value
\[ K(r) = \lfloor 2\pi r \rfloor. \] (6.4.2)
6.4. Implementation

The choice of the precision $\Delta(r)$ relies on similar considerations. Visual experiments show that the perceived precision of an angle between two crossing segments in an image is better for long segments than short ones. Now, recall that $2\Delta(r)$ is the angle of a branch (or sector) in a junction at scale $r$. We consider that the length of the arc defined by this sector should be a constant $w$ and not depend on $r$. This length is exactly $2\Delta(r) \times r$, which implies that $\Delta(r)$ should be chosen as inversely proportional to $r$. In practice, we choose $w = 5$. Thus

$$2\Delta(r) = \frac{w}{r} = \frac{5}{r}.$$  \hspace{1cm} (6.4.3)

It follows that for a given order $M$, the number of tests $\#J(M)$ can be computed as

$$\#J(M) = \frac{N}{M!} \cdot \sum_{r=r_{\text{min}}}^{r_{\text{max}}} \prod_{m=1}^{M} [2\pi r] \left(1 - (m-1) \frac{5}{r\pi}\right),$$  \hspace{1cm} (6.4.4)

where $N$ is the total number of pixels in the image. In the experimental section, the maximum order of junctions will be $M = 5$ and $r_{\text{min}}$ will be chosen as $r_{\text{min}} = 4$. The maximal scale is chosen as $r_{\text{max}} = 30$. Figure 6.4 illustrates the choices of $\Delta(r)$ for two different scale $r$.

![Figure 6.10: $\Delta(r)$ in function of $r$ for two choices of $r$. Observe that the length of the arc defined by a sector is a constant $w$ and not depend on $r$](image)

6.4.3 Direction refinement

Since directions in a junction are bisectors of angular sectors (see Equation (6.2.4) for the definition of $S_p(r, \theta)$), it may happen that the directions of some branches in a detected junction remain imprecise (see Figure 6.11a for an example). To overcome this problem,
Definition and Detection of Junctions

(a) before refinement

(b) after refinement

Figure 6.11: Direction refinement. In (a), one of the directions in the detected junction is not well located. (b) shows the refinement of all directions using Equation (6.4.5).

for a branch of direction $\theta$ centered at $p$, a refined direction $\hat{\theta}$ is computed as follows

$$\hat{\theta} = \arctan \frac{O_y}{O_x},$$

with

$$O_x = \sum_{q \in S_p(r,\theta)} \gamma_p(q) \cos \psi_q; \quad O_y = \sum_{q \in S_p(r,\theta)} \gamma_p(q) \sin \psi_q;$$

and

$$\psi_q = \begin{cases} \phi_q & \text{if } d_2(\phi_q, \theta) < \frac{\pi}{2} \\ \phi_q + \pi & \text{otherwise.} \end{cases}$$

Notice that after this refinement, two branches in a given junction may intersect. If this happens, the detected junction is removed. The whole refinement process is described in Algorithm ??.

6.4.4 Speed up

Following [156, 41, 40, 192], we propose to apply some pre-processing steps to select potential junction candidates and speed up the algorithm.

Since junctions are constituted of branches, it makes sense to take advantage from a fast segment detector to speed-up the algorithm. In this chapter, we make use of the fast Line Segment Detector (LSD) [210] 3, whose complexity is linear in the size of the image. In order to make sure that we won’t miss some junction candidates, the detection threshold $\lambda$ of the LSD is set to be much larger than $\epsilon$ (for instance, $\lambda = 10^6 \cdot \epsilon$). Once all possible line segments have been found, potential junction locations are restricted to a small neighborhood around each endpoint of those line segments. Figure 6.12 displays all line segments detected with $\lambda = 10^4$ for a given image and shows the corresponding junction candidates in red.

3The code of LSD can be downloaded from the IPOL website: http://www.ipol.im/pub/algo/gjmr_line_segment_detector/.
**Algorithm 10: Direction refinement**

**Require:** A junction \( j : (p, r, \{\theta_k\}_{k=1}^M) \)

1. (1) **Refinement of junction branches:**
   - for \( k = 1 \) to \( M \) do
     - \( O_x = 0, O_y = 0; \)
     - for \( q \in S_p(r, \theta_k) \) do
       - Compute \( \gamma_p(q) \) using Equation (6.2.6);
       - Compute \( \psi_q \) using Equation (6.4.6);
     - end for
     - \( O_x \leftarrow O_x + \sum_{q \in S_B} \gamma_p(q) \cos \psi_q; \)
     - \( O_y \leftarrow O_y + \sum_{q \in S_B} \gamma_p(q) \sin \psi_q; \)
   - Update the branch direction \( \theta_k = \arctan \frac{O_y}{O_x}. \)
   - end for

2. (2) **Check that branches are still disjoint.**
   - for \( m = 1 \) to \( M \) do
     - for \( k = m + 1 \) to \( M \) do
       - if \( d_{2\pi}(\theta_m, \theta_k) < \Delta(r) \) then
         - Remove \( j \), then break;
       - end if
     - end for
   - end for

---

Figure 6.12: Junction candidates using LSD. **Left:** an original image; **Mid:** all line segments detected with \( \lambda = 10^4 \); **Right:** all junction candidates.

Table 6.1 shows the number of detections and the running time when using or not the LSD preprocessing. The comparisons are implemented on three types of images: images with a strong geometrical content, images containing mostly textures and photographs of abstract paintings (see Figure 6.13). Using candidates from LSD clearly reduces the computing time, while the quantity of detections is not affected. Notice that this reduction strongly depends on image structures. The simpler the structures contained within the images, the larger the achieved reduction.
Figure 6.13: Test images for the speed up evaluation. **Top:** house, Lena and window; **Mid:** autumn, park and branches; **Right:** images of geometric paintings: *Geometric* by John Cooper, *Composition* by Charmion von Wiegand, and *Suprematism* by Kazimir Severinovich Malevich.

### 6.4.5 Algorithm pipeline

The pipeline of the whole a contrario junction detection algorithm is summarized in Algorithm ???. Remark that the only parameters to be set in Algorithm ?? are the detection threshold $\epsilon$, the maximum observation scale $r_{max}$ and the maximum order of junctions $M'$. 
Table 6.1: Number of detections and running time when using or not the LSD preselection. The comparisons are implemented on three types of images: images with a strong geometrical content, images containing mostly textures and photographs of abstract paintings. Num (L/T) stands for the numbers of L/T detected junctions respectively.

<table>
<thead>
<tr>
<th>Image</th>
<th>Using LSD candidates</th>
<th>Using all candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>class</td>
<td>name</td>
</tr>
<tr>
<td>nature</td>
<td>house</td>
<td>256 × 256</td>
</tr>
<tr>
<td></td>
<td>lina</td>
<td>256 × 256</td>
</tr>
<tr>
<td></td>
<td>window</td>
<td>768 × 576</td>
</tr>
<tr>
<td>texture</td>
<td>autumn</td>
<td>576 × 768</td>
</tr>
<tr>
<td></td>
<td>park</td>
<td>576 × 768</td>
</tr>
<tr>
<td></td>
<td>branches</td>
<td>536 × 819</td>
</tr>
<tr>
<td>geometry</td>
<td>geometric</td>
<td>655 × 518</td>
</tr>
<tr>
<td></td>
<td>composition</td>
<td>413 × 300</td>
</tr>
<tr>
<td></td>
<td>suprematism</td>
<td>400 × 640</td>
</tr>
</tbody>
</table>

**Algorithm 11:** Adaptive junction detection

**Require:** Image $I$, threshold $\epsilon$

1. Use LSD to preselect junction candidates $C$;
2. Detect $\epsilon$-meaningful junctions for all orders smaller than $M'$ and for all scales between $r_{min}$ and $r_{max}$ by using Algorithm 7. Store junctions of order $M$ in a list $Jlist(M)$.
3. For each $M \leq M'$, refine all junctions in $Jlist(M)$ by using Algorithm ??.
4. Keep only maximal junctions in the lists $Jlist(M)$ by using Algorithm ?? and ??.
Chapter 7

Detecting Junctions in Natural Images

In this chapter, we experimentally analyze the a contrario junction detection method. Section 7.1 illustrates the control of the number of false detections in a random noise image. We then investigate the invariance properties of the method, with respect to scale changes in Section 7.2 and with respect to contrast changes in Section 7.3. The proposed method is then compared with some state-of-the-art junction detection approaches in Section 7.4.

To clarify the following comparisons, we first summarize all the involved methods in Table 7.1. Notice that in Table 7.1, and also in this whole chapter, when we say an algorithm is contrast invariant, we refer to local contrast invariance, which means that the results of the algorithm are expected to be unchanged under local contrast changes.

7.1 Controlling the Number of False Detections

In the first experiment of this section, the goal is to check that the expectation of the number of $\epsilon$-meaningful detections under the background model $H_0$ is smaller than $\epsilon$ (Proposition 2). We then illustrate how the threshold $\epsilon$ controls the number of detections in natural images and in images of Gaussian noise.

7.1.1 False alarms when $H_0$ is “almost” satisfied

In order to test the a contrario model, we use a set of 1000 independent random images of size $100 \times 100$. Observe that for a Gaussian noise image $I$, the values of the contrast normalized gradient $\nabla I$ computed from Equation (6.2.2) are not independent. In order to get closer to the a contrario model, the random images we consider are created in the following way. We first consider large Gaussian noise images $J$ (pixel values are i.i.d. random variables following a standard normal distribution). Each of these images has a size $(l_n + 2)100 \times (l_n + 2)100$, where $l_n \times l_n$ is the size of the neighborhood $N_p$. We then downsample $\nabla J$ by a factor of $l_n + 2$ to obtain $\nabla I(p) \approx \nabla J((l_n + 2)p)$. This ensures that the values of $\|\nabla I\|$ are independent and follow approximately a Rayleigh distribution. The gradient orientations are also independent and follow a uniform distribution on $[0, 2\pi]$. 
Table 7.1: Classical approaches for junction detection.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cues for junction detection</th>
<th>Scale of junctions</th>
<th>Classification of junctions</th>
<th>Contrast changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parida's [156]</td>
<td>Integrate gradients and region consistency by using piecewise constant templates</td>
<td>Each junction is associated with an adaptive scale</td>
<td>Yes</td>
<td>Dependent</td>
</tr>
<tr>
<td>Cazorla's [40]</td>
<td>Gradients amplitudes (for the edge-based method) or piecewise region consistency (for the region-based method)</td>
<td>Junctions are detected at a given scale</td>
<td>Yes</td>
<td>Dependent</td>
</tr>
<tr>
<td>Sinzinger's [192]</td>
<td>Radial edges</td>
<td>Junctions are detected at a given scale</td>
<td>Yes</td>
<td>Robust</td>
</tr>
<tr>
<td>Maire's [129]</td>
<td>Edge map from learned global probability of boundary, which integrates gradients of brightness, color, and texture at different scales.</td>
<td>No scale definition for junctions</td>
<td>No</td>
<td>Dependent</td>
</tr>
<tr>
<td>Harris [88]</td>
<td>Autocorrelation matrix on image intensity.</td>
<td>No scale definition for junctions</td>
<td>No</td>
<td>Dependent</td>
</tr>
<tr>
<td>Our</td>
<td>Amplitudes and phases of gradients</td>
<td>Each junction is associated with an adaptive scale</td>
<td>Yes</td>
<td>Robust</td>
</tr>
</tbody>
</table>

The validation experiment consists of the following procedure: for different values of the detection threshold $\epsilon$ and for all images, all $\epsilon$-meaningful M-junctions (resp. L-junctions, Y-junctions and X-junctions) are detected. This is done thanks to a slight modification of Algorithm 7: line 7 of the algorithm, the NMS procedure is not used (all strengths of $w$ are considered, not only the semi-local maxima); line 11 of the algorithm, all possible junctions of order $M$ with branches in $\Theta$ are tested, and not only the best one, since we wish to count all $\epsilon$-meaningful junctions and not only maximal ones (see Section 6.3.6). This average number of detections over the 1000 images is then computed and noted $M_{\text{NFA}}(\epsilon)$.

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>$10^{-3}$</th>
<th>$10^{-2}$</th>
<th>$10^{-1}$</th>
<th>$10^{0}$</th>
<th>$10^{1}$</th>
<th>$10^{2}$</th>
<th>$10^{3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{\text{NFA}}$ of L</td>
<td>0</td>
<td>0.009</td>
<td>0.089</td>
<td>0.896</td>
<td>8.210</td>
<td>82.006</td>
<td>806.354</td>
</tr>
<tr>
<td>$M_{\text{NFA}}$ of T</td>
<td>0</td>
<td>0.007</td>
<td>0.075</td>
<td>0.766</td>
<td>7.231</td>
<td>68.521</td>
<td>536.756</td>
</tr>
<tr>
<td>$M_{\text{NFA}}$ of X</td>
<td>0.001</td>
<td>0.006</td>
<td>0.058</td>
<td>0.493</td>
<td>3.726</td>
<td>28.715</td>
<td>201.254</td>
</tr>
</tbody>
</table>

Table 7.2: Average numbers of $\epsilon$-meaningful junctions when $H_0$ is satisfied.

Table 7.2 illustrates the results of this experiment, with $\epsilon$ varying from $10^{-3}$ to $10^{3}$. Observe that the average number of detections is always smaller than $\epsilon$. 
7.1.2 Detections in Gaussian noise

As suggested in Section 7.1.1, the proposed method can control the number of false detections in Gaussian noise images. It is of interest to study the respective performance of the Harris detector and the “Pj on $gPb$” detector proposed by Maire et al. [129] in the same context. Figure 7.1 displays the detection results produced by those two methods on Gaussian noise images. Both of them detect many points. Recall that the Harris detector finds junctions by relying on the local maxima of the function $\text{det}(A) - \kappa \cdot \text{trace}^2(A)$, where $A$ is a local structure tensor and only the top 10% local maxima are kept. This results in the detection of many local maxima in Gaussian noise images, see Figure 7.1 (d). The detector “Pj on $gPb$” is based on the edge-map $gPb$ (see [129]), a boundary detector based on parameters that are learned on the manually segmented Berkeley database. Figure 7.1 (b) shows the thinned global Probability of boundaries ($gPb$). Figure 7.1 (c) illustrates the detected junctions, each of which is associated with a significance calculated from the salience of contours $gPb$ and the angles between contour segments. Figure 7.1 (e)-(h) display respectively the detected L-, Y-, X-, and all junctions in the image of Gaussian noise, using the method we introduced in the previous chapter. Notice that there is no detection with $\epsilon = 1$, which is consistent with the background model.

Figure 7.2 illustrates one more example on a textured image, where visually there is no junctions but where both the “Pj on $gPb$” detector and Harris detector give numerous detections, while the proposed method with $\epsilon = 1$ detects no junction, which is more consistent with our perception. Of course, one can argue that we can threshold the significance of the detections of “Pj on $gPb$” detector and Harris detector to obtain satisfying results. However, to our knowledge, there is no criterion to set such thresholds automatically.

Another point of interest is the behavior of junction detection as images become noisier. According to the proposed junction model, for a given $\epsilon$, when the noise level increases the number of detections is expected to decrease and finally should tend towards $\epsilon$. Figure 7.3 illustrates this point. For the house image, we create a sequence by adding Gaussian noises with zero mean and standard deviations in $\{8.1, 25.51, 80.6\}$ (selected as $\sigma * 255$, with $\sigma^2$ taking the values 0.001, 0.01 and 0.1) to the original image, see Figure 7.3 (a). Figure 7.3 (d) illustrates the detection results using our method with $\epsilon = 1$. We can see that the number of detections decreases when $\sigma$ becomes larger. Eventually, when $\sigma = 0.1$, only one junction is detected. The results of “Pj on $gPb$” detector and Harris detector are respectively displayed in Figure 7.3 (b) and (c). Observe that the number of detections explodes with $\sigma$.

It is worth noticing that in different ways, these three methods are not robust to noise. A practical way to avoid this could be to measure a priori the level of noise in the image and apply some denoising algorithm, as for instance the non-local means algorithm.

---

1 In all the experiments using “Pj on $gPb$”, we use the codes kindly provided by Michael Maire, one author of [129].

2 $\kappa$ is a tunable sensitivity and usually set to be 0.06.
7. Detecting Junctions in Natural Images

(a) white noise  
(b) $gPb$

(c) by “$Pj$ on $gPb$”  
(d) by Harris detector

(e) L-junctions  
(f) Y-junctions

(g) X-junctions  
(h) all junctions

Figure 7.1: Junction detection on an image of Gaussian noise, displayed in (a). (b) is the thinned global Probability of boundary ($gPb$) of (a). (c) shows the junctions detected by the method “$Pj$ on $gPb$” [129]; the significance of each junction is calculated from the salience of the contours composing the junction, i.e. $gPb$, and the angle between the contour segments. (d) is the result of the Harris detector, and shows the corner strengths. (e)-(h) are the detected L-, Y-, X- and all junctions by our method; The color (from blue to red) associated to each junction indicates its significance (from low to high). Refer to the text for more details.

7.2 Scale Invariance

In the literature, junction detection and scale selection classically involve the use of some scale space, usually the linear one. In this section, we first investigate the properties of the scale selection in our approach, and then compare it with other approaches, including the method proposed by Maire et al. [129] and the classical Harris detector.

7.2.1 Scale and resolution in the proposed approach

Let $I$ be an image and $I_s$ the low-resolution image obtained by zooming $I$ by a factor $s < 1$. If $j = \{p, r(1), \{\theta_m\}\}$ is a maximal meaningful junction in $I$, we can expect that a maximal meaningful junction will be detected in $I_s$ at the same location with a scale $r(s) = sr(1)$.

In order to investigate the scale property, we apply the proposed junction detection algorithm to a sequence of images with different resolutions. An original image $I$ is resized with a factor $s_i$ thanks to a bilinear interpolation. The zoom factors used in the experiment are $s_i \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, which implies that, including $I$, there are 8 images
7.2. Scale Invariance

Figure 7.2: Junction detection on a textured image, displayed in (a). (b) shows the detected junctions by the method “Pj on gPh” [129]; the significance of each junction is calculated from the salience of contours composing the junction, i.e. gPh, and the angle between the contour segments. (c) is the result of the Harris detector, corner strengths are shown in color. (d) shows the junctions detected by our method ($\varepsilon = 1$); the color (from blue to red) associated to each junction indicates its significance (from low to high). Refer to the text for more details.

in the sequence. For instance, if the size of $I$ is $576 \times 768$, the size of the smallest image in the sequence will be $172 \times 230$. Figure 7.4 shows one example of such an image sequence.

The testing procedure is then the following one:

(1) **Junction detection:** for an image $s_iI$ in the sequence, we detect all junctions in $s_iI$ by relying on Algorithm ???. This yields a junction list $J_{s_i}$ for each image $s_iI$, with $s_i \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ ($s_0 = 1$ being the finest scale and $s_7 = 0.3$ the coarsest scale).

(2) **Junction matching:** a junction $j_0 : \{p_0, r_0, \{\theta_m\}_{m=1}^{M_0}\}$ in $J_{s_0}$ and a junction $j_i : \{p_i, r_i, \{\theta_m\}_{m=1}^{M_i}\}$ in $J_{s_i}$ are matched if

$$j_i = \arg \min_{j \in E_i} \left\{ S(j_0, j) \right\} \quad (7.2.1)$$
Figure 7.3: Comparison of junction detection on a sequence of images with Gaussian noise of zero means and standard derivations as 8.1, 25.51 and 80.6 from left to right displayed in the first row. The second row illustrates the detected junctions by the method “Pj on \(gPb\)” [129]; The significance of a junction is calculated from the salience of contours composing the junction, i.e. \(gPb\), and the angle between the contour segments. The third row is the result of the Harris detector, the significance of each detection is the corner strength. The fourth row illustrates the junctions detected by our method (all types); The color (from blue to red) associated to each junction indicates its significance (from low to high). Observe that using our method, the number of detection decreases when \(\sigma\) becomes larger, while for the other two methods the number of detections explodes with \(\sigma\). Refer to the text for more details.
7.2. Scale Invariance

Figure 7.4: An example of a multi-resolution image sequence. The size of the original image $I$ is $576 \times 768$. A lower-resolution image $sI$ is obtained by down-sampling the original image with a zoom factor $s$, using bilinear interpolation.

$$E_i = \{ j \in J_{s_i} : S(j_0, j) < \tau, M_0 = M, \| p - s_i \cdot p_0 \|_2 < \rho \}$$  \hspace{1cm} (7.2.2)

with $\tau$ and $\rho$ being small values (for experiments, $\tau = \frac{\pi}{20}$ and $\rho = 3$) and

$$S(j, j_0) = \max_{\theta \in \{\theta_m\}_{m=1}^M} \min_{\theta' \in \{\theta'_m\}_{m=1}^M} d_{2\pi}(\theta, \theta').$$  \hspace{1cm} (7.2.3)

Observe that if the set is empty in Equation (7.2.2), no junction $j_k$ is matched with $j_0$. Once all junctions in the image sequence have been matched, we track all these junctions in the scale space. Thus, for each junction at the finest scale, we get a trajectory of junctions from the coarsest scale to the finest scale.

(3) **Statistics from junction trajectories**: Two statistics on the junction trajectories are investigated. First, we illustrate the scale variations along all trajectories in function of the zoom factor $s_i$. Second, we compute the repeatability rate of junction detections according to the image resolution. More precisely, let $J_{s_0}$ be the list of junctions at the finest resolution and $J_{s_0}(s_0I \rightarrow s_iI)$ be the list of junctions which are matched between $s_0I$ and $s_iI$ for all intermediate resolutions. The repeatability rate is defined as

$$R(s_i) = \frac{\# J_{s_0}(s_0I \rightarrow s_iI)}{\# J_{s_0}}.$$  \hspace{1cm} (7.2.4)

Ideally, if all the junctions at the finest resolution are also detected and matched in all coarser-resolution images from $s_0I$ to $s_iI$, then $R(s_i) = 1$ for all $s_i$.

Figure 7.5 illustrates these scale properties on three images. The largest scale in Algorithm ?? is set to be $r_{max} = 30$. In order to have complete trajectories in the scale
Figure 7.5: Illustration of the scale invariance along the scale space. The first row shows the tested images. The second row shows the scales of junctions along all the trajectories (curves in red) as a function of the zoom factor $s_i$ (the abscissa is $s_i/0.3$). The baselines $\{y = r \cdot s_i\}$, where $r$ changes from 1 to 90, are displayed in blue. The bottom row presents the repeatability rate of the junctions as a function of the zoom factor.

space, we are only interested in the junctions with scale smaller than $r_{\text{max}} \times 0.3 = 9$ in the coarsest image (as it will result in junctions of scale smaller than $r_{\text{max}} = 30$ in the original image). The middle row of Figure 7.5 shows the scales of junctions along all the trajectories (curves in red) as a function of the zoom factor $s_i$. The baselines $\{y = r \cdot s_i\}$, where $r$ changes from 1 to 90, are displayed in blue. We can see that the curves of scale changes are close to the baselines. This implies that the characteristic junction scale is quite robust to image resolution changes.

Figure 7.6 shows several examples of detected junctions along the junction trajectories. Observe that on each single example in Figure 7.6, we can see that the proposed characteristic scales really correspond to the sizes of local structures. For instance, the scale of the L-junctions at the corners of rectangles are the minimum width of the rectangles. Besides, observe that the scales of all the detected junctions seems to be invariant to resolution.
7.2. Scale Invariance

Figure 7.6: Examples of detected junctions along several junction trajectories: each row from left to right is a list of junctions detected at the same relative location in images, from the coarsest resolution to the finest resolution. Refer to the text for more details.

Observe, however, that the proposed algorithm is not strictly scale invariant since we use a fixed smallest scale of detection \( r_d \) at each point (see Section 6.3.5). This appears to us to be consistent with human perception of junctions: when several edges meet with a big gap, we probably disregard it as a junction, though this point clearly deserves more investigation. Figure 7.7 illustrates several such examples of failures with respect to scale invariance. In particular, examples on the second and fourth row are due to the non scale invariance of the maximum size of the gap \( (r_d) \) allowed. Notice also that the curvatures of edges depend on image resolution, so zooming out an image may produce some new \( L \) junctions, see Figure 7.8 for an example.

7.2.2 Comparison with classical methods

In [156], Parida et al. define a characteristic scale for their junction detector as,

\[
\hat{r} = \max \left\{ r; \frac{R(r)}{R(r+1)} < \tau \right\}
\]  

(7.2.5)
Figure 7.7: Some examples of failures in scale detection. Each row from left to right is a list of junctions detected at the same relative location of images from the coarsest resolution to the finest resolution. Refer to the text for more details.

Figure 7.8: A failure in scale invariance. Observe that the curvatures of edges increase when image resolution decreases. As a consequence, new L-junctions may emerge at coarse resolutions.

where \( R(r) := \int_{r_0}^{r} \int_{0}^{2\pi} (\frac{\partial I}{\partial s})^2 d\theta ds \) with \( r_0 \) as a given parameter. Although the scale depends heavily on the empirical value \( \tau \), their experiments show satisfying results when \( \tau = 2.1 \).

The Harris-Laplace detector combines the traditional 2D Harris corner detector with the idea of a linear scale-space representation in order to create a scale-invariant detector. The points detected are those that maximize the Laplacian of Gaussian (LoG) across scales (scale selection) and maximize the Harris corner measure in a local neighborhood (spatial selection). Some results of the Harris-Laplace detector on a synthetic and a real image are given in Figure 7.9 (b) and (d). It seems that the points are not accurately located, and the characteristic scale maximizing the LoG over the linear scale space don’t have an obvious meaning for junctions.

In contrast with the Harris-Laplace detector which uses a linear scale space, our NFA-based scale selection do not rely on smoothing (see Section 6.3). A comparison is given in Figure 7.9. Observe that the scales computed by our method in Figure 7.9 (a) and (c) correspond to the optimal size at which one can observe the junction in the image, which implies that the NFA of the junction is larger (the junction is less meaningful) at other scales (smaller or larger). For instance, the scale of the L-junctions located at the corner of a rectangle is generally chosen as the length of the smaller side of the rectangle, see Figure 7.9 (a).

In order to investigate the accuracy of junction locations, we test the different methods
Figure 7.9: Scale selection. (a) shows the characteristic scale of L-, Y- and X-junctions given by our approach, and (b) shows the characteristic scale of junctions given by Harris-Laplace on a synthetic image. (c) and (d) show the same comparison on the house image. The location of junctions is noted by a red cross and the scale is illustrated by a yellow circle. Refer to the text for more explanation.

on a sequence of images with different resolutions. The large resolution image is displayed on Figure 7.10 (a). The sequence is created by convolving this image with a Gaussian blur kernel and by adding Gaussian noise. The testing sequence consists of eight images different zoom factors in \( \{1, 0.9, 0.7, 0.8, 0.6, 0.5, 0.4, 0.3\} \). The original image is annotated by hand to create a ground-truth of junctions \( J_g \). The three approaches are then applied to the image sequence. Each method \( \mathcal{M} \) produces a list of junctions \( J_I^M \) on each image \( I \). \( J_I^M \) is compared with the ground-truth \( J_g \), and the location error of method \( \mathcal{M} \) on \( I \) is evaluated by the average difference of locations between the junction list and the ground-truth, as

\[
\text{error}(\mathcal{M}, I) = \frac{1}{\# J_I^M} \sum_{j \in J_I^M, j' \in J_g} \| p_j - p_{j'} \|_2
\]  

(7.2.6)

where only matched pairs \((j, j')\) are considered and \( p_j \) (resp. \( p_{j'} \)) is the location of the junction \( j \) (resp. \( j' \)). The average location errors of different methods are shown in Figure 7.10 (b). We can see that our approach has the smallest errors, followed by the Laplace-Harris detector and “\( P_j \) on \( gPb \)”. The lack of precision of these two methods is most probably due to the use of Gaussian smoothing.
7. Detecting Junctions in Natural Images

Figure 7.10: Average location error for different approaches: our approach, called ContJunct, “PJ on gPb” and the Harris detector. The comparison is led on a sequence of images with different resolutions. (a) shows an image in the sequence. (b) illustrate the location errors for different scales. Errors in (b) are computed by using Equation (7.2.6).

7.3 Contrast Invariance

In the proposed algorithm, robustness to local contrast changes is achieved by locally normalizing the image gradient. Observe that the normalized gradient \( \tilde{\nabla} I \) is invariant to any affine contrast change on the normalization window \( N_p \) (in experiments, a \( 5 \times 5 \) window). However, the approach is not totally invariant to any local contrast changes. In this part, we first discuss how to achieve total contrast invariance (by relying on the gradient orientation only) and then compare the proposed approach with other classical ones regarding this invariance.

7.3.1 Total contrast invariance

In order to achieve full contrast invariance, one possibility is to rely only on the gradient phase. Desolneux [58], Cao [31] and Grompone von Gioi et al. [210] have used gradient phases in the extraction of edges and curves. In our framework, it is straightforward to remove the weight of gradient amplitudes from the definition of junction strength in order to obtain the same kind of contrast invariance. More precisely, we can define a revised branch strength as

\[
\omega_p(r, \theta) = \sum_{q \in S_p(r, \theta)} \max (|\cos(\phi(q) - \alpha(pq))| - |\sin(\phi(q) - \alpha(pq))|, 0).
\]

With this definition, Equation (6.3.3) becomes

\[
\mathbb{P}_{H_0}[t(j) \geq t] = \prod_{m=1}^{M} \int_{t}^{+\infty} J(r, \theta_m) \int_{j=1}^{\star} \mu(dz),
\]

(7.3.2)
where $\mu = \frac{1}{2} \delta_0 + \frac{2}{\pi} \sqrt{\frac{1}{2-z^2}} H(z) dz$. The resulting junction detection algorithm is similar to Algorithm ??, where we replace Equation (6.3.3) by Equation (7.3.2).

Figure 7.11 compares the approach using contrast-weighted branch strength and branch strength relying only on the orientation of the gradient.

contrast robustness and this total contrast invariance. We can see that although the junction points produced by both methods are almost the same, the totally invariant approach appears to be less robust than the weighted one. Moreover, notice that the JPEG compression in images affects much more the results of totally invariant approach.

(a) L-, Y- and X- and all junctions obtained by the weighted contrast invariant method

(b) L-, Y- and X- and all junctions obtained by the totally contrast invariant method

(c) L-, Y- and X- and all junctions obtained by the weighted contrast invariant method

(d) L-, Y- and X- and all junctions obtained by the totally contrast invariant method

Figure 7.11: Comparison between the weighted contrast invariant junction detection algorithm ((a) and (c)) and the totally contrast invariant junction detection algorithm ((b) and (d)). Observe that while the junction points produced by both methods are almost the same, the totally invariant approach appears less robust than the weighted one. For instance, some junctions are misclassified.
7.3.2 Comparison with classical approaches

In this section, we propose to evaluate the robustness to contrast changes of different junction detection approaches. However, we should first observe that region-based methods, as the methods in [156] and [40], are not contrast invariant, and boundary-based methods, as the methods in [129] and in [192], are semi-contrast invariant, i.e. invariant to contrast changes applied on their junction detection window. The Harris detector, which relies on local gradient measures, is not invariant and yields large responses on high-contrasted boundaries.

To compare different methods regarding to contrast invariance, we test them on a sequence of images with illumination changes (see Figure 7.12). More precisely, we detect junctions in each image, by using our contrast-weighted method, our totally contrast invariant method (see Section 7.3.1), the method using “Pj on gPb” [129], and the Harris detector. For our approach, we also classify junctions into three types, L, Y, and X-junctions. The threshold ε is set to 1 and the threshold on local maxima in Harris detector is chosen as 0.06. The detection results are illustrated in Figure 7.12. Visually, we can see that for the proposed method, X-junctions are more robust than Y-junctions, and Y-junctions are more robust than L-junctions. Moreover, the results of the two a contrario methods are close to each other, and the totally contrast invariant version has less detections.

For the qualitative evaluation of those methods, we compute the repeatability rate of each method on the image sequence. More specifically, if $\mathcal{J}_0$ is the list of junctions in the first image, and $\mathcal{J}_i$ the list of junctions in the $i$-th image, we note $\mathcal{J}_0 \cap \mathcal{J}_i$ the set of junctions in $\mathcal{J}_i$ that can be matched with junctions in $\mathcal{J}_0$, where the matching criterion is as given by Equations (7.2.2) and (7.2.1). The repeatability rate [185] of the $i$-th image is then calculated as

$$ Repeat(i) = \frac{\# \{ \mathcal{J}_0 \cap \mathcal{J}_i \}}{\min(\# \mathcal{J}_0, \# \mathcal{J}_i)}. \quad (7.3.3) $$

The curves of repeatability rates for the different methods are shown in Figure 7.13. We can see that the Harris detector has the worst performance with respect to contrast changes, as it is the most contrast dependent. “Pj on gPb” gives better results than the Harris detector, possibly because it relies on an edge detector that is tuned to match boundaries annotated by humans, which may somehow eliminates contrast variations. Both the weighted and totally contrast invariant approaches perform better than the other ones, and the weighted one works slightly better than the totally contrast invariant one.

7.4 Comparison Evaluation on Junction Benchmark

In the previous section, we have investigated the properties of the proposed method and compared it with state-of-the-art approaches. The proposed junction detector has shown many advantages to others. However, there lacks ground truth for the qualitative evaluation of junction detection. To our knowledge, the only available benchmark for junction detection has been recently established by Maire et al. [129], where each image in the database is annotated by different users to form a ground truth. The idea here is that the human annotated benchmark is consistent with our visual perception, and is suitable for evaluating our “meaningful” junction detector.
7.4. Comparison Evaluation on Junction Benchmark

Figure 7.12: Contrast invariance comparison. For a quantitative evaluation, please refer to Figure 7.13.

The junction benchmark is adapted from the human ground truth of the BSDS, which is preliminarily used for boundary detections. From the ground truth segmentations, L-junctions are extracted as locations of high curvatures along the human annotated boundaries, and Y-junctions are extracted as points where three or more regions intersect. Some examples are shown in Figure 7.14, where L-junctions are marked in green and Y-junctions are marked in red. Notice that each image is annotated by a number of different users through drawing the boundaries in the image. Each junction detection algorithm is then evaluated by using the precision-recall curve, produced by the correspondence between machine and human marked junctions.

It’s worth noticing that different users might annotate differently the same image. Figure 7.15a shows the human agreement on the junction detection task. Green points represent all pairwise comparisons of junction locations between users who annotated the same image. The blue points represent the average of such agreements for each image. The overall average agreement (point in red) is the maximum performance a junction detection
Figure 7.12: (cont.) Contrast invariance comparison. For a quantitative evaluation, please refer to Figure 7.13.

Figure 7.15b presents the results of evaluating several junction detection algorithms on the BSDS-based junction benchmark. The baseline Harris detector on the grayscale-level image performs worst of all, with an F-measure of 0.28. Using learned boundary probability as input to Harris permits improve its F-measure to 0.34. Boundary-based algorithms, including “Pj on Canny”, “Pj on Pb” and “Pj on gPb”, perform better than Harris-like methods. The main reason is that they learn and combine different gradient cues at multiple scales. The good performances of approaches relying on the edge maps Pb and gPb can be explained by the fact that the edge detector is optimized to match human-detected edges. These are thereafter evaluated on a “ground truth” obtained using the same boundaries on which the learning was performed. It is thus natural that these approaches

\(\text{Let } Pr \text{ be the precision, i.e. the number of correct results divided by the number of all returned results, and } Re \text{ be the recall, i.e. the number of correct results divided by the number of results that should have been returned. The F-measure is defined as } F = \frac{Pr \cdot Re}{Pr + Re}.\)
Figure 7.13: Repeatability rate of different approaches regarding contrast changes. The image sequence and the corresponding results are shown in Figure 7.12.

Figure 7.14: Some examples of the human annotated junction benchmark [129]. Each image is annotated by different users, who draw the boundaries of the images. L-junctions (in green) are points with high curvatures on the annotated boundaries, and Y-junctions (in red) are points where different regions meet.

get closer to human agreement on this database. If we increase the normalizing window $N_p$ used to compute $\|\nabla I\|$ to $11 \times 11$, instead of $5 \times 5$, the performances of our approach improve. As shown in Figure 7.15b, the F-measure attains 0.37. Some particular results of the junctions benchmark are shown in Figure 7.16.
Figure 7.15: Junction detection benchmark. (a) Green points represent all pairwise comparisons of junction locations between users who annotated the same image. The blue points represent the average of such agreement for each image. The overall average agreement (point in red) is the maximum performance a junction detection algorithm could expect to achieve. (b) results of different methods.

7.5 More Experiments

In this section, we show more results using the proposed approach. These result aim at illustrating the scale selection, the location precision and the branch assignment of the method. Figure 7.17 presents all the junction detections on a synthetic image. We can observe that both the scale, location and the branches of junction are found with high accuracy. Moreover, the junctions are correctly classified. Figures 7.18 and 7.19 present the same results on two natural images, along with the results of the Harris detector and Maire et al. method. Finally, a comparison with the method of Sinzinger [192] is shown in Figure 7.20. Observe that X-junctions are rarely detected in nature images, since they often corresponds to transparency between objects [136]. To clarify the validation of the proposed method, we also show the detection of X-junctions on an image with transparency in Figure 7.21.
Figure 7.16: Some results of the junctions benchmark. Top line: original images. From the second to the fourth line: the detected L-junctions, Y-junctions, and all junctions by the proposed method. The last line: the results by \( P_j \) on \( gP_b \).
7. Detecting Junctions in Natural Images

(a) Left: all junctions, middle: L-junctions, right: Y-junctions

(b) Detailed junctions, ranked according to their NFA

Figure 7.17: All the junction detections on a synthetic image.
Figure 7.18: Specific junction detections on a natural image.
Figure 7.18: (cont.) Specific junction detections on a natural image.
(e) X-junctions

(f) Detailed junctions ranked according to their NFA

Figure 7.18: (cont.) Specific junction detections on a natural image.
Figure 7.19: Specific junction detections on a natural image.
Figure 7.19: (cont.) Specific junction detections on a natural image.
Figure 7.19: (cont.) Specific junction detections on a natural image.
7.5. More Experiments

(a) results obtained with the method of Sinzinger \[192\]

(b) all junctions detected with the proposed approach

Figure 7.20: Comparison with the method of Sinzinger \[192\]. Notice the differences between the results in texture and in low-contrasted regions.
Figure 7.20: (cont.) Comparison with the method of Sinzinger [192]. Notice the differences between the results in texture and in low-contrasted regions.
Figure 7.21: Detection of X-junctions in an image with transparency.
7. Detecting Junctions in Natural Images
Chapter 8

Conclusion and Perspectives

Conclusion

The first part of this manuscript proposes a shape-based invariant texture analysis scheme relying on the topographic map, a morphological multi-scale representation of images. The shapes contained in the topographic map can be seen as textons in the context of texture analysis. Local texture features are computed from the statistics of these shapes. We have shown that the morphological, granulometry-like indexing methods can deal with highly structured texture images and enable state-of-the-art results in invariant texture analysis, even in the case of non-rigid transforms. In addition, we have successfully applied the proposed texture descriptors, combined with an active contour model [95], to the problem of texture segmentation. The proposed approach also demonstrated good performance on the recognition of high resolution satellite images. Finally, we have suggested that comparing textures with the geodesic distances along the statistical manifold formed by PDFs significantly improves the recognition performances. Based on such geodesic distances, an a-contrario method has been introduced to set the matching threshold in the process of retrieving similar images from a database with a given example.

In the second part of this thesis, we have concentrated on a NPR problem and shown that the topographic map enables the creation of geometrical abstractions from digital photographs. The proposed method can be used to produce a wide range of abstraction effects. In contrast with the related methods in the field of NPR, the synthesis procedure presented in this part enables complex interactions between shapes, since the topographic map consists in a hierarchical structure, driven by the inclusion of shapes and accounting for all scales in the image. As an additional point, we have proposed the scale-ratio filter, which allows to suppress image structures according to the scale ratio between shapes and their parents along the topographic map.

Finally, the last part of this thesis has suggested an approach for detecting meaningful junctions in natural images. We handle the detection of junction as a problem of visual perception grouping. In particular, junctions have been taken as locations in images where different edges intersect. The detection of junctions in an image is subsequently governed by an a contrario model, established from the statistical properties of the image. The proposed method can identify the location, scale, and branches of each detected junction simultaneously. It is worth noticing that all these are achieved without using a linear
scale space, which is widely employed in classical approaches for interest points detection, such as SIFT [126]. In contrast with previous works dealing with junction detections, the proposed method finds junctions where are not only locally salient in small patches but also meaningful with respect to a global perception of the image. A perceptual significance is in fact associated to each junction and can be used to control the detection. In addition, the proposed method is also robust to changes of scale and contrast. Moreover, compared with classical junction detection methods, our method involves less parameters.

**Perspectives**

The work of this thesis opens a wide variety of perspectives. In what follows, we only list some most important ones.

1. In our present work on texture analysis, the hierarchical structure of the topographic map is only partially considered. It is of interest to further investigate the descriptive power of statistics on the tree of level lines, making use of specific neighborhoods and higher dependencies in the tree, possibly using probabilistic graphical models. One difficulty is to achieve this while preserving radiometric and geometric invariances. Next, and going beyond local contrast invariances, one could study the behavior of level line statistics under illumination changes in greater details [213]. We show in this manuscript that lines statistics yield efficient retrieval results on databases with varying illumination conditions. The next step could be either to explicitly model level lines variations or to investigate the ability of the topographic map to learn the effects of illumination changes using databases such as CUReT [51].

2. Besides the applications on invariant texture recognition, other possible applications of the proposed texture indexing scheme include the registration of non-rigid objects and shape from texture.

3. Texture synthesis could also be investigated under the framework of texture indexing. As shown in Section 2, shapes are adaptively extracted from the texture samples as textons. The synthesis of a new texture can be implemented by studying the organization and interactions between these shapes, for instance, by relying on the Markov point process [111]. Compared with classical texture synthesis approaches, this kind of synthesis is expected to be able to handle highly structured textures.

4. The image manipulation framework proposed in Chapter 5 has been partially tested for generating a wide range of abstractions in the present work. It is of great interest to further investigate it in the context of NPR, such as to study how it can be used to improve the optimization of the stroke-based image representations, see [92]. Using a map would enable the user to create a back-to-front painting strategy that place small bright objects on the large objects.

5. Other possible extensions on image synthesis is to apply it to manipulate videos; In this case, the abstraction both in spatial and in temporal domain should be taken into account. We have implemented some preliminary experiments, but further studies are demanded.
(6) One main problem in the field of NPR is that it is difficult to measure what is a good abstraction of an image. A possible way to do that is performing some Turing tests: mixing the automatically generated abstractions and the human abstract paintings and to check how our human being can distinguish them. Such experiments has been done in [230] for interactive image abstractions. As we can represent and generate abstraction of images, one interest direction is to investigate the statistics of abstract paintings by relying on their topographic map to develop some measurement of image abstractions.

(7) The junction detection algorithm proposed in Chapter 6 can be generally considered as an interest point detector and descriptor. It enables good invariant properties, such as scale invariance and contrast invariance, to image transforms. A further direction is to apply it to object recognition and image matching. On this subject, the readers are recommended to refer to some recent work [119, 211], which handle object recognition by learning features from junctions detected by classical methods.

(8) Observe that there is a strong link between the invariant texture analysis of Chapter 2 and the junction detection of Chapter 6: the detected junctions in texture images can be used to form a sparse representation of textures, which will allow for the analysis of textures with strong geometric and illumination changes.

Figure 8.1: A small testing texture dataset taken from the UIUC database [113]. It is composed of 4 classes with 10 samples per class.

We show a preliminary experiment on a very small texture dataset, composed of 4 classes with 10 samples per class (see Figure 8.1), taken from the UIUC database [113]. The idea is, inspired by the texton theory of Julesz [100], to represent each texture sample by a set of detected junctions. The histogram of junction classes (HJC), i.e. the frequency of L, Y, X, and 5-branch junctions, is used as a texture descriptor. The confusion matrix between these 40 texture samples is then computed by relying on the Jeffrey divergence, described in Section 3.1. Figure ?? illustrates such a confusion matrix by the HJC descriptor. Observe that the HJC descriptor can somewhat distinguish the 4 texture classes, but it is far from enough to produce results comparable to those shown in Chapter 3. This is because with the same HJC features, we could have images which are very different from the textures. Thus, further studies could concentrate on developing more discriminative texture features,
from the detected junctions.

Figure 8.2: The confusion matrix between texture samples.

(9) Since the junction detection algorithm produces meaningful detections in natural images, it is of interest to use the resulting junctions for computing the visual saliency in natural images [11]. For instance, let \( J = \{ j, j = \{ q, r, \theta_m \} \} \) be the list of detected junctions in a gray-level image \( I \). We define the visual saliency of a pixel \( p \in \Omega \) in the image \( I \), as

\[
\text{saliency}(p) = \sum_{q_{j}, j \in J} -\log(\text{NFA}(j)) \cdot G_r(p, q_j),
\]

where \( G_r(p, q) = e^{-\frac{\|p - q\|^2}{2r^2}} \) is an exponential kernel with size of \( r \), in order to propagate the saliency of a junction \( j \) at \( q \) to the pixels \( p \) around \( q \). A first experiment on several images is shown in Figure 8.3. The results are compared with a recent approach computing the visual saliency by self-resemblance [186].
Figure 8.3: A first experiment on computing visual saliency. **Left**: original images; **Middle**: the map of visual saliency computed from the junctions (all the junctions are detected in the images with $\epsilon = 1$). **Right**: the map of visual saliency obtained by self-resemblance in [186].
Appendix A

Appendix

A.1 Distribution of $\max(|\cos \theta| - |\sin \theta|, 0)$

Assume that $\theta$ is a random variable following a uniform distribution on $[0, 2\pi]$. If $Y = \max(|\cos \theta| - |\sin \theta|, 0)$, we have $Y \in [0, 1]$ with probability 1 and

$$Y = \begin{cases} |\cos \theta| - |\sin \theta| & \text{if } \theta \in [0, \frac{\pi}{4}] \cup \left[\frac{3\pi}{4}, \frac{5\pi}{4}\right] \cup \left[\frac{7\pi}{4}, 2\pi\right], \\ 0 & \text{otherwise.} \end{cases}$$

(A.1.1)

Note $\Theta = [0, \frac{\pi}{4}] \cup \left[\frac{3\pi}{4}, \frac{5\pi}{4}\right] \cup \left[\frac{7\pi}{4}, 2\pi\right]$, then $|\sin 2\theta| = 1 - Y^2$ for $\theta \in \Theta$. The cumulative distribution function of $Y$ can then be computed as

For $y < 0$, $F_Y(y) = P\left(\max(|\cos \theta| - |\sin \theta|, 0) \leq y\right) = 0$,

For $y = 0$, $F_Y(y) = P\left(\max(|\cos \theta| - |\sin \theta|, 0) \leq 0\right) = \frac{1}{2}$,

For $0 < y \leq 1$, $F_Y(y) = P\left(\max(|\cos \theta| - |\sin \theta|, 0) \leq y\right)$

$$= \frac{1}{2} + \left\{ P\left(\frac{1}{2} \arcsin(1 - y^2) \leq \theta < \frac{\pi}{4}\right) + P\left(\pi + \frac{1}{2} \arcsin(1 - y^2) \leq \theta < \frac{5\pi}{4}\right) + P\left(\frac{3\pi}{4} < \theta \leq \pi + \frac{1}{2} \arcsin(y^2 - 1)\right) + P\left(\frac{7\pi}{4} < \theta \leq 2\pi + \frac{1}{2} \arcsin(y^2 - 1)\right) \right\}$$

$$= \frac{1}{2} + \left(\frac{1}{2} + \frac{1}{\pi} \arcsin(y^2 - 1)\right),$$

For $1 < y$, $F_Y(y) = 1$.

(A.1.2)

Finally, the distribution of the random variable $Y$ can be written

$$\mu_Y(y) = \frac{1}{2} \delta_0(y) + \frac{2}{\pi} \frac{1}{\sqrt{2 - y^2}} 1_{[0,1]}(y)dy,$$  

(A.1.3)
where $\delta_0$ is the Dirac mass centered at $0$, $1_{[0,1]}$ is the indicator function of the interval $[0,1]$ and where $dy$ denotes the Lebesgue measure on $\mathbb{R}$.

## A.2 Distribution of $|\tilde{\nabla} I| \cdot \max(|\cos \theta| - |\sin \theta|,0)$

The goal of this appendix is to compute the distribution of the product $Z = XY$ when $X$ and $Y$ are two independent random variables, $X$ following a Rayleigh distribution of parameter $1$ and $Y$ following the distribution (A.1.3). We recall that the density of the Rayleigh distribution is

$$ f_X(x) = \chi(x,2) = \frac{x^{(2-1)}e^{-x^2/2}}{2^{3/2-1}\Gamma(2/2)} = xe^{-x^2/2}, \quad x \in [0,\infty). \quad (A.2.1) $$

Now, notice that if $Z = XY$, then $Z \in [0,\infty)$ with probability $1$ and the cumulative distribution function $F_Z(z)$ can be computed as follows,

For $z < 0$, $F_Z(z) = 0$.

For $z = 0$, $F_Z(z) = \frac{1}{2}$.

For $z > 0$, $F_Z(z) = \mathbb{P}(0 \leq XY \leq z)$

$$ = \int_0^1 \int_0^{z/y} f_X(x)dx\mu_Y(dy) $$

$$ = \frac{1}{2} + \frac{2}{\pi} \int_0^1 (1 - e^{-\frac{z^2}{y^2}}) \frac{1}{\sqrt{2-y^2}} dy. \quad (A.2.2) $$

The distribution of the random variable $Z$ can thus be written

$$ \mu_Z(z) = \frac{1}{2} \delta_0(z) + H(z) \cdot \frac{2z}{\pi} \left( \int_0^1 \frac{1}{y^2} e^{-\frac{1}{2}(\frac{z}{y})^2} \frac{1}{\sqrt{2-y^2}} dy \right) dz. \quad (A.2.3) $$

In addition,

$$ \int_0^1 \frac{1}{y^2} e^{-\frac{1}{2}(\frac{z}{y})^2} \frac{1}{\sqrt{2-y^2}} dy = \int_1^\infty e^{-\frac{x^2}{2(x^2-1)}} dx, \text{ with } x = \frac{1}{y}, \quad (A.2.4) $$

$$ = \frac{1}{2} \int_1^\infty e^{-\frac{t^2}{4(\sqrt{2}+1)}} dt, \text{ with } t = \sqrt{2x^2-1}, \quad (A.2.5) $$

$$ = \frac{\sqrt{\pi}}{2z} e^{-\frac{z^2}{4}} \cdot \left(1 - \text{erf}\left(\frac{z}{2}\right)\right) = \frac{\sqrt{\pi}}{2z} e^{-\frac{z^2}{4}} \cdot \text{erfc}\left(\frac{z}{2}\right), \quad (A.2.6) $$

where $\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ is the error function and where $\text{erfc}(z) = 1 - \text{erf}(z)$ is the complementary error function. Finally,

$$ \mu_Z(z) = \frac{1}{2} \delta_0(z) + H(z) \cdot \frac{1}{\sqrt{\pi}} e^{-\frac{z^2}{4}} \text{erfc}\left(\frac{z}{2}\right) dz. \quad (A.2.7) $$
Bibliography


