



Faculté des Sciences Appliquées

---

# Learning Visual Feature Hierarchies

---

Année académique  
2007 - 2008

Thèse de doctorat présentée par  
Fabien Scalzo  
en vue de l'obtention du grade de  
Docteur en Sciences (orientation Informatique)



---

---

# Learning Visual Feature Hierarchies

by Fabien Scalzo

Submitted on September 24, 2007 (revised on June 2008)  
in Partial Fulfillment of the Requirements for the Degree of  
Docteur en Sciences (orientation Informatique)

## Abstract

This dissertation addresses the problem of recognizing objects in images. Representation, detection, and learning are the main issues that need to be tackled in designing an object recognition system. Despite more than 20 years of research, this field still remains very challenging and generic aspects of the problem are largely unsolved.

This thesis proposes a framework for the statistical representation of visual features (from which objects are constituted) and their detection in images. The model essentially combines several key concepts that have been developed in the last couple of years in computer vision, machine learning, and computational neuroscience; spatial relations between local visual features, graphical models, and hierarchies of complex cells. This results in a compositional hierarchy of visual feature classes. Its strength is to provide a coherent and generic model by representing both local and global aspects through the combination of shape and appearance modalities.

Interestingly, the use of graphical models provides a convenient formalism to represent complex systems and to exploit efficient inference mechanisms. In this work, we exploit an iterative message-passing algorithm to infer the position of features and thus to detect objects, namely Nonparametric Belief Propagation (NBP). The hierarchical model is learned iteratively and composed in a bottom-up manner. A co-occurrence learning method is used to estimate both the structure and the parameters of the hierarchy.

We also summarize the state-of-the-art with respect to the detection and the description of local visual features. Finally, the behavior of our feature hierarchies is investigated across a variety of object recognition datasets. These experimental evaluations are organized around three recognition tasks of increasingly difficulty.

Thesis Supervisor: Professor Justus H. Piater



---

---

# Learning Visual Feature Hierarchies

par Fabien Scalzo

Thèse soumise le 24 septembre 2007 (révisée en Juin 2008)  
en vue de l'obtention du grade de  
Docteur en Sciences (orientation Informatique)

## Résumé

Cette thèse porte sur la reconnaissance visuelle d'objets, un domaine qui reste un défi majeur en vision par ordinateur. En effet, malgré plus de vingt années de recherche, de nombreuses facettes du problème restent à ce jour irrésolues. La conception d'un système de reconnaissance d'objets repose essentiellement sur trois aspects: la représentation, la détection et l'apprentissage automatique.

La principale contribution de cette thèse est de proposer un système générique pour la représentation statistique des caractéristiques visuelles et leur détection dans les images. Le modèle proposé combine différents concepts récemment proposés en vision par ordinateur, machine learning et neurosciences: à savoir les relations spatiales entre des caractéristiques visuelles, les modèles graphiques ainsi que les hiérarchies de cellules complexes. Le résultat de cette association prend la forme d'une hiérarchie de classes de caractéristiques visuelles. Son principal intérêt est de fournir un modèle représentant à la fois les aspects visuels locaux et globaux en utilisant la structure géométrique et l'apparence des objets. L'exploitation des modèles graphiques offre un cadre probabiliste pour la représentation des hiérarchies et leur utilisation pour l'inférence. Un algorithme d'échange de messages récemment proposé (NBP) est utilisé pour inférer la position des caractéristiques dans les images.

Lors de l'apprentissage, les hiérarchies sont construites de manière incrémentale en partant des caractéristiques de bas-niveaux. L'algorithme est basé sur l'analyse des co-occurrences. Il permet d'estimer la structure et les paramètres des hiérarchies.

Les performances offertes par ce nouveau système sont évaluées sur différentes bases de données d'objets de difficulté croissante. Par ailleurs, un survol de l'état de l'art concernant les méthodes de reconnaissances d'objets et les détecteurs de caractéristiques offre une vue globale du domaine.

Promoteur: Professeur Justus H. Piater



---

---

## Acknowledgments

I feel fortunate to have had the opportunity to work with Justus Piater, who has been my advisor and collaborator. I would like to thank him for enlightening my work with insightful remarks. I also appreciate that he gave me this independence to pursue my ideas both for research and teaching.

I would like to thank Pierre Geurts, Norbert Krüger, Jochen Triesch, Jacques Verly, and Louis Wehenkel for taking the time to being a part of the dissertation committee. I would not have been able to complete this PhD without the support and the confidence of Bernard Boigelot and Pierre Wolper who offered me this exciting position at the Montefiore Institute.

This work is the result of many discussions, interactions with exceptional researchers who helped me with their comments, code or advises to improve the quality of my own work. A special thanks to: Gyuri Dorkó, Björn Holmquist, Alexander Ihler, Kristian Kirk, Raphaël Marée, Jiri Matas, Krystian Mikolajczyk, Thomas Serre, Leonid Sigal, Erik Sudderth, Tinne Tuytelaars, Michel Vidal-Naquet and Ryan White. Thanks to Reza Amayeh, Pradeep Katta, and Alireza Tavakkoli for helping me with the English.

Bien plus qu'un simple travail, cette aventure fut surtout une aventure humaine: je remercie mes collègues pour avoir rendu ces moments plus agréables. Je remercie tout spécialement Renaud Dardenne, Sébastien Jodogne, ainsi qu'Axel Legay pour m'avoir constamment rappelé de rentrer chez moi quand nous étions les deux seuls dans le bâtiment. Merci à Mélanie Godart pour avoir rendu mes trajets dans le 48 vraiment amusants et à Renaud Detry pour l'impression du document.

Et bien sûr, tout ceci n'aurait pas été possible sans le soutien de ma famille.

Merci!





---

---

# Contents

---

<b>Abstract</b>	<b>i</b>
<b>Résumé</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>v</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Algorithms</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Objective . . . . .	1
1.2 Motivation . . . . .	2
1.3 Challenges . . . . .	3
1.4 Contribution . . . . .	4
1.4.1 Generic Feature Hierarchies . . . . .	4
1.5 Outline . . . . .	6
<b>2 Prerequisites</b>	<b>7</b>
2.1 Graphical Models . . . . .	8
2.1.1 Draw me a Graph . . . . .	9
2.1.2 Undirected Graphical Models . . . . .	10
2.1.3 Directed Graphical Models . . . . .	14
2.1.4 Inference on Graphical Models . . . . .	15
2.2 Belief Propagation . . . . .	16

---

2.2.1	BP in Graphs with Cycles . . . . .	18
2.2.2	Implementation Issues . . . . .	18
2.3	Nonparametric Belief Propagation . . . . .	19
2.3.1	Message representation . . . . .	20
2.3.2	Message product . . . . .	20
2.3.3	Message propagation . . . . .	21
2.3.4	NBP Algorithm . . . . .	22
2.4	Discussion . . . . .	23
<b>3</b>	<b>State of the art</b>	<b>25</b>
3.1	Where: Local Feature Detection . . . . .	26
3.1.1	Signal-based Methods . . . . .	27
3.1.2	Geometry-based Methods . . . . .	39
3.1.3	Fast Alternative Methods . . . . .	40
3.1.4	Discussion . . . . .	42
3.2	What: Local Description . . . . .	43
3.2.1	Convolution Coded Descriptors . . . . .	43
3.2.2	Histogram Descriptors . . . . .	47
3.2.3	Alternative Approaches . . . . .	53
3.2.4	Discussion . . . . .	57
3.3	How: Object Recognition Methods . . . . .	58
3.3.1	A Critical View on the History of Object Recognition . . . . .	58
3.3.2	Appearance-Only Models . . . . .	62
3.3.3	Statistical Part-Based Models . . . . .	65
3.3.4	Biologically Motivated Models . . . . .	70
3.3.5	Discussion . . . . .	71
<b>4</b>	<b>Feature Detectors and Descriptors: A Comparative Evaluation</b>	<b>73</b>
4.1	Image Dataset . . . . .	74
4.2	Evaluation of Feature Detectors . . . . .	75
4.2.1	Experimental Protocol . . . . .	75
4.2.2	Results . . . . .	77
4.2.3	Discussion . . . . .	78
4.3	Evaluation of Feature Descriptors . . . . .	78
4.3.1	Experimental Protocol . . . . .	78
4.3.2	Results . . . . .	80

---

4.3.3	Discussion . . . . .	81
<b>5</b>	<b>The Visual Feature Hierarchy</b>	<b>83</b>
5.1	The Search For Representation . . . . .	84
5.1.1	An Ideal Object Representation . . . . .	84
5.2	A Visual Feature Hierarchy . . . . .	86
5.2.1	Hierarchical Feature Set . . . . .	87
5.2.2	Flexible Spatial Relations . . . . .	90
5.3	Representing a Hierarchy via a Graphical Model . . . . .	95
5.3.1	Visual Feature Classes as Nodes . . . . .	95
5.3.2	Spatial Relations as Edges . . . . .	97
5.4	Inferring High-level Features . . . . .	101
5.5	Discussion . . . . .	102
<b>6</b>	<b>Statistical Learning of Hierarchies</b>	<b>107</b>
6.1	Learning Context . . . . .	108
6.2	Overview . . . . .	109
6.3	Motivation . . . . .	110
6.3.1	Combining Generative and Discriminative Models . . . . .	110
6.3.2	Task-Driven and Task-Independent Learning . . . . .	112
6.3.3	Structure Learning via an Incremental Strategy . . . . .	112
6.3.4	Co-occurrence Analysis . . . . .	114
6.4	Composing Features into Hierarchies . . . . .	116
6.4.1	Co-occurrence Learning Algorithm . . . . .	116
6.4.2	Local Feature Extraction . . . . .	118
6.4.3	Visual Classes . . . . .	120
6.4.4	Finding Correlated Feature Classes . . . . .	122
6.4.5	Estimating Spatial Relations . . . . .	131
6.4.6	Feature generation . . . . .	133
6.4.7	Adaptive Patch Features . . . . .	134
6.5	Discriminative Learning . . . . .	137
6.5.1	SVM for Graphical Models . . . . .	137
6.5.2	Combining SVM and Fisher score . . . . .	139
6.6	Discussion . . . . .	139

---

<b>7</b>	<b>Experimental Evaluation</b>	<b>143</b>
7.1	Datasets . . . . .	144
7.1.1	COIL-100 . . . . .	145
7.1.2	Ponce Group’s Object Recognition Database . . . . .	146
7.1.3	Butterflies . . . . .	146
7.1.4	Soccer . . . . .	149
7.2	View-Specific Object Recognition . . . . .	150
7.2.1	Experimental Protocol . . . . .	150
7.2.2	Parameters and Implementation . . . . .	151
7.2.3	Evaluation . . . . .	152
7.2.4	Discussion . . . . .	157
7.3	Multiple Viewpoint Object Recognition . . . . .	157
7.3.1	Experimental Protocol . . . . .	158
7.3.2	Parameters and Implementation . . . . .	158
7.3.3	Evaluation . . . . .	159
7.3.4	Discussion . . . . .	160
7.4	Object Class Recognition . . . . .	167
7.4.1	Experimental Protocol . . . . .	167
7.4.2	Parameters and Implementation . . . . .	168
7.4.3	Bag-of-Features . . . . .	173
7.4.4	Evaluation . . . . .	174
7.4.5	Discussion . . . . .	178
<b>8</b>	<b>Conclusions</b>	<b>191</b>
8.1	Summary of the Contributions . . . . .	191
8.2	Extensions . . . . .	193
8.2.1	Multidimensional Feature Hierarchies . . . . .	193
8.2.2	Reinforcement Learning . . . . .	194
8.2.3	Tracking with Feature Hierarchies . . . . .	195
8.3	Suggestions for Future Research . . . . .	196
8.3.1	Learning Pertinent Combinations . . . . .	196
8.3.2	Improving Feature Hierarchies . . . . .	197
8.3.3	Appearance Model . . . . .	199
<b>A</b>	<b>Kernel Density Estimation</b>	<b>237</b>

---

---

## List of Figures

---

1.1	Hierarchical decomposition of Leonardo da Vinci's Vitruvian Man . . .	5
1.2	The overall structure of our hierarchical object model . . . . .	6
2.1	Illustration of three different graphical model formalisms . . . . .	13
2.2	Pairwise Markov Random Field . . . . .	14
2.3	Message passing recursions underlying the Belief Propagation algorithm	17
3.1	Analysis of cornerness in the Harris detector . . . . .	29
3.2	Computation of the intrinsic scale for blob features . . . . .	32
3.3	Maximally Stable Extremal Regions (MSER) . . . . .	36
3.4	Intensity Extrema-Based Region Detector (IBR) . . . . .	38
3.5	Illustration of three strategies for spatial sampling . . . . .	41
3.6	Construction of SPIN images . . . . .	49
3.7	Scale Invariant Feature Transform (SIFT) . . . . .	52
3.8	Correspondence of triplets in affine transform related images . . . . .	55
3.9	Overview of the history of object recognition . . . . .	60
3.10	Object Recognition topologies under Graphical Model formalism . . .	68
3.11	Object Recognition topologies, a visual example . . . . .	70
4.1	Image sets used for evaluating feature detectors . . . . .	76
4.2	Performance evaluation of various type of detectors . . . . .	79
4.3	Performance evaluation of various type of descriptors . . . . .	82
5.1	The overall structure of our hierarchical object model . . . . .	86
5.2	Instances of two feature classes . . . . .	88
5.3	Spatial relation between a compound feature and its subfeatures . . .	90

---

5.4	Two-dimensional parametric spatial relation . . . . .	92
5.5	One-dimensional parametric spatial relation . . . . .	92
5.6	The proposed hierarchy under the graphical model formalism . . . . .	96
5.7	Visual interpretation of the mapping of a spatial relation . . . . .	99
5.8	Illustration of the message product during NBP . . . . .	105
5.9	Illustration of an upward message-passing iteration during NBP . . . . .	106
6.1	Overview of different aspects of our learning strategies . . . . .	111
6.2	Illustration of the feature extraction process . . . . .	118
6.3	Bayesian Information Criterion (BIC) . . . . .	121
6.4	Visualization of the number of co-occurrences between all the possible pairs of feature classes. . . . .	124
6.5	Orientation normalization of random patches . . . . .	127
6.6	Point density normalization for different feature instances . . . . .	128
6.7	Point density normalization for different feature instances . . . . .	129
6.8	Effect of the normalization method of the potential of a feature . . . . .	130
6.9	Creation of a new visual feature in the graphical model. . . . .	133
6.10	Creation of the observation potential in the graphical model. . . . .	134
6.11	Adaptive selection procedure . . . . .	136
6.12	Learning of a multi-class SVM classifier . . . . .	138
7.1	First 36 objects of the COIL-100 dataset . . . . .	145
7.2	18 training views of an object of COIL-100 . . . . .	146
7.3	Ponce Group’s Object Recognition Database: training images . . . . .	147
7.4	Ponce Group’s Object Recognition Database: test images . . . . .	147
7.5	The butterfly dataset . . . . .	148
7.6	The Soccer dataset . . . . .	149
7.7	Detection of five object models on a series of images differing in view- ing angle . . . . .	152
7.8	Detection results on a series of COIL-100 object . . . . .	153
7.9	Convergence of NBP under viewpoint variations . . . . .	154
7.10	Detection of partially occluded objects . . . . .	155
7.11	Inference of missing features during detection . . . . .	156
7.12	Four different hierarchies learned on COIL-100 objects . . . . .	160
7.13	Confusion matrices for one- to six-level models . . . . .	161
7.14	Extraction of random patches and learned visual codebooks . . . . .	162

---

7.15	Classification error on COIL-100 [NNM96] versus the mean number of feature classes at the top level. . . . .	164
7.16	ROC curves obtained by our hierarchical framework and two of the best state-of-the-art methods. . . . .	164
7.17	Illustration of spatial relations identified at the first level. . . . .	165
7.18	Recognition of two objects in Ponce’s object database [RLSP06] . . .	166
7.19	Local regions extracted from multiple feature detectors . . . . .	170
7.20	Adaptive Patch Features for different spatial relations . . . . .	172
7.21	Automatic threshold selection based on the Mutual Information for six visual class of the codebook . . . . .	175
7.22	Illustration of the “most useful” low-level features during detection . .	176
7.23	Detection of <i>Zebra</i> butterflies using our hierarchical model . . . . .	180
7.24	Detection of <i>Admiral</i> butterflies using our hierarchical model . . . . .	185
8.1	Two extensions of the hierarchical model . . . . .	198
8.2	Two extensions of the hierarchical model . . . . .	199
A.1	Illustration of the effect of the bandwidth size on the density during a Kernel Density Estimation (KDE) . . . . .	238
A.2	Three scenario that may occur during a Kernel Density Estimation (KDE) . . . . .	240

---



---

---

## List of Tables

---

6.1	Classification of the difference between two successive BIC values into 4 classes. . . . .	121
7.1	Classification results for the Soccer dataset . . . . .	177
7.2	Classification results for the Butterflies dataset . . . . .	178



---



---

## List of Algorithms

---

1	NBP update of an outgoing nonparametric message . . . . .	22
2	Harris-Laplace Detector . . . . .	33
3	Iterative Scale Selection( $\mathcal{P}_\sigma^x$ ) . . . . .	33
4	Affine Invariant Interest Point Detector (Simplified) . . . . .	34
5	Intensity-based Regions . . . . .	37
6	Randomized Grid Generation . . . . .	42
7	Scale Invariant Feature Transform (SIFT) $\{x, y, \sigma\}$ . . . . .	51
8	NBP update of an outgoing nonparametric message . . . . .	103
9	$\bar{\vartheta}_t^i = \text{pose}(\bar{x}_t^i, \{\mu_{k,t}\}_{i=1}^N, \{\vartheta_{k,t}\}_{i=1}^N)$ . . . . .	103
10	$\bar{\theta} \leftarrow \text{WeightedCircularMean}(\bar{x}, \{\mu\}_{i=1}^N, \{\theta\}_{i=1}^N)$ . . . . .	103
11	Co-occurrence Learning: learn() . . . . .	117
12	Correlation Extraction: extract( $\mathcal{G}$ ,level) . . . . .	123
13	Direction of the point density . . . . .	126
14	Fisher score for feature selection . . . . .	140



# Introduction

---

In this opening chapter, we present the objective of this thesis and we aim to provide intuitive answers to the following questions: Why is it worth doing research on visual recognition? What are the underlying mechanisms? Why is it not yet solved and very challenging? And finally, how do we approach the problem in this thesis and how is it organized?

## 1.1 Objective

Our goal is to develop a flexible representational framework for visual features. Such a representation should be able to reflect the aggregative nature of features and their spatial relations in a flexible way.

Specifically, we aim to exploit this representation in a system that would be able to identify previously learned object classes within images. Although promising attempts have been made to represent visual features in a generic way, there is still a lot of work ahead to exploit these models to recognize objects in real world conditions. Representation is still the main weakness of current approaches and the gap between low-level features and objects remains problematic; *How can a visual system exploit basic low-level features to identify higher level stimuli, such as objects?*

This question is the central theme around which this thesis is written. In addition, we also want to develop methods that can automatically learn the shared visual aspects between images of the same object class.

## 1.2 Motivation

Vision is one of the most extraordinary sense that we have been given. Man has, since the early times, exploited this ability to live and improve his living conditions; to find food and shelter, to mate, to communicate, to learn, to orient himself, to manipulate and construct tools, to build houses, to do art, etc.

Under the realm of visual perceptions, our world is a rich and complex source of information. Humans have a remarkable ability to exploit these perceptions daily. The human vision system is one of the most sophisticated visual recognition system known. It works so well in our everyday perceptual tasks that we do not have to “think” about it. Therefore it can be considered as a good source of inspiration for computer vision research. However, our visual system is not perfect. IBN AL-HAYTHAM mentioned in his *Book of Optics* (written around 1020) that personal experience has an effect on what people see and how they see. Our visual perceptions are subjective and can be affected by illusions. Moreover there are also physiological limitation factors; such as the field of view, the contrast sensitivity, the maximum perceivable resolution, the perceived colors, the maximum speed of moving objects, and the variations... For instance, the human eye is physically incapable of capturing a whole scene in full detail [Cat04].

Summing up, we live in a world in which many of our daily tasks involve visual recognition. The automation of some of these tasks is a long term goal in which computers will play a fundamental role. They will go beyond our visual abilities and perform much of our tasks swifter and with more reliability. Among the current trends of applications, we can mention; image retrieval (google image), video search, web search, security, online dating, airport baggage screening, helmet-mounted sight-and-display technology, devices for blind people, etc.

Therefore we can easily understand why computer vision research has become so popular. Nevertheless, despite more than fifty years of research in artificial intelligence, robotics and computer vision, many visual tasks that are easily performed by humans are still unsolved by machines. In the area of object recognition, several researchers agree that we do not seem to be closer to a generic solution nowadays than we were twenty years ago. Therefore it is a very challenging field of research.

In object recognition research, we aim at developing algorithms and representations that will allow a computer to autonomously analyze visual information to perform recognition tasks, with the belief that in doing so we will see a significant impact on our daily life.

## 1.3 Challenges

The area of object recognition is still unsolved and challenging. Because our visual world is constantly changing, an object recognition system has to be robust to some variations. The main variations a system should be able to deal with are the following;

1. Changes of aspect:

An image only captures a given viewpoint of the object. Many objects have different visual appearance depending on which angle they are seen. Different views of an object can look very different.

2. Viewpoint:

Objects can also be subject to in-plane transformations (translation, rotation, scaling, skews) and out-of-plane transformations (foreshortenings) that change their appearance.

3. Illumination:

Change in the lighting of the object can be artificial, natural, and induced by shadows. In all the case, the illumination variation will modify the pixel values in the image. Linear or non-linear transformations may be applied to rescale the pixel values.

4. Background clutter:

Another kind of variation can arise from the background. The background of the image may contain many distractors as well as other objects.

5. Occlusion:

Since we are interested in two-dimensional images of three-dimensional objects, self-occlusion is inevitable and has to be taken into account. Additionally, some parts of the object may be obscured by another object instance. This is another type of issue with the same effect: a part of the object is not visible.

6. Intra-class variation:

The object class itself can have a large degree of visual variability. The variability can take various forms: in the geometry, appearance, and/or texture.

## 1.4 Contribution

I believe that to make real progress in object recognition, we should not only focus on specific difficulties (*e.g.* varying illumination and viewpoint, occlusion) but we should also try to find more generic answers to more fundamental aspects of the problem. What are these fundamental aspects? By reviewing the research in the field, we can identify some questions that appear to be essential for visual recognition:

- *How to represent visual information? How to represent the appearance and the spatial structure of objects in a unified framework?*
- *How can the visual system be adaptive and learn the shared visual aspects of the instances of a given object class, and thus learn how to recognize objects?*
- *And finally how to propagate information from low-level observations (pixels) to create globally consistent scene interpretations?*

This thesis will focus on these questions. The main contribution will be to develop and present a new hierarchical model for visual features that we introduce in the following section.

### 1.4.1 Generic Feature Hierarchies

Digital images processed by computer systems are made up of a large set of pixels. In practice, instead of using them directly it is convenient to exploit a smaller number of so-called “visual features”. They generally abstract a set of visual properties (shape, color, or texture) computed locally or globally. Parts, objects, and scenes can be defined in terms of their visual features. It is common for recognition systems to represent an object as a set of visual features arranged spatially.

If we turn to the nature of objects, we see that many real world object classes exhibit a hierarchy in the structure of their parts. For instance, a face is made up of two eyes, each eye is made up of a eyeball, each eyeball is made up of an iris, etc. Another illustration is shown in Figure 1.1. Such a hierarchy is also reflected in the human visual system. A large body of evidence suggests a gradual increase in both the invariance properties and the complexity of the preferred stimuli of neurons along the visual stream.

These observations constitutes strong motivations to design a hierarchical model of features. In this thesis, we define such an object model which consists of a set



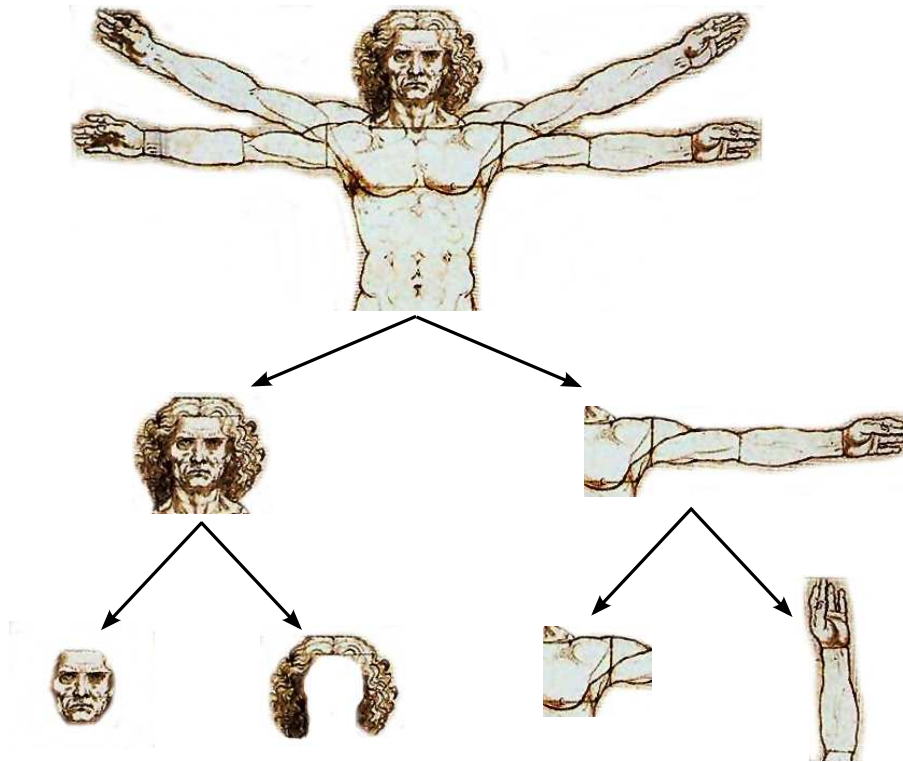


Figure 1.1: Leonardo da Vinci’s Vitruvian Man (1492) is decomposed into a hierarchy of parts. To be generic, the hierarchy should exhibit some invariance in the appearance of the parts and some flexibility for their relative positions.

of generic features organized in a hierarchy (see Figure 1.2). At the bottom level lie *primitive features* that are directly extracted by feature detectors. Subsequent higher levels consist of spatial compositions of more elementary features. These high-level features, or *compound features*, depict the relative configuration of two features from a lower level of the hierarchy. Nodes represent features and edges correspond to spatial relations between them.

The use of the graphical model formalism to represent the hierarchy allows us to pose detection as an inference process and to use Nonparametric Belief Propagation (NBP) to propagate evidence.

The structure of the graphical model itself is constructed by a co-occurrence learning algorithm in an iterative, bottom-up fashion. At each level of the hierarchy, pairs of features are identified that tend to occur at stable positions relative to each other, by clustering the configurational distributions of observed feature co-occurrences using the Expectation-Maximization algorithm.

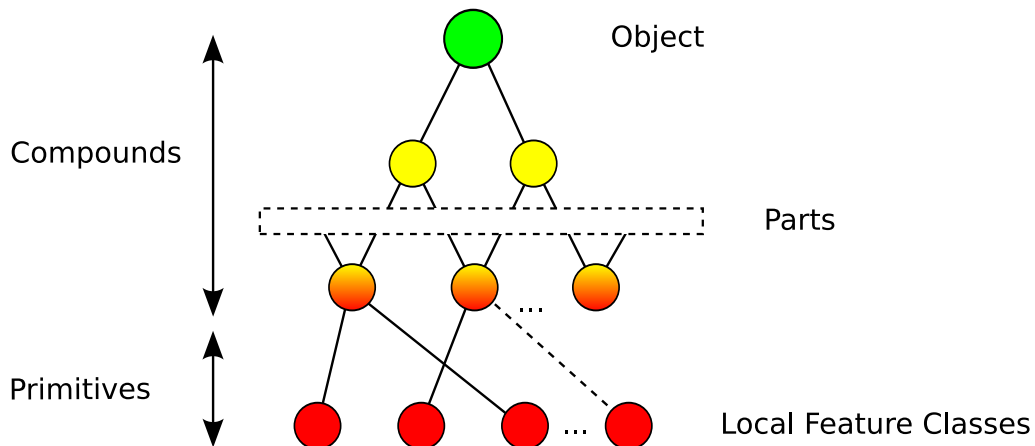


Figure 1.2: The overall structure of our hierarchical object model. Visual feature classes are represented by nodes, and edges are used to define their relative positions. We distinguish between primitives (in red) that correspond to low-level features and compound features that are constrained by spatial relations with lower level features.

## 1.5 Outline

The thesis is organized as follows: Chapter 2 starts by presenting prerequisites on different graphical model formalisms. We examine their properties and their use to perform inference via iterative message-passing algorithms; Belief Propagation and its nonparametric form (NBP) are successively explained.

Chapter 3 reviews existing work in the field of object recognition. We present an extensive discussion on the feature detectors and descriptors that produce the basic inputs to these models. Their performance in terms of repeatability across different image variations are evaluated in Chapter 4.

Chapter 5 introduces a new statistical model for representing visual features that can be used to perform object recognition. This model takes the form of a hierarchy of visual feature classes where the geometrical structure of the object is represented by local spatial relations (Figure 1.2).

Chapter 6 presents a co-occurrence learning method from which both the structure and the parameters of the hierarchy can be estimated. The proposed learning framework composes the hierarchical model iteratively in a bottom-up manner.

Finally, Chapter 7 investigates the behavior of our feature hierarchies across a variety of object recognition datasets. These experimental evaluations are organized around three recognition tasks of increasingly difficulty.

# Prerequisites

---

Statistical methods play a major role in the design of current approaches to object recognition. Graphical model formalisms provide a powerful framework that is general enough to deal with a wide variety of applications. Their strength relies in their capacity to encode the statistical structure of a problem in a sparse and flexible manner. Furthermore, they allow us to exploit standard learning strategies and efficient inference algorithms.

In this thesis, graphical models will be applied to modeling spatial relations between visual features at the different levels of abstraction. The ultimate goal is to provide a statistical framework to perform object recognition.

Depending on the nature of the problem, graphical models may appear under different forms in the literature. In this background chapter, we start by reviewing different graphical model formalisms in Section 2.1. Then, in subsequent sections, we show how these graphs can be used to perform inference via iterative message-passing algorithms; Belief Propagation (Section 2.2) and its non-parametric form (NBP) (Section 2.3) are successively presented. These are particularly emphasized because the framework developed in this thesis will use them to detect objects.

## 2.1 Graphical Models

Graphical models provide a general and powerful method for encoding the statistical structure of a set of random variables. Their genericity makes them convenient to represent a variety of problems in computer vision. M. I. JORDAN [Jor99] gives a concise and insightful introduction to graphical models:

“*Graphical models are a marriage between probability theory and graph theory. They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering – uncertainty and complexity – and in particular they are playing an increasingly important role in the design and analysis of machine learning algorithms. Fundamental to the idea of a graphical model is the notion of **modularity** – a **complex system is built by combining simpler parts**. Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data. The graph theoretic side of graphical models provides both an intuitively appealing interface by which humans can model highly-interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms.*

”

In other words, graphical models offer several useful properties to represent complex systems in probabilistic terms. If considered under the point of view of object recognition, these properties are particularly relevant. Indeed, the modularity of graphical models allows to represent object models in terms of object parts that are often simpler to manipulate. Therefore it is not surprising that object recognition frameworks in the literature tend to make use of graphical models (state-of-the-art approaches will be reviewed in Chapter 3).

The objective of this section is to focus on the key aspects offered by graphical models. A short introduction will first be given to the basis required to understand the model presented in this work. More complete discussions about graphical models can be found in several books [Whi90, Lau96, Jor99, Bis06].

The following subsections will introduce and compare different families of graphical models. For a better understanding, graphical models are grouped into two different classes: undirected and directed, that are described respectively in Section 2.1.2 and Section 2.1.3.

### 2.1.1 Draw me a Graph

We begin by presenting basic concepts from graph theory that are useful in describing graphical models. Intuitively, a graph comprises nodes (*i.e.* vertices) that are connected by edges (*i.e.* arcs, links). In this work, we distinguish between three different types of graphs: undirected graphs, directed graphs and hypergraphs:

**Definition 2.1.** A *graph*  $\mathcal{G}$  is a mathematical structure  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  composed of two sets: a finite set of nodes  $\mathcal{V}$  and a set of edges  $\mathcal{E}$ .

**Definition 2.2.** An *undirected graph* is a graph  $\mathcal{G}$  in which each *edge*  $e \in \mathcal{E}$  is defined as an *unordered* pair of nodes  $\{u, v\}$ ,

$$\mathcal{E} \subseteq \{\{u, v\} | u, v \in \mathcal{V}\} \quad (2.1)$$

**Definition 2.3.** A *directed graph* is a graph  $\mathcal{G}$  in which each *edge*  $e \in \mathcal{E}$  is defined as an *ordered* pair of nodes  $(u, v)$ . For simplicity, the set of directed edges is denoted  $\vec{\mathcal{E}}$ ,

$$\vec{\mathcal{E}} \subseteq \{(u, v) | u, v \in \mathcal{V}\} \quad (2.2)$$

**Definition 2.4.** A *hypergraph* is a graph  $\mathcal{G}$  in which each *edge*  $e \in \mathcal{E}$  is defined as an *unordered* and *nonempty* set of nodes  $\{v_1, \dots, v_n\} \subseteq \mathcal{V}$ ,

$$\mathcal{E} \subseteq \{\{v_1, \dots, v_n\} | v_1, \dots, v_n \in \mathcal{V}\} \quad (2.3)$$

**Definition 2.5.** In an undirected graph, a *clique* is a set of nodes  $C \subseteq \mathcal{V}$  for which every pairs  $\{u, v\} \in C$  are connected by an edge. If the entire graph forms a clique, it is said to be *complete*. A clique  $C$  is called *maximal* if no other node can be added such that the set remains a clique,  $\nexists C' \subseteq \mathcal{V} : C \subset C'$  and  $C'$  is a clique.

In directed graphs, we find the following additional elements:

**Definition 2.6.** A node  $v_p$  is said to be a *parent* of a node  $v$  if there exists one directed edge from  $v_p$  to  $v$ ,  $(v_p, v) \in \vec{\mathcal{E}}$ .

**Definition 2.7.** A node  $v_c$  is said to be a *child* of a node  $v$  if there exists one directed edge from  $v$  to  $v_c$ ,  $(v, v_c) \in \vec{\mathcal{E}}$ .

## From Graph Theory to Graphical Models

In probability theory and statistics, a graphical model defines the independence structure between a set of random variables by a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  in which each node  $v_i \in \mathcal{V}$  is associated to a random variable  $x_i$ , and the edges  $e_i \in \mathcal{E}$  between the nodes express relationships between these variables. Graphical models also allow to use different types of variables within the same model; following their nature, they can be either directly observable  $y_i$  or hidden  $x_i$ , continuous or discrete. From a global point of view, the graph represents a decomposition of the joint, global distribution into a product of factors each depending only on a subset of the variables.

Intuitively, the three types of graph that have been presented in the preceding section have a direct correspondance in terms of graphical models; Undirected and directed graphical models, and Factor graphs (as illustrated in Figure 2.1). From these three types, we can extract two main families depending if they contain directed or undirected edges. In the next sections, we provide a brief introduction to them.

### 2.1.2 Undirected Graphical Models

This section introduces undirected graphical models, from which three main families of graph can be identified; Markov Random Fields (MRFs), Pairwise Markov Random Fields (PMRFs) and Factor Graphs. In the following paragraph, we show how graph separation and conditional independence are used to encode the Markov properties of the random variables.

#### Graph Separation

Graph separation and conditional independence are essential properties of graphical models that allow to factorize the computation of the complete joint distribution of a set of random variables.

**Theorem 2.8.** Given an undirected graphical model that associates a random variable  $x_i \in X$  to each vertice  $i \in \mathcal{V}$  in the undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ,

- Let  $f, g$  and  $h$  denote three disjoint subsets of nodes  $f, g, h \subset \mathcal{V}$ .

Set  $h$  is said to *separate* sets  $f$  and  $g$  if every path connecting  $f$  and  $g$  passes through some node in  $h$ .

Under these conditions, the probability distribution of the random variables  $X_f$ ,  $X_g$  conditioned on the separating set  $X_h$  can be factorized as:

$$p(X_f, X_g | X_h) = p(X_f | X_h) p(X_g | X_h) \quad (2.4)$$

The two sets of random variables,  $X_f$ ,  $X_g$ , are said to be *conditionally independent* given  $X_h$ . This property plays a fundamental role in graphical models and Markov Random Fields (MRFs). It allows for global inference using only local computations.

### Markov Random Fields

A *Markov Random Field* (MRF) [Spi71, Pre74], also known as *Markov network*, is an undirected graphical model that characterizes the joint probability distribution of a set of random variables  $\{x_1, \dots, x_n\} \in X$  via the notion of conditional independencies.

A probability distribution  $p(x_i)$ , where  $x_i \in X$ , is said to define a Markov random field if it depends only on the knowledge of the outcome of its neighbors [KS80],

$$p(x_i | X_{/i}) = p(x_i | \{x_j \in \mathcal{N}_i\}) \quad (2.5)$$

where  $X_{/i}$  represents all the random variables in the graph except  $x_i$  and  $\mathcal{N}_i \subset X$  is the set of neighbors of  $x_i$ .

MRFs have been popularized in the 1970s when the Hammersley-Clifford theorem [Cli90] demonstrated a method for constructing them. This theorem relates the conditional independence structure specified by a graph to the distribution of the random variables. It shows that a probability distribution can naturally be parameterized via *potential functions* defined on the *cliques* of the undirected graph.

Let  $C$  denotes a clique,  $X_C$  the set of variables in that clique and  $\mathcal{K}$  the set of cliques of the undirected graph  $\mathcal{G}$ .

**Definition 2.9.** A *potential function* for a clique  $C$  is a non-negative function  $\psi_C(X_C)$  of all possible realizations of its variables  $X_C = \{x_j, j \in C\}$ .

**Theorem 2.10.[Hammersley-Clifford.]** A probability distribution defined as a normalized product of positive potential functions  $\psi_C$  on those cliques  $C \in \mathcal{K}$  is always Markov with respect to the graphical model:

$$p(X) = \frac{1}{Z} \prod_{C \in \mathcal{K}} \psi_C(X_C) \quad (2.6)$$

where  $Z$  is a normalizing constant,

$$Z = \sum_{x_i \in X} \prod_{C \in \mathcal{K}} \psi_C(X_C) \quad (2.7)$$

Summarizing graph structure and Hammersley-Clifford theorem lead to the following definition.

**Definition 2.11.** A *Markov random field* consists of an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where each vertex  $i \in \mathcal{V}$  is associated to a random variable  $x_i \in X$  and each edge  $\{u, v\} \in \mathcal{E}$  represents a dependency between two random variables  $u$  and  $v$ . A set of potential functions  $\psi_C$  are defined for each clique  $C \in \mathcal{K}$  in the graph.

### Pairwise Markov Random Fields

In many applications, potential functions defined over a large set of random variables may become intractable in Markov Random Fields (MRFs). Therefore, it is convenient to consider a subclass of Markov random fields, called *Pairwise Markov Random Fields* (PMRFs).

Unlike MRFs, that define potential functions on cliques of any size, pairwise MRFs factorize the joint probability distribution in terms of pairwise potential functions, that are defined only between pairs of neighboring nodes. Because pairs of neighboring nodes always define cliques, the Hammersley-Clifford Theorem [Cli90] guarantees that pairwise MRFs are Markov with respect to the graphical model.

The joint probability distribution  $p(X)$  of a set of random variables  $X$ , corresponding to a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , can be decomposed in terms of pairwise potentials:

$$p(X) = \prod_{i \in \mathcal{V}} \phi_i(x_i) \prod_{\{i,j\} \in \mathcal{E}} \psi_{ij}(x_i, x_j) \quad (2.8)$$

where  $\phi_i(x_i)$  is the local potential, and  $\psi_{ij}(x_i, x_j)$  is the pairwise potential between random variables  $x_i \in X$  and  $x_j \in X$ .

Until here we have not mentioned observable random variables. In most applications, inference can be posed as the estimation of a set of hidden random variables, denoted  $X$ , based on observations  $Y$ . This estimation amounts to computing the posterior distribution  $p(X|Y)$ , where  $Y$  represents the set of observed random variables. Similarly to hidden nodes  $x_i \in X$ , each observed random variable  $y_j \in Y$  has a corresponding node  $j \in \mathcal{V}$  in the graph  $\mathcal{G}$  (an example of PMRF is given in Figure 2.2).



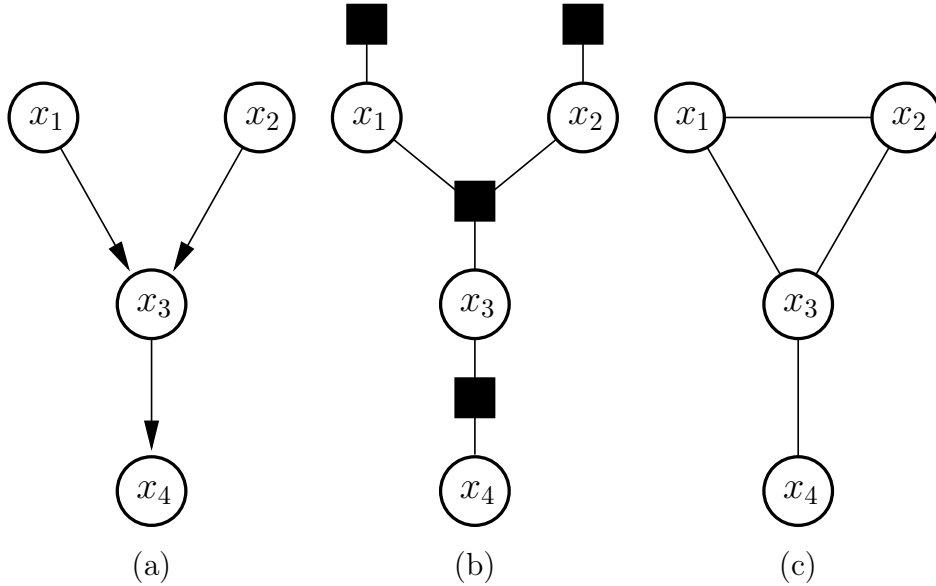


Figure 2.1: Three graphical representations of a distribution over four random variables (see [KFL01]). (a) A directed graph. (b) A Factor graph expressing the factorization underlying  $\mathcal{G}$ . (c) An undirected graph capturing the Markov structure of  $\mathcal{G}$ .

When observations are available, pairwise MRFs can be used to express the internal structure of the desired posterior distribution  $p(X|Y)$ . This is done by introducing local observations  $y_i$  in the probability model through local potential functions  $\phi_i(x_i, y_i)$  that link a hidden random variable  $x_i \in X$  to a corresponding observed random variable  $y_i \in Y$ .

Given an undirected graph, a pairwise MRF factorizes the joint distribution as a product of potential functions intuitively defined on that graph's edges:

$$p(X|Y) = \prod_{i \in \mathcal{V}} \phi_i(x_i, y_i) \prod_{\{i,j\} \in \mathcal{E}} \psi_{ij}(x_i, x_j) \quad (2.9)$$

Pairwise MRFs are widely used in many computer vision applications such as image restoration, denoising, and segmentation.

### Factor graph

Markov Random Fields seek to represent factorized probability distributions. Another means to describe global distributions in a graphical model form is to use a Factor graph [KFL01]. Factor graphs aim at generalizing the notion of factors (*i.e.* potentials) by introducing the factorised distribution into the graph structure.

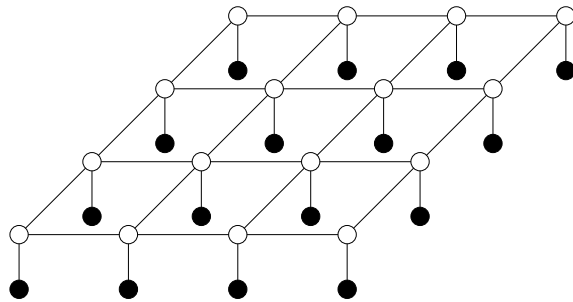


Figure 2.2: Pairwise Markov Random Field. The black circles correspond to observations and the white circles to the hidden variables.

Whereas MRFs defined potentials on cliques and PMRFs on edges, Factor graphs represent them explicitly through *hyperedges*. This allows to avoid ambiguous relations that may occur in MRFs (as mentioned in [Sud06]).

In contrast with MRFs and pairwise MRFs, Factor graphs define the joint distribution as a normalized product of local potential functions  $\psi_f(x_f)$  defined on hyperedges  $f \in \mathcal{E}$ :

$$p(X) = \prod_{f \in \mathcal{E}} \psi_f(x_f) \quad (2.10)$$

$$\Leftrightarrow p(X) = \psi_{12}(x_1, x_2) \psi_{123}(x_1, x_2, x_3) \psi_{23}(x_2, x_3) \quad (2.11)$$

An example of Factor graph constructed from a directed graph is illustrated in Figure 2.1. In that figure, the circles are variable nodes, and the black boxes are factor nodes.

### 2.1.3 Directed Graphical Models

The second main class of graphical models is commonly called *Directed Graphical Models*. In the literature, most common instances of this model are referred to as a *Bayesian networks*, or *belief networks* [Win72, Har83]. Typically, a Bayesian network is a acyclic directed graph (DAG) which is used to represent conditional dependence or independence assumptions over a set of random variables.

Similarly to undirected graphical models, nodes are used to depict the random variables of the system. The main difference lies in the use of directed edges to model the statistical relationships between variables. In a Bayesian network, directed edges are used to represent conditional dependence assumptions.

Instead of using a product of local potentials to define the joint probability distribution of the random variables, the factorization is now defined as the product of the conditional distributions of each node given its parents (Equation 2.12). In other words, the joint probability distribution  $p(X)$  is factorized as

$$p(X) = \prod_{i \in \mathcal{V}} p(x_i | x_{\rho_i}) \quad (2.12)$$

where  $x_i \in X$  is the random variable of the node  $i$ ,  $\rho_i$  denotes the set of parents of node  $i \in \mathcal{V}$  and  $x_{\rho_i}$  represents its corresponding set of random variables. Note that if a node  $i$  has no parent, such that  $\rho_i = \emptyset$ , then the distribution can be written as

$$p(x_i | x_{\rho_i}) = p(x_i) \quad (2.13)$$

Bayesian networks have been used in object recognition [PG00b], gene regulatory networks, medicine, engineering, text analysis, image processing, data fusion, and decision support systems.

### 2.1.4 Inference on Graphical Models

A strength of the Graphical Models presented above is that they provide a convenient way to factorize and represent complex systems in terms of simpler local functions. Interestingly, this property facilitates solving of complex inference problems that arise in many computer vision applications.

As it has been mentioned previously, a typical occurring inference scenario is to assume that some of the nodes  $y_i \in Y$  in the graph can be observed in the environment. The goal is to compute the posterior marginal distributions  $p(x_i|Y)$  of one or more unobserved random variables  $x_i \in X$ ,

$$p(x_i|Y) = \int_{\mathcal{X}_{\setminus x_i}} p(X|Y) d\mathcal{X}_{\setminus x_i} \quad (2.14)$$

where  $\mathcal{X}_{\setminus x_i}$  represents the joint space made of all the random variables in  $X$  except  $x_i$ .

In many cases, the naive exact inference of problems arising in graphical models becomes quickly intractable. By analyzing this exact computation [YFW03], we can observe that many computations are repeated. In many cases, it is possible to efficiently perform the inference by reusing intermediary factors and thus computing the posterior marginals in an incremental fashion. This idea leads to more efficient solutions to the exact inference problem.

The most popular algorithm to perform global inference efficiently through local computations is known as Belief Propagation (BP) [LS88, Pea88, SS90]. In the following sections, we will focus on presenting the standard BP algorithm (Section 2.2) and its stochastic extension, called Nonparametric Belief Propagation (NBP) (Section 2.3). As we shall see, the structure of the graph can be exploited by inference algorithms and to make the structure of those algorithms transparent.

For each kind of graphical model, it is possible to find the corresponding BP algorithm [YFW03]. We will consider BP applied to the pairwise MRF's, because it has only one kind of message, while the BP algorithms for the other graphical models are generally described using two kinds of messages (upward and downward).

## 2.2 Belief Propagation

Belief Propagation, also called sum-product algorithm, is an efficient iterative algorithm for computing marginal probability distribution  $p(x_i)$  of each random variable  $x_i \in X$  of the graphical model. It was independently formulated by LAURITZEN *et al.* [LS88] and PEARL [Pea88].

BP takes the form of a message-passing algorithm between hidden nodes (*i.e.* hidden random variables), where at each iteration, each node calculates messages to send to its neighbors. A message is depicted by a variable of the same dimensionality as the destination hidden node. Intuitively, a message  $m_{ij}(x_j)$  sent from a hidden node  $i$  to the hidden node  $j$  is proportional to how likely node  $i$  thinks (or believes) it is that node  $j$  will be in the corresponding state. Following the standard BP notation for pairwise MRFs, a message  $m_{i,j}(x_j)$  from hidden node  $i$  to  $j$  is written:

$$m_{i,j}(x_j) \leftarrow \int_{\mathcal{X}_i} \psi_{i,j}(x_i, x_j) \phi_i(x_i, y_i) \prod_{k \in \mathcal{N}_i \setminus j} m_{k,i}(x_i) dx_i \quad (2.15)$$

where  $\mathcal{N}_i \setminus j$  is the set of neighbors of node  $i$  where node  $j$  is excluded,  $\psi_{i,j}(x_i, x_j)$  is the pairwise potential between random variables  $x_i, x_j$ , and  $\phi_i(x_i, y_i)$  is the local potential between the hidden and the observable random variables  $x_i, y_i$ .

The messages received by each node over its edges are combined at each iteration to compute incremental updates of marginal distribution estimates  $b_i(x_i)$  (Equation 2.16). These estimates are referred to as *beliefs*, by analogy with expert systems developed in the artificial intelligence community [Pea88],

$$b_i(x_i) \leftarrow \phi_i(x_i, y_i) \prod_{k \in \mathcal{N}_i} m_{k,i}(x_i) \quad (2.16)$$

If the graph is a polytree, it can be shown that the *beliefs*  $b_i(x_i)$  will eventually be equal to the exact posterior marginal distributions at each node of the graph (Equation 2.17). This will occur after  $n$  message-passing operations, where  $n$  is proportional to the longest path in the graph.

$$\forall x_i \in \mathcal{G} : b_i^n(x_i) = p(x_i|Y) \tag{2.17}$$

if no loops in  $\mathcal{G}$

In Figure 2.3, we summarize the computation of a BP message update, and the corresponding message products which provide marginal distributions.

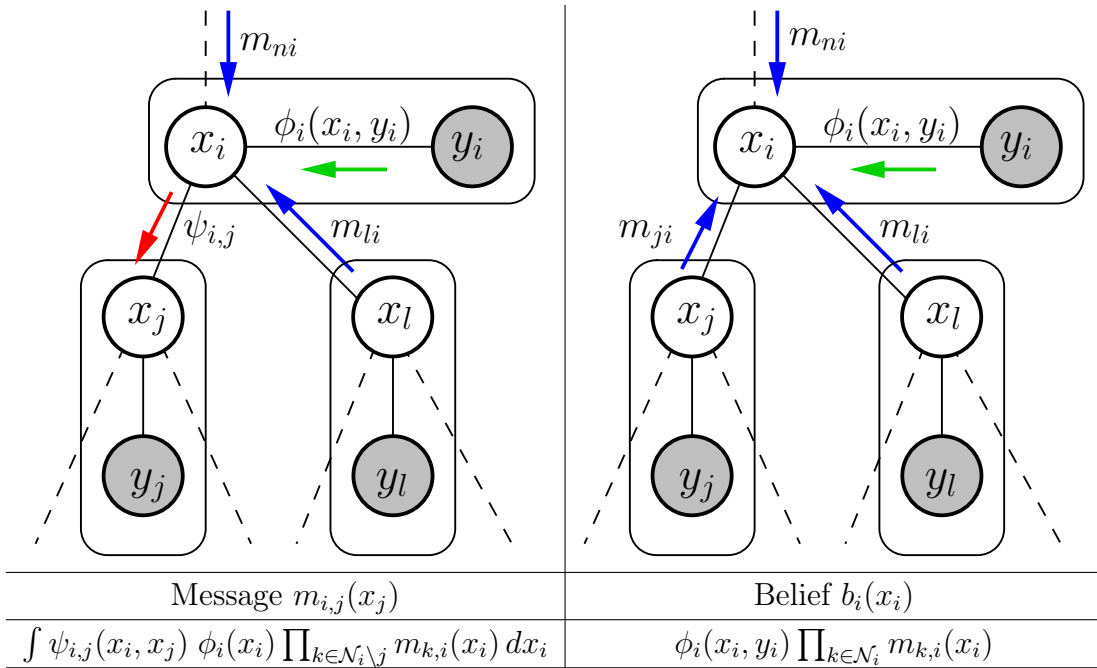


Figure 2.3: Message passing recursions underlying the BP algorithm. Left: A new outgoing message (red) is computed from all other incoming messages (blue) together with the local observation (green). Right: Marginal probability distribution estimates are determined from the product of the local observation potential  $\phi_i(x_i, y_i)$  with messages sent from neighboring nodes.

### 2.2.1 BP in Graphs with Cycles

The BP algorithm [Pea88] was originally designed to be performed on polytrees. On graphs with cycles, it is not guaranteed that the beliefs computed by BP will eventually converge to the true marginals  $p(x_i|y)$ . However, several methods have been proposed to perform inference on graph with cycles.

Rather than using these graph transforms, we focus on an alternative method known as loopy belief propagation [Pea88].

As it has been shown previously, BP algorithm proceeds entirely via a series of local message updates and can produce the exact posterior marginals after a given number of iterations. Given a graph with cycles, loopy BP iterates a parallel form of these message updates. Typically, the iteration process is simply repeated for a pre-defined number of iterations. In many applications, this straightforward method converges to beliefs which often closely approximate the true posterior marginals [FM98, MWJ99, WJW02, WF01]. However, the derivation of BP provides no justification for loopy BP, other than the intuition that messages are computed locally and that it should work well. For a stronger conceptual basis and detailed reasoning, it is possible to justify loopy Belief Propagation by considering its variational interpretation [Sud06].

### 2.2.2 Implementation Issues

In general, most common applications perform exact BP inference on discrete-valued or Gaussian random variables. In the discrete case, the belief  $b_i(x_i)$  takes the form of a vector of probabilities corresponding to an estimate of the discrete probability density function. Similarly, the potential function  $\psi_{ij}$  may be written as a matrix, and the convolution as a matrix-vector product [Pea88].

However, in many problems, variables are formulated as continuous-valued, possess non-linear relationships and non-Gaussian uncertainty. In these cases, Gaussian approximations may not be acceptable, and discretization of the state space may result either in lack of precision, or become computationally intractable due to the large state space.

Until recently, it was not possible to apply BP on high-dimensional data. It is now possible to apply Nonparametric Belief Propagation (NBP). This technique combines the advantages of nonparametric representation and message-passing inference algorithm. It is described in the following section.

## 2.3 Nonparametric Belief Propagation

Several nonparametric extensions of Belief Propagation (BP) algorithm [Pea88] have been proposed recently in the literature [Isa03, SIFW03, HW04, HYW05]. In this section, we focus on the first two nonparametric approaches that have been formulated; PAMPAS [Isa03] and Nonparametric Belief Propagation (NBP) [SIFW03]. These algorithms have been proposed in a short period of time and only differ in minor details. They are often referred to as NBP in the literature. These nonparametric approaches allow to solve high-dimensional, non-Gaussian inference problems in an efficient and statistically well founded way.

The main concept behind NBP algorithm is to exploit a nonparametric representation in order to approximate arbitrary continuous distributions that occur in messages and belief update operations. To better illustrate the specificities of NBP, the computation of message updates is divided in two phases:

$$m_{t,s}(x_s) \leftarrow \underbrace{\int_{\mathcal{X}_t} \psi_{t,s}(x_t, x_s)}_{(2)} \underbrace{\phi_t(x_t, y_t) \prod_{k \in \mathcal{N}_t \setminus s} m_{k,t}(x_t)}_{(1)} dx_t \quad (2.18)$$

First, the product of incoming messages  $m_{i,t}(x_t)$  is combined with the local observation potential  $\phi_t(x_t)$  to define (1) the distribution of the intermediate random variable  $\beta_{ts}(x_t)$ <sup>1</sup>:

$$\beta_{ts}(x_t) = \phi_t(x_t) \prod_{i \in \mathcal{N}(t) \setminus s} m_{i,t}(x_t) \quad (2.19)$$

Second,  $\beta_{ts}(x_t)$  is transformed via the pairwise potential  $\psi_{ts}(x_t, x_s)$  and integrated (2) to produce the message  $m_{ts}(x_s)$  sent to the destination node  $s$ :

$$m_{t,s}(x_s) = \int_{\mathcal{X}_t} \psi_{ts}(x_t, x_s) \beta_{ts}(x_t) dx_t \quad (2.20)$$

Focusing on these two steps of a local message update, we describe in the next subsections the strategy employed by NBP to keep the message update operations tractable. Specifically, we describe the message representation (Section 2.3.1), the computation of the product of Gaussian Mixtures (Section 2.3.2), and the propagation of messages (Section 2.3.3). Finally, the general NBP message update algorithm is presented (Section 2.3.4).

<sup>1</sup>Please note that this intermediate variable is not equivalent to the local belief  $b_i$  (Equation 2.16) since the incoming message from  $s$  is not taken into account.

### 2.3.1 Message representation

The representational approach in NBP is directly inspired by Particle Filtering methods. Each message is approximated by a set of samples. However, despite its name, NBP is not fully nonparametric (but rather semi-parametric) since each sample is associated with a weighted regularizing kernel. This is done to make message products well defined, and to make the integral of the messages tractable.

A message sent from  $t$  to  $s$  is approximated as a mixture of  $M$  Gaussian kernels:

$$m_{t,s}(x_s) = \sum_{i=1}^M w_s^{(i)} \mathcal{G}(x_s; \mu_s^{(i)}, \Sigma_s) \quad (2.21)$$

Here  $\mathcal{G}(\cdot; \cdot)$  is the Gaussian density function,  $w_s^{(i)}$  is the weight associated with the  $i$ -th kernel mean  $\mu_s^{(i)}$  and  $\Sigma_s$  is the variance parameter that is also called bandwidth. The weights  $w_s^{(i)}$  are normalized to sum to 1.

If the bandwidth parameter is held constant across all the samples of the density estimate, it is said to be *fixed*. However, estimating multimodal densities with a fixed bandwidth kernel may affect the quality of the estimation. Therefore, in order to better capture the distribution it is possible to use a different bandwidth at each sample, in this case it is said to be *variable*.

The distributions in Nonparametric Belief Propagation [SIFW03] make use of a fixed bandwidth whereas the PAMPAS algorithm [Isa03] associates a different bandwidth to each kernel center. Different methods to estimate this bandwidth are described in Appendix A.

### 2.3.2 Message product

As each message is represented by a Gaussian mixture (Equation 2.21), the computation of the message product (Equation 2.19) amounts to computing a product of Gaussian densities.

Furthermore, assuming that the local observation potentials are also defined as weighted Gaussian mixtures, the product leading to the intermediate variable  $\beta_{ts}$  may be computed exactly and then resampled to a reasonable size.

The product of  $d$  Gaussian densities [SIFW03] of mean  $\mu_j$  and covariance  $\Sigma_j$  is a Gaussian of mean  $\bar{\mu}$  and variance  $\bar{\Sigma}$  given by

$$\bar{\Sigma}^{-1} = \sum_{j=1}^d \Sigma_j^{-1} \quad \bar{\Sigma}^{-1} \bar{\mu} = \sum_{j=1}^d \Sigma_j^{-1} \mu_j \quad (2.22)$$



The product of  $d$  Gaussian mixtures of  $M$  components is a Gaussian mixture of  $M^d$  components. The weight  $\bar{w}$  associated with component  $\mathcal{G}(x; \bar{\mu}, \bar{\Sigma})$  of the product mixture is

$$\bar{w} \propto \frac{\prod_{j=1}^d w_j \mathcal{G}(x; \mu_j, \Sigma_j)}{\mathcal{G}(x; \bar{\mu}, \bar{\Sigma})} \quad (2.23)$$

where  $\{w_j : j = 1, \dots, d\}$  are the weights associated with the  $d$  factor Gaussians of  $\mathcal{G}(x; \bar{\mu}, \bar{\Sigma})$ .

The exact computation of a product requires explicit computation of  $M^d$  components; the cost of this operation is therefore exponential in the number of components. To avoid a dramatic increase in the number of components during inference, NBP proposes to use a sampling method to reduce the  $M^d$  components.

Moreover, NBP demonstrated that another strategy could be used to compute the message product. Instead of computing the exact product and then resampling, it is possible to compute an approximation by drawing  $M$  independent samples without computing the  $M^d$  components explicitly. This is done using a Gibbs sampler [GG84].

### 2.3.3 Message propagation

Intuitively, the propagation step (Equation 2.18 (2)) of a message  $m_{ts}(x_t)$  can be thought of as a probabilistic mapping of the belief available at the source node  $x_t$  to the destination node  $x_s$ . Given a set of samples  $x_t^{(i)}$  obtained from the product of incoming messages at  $x_t$ , NBP has to convolve these beliefs  $x_t^{(i)}$  with the pairwise potential  $\psi_{ts}(x_t, x_s)$  (Equation 2.20). To do so, the algorithm stochastically approximates this convolution, and thus provides a consistent nonparametric estimate of the outgoing message [Ihl05].

The way the approximation of this stochastic convolution (Equation 2.20) is computed depends on the type of the pairwise potential (parametric, non-parametric, analytic, ...). In the general case, NBP requires the pairwise potential  $\psi_{ts}(x_t, x_s)$  to be decomposed into its marginal influence on  $x_t$  and the conditional distribution  $\psi_{ts}(x_s|x_t)$  it defines between  $x_t$  and  $x_s$ . A commonly used hypothesis [SIFW03, Isa03, Tam05] to simplify this decomposition is to assume that the marginal influence defined by the potential on  $x_t$  is constant and can therefore be neglected. Interestingly, this will be the case in the model presented in this thesis.

Under these assumptions, the stochastic convolution is completed by drawing a sample  $x_{ts}^{(i)}$  from the conditional  $\psi_{ts}(x_s|x_t^{(i)})$  obtained for each  $x_t^{(i)}$ . Finally, the

last step is to select a kernel bandwidth  $\Sigma_{ts}$  to obtain the nonparametric density estimate <sup>2</sup>.

### 2.3.4 NBP Algorithm

We provide in Algorithm 1 [SIFW03] a summary of the main operations performed during the NBP message update and that have been described above.

---

**Algorithm 1** NBP update of an outgoing nonparametric message

---

Given input messages  $m_{kt}(x_t) = \{\mu_{k,t}^i, \Sigma_{k,t}^i, w_{k,t}^i\}_{i=1}^M$  received from nodes  $k \in \mathcal{N}_t \setminus u$

1. (a) // *Compute The Exact Incoming Message Product*  

$$\beta_{ts}(x_t) \leftarrow \phi_t(x_t) \prod_{i \in \mathcal{N}(t) \setminus s} m_{it}(x_t)$$
// *Draw samples*  
Draw  $M$  weighted samples  $\{\bar{x}_t^i, \bar{\Sigma}_t^i, \bar{w}_t^i\}_{i=1}^M$  from the product  $\beta_{ts}(x_t)$   
or  
(b) // *Compute The Gibbs Incoming Message Product*  

$$\{\bar{x}_t^i, \bar{\Sigma}_t^i, \bar{w}_t^i\}_{i=1}^M \leftarrow \text{Gibbs}(\phi_t(x_t) \prod_{i \in \mathcal{N}(t) \setminus s} m_{it}(x_t))$$
  2. // *Map the Potential*  

$$\{x_{tu}^i, w_{tu}^i\} = \text{apply } \{\bar{x}_t^i, \bar{w}_t^i\} \text{ on the potential } \psi_{t,u}$$
  3. // *Adjust the Kernel bandwidth (see Appendix A)*  
    - (a) // *fixed bandwidth*  

$$\Sigma_{tu} = kde(x_{tu})$$

$$\forall i \in [1 \dots M], \Sigma_{tu}^i = \Sigma_{tu}$$
or  
(b) // *variable bandwidth*  

$$\forall i \in [1 \dots M], \Sigma_{tu}^i = kde(x_{tu}, i)$$
  4. *Compose The Outgoing message*  

$$m_{tu}(x_u) = \{x_{tu}^i, \Sigma_{tu}^i, w_{tu}^i\}_{i=1}^M$$
- 

<sup>2</sup>see Appendix A for more details on kernel density estimations.

## 2.4 Discussion

In this chapter, different graphical model formalisms have been presented. We have also discussed efficient inference techniques that can be applied to these models, namely Belief Propagation and its nonparametric extensions (NBP and PAMPAS).

During the analysis of the available graphical model representations, Pairwise Markov Random Fields appeared in general to be the most convenient formalism for our purpose. Contrary to Bayesian Networks, they are able to cope with loopy graphs and offer better computational performance in comparison with Factor graphs by the use of only pairwise potentials.

However, the use of PMRFs, and of graphical models in general, presents some weaknesses that we should keep in mind. All the models assume the presence of well defined potential functions. In our case these potentials will have to be learned (Chapter 6).

Despite these drawbacks, using NBP in graphical models seems to be promising computationally and from the biological point of view [LM03]. Recently, several applications have successfully exploited Nonparametric versions of Belief Propagation on graphical models, these include; Hand Pose Estimation [SMFW04b], Stereo Vision, Self-Calibration in Sensor Networks [IFMW04], Articulated Body Tracking [SISB03], and tracking [SBR<sup>+</sup>04]. These are strong motivations to pursue the design of an object recognition system based on these concepts.



## State of the art

---

In this chapter, a discussion is presented on state-of-the-art research in connection with the present work. This is initiated by a brief insight into the use of local visual features to solve recognition problems. Then, in Section 3.1, we present techniques to find these local features in the image. A variety of these methods will be exploited along this thesis. In practice, these detection techniques are often coupled to local description methods to form powerful feature extractors. Most popular descriptors are presented in Section 3.2 where we show how a compact and semi-invariant description can be extracted from the neighborhood of these potentially interesting points. Finally, the subsequent parts (Section 3.3) of this chapter highlight representative research directions in object recognition developed during the last few years.

### **From Local to Global Visual Features**

It is often said that a central task for computer vision systems is to extract a computational description of the world based on images. For many vision applications, it is convenient to reduce the visual input space of the system to a set of visual features.

**Definition 3.1.** A *Visual Feature* is a generic term that is defined as a prominent or distinctive visual aspect, quality, or characteristic.

More specifically, a visual feature can be quantitative or qualitative, localizable or not, local or global. Global visual features are computed on the whole image. Among them we can find color histograms, texture values, and global shape parameters. These approaches have been widely used in industrial systems with a lot of success. However in some situations (*e.g.* in the presence of occlusion), they cannot always ensure satisfactory results. That is why most of the state-of-the-art methods have focused on developing systems that exploit local visual features. These are locals, in the sense that they are computed on a limited size neighborhood and are therefore robust to partial occlusion of the scene. Their usefulness is used in several domains of computer vision: starting from stereo matching, image retrieval, 3D reconstruction, and object tracking. Local features can also be very useful in object recognition. In this thesis, we will exploit local features as the visual input of our recognition system.

### 3.1 Where: Local Feature Detection

This section is dedicated to the description of state-of-the-art methods for detecting local visual features in images. The purpose is not to take the reader through an exhaustive list of methods but rather to present a general survey of the field. The questions addressed in this section are the following:

1. What do we expect from a local feature detector?
2. What are the available techniques and what are their properties?

Ideally, local features should be located in a potentially informative neighborhood and extracted reliably. This means that the feature should present a reasonable degree of robustness. The invariance should generally cover heterogeneous combinations of scale, intensity, and geometric transformations as well as blur and image compression artifacts. The key for a detector is to offer a good trade-off between robustness and precision.

If we take a bird's-eye view on the available detection techniques, three categories appear: signal-based methods (Section 3.1.1), geometric (Section 3.1.2), and some other methods that we call *fast alternative* approaches (Section 3.1.3). Some of these methods are detailed in the next subsections where the second question is considered.

### 3.1.1 Signal-based Methods

In these methods the detection is done directly on the image intensity. A class of signal-based methods has popularized the use of local features, they are called *points of interest*. In 1977, MORAVEC [Mor77] was among the first to develop this notion, which he defined as follows:

**Definition 3.2.** *Points of interest* are defined as occurring when large intensity variations are present in every direction.

MORAVEC obtained them by computing a local auto-correlation in four directions and taking the highest result as measure of interest. However, this operator has a critical weakness: the use of only four directions for finding the local auto-correlation lead to noisy responses. One year later, BEAUDET [Bea78] proposed an interest point detector which enhanced high curvature edges by calculating the image Gaussian curvature that is based on the second derivatives of the image.

A few years later, FÖRSTNER [För86] described a detector based on a similar idea that Moravec's interest operator, but this time the measure of auto-correlation is estimated from first order image derivatives. The use of first order derivatives leads to a better robustness to noise. HARRIS AND STEPHENS [HS88] modified FÖRSTNER's detector by using another operator to select interest points which became very popular.

In the next paragraphs, we review current signal-based methods to detect local features. We present successively Harris detector and its various extensions, Hessian, difference of Gaussian and MSER.

#### Harris Detector

The Harris interest point detector [HS88], also called Plessey detector, has been popular in the 90's. Its robust and reliable detection combined with its generality are the main reasons of its success. Originally based on the work of MORAVEC [Mor77] and FÖRSTNER [För86], the idea is, as previously said, to use the variation of the auto-correlation over different orientations.

HARRIS AND STEPHENS [HS88] have shown the possibility to use directly the auto-correlation matrix (*i.e.* second moment matrix) for detecting interest points. This matrix <sup>1</sup> (Equation 3.1) describes the gradient profile in the local neighborhood

---

<sup>1</sup> This formulation [Sch96] uses a Gaussian function to compute derivatives. Note that the original Harris detector uses the mask  $[-2 \ -1 \ 0 \ 1 \ 2]$  to compute derivatives.

of a point. It is sensitive to discontinuities of the image signal.

$$M(\sigma_I, \sigma_D) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} = \sigma_D^2 g_{\sigma_I} * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \quad (3.1)$$

where  $I$  is the image convolved by a Gaussian derivative kernel,  $g_\sigma$  is a Gaussian kernel (Equation 3.2) of standard deviation  $\sigma$ ,  $\sigma_I$  is the integration scale and  $\sigma_D$  the derivation scale.

$$g_\sigma = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3.2)$$

Given any image  $I$ , its Gaussian derivatives  $I_x, I_y$  in both directions are defined as follows:

$$I_x = \frac{\partial}{\partial x} g_\sigma * I \quad (3.3)$$

$$I_y = \frac{\partial}{\partial y} g_\sigma * I \quad (3.4)$$

Two parameters ( $\sigma_I, \sigma_D$ ) are used to define the scale at which the detector operates (the terminology is inspired by [GL96]):

**Integration scale ( $\sigma_I$ ):** It defines the standard deviation of the Gaussian kernel that is used to smooth the image derivatives in the neighborhood of each point.

**Derivation scale ( $\sigma_D$ ):** It determines the standard deviation of Gaussian windows that are used to compute the local derivatives. The image derivatives are then averaged in the neighborhood of the point by smoothing with a Gaussian window of size  $\sigma_I$ . By convention, the derivation scale  $\sigma_D$  can be set proportional to the integration scale  $\sigma_D = s \sigma_I$ . However, if the value of  $s$  is too small,  $\sigma_D$  will also be small and the smoothing will be too large, therefore information will be lost. Experiments [MS02] have shown that the best repeatable results are obtained with a value of  $s \in [0.5, \dots, 0.75]$ . In our work, we set  $s$  to 0.6.

Generally, the integration scale parameter  $\sigma_I$  has to be fixed manually, thus it requires the user to have an approximate idea of the feature size. To avoid this manual setting, automatic scale selection methods have been proposed and will be discussed in a subsequent paragraph of this section.

The main contribution of Harris and Stephens is to provide an operator  $R$  on the second moment matrix  $M$  that is proportional to bidimensional variations of the intensity. This operator decides when the eigenvalues are sufficiently large to



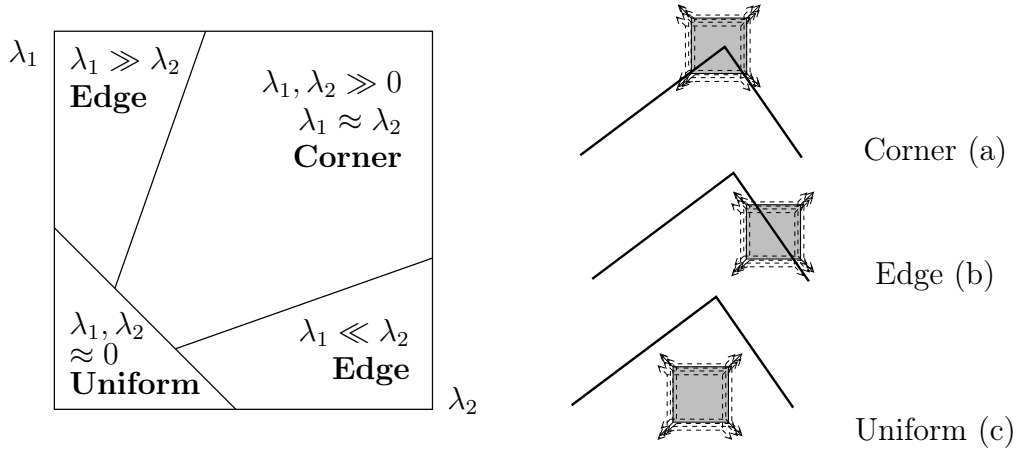


Figure 3.1: Analysis of cornerness in the Harris interest point detector [HS88]. Classification of an image point using the eigenvalues of  $M$ . (a) A corner is observed through a small window that is moved around. Shifting this window in *any direction* should give a significant change in the filter response. (b) When the window covers an edge, there is no change along the edge direction. (c) Over an uniform region, there is no change in any direction. Reproduced with permission from [Jod06].

establish the presence of an interest point. Intuitively, we can see that eigenvalues  $\lambda_1$  and  $\lambda_2$  represent the two principal curvatures of the gradient for each point. This property enables the extraction of points for which both curvatures are high, which implies with good confidence that the window is over a corner. This notion is measured through a difference between the trace and the determinant of the second moment matrix.

Interest points are selected where the cornerness measure  $R$  is above a given threshold  $t$ :

$$R = \det(M) - \alpha \text{trace}(M)^2 > t \quad (3.5)$$

$$R = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2 > t, \quad \alpha = 0.04 \quad (3.6)$$

For any symmetric matrix the trace is the sum of the eigenvalues:

$$\text{trace}(M) = \sum_i \lambda_i(M) \quad (3.7)$$

The determinant is the product of the eigenvalues:

$$\det(M) = \prod_i \lambda_i(M) \quad (3.8)$$

As it is illustrated in Figure 3.1, if one eigenvalue dominates, the area is considered as an edge; and if both eigenvalues are low, the area can be considered as uniform. In the classic Harris detector [HS88], the threshold parameter  $t$  has to be fixed by the user. To avoid this, I implemented a simple procedure for an automatic *threshold* selection. This can adjust the right *threshold* value to obtain a given number of interest points.

### Harris Color Points of Interest

The Harris interest point detector [HS88] originally operates on gray level image intensities. A straightforward generalization of the Harris detector to color images was introduced by DERICHE *et al.* [DGM98, Gou00, GMDP00]. The second moment matrix  $M$  was adapted by replacing the original grey-level derivatives by the combination of derivatives now applied on each color channel separately:

$$M(\sigma_I, \sigma_D) = \sigma_D^2 g_{\sigma_I} * \begin{bmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & R_y^2 + G_y^2 + B_y^2 \end{bmatrix} \quad (3.9)$$

Here  $R, G, B$  are responses to the first Gaussian derivative for each channel of image intensity. Following their experiments [GMDP00], it appears to be one of the most stable color interest point detectors with regard to image rotation, noise, illumination, and viewpoint changes.

### Scale Invariant Interest Points

The Harris detector considers a single fixed integration scale, and therefore fails in the presence of scale changes. This may become problematic since a real-world object is typically composed of structures with various sizes. This motivates the introduction of detectors that can automatically select, for each location in the image, the scale that best suits the neighborhood.

Several strategies are possible to determine the scale of each interest point automatically. The “naive” multi-scale Harris detector [Mik02] works as follows: it detects interest points at several scales and looks for points which are maxima over scales. A feature point is detected if a local maximum is present in a surrounding 3D cube and if its value is above the threshold. Unfortunately, experiments [Mik02] have shown that this scale adapted interest function rarely reaches maxima over scales. Many points are consequently lost and so is the repeatability of the detector.

To overcome this challenging task, researchers turned to scale space theory. LINDEBERG [Lin98] extensively studied the use of dedicated scale selection operators.

In order to select the scale of a point, the idea is to apply a given operator at several scales, and then to select scales where measures reach local maxima. Below, we describe two operators [Lin98] which are designed to recover the scale of different image structures.

**Blob Operator:** The maximum response of a Laplacian filter (*i.e.* Laplacian-of-Gaussians, LoG) (Equation 3.10) computed across several scales, can be used to recover the characteristic scale of circular regions. This kernel is well adapted to blob detection due to its circular symmetry (see Figure 3.2). It also demonstrates empirically fair results for the characteristic scale selection of other local structures such as corners, edges, and multi-junctions.

$$Laplacian(\sigma) = \sigma^2 |I_{xx}(\sigma) + I_{yy}(\sigma)| \quad (3.10)$$

**Corner Operator:** This operator is a specific junction operator. The curvature is multiplied by the gradient magnitude. This leads to the following operator:

$$\tilde{K}_{norm}(\sigma) = \sigma^2 (I_x^2(\sigma)I_{yy}(\sigma) - 2I_x(\sigma)I_y(\sigma)I_{xy}(\sigma) + I_y^2(\sigma)I_{xx}(\sigma)) \quad (3.11)$$

The weakness of this kind of technique concerns the discretization of the scale space. A common heuristic is to consider scales spaced out by a factor of 1.2 such as:  $\sigma = 1.2^n$ , where  $n = [0 \dots 17]$ .

### Harris-Laplace Detector

MIKOLAJCZYK *et al.* [MS01] unified the technique of LINDBERG scale selection [Lin98] and Harris detector [HS88] in a new scale-invariant interest point detector, named the *Harris-Laplace* detector.

The detector (Algorithm 2) first constructs a scale-space representation of the Harris measure. This is done by computing the second moment matrix  $M(\sigma_I)$  (Equation 3.1) for the entire image at each considered scale  $\sigma_I$ . Local maxima and minima are searched *separately* at each level of the scale space representation.

The second step is to consider spatial local extrema. If they are also a local maximum in the scale dimension, by evaluating the normalized Laplacian, then the point is selected as *Harris-Laplace* interest point. In the scale space, a point at a scale  $s$  is a local maximum if its Laplacian response is greater than its two nearest scales  $s - 1$  and  $s + 1$ . This method allows to extract a set of characteristic locations

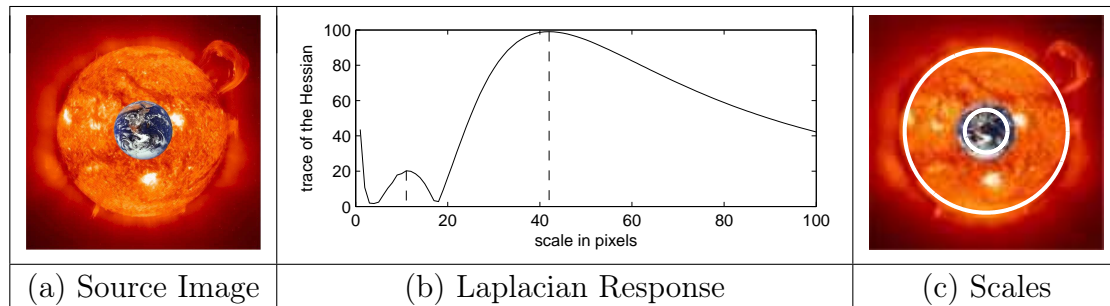


Figure 3.2: Illustration of the computation of the intrinsic scale for blob features. (a) The source image the size of which is  $121 \times 121$  pixels. (b) The scale-space signature  $r(60, 60; \sigma)$  for the location at the center of the image. Each local maximum in this signature defines an intrinsic scale for this location: In this example, two intrinsic scales are detected at  $\sigma = 11$  pixels and  $\sigma = 42$  pixels. Thus, there are typically multiple intrinsic scales associated with each image location. (c) Two circles whose radii correspond to one computed intrinsic scale are overlaid on the source image. The smaller intrinsic scale corresponds to the Earth, whereas the larger is induced by the sun (reproduced with permission from [Jod06]).

with associated scales. In experiments [MS01], it has been shown that this detector performs reliable detection in presence of large scale changes.

However, displacements may occur between local maxima at different scales. Therefore, comparing the same spatial point in another scale is biased. To solve this problem, an extension was introduced [Mik02, MS04] to perform an iterative adaptation of the point location in order to select the scale invariant interest points. This procedure is given in Algorithm 3.

### Affine Invariant Detector

The principal weakness of the Harris-Laplace detector [MS01] is its imprecise detection in the presence of viewpoint changes. A small displacement of the camera leads to an object surface deformation. Such a transformation introduces significant changes in the point location as well as in the scale and the shape of its neighborhood. If one tries to match points between affine transformed related images, one will probably see a displacement of the local maximum. This can be an issue in some applications.

MIKOLAJCZYK *et al.* [MS02] improved the Harris-Laplace detector [MS01] and presented a new approach for detecting affine invariant interest points. Their iter-

---

**Algorithm 2** Harris-Laplace Detector

---

```

1:  $\Sigma \leftarrow$  select  $n$  scales  $\{\sigma_{i=1..n}\}$ 
2:  $\mathcal{T} \leftarrow$  choose a threshold
3: for each scale  $\sigma_i \in \Sigma$  do
4:   Compute the matrix  $M(\sigma_I, \sigma_D)$  (Equation 3.1) where  $\sigma_I = \sigma_i$  and  $\sigma_D = 0.6 \times \sigma_i$ 
5:   Compute  $R$  (Equation 3.5) for each point  $x$  in the image
6:    $\mathcal{P}_{\sigma_i}^x \leftarrow$  Extract local maxima in  $R$  above threshold  $\mathcal{T}$ 
7:    $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}_{\sigma_i}^x$  // where  $\mathcal{P}_{\sigma_i}^x$  is the response  $R$  computed at point  $x$  with scale
       $\sigma_i$ 
8: end for
9: for each point  $\mathcal{P}_{\sigma_i}^x \in \mathcal{P}$  do
10:  if ITERATIVE VERSION then
11:     $\mathcal{S} \leftarrow \mathcal{S} \cup$  Iterative Scale Selection( $\mathcal{P}_{\sigma_i}^x$ )
12:  else
13:    if  $\mathcal{L}_{\sigma_i}^x > \mathcal{L}_{\sigma_{i+1}}^x \wedge \mathcal{L}_{\sigma_i}^x > \mathcal{L}_{\sigma_{i-1}}^x$  // evaluate Laplacian response then
14:       $\mathcal{S} \leftarrow \mathcal{S} \cup \{x, \sigma_i\}$ 
15:    end if
16:  end if
17: end for

```

---



---

**Algorithm 3** Iterative Scale Selection( $\mathcal{P}_{\sigma}^x$ )

---

```

1: repeat
2:    $\sigma' = \sigma$ 
3:    $x' = x$ 
4:    $\sigma' \leftarrow$  Find the local extremum over scale for the point  $x$  (by evaluating the
      Laplacian), otherwise reject the point. The investigated range of scales is
      limited to  $t\sigma$  with  $t \in [0.7, \dots, 1.4]$ 
5:    $x' \leftarrow$  Detect the spatial location of a maximum of the Harris measure  $R$ 
      nearest to  $x$  for the selected scale  $\sigma'$ 
6: until  $\sigma' = \sigma \wedge x' = x$ 
7: Return  $\{x', \sigma'\}$ 

```

---

ative method can deal with significant affine transformations including large scale changes. The approach is based on two key ideas:

- the scale of a local structure can be computed by considering the local maximum of Laplacian responses over scales, and the location of the points can be iteratively adjusted (similarly to the Harris-Laplace detector).
- the second moment matrix (Equation 3.1) computed on a point can be used to normalize a region in an affine invariant way [LG97, Bau00].

The main contribution of the Harris-Affine detector is to provide an iterative procedure (Algorithm 4) that modifies location, scale, and neighborhood of each initial interest point and converges to affine invariant points by repeatedly normalizing the image neighborhood.

A quantitative comparison [Mik02] with existing detectors shows a significant improvement in the presence of large affine deformations. However, the gain on view-point invariance is lost on repeatability. In our implementation, we observed that many points do not converge to stable normalized region.

---

**Algorithm 4** Affine Invariant Interest Point Detector (Simplified)

---

```

1:  $\{X, \Sigma\} \leftarrow$  Apply Harris detector at several scales
2: for each point  $x \in X$  of scale  $\sigma \in \Sigma$  do
3:    $x^k \leftarrow x$  // spatial location
4:    $\sigma^k \leftarrow \sigma$  // integration scale
5:   repeat
6:     // Find the local extremum over scale around  $\sigma^k$  by evaluating the Laplacian
7:      $\{\sigma_I, \sigma_D\} \leftarrow$  select integration and derivation scale at point  $x^k$ 
8:     compute transformation  $\mathcal{T} \leftarrow \sqrt{M(x^{(k)}, \sigma_I, \sigma_D)}$ 
9:     normalize  $\mathcal{T}$  to  $\lambda_{max}(\mathcal{T}) = 1$ 
10:    normalize window centered on  $x^{(k)}$  using  $\mathcal{T}$ 
11:     $x^{(k+1)} \leftarrow$  compute the nearest interest point to  $x^{(k)}$  for  $M(\sigma_I, \sigma_D)$ 
12:     $\sigma^{(k+1)} \leftarrow \sigma_I$ 
13:     $k \leftarrow k + 1$ 
14:  until  $\frac{\lambda_{min} M(x^k, \sigma_I, \sigma_D)}{\lambda_{max} M(x^k, \sigma_I, \sigma_D)} < \tau$ 
15:  // if no convergence reject  $x$ 
16: end for

```

---

### Difference of Gaussians

Inspired by LINDBERG’s scale selection theories [Lin98], LOWE *et al.* [Low99, BL02] have developed an efficient scale-invariant feature detector. This has some similarities to the non-iterative Harris-Laplace but it approximates the Laplacian by a Difference-of-Gaussians (DoG) that can be computed more efficiently:

$$DoG = (G_{\sigma_1} - G_{\sigma_2}) * I = (G_{\sigma_1} * I) - (G_{\sigma_2} * I) \quad (3.12)$$

The gain in computational cost is possible thanks to a *Gaussian pyramid* representation of the image. Each level in this pyramid is the smoothed version of the lower level, starting with the bottom level containing the source image. The top level corresponds to the coarsest scale, and is limited by the size of the image.

The detector uses the *Gaussian pyramid* to build a *DoG pyramid*. Each level of the DoG pyramid is obtained by subtracting two successive levels of the Gaussian pyramid. LOWE *et al.* proposed to identify local 3D extrema  $(x, y, \sigma)$  in the DoG pyramid. Then, they locate the extrema to sub-pixel / sub-scale accuracy by fitting a second degree function to the scale-space Laplacian.

The main drawback of this technique is the need of a “cleaning step” where edge responses are eliminated.

### Hessian Interest Points

Hessian interest point detector was proposed by BEAUDET [Bea78]. It enhances high curvature points by calculating image Gaussian curvature. Similarly to the Harris detector [HS88], it was extended more recently by using Gaussian kernels to smooth the derivative responses. The Hessian is the matrix of second partial derivatives expressed as

$$\mathcal{H}(\sigma_I, \sigma_D) = \begin{bmatrix} h_{xx} & h_{xy} \\ h_{yx} & h_{yy} \end{bmatrix} = \sigma_D^2 g_{\sigma_I} * \begin{bmatrix} I_{xx}(\sigma_D) & I_{xy}(\sigma_D) \\ I_{xy}(\sigma_D) & I_{yy}(\sigma_D) \end{bmatrix} \quad (3.13)$$

$$h_{xx}h_{yy} - h_{xy}^2 > threshold \quad (3.14)$$

where  $\sigma_I$  is the integration scale,  $\sigma_D$  the derivation scale,  $g$  the Gaussian, and  $I$  the image smoothed by Gaussian derivative.

Multi-scale and affine extensions are straightforward and similar to those of the Harris interest point detector.

### Maximally Stable Extremal Regions

The concept of Maximally Stable Extremal Region (MSER) was proposed by MATAS *et al.* [MCUP02]. An Extremal region consists of a subset of connected pixels which are all brighter (MSER+) or darker (MSER-) than all the pixels on the region's boundary. These regions are obtained through a watershed flooding segmentation algorithm [VS91] that is applied to the image intensities. To define the regions, the algorithm selects segment boundaries that are stable when considering successive thresholded transformations of the original image.

By construction, the selected regions are quite often uniform regions surrounded by a highly contrasted boundary. Therefore, the reliable matching of such regions is difficult. In order to more easily describe distinctively each region, an ellipse is fit to it. The position of the regions are computed from the average pixels locations. The size is given by a geometric mean of the eigenvalues of the second order moments matrix, computed from the pixel locations. The need for this operation is closely linked to the following section, where we consider feature extraction for describing regions or points with their neighborhood.

Interestingly, this type of feature is invariant to affine and photometric transformations. It is also really fast; the computational complexity is linear in the number of pixels. It has been successfully used in Image Retrieval [OM02a] and Stereo Matching [MOC02, CMO03]. MSER often demonstrate high precision and repeatability in object recognition tasks. However, the main weakness resides in the number of regions detected which is often too small.

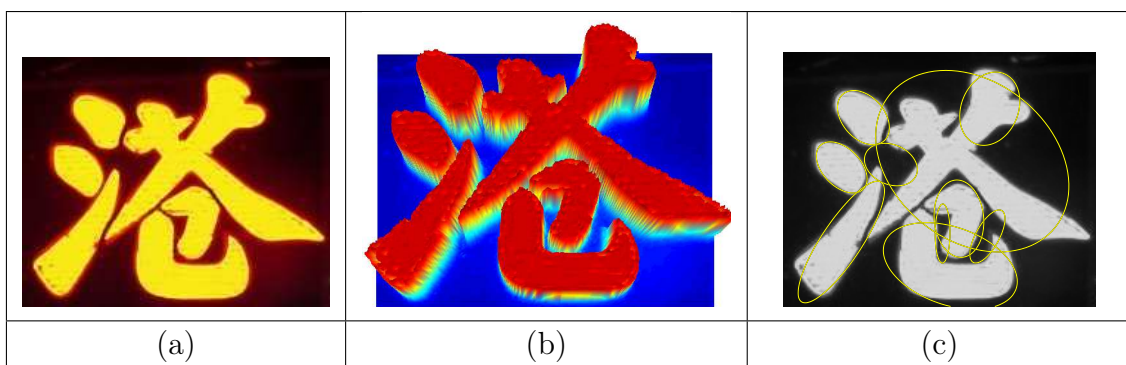


Figure 3.3: An image (a) and its topographical intensity surface (b). After detection (c), ellipses are used to fit MSER regions.



**Intensity Extrema-Based Region Detector**

Recently, TUYTELAARS AND VAN GOOL [Tuy00, TG00, TVG04] proposed a method to detect and extract affine invariant regions. The method, of which an outline is presented in Algorithm 5, starts from intensity extrema detected at several scales, and explores the image around them in a radial way. A function applied on each ray is used to determine regions of arbitrary shape, which are then replaced by ellipses.

Given a local extremum, the intensity profile along each ray originating from the extremum is studied by evaluating the function  $f_I(t)$ :

$$f_I(t) = \frac{\text{abs}(I(t) - I_0)}{\max\left(\frac{\int_0^t \text{abs}(I(t) - I_0) dt}{t}, d\right)} \quad (3.15)$$

where  $t$  is the Euclidean length along the ray,  $I(t)$  the intensity at position  $t$ ,  $I_0$  the intensity value at the extremum, and  $d$  a small number which has been added to prevent a division by zero.

The authors showed that the point for which function  $f_I(t)$  reached an extremum generally offers affine invariant properties. Typically, a maximum is reached at positions where the intensity suddenly increases or decreases. Once the maximum is detected, all corresponding points of  $f_I(t)$  along rays originating from the same local extremum are linked to define an affine covariant region. This irregular region is replaced by an ellipse having the same shape moments up to the second order (Figure 3.4).

---

**Algorithm 5** Intensity-based Regions

---

- 1:  $X \leftarrow$  Detect intensity extrema in image
  - 2: **for each** point extrema  $x \in X$  **do**
  - 3:   Consider intensity profile along rays
  - 4:    $l_{max} \leftarrow$  Select maximum of invariant function  $f(t)$  along each ray
  - 5:    $r_x \leftarrow$  Connect all local maxima  $l_{max}$  to create a region
  - 6:    $e_x \leftarrow$  Fit an ellipse for the region  $r_x$
  - 7:    $S \leftarrow S \cup \{x, e_x\}$
  - 8: **end for**
- 

**Salient regions**

The detector of KADIR AND BRADY [KB01] is another way to detect features in the scale space of the image. In contrast with *Harris-Laplace* detector [MS01], it does

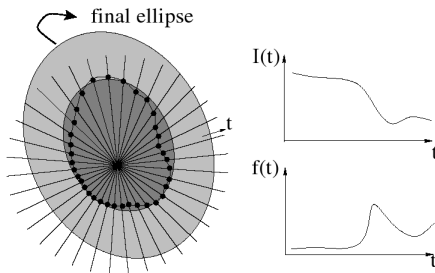


Figure 3.4: The intensity along “rays” emanating from a local extremum are studied. The point on each ray for which a function  $f(t)$  reaches an extremum is selected. Linking these points together yields to an affine invariant region, to which an ellipse is fitted using moments. Reproduced with permission from Tuytelaars [Tuy00].

not start with interest points but rather tries to find regions that are salient directly over both location and scale. To do so, it first constructs an intensity histogram for each point in the image using a circular region. Then the entropy of each histogram is calculated. For a given point  $x$ , local entropy can be defined as:

$$\mathcal{S}_{D,W_x^\sigma} = - \sum_i P_{D,W_x^\sigma}(d_i) \log_2 P_{D,W_x^\sigma}(d_i) \quad (3.16)$$

where  $W_x^\sigma$  is the local neighborhood extracted at scale  $\sigma$ ,  $D$  is a descriptor that takes on values  $d_{i=1,\dots,r}$ ,  $P_{D,W_x^\sigma}(d_i)$  is the probability of descriptor  $D$  taking the value  $d_i$  in the local window  $W_x^\sigma$ .

The saliency of each location and scale is measured. This leads to a 3-D saliency map. Regions of high saliency are clustered over both location and scale. The  $N$  centroids of the clusters are selected as output features.

This framework was recently extended by including invariance to affine transformations [KZB04]. Despite its known stability when only a small number of regions per image are required, this detector suffers from a prohibitive computational cost that reduces its popularity.

### Wavelet-based Interest Points

Harris-like interest point detectors may present some weakness for the detection of features in natural images. Specifically, interesting features are not necessarily located at corner locations. From these observations, LOUPIAS *et al.* [LS99, SL03] presented another class of interest points based on a wavelet image representation.

The wavelet image is obtained as the convolution of image with a wavelet function computed at different scales. A recursive process tracks interesting locations in different resolutions to keep the points presenting the highest response to a saliency measure.

Precision, repeatability and information content experiments performed on a small number of images showed that the wavelet-based interest point detectors provide better results compared to the classical Harris detector in some conditions [STL<sup>+</sup>02]. Also, this detector is not restricted to image regions where the signal vary two-dimensionally.

### 3.1.2 Geometry-based Methods

In geometry-based methods, edges are often used as a starting point. A preliminary step using a standard edge detector (*e.g.* Sobel [Pin69], Canny [Can86]) is often performed. Intuitively, an edge is detected if there is a strong contrast transition in one direction at a given point.

Since the first attempts to perform object matching in the 70's, edges have generally been popular and considered as strong cues in many other applications. More generally, it is common to use edge points as input to find more robust and informative locations. Among all the possible geometric-based approaches, we mention here two different directions that are pursued by the computer vision community.

- One possibility is to construct a polygonal approximation of edges. The idea is to link edge points to produce segments. Local features can either be considered at each segment intersections [HSV90], or defined by the region constructed from segments [TVG04].
- The Shape Skeleton [Blu67, GS99, Xia89] is obtained by repeatedly thinning until it becomes a one pixel width network. Thinning strategies generally work on the principle of stripping away successive layers of shape boundary points on the condition that the removal of a point does not change the connectedness of the shape. When all allowable points have been removed the shape skeleton is left. By their nature, thinning algorithms are sensitive to occlusion (to a thinning algorithm an occluded shape looks like a different shape with different topology) and in general recognition schemes based on skeletons cannot cope with occlusions.

The known problem of this kind of methods is that edge detection is sensitive to noise and has some difficulties in the presence of complex images or structures. Sometimes the class of object or the texture does not allow to detect lines. These arguments reduce the generality of this approach and lead us to consider other techniques to detect features more robustly.

### 3.1.3 Fast Alternative Methods

The role of signal and geometry-based methods is to reduce the visual input space of the system in focusing on some image area that are believed to be pertinent. However, in the case of object recognition, the *computational cost* to perform the detection and the fact that a detector may *miss crucial locations* has motivated some researchers to use different approaches. These alternative approaches do not rely on image intensity but rather use a stochastic process to generate image locations. Among these methods, we review fixed grid, fully random, and randomized grid detection.

#### Fixed Grid

The fixed grid detector is the most basic strategy to reduce the input space to a set of feature locations. It simply returns a set of image locations  $\{x_{ij}, y_{ij}\}$ , each of them being sampled along a uniform, regularly spaced grid.

Surprisingly, this simple detector gave good results in some applications such as scene classification [VS04, FP05]. In contrast with interest point detectors that are often distributed along highly contrasted image structure, it has the inherent advantage that the images are densely covered. However, we observed that the use of regularly spaced grid may suffer from singularities in some situations. For instance, if the grid is placed on an image representing a horizontal line, the appearance of *all* the features extracted along the line will be influenced by vertical displacement of the image. Also they will all have the same appearance. As we observed in our experiments, this may induce some generalization problems during learning, particularly for the recognition of object classes.

#### Full Randomized

Another random strategy has been proposed by MARÉE *et al.* [MGPW05c]. This stochastic detector is said fully randomized. In its basic version, the randomized

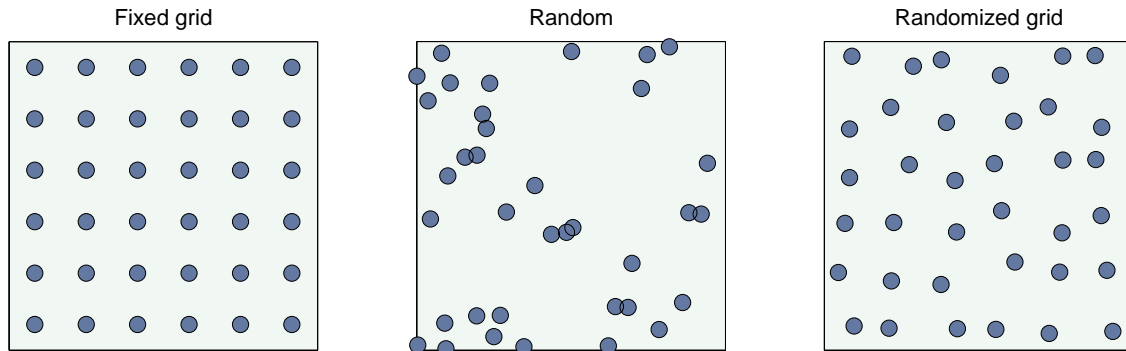


Figure 3.5: Illustration of three strategies for spatial sampling. (left) The fixed grid is the most uniform sampling strategy but may suffer from singularities. (center) A poor choice of a Random selection method of points may lead to a bad covering of the image. Randomized grid method (right) presents the advantages of using a grid sampling but avoid singularities by introducing a small random displacement in the point location.

detector selects  $k$  interest points in the image at random. MARÉE *et al.* have shown remarkable performance of this approach for image classification [MGPW05c]. In our experiments we observed that such a detector may not well cover the image when the number of generated points becomes low. Our observations were confirmed in the context of image interpolation [Kir03] where a randomized grid sampling gave more accurate results than a random strategy.

### Randomized Grid

The weaknesses of the fixed grid method can be remedied by randomizing the position of points. This stochastic process corresponds intuitively to “shaking the grid”.

The method we propose first uses a fixed grid to generate a basic set of points  $\{x_{ij}, y_{ij}\}$ . Then a random shift is introduced to these coordinates of points, which has the effect of shaking the grid. Thus, the points move away from the initial positions into random directions. The shift, which is different for each point, is assigned in the form of a percentage  $\delta$  of the interval  $\tau$  between the rows and the columns. An outline of the procedure is given in Algorithm 6.

This method offers the same advantages of a fixed grid detector; very low computational cost and dense covering of the image. In addition, it also prevents the system from singularities and demonstrated more reliable results for different scenarios in our recognition framework.

---

**Algorithm 6** Randomized Grid Generation

---

```

1:  $X' \leftarrow \{\}$ 
2:  $X \leftarrow$  generate a fixed grid
3:  $\tau \leftarrow$  distance between two points of the grid
4:  $\delta = .5$  // percentage of the interval  $\tau$ 
5: for each point  $\{x_{ij}, y_{ij}\} \in X$  do
6:   // Compute random shifts within the interval
7:    $s_x = \text{rand}(-\tau\delta, \tau\delta)$ 
8:    $s_y = \text{rand}(-\tau\delta, \tau\delta)$ 
9:    $\{x'_{ij}, y'_{ij}\} = \{x_{ij} + s_x, y_{ij} + s_y\}$ 
10:   $X' \leftarrow X' \cup \{x'_{ij}, y'_{ij}\}$ 
11: end for

```

---

### 3.1.4 Discussion

In this section, we have focused on providing a summary of the most popular approaches to detect local features. These techniques are extremely diversified and rely on different properties of the image values. Therefore they may have different behavior depending on the type of image. In the following chapter, we will evaluate their performances on a set of images.

The common point between these techniques is to provide a set of local, potentially interesting regions in the image. The description of these regions is the other task involved in the feature extraction process. It is discussed in the following section.

## 3.2 What: Local Description

In the previous section, we presented state-of-the-art approaches to the detection of local features. These relatively robust image locations were selected either from a stochastic process or on the basis of the intensity structure of their neighborhood. We will see in this section how it is possible to extract a both compact and robust representation of these local image regions.

Many different local descriptors have been proposed in the literature. A direct approach is to sample the local image intensities around the interest point at the appropriate scale. However, for some applications this sampling is too sensitive to changes such as affine, photometric, 3D viewpoint change, or non-rigid deformations. Moreover, the high-dimensionality can also be prohibitive. In order to be able to deal with complex transformations of the signal and to obtain a compact representation, the computer vision community has focused on the development of local image descriptors that present interesting properties: invariance to image rotation, illumination variation, and view-point change.

The main difficulty for these methods is to find the best compromise between degree of invariance and selectivity (*i.e.* discriminative power). For instance, let us consider a descriptor that consists of pixel values. This descriptor will be very distinctive, but will also present a weak invariance degree and will not generalize well. On the other hand, descriptors with a high degree of invariance may have a reduced discriminative power.

Various high performance techniques to describe local features have been developed in the last few years and will be reviewed in this section. The available methods are classified in three main classes. After reviewing Convolution Coded (Section 3.2.1) and Histogram based Descriptors (Section 3.2.2), we explore alternative approaches (Section 3.2.3). Finally, we terminate this overview of local description techniques by a discussion in Section 3.2.4.

### 3.2.1 Convolution Coded Descriptors

A commonly used approach to describe an image region is to extract the responses of a Gaussian filter (or its derivatives). These techniques, called convolution coded, are very compact and present fair invariance properties to simple image transformations.

## Differential Invariants

Intuitively, *Differential Invariants* are constituted of responses of several combinations of Gaussian derivative convolutions applied on a given local region.

This is motivated by the Taylor expansion that is a classical technique to describe a function locally in terms of its derivatives. KOENDERINK *et al.* [KvD87] exploited this property to characterize the intensity function of an image region. The so-called *local jet* descriptor uses the Gaussian filter to compute local image derivatives in a more robust way. The *local jet*  $J^N[I]$  computed up to the order  $N$  for the image  $I$  is written as:

$$J^N[I](x, \sigma) = \{L_{i_1 \dots i_n}(x, \sigma) | (x, \sigma) \in I \times \mathbb{R}^+; n = 0, \dots, N\} \quad (3.17)$$

Here the term  $L_{i_1 \dots i_n}(x, \sigma)$  represents the convolution of the image with the  $n$ -th Gaussian derivatives such that each  $i_k \in \{x, y\}$ .

In practice, the set of invariants is usually limited to third order. Equation 3.18 shows the elements of the differential invariant descriptor  $\vec{D}$  up to second order. The second term is given in tensorial notation [SM97, WCT98] (*i.e.* Einstein summation convention):

$$\vec{D}[0..4] = \begin{bmatrix} L \\ L_i L_i \\ L_i L_{ij} L_j \\ L_{ii} \\ L_{ij} + L_{ji} \end{bmatrix} = \begin{bmatrix} L & \leftrightarrow \text{Intensity} \\ L_x L_x + L_y L_y & \leftrightarrow \text{Gradient Magnitude} \\ L_{xx} L_x L_x + 2L_{xy} L_x L_y + L_{yy} L_y L_y \\ L_{xx} + L_{yy} & \leftrightarrow \text{Laplacian} \\ L_{xx} L_{xx} + 2L_{xy} L_{xy} + L_{yy} L_{yy} \end{bmatrix} \quad (3.18)$$

Likewise Differential Invariants for color images [MGD98, GMP98, GSvdB01, Hal01] combine Gaussian kernel responses of RGB color channels. Here is the composition of such a descriptor using only first order invariants:

$$\vec{D}_{col}[0..7] = \begin{bmatrix} R & \leftrightarrow \text{Red Channel Intensity} \\ R_x R_x + R_y R_y & \leftrightarrow \text{Red Channel Magnitude} \\ G & \leftrightarrow \text{Green Channel Intensity} \\ G_x G_x + G_y G_y & \leftrightarrow \text{Green Channel Magnitude} \\ B & \leftrightarrow \text{Blue Channel Intensity} \\ B_x B_x + B_y B_y & \leftrightarrow \text{Blue Channel Magnitude} \\ (R_x^2 + R_y^2)(G_x^2 + G_y^2) & \leftrightarrow \text{Red and Green Magnitude} \\ (R_x^2 + R_y^2)(B_x^2 + B_y^2) & \leftrightarrow \text{Red and Blue Magnitude} \end{bmatrix} \quad (3.19)$$



### Steerable Filters

Pursuing the direction suggested by the Differential Invariants [KvD87], where descriptors are constructed from Gaussian derivative kernel convolutions, FREEMAN AND ADELSON [FA91] introduced a new set of filters, called steerable filters, and defined it as:

**Definition 3.3.** A filter set forms a *steerable filter* class if a copy of the filter at any orientation can be computed as a linear combination of the basis filters.

Interestingly, Gaussian derivative kernels are steerable. Therefore, it is possible to use this property to describe a local neighborhood. The oriented derivative of a Gaussian of order  $d$  at orientation  $\theta$  is written as  $L_d^\theta(x, \sigma)$ . Here is a general formulation for the first three derivatives (following the notation in [Pia01]):

$$L_d^\theta = \sum_{k=0}^d L_d^{\theta_{k,d}} c_{k,d}^\theta \quad (3.20)$$

where

$$\begin{aligned} c_{k,1}^\theta &= \cos(\theta - k\pi/2) & k &= 0, 1 \\ c_{k,2}^\theta &= \frac{1}{3}(1 + 2\cos(2(\theta - k\pi/3))) & k &= 0, 1, 2 \\ c_{k,3}^\theta &= \frac{1}{2}(\cos(\theta - k\pi/4) + \cos(3(\theta - k\pi/4))) & k &= 0, 1, 2, 3. \end{aligned} \quad (3.21)$$

The steerable descriptor will have the following composition:

$$\vec{D} = \left( L_1^{\theta_{0,1}}(x, \sigma), L_2^{\theta_{0,2}}(x, \sigma), L_2^{\theta_{1,2}}(x, \sigma), L_2^{\theta_{2,2}}(x, \sigma), \right. \\ \left. L_3^{\theta_{0,3}}(x, \sigma), L_3^{\theta_{1,3}}(x, \sigma), L_3^{\theta_{2,3}}(x, \sigma), L_3^{\theta_{3,3}}(x, \sigma) \right)^T \quad (3.22)$$

where the  $\theta_{k,d}$  orientations are defined by

$$\theta_{k,d} = \frac{k\pi}{d+1}, k = 0, \dots, d \quad (3.23)$$

and the particular reference orientation can be computed according to the dominant gradient orientation:

$$\tan \theta_g = \frac{L_y}{L_x} \quad (3.24)$$

If the gradient is well defined in the local region, this technique is particularly efficient. However, the computation of  $\theta_g$  orientation is a weakness in presence of circularly uniform regions.

### Complex filters

Instead of using Gaussian filters directly to compute invariants, BAUMBERG [Bau00], SCHAFFALITZKY, AND ZISSERMAN [SZ02] proposed to use complex filters that are derived from the following equation:

$$K(x, y, \theta) = f(x, y) e^{i\theta} \quad (3.25)$$

where  $f(x, y)$  is a function applied on the image and  $\theta$  the orientation.

BAUMBERG uses Gaussian derivative convolution for  $f(x, y)$  whereas SCHAFFALITZKY AND ZISSERMAN apply the polynomial

$$K_{m,n}(x, y) = (x + iy)^m (x - iy)^n g(x, y) \quad (3.26)$$

where  $g(x, y)$  is a Gaussian. Different filters are computed by varying  $m$  and  $n$  such that  $m + n \leq 6$ . The final descriptor consists of 15 absolute values; one for each filter response.

Complex filters differs from Gaussian derivatives by a linear coordinates change in filter response space [MS03]. Therefore they tend to exhibit different properties and the choice depends on the type of image.

### Phase-based Descriptors

Another approach to local description is based on the phase and amplitude responses of complex-valued steerable filters (CARNEIRO *et al.* [CJ02, CJ03]). It exploits the fact that phase data is often locally stable with respect to scale changes, noise and common brightness changes.

The so-called *phase-based* local feature is a complex representation [Bau00] of local image region that is obtained through the use of quadrature pair filters, tuned to a specific orientation and scale. More specifically, they used the steerable quadrature filter pairs:

$$g(x, \sigma, \theta) = L_2(\sigma, \theta) * I(x) \quad (3.27)$$

$$h(x, \sigma, \theta) = H_2(\sigma, \theta) * I(x) \quad (3.28)$$

where  $L_2(\sigma, \theta)$  is the Laplacian,  $H_2(\sigma, \theta)$  is the approximation of the Hilbert transform of  $L_2(\sigma, \theta)$  and  $\sigma$  is the standard deviation of the Gaussian kernel used to derive  $L_2(\sigma, \theta)$  and  $H_2(\sigma, \theta)$ . A complex polar representation can be written as:

$$g(x, \sigma, \theta) + ih(x, \sigma, \theta) = p(x, \sigma, \theta) e^{i\phi(x, \sigma, \theta)} \quad (3.29)$$

It uses steerable filters to make the features invariant to rotation. To reduce the system's sensitivity to brightness changes, they add a constraint to the minimum absolute amplitude which is similar to contrast normalization [Car04]. The final descriptor is constructed from the complex filter responses.

Empirical results compared their phased-based descriptor with another based on differential invariants. They show that phase-based local feature leads to better performance when dealing with common illumination changes and 2-D rotation.

### 3.2.2 Histogram Descriptors

Histogram-based approaches have been widely used in image retrieval applications. Originally, they were constructed from the entire image intensity values. Global methods mainly used them because of their dimensionality reduction facilities. More recently, they have been extended to describe local regions efficiently using either color, or information obtained from different filters (*e.g.* gradients) applied on the image region.

In the common mathematical sense, a histogram is simply a mapping that counts the number of observations that fall into various disjoint categories (known as bins). Thus, if we let  $N$  be the total number of observations and  $n$  be the total number of bins, the histogram  $h_k$  meets the following conditions:

$$N = \sum_{k=1}^n h_k \quad (3.30)$$

In the following, description techniques based on histograms are presented. They mainly differ in the nature of the partitions (*e.g.* color, filter response,...) in which they will accumulate votes. We subsequently describe Color and Texture Histograms, Spin Images, Shape Context, and the Scale Invariant Feature Transform (*i.e.* SIFT).

#### Color Histograms

A straightforward way to build an histogram is to vote for the intensity value corresponding of each color channel in a reduced discretized color space. This concept was introduced under the name of *color histograms* [SB91]. They were originally tri-dimensional and based on widespread RGB colorspace.

By construction, this approach is robust to scale and rotational transforms. Unfortunately, it completely discards the spatial structure of the scene. Moreover,

the use of color histograms based on RGB color space is not ideal. Indeed, this colorspace is very sensitive to the illumination conditions and is not perceptually uniform. This leads to unexploited and redundant bins if using a uniform quantization. The computer vision community has investigated the normalization of color histograms [JV96], as well as color spaces that improve uniformity such as HSV [SC95], CIEl<sub>uv</sub> [STLC97] or CIEl<sub>ab</sub> [CTB<sup>+</sup>99]. These color spaces often allow for better recognition.

Although they are very popular in global methods, the use of color histograms to describe local regions is marginal. We noticed image retrieval [CCH01], color image segmentation [LP99] and object tracking [ZK04].

### Texture Histograms

The texture histogram (*i.e.* texton histogram) exploits different filter outputs to encode the distribution of the local intensity spatial structure (*i.e.* *texture*) over an image region. To build a texture histogram, a three-step procedure is applied.

- the image region is first filtered using a filter bank.
- then each pixel is represented by a multidimensional feature vector obtained by concatenating the corresponding filter responses.
- finally, the spatial distribution of the representative local structural features over the region is approximated by computing a multidimensional histogram.

Possible techniques to compute texture histograms include; the use of first-order Gaussian derivatives, the Laplacian as linear filters [SC96, SC00], and multi-scale measure based on Gaussian derivatives [LS03]. Recently, advances [OD04] in texture histograms focused on local height variations to compensate the shadowing and occlusions effects.

### Spin Images

SCHMID *et al.* [SLP03, Laz06] proposed to use an intensity-based rotation-invariant descriptor in a texture recognition system. The idea is based on the spin images introduced by JOHNSON *et al.* [JH99].

**Definition 3.4.** An intensity domain SPIN image is a 2D histogram encoding the intensity distribution of an affine-normalized patch. The two dimensions of the histogram are the distance from the center  $d$  and the intensity value  $i$ .

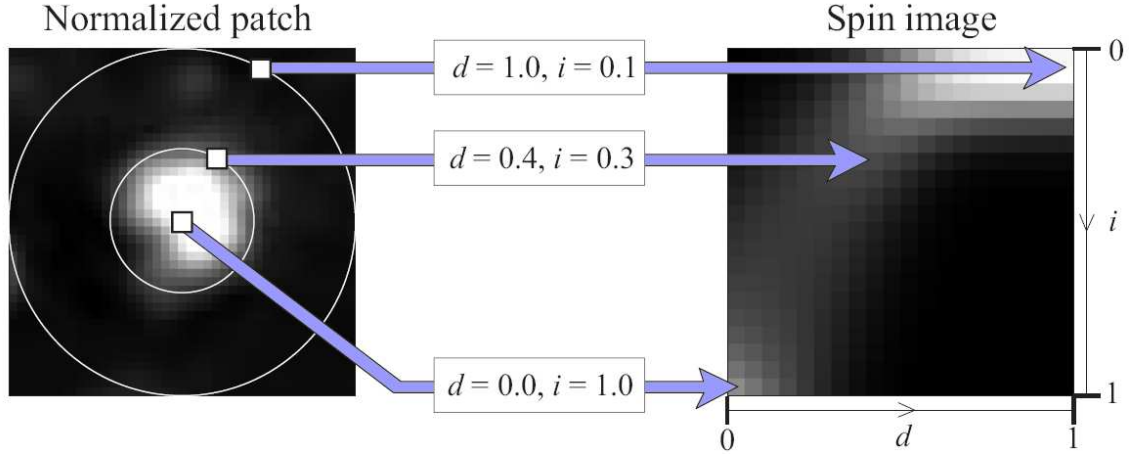


Figure 3.6: Construction of SPIN images. Three sample points in the normalized patch (left) map to three different locations in the descriptor (right) (Modified from [Laz06]).

As it can be seen in Figure 3.6, the slice of the SPIN image corresponding to a fixed distance is the histogram of the intensity values of pixels located at a distance from the center. Since the  $d$  and  $i$  parameters are invariant under orthogonal transformations of the image neighborhood, spin images offer an appropriate degree of invariance for representing affine-normalized patches.

In their implementation [SLP03], 10 bins were used for distance and intensity value, resulting in 100-dimensional descriptors. More precisely, they implemented the spin image as a “soft histogram”. In this type of histogram, each pixel within the considered region contributes to more than one bin. The contribution  $c$  of a pixel located in  $x$  to the bin indexed by  $(d, i)$  is computed as:

$$c_x(d, i) = e\left(-\frac{(|x - x_0| - d)^2}{2\alpha^2} - \frac{|I(x) - i|^2}{2\beta^2}\right), \quad (3.31)$$

where  $x_0$  is the location of the center pixel, and  $\alpha$  and  $\beta$  are the parameters representing the “soft width” of the two-dimensional histogram bin. To obtain good results, the patch size should be around  $10 \times 10$ .

### Shape Context

Another kind of local descriptor based on histogram computation has been introduced by BELONGIE *et al.* [BMP00]. The basic idea behind shape contexts is to

encode the relative distribution of neighbors to each edge point of a given contour. Thus, this detector requires a preliminary step that used a standard edge detector [Can86] to find the position of edge points.

Given a set of  $n$  points from an image sampled along a contour  $P$ , such as

$$P = \{p_1, p_2, \dots, p_n\}, p_i \in R^2, \quad (3.32)$$

Shape Context uses a log-polar decomposition of the circular image region (with a precision of 10 degrees) of each point  $p_i$ , and counts neighboring edge points in each spatial bin. This results in a 36-dimensional descriptor. Scale and shift invariance are obtained by normalizing distances by the mean distance between all points.

This technique is especially well suited when edges can easily be detected in images. It was successfully used in several image matching applications [BMP01, BMP02, MBM05].

### SIFT Descriptors

Currently, the most employed histogram-based technique is the Scale Invariant Feature Transform (SIFT) [Low99, Low04]. This descriptor was introduced by LOWE and produces a scale and orientation invariant characterization of interest points.

The idea is first to compute the orientation of the gradient and its magnitude at each sample point of the region around the interest point. Each region is divided in subregions of size  $4 \times 4$ . Values of each sample point of a subregion are then accumulated into orientation histograms (8 orientations) where each column corresponds to the sum of the gradient magnitudes in that direction within the subregion. The computed SIFT descriptor simply stores these values and has a dimension of  $8 \times 4 \times 4 = 128$  elements. The general procedure to compute SIFT descriptors is given in Algorithm 7 and illustrated in Figure 3.7 where a SIFT descriptor is computed on a Southern Crab Nebula image acquired by the Hubble telescope.

This technique leads to a high dimensional, but not prohibitive, description of interest points and generally offer good results for a wide variety of context. Indeed, the SIFT approach has successfully been used in many various projects such as object recognition, motion capture [PH03], or robot localization [SLL01].

The high performance obtained by SIFT descriptor can essentially be explained by two factors. First, it takes parts of the advantages of histogram techniques that reliably represent the region in a compact and robust way. Second, the use of gradient information describe the spatial structure in a sparse fashion and present good invariance properties in presence of illumination changes.

---

**Algorithm 7** Scale Invariant Feature Transform (SIFT)  $\{x, y, \sigma\}$ 

---

1: // **estimate local orientations**2: **for each** point  $\{x', y'\}$  in the neighborhood of  $\{x, y\}$  **do**3:   Compute the gradient magnitude and orientation at  $x', y'$  for given scale  $\sigma$ 

$$m(x', y', \sigma) = \sqrt{L_x^\sigma(x', y')^2 + L_y^\sigma(x', y')^2}$$
$$\tan\theta(x', y', \sigma) = (L_y^\sigma(x', y')/L_x^\sigma(x', y'))$$

4: **end for**5:  $M \leftarrow$  Form a 36-bin histogram from gradient orientations  $\theta(x', y', \sigma)$  where each orientation is weighted by its magnitude and by a circular Gaussian centered at  $\{x, y\}$ 6:  $\theta \leftarrow \max(M)$  // *Locate the highest peak in the histogram*7:  $\theta_{acc} \leftarrow$  fit a parabola to the 3 histogram values closest to the maxima  $\theta$  to interpolate the angle with a better accuracy.8: // **build the descriptor**9:  $W \leftarrow$  decompose the region centered on point  $\{x, y\}$  at scale  $\sigma$  into 16 windows.10: **for each** window  $w_i \in W$  **do**11:    $H_i \leftarrow$  compute a 8 bins histogram of its gradient orientations relative to  $\theta_{acc}$ 12:   weight each observation by the magnitude and a Gaussian centered at  $\{x, y\}$ 13: **end for**14: return  $\{H\}_{i=1\dots 16}$  // *SIFT descriptor*

---

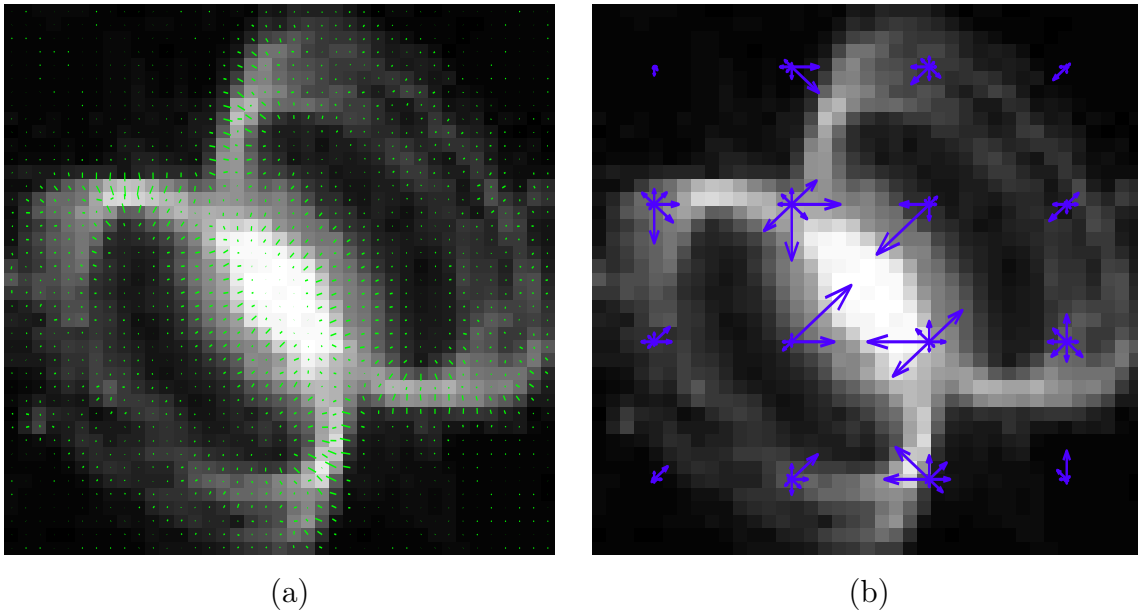


Figure 3.7: Center of the Southern Crab Nebula He2-104 (<http://hubblesite.org>). (a) Per pixel image gradients computed using local differences. (b) The SIFT descriptor for this image. The SIFT descriptor uses a 4x4 grid of histograms, where directions are quantized to 8 bins. The overall descriptor contains 128 bins, and is robust to small translations and warpings of the image.

The large attention given by researchers has led to several evolutions of the original SIFT descriptor:

**PCA-SIFT** [KS04] applies PCA to the normalized gradient image to produce a 36-dimensional vector.

**GLOH** *Gradient Localization and Orientation Histograms* [MS05] applies PCA to a log-polar location grid.

**RIFT** *Rotation Invariant Scale Transform* [Laz06] perform a rotation normalization before the computation of orientation histograms. To maintain rotation invariance, this orientation is measured at each point relative to the direction pointing outward from the center.

**SURF** *Speeded Up Robust Feature* [BTVG06] is an approximation of the SIFT descriptor. It performs at a lower cost in complexity and is therefore suitable for real-time applications. The computation can be carried on about 3-4 times faster (the running time on a  $800 \times 640$  image is 350ms).



### 3.2.3 Alternative Approaches

Besides convolution coded and histogram-based descriptors, many other techniques can be used to characterize the intensity of a local region. We review here some that have been considered in this thesis.

#### Patches of Intensity Values

A simple and straightforward technique to describe a region is to resample it to a square of fixed size (*e.g.* by bilinear interpolation), and to directly use the raw pixel values as a local descriptor. This idea has demonstrated comparable performance to state-of-the-art local description techniques [EC04]. Furthermore, it has been successfully applied to image classification by MARÉE *et al.* [MGPW05c]. Typically, extracted patches are resized to  $11 \times 11$  pixels, thus leading to  $363 = 11 \times 11 \times 3$  values in the case of color images. HSV color space, which defines a color in terms of hue, saturation, and value (or lightness), is used. It is often preferred to RGB because it allows for more robustness against illumination changes. This is because the distance function can more easily balance the importance of the lightness term.

Nevertheless, the major drawback of the basic version of this technique is that it is sensitive both to scale, orientation, and illumination changes. To obtain scale invariant patches, we proposed [SP06] to exploit Lindeberg's scale selection process to select the appropriate size at which the patch should be extracted<sup>2</sup>. Likewise, orientation is obtained by computing the gradient direction for the given region.

Another problem of patch extraction comes from the fact that a small displacement can induce a large difference in the distance measure between two descriptors. BROWN *et al.* [BSW05] explained how image patches can be made less sensitive to the exact feature location by sampling the pixels at a lower frequency (typically, the frequency at which the interest points are located). Given an oriented point  $(x, y, l, )$ , they sample a  $8 \times 8$  patch of pixels around the sub-pixel location of the point, using a spacing of  $s = 5$  pixels between samples. To avoid aliasing, the sampling is performed at a higher pyramid level, such that the sampling rate is approximately one per pixel. To obtain illumination invariance, the descriptor vector is normalized so that the mean is 0 and the standard deviation is 1. These manipulations have the effect of robustifying the patches to obtain a fair degree of invariance to most common image variations.

---

<sup>2</sup> The size of the patch is critically linked to the environment in which the task is performed. In Chapter 6, the best size and shape will be learned to produce *adaptive patch features*.

## Interest Point Groups

The description techniques presented so far were designed to describe the surrounding region of each point separately. When stable feature points can robustly be identified, an alternative is to use groups of features to describe the intermediate region between them. In this way, there is no one-to-one mapping between points and descriptors.

This idea was explored in the context of image matching by BROWN AND LOWE [BL02, Bro05]. They demonstrated that relative positions between interest point groups can be used to compute local 2D transformation parameters that relate two images. By using different number of points (2, 3, or 4), which are nearest neighbors, they were able to form local descriptors invariant to any 2D projective transformation (similarity, affinity, or homography).

We exploited these features as input into a hierarchical object recognition framework [Sca04]. Whereas interest points were originally located at difference-of-Gaussian function (DoG), we proposed to use Harris-Laplace points as inputs. This was motivated by the higher repeatability rate of the Harris-Laplace detector [SMB98]. Specifically, the class of point group we used is the triplet class (*i.e.* a set of three points). As mentioned by ISAKSSON [Isa02, GM04], a triplet invariant  $T = (f_1, f_2, f_3)$  is a three-tuple of local features (*e.g.* interest points) associated with a description function  $D = D(T)$ , that offers the three following invariance properties:

**Orientation invariance:**  $D$  is invariant to rotations of the object around arbitrary points in the image plane.

**Scale invariance:**  $D$  is invariant to scale changes of the object.

**Order invariance:**  $D$  is invariant to the ordering of  $f_1, f_2$ , and  $f_3$ .

In our implementation, triplets are formed by grouping interest points which are nearest neighbors<sup>3</sup>. After verifying that the points are not collinear, we retrieve the 2D transformation parameters  $t_{2D}$  to a canonical frame (made of  $13 \times 13$  values). In order to retrieve the right correspondence between the observation and the reference triple, for each point of the triple, we consider the angle formed by the two others. We assign the base point of the affine transformation to the point with the greatest angle.

<sup>3</sup>A Delaunay triangulation can also be applied to obtain a set of triangles.

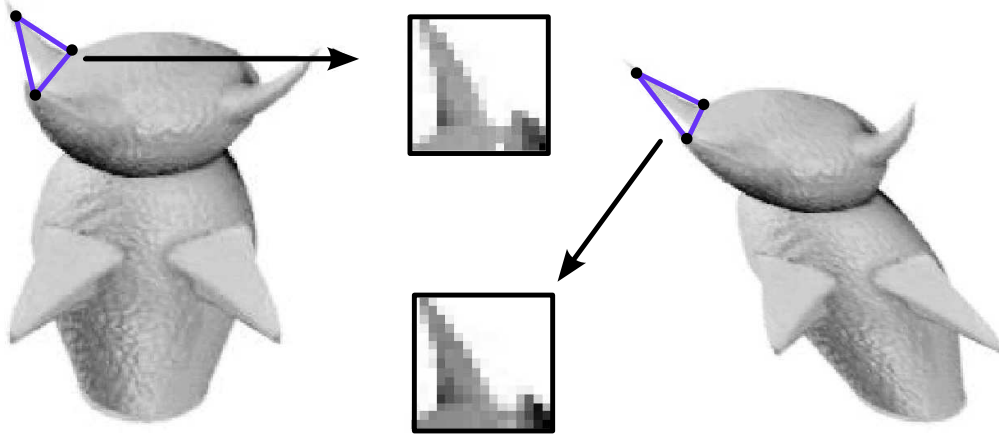


Figure 3.8: Correspondence of triplets of interest points in affine transform related views of a “Greeble” [GT97].

In parallel, other researchers have recently used groups of interest points. Among the successful applications, we notice; object class recognition based on histograms computed on triplets [TC04], 3D object recognition [GM04], and patch pairs [JM05]. Similar ideas were incorporated into powerful learning frameworks. For instance, LAZEBNIK *et al.* [LSP04] proposed to learn semi-local affine parts. Each of these consists of an interest point group that is first computed using a triplet of interest points, and then subsequently enriched by adding new points into the invariant set.

The use of interest point groups in the context of object recognition presents several advantages. First, it is fast and doesn’t require the application of time consuming image convolutions. Also it naturally offers invariance to rotation and scale and also to weak affine deformations. Furthermore, in presence of a low number of features, it allows to better cover the object. Despite these striking advantages, we observed that triplets considerably decrease the repeatability rate of the individual detectors [Det05]. This is explained by the fact that whenever a point of the triplet is missing the entire triplet is lost.

### Principal Component Analysis

Local description of an image region can be posed as a dimensionality reduction problem. A general solution is to use *Principal Component Analysis* (PCA) [SK87] to project previously normalized regions into a low-dimensional subspace. They

emphasized that PCA can be used to extract the first  $N$  eigen-images that best explain the variations in the training images.

Mathematically, the eigen-images correspond to the first eigen-vectors of the covariance matrix of the model database. Any normalized image can be projected as a vector of real numbers, and can be approximately reconstructed as the linear combination of the eigen-patches with coefficients.

PCA was originally used to describe entire images. Successful applications of PCA include face recognition [SK87, TP91], tracking [NMN94], object recognition [MN95a], and image retrieval [SW96]. Due to its popularity in global approaches, PCA spread also in local appearance methods [OI97, CdVC98, PPJV01]. Instead of being applied on the entire image, PCA is here carried out on local image regions. As already mentioned, a nice combination of *SIFT* [Low99] and *PCA* leads to a new class of low-dimensional descriptor, namely *PCA-SIFT* [KS04].

### Moment Invariants

Moment invariants are properties of connected regions in binary images that are invariant to translation, rotation, and scale. They are useful because they define a simply calculated set of region properties that can be used for shape classification and part recognition.

Generalized moment invariants [FS93, GMU96] have been introduced to describe the multi-spectral nature of the image data. The invariants combine central moments defined as

$$M_{pq}^a = \int \int_{\Omega} x^p y^q [I(x, y)]^a dx dy \quad (3.33)$$

where  $p + q$  is the order and  $a$  the degree.

The moments characterize shape and intensity distributions in a region. They are independent and can be easily computed for any order and degree. However, the moments of high order and degree are sensitive to small geometric and photometric distortions. Computing the invariants reduces the number of dimensions. These descriptors are therefore more suitable for color images where the invariants can be computed for each color channel and between the channels. TUYTELAARS *et al.* [TG00, TVG04] used these descriptors to represent the affine invariant regions.

### Biologically Motivated Multi-modal Features

The Ecovision features [KW04, KFW04, Pug06] are compact coding of image information represented in terms of local multi-modal image descriptors. Their strength

is to be multi-modal, in other words they combine different measures to improve the description quality. Both geometric information and structural image information are combined within these features.

Several local filters are applied to compute the following modalities:

**Orientation:** It corresponds to the dominant direction of the gradient direction,  $\theta \in [0, \pi]$ .

**Contrast Transition:** It is a description of the contrast transition. The contrast transition is coded in the phase of the applied filter and represents the local symmetry.

**Color:** Depending on the phase, two or three colors are sampled from the average color in the different image regions. If the transition is an edge, two colors are sampled, from both sides of the edge. In the other cases, three colors are sampled, the third one being the color of the line itself.

**Intrinsic dimensionality:** It characterizes the degrees of freedom of an image patch [CV01, FK03]. This information is used to distinguish between three types of region: homogeneous image patches, lines (*i.e.* edges) and junction-like structures.

### 3.2.4 Discussion

In this section, we have presented several semi-invariant local description techniques. In the context of object recognition, the system architecture should be independent of the description method. In contrast with local detection methods that may have a deep effect on the results by eliminating evidence in the image, the description methods do generally not (and should not) have a significant impact on the system performance<sup>4</sup>. Indeed, spatial relations, co-occurrences, and scene context can also be exploited to improve the recognition process. This leads us to the following section where we focus on the state-of-the-art object recognition methods.

---

<sup>4</sup>We will try to verify this assertion in Chapter 4 by evaluating local descriptors.

### 3.3 How: Object Recognition Methods

Sections 3.1 and 3.2 have respectively proposed an introduction to the detection (*i.e.* “Where” step) and to the description (*i.e.* “What” step) of local features in images. This chapter is now completed by presenting an overview of the main available methods to recognize objects. All these methods propose solutions to the following general question;

*How can a computer vision system recognize objects?*

We will see in this section that the expectations have evolved since the early attempts; it has passed from specific instance matching to the recognition of object categories. Nevertheless, as it has already been mentioned in the introductory chapter, actual techniques should be able to “recognize” objects in presence of real world conditions (*e.g.* with changes in viewpoint, illumination, . . .).

In order to give an insight into the extensive number of methods, a brief look at historical developments is first presented in the next subsection (Section 3.3.1). Then we take a closer look at the contemporary literature, and we analyze the specific merits and limitations of each approach. As it has been explained before, object recognition is made of three parts; learning, representation, and detection. Throughout this section, we will mainly focus on these points to describe the methods.

For readability purposes we split the approaches explained in this section into three categories: (1) approaches that are based only on appearance (Section 3.3.2), (2) statistical approaches based on appearance and geometry (Section 3.3.3) and (3) biologically motivated methods (Section 3.3.4).

This literature overview is current state-of-the-art and it should be mentioned that many of the approaches were not yet published when we started the work on our approaches (others published after or at the same time as our papers).

#### 3.3.1 A Critical View on the History of Object Recognition

Historically, early attempts at computerized object recognition took place in the 1950’s [Din55, Cho57, GSTK58, Ung59]. Much of this research has focused on two-dimensional pattern classification applications such as character recognition, fingerprint analysis, and microscopic cell classification. These experiments generally relied on basic correlation and template matching techniques. Results were satisfying on very constrained problems but did not generalize well.

Since that time, progress has been gradual but subsequent approaches have tended to focus on establishing theoretical frameworks where computers could carry out the necessary reasoning using mathematical tools. Therefore computer vision became intrinsically linked to many aspects of artificial intelligence and statistics. As consequence of this, the evolution of object recognition has been largely dictated by findings in these connected fields.

Every couple of years, a new technique slightly outperformed the preceding ones leading to an evolution of the best recognition system performance. Among the popular techniques, we can mention range data analysis [Agi72, Bin71] (range images store the depth of the scene, rather than intensity), alignment techniques [Low87, UB91] to find the best match, geometric invariants [RFZM93] where small sets of points are used to compute a viewpoint-independent descriptor which can act as a key for hashing into a database where models are stored.

Rather than reviewing the history of techniques <sup>5</sup> (of which an overview can be seen in Figure 3.9) it would be interesting to see what we can learn from this history. Many times, researchers believed to be close to a general and complete solution to this problem. However, it seems now much more difficult than expected. The computer vision community now admits that 50 years of research was not sufficient to solve the problem of object recognition. Therefore, it is crucial to understand why researchers failed to propose a universal approach to object recognition. In the following, a few possible explanations are described; they have been collected from various readings, conferences, and informal discussions.

### 1. A Difficult Problem

Recognizing a previously seen object in images seems natural and easy for human beings, but is actually widely acknowledged as a very difficult problem for computers. We already mentioned most of the basic reasons (viewpoint, clutter, occlusions,...) but the formulation of the problem itself is far from trivial. Typically, in computer vision, much attention is paid to the technical issues rather than on the understanding of the problem itself which most of the time is described by either too naive or too complex theories. Sometimes the solution may lie in the problem understanding itself. This was noticed by MARR AND POGGIO [MP77] in their recognition research in the field of

---

<sup>5</sup>Note that complete reviews on object recognition can be found in the following papers [DFP97, Mun98, Mun03, Fer05]

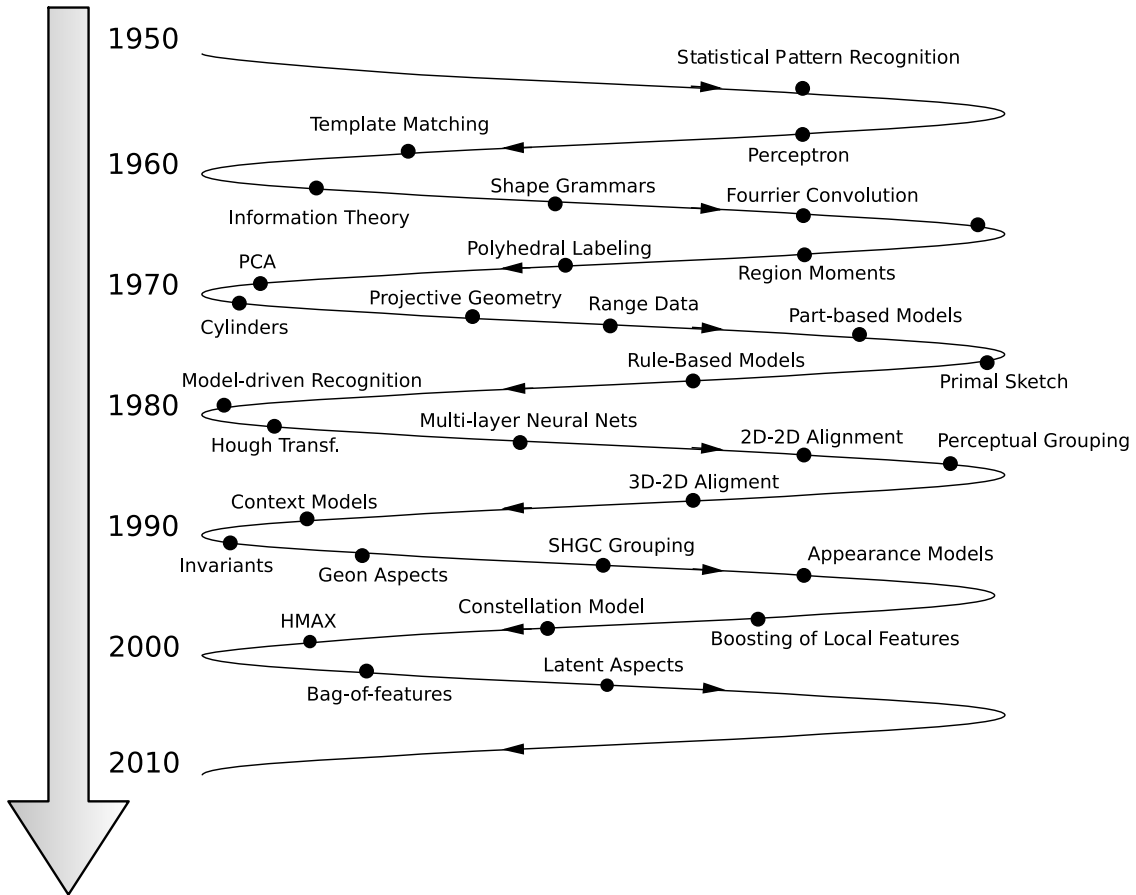


Figure 3.9: A history of some of the key ideas and paradigms for object recognition by computer (Modified from [Mun98]).

computational neuroscience. The following sentence [MP77] was considered by EDELMAN [EV01] as a central legacy of MARR’s career:

“ *The understanding of any information processing system is incomplete without insight into the problems it faces, and without a notion of the form that possible solutions to these problems can take.* ”

## 2. A Biased Interest

Historically, most advanced applications in computer vision have often been intrinsically linked to military applications such as recognizing buildings in aerial images, locating industrial parts in bins, and more recently biometric



applications (face recognition, iris). The large amount of research in these areas, and their quality may have led many researchers to re-use their models in object recognition. Thus reducing object recognition to an image matching problem where the detection consists in a search for correspondences between model and image features. Typically, this minimizes the importance of the learning process and focuses on straightforward recognition methods.

### 3. Empirical Methods

If we take a look at the historical overview shown in Figure 3.9, we clearly see that a large variety of methods and concepts have been employed. We have to acknowledge that a large part of research in this field has been dictated by empirical findings rather than strong high-level theories. Therefore there is currently a lack in both theoretical and representational knowledge of objects.

### 4. Biological and Psychological Evidence

In the 1920's, the *Gestalt* theory [Wer23] already proposed precise models of perception mechanism well before the first object recognition experiments on computers. It was shown that a number of factors determine grouping and therefore our way to perceive objects: proximity, similarity, common fate, good continuation, and closure. Moreover, the *Gestalt* theory also explored the learning theories of such properties, and how the recognition models (rules) are indeed obtained from past experience [Köh47]. These models were not only pure psychological theories but were first validated on animals (cats and monkeys) and later on humans. Logically, they should have constituted a big inspiration for many computer scientists. However, they have been marginalized and clearly neglected by researchers in computer vision. We must now acknowledge that some recent advances in object recognition tend to be close to this *Gestalt* theory.

Since we aim at developing a visual recognition system, it is essential for us to keep those issues in mind.

### 3.3.2 Appearance-Only Models

A straightforward approach to tackle the problem of object recognition is to consider local appearance-only models. The first attempts go back to the work of MEL [Mel97]. Then this idea was expanded by SIVIC *et al.* [SZ03]. A few months later, a paper using a similar idea [DWF<sup>+</sup>04] used the term “Bag of features” model (or “Bag of keypoints”) to name this kind of technique. The choice of this name is an analogy to the “Bag of words” representations used in text document analysis [Joa98]: local visual features are the visual equivalents of individual “words” and the image is treated as an unstructured set (*i.e.* bag) of these.

Originally presented as an object matching application [SZ03], its first application in object class recognition was proposed by CSURKA *et al.* [DWF<sup>+</sup>04, WAC<sup>+</sup>04]. The main idea behind this kind of framework is to represent each object class as a set of local visual feature classes without using any geometric information between parts.

In order to build such a *geometry free* model, they proposed to extract a collection of local features from training images. A visual codebook is obtained by clustering descriptors of local features using k-means [HW79]. Traditionally each cluster center corresponds to a visual word and is described by an appearance vector. Then for each training image, the system counts the number of occurrences of each feature class presents in the visual codebook. This leads to a vector that is labelled according to the corresponding object class. All the vectors are used to train a robust classifier. Recognition follows the same idea, except that support vector machines (SVM) that have been learned are now used to predict the class label. The original work achieved excellent classification performance on standard object recognition databases despite the lack of any location and geometrical information in the model. These astonishing results incited many other researchers to explore this technique.

Instead of using k-means to cluster the visual feature space, DORKÓ *et al.* [DS05] exploited the Expectation-Maximization (*EM*) algorithm to estimate a Gaussian mixture model (*GMM*) [Bis95]. Then a feature selection based on the mutual information criterion [Pap91] allows the system to find the most discriminative feature classes. The classifiers are built on the basis of these discriminant features. Experiments were conducted with success on a database consisting of different object classes. Results confirmed the importance of the feature selection step since many common features such as edges occur very often but are not useful to differentiate object classes. Other similar techniques [WCM05] try to merge classes of the visual

codebook.

Another approach is to build a robust classifier from a combination of weak classifiers, such as AdaBoost [FS97]. Because of its simplicity, this was largely used in bag-of-features approaches. Among the most successful applications, we can mention the research of OPELT *et al.* [OFPA04, OP05, Ope06] on recognition of object categories. Whereas other bag-of-features approaches only use one kind of local feature, they proposed to use several types of detectors and local descriptors (such as SIFT, differential invariants, ...). They considered them as the weak classifiers and combined them within a single robust classifier through AdaBoost.

MARÉE *et al.* [MGPW05c] have proposed to use a new kind of decision trees (called *Extra-Trees*) [GEW06] in order to solve visual classification problems. This “bag-of-features” approach describes images by a collection of local image patches randomly extracted from images. Decision trees are built in a supervised manner on the unstructured feature set obtained from the training set. During recognition, randomly extracted patches are used as input to the decision trees to predict the presence of the class. Contrary to other bag-of-feature methods, an advantage is that it does not require the construction of a visual codebook. This approach has been evaluated on various image classification datasets involving the classification of digits [Mar05], faces [MGPW04], objects [MGPW05c], buildings [MGPW05b], photographs, and biomedical images [MGPW05a].

The performances of bag-of-features approaches are intrinsically linked to the classification method. Kernel-based learning methods [SS01] have recently gained interest in pattern recognition. Kernel methods offer a modular framework that exploits a Kernel function to transform the data into a higher dimensional space. For instance, the Kernel function can be used to approximate the partial matching between two feature sets. Using this concept, GRAUMAN AND DARRELL [GD05] have developed a kernel that approximates the optimal partial matching between two unordered sets of local features. This is done using a pyramid structure. More recently, it was extended to cope with the image categorization problem [LSP06].

One of the best performing kernel-based method is the framework developed by ZHANG *et al.* [ZMLS07] which uses a kernel based on the Earth Mover’s Distance [RTG00]. Mercer Kernels [Lyu04, CWN04] are also very efficient to match the features. They have been showed to achieve excellent results on various object recognition problems.

At the moment, systems based on Kernels generally obtain the best recognition rate in categorization tasks. This can be explained by the fact that they have recently

allowed to add spatial information to an already high-performance framework only based on appearance.

Largely influenced by appearance-only approaches, gradually more complex models have been developed. They tried to add spatial information in a similar framework. Among them [OPZ06, Ope06, LSP04, Laz06] we can mention AGARWAL *et al.* [AR02], who included pairwise relations between features. In a similar spirit, one follower of the bag-of-features model are the latent probabilistic models; which is described in the following.

### Latent Semantic Analysis

The Bag-of-features strategy inspired more complex models, like the Latent Semantic Analysis (LSA) [DFL<sup>+</sup>88, DDF<sup>+</sup>90]. This method has also been used in the text analysis community to extract coherent components from a collection of documents.

In computer vision, researchers turned recently to probabilistic versions of LSA, namely probabilistic Latent Semantic Analysis (pLSA) [Hof99, Hof01] and its Bayesian form, the Latent Dirichlet Allocation (LDA) [BNJ03]. The general idea is to assume that visual features are generated from latent aspects (or topics). These topics can be represented by hidden high-level variables that relate the observed visual features with their class label.

The construction of a model using such a technique can be described as follows. It starts from a set of visual words  $\mathcal{V} = \{w_i\}$  and each image is described by a vector of fixed size  $\mathcal{V} = |\mathcal{V}|$  where each bin contains the number of occurrences of word  $w_i$  in the image. A set of  $\mathcal{N}$  images is described using a  $\mathcal{V} \times \mathcal{N}$  matrix. From this relationship between images and visual features, topic models try to find an indirect relationship through topics: first between images and topics, then between topics and features, with the assumption that these two relationships are independent. This is generally estimated through an Expectation-Maximization algorithm. Note that it is possible to add spatial informations into the models.

This kind of approach has recently been applied to various applications of recognition such as scene [FP05] and object categorization [STFW05, LJ06], unsupervised learning of object class [SRE<sup>+</sup>05].

### 3.3.3 Statistical Part-Based Models

The geometric structure of an object may be a determinant factor in the process of recognition. Most advanced methods try to integrate this information in their models. However, this introduced several problems that have to be taken into account such as scale change, rotation, deformation, and occlusion. The most popular way to include geometrical information into the object model is to consider the object as a set of spatially related parts.

Part-based models mainly originate from FISCHLER AND ELSCHLAGER [FE73] theories. Their object model was a combination of parts in a geometrical model. Each part represented local visual properties and spatial configurations were captured by relational functions. Since those early developments [Yui91, BP93, LVB<sup>+</sup>93, AG99], this model gradually evolved to integrate more robust part detectors, better models of uncertainty (Non-Gaussian probability distributions, mixture models, particle sets), inference (Belief Propagation), and powerful learning (EM) and boosting (AdaBoost) methods.

At the end of the last century, a new probabilistic part-based model emerged from the work of BURL *et al.* [BP96, Bur96]. It was later generalized under the name of “Constellation Model” by its followers WEBER *et al.* [BWP98, Web00]. A few years later, FERGUS *et al.* [FPZ03, Fer05] added appearance variability and scale invariance to this approach. Other extensions such as the use of more robust local features have also been proposed. In parallel several other similar models have been developed, they often only differs in minor technical issues (*e.g.* feature detector, ...). In the next paragraphs, we propose an overview of the constellation model, hierarchical part-based models, and shape matching methods.

#### Constellation Model

The Constellation Model is a statistical model designed for recognizing object classes. It represents an object model as a set of parts. Each part  $p$  has an appearance  $\mathcal{A}_p$  and relative scale  $\mathcal{S}_p$ . Shape  $\mathcal{X}$  is object centered (*i.e.* relative to the center of the object) and represented by the mutual spatial position of the parts.

Since the entire model  $\mathcal{M} = \{\mathcal{A}, \mathcal{S}, \mathcal{X}\}$  is probabilistic, appearance, scale, and shape are modeled by Gaussian probability density functions. Typically, the appearance  $\mathcal{A}_p$  of a part  $p$  is represented by a Gaussian density within a feature space reduced using Principal Component Analysis (PCA). This distribution is described by a mean  $\mu_p^{\mathcal{A}}$  and corresponding variance  $\Sigma_p^{\mathcal{A}}$ . Similarly, the scale of a part is de-

notated by a Gaussian which has parameters  $\mathcal{S}_p = \{\mu_p^S, \Sigma_p^S\}$ . Finally, the shape is represented by a joint Gaussian density of the locations of features within a scale-invariant space;  $\mathcal{X}_p = \{\mu_p^X, \Sigma_p^X\}$ . Equation 3.34 illustrates the form of these matrices for a three-part model.

$$\mu^X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \quad \Sigma^X = \begin{pmatrix} x_1x_1 & x_1x_2 & x_1x_3 & x_1y_1 & x_1y_2 & x_1y_3 \\ x_2x_1 & x_2x_2 & x_2x_3 & x_2y_1 & x_2y_2 & x_2y_3 \\ x_3x_1 & x_3x_2 & x_3x_3 & x_3y_1 & x_3y_2 & x_3y_3 \\ y_1x_1 & y_1x_2 & y_1x_3 & y_1y_1 & y_1y_2 & y_1y_3 \\ y_2x_1 & y_2x_2 & y_2x_3 & y_2y_1 & y_2y_2 & y_2y_3 \\ y_3x_1 & y_3x_2 & y_3x_3 & y_3y_1 & y_3y_2 & y_3y_3 \end{pmatrix} \quad (3.34)$$

During learning, the system estimates the parameters of the model  $M$  for a given number of parts  $P$ , where

$$M = \{\mu^A, \Sigma^A, \mu^S, \Sigma^S, \mu^X, \Sigma^X\}_{1\dots P} \quad (3.35)$$

The Expectation-Maximization (EM) algorithm is used to find the parameters that maximize the likelihood  $\hat{\theta} = \arg_{\theta} \max p(\mathcal{A}, \mathcal{S}, \mathcal{X}|\theta)$  (*i.e.* the probability of observing the training data given the model parameters). Recognition is performed on a test image by first detecting local features, and then evaluating the regions in an exhaustive manner, using the model parameters estimated during the learning.

This framework has demonstrated very good results on several difficult image databases. The novelty of the constellation model is to learn appearance and shape variability within the same model. The counterpart of this is the complexity of learning which becomes intractable when more than 6-7 parts have to be used. The algorithm also required the user to pre-determine the number of parts. Moreover it is only able to learn a given view of an object and cannot deal with large viewpoint changes.

To reduce the complexity in both learning and recognition, FERGUS *et al.* [FPZ05] proposed to prune out the spatial relations by using a more compact topology. This led to a new evolution of the model called Star Shape Model (Figure 3.10). It is a graph in which all edges are incident to a reference node. This notion was extended to  $K$ -Fan models by FELZENSZWALB AND HUTTENLOCHER [FH05]. They compared the performance of object models built with different numbers of reference nodes (*e.g.* 1, 2 or 3). Results indicated that an increasing number of reference nodes leads to a better recognition.

### Hierarchical Models

Before the emergence of appearance-based models, hierarchical representations were common in computer vision (*e.g.* [MN78, Ett88, MGA89, Coo89]). Objects were represented as compositional hierarchies of rigid primitive forms, and detection was performed by searching for appropriate combinations of these primitives. With the advance of statistical learning tools and local appearance description methods, object recognition has turned to very simple representations. They are simple in the sense that the object is only represented by one level of visual features. A few years ago, hierarchical systems have regained their popularity [Pen90, SPS00, Pia01]. However, the coupling of appearance methods and hierarchical representation was still problematic on real conditions images. To cope with this problem, different models [BT05, EU05a, FBL06, AT06, OB06] have been proposed recently. The first two models are reviewed below:

**Three-Layer Model.** BOUCHARD *el al.* [BT05, Bou05] have introduced a three-layer generative model for coding the geometry and appearance of visual object classes. The object model is a collection of connected parts containing assemblies of subparts. It is illustrated in Figure 3.10 (bottom center). Spatial relations are described by parametric, probabilistic spatial transformations that follow a Gaussian distribution, of mean  $\bar{\mu}_{qp}$  (and variance  $\Sigma_{qp}$ ):

$$\mu_{qp} = \begin{pmatrix} s & 0 & a \\ 0 & s & b \\ 0 & 0 & 1 \end{pmatrix} \quad (q \text{ being } p\text{'s parent}) \quad (3.36)$$

where  $s$  is the relative scale and  $(a, b)$  is the relative translation.

For more simplicity, they used an object centered model where each part  $p$  has a relative position  $\mu_p$  to the center of the object. The probability density model for features of class  $p$  and location  $\mu_p$  takes the form of a mixture of transformations (*i.e.* relative positions) given its parents locations  $\mu_q$ :

$$P_p^{\text{loc}}(\mu_p | \{\mu_q\}) = \sum_q \tau_p(q) \mathcal{N}(\mu_q \mu_p^{-1} | \bar{\mu}_{qp}, \Sigma_{qp}) \quad (3.37)$$

where the mixture weights  $\tau_p(q)$  are the model parameters representing the prior probabilities of  $p$ 's parent being  $q$ .

First level features are located with the Harris-Laplace detector and the SIFT descriptor is used to extract an appearance vector  $\mathcal{A}$  of the local region which

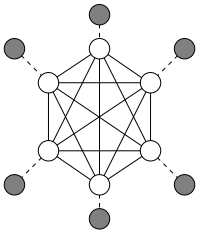
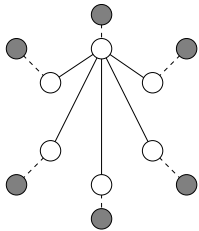
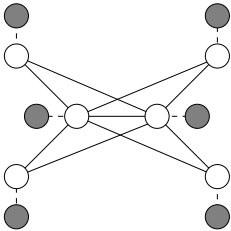
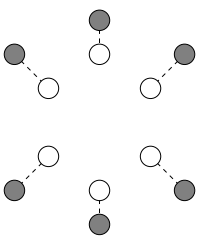
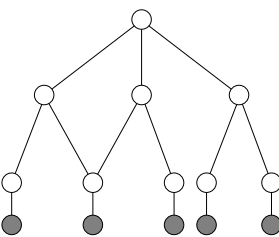
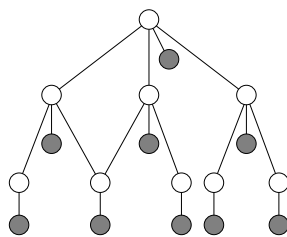
		
Constellation Model [FPZ03]	Star Shape [FPZ05]	K-Fan [CFH05]
		
Bag of features [SZ03]	Hierarchy [RP99, BT05, SP05]	Appearance Hierarchy [EU05a] & This thesis

Figure 3.10: Different Object Recognition topologies under Graphical Model formalism. Each white circle represents an object part (*i.e.* feature class) and denote a hidden variable. An object part may also have an appearance likelihood if it is linked to a dark circle. Edges denote spatial relations between feature classes.



is assumed to be Gaussian with variance  $\Sigma^A$ . Similarly to the constellation model, the Expectation-Maximization (EM) algorithm is used to find the parameters  $\{\bar{\mu}_p^P, \Sigma_p^P, \mathcal{A}_p, \Sigma_p^A, \tau_p, \pi_p\}_{1\dots P}$  that maximize the likelihood of the model. Some experiments on real images demonstrated the ability of the model to fit complex natural object classes when orientation of the object was assumed to be known.

**Top-down Hierarchical Decomposition.** EPSHTEIN *et al.* [EU05a, EU05b] introduced a method for automatically extracting hierarchical feature models for object recognition. The extraction process proceeds in a top-down manner. It first extracts informative top-level fragments, and then employs a recursive strategy to break down object parts successively into their own optimal components. The hierarchical decomposition terminates with simple features that cannot be decomposed into simpler features. Typically, these hierarchical models are constituted of three to four different levels.

The strength of this method is that it is able to learn the entire hierarchy, the different features and sub-features, and their optimal parameters during the training phase. Experiments demonstrated that the use of feature hierarchies significantly improved classification results compared with similar non-hierarchical features.

### Shape Matching

The problem of object recognition can be posed as the matching between two object shapes. BERG *et al.* [BBM05] presented a framework that uses a deformable shape matching algorithm to recognize object classes. Their method sets up correspondence as an integer quadratic programming problem, where the cost function relies on similarity of corresponding geometric blur descriptors as well as the geometric constraints between pairs of corresponding feature points.

DICKINSON *et al.* [SD02] developed several approaches to the shape matching problem using graph-based models. They have also widely contributed to the enhancement of hierarchical representations of objects. In contrast with other methods using EM-like algorithms, they formulate learning differently. The problem for them is to find the lowest common abstraction among a set of graphs [Seg88]. Object recognition is obtained through the matching between image features and model features. To perform this matching, they present a framework capable to

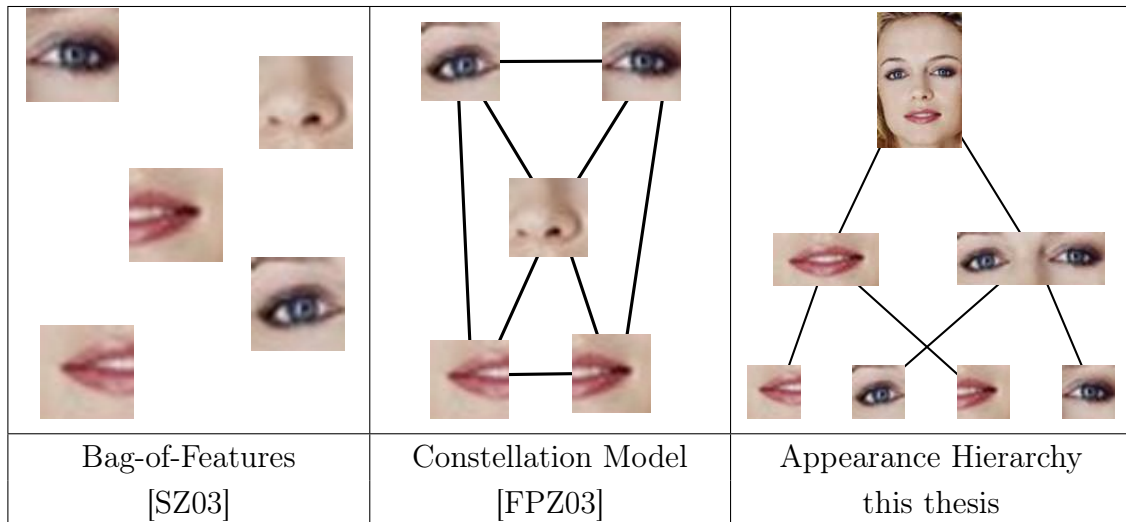


Figure 3.11: Intuitive visual overview of different Object Recognition topologies.

find many-to-many correspondences [KSDD03] established by the Earth Mover’s Distance.

Such matching methods have been employed in object class recognition [DSD<sup>+</sup>04, DSK<sup>+</sup>05, LSD05] and Skeleton shape matching [vEMT<sup>+</sup>06]. The representational power of their hierarchy is very inspiring and their learning method is efficient. However, the major problem by using skeleton-based representation is occlusion because it modified the shape.

### 3.3.4 Biologically Motivated Models

Many biologically plausible models are available in the neuroscience literature [Fuk80, Mel97, LBBH98, SUS02] to explain the mechanism of visual recognition<sup>6</sup>. However, only a few of them have been implemented and evaluated on standard object recognition databases. HMAX is one of them, it is a hierarchical computational model of the visual cortex proposed by RIESENHUBER AND POGGIO [RP99] and inspired by Fukushima [Fuk80]. It is originally based on experimental findings on the ventral visual pathway and demonstrated its ability to make predictions of biological models.

In its basic formulation, HMAX architecture is made up of alternating levels of  $S$  units, which perform pattern matching, and  $C$  units, which take the *max* of the  $S$  level responses. The first layer,  $S_1$ , consists of a set of oriented Gaussian-

<sup>6</sup>see the PhD thesis of SERRE [Ser06] for a complete overview.

derivative filters computed at different scales. The second  $C1$  layer is obtained by performing max operations over  $S1$  filters that present the same orientation, but different scales and positions over its neighborhood. In the  $S2$  layer, the simple features from the  $C1$  layer are combined to form intermediate feature detectors. Finally, each  $C2$  layer unit takes the max over all  $S2$  units differing in position and scale for a specific feature. Instead of taking the maximum value, it is also possible to compute the *sum* of  $S1$  responses. Alternating between *max* and *sum* operations gives both invariance and selectivity to the HMAX hierarchy.

Recently, several improvements [SWP05, Ser06] have been proposed to the original framework. The most interesting is the learning of a generic shape codebook from V2 to IT, which provides a rich representation to task-specific categorization circuits in higher brain areas. The hierarchical architecture builds progressively more invariance to position and scale while preserving the selectivity of the units. This vocabulary of tuned units is learned from natural images during a developmental-like, unsupervised learning stage in which each unit in the intermediate layers becomes tuned to a different patch of a natural image.

In the same spirit, WERSING *et al.* [WK02, WK03] proposed another biological hierarchical model using different feature matching and pooling stages. Another improvement was recently proposed by MUTCH *et al.* [ML06] where they applied a similar framework to perform object categorization. They showed that such a biologically-based model can compete with other state-of-the-art approaches to object categorization, strengthening the case for investigating biologically-motivated approaches to object recognition.

### 3.3.5 Discussion

Despite its simplicity and computational efficiency, a bag-of-features is an extremely impoverished representation for object classes. It ignores all geometric information about the object class, and therefore fails to represent the geometric structure of the object class. Moreover it is not able to distinguish between foreground and background features<sup>7</sup>. It can also be adversely affected by clutter and influenced by discriminant background features when constructing the classifier. Therefore it is understandable that some researchers [Ope06, Tar06, Tar04, Mun98] do not consider bag-of-features models as real (geometric) object models.

---

<sup>7</sup>Interestingly, because of this, bag-of-features systems can exploit the background information to recognize objects. Some objects can be recognize only using the contextual information.

An example of arguments in that sense can be found in a conversation between T. KANADE [TCRK01] and J. L. MUNDY [Mun98, Mun03]:

“ *Dr Mundy said that [...] we need language for describing objects. I'm not sure about that. [...] if we suddenly begin to say, humans seem to describe them by language, and therefore we need language as a tool, I think that's wrong. [...] If there is any theory here, somehow we have to develop sound mathematical theory for perceptual grouping that relates observable properties with the description of the object, not a linguistic theory that relates symbolically represented properties with objects. Simply saying that geometry is done and the language to describe functions is the next direction sounds like we are going back to the old days before geometry pattern recognition, when all sorts of soft AI-ish ideas were dominant.* ”

The main problem is that the development of a theoretical framework to represent visual features is often more complex. After evaluating feature detectors and descriptors in Chapter 4, we will focus in Chapter 5 on the definition of such a representation which seems to be a critical point in current approaches.

# Feature Detectors and Descriptors: A Comparative Evaluation

---

After having reviewed the main methods to detect and describe local visual features from images, we now evaluate some of them through a common experimental protocol. The particular interest of this chapter is to explore the following question:

*What are the best performing detectors and descriptors?*

The idea here is to compare them quantitatively by evaluating their performance on an image matching task. To be able to evaluate available techniques, we need to consider some formal criteria that make them desirable for object recognition. Among the properties that may characterize the performance of a local detector, we can find:

**Precision:** the distance between the detected point and the effective point,

**Robustness:** the capability of the detector of coping well with variations,

**Density:** the average number of detected regions per fixed amount of pixel area,

**Information content:** the distinctiveness of the local region at interest points,

**Complexity:** the number of operations needed to achieve the detection.

In this work, we are mainly concerned with robustness, precision and density matters. Robustness and precision properties can be quantified using a *repeatability measure* that is obtained by computing the average number of correspondences between two images.

**Definition 4.1.** *Repeatability* is the ability to detect features across views that only differ by geometric and/or photometric transformations.

The efficacy of a descriptor mainly relies on two properties:

**Invariance:** is the degree to which the same feature can be matched regardless of photometric and geometric variations,

**Selectivity:** measures how well two different features can be discriminated.

Before describing the protocol used and discussing the results obtained by the detectors and the descriptors, respectively in Section 4.2 and Section 4.3, we introduce in the following section the image dataset that will be used in our these experiments.

## 4.1 Image Dataset

The images used to evaluate the feature detectors and descriptors are illustrated in Figure 4.1. The first image of each set is considered as the reference image. Various different changes in imaging conditions are examined: viewpoint and scale changes (a,b), image blur (c), and illumination change (d).

The first two image sets were taken by JODOGNE [JP05b]. It consists of pictures of the Montefiore Institute (University of Liège). Originally, they have been used to simulate a navigation task in a closed-loop learning environment [JP05b, Jod06]. Each set consists of color images taken from a fixed position of the camera, so that images are related by homographies (plane projective transformation). We choose an image of the set as the reference image. The homographies between this reference image and the other images are computed using a standard robust homography estimation algorithm [HZ04]. This algorithm uses correspondences of feature points to estimate the homography. To minimize the influence of one detector versus the others during the estimation the homographies, all the detectors were used to produce the features.

The third image set (Hong-Kong Tramways) consist of the same image repeatedly blurred by a Gaussian kernel ( $\sigma = 10$ ). The final set (Hong-Kong Tramways) consists of images with severe progressively decreasing lighting condition.

## 4.2 Evaluation of Feature Detectors

In the following, we present an evaluation of some of the most popular feature detectors: Harris-Laplace [MS01], Harris-Affine [MS02], Hessian-Affine, Hessian-Laplace, MSER [MCUP02] and EBR [TVG04]. We begin by presenting the protocol used to evaluate the methods. Then the results obtained for the tested detectors are presented and discussed.

### 4.2.1 Experimental Protocol

Through these experiments, we try to measure how well a detected region in the reference image can match the same area in another view of the same scene. The homography between the images is used to determine ground truth correspondences for the detectors. This is possible since we know for each position in the reference image the corresponding position in any other view.

The experimental protocol presented in this section is similar in organization to that used by MIKOLAJCZYK *et al.* [MTS<sup>+</sup>05] to evaluate affine invariant detectors. Here, the objective is to measure the repeatability and the density of different detectors across various image transforms and degradations.

Each local feature is not only defined by a position  $(x, y)$  in the image but also by a scale factor  $s$ . This determines a circular local region around the feature point. In the case of affine invariant detector, such as MSER, Hessian-Affine, EBR and Harris-Affine, the region is elliptical. To quantify the repeatability, the idea is to calculate the relative amount of overlap between the detected regions in the reference image and the regions detected in the other image projected onto the reference image using the homography relating the images [MTS<sup>+</sup>05]. More precisely, two regions are said to correspond if the overlapping error is sufficiently small:

$$1 - \frac{R_{\mu_a} \cap R_{H^T \mu_b H}}{R_{\mu_a} \cup R_{H^T \mu_b H}} < \mathcal{E}_O \quad (4.1)$$

where  $R_\mu$  represents the region defined by  $x^T \mu x = 1$  and  $H$  is the homography matrix relating the two images. The union of the regions is  $R_{\mu_a} \cup R_{H^T \mu_b H}$  and  $R_{\mu_a} \cap R_{H^T \mu_b H}$

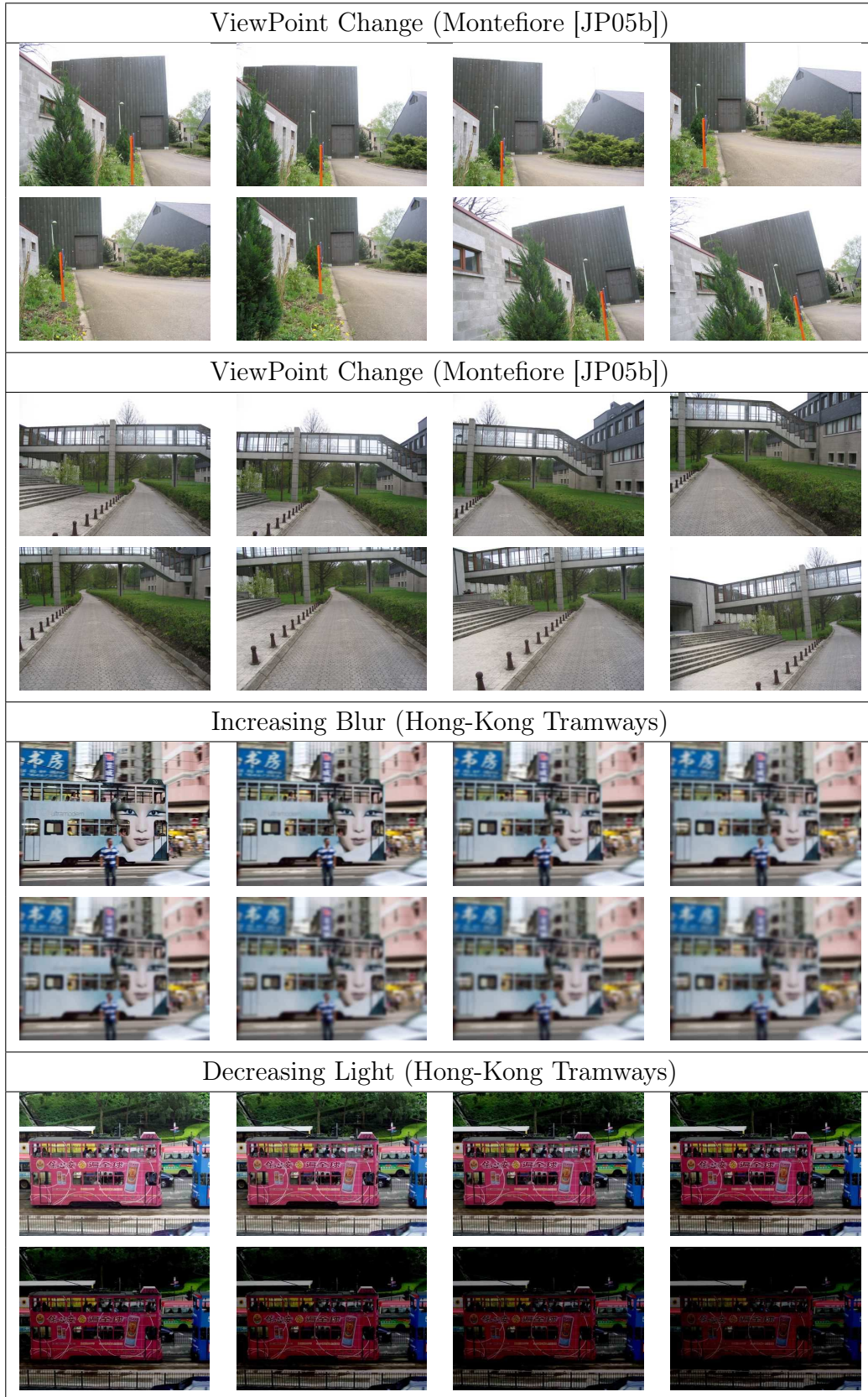


Figure 4.1: Image sets used for evaluating feature detectors.



is their intersection. During the experiments, the maximum overlapping error  $\mathcal{E}_O$  was fixed to 40%.

To measure repeatability, only the regions located in the part of the scene present in both images were taken into account. The scale factor between regions of two images can influence the measure. To compensate this effect, regions are rescaled by computing a scale factor to transform the region from the reference image to a canonical region. Prior to the estimation of the overlapping error (Equation 4.1), this relative scale factor is applied to both reference and detected region which has been mapped onto the reference image.

## 4.2.2 Results

In our experiments, we provide repeatability measures and numbers of occurrences for each detector on the all set of images. These measures can be shown to have a direct impact on the accuracy of an object recognition system based on local feature detectors.

By considering only the repeatability results, maximally extremal stable regions (MSER) generally perform best, but seem less robust in the presence of blur. Harris-affine and EBR are generally below the other detectors. The relatively poor results of EBR method can be explained by the need of reliable edges to compute the regions. However, is it surprising to see that scale-invariant detectors outperform their corresponding affine invariant pairs in terms of repeatability and frequency. Even in the presence of large viewpoint changes, the results of Laplacian-based detector are higher. We explain this by the type of structure detected in the viewpoint change images. Indeed, Harris and Hessian affine detectors obtain good results when an elliptical region can easily be found in the neighborhood of the point. For cross-like structures, which is the most common pattern detected in our images, affine invariant detectors do not converge to a stable shape during normalization.

By looking at the number of occurrences, we observe that MSERs have a serious drawback. Only a very low number of features is detected. In contrast, Hessian detectors generally offer the highest number of points.

The examination of these results also shows that Laplacian-based feature detectors outperform their corresponding affine invariant pairs in terms of repeatability and frequency. At first sight, one can thus conclude that Laplacian-based detectors should be preferred. However, it is important to note that these comparisons do not take into account the information content of the detected regions. Affine invariant

detectors also provide a more precise shape on the point neighborhood which can be very useful during the characterization.

A conclusion from our experiments, that is consistent with other comparisons, is that the detectors based on the Hessian outperform the Harris detector in all cases.

### 4.2.3 Discussion

The last years developments in feature detection has coincided with the emergence of several new affine invariant techniques. Although very promising at first sight, they are less repeatable than classical multi-scale detectors in most situations. In particular conditions, such as severe affine deformations, they have been shown to outperform other methods. However, these results were not confirmed in our experiments (probably because of the nature of the images).

We also observed that performance offered by feature detectors largely depends on the conditions in which they are applied. Therefore, we consider, as other researchers [OFPA04, OP05], that the choice of a particular detector is not an efficient strategy; it cannot be done without a loss of performance in some conditions. Each detector covers a particular aspect of the image. Some of them are really good for textured scenes (Hessian) while other are more efficient on uniform regions (MSER). Moreover, the type of images that may occur in object recognition tasks is a priori unknown. Ideally, a system should be able to function with any kind of image. The choice of particular detector will critically eliminate a large part of the visual information available in the image. Therefore, a reasonable choice is to use different detectors to obtain better results by combining their specific properties.

## 4.3 Evaluation of Feature Descriptors

In this section, we evaluate the performance of nine feature descriptors; Steerable Filters [FA91], Differential Invariants [KvD87], Complex Filters [SZ02], Shape Context [BMP00], Spin Images [SLP03], Moment Invariants [FS93], and finally SIFT descriptor [Low99] and its extensions PCA-SIFT [KS04] and GLOH [MS05].

### 4.3.1 Experimental Protocol

The image set (Figure 4.1) used in these experiments is the same that has been used to evaluate local feature detectors in the preceding section. Similarly, the first

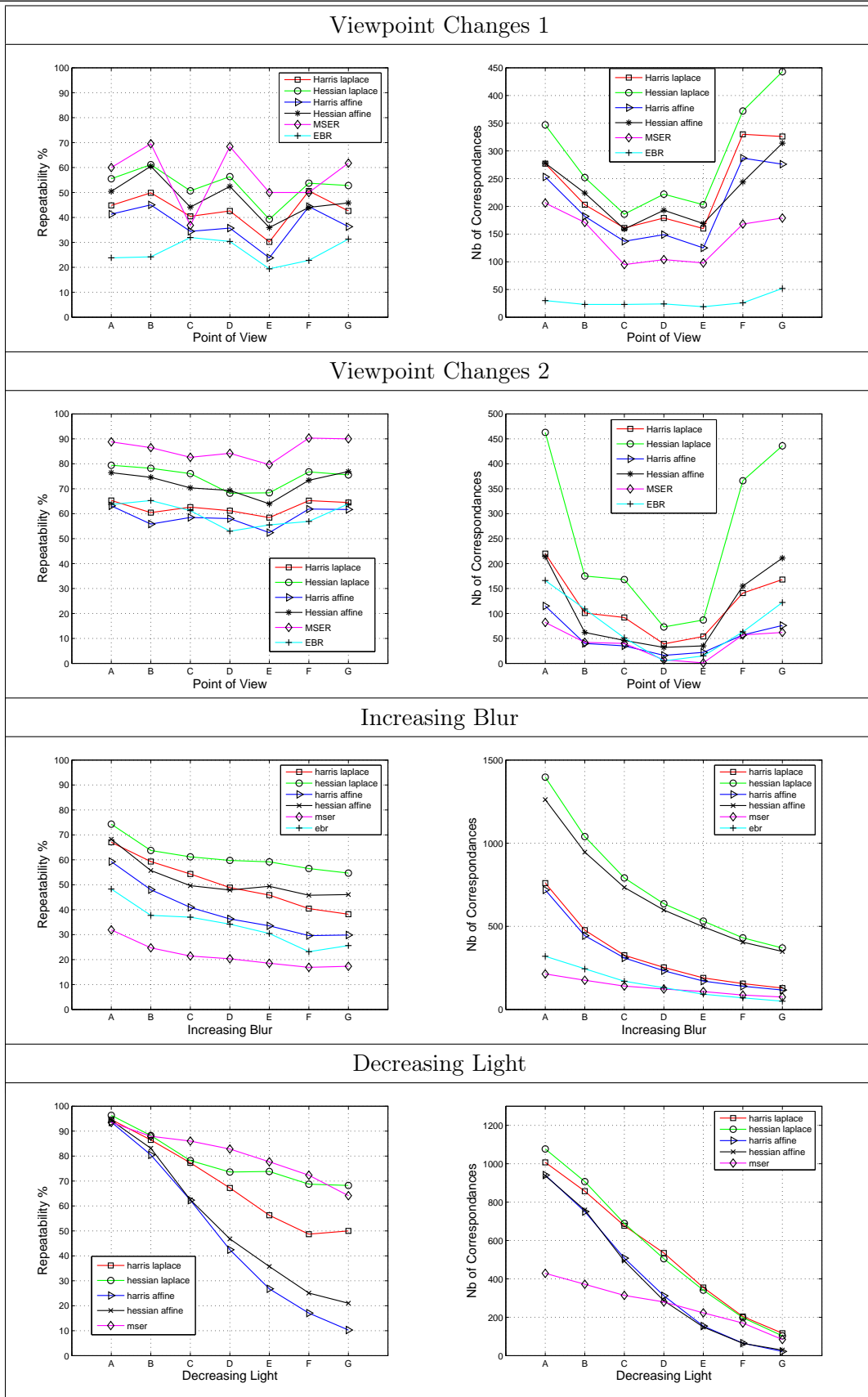


Figure 4.2: Performance evaluation of various type of detectors.

image of each set is considered as the reference image and four different changes in imaging conditions are examined: viewpoint and scale changes (a,b), image blur (c), and illumination change (d).

The evaluation criterion is similar to that used in other state-of-the-art descriptor evaluations [KS04, MS05]. It is based on a measure that is proportional to the ratio between the number of correct and false matches obtained for each image pair. First, the iterative *Harris-Laplace* detector [Mik02] is applied to each image to produce multi-scale feature points. Each of them is represented by a circular region that is used to compute the different type of descriptors.

Once a set of descriptors has been computed for a given image pair, each descriptor from the reference image is compared with each descriptor from the transformed image and the number of correct matches  $TP$  as well as the number of false matches  $FP$  are quantified. Two regions  $A$  and  $B$  are matched if the Euclidean distance between their descriptors  $D_A$  and  $D_B$  is below a given threshold  $t$ .

The matching score  $\mathcal{M}$ , or recall, that is used to construct our left plots in Figure 4.3 is defined as the number of correctly matched regions  $TP$  with respect to the total number  $TC$  of corresponding regions between two images of the same scene (computed with the *overlap error*):

$$\mathcal{M} = TP/TC \quad \text{for } \mathcal{P} \in [0.4, 0.5[ \quad (4.2)$$

We compute the matching score  $\mathcal{M}$  for a predefined precision  $\mathcal{P}$  interval;  $\mathcal{P} \in [0.4, 0.5[$ . This is represented as the ratio between the number of false matches  $FP$  and the total number of matches  $TP + FP$ :

$$\mathcal{P} = FP/(TP + FP) \quad (4.3)$$

The value of threshold  $t$  used to compute the matches was automatically adjusted to obtain the matching score  $\mathcal{M}$  for a fixed precision interval  $\mathcal{P}$ . The matching measure  $\mathcal{M}$  involved in these experiments can be considered as a good indicator for both invariance and selectivity and is therefore useful to evaluate the performance of the descriptors.

### 4.3.2 Results

In most images, SIFT and its GLOH extension generally perform best. Shape context is a little below but offers a descriptor of lower dimensionality (36 values) and seems to be reliable in the tested conditions.

On the contrary, Differential Invariants can be observed to give poor results in most cases. This can partially be explained by the very low dimensionality of the descriptor (only 8 values). It can also be noticed that Spin Images are the most sensitive to blur and illumination changes. It is not surprising since this descriptor uses raw pixels values.

Finally, Steerable Filters can be considered as a good compromise considering the low-dimensionality (14) in comparison with SIFT like descriptors (128).

### 4.3.3 Discussion

In this section, we have evaluated several description techniques on an image matching task. High-dimensional descriptors based on sparse gradient histograms, like *SIFT*, usually offer better performances than the low-dimensional descriptors like Steerable filters or differential invariants.

In the context of object recognition, the system architecture should be independent of the description method. In contrast with local detection methods that may have a deep effect on the results by eliminating evidence in the image, the description method does generally not (and should not) have a determinant impact on the system performance.

Sec. 4.3. Evaluation of Feature Descriptors

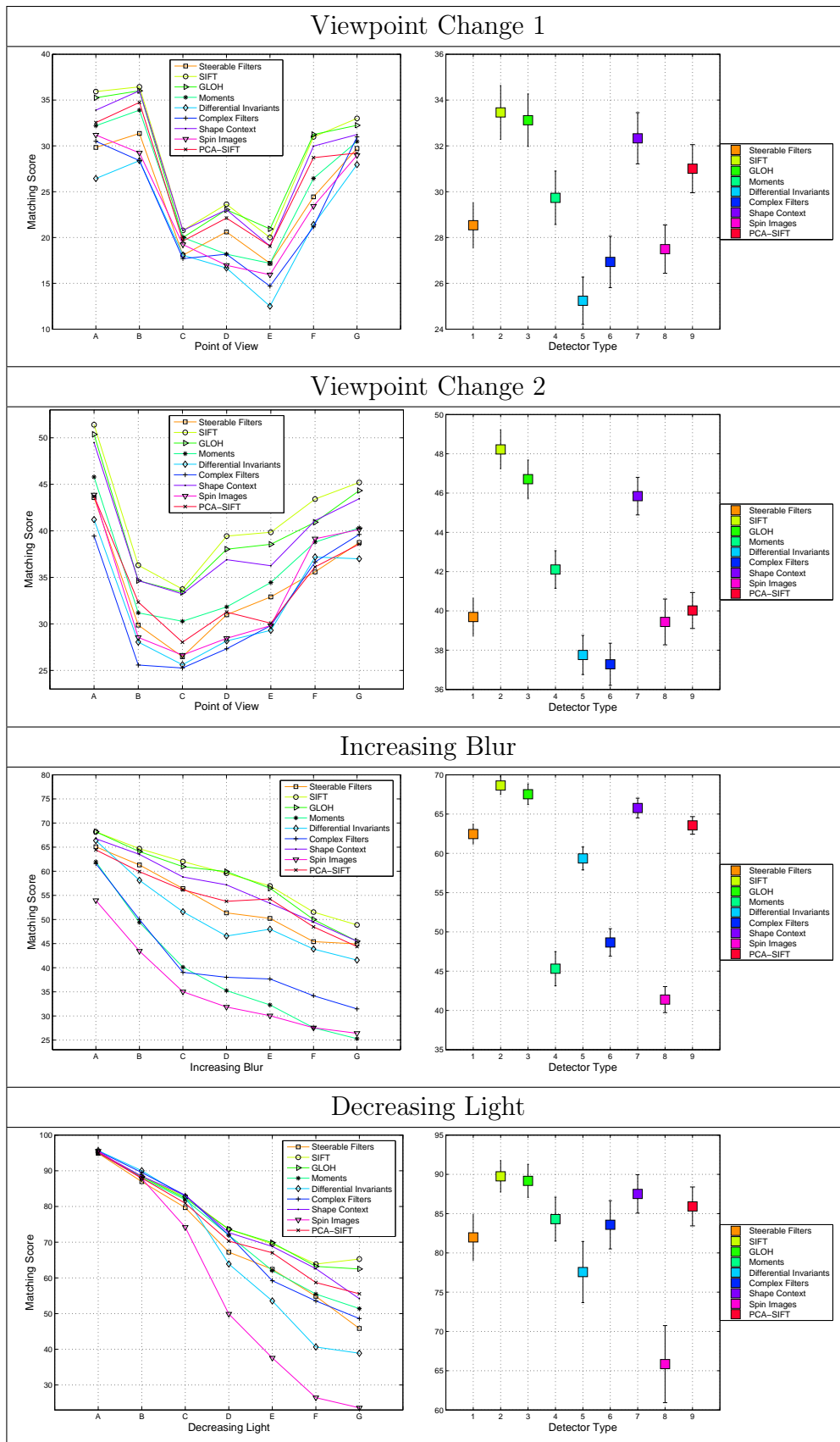


Figure 4.3: Performance evaluation of various type of descriptors.

# The Visual Feature Hierarchy

---

In this chapter, we introduce a new model to tackle the problem of object recognition. This model takes the form of a hierarchy of visual feature classes. Specifically, it starts at the first level from basic visual features that are easily extracted from standard feature detectors. Then these primitive visual features are combined using geometrical and appearance properties of their composition. This structure is repeated recursively at each level of the hierarchy to finally lead to high-level abstractions. This fills the gap between low-level and higher level visual concepts. Another impact of this representation strategy is that the model is capable of naturally integrating bottom-up and top-down interaction mechanisms within the same framework. Moreover it is coherent with recent advances in the field of neuroscience where it is now widely accepted that the recognition process in the visual cortex follows a hierarchical path of increasingly complex cells [RP99].

After presenting a preliminary description of the problem of finding appropriate representations of objects in Section 5.1, one of the main contributions of this thesis is described in Section 5.2, namely a new hierarchical representation of visual features. Then in Section 5.3, we explain how this representation can be integrated in a graphical model formalism. This probabilistic formulation allows to pose detection as an inference process. In Section 5.4, it is explained how Nonparametric Belief Propagation (NBP) can be used for inferring precise locations and pose of high-level features. Finally, we conclude this chapter by giving a critical analysis of the proposed representational framework.

## 5.1 The Search For Representation

Representation is, with detection and learning, one of the three essential steps to achieve object recognition [BP96]. In general, rather than being learned by the computer, the structure of object representation is designed manually to fulfill a specific task. This leads to a lack of generality. For instance, in the Constellation Model [FPZ03], the number of object parts has to be defined by the user. Moreover the representational power of many recognition systems is often a central point and the main cause of weaknesses. Indeed, no matter how good its perceptions are and how well it is able to learn robustly, a recognition system will always be limited if it cannot represent information properly.

However, the evaluation of a representation and its genericity is problematic because it relies on the conditions in which the tasks are performed and on a variety of qualitative criteria. In this section, we emphasize a few properties that a generic object representation should be able to provide. The understanding of the underlying issues is a preliminary requirement to the design of a new framework.

Note that object representation can be addressed through many overlapping disciplines such as ontologies, artificial intelligence (AI), or neurocognitive systems. Each of them focuses on specific aspects to model the real world (which is infinite). These have some connections to this work, however, we deliberately avoid to consider them to keep this chapter in a reasonable length.

### 5.1.1 An Ideal Object Representation

In the context of object recognition, we believe that a representation should satisfy the following intuitive definition:

**Definition 5.1.** A *Representation* is a formal scheme that makes explicit certain entities of information that may or may not be directly accessible to the perception process. These entities are stored in an invariant way and can be retrieved efficiently. A representation reduces the perceptual space to a more compact organization of these entities of which relations are defined together with their semantics.

This general definition can be refined in more specific terms. MARR [Mar82] proposed five criteria for evaluating object representations:

1. *Accessibility*: needed information should be directly available from the model rather than derivable through heavy computation,



2. *Scope*: a wide range of objects should be representable,
3. *Uniqueness*: an object should have a unique representation,
4. *Stability*: small variations in an object should not cause large variations in the model and
5. *Sensitivity*: detailed features should be represented as needed.

A few years later, FISHER [Fis89] completed these criteria by the *conceptual economy* criterion. First, this states that there should be only a single representation of any particular shape (multiple instances of that shape should refer to the single representation). Second, features that are distinctly characterized as a whole, irrespective of their substructures, should be represented simply by reference to that whole, with the details of that feature represented elsewhere.

Many other researchers have emphasized the need of a suitable representation, but only a few of them have proposed ideas of solutions. Among them, KESELMAN AND DICKINSON [KD05] recently mentioned

“*To make real progress on the problem of generic object recognition, we must address the representational gap [...] Not only must we continue to push the technologies of segmentation and perceptual grouping, we must be able to generate image abstractions that may not exist explicitly in the image, but which capture the salient, invariant shape properties of a generic model.*”

The representation and learning of new *image abstractions that may not exist explicitly in the image* is a major objective of the framework that is presented in this thesis. The development of such a capability will offer more selectivity in the representation by making possible the integration of high-level appearance models.

In a recent talk, TARR [Tar06] has mentioned a few lines of research in object recognition that can be considered as promising. Among them we can find the challenging task of *integrating feature-part hierarchies and their spatial relations into unitary models*. The definition and the learning of a feature hierarchy are the main purposes of this thesis. The first step toward such a model is presented in the following section.

## 5.2 A Visual Feature Hierarchy

In this section, we introduce a new object model that essentially combines several key concepts that have been developed the last couple of years in computer vision, machine learning, and computational neuroscience; spatial relations between local visual features [Sch96, Pia01], graphical models [Pea88, PFZ03], and hierarchies of complex cells [FMI83, RP99]. This results in a compositional hierarchy of visual feature classes.

In this topology, each feature class is represented as an entity and is related via spatial relations to some other feature classes from a higher and/or lower level of abstraction (see Figure 5.1). The purpose of this is to provide a coherent and generic object model by representing both local and global aspects through the combination of shape and appearance modalities.

The model is best explained by first considering the visual features classes represented in the hierarchy (Section 5.2.1). Then the pairwise relations are described in terms of geometrical relations between features (Section 5.2.2). This hierarchy is naturally integrated in a graphical model formalism (Section 5.3) that allows efficient inference mechanisms for detecting higher level features (Section 5.4).

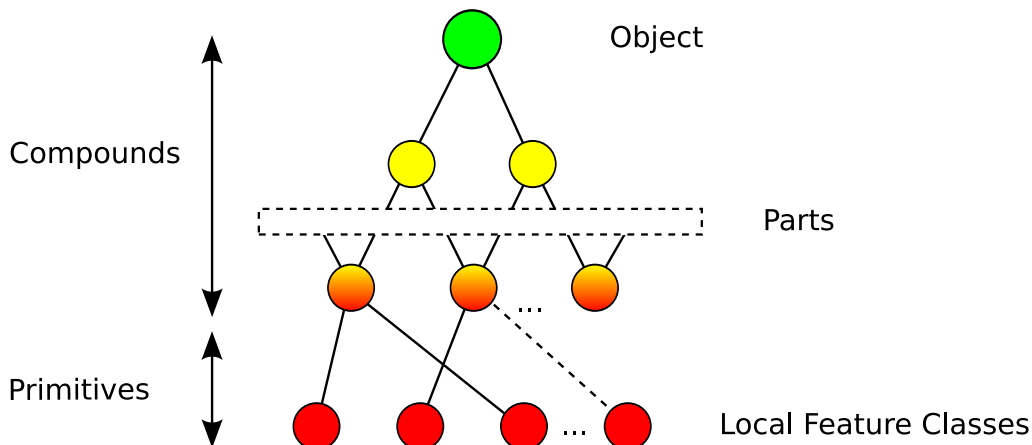


Figure 5.1: The overall structure of our hierarchical object model. Visual feature classes are represented by nodes and edges are used to define their relative positions. We distinguish between primitives (in red) that correspond to low-level features and compound features that are constrained by spatial relations with lower-level features.

### 5.2.1 Hierarchical Feature Set

In the literature, object models are often defined as a set of local visual features. The main drawback of feature-based approaches comes from the artificial separation between the object and its parts. Indeed, complex appearance variations occurring on these parts often forces the system to set a pre-defined granularity level and therefore to ignore a good deal of the available image information.

If we turn to the nature of objects, we see that many real world object categories exhibits a hierarchy in the structure of their parts. For instance, a face contains two eyes, each eye contains a eye globe, each eye globe contains an iris, . . . Hierarchical models aim at representing this naturally-occurring structure of information. Interestingly, a hierarchy often allows to obtain more flexibility in both shape and appearance, and often reduces the complexity of the learning process.

The proposed object model  $\mathcal{M}$  defines spatial relations  $\mathcal{S}$  between a set of feature classes  $\mathcal{F}$  using a hierarchical topology.

A feature class is said to be a

- *subfeature*  $f_s$  of  $f$ , if there exists a spatial relation  $\{f, f_s\} \in \mathcal{S}$  with feature  $f$  and if feature  $f_s$  lies at a lower level than feature  $f$  in the model  $\mathcal{M}$ ,
- *parent feature*  $f_p$  of  $f$  if there exists a spatial relation  $\{f, f_p\} \in \mathcal{S}$  with feature  $f$  and if feature  $f_p$  lies at a higher level in the model  $\mathcal{M}$ .

Any feature class  $f \in \mathcal{F}$  in the model  $\mathcal{M}$  is related to some subfeatures and/or parent features  $f'$  through spatial relations  $\{f, f'\} \in \mathcal{S}$ , such that the levels of  $f$  and  $f'$  in the model is different for any spatial relation  $\{f, f'\} \in \mathcal{S}$ .

By looking deeper into the structure of our topology (Figure 5.1), it is natural to distinguish between two types of features classes; those that lie at the first level of the hierarchy, namely the *primitives*, and the others; the *compound features*.

- *Primitive features* have no children and lie at the first level of the hierarchy. They are abstractions corresponding to low-level visual features. Typically, these classes originate from local visual feature detectors.
- *Compound features* have at least one subfeature (*i.e.* child). They consist of flexible spatial combinations of other subfeatures and tend to represent more global aspects.

Therefore, an object is simply considered as a particular kind of compound feature. The difference lies in the semantic level of interpretation.

## Appearance Model

Besides its structural information, any feature class  $f \in \mathcal{F}$  can be associated to an appearance model  $\mathcal{A}_f$ <sup>1</sup>. There are many ways of expressing such a model. Interestingly, our representation is general enough to be independent from the chosen technique. A common solution is to associate to each class  $f \in \mathcal{F}$  a mean appearance vector  $\mu_f^A$  of  $n$  elements in  $\mathbb{R}$  and corresponding covariance matrix  $\Sigma_f^A$ . We adopt this technique and assume that appearance vectors follow an unimodal Gaussian distribution:

$$\mathcal{A}_f = (\mu_f^A, \Sigma_f^A) \text{ where } \begin{aligned} \mu_f^A &\in \mathbb{R} \times \{1 \dots n\} \\ \Sigma_f^A &\in \mathbb{R} \times \{1 \dots n\} \end{aligned} \quad (5.1)$$

The features considered here are intrinsically linked to a given characteristic appearance. A feature class should represent any relevant aspect of the object and can possibly be non-visual and non-geometric. Because the nature of the features in a recognition system depends on the task, it can be useful for a wide range of applications to include non-visual features that can be obtained from other sensors.

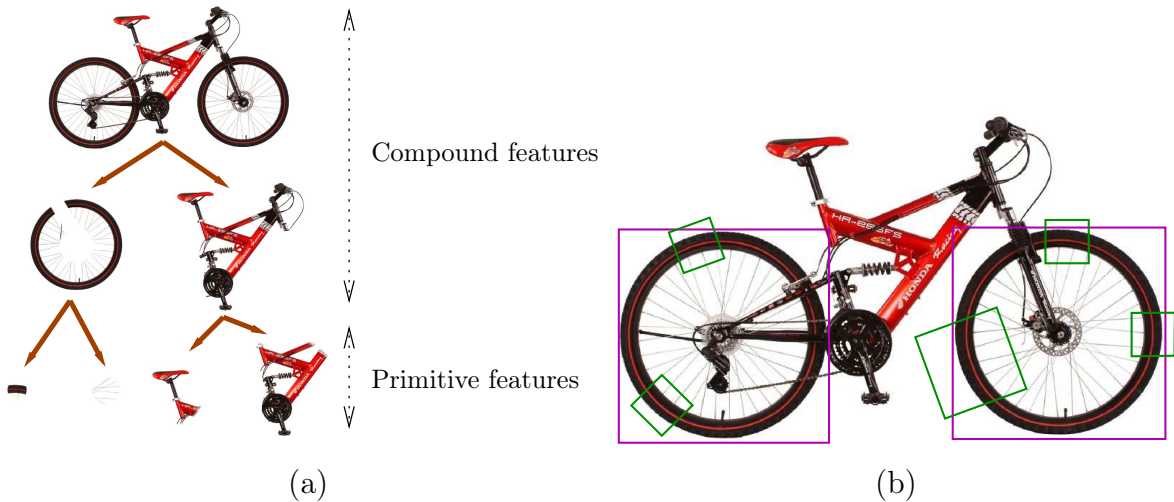


Figure 5.2: An illustrative example of feature hierarchy (a). Instances of two generic feature classes in the bike scene (b); Piece-of-tire primitives (green) and two wheel compounds (purple) are detected in the image. Reproduced with permission from [DP07].

<sup>1</sup>This information is not compulsory for a compound feature since it is always possible to predict its position from its subfeatures.

**Instantiating a feature class**

In general, the instantiation of a feature class results from a detection process on a given image. As shown in Figure 5.2, a feature may appear several times in the image and may have different orientations and scales.

Our model reflects these properties by associating three parameters  $\{x, \vartheta, w\}$  to each *instance* of a visual feature class  $f \in \mathcal{F}$ :

$$\forall f \in \mathcal{F}, \text{inst}(f, I) \rightarrow \{x, \vartheta, w\}_{0\dots k} \quad (5.2)$$

where  $\text{inst}(f, I)$  is a detector that produces  $k$  instances of a feature class  $f$  on image  $I$ ,  $x \in \mathbb{R}^2$  is the position in the image coordinates,  $w \in \mathbb{R}$  the weight that represents the likelihood of this specific instance and  $\vartheta \in \mathbb{R}^3 \times [0, 2\pi[$  is the local transform parameter that represents the pose of the feature and is defined as

$$\vartheta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \quad (5.3)$$

where  $\theta, a, s_x, s_y$  are respectively the orientation, skew and scale parameters.

In practice, we observed that affine invariant detectors are not reliable enough to obtain stable affine parameters and to keep track of them during the inference process. Therefore, the parameter  $\vartheta$  that is often simplified to include only rotation  $\theta$  and scale  $s$  parameters:

$$\vartheta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \quad (5.4)$$

### 5.2.2 Flexible Spatial Relations

Our model represents the shape of an object in terms of relative positions between visual feature classes. It is a sparse model in the sense that the entire object is represented through local pairwise relations between features. An advantage of this is that it helps the system to keep a tractable complexity.

In order to represent a spatial relation  $\{i, j\} \in \mathcal{S}$  between two feature classes, the object model  $\mathcal{M}$  associates two models of relative position  $\{s_{i \rightarrow j}, s_{j \rightarrow i}\}$ ; one relative to each class. These models aim at informing the system where a feature is expected to be found with respect to the reference feature (*i.e.* source). In the ideal case, these relations should be able to represent stable spatial dependencies in a flexible way (by offering invariance to scale, orientation). Figure 5.3 illustrates spatial relations in an artificial example. In our model, they occur only between features of subsequent levels in the hierarchy. No explicit relations are defined directly between features of the same level.

The relative position model can be defined either parametrically through a Gaussian mixture or nonparametrically as a set of particles. In the following, we present these two options and particularly focus on the three different parametric formulations of the spatial relations.

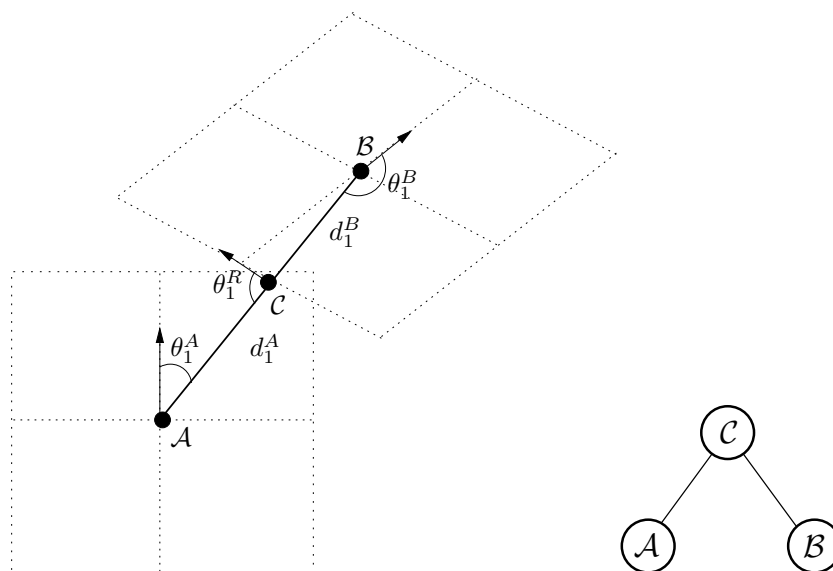


Figure 5.3: Spatial relations between a compound feature  $\mathcal{C}$  and two subfeatures ( $\mathcal{A}$ ,  $\mathcal{B}$ ).  $\{\theta_1^A, d_1^A\}$  and  $\{\theta_1^B, d_1^B\}$  represent the direction and the distance to  $\mathcal{C}$  respectively from  $\mathcal{A}$  and  $\mathcal{B}$ .  $\theta_1^R$  is the relative orientation of  $\mathcal{C}$  computed from  $\mathcal{A}$ .

### Parametric Relations

Representing spatial dependencies between features through parametric relations is not new. Different approaches [WPW00, PG00b, FPZ03, VS04] have successfully exploited them to perform object recognition. However, representing flexible spatial dependencies in hierarchical models is a new and challenging problem that we propose to tackle in this section by presenting pairwise spatial relations.

In its parametric form, our model approximates geometrical relations  $s_{i \rightarrow j}$  between two feature classes  $i, j$  by a Gaussian mixture of  $k$  components, each representing a likely relative position  $\mu_k \in \mathbb{R}^n$  of one of the two features  $f_j$  with respect to the other, the reference feature  $f_i$ :

$$s_{i \rightarrow j}(x; \Theta) = \sum_{k=1}^K w_k \mathcal{G}(x; (\mu_k, \Sigma_k)) \quad (5.5)$$

where  $\Theta = (w_{1 \dots K}; \mu_{1 \dots K}; \Sigma_{1 \dots K})$  defines the model parameters, and  $x \in \mathbb{R}^2$  is the relative image space normalized with respect to the reference feature  $i$ . Mixing weights  $w_{1 \dots K}$  must be positive and sum to unity.

Depending on the dimensionality  $n$  of relative position parameters  $\mu_{1 \dots K} \in \mathbb{R}^n$ , it is possible to distinguish between different types of spatial relations:

1. Typically, relative positions  $\mu_{1 \dots K}$  between features are defined in two-dimensional planar coordinates  $\mathbb{R}^2$ . This space is normalized either with the local affine deformation  $\vartheta_i$  or at least normalized with the orientation  $\theta_i$  and scale  $s_i$  of the instance of the reference feature  $i$ . An illustrative example of such a spatial relation is given in Figure 5.4. It shows a model of two components representing two possible positions in the affine normalized neighborhood.
2. Another possibility is to define the relation  $s_{i \rightarrow j}$  in terms of relative distance between features. Such spatial relation is defined by a one-dimensional relative distance  $\mu_{1 \dots K} \in \mathbb{R}$  and covariance in the affine-normalized neighborhood of the reference feature class (see Figure 5.5).
3. Finally, this representation can also accommodate relations that are defined on more than two dimensions  $\mathbb{R}^n, n > 2$ . Similarly to the two-dimensional case, a straightforward solution is to consider a relative position of the normalized neighborhood. This can be particularly useful to model spatial relations for a three-dimensional object.

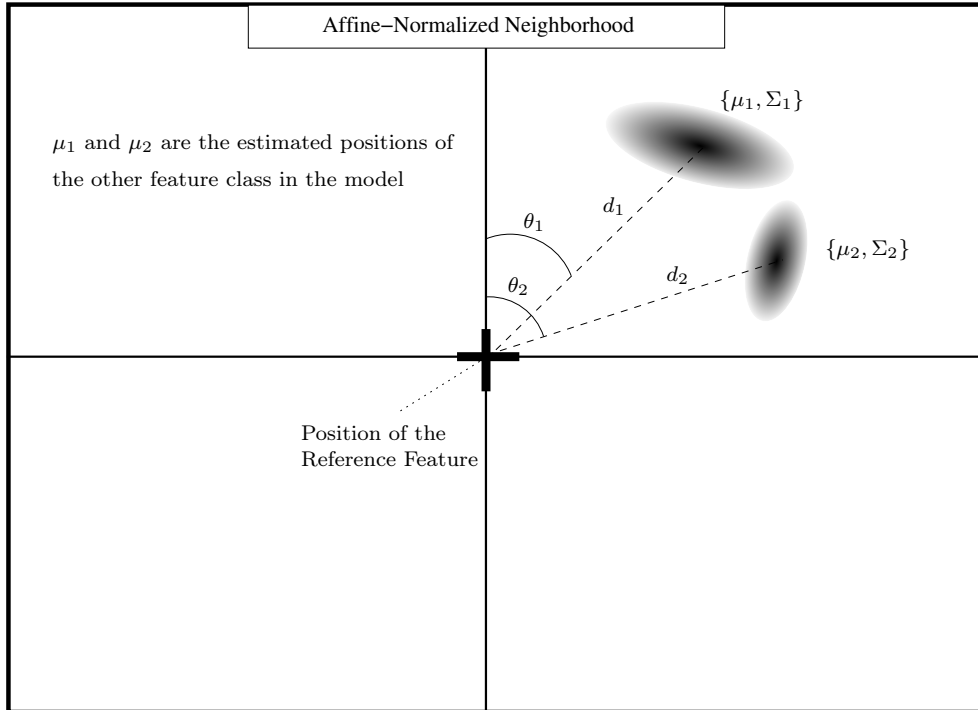


Figure 5.4: Two-dimensional parametric spatial relation between two features.

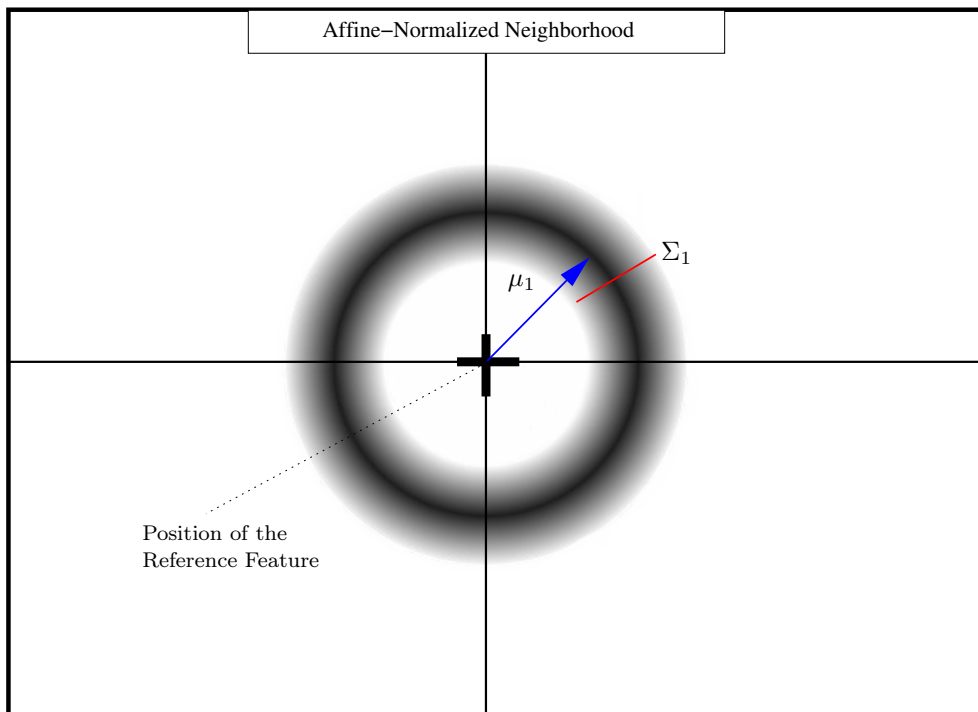


Figure 5.5: One-dimensional parametric spatial relation between two features.



### Properties

As previously shown in Equation 5.5, a spatial relation may contain several components  $\mu_{1\dots k}$ , each depicting a relative position from one feature class to the other. Given the position of one feature in the image, called instance, it is possible to predict the position of an instance of the other feature that is linked through the spatial relation.

An interesting property of our hierarchical structure is that the position of a compound feature  $\mathcal{C}$  in the image can be obtained by its children  $\mathcal{A}, \mathcal{B}$ , no matter which one is used to predict it (Figure 5.3).

More precisely,

Let  $\{x_A, y_A\}, \{x_B, y_B\}$  be the position of an instance of feature class  $A$  and  $B$ ,

Let  $\mu_e^A \in s_{A \rightarrow C}$  and  $\mu_e^B \in s_{B \rightarrow C}$ , be two relative position components obtained from their respective spatial relation model (Figure 5.4),

Let  $\{x_{A \rightarrow C}, y_{A \rightarrow C}\}, \{x_{B \rightarrow C}, y_{B \rightarrow C}\}$  be the positions of instance  $\mathcal{C}$  predicted from the instances  $\{x_A, y_A\}, \{x_B, y_B\}$ .

Specifically, the position  $\{x_{A \rightarrow C}, y_{A \rightarrow C}\}$  of the feature  $\mathcal{C}$  obtained from  $\mathcal{A}$  is obtained by applying the local pose  $\vartheta_A^{inst}$  of the instance on the relative position  $\mu_e^A$  in the model  $s_{A \rightarrow C}$  and then by translating the resulting point by the position  $\{x_A, y_A\}$  of the feature  $A$  in the image coordinates:

$$\begin{aligned}
 \begin{bmatrix} x_{A \rightarrow C} \\ y_{A \rightarrow C} \end{bmatrix} &= \begin{bmatrix} \mu_e^A(0) \\ \mu_e^A(1) \end{bmatrix} \times \vartheta_A^{inst} + \begin{bmatrix} x_A \\ y_A \end{bmatrix} & (5.6) \\
 &= \underbrace{\begin{bmatrix} \mu_e^A(0) \\ \mu_e^A(1) \end{bmatrix}}_{\text{Relative position}} \times \underbrace{\begin{bmatrix} \cos(\theta_A) & -\sin(\theta_A) \\ \sin(\theta_A) & \cos(\theta_A) \end{bmatrix}}_{\text{Pose of A}} \underbrace{\begin{bmatrix} s_A^x & 0 \\ 0 & s_A^y \end{bmatrix}}_{\text{Pose of A}} \underbrace{\begin{bmatrix} 1 & a_A \\ 0 & 1 \end{bmatrix}}_{\text{Position of A}} + \underbrace{\begin{bmatrix} x_A \\ y_A \end{bmatrix}}_{\text{Position of A}}
 \end{aligned}$$

The derivations for computing the position  $\{x_{B \rightarrow C}, y_{B \rightarrow C}\}$  from  $\mathcal{B}$  follow similarly:

$$\begin{aligned}
 \begin{bmatrix} x_{B \rightarrow C} \\ y_{B \rightarrow C} \end{bmatrix} &= \begin{bmatrix} \mu_e^B(0) \\ \mu_e^B(1) \end{bmatrix} \times \vartheta_B^{inst} + \begin{bmatrix} x_B \\ y_B \end{bmatrix} & (5.7) \\
 &= \begin{bmatrix} \mu_e^B(0) \\ \mu_e^B(1) \end{bmatrix} \begin{bmatrix} \cos(\theta_B) & -\sin(\theta_B) \\ \sin(\theta_B) & \cos(\theta_B) \end{bmatrix} \begin{bmatrix} s_B^x & 0 \\ 0 & s_B^y \end{bmatrix} \begin{bmatrix} 1 & a_B \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} x_B \\ y_B \end{bmatrix}
 \end{aligned}$$

Then the position  $\{x_C, y_C\}$  of the instance of  $\mathcal{C}$  derived from instance  $\mathcal{A}$  is equal to the position of  $\mathcal{C}$  derived from instance  $\mathcal{B}$ :

$$\begin{bmatrix} x_C \\ y_C \end{bmatrix} = \begin{bmatrix} x_{\mathcal{A} \rightarrow \mathcal{C}} \\ y_{\mathcal{A} \rightarrow \mathcal{C}} \end{bmatrix} = \begin{bmatrix} x_{\mathcal{B} \rightarrow \mathcal{C}} \\ y_{\mathcal{B} \rightarrow \mathcal{C}} \end{bmatrix} \quad (5.8)$$

### Representing Relative Pose

Beside the *relative position* parameters  $\Theta = (w_k, \mu_k, \Sigma_k)_{\{k=1 \dots K\}}$  that are used to retrieve the position of the other feature, another piece of information that can be exploited to improve the spatial relation is the *relative pose* between two geometrically related features. Intuitively, this allows to compute the pose (or at least the orientation) of one feature  $\mathcal{C}$  given another feature  $\mathcal{A}$ .

To this purpose, our model associates a local affine measure  $\vartheta_{k=\{1 \dots K\}}^R$  to each component of the mixture  $S_{i \rightarrow j}$ . This pose is set relative to the pose of the reference feature  $i$  and is consequently invariant to any two-dimensional rotation and scale variation. The relative position  $\Theta$  is extended with the relative pose  $\vartheta^R$  as follows:

$$\Theta \leftarrow (\Theta_k \cup \vartheta_k^R)_{\{k=1 \dots K\}} \quad (5.9)$$

$$\Leftrightarrow \Theta \leftarrow (w_k, \mu_k, \Sigma_k, \theta_k^R, a_k^R, s_k^R)_{\{k=1 \dots N\}} \quad (5.10)$$

This will help the system to keep track of feature pose at every level of the hierarchy. Figure 5.3 shows the relative orientation  $\theta_k^R$  between features  $\mathcal{A}$  and  $\mathcal{C}$ .

### Nonparametric Relations

Rather than using a Gaussian Mixture to model spatial relations, it is possible to use a nonparametric distribution. This is particularly suitable when the number of training samples is too small or when the distribution cannot be covered efficiently by parametric methods. One main advantage of this type of method is its ability to obtain the relations directly from experimental data. However, the complexity during inference may quickly become intractable.

Nonparametric relations are defined as sets of  $P$  particles where each particle is defined by a relative location  $\mu \in \mathbb{R}^n$ , a weight  $w$ , a variance  $\Sigma$ , and a relative orientation  $\theta^R$  of the feature from the reference feature:

$$s_{i \rightarrow j} = \{\mu_l, w_l, \Sigma_l, \theta_l^R\}_{\{l=1 \dots P\}} \quad (5.11)$$

## 5.3 Representing a Hierarchy via a Graphical Model

In this section, we describe how the proposed hierarchy can be represented in a graphical model (see Figure 5.6). As it has been discussed in Chapter 2, graphical models generally provide a convenient formalism to represent complex systems and to exploit efficient inference mechanisms. Pairwise Markov Random Fields (PMRF's) appear to be adapted for representing objects. They have the advantage to offer low complexity and are able to deal with loopy graphs.

Another reason to use PMRF's is that the correspondence between our hierarchical model and its representation in a PMRF is relatively straightforward. The main idea is to represent feature classes by nodes and spatial relations by statistical relations between nodes, represented by edges. Therefore, whereas our model  $\mathcal{M} = \{\mathcal{F}, \mathcal{S}\}$  is constituted of visual classes  $\mathcal{F}$  and spatial relations  $\mathcal{S}$ , the graphical model  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  uses nodes  $\mathcal{V}$  and edges  $\mathcal{E}$  to represent the object structure. However, there are different types of nodes in the graph and different ways to express geometric dependencies in edges. These topics are addressed in the following sections.

### 5.3.1 Visual Feature Classes as Nodes

In PMRF terminology, it is common to differentiate between *hidden* and *observable* nodes (*i.e.* random variables). We propose to use a pair of these nodes, hidden  $x_i \in x$  and observable  $y_i \in y$ , to represent two aspects of a visual feature class  $f_i \in \mathcal{F}$ :

$$\forall f_i \in \mathcal{F}, f_i \rightarrow \{x_i, y_i\} \quad (5.12)$$

Intuitively, observable nodes  $y_i$  represent the output of local feature detectors following some basic measurements directly done on the image. Hidden nodes  $x_i$  represent a spatial density about the location and pose of the feature on the given image. A difference between these two types of node resides in the fact that observations  $y_i$  are only obtained from appearance measures and are therefore sensitive to occlusions and detector weaknesses whereas hidden node annotations are estimated through an inference process that exploits all the observations  $y$  in the graph  $\mathcal{G}$  together with the shape model consistency.

Some features  $f_i$  may not have direct observations  $y_i$  available. In that case, the feature will be represented by a hidden node  $x_i$  only,  $f_i \rightarrow \{x_i\}$ . The instantiation of hidden and observable nodes for a given image is explained below.

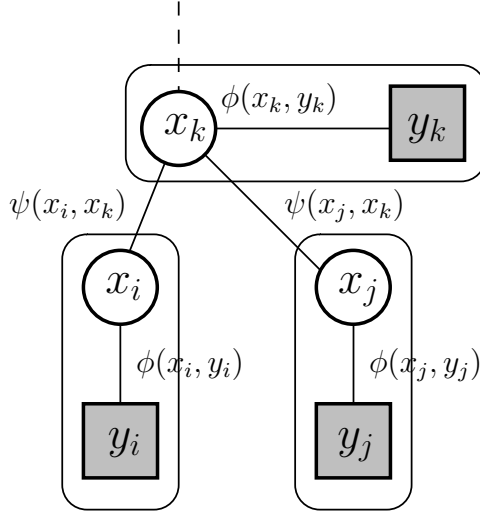


Figure 5.6: Illustration of the proposed representation. Each feature has an observable variable  $y_i$  and a hidden variable  $x_i$  linked through a local observation potential  $\phi(x_i, y_i)$ . Each pairwise potential  $\psi(x_i, x_j)$  encodes the spatial relation between two feature classes  $i, j$ .

### Hidden Nodes

Each visual feature has an associated hidden random variable  $x_i$  that is depicted by a white node in the graph shown in Figure 5.6. This hidden variable  $x_i$  is a continuous density function defined on the image space  $\mathbb{R}^2$  (*i.e.* feature position) and the feature pose  $\mathbb{R}^3 \times [0, 2\pi[$ .

However, the direct instantiation of such a continuous space ( $\mathbb{R}^2 \times \mathbb{R}^3 \times [0, 2\pi[$ ) should be avoided to keep the problem tractable in terms of complexity. A solution is to use a nonparametric density estimate to approximate the distribution. However, using a high-dimensional circular space for the particles may also lead to various problems during inference. Instead, we represent each hidden variable in the image space coordinates. We keep track of the pose information of each sample separately. The nonparametric representation used to model the spatial density distribution of each hidden random variable  $x_i \in x$  is composed of  $n$  samples:

$$p = \{\mu, \Sigma, w\}_{j=1\dots n} \quad (5.13)$$

where  $\mu \in \mathbb{R}^2$  is the position in the image,  $w \in \mathbb{R}$  is the weight and  $\Sigma \in \mathbb{R}^2$  is the variance associated to this sample.

To each sample  $p_j$  is also associated a parameter  $\vartheta_j$  that corresponds to the pose of the feature (orientation  $\theta$ , scales  $s_x, s_y$ , and skew  $a$ ). We can summarize the

position of samples  $p$  and their corresponding pose  $\vartheta$  as parameter  $\Theta$ :

$$\Theta \leftarrow \{p_j, w_j, \vartheta_j\}_{j=1\dots n} \quad (5.14)$$

$$\Leftrightarrow \Theta \leftarrow \{\mu_j, \Sigma_j, w_j, \{\theta, a, s_x, s_y\}_j\}_{j=1\dots n} \quad (5.15)$$

### Observable Nodes

Contrary to many approaches that use pixel values directly as input, we allow the system to consider the image as a collection of features. These are particularly convenient to reduce the large visual input space to a smaller set of features.

Our observable nodes  $y_i \in y$  are annotated with a set of features obtained from feature detectors. Each observed feature (*i.e.* instance) is defined as a triple  $\{\mu, \vartheta, \mathcal{D}\}$  where  $\mu \in \mathbb{R}^2$  is a location in the image,  $\vartheta$  is the local affine deformation matrix, and  $\mathcal{D} \in \mathbb{R}^{N_d}$  is a local descriptor (*i.e.* a vector of real numbers) that summarizes the appearance around the neighborhood of the point  $\mu$ .

Another particularity of our approach is that an observable node  $y_i \in y$  can be associated to several feature detectors  $K_{j=1\dots n}$ . Therefore a given feature class can be located in the image by different detectors (thus improving the overall robustness of the system). Moreover, these detectors capture specific measurements and are sensitive to different image modalities. Multiple types are needed since no one type of feature can represent all types of object.

The union of the local feature locations obtained from each detector  $K_i$  associated to the current observable node  $y_i$  defines its annotation for a given image  $I$ :

$$\mathcal{O}_{y_i}(I) \leftarrow \bigcup_{K_i \in K_{y_i}} K_i(I) \quad (5.16)$$

$$\Leftrightarrow \mathcal{O}_{y_i}(I) \leftarrow \bigcup_{K_i \in K_{y_i}} \{\mu, \vartheta, \mathcal{D}\}_{k=1\dots n} \quad (5.17)$$

### 5.3.2 Spatial Relations as Edges

In Pairwise Markov Random Fields, two different types of edges can be found. The first type connects pairs of hidden nodes and is associated to a pairwise potential  $\psi(x_i, x_j)$  that is used to represent the relationship between the two nodes.

The second type of edges links pairs of hidden  $y_i$  and observable nodes  $x_i$  and corresponds to an observation potential  $\phi(x_i, y_i)$ . This kind of potential is used to incorporate the observations in the model during the inference process. In the following, we provide detailed information on pairwise and observation potentials.

### Pairwise potentials

Intuitively, pairwise potentials  $\psi(x_i, x_j)$  are used to represent the relationship existing between two hidden random variables. In this work, we do not represent the relationships through pairwise potentials explicitly but we rather use conditional distributions. To better understand the motivation of this choice, let us to go back to the description of the NBP algorithm (Section 2.3). During the computation of an outgoing message  $m_{ij}$ , we saw that when the marginal influence is constant, the outgoing message can be computed by using the conditional distribution  $\psi(x_j|x_i)$ . This means that in practice we do not need to represent pairwise potentials  $\psi(x_i, x_j)$  explicitly to perform NBP but we can instead use the conditional distributions  $\psi(x_i|x_j)$  and  $\psi(x_j|x_i)$  they define <sup>2</sup>.

Defining the relations under these terms facilitates the correspondence between our spatial relation  $\mathcal{S}_{i \rightarrow j}$  and the conditional functions  $\psi(x_j|x_i)$ . Indeed, the same representation in terms of Gaussian mixtures can be used. Because of their lower dimensionality, conditional functions are also easier to learn. During inference, the application of the potential can be thought of as a mapping where each sample  $\mu_i$  is moved to the  $N_{ij}$  directions of the Gaussian mixture:

$$\psi(x_j|x_i) = \sum_{k=1}^{N_{ij}} w_i^k \mathcal{G}(x_j; \gamma_{i,j,k}(\mu_i, \vartheta_i), \Sigma_i), \text{ where } \sum_{k=1}^{N_{ij}} w_i^k = 1 \quad (5.18)$$

where  $w_i^k$  is the relative weight of an individual component. Function  $\gamma_{i,j,k}$  is a mapping that computes the position of the samples for the  $k$ -th Gaussian component. Specifically, this conditional function moves the samples of  $x_i$ , denoted  $\mu_i \in x_i$ , using the  $k$  relative positions  $\mu_{ijk}$  of the model. This process is illustrated in Figure 5.7 and the function  $\gamma_{i,j,k}$  is formalized below:

$$\begin{aligned} \gamma_{i,j,k}(\mu_i, \vartheta_i) &= \mu_i + (\mu_{ijk} \vartheta_i) \\ \Leftrightarrow \gamma_{i,j,k}(\mu_i, \vartheta_i) &= \begin{bmatrix} \mu_i^x \\ \mu_i^y \end{bmatrix} + \begin{bmatrix} \mu_{ijk}^x \\ \mu_{ijk}^y \end{bmatrix} \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} s_i^x & 0 \\ 0 & s_i^y \end{bmatrix} \begin{bmatrix} 1 & a_i \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (5.19)$$

---

<sup>2</sup>A similar approach has been presented by ISARD [Isa03] and successfully used by SIGAL *et al.* [SISB03, SBR<sup>+</sup>04, SB06] for tracking purposes.

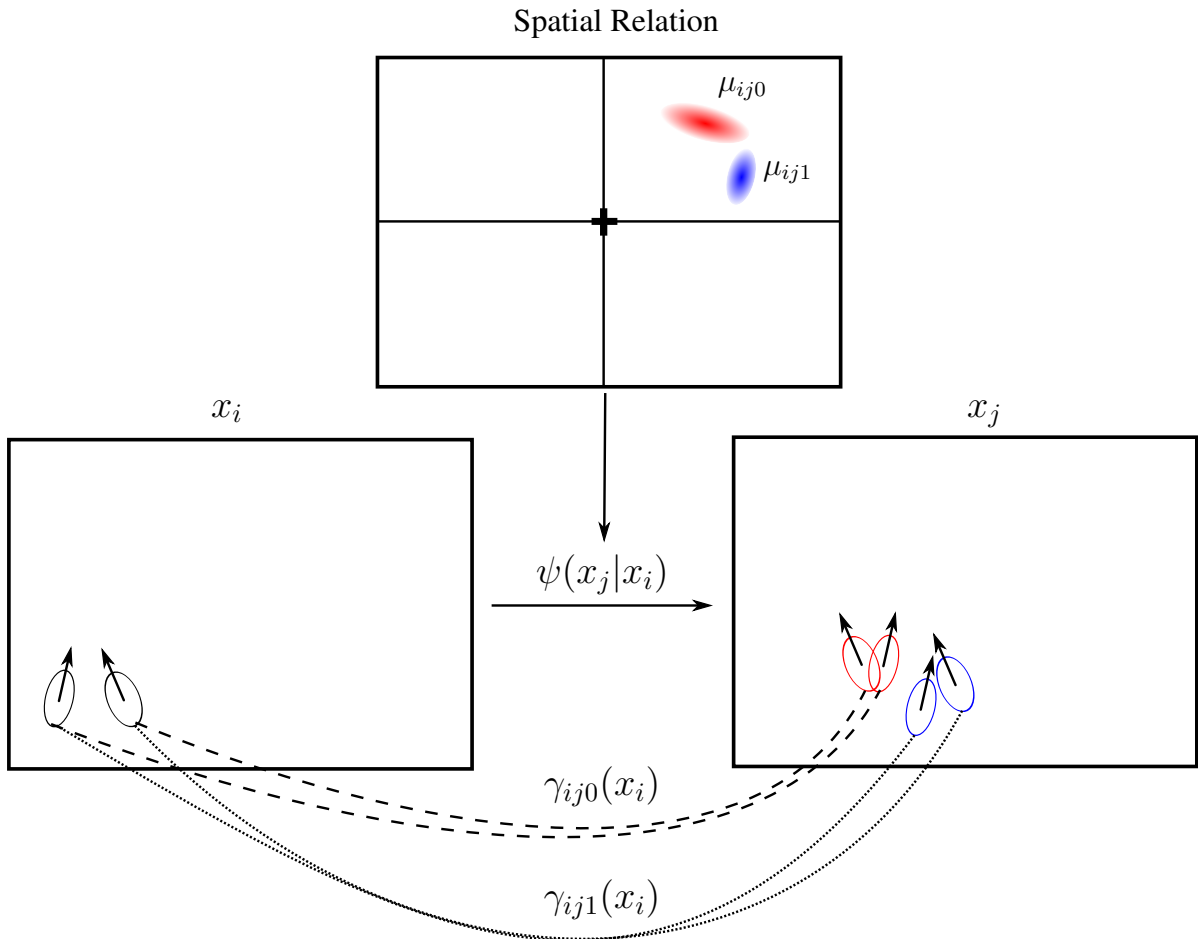


Figure 5.7: Visual interpretation of the mapping of a spatial relation. During the application of a conditional  $\psi(x_j|x_i)$ , each instance of the source feature  $x_i$  is mapped to the target space by using the different relative positions  $\mu_{ij0}, \mu_{ij1}$ .

### Observation potential

Observation potentials,  $\phi(x_i, y_i)$ , correspond to the likelihood parts in the standard Bayesian formulation of an inference problem. They represent the compatibility between a hidden random variable  $x_i$  and its corresponding image evidence  $y_i$ .

The observed features  $\mathcal{O}_{y_i}(I)$  of the image  $I$  may appear differently from the features of the unknown true scene due to a number of variation factors. This includes image noise, errors of feature extraction algorithms, and others artifacts. It is the purpose of the likelihood function (or observation potential)  $\phi(x_i, y_i)$  to describe these differences in probabilistic terms.

Given a set of observed features  $\mathcal{O}_{y_i}(I) = \{\alpha, \vartheta, \mathcal{D}\}_{k=1\dots n}$  at node  $y_i$ , the observation potential is formulated by creating a spatial Gaussian  $g_k$  at point  $\alpha_k$  weighted by a similarity measure  $w$  with the feature appearance model  $\mathcal{A}_i$ . The likelihood  $\mathcal{L}$  for a given point  $t$  in the image corresponds to the maximum response among all weighted Gaussians  $g_k$  at point  $t$ , that is

$$g_k = w \mathcal{G}(\alpha_k, \Sigma) \text{ where } w = e^{-\lambda(\mathcal{D}_k, \mathcal{A}_i)} \quad (5.20)$$

$$\mathcal{L}(t) = \operatorname{argmax}_k g_k(t) \quad (5.21)$$

where  $\lambda(\mathcal{D}_k, \mathcal{A}_i)$  is the *Mahalanobis distance* between an observation  $\mathcal{D}_k$  and the appearance model  $\mathcal{A}_i$ , and  $\Sigma$  is set proportional to the scale in  $\vartheta_k$ .

The *Mahalanobis distance* [Mah36] is a generalization of the Euclidean distance and is suitable to be used as the metric on the visual feature space. It differs from the Euclidean distance in that it takes into account the correlations of the data set and is not dependent on the scale of measurements.

Formally, the Mahalanobis distance from a group of values with mean  $\mu = (\mu_1, \mu_2, \dots, \mu_p)^T$  and covariance matrix  $\Sigma$  for a multivariate vector  $x = (x_1, x_2, \dots, x_p)^T$  is defined as:

$$D_M^2(\vec{x}, \vec{y}) = (x - y)^T \Sigma^{-1} (x - y) \quad (5.22)$$

If the covariance matrix  $\Sigma$  is diagonal, then the resulting distance measure is called the “normalized Euclidean distance”:

$$D_E^2(\vec{x}, \vec{y}) = \sum_{i=1}^p \frac{(x_i - y_i)^2}{\sigma_i^2} \quad (5.23)$$

where  $\sigma_i$  is the standard deviation of the  $x_i$  over the training set.



## 5.4 Inferring High-level Features

In our system, computing the presence of features in an image amounts to estimating the posterior belief associated with the hidden nodes given all observations. Thus, detection of hierarchical features can be posed as inference in the graphical model. One way to do this is to use Nonparametric Belief Propagation (NBP) [SIFW03]. However, a few extensions are necessary to apply NBP in our framework, essentially because we need to keep track of the orientation (pose) of samples during inference. After a brief review about NBP, our message update algorithm is presented.

As it has been discussed in Chapter 2, NBP is an inference algorithm for graphical models that generalizes particle filtering and propagates information via a series of local message-passing operations. Reasons to use NBP are multiple; it allows efficient inference in high-dimensional space, in presence of complex likelihoods or potentials and even on loopy graphs. It is also motivated by recent advances in neurosciences [LM03].

In NBP, a message  $m_{ij}$  from node  $i$  to  $j$  is written

$$m_{i,j}(x_j) \leftarrow \int_{\mathcal{X}_i} \psi_{i,j}(x_i, x_j) \phi_i(x_i, y_i) \prod_{k \in \mathcal{N}_i \setminus j} m_{k,i}(x_i) dx_i \quad (5.24)$$

where  $\mathcal{N}_i$  is the set of neighbors of node  $i$ ,  $\psi_{i,j}(x_i, x_j)$  is the pairwise potential between nodes  $i, j$ , and  $\phi_i(x_i, y_i)$  is the local observation potential. After any iteration, each node can compute an approximation  $\hat{p}(x_i|y)$  to the marginal distribution  $p(x_i|y)$  by combining the incoming messages with the local observation:

$$\hat{p}(x_i|y) \leftarrow \phi_i(x_i, y_i) \prod_{k \in \mathcal{N}_i} m_{k,i}(x_i) \quad (5.25)$$

A particularity of NBP is that it exploits a sampled-based representation to approximate messages and beliefs:

$$m_{i,j}(x_j) \leftarrow \{\mu_{ij}^k, \Sigma_{ij}^k, w_{ij}^k\}_{k=1}^K \quad (5.26)$$

In our framework, the position of samples  $\mu_{ij}^k \in \mathbb{R}^2$  is defined in the image space. Our inference method extends NBP by maintaining the pose of samples  $\vartheta_{i,j}$  in parallel by using additional parameters:

$$\vartheta_{i,j} \leftarrow \{\theta^k, a^k, s_x^k, s_y^k\}_{k=1}^K \quad (5.27)$$

The computation of an outgoing message in our framework is summarized in Algorithm 8. First, it starts by computing the exact product of incoming messages,

$\beta_{ts}(x_t) \leftarrow \phi_t(x_t) \prod_{i \in N(t) \setminus s} m_{it}(x_t)$ . From the produced variable  $\beta_{ts}(x_t)$ ,  $M$  weighted samples  $\{\bar{x}_t^i, \bar{\Sigma}_t^i, \bar{w}_t^i\}_{i=1}^M$  are drawn.

A pose is assigned to each newly created sample by using a procedure,  $pose()$ , that computes the mean pose (of the incoming message samples) around the sample point. Then the conditional  $\psi(x_j|x_i)$  can be applied to move each sample independently by using the relative position in the model. Resulting samples  $\{x_{tu}^i, w_{tu}^i\}$  are then assigned a new variance  $\Sigma_{tu}^i$  by using a k-nearest KDE estimation (See Appendix A). Finally, the pose  $\vartheta_{tu}^i$  of each sample is modified following the relative pose between the source and the target node.

The outgoing message is formed by using the parameters describing the location  $x_{tu}^i$ , variance  $\Sigma_{tu}^i$ , weight  $w_{tu}^i$ , and pose  $\vartheta_{tu}^i$  of the  $M$  samples:

$$m_{tu}(x_u) = \{x_{tu}^i, \Sigma_{tu}^i, w_{tu}^i, \vartheta_{tu}^i\}_{i=1}^M \quad (5.28)$$

The product of incoming messages, received by higher and lower level nodes, is detailed in Figure 5.8. It allows to obtain a localization of the feature. Figure 5.9 illustrates the message-passing algorithm by presenting the NBP detection process on an object.

### Why separate position and pose during NBP?

It is clear that the computation of the pose of samples could be integrated within NBP by using a higher dimensional space. This has been done by DETRY [Det06]. However, to be able to cover such a higher-dimensional space properly, the system would require a much larger number of samples. A consequence of this is that the inference process would require more computational complexity, especially for computing the products of messages.

## 5.5 Discussion

This chapter has presented a new approach to represent visual features. It extends recent work [Pia01, FPZ03] by introducing a flexible hierarchy of spatial dependencies between features.

This hierarchy allows to naturally overcome the limited descriptive power of individual primitive features by composing them into compound features. The same structure is repeated recursively in higher levels of the representation. Since compound features gradually contain more parameters and naturally cover a larger image

---

**Algorithm 8** NBP update of an outgoing nonparametric message
 

---

Given input messages  $m_{kt}(x_t) = \{\mu_{k,t}^i, \Sigma_{k,t}^i, w_{k,t}^i, \vartheta_{k,t}^i\}_{i=1}^N$  received from nodes  $k \in \mathcal{N}_t \setminus u$

1. // Compute The Exact Incoming Message Product

$$\beta_{ts}(x_t) = \phi_t(x_t) \prod_{i \in \mathcal{N}(t) \setminus s} m_{it}(x_t)$$

2. // Draw samples

$$\{\bar{x}_t^i, \bar{\Sigma}_t^i, \bar{w}_t^i\}_{i=1}^M = \text{Draw } M \text{ weighted samples from the product } \beta_{ts}(x_t)$$

3. // Compute the pose of each sample  $\bar{x}_t^i$  (Algorithm 9)

$$\{\bar{\vartheta}_t^i\}_{i=1}^M = \text{pose}(\bar{x}_t^i, \{\mu_{k,t}^i\}_{i=1}^N, \{\vartheta_{k,t}^i\}_{i=1}^N)$$

4. // Map the Conditional (Equation 5.18)

$$\{x_{tu}^i, w_{tu}^i\}_{i=1}^M = \text{apply } \{\bar{x}_t^i, \bar{w}_t^i, \bar{\vartheta}_t^i\}_{i=1}^M \text{ on the Conditional } \psi(x_u|x_t)$$

5. // Adjust the variance (Appendix A)

$$\{\Sigma_{tu}^i\}_{i=1}^M = \text{k-nearest kde}(\{x_{tu}^i\}_{i=1}^M)$$

6. // Map the relative pose (Equation 5.9)

$$\{\vartheta_{tu}^i\}_{i=1}^M = \{\bar{\vartheta}_t^i\}_{i=1}^M \times \vartheta_{tu}^R$$

7. // Compose the outgoing message

$$m_{tu}(x_u) = \{x_{tu}^i, \Sigma_{tu}^i, w_{tu}^i, \vartheta_{tu}^i\}_{i=1}^M$$


---

---

**Algorithm 9**  $\bar{\vartheta}_t^i = \text{pose}(\bar{x}_t^i, \{\mu_{k,t}^i\}_{i=1}^N, \{\vartheta_{k,t}^i\}_{i=1}^N)$ 


---

$$\{\theta, a, s(x), s(y)\}_{i=1}^N \leftarrow \{\vartheta_{k,t}^i\}_{i=1}^N$$

$$\bar{\theta} \leftarrow \text{WeightedCircularMean}(\bar{x}_t^i, \{\mu_{k,t}^i\}_{i=1}^N, \{\theta^i\}_{i=1}^N)$$

$$\{\bar{a}, \bar{s}_x, \bar{s}_y\} \leftarrow \text{gaussianWeightedMean}(\{a^i, s_x^i, s_y^i\}_{i=1}^M, \{\mu_{k,t}^i\}_{i=1}^M, \bar{x}_t^i)$$

$$\bar{\vartheta}_t^i \leftarrow \{\bar{\theta}, \bar{a}, \bar{s}(x), \bar{s}(y)\}$$


---

---

**Algorithm 10**  $\bar{\theta} \leftarrow \text{WeightedCircularMean}(\bar{x}, \{\mu\}_{i=1}^N, \{\theta\}_{i=1}^N)$ 


---

$$\{w\}_{i=1}^N = \text{evaluate a gaussian centered at } \mathcal{G}(\bar{x}) \text{ for each } \mu_i$$

$$\mathcal{C} = \sum_{i=1}^N w_i \cos \theta_i$$

$$\mathcal{S} = \sum_{i=1}^N w_i \sin \theta_i$$

$$\text{return } \bar{\theta} = \arctan(\mathcal{S}/\mathcal{C})$$


---

area than primitives do, they should provide a potentially more specific and robust description of relevant aspects of shape and appearance.

We can notice several other properties that are inherent to our hierarchy:

- + Flexible representation of spatial relations,
- + Natural separation of appearance and shape,
- + Fine-grained representation of high-level spatial relations,
- + Appearance model at high levels,
- + Different levels of abstraction within the same model,
- + Viewpoint invariance (rotation, location, scale),
- + Top-down and bottom-up influence during detection,
- + Sparse model,
- + Similarities with neuroscience models of the visual cortex.

The hierarchical model has also a few limitations that we should keep in mind:

- The object is assumed to be made up of a set of parts,
- The relative location between the parts is expected to be captured by a mixture of Gaussians,
- The distribution of appearance for each part is represented by an unimodal Gaussian,
- Only pairwise relations are represented.

The efficacy of the model depends on its structure and the accuracy of its parameters. Therefore, the learning strategy of such a model is crucial. In the next chapter, we will focus on the incremental composition of such a model and describe how it can be constructed.

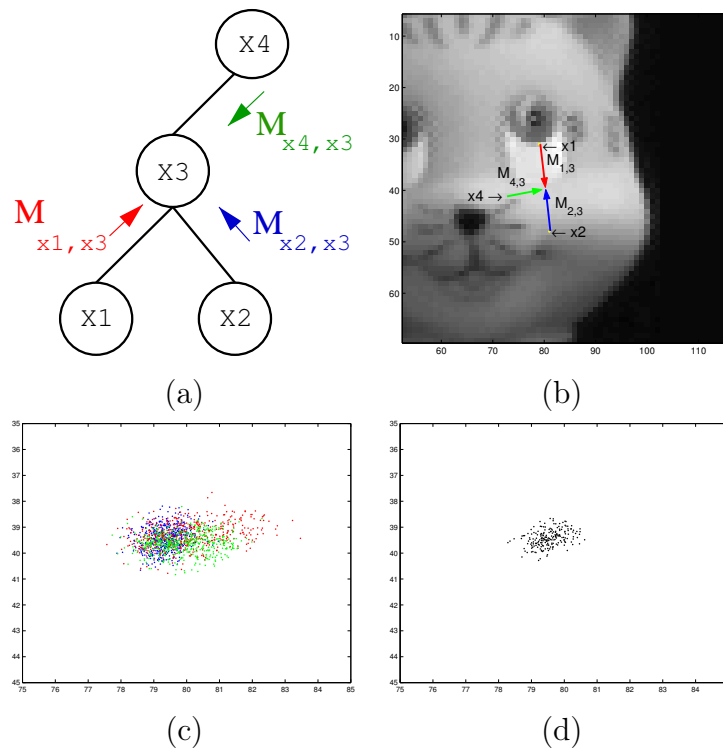


Figure 5.8: During an iteration of NBP (a, b), feature  $x_3$  received messages from subfeatures  $x_1, x_2$  and parent  $x_4$ . Even if individual messages contain uncertain information about the location of a feature (c), the product of the incoming messages constrains the location of feature  $x_3$  (d).

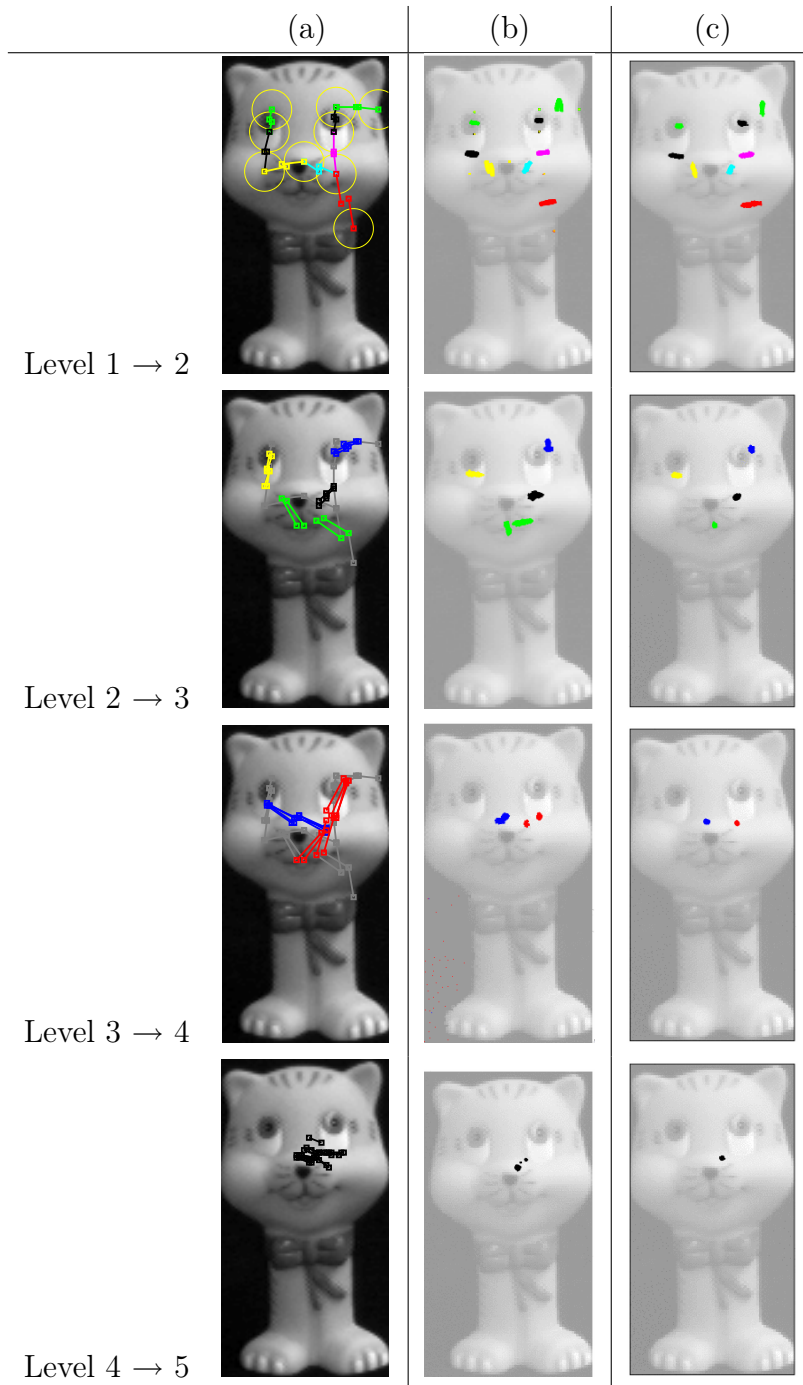


Figure 5.9: Illustration of an upward message-passing iteration during the NBP detection process. Starting from the first level (a), the detection process uses the presence of primitives to predict the location of the second level features, top (b). The product of these messages (c) refines the belief to a more precise localization. This product is then used for the next level (a). At the end of this simple example, we obtain the final set of samples that corresponds to the localization of the object.

# Statistical Learning of Hierarchies

---

This chapter provides another cornerstone of the recognition framework by introducing a statistical learning method from which the proposed hierarchy can emerge. In our context, the purpose of learning is a threefold one. First, the structure of the model itself is initially unknown; therefore it should be constructed automatically and incrementally. At the same time, a second goal is to estimate the parameters of the model that best fit the training data. These parameters comprise spatial relations, appearance and shape between visual feature classes. Third, for recognition purposes, the system should be able to exploit these models to predict the presence of the most likely object class in the image.

To address these challenges, the proposed learning framework is divided in two parts. The first part composes the hierarchical model iteratively in a bottom-up manner (Section 6.4). It starts with simple low-level features and gradually generates more complex features. For each newly created feature, the system learns in parallel a generative model of its spatial relations. The second part exploits the feature hierarchies previously learned, and creates a discriminant model that can predict the presence or the absence of an object class in the image (Section 6.5).

Before entering the technical discussion, we introduce in Section 6.1 the general learning environment in which the framework will perform. An overview of the system is given in Section 6.2 and Section 6.3 gives the rationale for the structure and the strategies used in our statistical learning method.

## 6.1 Learning Context

The structure of a learning process is often shaped by the environmental context in which it performs. Depending on the level of supervision available, it is common to differentiate between supervised and unsupervised learning methods. Supervised methods learn a model from input to output values, both observed by the system. In contrast with those methods, unsupervised learning methods try to discover structure in the data without any external measure of success.

In object recognition, this terminology is extended to better match with the reality of the problem. The term “supervision” is taken in a more general sense. Specifically, it is possible to distinguish between three types of supervision: the human effort required to train the models, the degree of generality of the images, and the number of training images required. For the first type, we can mention four levels of supervision [Ope06]:

- *Unsupervised*: the data available to the system consist of a set of unordered and unlabeled images with no information about the object locations.
- *Weakly Supervised*: here the object labels are available during training. This labelling associates to each image the label (the name or the class) of the object present in the image. Therefore, the learning system is presented with a collection of images  $\mathcal{I}_{1..n}$  containing examples of objects belonging to a given class  $\mathcal{O}$ :

$$\mathcal{T} \leftarrow \{\mathcal{I}_i, \mathcal{O}_i\}, i = \{1 \dots n\}, \mathcal{O}_i \in \{1 \dots N_o\} \quad (6.1)$$

where  $\mathcal{T}$  is the training set presented to the learning algorithm,  $\mathcal{O}_i$  denotes the object label of an image  $\mathcal{I}_i$ ,  $n$  is the number of training images and  $N_o$  stands for the number of object labels.

- *Supervised*: in addition to the object labelling information, supervised object recognition methods are given bounding boxes that determine the location of the object instances in those images.
- *Highly Supervised*: these methods have object labelling information and bounding boxes available. Moreover the objects are segmented in the training images such that each pixel of the image is either a part of the object or the background.

Since it only receives the object labelling information, the learning framework presented in this work is *Weakly Supervised*.



## 6.2 Overview

In this section, we introduce the learning method that is used to construct the feature hierarchies and to learn a classifier from them. From a general point of view, the challenge of learning hierarchies is to make sense of the data available by automatically learning the visual consistency between instances of the same object class. To this end but also to perform recognition, our system exploits different learning paradigms:

### 1. Structural Learning

For each object class, the structure of the graphical model is constructed iteratively in a bottom-up manner. At each level of the hierarchy, pairs of features are identified that tend to occur in the same relative neighborhood.

### 2. Generative Learning

When two features are combined to produce a high-level feature, the system learns a generative model of the spatial relations existing between them. This model, which is represented as a mixture of Gaussians, is estimated by clustering the configurational distributions of observed feature co-occurrences using Expectation-Maximization. In addition, an appearance model can also be estimated for the newly created feature. By doing so, the feature becomes directly observable in the image.

### 3. Discriminative Learning

Once a hierarchical model has been learned for each object class, they can be used for detection in new images. To perform recognition, a discriminative model is learned on the top of the feature hierarchies. The general idea is to construct a multi-class Support Vector Machine classifier (SVM) [BGV92] from the activation of the features.

In practice, the first two phases are applied in parallel to build the hierarchies. They are referred to as the *co-occurrence learning algorithm* and will be presented in Section 6.4 under Algorithm 11. The third phase, which will be explained in Section 6.5, is independent and is applied separately once a model has been learned for each object class.

Figure 6.1 illustrates a summary of the learning techniques employed in this thesis. The system starts building the model upwards from the input set of observations with the co-occurrence learning algorithm (Figure 6.1 (a)). It is possible to

improve the overall structure of the model by learning an observation model (*i.e.* likelihood) for each feature (*i.e.* hidden variable) in the graphical model (Figure 6.1 (b)). Therefore, hidden variables learned from the co-occurrence algorithm become observable. This leads to potentially more informative data that can be exploited as input by a supervised learning process. Then during a second phase, the system exploits the full hierarchy (the primitives and the newly created variables) as inputs in a discriminative learning procedure (Figure 6.1 (c)).

## 6.3 Motivation

The learning method introduced in the preceding section is grounded on three different learning paradigms that naturally give rise to the following questions:

- Why combine generative and discriminative models?
- How is the learning related to the task?
- Why use incremental learning?
- Why use co-occurrence statistics as a criterion to compose features?

In this section, we present a concise overview of the motivations behind these questions. Specifically, Section 6.3.1 motivates the use of both generative and discriminative models within the same framework. Some insight is given in Section 6.3.2 about the relationship between our learning system and the task. Then we review the reasons of using incremental learning to create higher-level visual abstractions in Section 6.3.3. Finally, we answer the last question in Section 6.3.4 and present a discussion about the use of co-occurrence analysis.

### 6.3.1 Combining Generative and Discriminative Models

The combination of generative and discriminative models within a single framework has demonstrated good results in several visual recognition applications [HWP05, CLS05, KPM06].

In general, representing the geometry of object parts is more suitable with generative models. This can be explained because different object classes may have similar parts (*i.e.* sharing similar feature configurations). Therefore the representation of spatial relations in terms of the object class boundaries may fail.

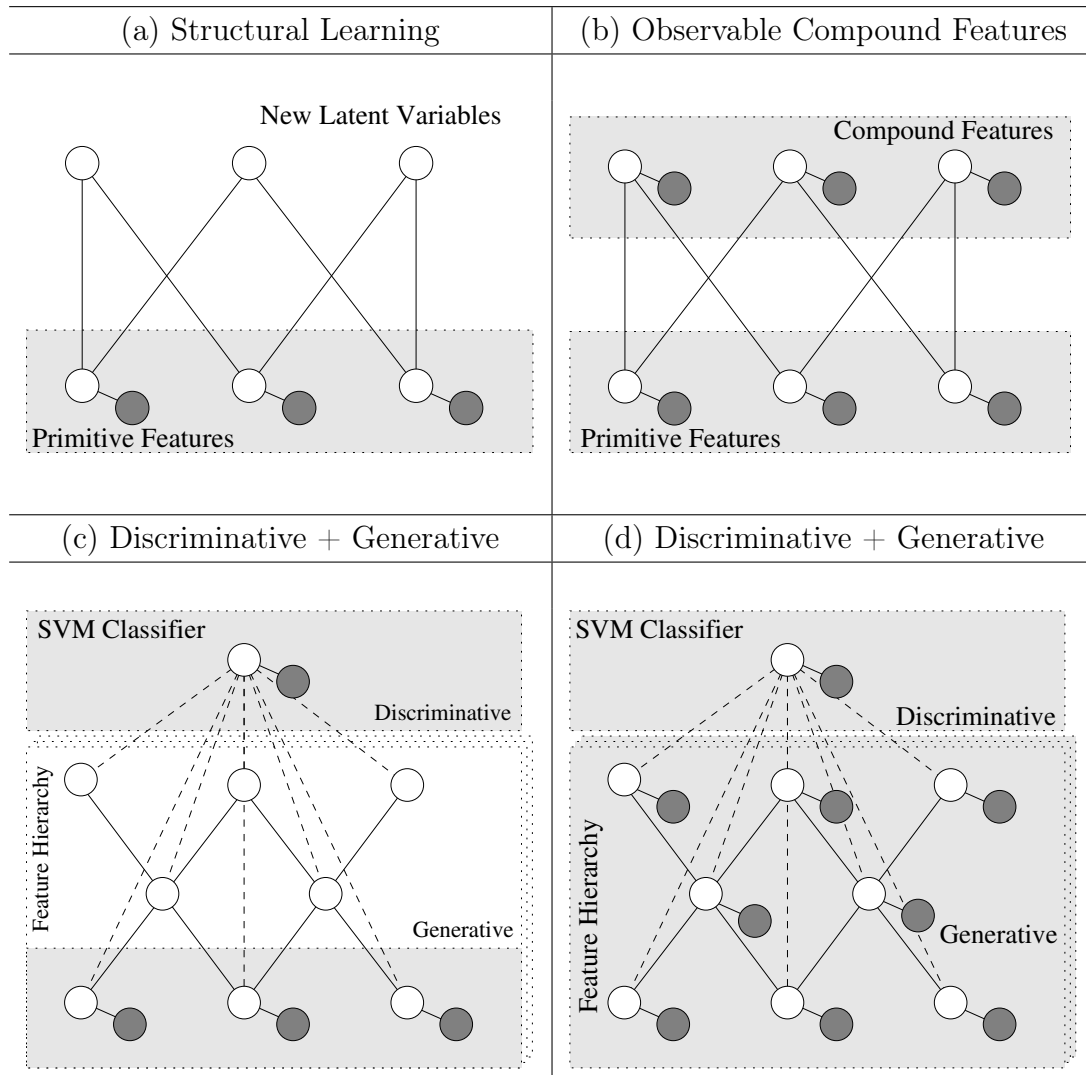


Figure 6.1: A graphical model formalism is used to illustrate the different aspects of our learning strategies. Hidden (or latent) variables are represented by white circles and observations by shaded circles. During our co-occurrence learning (a), the system tries to establish the relation between the inputs and some latent variables. The model is generative. A natural extension to this strategy is to learn an observation model for each variable learned from the co-occurrence algorithm (b). This is expected to give better results since more information can be extracted from the environment. Once a model has been learned for each object class, a discriminant model (c) is trained to predict the object class label from the feature activation of the hierarchy. The corresponding model where the full hierarchy is observable is shown in (d).

However, for recognition, the process has to predict the presence or the absence of the object class in the image given the features. To do this, discriminative models are most often used in the literature and generally offer good results.

Summing up, we expect that the use of generative models to estimate the parameters (geometry and appearance) of the models, and discriminative models to learn the classifier to predict the class label, will be beneficial to the system performance.

### 6.3.2 Task-Driven and Task-Independent Learning

Another way to justify the structure of our learning framework is to consider how the learning framework is related to the task. If we turn on the literature of object recognition, we see that the problem is currently too vast and complex to be addressed by a single framework. Researchers often consider subclasses of the main problem (*e.g.* detection, categorization, matching, *etc.*). Therefore state-of-the-art solutions can be differentiated between the task they aim to complete. Most often those frameworks exploits a *Task-Driven* learning strategy. In contrast with these methods that learn features in a *task-driven* way (something we did in [JSP05]), our co-occurrence learning algorithm is *task-independent*.

There are several advantages to perform learning in a *task-independent* way. The most important reason is that some visual features become only useful for the task when they are combined with others, in higher level in the hierarchy. However, we do not know in advance at which level in the hierarchy a feature will become useful. Therefore by learning these reliable structures in a task-independent way, we may find complex configurations of features that would have been more difficult to find in a task driven way.

Moreover, by using such a learning strategy we expect that the feature hierarchies learned in a task-independent way can be useful for different tasks (matching, detection, *etc.*).

### 6.3.3 Structure Learning via an Incremental Strategy

Statistical Learning methods for Object Recognition are generally concerned with the estimation of the object model parameters (appearance and geometry). The structure of the statistical object model is often defined a priori [FPZ03, BT05, CFH05, LTGK06] by manually selecting an appropriate topology (*i.e.* number of parts). For instance, the constellation model [FPZ03] generally assumes that an ob-

ject model is composed of four inter-related different parts. This a priori information allows the system to learn models that are robust to large appearance variations. However, it constitutes a strong requirement that reduces the generality of this kind of approach.

A more realistic, but also more challenging way to learn object models is to allow the learning scheme to construct their structure in an automatic (possibly incremental) fashion [PG99, PG00a, KGA02, OPZ06]. In this thesis, unlike many existing frameworks, we turn to this type of statistical learning. In contrast with previous work [PG99], the term *incremental* does not mean that the system processed images *serially* but rather that the system gradually organizes its perceptions in ever higher level abstractions.

Despite the fact that it is often more difficult to set up, three main motivations lead us to build a learning system that performs in an incremental fashion; intuition, biology and psychology. First, it is my intuition that we could improve our ability to recognize unfamiliar object classes if we have the opportunity, the desire, and the time to learn them by accumulating experience and thus refining our perception of these special classes of object. Second, this intuition is strengthened by the biological point of view [Kan79, Cha02]. It has been shown that certain primary animals with only very rudimentary nervous systems exhibit interesting behaviors and are able to learn to survive in their environment. Therefore the amount of experience accumulated by an animal is a factor that improves its expertise and its ability to perform given tasks. Third, these concepts have also been observed in humans and formalized in the Gestalt Psychology [Wer23, Köh47]. They have been proven to be essential factors to the learning of human vision abilities [Wer23]:

“ *Another Factor is that of past experience or habit. Its principle is that if AB and C but not BC have become habitual (or "associated ") there is then a tendency for ABC to appear as AB/C. Unlike the other principles with which we have been dealing, it is characteristic of this one that the contents A, B, C are assumed to be independent of the constellation in which they appear. ...*

”

We can see that the notion of time and expertise is natural in incremental learning. All these reasons lead us to consider incremental learning as a basis in our object learning framework.

Our recognition system relies on the learning of reliable object models that are represented by feature hierarchies. From a theoretical point of view, this learning can be seen as the act of bridging the gap between low-level and high-level visual concepts (*i.e.* between visual primitives and object classes). The system will proceed incrementally from the observable inputs into more abstract levels of representation. Each additional level in the hierarchy can be seen as an additional step towards the high-level abstractions.

Incremental learning performed in a task-independent way inevitably requires an internal criterion for combining low-level features into higher level concepts. A commonly used technique is to rely on the analysis of correlation between variables. In the case of object recognition, the notion of correlation is generalized to spatial correlation which means that two visual features are likely to co-occur in the same images. This concept is detailed in the next section.

### 6.3.4 Co-occurrence Analysis

One simple and widely-used method for establishing statistical associations between features is via co-occurrence analysis. This method is very commonly used in various domains. For instance it has been used to validate learning models in psychology [FA01], to predict Gestalt rules in neuroscience [EG98, Krü98, GPSG01] and to retrieve high-level concepts in data mining and text document analysis [WR93, SC99, MAF<sup>+</sup>04]. This section gives an insight into the motives that lead researchers to explore the analysis of co-occurrences through these different fields.

**Definition 6.1.** A *Co-occurrence* is an event or situation that happens at the same time as or in connection with another.

A high probability of co-occurrence is often considered as synonym of stability between two features. Specifically, it can be useful to efficiently find more complex structures that tend to be repeated across different images. Co-occurrence assumes interdependency of the two terms and can also be interpreted as an indicator of semantic proximity.

In experimental psychology [FA01, FA02, FA05], it has been shown that human observers paid more attention not only to feature pairs that often co-occurred in the images as embedded elements, but also to pairs that had higher predictability (conditional probability) between the constituent features. The authors suggested from these findings that subjects learn higher level visual features based on the statistical coherence of features within the images. This unsupervised learning ability

of human observers assumed that they extract the joint and conditional probabilities of shape co-occurrences during passive viewing of images.

In neuroscience, some researchers focus on bridging the gap between the Gestalt model of Psychology [Wer23] and statistical measurements in images [Zhu03]. Likewise, co-occurrence analysis has been proven to be essential in the understanding of many Gestalt principles; such as proximity, collinearity [Krü98, KW02], parallelism, etc. These are based on various formulations of co-occurrence statistics of oriented filter responses in natural images [SCGM01]. They find that contour detection performance is quantitatively predicted by a local grouping rule derived directly from the co-occurrence statistics [GPSG01].

Among the broad variety of possible applications, the most popular use of co-occurrence statistics is certainly the text document analysis [HP98, BWL02] and data mining [SKW05, SPKW06] (*e.g.* personal shopping profiles). Deriving concepts from co-occurrence analysis of text documents often leads to the definition of a hierarchical structure. In general, there are two ways to construct the structure: bottom-up (upward) or top-down (downward). The top-down method starts with a high-level model to understand texts. This manner is efficient when the documents are tightly structured, but remains challenging in the general case. The bottom-up manner starts with observable data to build higher-level conceptual entities and is often more flexible.

If we take a bird's eye view on the subject, we observe that the study of co-occurrence is a common factor of learning between the different fields we have discussed. Therefore, the choice of co-occurrence analysis as a basis for our learning algorithm seems appropriate. It is coherent with BARLOW's theory of visual recognition [Bar89, Bar94]:

“ *Detecting “suspicious coincidences” of elements during recognition is a necessary prerequisite for efficient learning of new visual features.* ”

Intuitively, two candidate features should be combined into a composite object if they have a high probability of their joint appearance. This insight provides a strong motivation in favor of co-occurrence analysis for in visual recognition [EHYI01].

This thesis claims that feature composition for recognition tasks can be accomplished on the basis of correlation between features, and that such a feature composition procedure can be beneficial to machine learning algorithms for object recognition. Having reviewed the main ideas behind our framework, we describe, in the next section, our co-occurrence learning algorithm.

## 6.4 Composing Features into Hierarchies

This section provides the description of a co-occurrence learning algorithm which is designed to produce visual feature hierarchies. In probabilistic terms, co-occurrence learning first aims at defining the hierarchical latent structure of the Pairwise Markov Random Field by defining nodes compositions through edges. Its second purpose is to estimate the model parameters (*i.e.* spatial relations) through conditional functions by maximizing their likelihood on the image training set.

### 6.4.1 Co-occurrence Learning Algorithm

The basic concept behind this learning algorithm is to accumulate statistics of the relative positions of observed features in order to find frequently-occurring feature co-occurrences. The structure of the model is built incrementally by combining spatially correlated feature classes into new feature abstractions. The learning of the model can be summarized by an iterative procedure, whose outline is given in Algorithm 11.

First, a clustering algorithm (K-means [HW79]) is applied to the set of descriptors  $\mathcal{D}$  of local regions previously extracted from the training set. This yields a visual codebook that is used to create the first level of the graph  $\mathcal{G}$ . Each feature class is associated with a visual word of the codebook to create an appearance class. After clustering, the training procedure accumulates information on the relative positions  $\Lambda$  of features and their image locations  $\Phi$ . It extracts those feature pairs  $\mathcal{C} \leftarrow [f_i, f_j]$  that tend to be located in the same neighborhood.

Then it estimates the parameters of their geometric relations  $\mathcal{S}_{ij}$  using Expectation-Maximization (Section 6.4.5). It selects the closest relations and estimates their shape  $\mathcal{X}$  and their appearance model  $\mathcal{A}$  using an adaptive patch (Section 6.4.7). This key component describes the appearance of the new feature whose width and height are automatically determined. The optimality criterion is based on minimum-variance analysis, which first computes the variance of the appearance model for various patch deformations, and then selects the patch dimensions that yield the minimum variance over the training data.

Finally, it generates new visual feature classes by adding new nodes in the graphical model (Section 6.4.6). The same process is applied iteratively to each new level in the graph. In the following sections, we describe the main steps of this co-occurrence based learning procedure.



---

**Algorithm 11** Co-occurrence Learning: learn()

---

```
1: // extract a set of low-level visual feature occurrences from the training set
2:  $\{\mathcal{D}\} \leftarrow$  regions extracted from the training set
3: // find primitive classes  $f_p$  by applying K-means clustering on  $\mathcal{D}$ 
4:  $\{f_p, \mathcal{A}_p\} \leftarrow$  K-means( $\mathcal{D}$ )
5: // construct the first level of the graph
6:  $\mathcal{G} \leftarrow$  create( $f_p, \mathcal{A}_p$ )
7: for each level  $< nLevels$  do
8:   // extract co-occurrence statistics: correlated features  $\mathcal{C}$ , their relative positions  $\Lambda$  and image locations  $\Phi$ 
9:    $\mathcal{C}, \Lambda, \Phi \leftarrow$  extract( $\mathcal{G}$ , level)
10:  for each correlated feature class pair  $[f_i, f_j] \in \mathcal{C}$  do
11:    // estimate spatial relational model  $\mathcal{S}_{ij}$  between  $i$  and  $j$ 
12:    if Parametric then
13:       $\mathcal{S}_{ij} \leftarrow$  EM( $\Lambda_{i,j}$ ) // where  $\Lambda_{i,j}$  is the set of relative positions btw  $i$  and  $j$ 
14:    else
15:       $\mathcal{S}_{ij} \leftarrow$  Resample( $\Lambda_{i,j}$ )
16:    end if
17:    // store the model  $\mathcal{S}_{ij}$  into the set  $\mathcal{S}$  which contains the candidate models for the current level
18:     $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_{ij}$ 
19:  end for
20:  // keep the closest spatial relations
21:   $\mathcal{S}' \leftarrow$  closest( $\mathcal{S}$ )
22:  // estimate shape, appearance model
23:   $[\mathcal{X}, \mathcal{A}] \leftarrow$  adaptivePatch( $\mathcal{S}', \Phi$ )
24:  // connect new nodes to the graph
25:   $\mathcal{G} \leftarrow$  generate( $\mathcal{X}, \mathcal{A}, \mathcal{S}', \mathcal{G}$ )
26: end for
```

---

## 6.4.2 Local Feature Extraction

The purpose of the feature extraction is to reduce the visual input space to a set of local descriptors  $\mathcal{D}$ . To be called *generic* a recognition framework should ideally be able to learn different object classes without restrictions concerning their shape, appearance or texture. A common problem in object recognition is that different object classes might be described by different visual properties. Most of existing approaches only use one kind of feature detector. However, none of these one-type detectors is able to cover the different classes.

OPELT *et al.* [OPFA06] recently combined multiple methods to capture the main characteristics of various object categories. This improves the generality of their approach and therefore motivates us to utilize a similar feature extraction scheme. As illustrated in Figure 6.2, the feature extraction process is composed of two phases. The first one locates regions of possible interest in images; various interest point extraction techniques are used. The second computes local descriptors on previously detected regions. Below we give more details concerning these two operations.

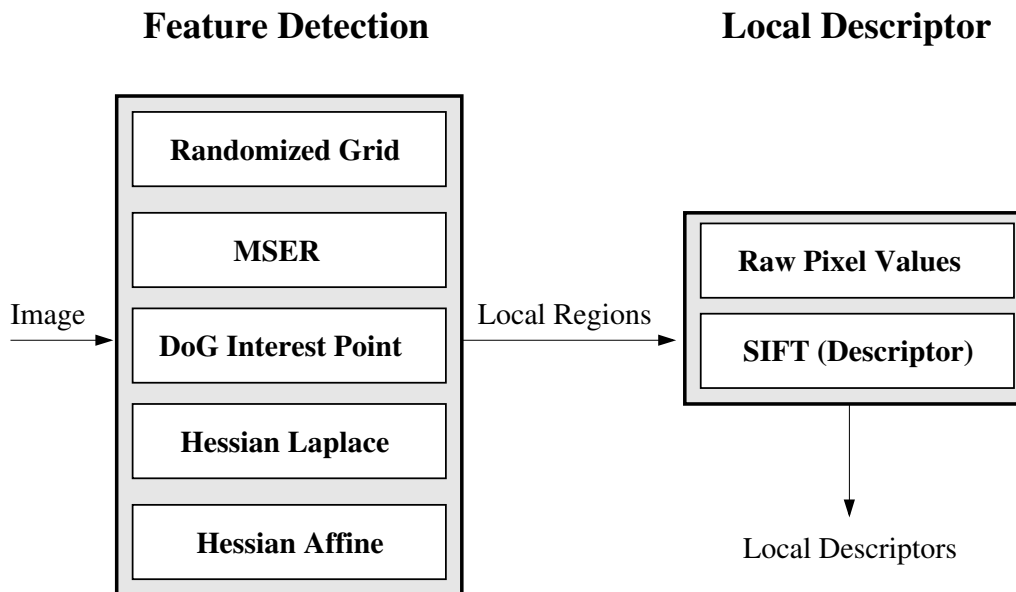


Figure 6.2: Illustration of the feature extraction process. Each image is first processed by standard feature detectors to produce a subset of local regions. These are then extracted and described in an invariant manner by some feature descriptor.

## Region Detection

As it has been discussed in Chapter 3, feature detectors use various kind of image measurements to locate potentially interesting areas in images. Depending on their type, detectors can be invariant to a certain degree of geometric and/or photometric transformations. The detectors provide region locations, orientations and scales. For some of them (MSER [OM02a] and Hessian-Affine [MS04]), an affine normalization is also available. This information is later used to compute descriptors. In our framework, we are able to use five different detectors jointly:

**Maximally Stable Extremal Regions** (MSER) [OM02a] define extremal regions that are derived from a watershed segmentation algorithm. These regions remain stable over a large threshold range. The region center, size and orientation are given by corresponding information of the surrounding ellipse.

**DoG** regions are localized at local scale-space maxima of the difference-of-Gaussian. This detector is often combined with SIFT descriptors.

**Hessian-Laplace and Hessian-Affine** regions [MS04] are localized in space at the local maxima of the Hessian determinant and in scale at the local maxima of the Laplacian-of-Gaussian.

**Randomized Sampled Grid** is constructed from evenly sampled grid spaced at  $s \times s$  pixels for a given image. Then each location is displaced by a random amount chosen between  $[0, s_r[$ . The size of the patch is proportional to the local scale [Lin98]. The orientation of each grid location corresponds to the dominant gradient orientation.

## Region Description

In order to make the information suitable for the learning algorithm, each region is represented by a local descriptor. It describes the appearance of a region of interest. In this framework, we used two different kinds of descriptors:

**SIFT** descriptors are histograms of gradient locations and orientations, where locations are quantized into 4x4 location grid and the gradient angle is quantized into 8 orientations. The resulting descriptor is a 128 dimensional vector.

**HSV pixels values** Each image region is mapped to a  $13 \times 13$  window of pixels. We obtain orientation invariance by normalizing the region with the orientation obtained from the detectors.

### 6.4.3 Visual Classes

The first step towards the learning of an object model consists in the definition of a basis of low-level feature classes (*i.e.* primitives) from which the hierarchy will be constructed. Each of these classes, denoted  $f_p$ , lies at the first level in the graph and is associated to an appearance model  $\mathcal{A}_p$ .

To obtain these models, our strategy is to perform a K-Means algorithm [HW79] to cluster the descriptors from all training images into a fixed number  $k$  of partitions. This technique is very commonly used by both bag-of-features and statistical object recognition systems [OPZ06] to obtain a reduced set of visual classes. Various similarity metrics can be used to differentiate two observed descriptors in the algorithm. In our system, we use the Euclidean distance, which defines the distance between two descriptors  $P = [p_1, p_2, \dots, p_n]$ , and  $Q = [q_1, q_2, \dots, q_n]$ , in Euclidean  $n$ -space as:

$$d(P, Q)^2 = \sum_{i=1}^n (p_i - q_i)^2 \quad (6.2)$$

K-Means clustering finds a grouping of the observations that minimizes the within-cluster sum-of-squares. Each observation, represented by a descriptor, is grouped so that it is assigned to one of the  $k$  clusters. Clusters with too small number of members are eliminated. The centroids of these clusters become the mean  $\mu_p^A$  of the appearance model  $\mathcal{A}_p$  of our primitives  $f_p$ . The covariance  $\Sigma_p^A$  is computed using the members assigned to the class.

#### How many clusters should be formed?

Generally the number of clusters in the dataset is not known in advance. Several algorithms have been proposed to determine this value  $k$  automatically. A usual idea is to run the algorithm with different values of  $k$  and score each clustering model using a likelihood criterion. In this work, the number of classes is selected according to the Bayesian Information Criterion (BIC) [Sch78]<sup>1</sup>:

$$BIC = -2L(X|C) + \frac{p}{2} \log(n) \quad (6.3)$$

where  $L(X|C)$  is the log-likelihood of the dataset  $X$  according to model  $C$ ,  $n$  is the number of observations in the dataset and  $p = k(d+1)$  is the number of free parameters in the model  $C$  with dimensionality  $d$  and  $k$  cluster centers. It is used to support

<sup>1</sup>other scoring functions are available: Akaike's Information Criterion (AIC), Integrated Completed Likelihood (ICL), Normalized Entropy Criterion (NEC), cross-validation criterion (CV).

models that contain a fewer number of clusters. A standard convention [KR93] to select the right number of clusters is to consider BIC logarithm differences (Table 6.1). The choice is done on the model where a decisive variation is detected in BIC differences (Figure 6.3). More robustness can be obtained by averaging the BIC differences over a small interval (+-1, +-2). We observed that our method was not critically linked to the selection of a cluster number  $k$  as long as it was not too small. Therefore we choose a low threshold to select the number of visual classes. Complete studies of this technique can be found in the literature [Sch78, KW95, GH03, FR98].

differences ( $2 \log_e BIC$ )	evidence
less than 2	weak
between 2 and 6	positive
between 6 and 10	strong
greater than 10	very strong

Table 6.1: The difference between two successive BIC values can be classified in 4 classes; differences of less than 2 correspond to weak evidence of decisive variation, between 2 and 6 to positive evidence, between 6 and 10 to strong evidence, and greater than 10 to very strong evidence.

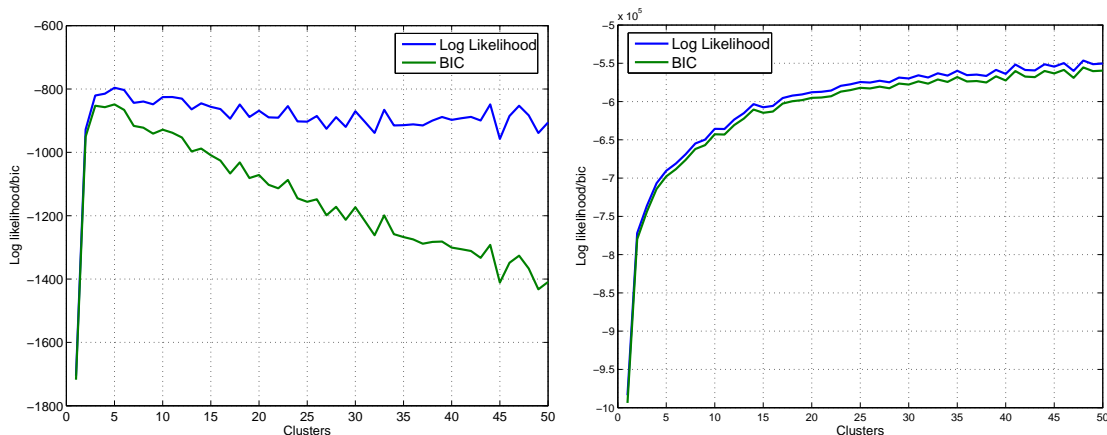


Figure 6.3: Bayesian Information Criterion (green) and Log-Likelihood (blue) measure for different number of clusters in k-means are illustrated on *iris* [DJNM98] and *letter* [MSTC94] databases. In the ideal case (left), BIC function reaches a peak at the best number of clusters (3). When the clusters are more fuzzy defined (right), the function quickly increases at the beginning and then tends to remain constant.

### 6.4.4 Finding Correlated Feature Classes

The composition of new features is based on the finding of spatially correlated feature classes. The basic idea behind this process is a twofold one. First to collect information concerning the relative positions  $\Lambda$  and locations  $\Phi$  between features from training images. Second to extract pairs of feature classes  $\mathcal{C}$  that have a large number of local *co-occurrences* and therefore tend to be located in the same neighborhood. Interestingly, these co-occurrence statistics are collected from multiple feature instances within one or across many different images. We keep of the feature co-occurrence locations  $\Phi$  is the image for facilitating the estimation of their appearance model. The procedure to find correlated features and extract their relative positions is explained below and summarized in Algorithm 12.

#### Algorithm

The execution of this algorithm intends to provide three different sets of data; the pairs of correlated feature classes  $[f_i, f_j] \in \mathcal{C}$ , their relative positions and poses  $[p_r, \vartheta_r] \in \Lambda$  and the positions  $\Phi$  of their co-occurrences across the training set.

The Algorithm assumes that feature instances have previously been extracted from the feature detectors. For each possible pair of feature instances observed in the same image, it computes the relative position  $p_r$  of one feature  $o_j$  versus the other; the reference feature  $o_i$ . This position is relative in the sense that it is normalized with either the local affine pose  $\vartheta_i$  of  $o_i$ , its scale  $s_i$  or its orientation  $\theta_i$ . These normalization procedures are encompassed under the function  $\mathcal{N}_{o_i}(o_j)$ , and detailed in the next paragraph.

Once the relative position  $p_r$  is known, the function tests if the point lies in the neighborhood of the reference feature. This is done by using a threshold on the relative scale-normalized distance between the feature instances  $o_i, o_j$ . If it satisfies this condition, the system computes the relative pose (or orientation) from the reference feature instance to the other and sets the middle position between them as the position of their composition.

At the same time, for each feature pair, a co-occurrence measure is calculated as the counts of simultaneously observing features  $i$  and  $j$  in the normalized image neighborhood. This is illustrated in Figure 6.4. A feature pair is considered to be spatially correlated if this number is above a predefined threshold. After the completion of this function, we obtain a list  $\Lambda_{i,j} \in \Lambda$  concerning the relative poses  $\vartheta_r$  and locations  $p_r$  of correlated feature classes  $[f_i, f_j] \in \mathcal{C}$ .

---

**Algorithm 12** Correlation Extraction:  $\text{extract}(\mathcal{G}, \text{level})$ 

---

```
1:  $\mathcal{C} \leftarrow \{\}$  // correlated feature class pairs
2:  $\Lambda \leftarrow \{\}$  // co-occurrence statistics (relative positions and poses)
3:  $\Phi \leftarrow \{\}$  // locations of co-occurrences in images
4: Successively extract each image  $I$  from the training set
5: Detect all features  $f_I \in \mathcal{G}$  for the given level in image  $I$ 
6: for all feature class pairs  $[f_i, f_j] \in f_I$  do
7:   for all instances  $o_i \in \mathcal{O}_i(I)$  of  $f_i$  do
8:     for all instances  $o_j \in \mathcal{O}_j(I)$  of  $f_j$  do
9:       // Compute the relative position  $p_r$  of  $o_j$  given  $o_i$ 
10:       $p_r = \mathcal{N}_{o_i}(o_j)$ 
11:      if  $\text{distance}(o_i, p_r) < t_d$  then
12:        // Compute the relative pose  $\vartheta_r$  of  $f_j$  given  $f_i$ 
13:         $\vartheta_r = |\vartheta_j - \vartheta_i|$ 
14:        // Store the observation
15:         $\Lambda_{i,j} \leftarrow \Lambda_{i,j} \cup \{p_r, \vartheta_r\}$ 
16:        // Store the middle position of the co-occurrence
17:         $\Phi \leftarrow \Phi \cup |o_i + o_j|/2$ 
18:      end if
19:    end for
20:  end for
21:  if  $\text{size}(\Lambda_{i,j}) > t_c$  then
22:     $\mathcal{C} \leftarrow \mathcal{C} \cup [f_i, f_j]$ 
23:  end if
24: end for
25: return  $\mathcal{C}, \Lambda, \Phi$ 
```

---

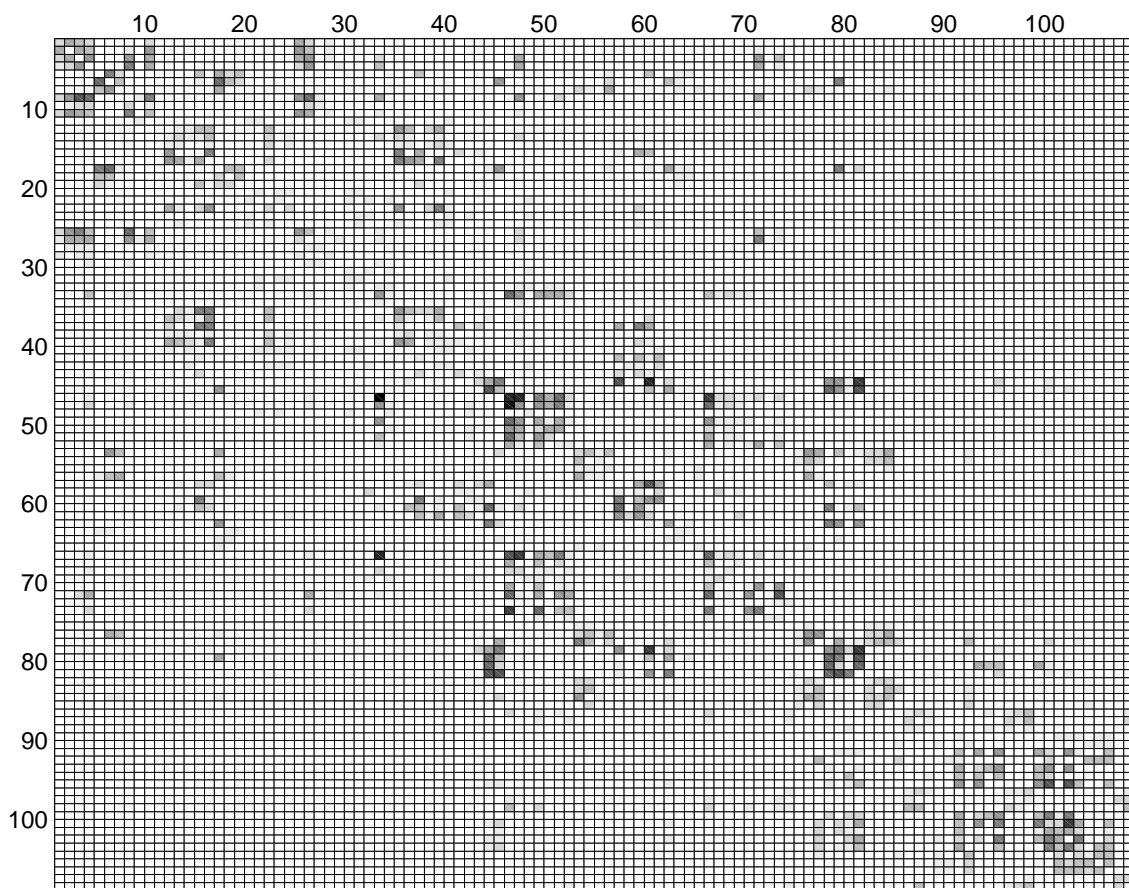


Figure 6.4: Visualization of the number of co-occurrences  $\Lambda$  between all the possible pairs of feature classes. They are extracted from interest points for an object of COIL-100 using Algorithm 12. Darker squares indicate a strong neighbor relationship between two features whereas white indicates that the features were never observed in the same neighborhood. Note that the table is not completely symmetric since the size of the neighborhood taken into account is set proportional the scale of the reference feature.



**Neighborhood Normalization**

During learning, visual features may occur at different orientations and scales in the image. To extract invariant relations between features, we normalize the neighborhood of each reference feature to a canonical frame. There are several strategies to achieve such a normalization. In the following, we explain three methods that have been used in our framework.

**1. Gradient Orientation.** A natural choice to normalize a local image region is to use the local orientation  $\theta_i$  associated with the reference point  $p_i = \{x_i, y_i\}$ . For observable features, this orientation corresponds to the gradient direction computed on the local region:

$$\tan \theta_i = \frac{L_y(x_i, y_i)}{L_x(x_i, y_i)} \quad (6.4)$$

$$L_x = \frac{\partial}{\partial x} g(\sigma) \otimes I \quad L_y = \frac{\partial}{\partial y} g(\sigma) \otimes I \quad (6.5)$$

where  $L$  is the image convolved by a Gaussian derivative kernel,  $g(\sigma)$  is a Gaussian kernel and  $\sigma$  is its standard deviation.

The normalization of a point  $p_j = \{x_j, y_j\}$  of the neighborhood with respect to the reference feature position  $p_i = \{x_i, y_i\}$  and orientation  $\theta_i$  can be written:

$$\begin{bmatrix} x_j^r \\ y_j^r \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(-\theta_i) & -\sin(-\theta_i) \\ \sin(-\theta_i) & \cos(-\theta_i) \end{bmatrix}}_{\text{orientation normalization}} \begin{bmatrix} x_j^t \\ y_j^t \end{bmatrix} \quad (6.6)$$

$$\begin{bmatrix} x_j^t \\ y_j^t \end{bmatrix} = \underbrace{\begin{bmatrix} -x_i \\ -y_i \end{bmatrix}}_{\text{translation}} + \begin{bmatrix} x_j \\ y_j \end{bmatrix} \quad (6.7)$$

**2. Local Affine Pose.** The local orientation may not be sufficient to capture stable relations in the presence of scale and viewpoint changes. To take into account these effects, a second possibility is to exploit the local affine parameters  $\vartheta_i = \{\theta, a, s_x, s_y\}$  of the reference feature to normalize a point  $p_j = \{x_j, y_j\}$  of the region:

$$\begin{bmatrix} x_j^r \\ y_j^r \end{bmatrix} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} s_x^{-1} & 0 \\ 0 & s_y^{-1} \end{bmatrix} \begin{bmatrix} 1 & a^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_j^t \\ y_j^t \end{bmatrix} \quad (6.8)$$

---

**Algorithm 13** Direction of the point density

---

- 1: Let  $p_r$  be a reference point,  $N$  the set of all image points
  - 2: and  $\mathcal{T}$  a histogram constructed on 36 bins
  - 3:  $k = 3N^{1/2}$  // Compute the number of neighbors
  - 4:  $P = N(1 \dots k)$  // Extract  $k$  nearest neighbors
  - 5: **for each** point  $p_j \in P$  **do**
  - 6:  $\tan(\alpha) = (p_r^y - p_j^y)/(p_r^x - p_j^x)$  // Compute the direction from  $p_r$  to  $p_j$
  - 7:  $\mathcal{T}(\text{round}(\alpha/10)) = \mathcal{T}(\text{round}(\alpha/10)) + 1$  // Vote for the orientation
  - 8: **end for**
  - 9:  $\theta_r = 10 \operatorname{argmax}_\alpha \mathcal{T}(\alpha)$  // Extract the most observed orientation
- 

**3. Direction of the point density.** In many situations, the local measures (affine pose or orientation) of the reference feature are not reliable enough to allow a stable normalization. This occurs typically when local features are extracted at random image locations (*e.g.* randomized grid). Uniform regions often leads to unstable gradient direction. This motivates us to propose a method to normalize points that does not rely on the structure of the intensity signal. To this end, we introduce a simple process that is designed to normalize feature locations when clusters of points can be identified. The main assumption to consider is that the direction of the point density is stable enough to be used as a canonical direction.

The process that computes the orientation of a reference point is summarized in Algorithm 13 and illustrated in Figures 6.5, 6.6 and 6.7. It starts by extracting the  $k$  nearest neighbors that will be used to compute the orientation. An empirical choice for the integer  $k$  is  $k = 3N^{1/2}$ , where  $N$  is the total number of points ( $n > 30$ ). Then for each of the  $k$  neighbors of the reference point, it computes the direction  $\alpha$  from  $p_i$  to  $p_j$ . Each observed angle is accumulated into a histogram  $\mathcal{T}$  of 36 bins (one bin every 10 degrees). Finally, the orientation is obtained by extracting the direction that receives the highest number of votes.

This method is efficiently implemented using a KD-tree data structure to reduce the computational cost of this method. This leads to an  $\mathcal{O}(\log N)$  nearest-neighbor algorithm. In contrast with signal based approaches, it is often faster since it does not require any convolutions. It is also more convenient for extracting sample orientations of high-level features.

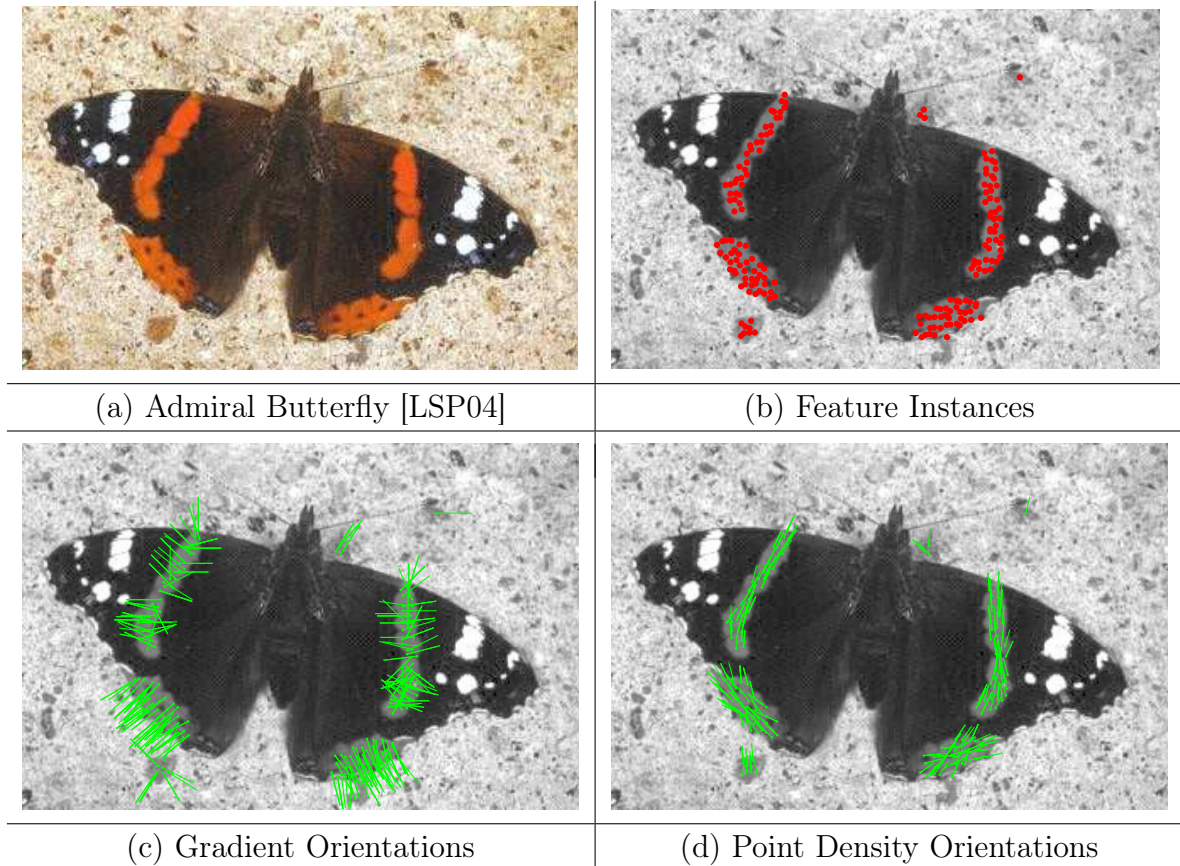


Figure 6.5: A color image of a butterfly (a) [LSP04] is processed by a detector during detection. Random patches corresponding to a given feature class are depicted by red points (b). We show in the two bottom figures, the orientations obtained by using gradient orientation (c) and the direction of the point density (d). The orientation of each feature instance is represented by a green line. In contrast with gradient-based method that leads to brittle orientations, the direction of the point density relies on the shape of the neighbor distribution and allows to extract more stable information.

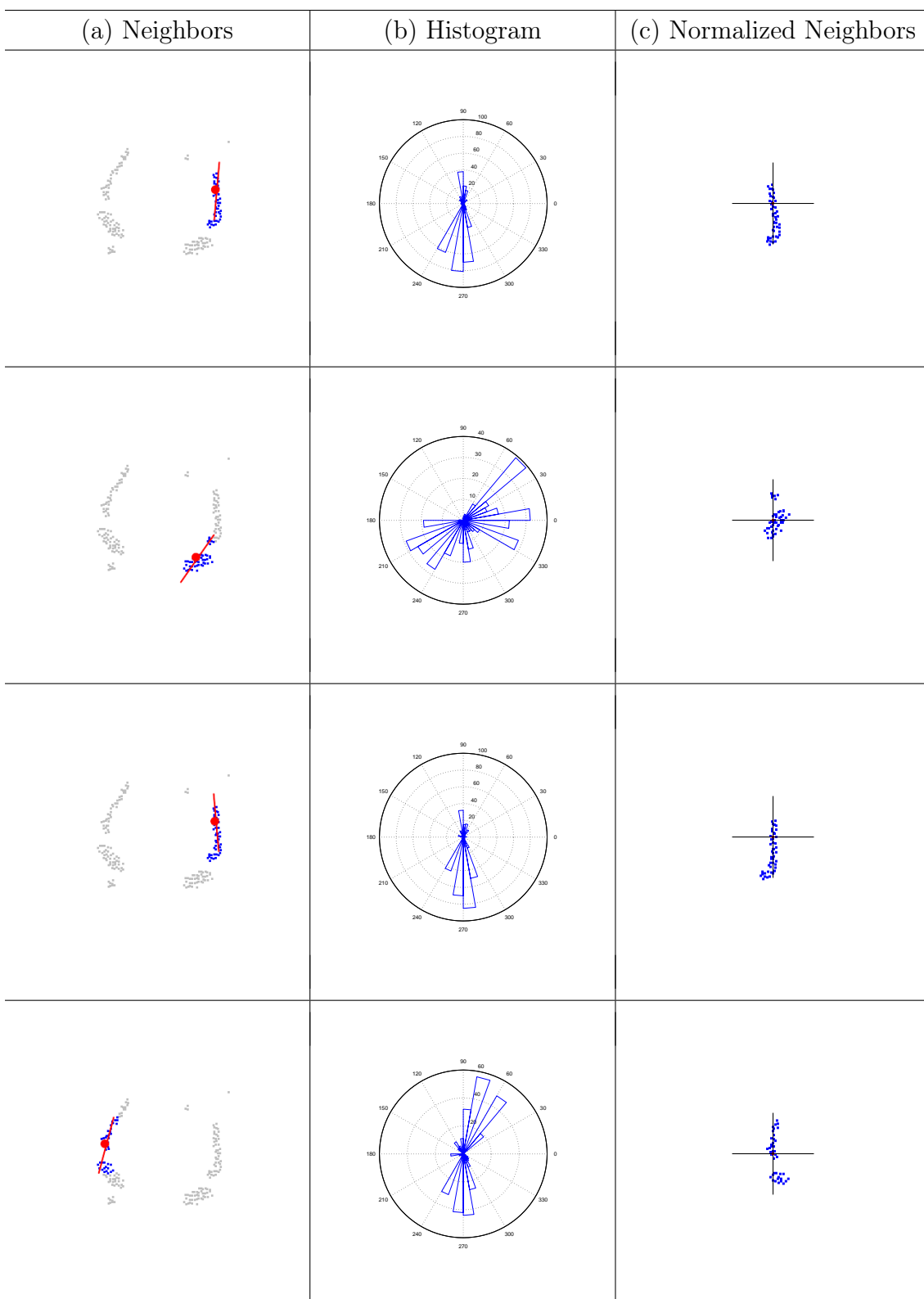


Figure 6.6: Illustration of the point density normalization for different feature instances obtained from the image shown in Figure 6.5 (b). For each line, the reference point is depicted by a red point and its orientation, which is represented by a red line, corresponds to the maximum bin in the histogram shown in the middle column...

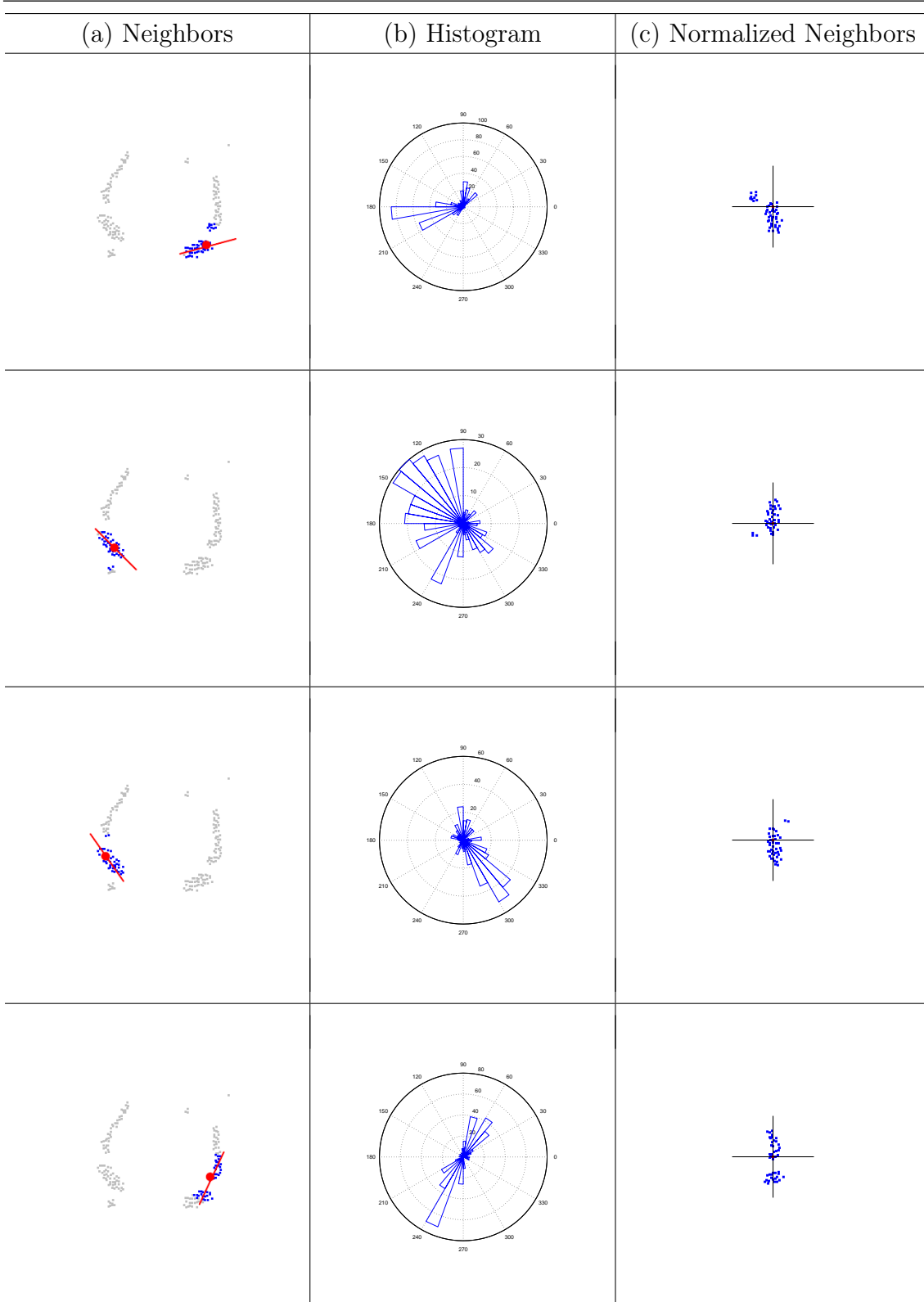


Figure 6.7: ...The normalized neighborhood is shown on the right and can be used to estimate a potential function (Figure 6.8) for this feature.

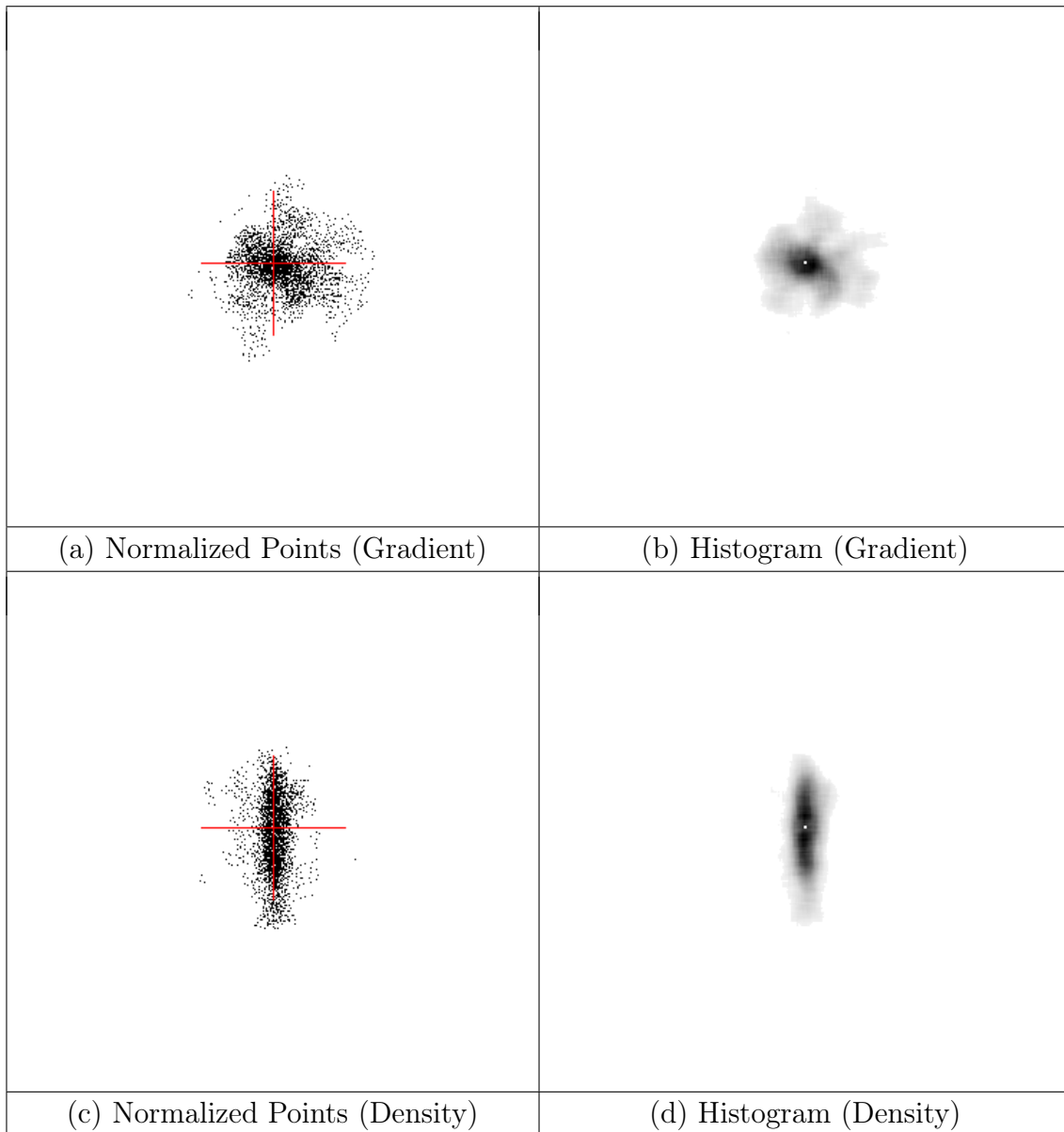


Figure 6.8: Illustration of the effect of the normalization method of the potential function of a single feature class (a feature class combined with itself). The first columns shows the set of all normalized points extracted during learning. For a better visualization, the second column illustrates a histogram constructed from the observed points (shown on the left). We can observe that the potential obtained by the gradient method does not reflect the elongated structure of the pattern. The direction of the point density clearly allows a more intuitive and sharp potential function.

### 6.4.5 Estimating Spatial Relations

Once reliable co-occurrence statistics (*i.e.* spatial relations) have been extracted from training images, our method estimates a model based on these observations. Two different models of spatial relation, parametric and nonparametric, are presented below. Parametric models are represented by a mixture of Gaussian distributions. For nonparametric relations, observations are resampled to a fixed number of samples whose variance is estimated by a kernel density estimation (KDE). The estimated geometric relations will then be used by the feature generation process in order to create new features in the graph.

#### Parametric Representation

In principle, a sample of observed spatial relations  $r \leftarrow \Lambda_{i,j}$  between two given features can be approximated by a Gaussian mixture, where each component  $k$  represents a cluster of relative positions  $\mu_k$  of one of the two features  $f_j$  with respect to the other, the *reference feature*  $f_i$ :

$$p(r; \Theta) = \sum_{k=1}^K w_k \mathcal{G}_k(r; (\mu_k, \Sigma_k)) \quad (6.9)$$

where  $\mu_k, w_k, \Sigma_k$  are respectively the mean, weight, and standard deviation of the  $k$ -th Gaussian component.

A common way to estimate these parameters is to use an Expectation-Maximization (EM) algorithm to fit the model to the observed spatial relations. To estimate the relative position between two features  $[f_i, f_j] \in \mathcal{S}$ , EM maximizes the likelihood of the observed spatial relations over the model parameters  $\Theta = (w_{1..K}; \mu_{1..K}; \Sigma_{1..K})$ . The Expectation (E) and Maximization (M) steps of each iteration of the algorithm are defined as follows:

**Step E** Compute the current expected values of the component indicators  $t_{ik}$ ,  $1 \leq i \leq n$ ,  $1 \leq k \leq K$ , where  $n$  is the number of observations in  $\Lambda_{ij}$ ,  $K$  is the number of components and  $q$  is the current iteration:

$$t_{ik}^{(q)} = \frac{\hat{w}_k^{(q)} \mathcal{G} \left( r_i; \hat{\mu}_k^{(q)}, \hat{\Sigma}_k^{(q)} \right)}{\sum_{l=1}^K \hat{w}_l^{(q)} \mathcal{G} \left( r_i; \hat{\mu}_l^{(q)}, \hat{\Sigma}_l^{(q)} \right)} \quad (6.10)$$

**Step M** Determine the value of parameters  $\Theta^{q+1}$  containing the estimates  $\hat{w}_k, \hat{\mu}_k, \hat{\Sigma}_k$

that maximize the likelihood of the data  $r$  given the  $t_{ik}$ :

$$\hat{w}_k^{(q+1)} = \frac{\sum_{i=1}^n t_{ik}^{(q)}}{n} \quad (6.11)$$

$$\hat{\mu}_k^{(q+1)} = \frac{\sum_{i=1}^n t_{ik}^{(q)} r_i}{\sum_{i=1}^n t_{ik}^{(q)}} \quad (6.12)$$

$$\hat{\Sigma}_k^{(q+1)} = \frac{\sum_{i=1}^n t_{ik}^{(q)} \left( r_i - \hat{\mu}_k^{(q+1)} \right) \left( r_i - \hat{\mu}_k^{(q+1)} \right)^T}{\sum_{i=1}^n t_{ik}^{(q)}} \quad (6.13)$$

The number of parameters  $K$  and their initial values in EM are unknown a priori. They may have a large impact on the estimation accuracy. In practice, the K-Means algorithm is used to initialize the estimation process, and the BIC criterion is used to select an appropriate number of relations  $K$ .

When the model parameters  $\Theta = (w_{1\dots K}; \mu_{1\dots K}; \Sigma_{1\dots K})$  are estimated between two features  $i$  and  $j$ , They are stored in a table  $\mathcal{S}$  at the corresponding entry  $\mathcal{S}_{i \rightarrow j}$  of the feature pair  $[f_i, f_j]$ .

### Nonparametric Representation

Rather than using a Gaussian mixture to model spatial relations, it is possible to use nonparametric distributions. Nonparametric relations are defined as sets of particles where each particle is defined by a location  $\mu \in \mathbb{R}^n$ , a weight  $w$ , a variance  $\Sigma$  and the relative orientation of the feature from the reference feature:

$$\mathcal{S}_{i \rightarrow j} = \{\mu_p, w_p, \Sigma_p\}_{\{p=1\dots P\}} \quad (6.14)$$

First, the system resamples the set of observed positions  $\Lambda$  to a more tractable number of particles  $\{\mu_p, w_p\}_{\{p=1\dots P\}}$  where  $P$  should be at least 100. Then, it computes the new variance values  $\{\Sigma_p\}_{\{p=1\dots P\}}$  by applying a KDE (Appendix A) on the set of particles.



### 6.4.6 Feature generation

During the preceding steps, the learning process has identified reliable spatial relations  $\mathcal{S}'$  between feature pairs. To incorporate these relations into the graphical model, the system generates a new hidden node  $x_n$  for each pair of spatially related features  $(x_i, x_j)$  that appears in  $\mathcal{S}'$ . The newly created node  $x_n$  corresponds to a higher-level feature and is linked to its subfeature nodes  $(x_i, x_j)$  by four conditional density functions  $\psi(x_i|x_n)$ ,  $\psi(x_n|x_i)$ ,  $\psi(x_j|x_n)$ , and  $\psi(x_n|x_j)$ . For more simplicity, pairs of conditionals are depicted by a potential in Figure 6.9.

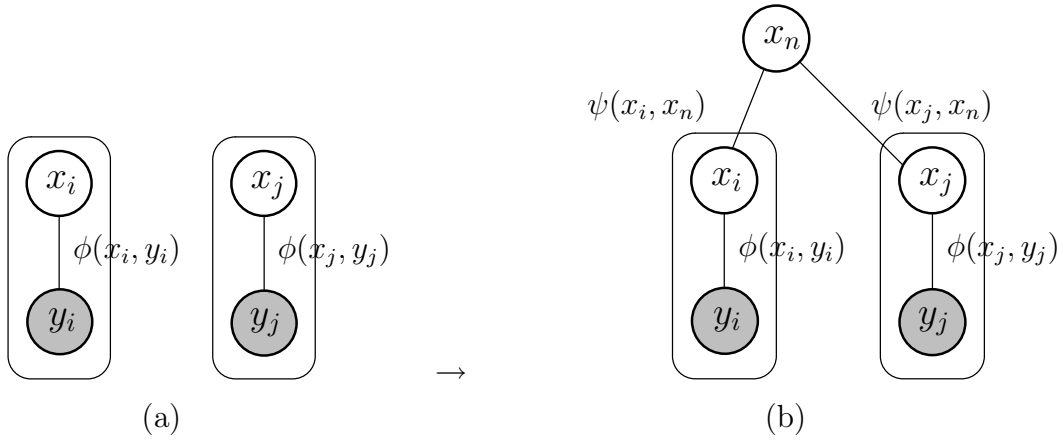


Figure 6.9: Creation of a new visual feature in the graphical model. A new hidden node  $x_n$  is connected to the two correlated features  $x_i, x_j$  by edges. Each edge is annotated by a potential function that represents a spatial relation between the new feature and its children. In our framework, each potential  $\psi(x_i, x_n)$  is decomposed in two conditionals,  $\psi(x_i|x_n)$ ,  $\psi(x_n|x_i)$ .

A conditional  $\psi(x_n|x_i)$  between two random variables is defined by means of a mapping function  $\gamma_{n,i,k}$  (Equation 6.15) that moves each sample of  $x_i$  with the  $k$ -th relative position  $\mu_{ink}$  from  $x_i$  to  $x_n$ . To ensure symmetry, we set this position  $\mu_{ink}$  to the midpoint between its subfeatures, thus to the half distance of the relative position  $\mu_{ijk} \in \mathcal{S}'_{ij}$  of feature  $x_j$  from  $x_i$ ,  $\mu_{ink} = \mu_{ijk}/2$ .

$$\gamma_{i,n,k}(x_i) = \mu_i + \vartheta_i \mu_{ijk}/2 \quad (6.15)$$

where  $\mu_i, \vartheta_i$  are the position and pose of a feature occurrence of  $x_i$  and  $\mu_{ijk}$  is the  $k$ -th relative position from  $i$  to  $j$  that has been estimated with the EM. The other conditionals  $\psi(x_i|x_n)$ ,  $\psi(x_j|x_n)$ ,  $\psi(x_n|x_j)$  are defined similarly.

In parallel, we also keep trace of relative pose between the correlated pair and the newly created feature. The relative pose of the new feature  $x_n$  is set to a

canonical orientation in the normalized neighborhood of the first feature  $i$ . For each component of the spatial relation, it is set to the orthogonal direction from feature  $i$  to feature  $j$ :

$$\theta_k^R = \theta_k - (\pi/2) \quad (6.16)$$

When newly created features  $x_n$  are expected to be observable, an observation node  $y_n$  can be associated and added into the vertex set of graph (Figure 6.10). The hidden node is linked to the observation by adding an observation potential  $\phi(x_n, y_n)$ .

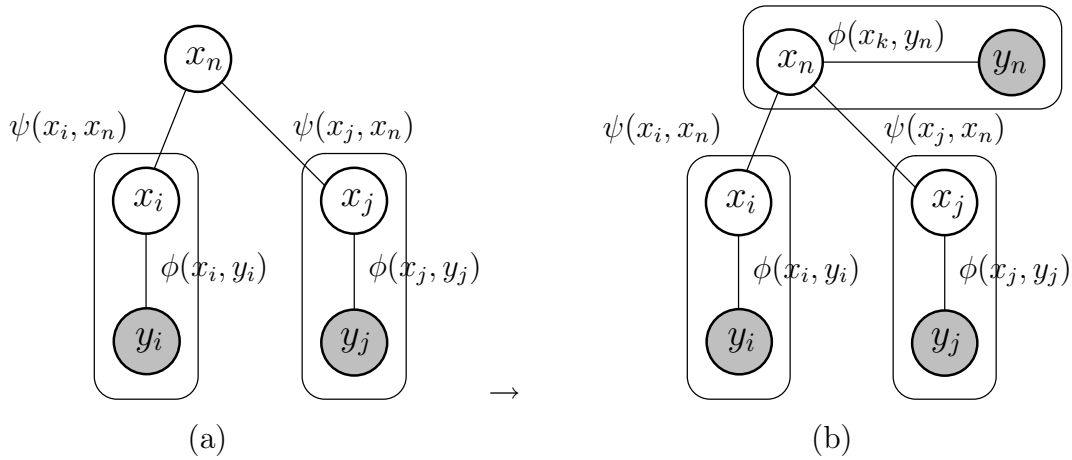


Figure 6.10: Creation of the observation part of a visual feature in the graphical model. A new observable node  $y_n$  is connected to the new hidden node  $x_n$  via an observation potential  $\phi(x_n, y_n)$ .

For observable features, each new feature  $x_n$  is associated to the shape  $\mathcal{X}_{ij}$  and appearance  $\mathcal{A}_{ij}$  of the feature combination. These parameters are explained below.

### 6.4.7 Adaptive Patch Features

A high-level feature may not only be defined by a spatial configuration  $\mathcal{S}_{ij}$  of lower-level features, but also by an appearance  $\mathcal{A}_{ij}$  over a region of shape  $\mathcal{X}_{ij}$ . In this section, we propose an efficient method to estimate these parameters from a set of previously extracted positions  $\Phi_{ij} \in \Phi$  of the feature pair  $[f_i, f_j]$ .

In general, the scale at which the appearance should be extracted is unknown a priori. A naive approach would be to derive it from the distance between its parts. We consider this as an initial reference scale  $s_{init}$ ; however, the optimal size of this region critically depends on the class and on the type of its neighborhood (*e.g.*

region, edge, corner, ...). Too small or too large regions may result in information loss and inaccurate models. Therefore, it is desirable to estimate a specific spatial extent for each novel feature to compute its appearance. To do so, we use two scale factors  $s_x, s_y$  relative to the initial scale  $s_{init}$ , one for each dimension of the neighborhood, normalized with respect to the gradient orientation.  $N_s$  pairs of scale factors are uniformly extracted from  $[0.1, 2.0[$ .

The optimal relative region size  $\mathcal{X}_{ij} \leftarrow [s_x, s_y]$  is selected by applying a minimum variance analysis method. It starts by extracting appearance vectors at the detected locations  $\Phi_{ij}$  of the combination in the training images for the set of scale factors  $[s_x, s_y]_{N_s}$ .

For each scale pair  $[s_x, s_y]_j$ , a trimmed mean  $\mathcal{M} \in \mathbb{R}^N$  is computed from the extracted appearance vectors:

$$\forall \mathcal{M}^i \in \mathcal{M}, \mathcal{M}^i = \frac{\sum_{(th_1 < a_i < th_2)} a_i}{\sum_{(th_1 < a_i < th_2)} 1} \quad (6.17)$$

It is used to compute a vector of dimension-wise variances  $\sigma \in \mathbb{R}^N$ . Then we select the scale factor pair  $[s_x, s_y]_{\min}$  with the minimum sum of variances over all  $N$  dimensions:

$$[s_x, s_y]_{\min} = \operatorname{argmin}_{[s_x, s_y]_j \in N_s} \sum_{i=0}^N \sigma_i^{[s_x, s_y]_j} \quad (6.18)$$

Here,  $\sigma^{[s_x, s_y]_j}$  is the variance vector corresponding to a relative window size of  $[s_x, s_y]_j$ . This optimal scale selection procedure is illustrated in Figure 6.11. The shape model of the newly created compound feature class is then set to  $\mathcal{X}_{ij} = [s_x, s_y]_{\min}$ , and the appearance model  $\mathcal{A}_{ij}$  to the mean appearance vector  $\mu^{\mathcal{A}} = \mathcal{M}^{[s_x, s_y]_{\min}}$  and its corresponding variance  $\Sigma^{\mathcal{A}} = \sigma^{[s_x, s_y]_{\min}}$ . During our experiments, appearance vectors are represented as color pixel values in the HSV colorspace. Note that any other description method (such as SIFT, ...) can be used to represent them.

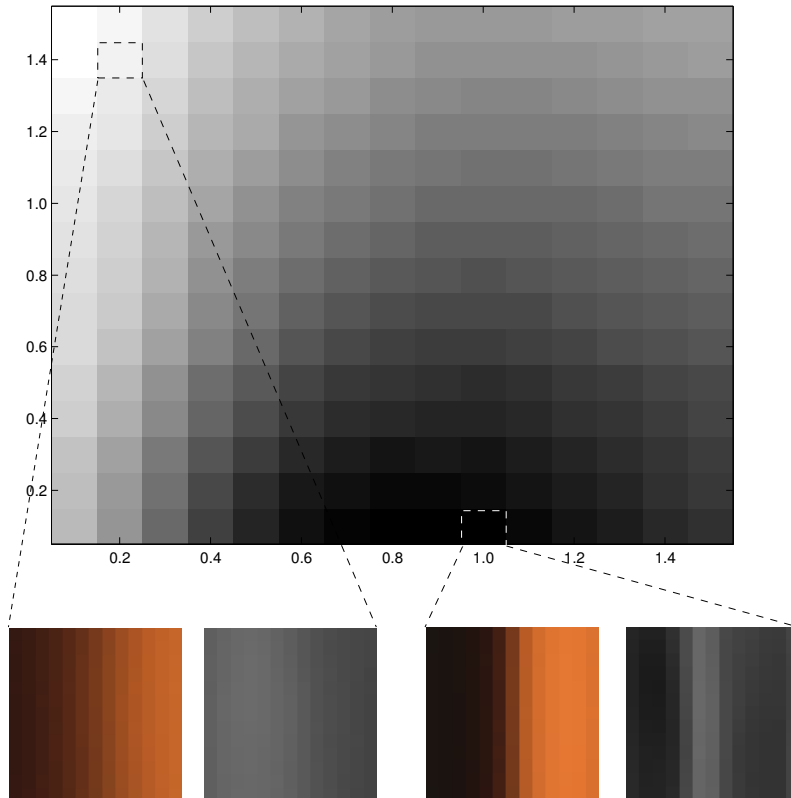


Figure 6.11: Illustration of the adaptive selection procedure. The gray value at each bin is proportional to the sum of dimension-wise variances for a pair of scale factors  $[s_x, s_y]$ . During the extraction process, each local patch is normalized in the local gradient direction computed at scale  $s = (s_y + s_x)/2$  and resampled into a patch of  $13 \times 13$  pixels. The trimmed-mean appearance and variance vectors corresponding to the optimal relative scale pair are shown on the bottom right.

## 6.5 Discriminative Learning

Once a graphical model has been learned for each object class, they can be used for detection in new images. Since each model has been constructed from co-occurrence statistics and without using discriminant information, some features in the graphical model are not useful to differentiate objects. Discriminant features might be spread over different levels in the graph. In this section, we present a method to build a classifier from the feature hierarchies. The general idea is to construct a multi-class Support Vector Machine classifier (SVM) [BGV92] from the maximum activation of features obtained from detection (NBP) (as shown in Figure 6.12).

As we said, many features in our representation are useless for an object recognition task. It has been shown that SVM can indeed suffer in high dimensional spaces where many features are irrelevant [WMC<sup>+</sup>00]. A way to bypass this naturally occurring problem is to first perform a feature selection to eliminate useless features. Then a robust SVM classifier can be learned from the response of these selected features.

In the next subsections, we explain how SVM can be applied to a graphical model (Section 6.5.1) and we present a way to precede this operation by a feature filtering step [WMC<sup>+</sup>00, LCS06, CL06].

### 6.5.1 SVM for Graphical Models

The Support Vector Machine (SVM) algorithm [BGV92] is a machine learning algorithms commonly used for many complex classification problems. In the common case, SVM algorithm finds a hyperplane that maximizes the margin of separation in the feature space between classes. This hyperplane is defined by a subset of examples which trace the boundary between classes.

During training, the classifier uses a set of input vectors of features together with their class label. The main issue is to convert our graphical models  $\mathcal{G}_q \in \mathcal{G}$  to a single input vector  $\mathcal{Z}$  for the classifier. To this end, we consider each node  $x_i \in \mathcal{G}_q$  of the graphical model of an object  $q$  as an element  $e_i$  of the SVM input vector. The value of the element  $e_i$  will correspond to the maximum activation of the node  $x_i$  for the current image. The maximum value is obtained by evaluating the kernel density at each location in the image. This process is repeated for each object  $q$ . Finally, we concatenate the vectors  $\mathcal{Z}_q$  of each object class into a single vector  $\mathcal{Z}$

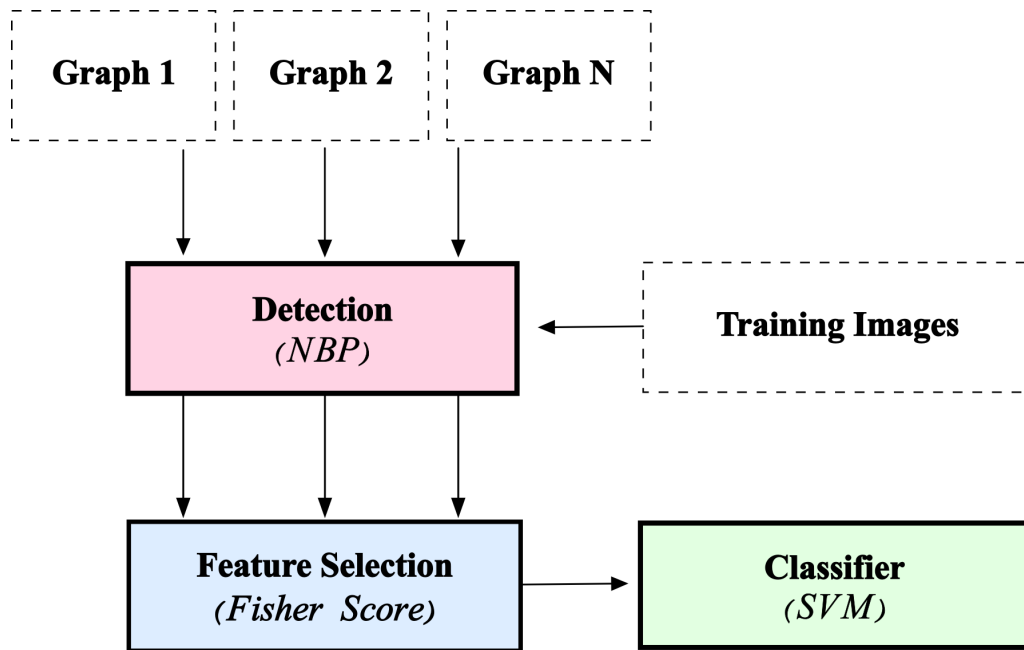


Figure 6.12: Illustration of the learning of a multi-class SVM classifier. Detection (NBP) is performed on each graphical model for the training images. Then a feature selection method based on the Fisher Score is applied. For each image, the maximum responses of the selected features are merged into a single vector that is used to train a SVM classifier.

which corresponds to the input vectors of the SVM classifier:

$$\mathcal{Z} = \{\mathcal{Z}_1 \mathcal{Z}_2 \dots \mathcal{Z}_q\} \quad (6.19)$$

$$\Leftrightarrow \mathcal{Z} = \{\{e_{i=1\dots N^1}\}_1 \{e_{i=1\dots N^2}\}_2 \dots \{e_{i=1\dots N^q}\}_q\} \quad (6.20)$$

where each  $N^q$  corresponds to the number of nodes in the graphical model of object  $q$ , and the dimensionality of the vector  $\mathcal{Z}$  is  $\sum_{j=1}^q N^j$ . During recognition, vectors  $\mathcal{Z}_q$  are obtained by processing the input image in each graphical model  $\mathcal{G}_q$ , thus obtaining  $q$  vectors of activation  $\{e_{i=1\dots n^q}\}$ .

### 6.5.2 Combining SVM and Fisher score

It has been shown that SVM performance can degrade in high-dimensional spaces with many irrelevant features [WMC<sup>+</sup>00]. One way to bypass this problem is to perform feature selection [WMC<sup>+</sup>00, LCS06, CL06] to eliminate useless features. We employ a conventional feature selection procedure (Algorithm 14) based on the Fisher score. It computes the recognition rate (on the training set) for a set of Fisher score thresholds  $\mathcal{T}_i \in \mathcal{T}$ . Then it selects the threshold  $\mathcal{T}_i$  with the best validation rate.

The Fisher score measures the discriminatory power between two sets of real numbers. Given training vectors  $x_{k=1,\dots,m}$ , if the number of positive and negative instances are  $n_p$  and  $n_n$ , respectively, then the  $F$ -score of the  $i$ -th feature can be expressed as

$$F(i) = \left| \frac{\mu_i^+ - \mu_i^-}{\sigma_i^+ + \sigma_i^-} \right| \quad (6.21)$$

where  $\mu_i^\pm$  is the mean value for the  $i$ -th feature in the positive and negative classes, and  $\sigma_i^\pm$  is the standard deviation.

## 6.6 Discussion

In this chapter, we have introduced a method to learn visual feature hierarchies. This construction is incremental and based on the analysis of co-occurrences between feature pairs. The learning process is basically task-independent but can naturally be used to perform object recognition by using a learning a discriminative layer on the top of the hierarchies.

The main contributions of this chapter are the following;

---

**Algorithm 14** Fisher score for feature selection

---

```

1: Calculate  $F$ -score of every feature  $\mathcal{Z}_i \in \mathcal{Z}$ 
2:  $\mathcal{T}_{0...N} \leftarrow$  thresholds on  $F$ -scores
3: for each threshold  $\mathcal{T}_j \in \mathcal{T}$  do
4:   for each  $\mathcal{Z}_i \in \mathcal{Z}$  do
5:     if  $F\text{-score}(\mathcal{Z}_i) < \mathcal{T}_j$  then
6:       remove feature  $\mathcal{Z}_i$ 
7:     end if
8:   end for
9:    $\{Train, Test\} \leftarrow$  randomly split(trainingSet)
10:  Train a SVM classifier on Train
11:   $\mathcal{R}_j \leftarrow$  Calculate the prediction rate on Test
12: end for
13:  $\mathcal{T}_s \leftarrow$  Select threshold  $\mathcal{T}_j \in \mathcal{T}$  with best rate  $\mathcal{R}_j$ .
14: for each  $F\text{-score}(f_i) < \mathcal{T}_s$  do
15:   remove features  $f_i$ 
16: end for

```

---

1. an incremental statistical method to learn high-level visual features by the analysis of feature co-occurrences,
2. a new kind of adaptive patch feature whose width and height are automatically determined,
3. a new technique to normalize randomly extracted image patches,
4. a way to learn discriminative models from previously learned visual feature hierarchies.

However, there are also some assumptions in this model that we should mention. First, the use of co-occurrence analysis assumes that it is possible to find correlated feature pairs in the image and that they are useful in order to recognize an object. If the visual similarities between training images cannot be translated in terms of feature co-occurrences, our method will fail to learn object models. Therefore, the method relies on the robustness of local feature detectors.

Another assumption we made is that relatively rigid spatial relations can be identified. Some object classes, however, have a highly flexible structure. To be



able to represent these classes, more complex models of spatial relation would be required.

Since our method is based on statistical relevance, we also assume to have a reasonable number of training images (more than 10).



# Experimental Evaluation

---

Current approaches to object recognition are typically dedicated to a task performed in specific conditions and do not generalize well. In contrast with these approaches, the hierarchical model proposed in this thesis has been thought to be generic enough to be applied in different contexts. In this chapter, we investigate its behavior across a variety of object recognition datasets (Section 7.1). These experimental evaluations are organized around three different tasks:

- The first set of experiments presented in Section 7.2 focuses on the recognition of specific objects in stable imaging conditions. The models are trained around the frontal view of each of object. Various tests of robustness are made to measure the viewpoint invariance and the robustness to clutter and occlusions.
- In Section 7.3 the system is generalized to learn object models that cover all the views of the object. Multiple views are embedded within the same graph. An additional difficulty of these experiments is that images are taken under real conditions (*i.e.* with large photometric and affine variations).
- Finally, more challenging datasets are used in Section 7.4 to evaluate our hierarchies on object classes. These images contain a large amount of background clutter, rotation, scale variation, natural lighting and compression artifacts as well as multiple instances.

For each experiment, the precise protocol is first presented and followed by a discussion of the results. Most often the dataset is split into two separate sets. The model is typically trained on the first set and tested on the second. But before presenting the experiments, we describe the datasets and their specific properties.

## 7.1 Datasets

In this section, we detail the datasets that have been used in the evaluation of our feature hierarchies. In general, the choice of the dataset used to evaluate an object recognition framework is critically linked to the method itself. To demonstrate the specific properties of their approach, researchers often create their own image dataset to evaluate their method. Therefore these are often biased in favor of their work. For instance, the K-fan [CFH05] and the Constellation model [FPZ03] were evaluated on single viewpoint datasets of object classes; such as rear view of cars [PU01], side view of planes or motorbike [us01] and front views of faces [Web99]. Object recognition with local affine frames [LSP04] was performed on butterflies because the geometry of a butterfly is locally planar for each wing. Moreover, the species identity of a butterfly is determined by a basically stable geometric wing pattern. In the same vein, bag-of-features approaches [WAC<sup>+</sup>04] are often evaluated on databases that can be distinguished without using spatial relations between local features.

In contrast with those works and similarly to more generic attempts to object recognition [OFPA04], we propose to evaluate our framework on a few standard object recognition datasets. Each of these datasets has its own properties, advantages and disadvantages, but none of them is especially tuned to our method. In the subsequent sections, we describe these datasets. We particularly emphasize the following variations, as mentioned in [OFPA04]:

**Intra-class variability:** How much the appearance of the different object instances varies from image to image.

**Occlusion:** The degree to which parts of the objects are occluded in different images. If the visible portion of the object is small, clearly the object becomes hard to detect.

**Viewpoint variation:** Are the different instances of same pose and aspect?

**Background clutter:** What portion of the image does the object typically occupy? If this is large then recognition is likely to be easier than if it is small.

**Quantity of training data:** How much data is available to train from.

**Multiple instances:** Finding many (possibly overlapping) instances is harder than finding a single occurrence.

### 7.1.1 COIL-100

Columbia Object Image Library (COIL-100) [NNM96]<sup>1</sup> is a database of color images of 100 objects (Figure 7.1). During acquisition, each object was placed on a turntable and images were captured at pose intervals of 5 degrees all around the object. Therefore for a given object, 72 colour images are available. In the classical scenario, training is performed on 18 views equally distributed to cover the object (Figure 7.2). Recognition is tested on the remaining 54 images.

The dataset contains viewpoint changes that inevitably induce large appearance variations. Depending on the viewpoint, some objects may look very similar. Despite this difficulty, the recognition conditions of this dataset are considered as ideal; the color images possess a fair resolution (128x128), a large number of training images are available and there is no background clutter nor occlusion. Moreover, since it is a specific instance recognition dataset, there is no intra-class variability. The main difficulty is that the same object is described by multiple views.

Originally, the images of the COIL database have been used as a benchmark for testing an appearance-based recognition system [MN95b] based on the notion of parametric eigenspace. It was later exploited to evaluate many multiple-view object recognition systems; such as Support Vector Machines (SVM) [PV98], local affine-frames [OM02b], hierarchies of complex cells [WK03], decision trees [MGPW05b], self-organized model graph [WvdMW06], etc.

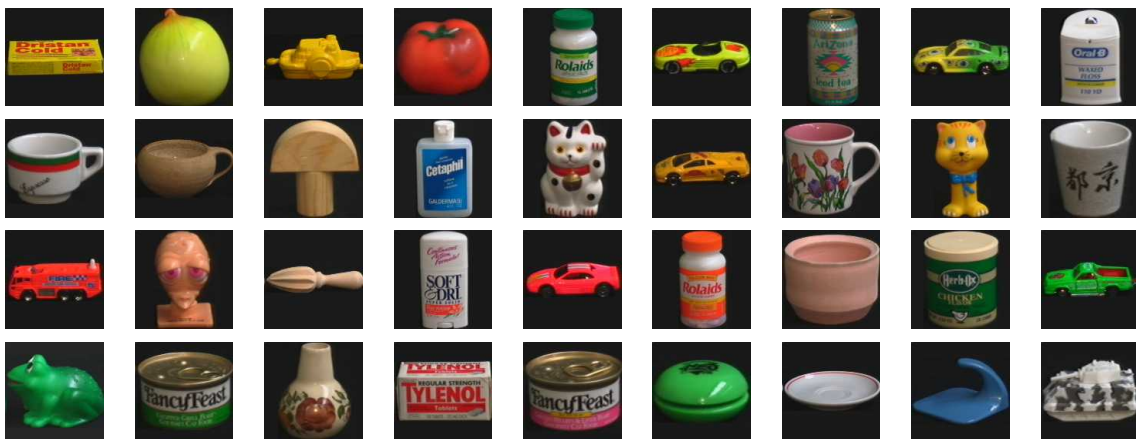


Figure 7.1: Illustration of the first 36 objects of the COIL-100 dataset [NNM96].

<sup>1</sup>available at <http://www.cs.columbia.edu/CAVE/>



Figure 7.2: COIL-100 [NNM96]: 18 training views of an object.

### 7.1.2 Ponce Group’s Object Recognition Database

Ponce Group’s Object Recognition Database [RLSP06]<sup>2</sup> consists of 8 specific objects represented by a total of 161 training images. Depending on the object, the number of available training image varies between 16 and 29. Similarly to the COIL-100 image library, the training set covers different poses of the object in a uniform background. However, the training views are not taken on a turntable but rather captured manually at different arbitrary poses around the object.

The testing set is composed of 51 cluttered images. Each scene comprises at least 1 and at most 6 of the training objects but may contain many other distracting objects. The images were taken in real conditions and contain rescaled, rotated, partially occluded and differently illuminated instances of the objects. A particularity of these color images is the high resolution (between 1.2 Mpix (1280x960) and 3.7 Mpix (2200x1700)).

This database has been introduced for the evaluation of a 3D object recognition system [RLSP06]. Three-dimensional models were constructed using affine-invariant patches and multi-view spatial constraints between them. It has also been used to evaluate other object recognition frameworks [PL00, MH03, MMP04, FTVG06].

### 7.1.3 Butterflies

In contrast with tasks involving the recognition of specific object instances such as COIL-100 and Ponce Group’s image set, the Butterflies dataset [LSP04] is designed to evaluate the ability of the system to recognize classes of object. When two different butterflies are said to belong to the same class, it signifies that they share some visual similarities. Seven classes of butterflies are distributed among the 619 images of the dataset. Each training image only contains one of the seven classes but several instances of a given class may occur.

<sup>2</sup>available at [http://www-cvr.ai.uiuc.edu/ponce\\_grp/data/](http://www-cvr.ai.uiuc.edu/ponce_grp/data/)

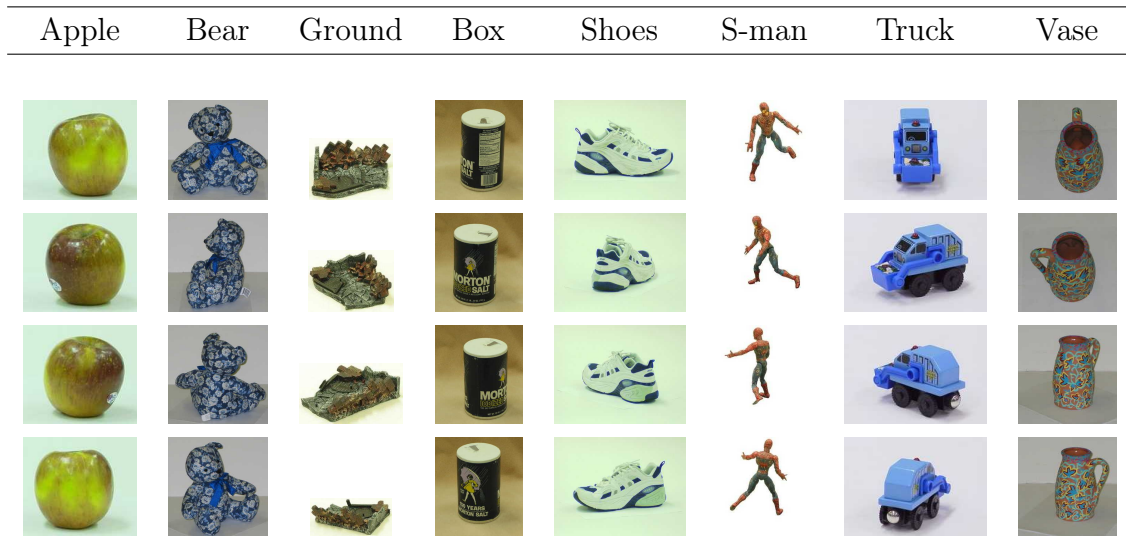


Figure 7.3: Ponce Group’s Object Recognition Database [RLSP06]: four training images are shown for each object. To avoid any background influence during learning, objects were automatically segmented.

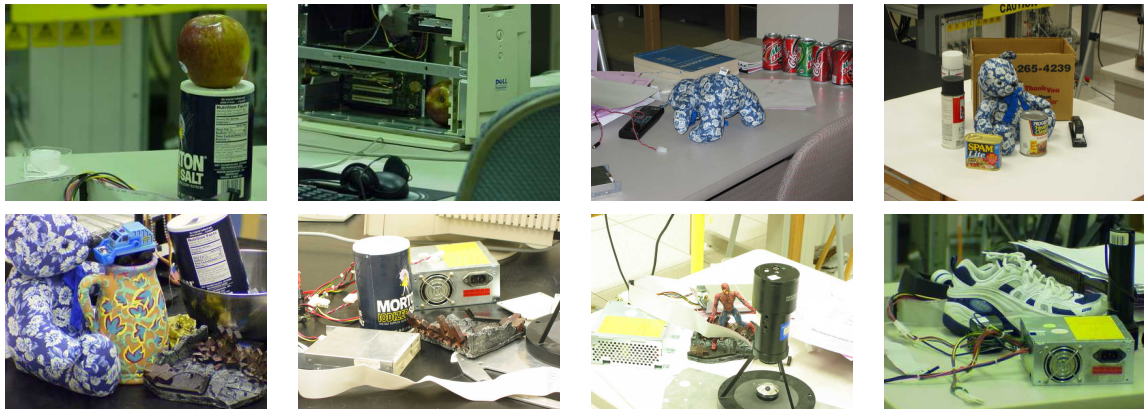


Figure 7.4: Ponce Group’s Object Recognition Database [RLSP06]: eight test images containing different objects. The apple can be observed in the first two images. The first one is very similar to the training instances whereas the second one is clearly more challenging to find, even for the human eye. Only a small portion of the object is visible.

A specificity of the Butterflies dataset is that the images were acquired from the Internet and are thus extremely diverse in terms of resolution and quality. Indeed, a wide variety of artifacts (blur, lack of focus, resampling, compression) can be noticed in these natural images (Figure 7.5).

In previous datasets, learning was facilitated by the use of training images containing objects on a uniform background. This enabled the use of segmented object for learning. Here, training images have the same difficulties as testing images. They contain scaled, rotated instance of the object in a cluttered scene. The intra-class variability is another factor that the system has to deal with. Even if each class is roughly defined by a wing pattern, some of them may present a high intra-class variability: two images of the same class may have different appearances.

As was mentioned by S. LAZEBNIK [LSP04], the creator of this database, butterfly recognition was beyond the capabilities of many recognition systems such as the constellation model. Recent discussions with the author of a high-performance image classifier, R. MARÉE [Mar05], confirmed the special level of difficulty of this task. The main problem is that the recognition of butterflies requires a large number of parts to be adequately represented, while the clutter is measured by hundreds or even thousands of regions. Moreover, the levels of invariance, in terms of translation and scale, exhibited by existing algorithms are clearly insufficient for recognizing butterflies, which can and do appear at a wide range of scales and orientations. Finally, a geometry-free approach (*i.e.* bag-of-features) does not generally give competitive results because the background in the training images may be wrongly discriminant and would mislead the training of the classifier.

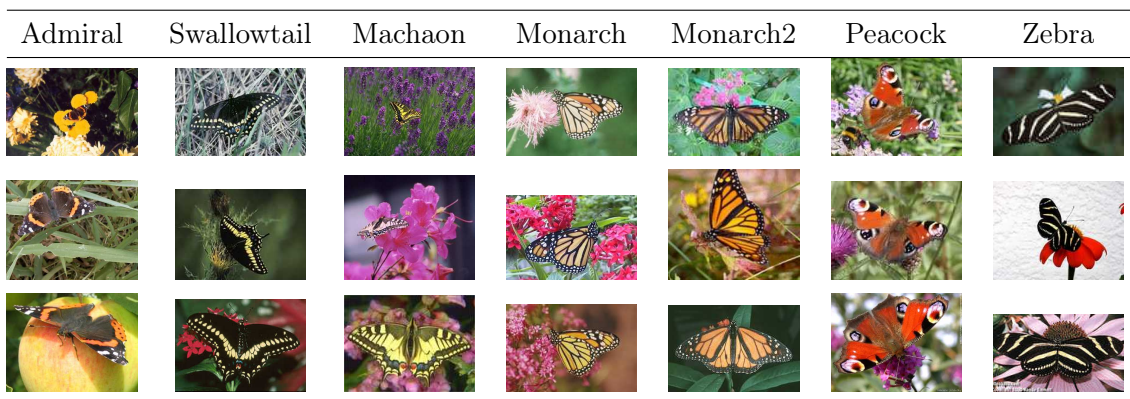


Figure 7.5: The butterfly dataset presented by LAZEBNIK *et al* [LSP04]. Three samples of each class are shown in each column. Clutter, large viewpoint changes and intra-class variability make this recognition task very challenging.



### 7.1.4 Soccer

The most recent dataset used in our experiments is named Soccer and has been introduced by VAN DE WEIJER [VdWS06]. This dataset comprises seven soccer teams and contains 280 images collected from the Internet. For each team, the image set is divided into 25 training and 15 testing images. The conditions of this dataset are very similar to the Butterflies dataset in the sense that images have very different qualities due to the real acquisition conditions. Therefore the lighting, pose variation, scale, occlusions and clutter are common among these images. These effects can be observed in Figure 7.6 where three images for each class are illustrated.

In addition to the variations in acquisition conditions, the local planarity of butterflies does not hold for a soccer player. The variation of the appearance is more challenging and the interest points are rare because of the low resolution of images. In contrast with previous datasets where different objects could be differentiated using grayscale features, the use of color information is now a necessary requirement to recognize the team of a soccer player.

For experiments, each image is assigned to a given team, however an extra difficulty is added by allowing players of various other teams (that are not from the 7 classes) to appear in the image. Players from other teams may have colors similar to the learned teams and therefore complexify the recognition process.



Figure 7.6: The Soccer dataset [VdWS06]. Three samples of each class are shown in each column. Most of teams could not be differentiated without the use of color features.

## 7.2 View-Specific Object Recognition

In this section, we evaluate our feature hierarchies on the task of recognizing given objects around a specific viewpoint. Experiments are conducted on a subset of the Columbia University Object Image Library (COIL-100) [NNM96]. The purpose of these basic tests is twofold. First, it is to demonstrate the ability of the system to learn accurate object models from co-occurrence statistics. Second, it is to evaluate these models through the inference process which locates the object instances in previously unseen images.

After introducing the experimental protocol in Section 7.2.1 and presenting the parameters of the model in Section 7.2.2, we focus in Section 7.2.3 on evaluating the degree of invariance the learned models and the convergence of the beliefs (posterior marginals) during inference. In addition, the inference mechanism is illustrated by a few didactic examples where we demonstrate the robustness of the models to clutter and occlusions.

### 7.2.1 Experimental Protocol

Our experimental protocol consists of two distinct steps: learning and detection. First, objects models are learned separately on a set of training images, then the models are used for detection in previously unseen images.

**Learning** is performed by using our co-occurrence learning method that is presented under the Algorithm 11 in Chapter 6. Learning is weakly supervised (Section 6.1) in the sense that the system exploits only the object labelling information. A hierarchy is composed separately for each object. To do so, the system utilizes 5 views, spaced by 10 degrees, around both sides of the frontal pose of the object. Here, the learning process aims at constructing a model that is tuned for a given view or aspect of the object.

**Detection** on a new image is achieved by first extracting local features in images using the Harris interest point detector at a fixed scale. Then Nonparametric Belief Propagation (NBP) (Chapter 5) is applied for inference on previously learned hierarchies.

## 7.2.2 Parameters and Implementation

In order to avoid an excessive growth of the graph due to the feature combinatorics, we only keep the most salient spatial relations between features. These are selected by considering closest relations among the strongest correlated feature pairs. Spatial relations are estimated in a two dimensional neighborhood with a relative orientation. Note that in these models, no appearance model is associated to high-level features. Therefore, high-level features are considered as hidden nodes in the graphical model.

To facilitate the interpretation of our models, the composition of new features was constrained to limit the number of levels to five and to obtain only one node at the top level of the graph. Finally, a graph pruning step was performed to eliminate the features that are not descendants of the top level node.

For these experiments, the major portion of the application code consists in a Matlab implementation. The application also relies on several other external tools. For instance, in order to extract local features from images, we implemented Harris interest point detector [HS88]. These features are detected at a single pre-defined scale ( $\sigma = 3$ ) and their appearance is described by a descriptors comprising 13x13x3 pixel values in the RGB colorspace. Rotational invariance is obtained by normalizing each region with the dominant gradient direction of the gray-level intensity.

The K-Means algorithm is used to produce a set of  $K$  low-level feature classes. The number  $K$  of classes (between 16 and 60 in our experiments) is selected according to the BIC criterion [Sch78]. K-Means clustering is performed using an open-source C++ implementation of an efficient version of the algorithm [KMN<sup>+</sup>02, KMN<sup>+</sup>04]<sup>3</sup>.

For inference, Nonparametric Belief Propagation (NBP) has been implemented using the KDE (Kernel Density Estimation) toolbox<sup>4</sup> for the efficient computation of message products. Gibbs sampling was used to keep a constant number of samples during the computation of the message products. Note that for detection, each message is sampled with 300 values.

---

<sup>3</sup><http://www.cs.umd.edu/~mount/Projects/KMeans/>

<sup>4</sup><http://ssg.mit.edu/~ihler/code/>

### 7.2.3 Evaluation

In the first series of tests presented in Figures 7.7 and 7.8, the invariance degree of the learned object models is evaluated by running the detection process on a set of images differing in viewing angle by increments of 5 degrees. The graph presented in Figure 7.7 illustrates the viewpoint invariance of five object models of COIL-100. The models responded maximally around the training views. We observe that the response quickly falls at  $\pm 40$  degrees. This is caused by the loss of pertinent features in the view. The models obtain an average viewpoint invariance over 80 degrees. These results are remarkable considering the fact that we did not use affine-invariant features at the leaf level.

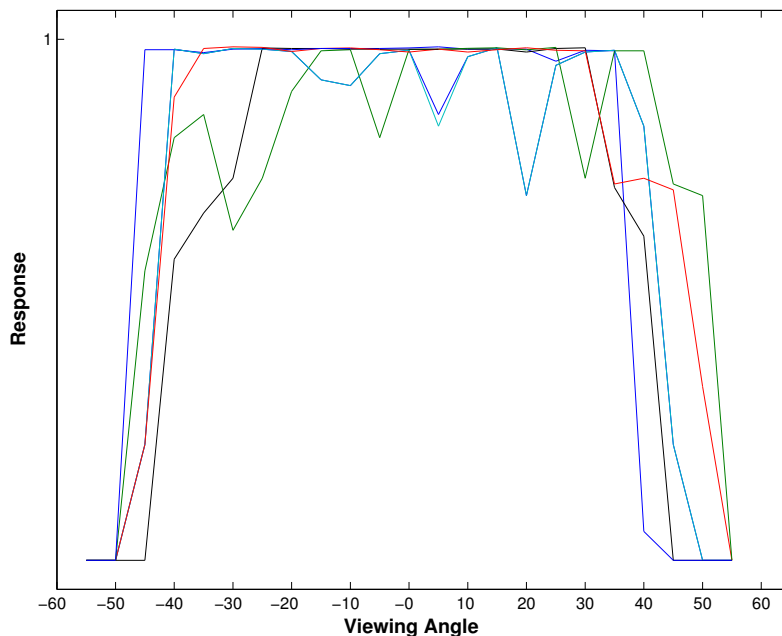


Figure 7.7: Maximum response of five object models on a series of images differing in viewing angle. The maximum response is obtained by evaluating the kernel of the top level feature and extracting the maximum value.

The second test, shown in Figure 7.9, demonstrates the convergence of the detection process using NBP. This is done by measuring the standard deviation of the sample distribution. This evaluation is repeated across 23 message-passing iterations. We observe convergence to the optimal solution in less than 7 iterations. In general, the number of iterations required for convergence depends on the number of nodes and levels in the graph.

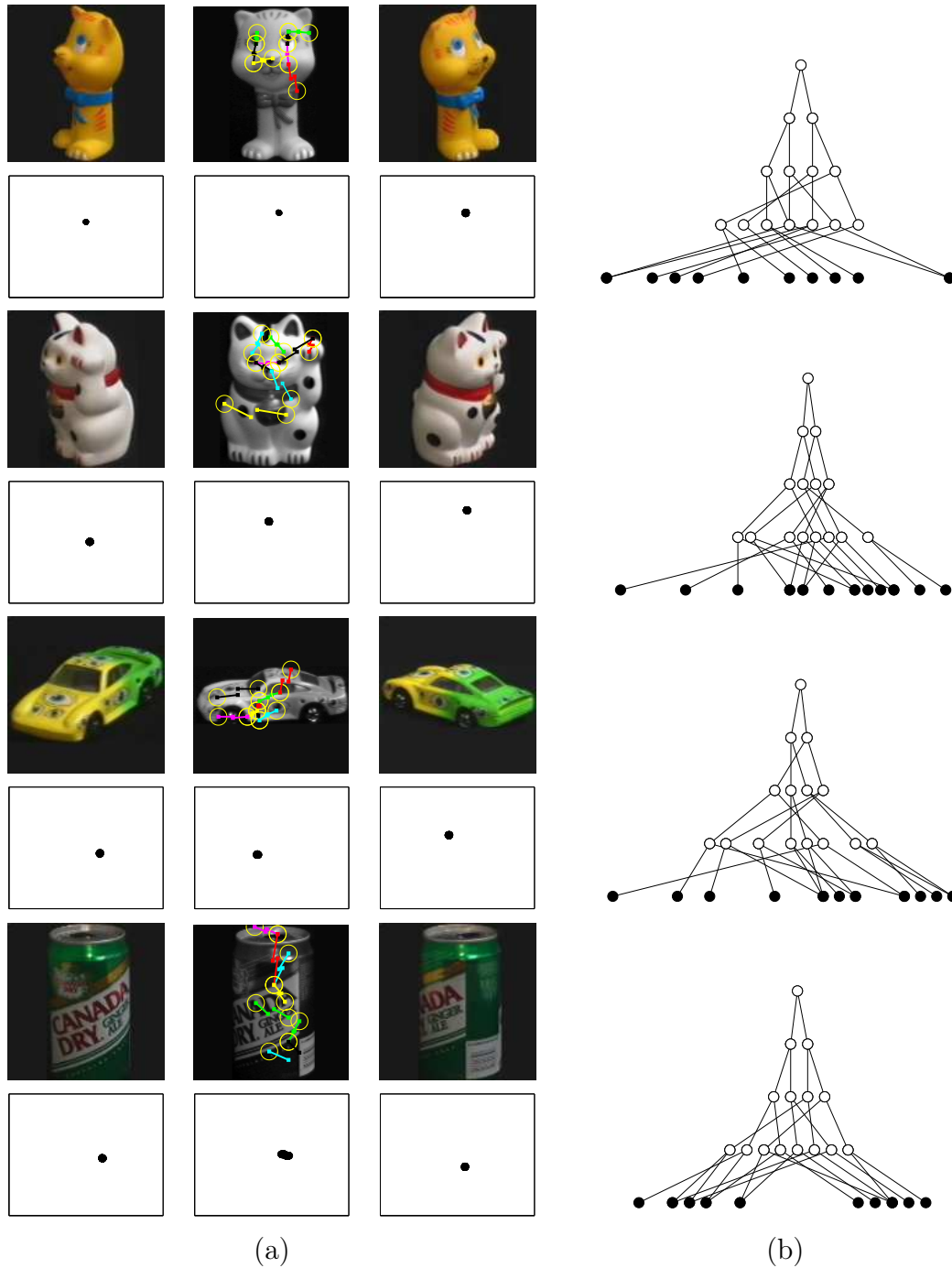


Figure 7.8: Detection results on a series of COIL-100 objects (a). The geometrical models between low-level features are shown in the top center images. The learned graphical models are illustrated on the right (b). Each circle corresponds to a primitive and is associated to another one to create a new compound feature. On the bottom of each image, we illustrate the detection results of our models with NBP after six iterations.

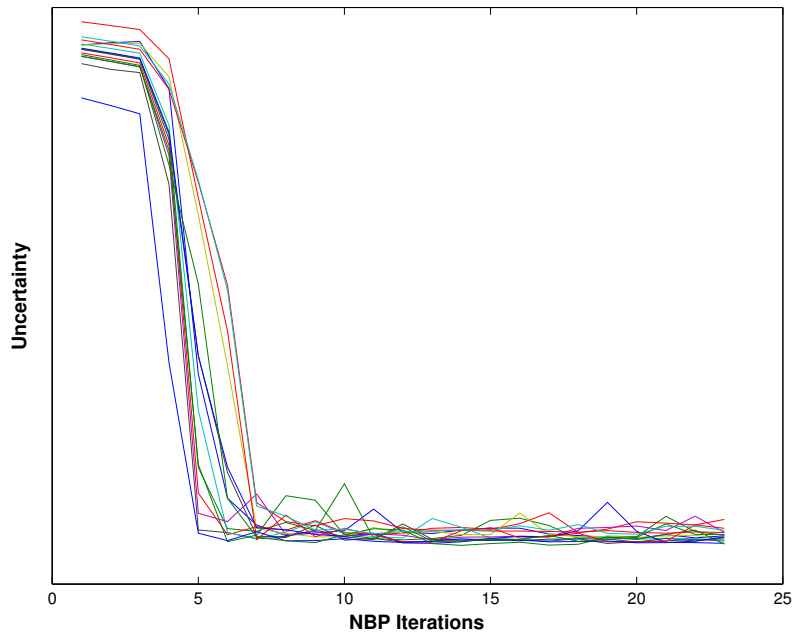


Figure 7.9: Convergence of NBP to the optimal solution during the detection on different views of the objects. The oscillations that may occur are mainly caused by the sampling process.

The robustness of our hierarchies in the presence of clutter and occlusions is illustrated in Figure 7.10. A test image is created that contains two different learned objects as well as many distractors and occlusions. A large number of primitives (interest points) are detected in the image (Figure 7.10 (a)). In this experiment, we add extra difficulty by assigning every interest point to the most similar feature class, without requiring a minimum degree of similarity. This results in noisy detection data. However, the use of geometric relations to infer the presence of higher-level features allows an unambiguous detection (Figure 7.10 (b)). Only a few features are needed to detect the objects.

Another interesting property of detection as inference in a graphical model is that it enables the inference of the localization of occluded features. Figure 7.11 illustrates that NBP correctly infers the localization of the missing features for a partially occluded object.

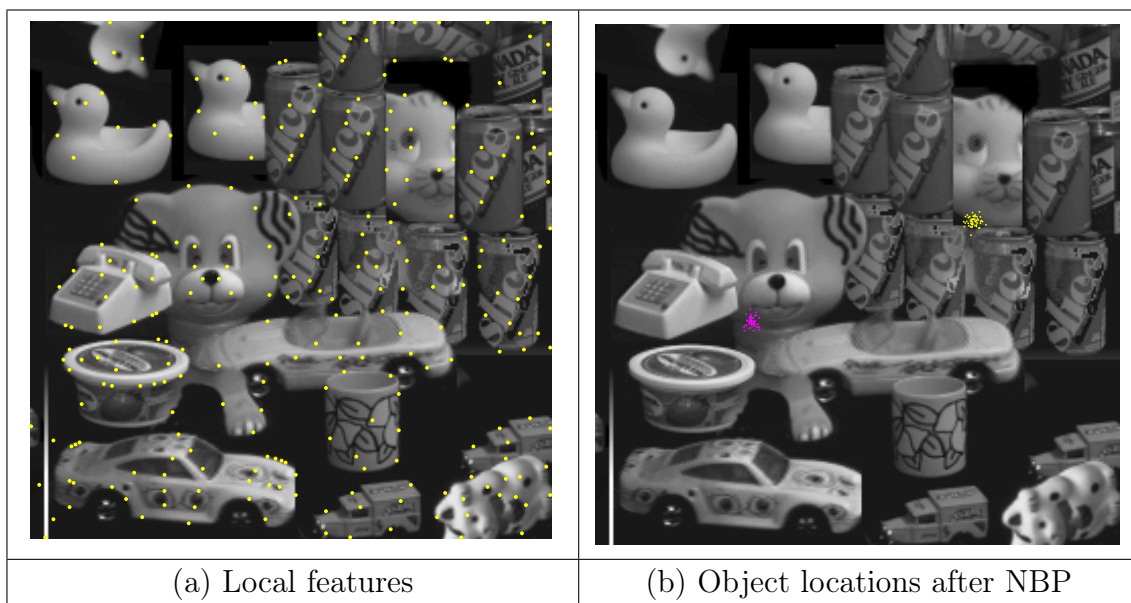


Figure 7.10: Harris interest points (a) are detected on a scene made of different objects of COIL-100. The position of two previously learned objects is correctly inferred by NBP (b). Samples corresponding to the posterior marginals (beliefs) for two objects are depicted by yellow and pink points.

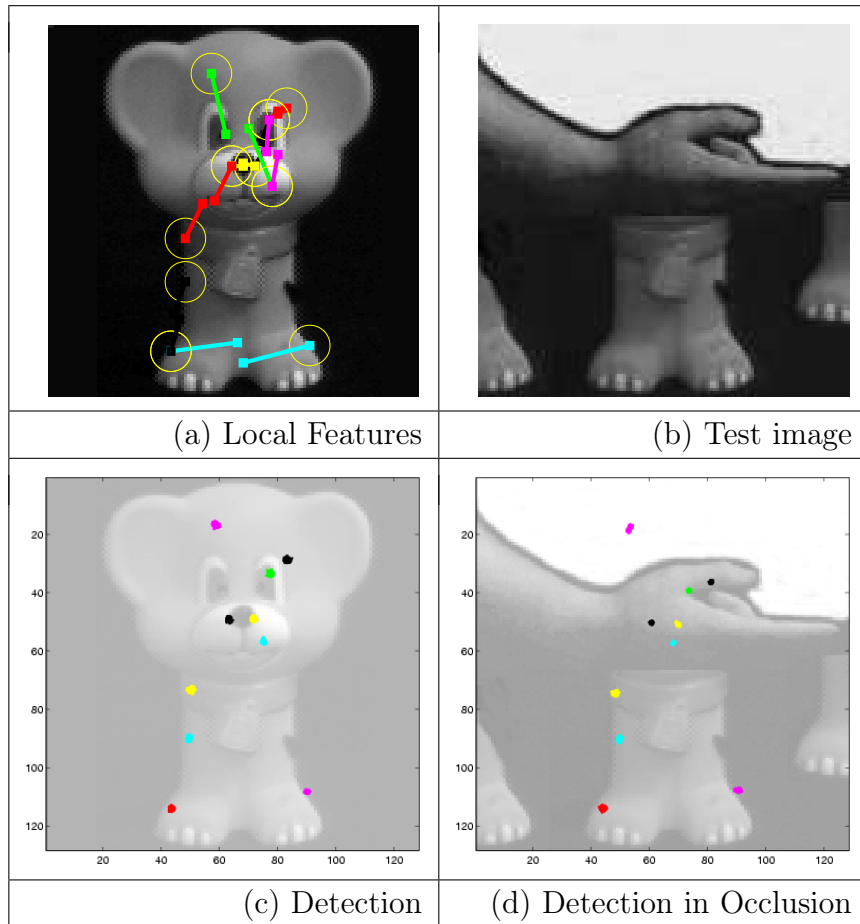


Figure 7.11: Spatial relations between the features of the first level are presented in the top left image. The results of the detection process after 6 iterations can be observed (for the first level) in the bottom images. For the occluded object, NBP correctly inferred the presence of missing first level features.



### 7.2.4 Discussion

In this section, we illustrated through simple experiments the ability of our system to learn efficient feature hierarchies and to use them for detection in previously unseen images. The framework offers several interesting properties such as the natural inference of occluded features and an intuitive representation. Moreover the use of a hierarchical representation enables to model high-level dependencies and variability properties between features that would not be possible to represent by a single-level set of features. Taking advantage of graphical models, we represent shape and appearance separately. This allows us to deal with shape deformation and appearance variability at different levels within a single framework. Moreover, our topology is invariant to rotation and translation of the object.

However, the framework presented in this section, like many others [Pia01, Fer05, Laz06, Ope06, Gra06], relies exclusively on the detection of interest points and their spatial relations. Therefore if the detector fails to detect local features, it will inevitably affect the performances of our hierarchies. To overcome this problem, we will exploit, in the next section, extracted features along a randomized grid.

## 7.3 Multiple Viewpoint Object Recognition

Evaluating object models tuned for a specific view may reveal some interesting properties. However in many real world situations, objects generally appear at different poses in the scene. This induces some variations in images such as shape deformations, appearance changes, shadowing effects, and self-occlusions. Because of these factors, Multiple Viewpoint Object Recognition is clearly more challenging.

In this section, we investigate the behavior of our hierarchies on two multiple-view object recognition tasks, namely COIL-100 [NNM96] and Ponce Group's object recognition dataset [RLSP06]. Through these experiments, we aim to demonstrate that several views can be learned within a single object hierarchy. In addition, we want to quantify the gain of each additional level in the hierarchy on recognition performances. In comparison with previous test images, an additional difficulty in Ponce's dataset [RLSP06] comes from the use of real scenes, where clutter, occlusions, and non-uniform illumination changes are common. The robustness and accuracy resulting from our framework is compared with the best state-of-the-art methods.

### 7.3.1 Experimental Protocol

Similarly to the previous experiments, the scenario used to evaluate our framework on the two multi-view datasets consists of a learning and a detection phase. Our co-occurrence learning algorithm is first applied separately on the training views of each object. This learning process generates a graphical model for each object in the dataset. They are then exploited for detection in the test images.

For the COIL-100 task, the first 25 objects are considered. In contrast with the previous set of experiments (Section 7.2) where 5 neighboring views were used to train the models, we now utilized 24 of the 72 available views uniformly distributed around each object. Detection is tested against the 48 remaining images of each object, leading to a total of 1200 ( $48 \times 25$ ) test images.

The second series of experiments was conducted on Ponce’s object database where each object model is constructed separately using 16 to 20 training images, except for the apple which is modeled from 29 images. For recognition experiments, all eight of our object models are evaluated against a set of 51 test images. Each test image containing instances of up to five previously learned objects. However, most of them only contain one or two.

### 7.3.2 Parameters and Implementation

For these experiments, most of the parameters used are similar the one detailed in the previous section (K-means, NBP, ...). We mention here a few extensions that have been introduced to cope with a multiple-view recognition task.

The most significant change to represent multiple views within a single hierarchy is to allow several nodes at the top level of the graph. Each of these nodes representing a given visual aspect of the object. Four different graphical models corresponding to objects of COIL-100 are shown in Figure 7.12.

In our experiments, both on COIL-100 and Ponce’s dataset, it appears that the framework gave the best recognition performance with 7 levels. By looking closely at the graphical model, we observed that further levels contained less repeatable combinations and eliminated high-level features that were useful for recognition. For learning on Ponce’s database, we kept the best five spatial relations of each feature to construct a hierarchy. Because of the smaller resolution of images in COIL-100, the combination of each feature class was limited to its best three spatial relations. In order to reduce the computational cost during inference, spatial relations were defined only in terms of distance between feature classes.

Instead of extracting local features at interest point locations, our primitives are now extracted using a randomized grid and described by rotation-invariant descriptors of  $13 \times 13 \times 3$  pixel values. As we mentioned in Chapter 3, a randomized grid has several advantages in comparison with methods based on interest points or uniform grid. First, it allows to cover all the object; even the poorly textured regions that are often missed by interest points. Second, it is not subject to the sensibility that may arise from a uniform grid (if the uniform grid is close to a line a small decay will have a big impact on all the descriptors on that line). Figure 7.14 illustrates patches extracted along a randomized grid together with the visual codebook associated to the object. On both dataset, the training images used in our experiments were previously automatically segmented to eliminate the uniform background.

### 7.3.3 Evaluation

The impact of the hierarchical representation is first evaluated on COIL-100. Figure 7.13 gives the confusion matrices concerning the first 25 object models for increasing number of levels in the hierarchy. Brightness indicates the total number of detections. As we can observe, a single level of feature combinations is weakly discriminant. Thanks to our hierarchy, selectivity arises from spatial combination of high-level features. Figure 7.15 shows a parallel between the evolution of the global classification error rates and the number of feature classes at the top level. Interestingly, the number of visual classes is approximately the same in level 1 and 6, but the latter reduces significantly the error rate. These classification error rates can be considered as acceptable, but are clearly not excellent. In comparison, Marée [MGPW04] obtained less than 1% of error on the all dataset. To give more intuition on which features are represented in our models, we show in Figure 7.17 the best spatial relations identified at the first level of the graph for different objects. It can be seen that the features are often able to cover a large part of the object whereas methods based on interest points would have some difficulties in more uniform regions.

The recognition results on Ponce’s dataset are shown in Figure 7.16 as a Receiver Operating Characteristic (ROC) curve. To perform recognition and produce these curves, we evaluated the kernel densities of the features at each location in the image and summed the best five feature responses. These results are comparable to the state-of-the-art methods using 2D models [FTG04] or explicit 3D matching [RLSP06]. This is remarkable since we did not use any discriminative

information to build our models. The method proposed by FERRARI *et al.* [FTG04] considers all training images independently which essentially reduces recognition to the matching of robust features. By using statistical learning and a flexible representation, we obtain good results with simple, randomly extracted features.

### 7.3.4 Discussion

We demonstrated that our feature hierarchies, first designed to be applied on single view object recognition tasks, can be generalized to perform object recognition tasks using multiple views.

Our hierarchical framework which is generic enough to integrate randomly extracted features, has demonstrated, during experimental evaluation, the ability to attain reasonable results on difficult databases where large viewpoint change, clutter, occlusions and illumination variations were common.

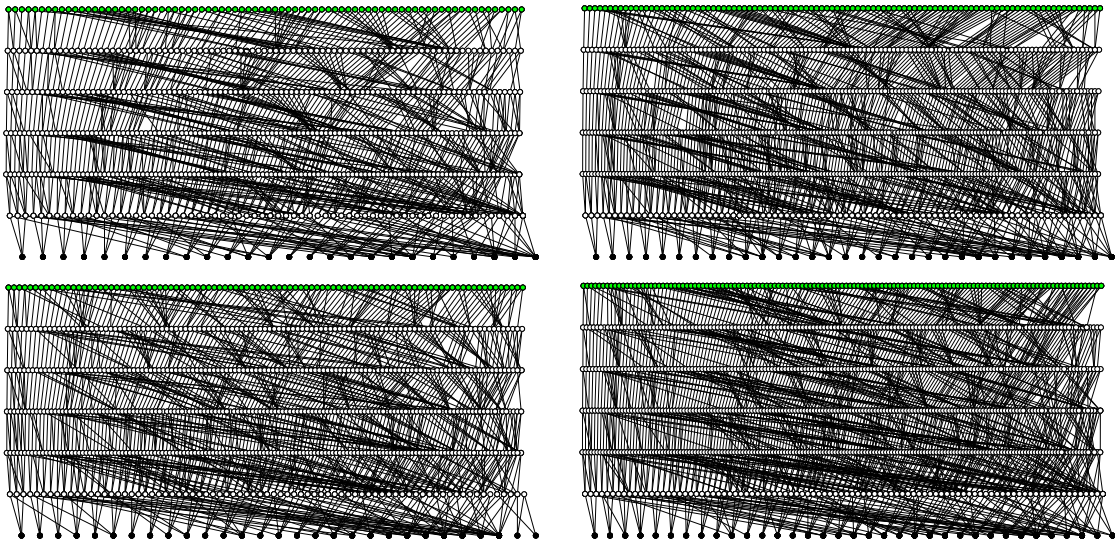


Figure 7.12: Four different hierarchies learned on COIL-100 objects. Each graphical model comprises seven levels of increasingly complex features.

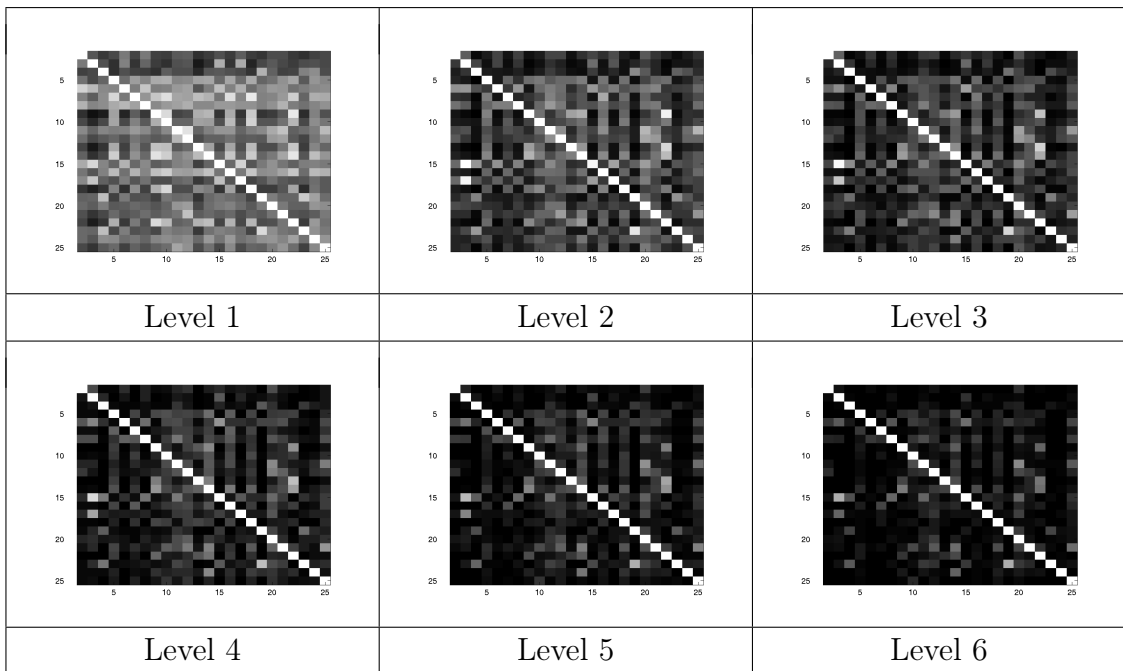


Figure 7.13: Confusion matrices for one- to six-level models (COIL-100 [NNM96]).

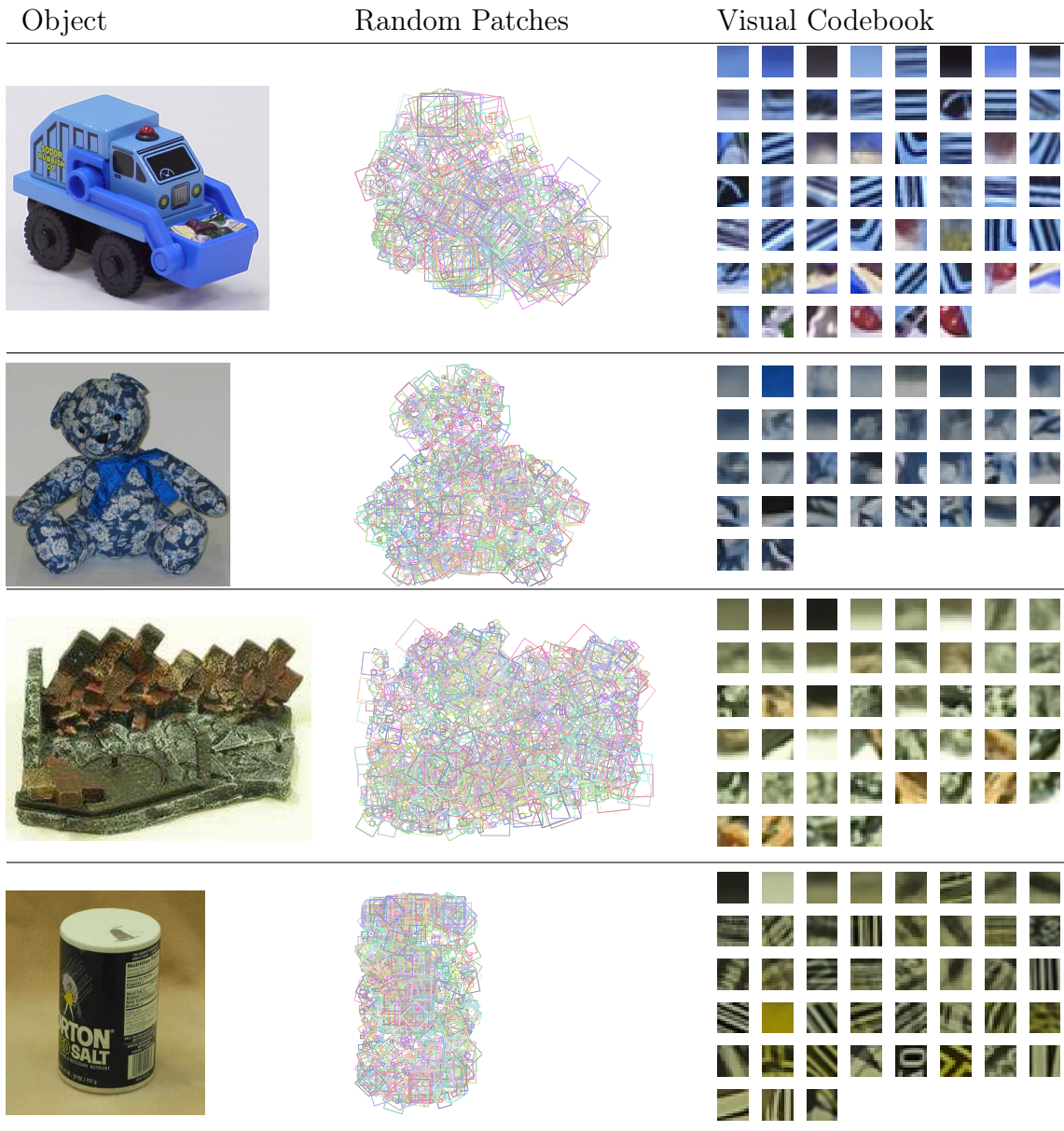
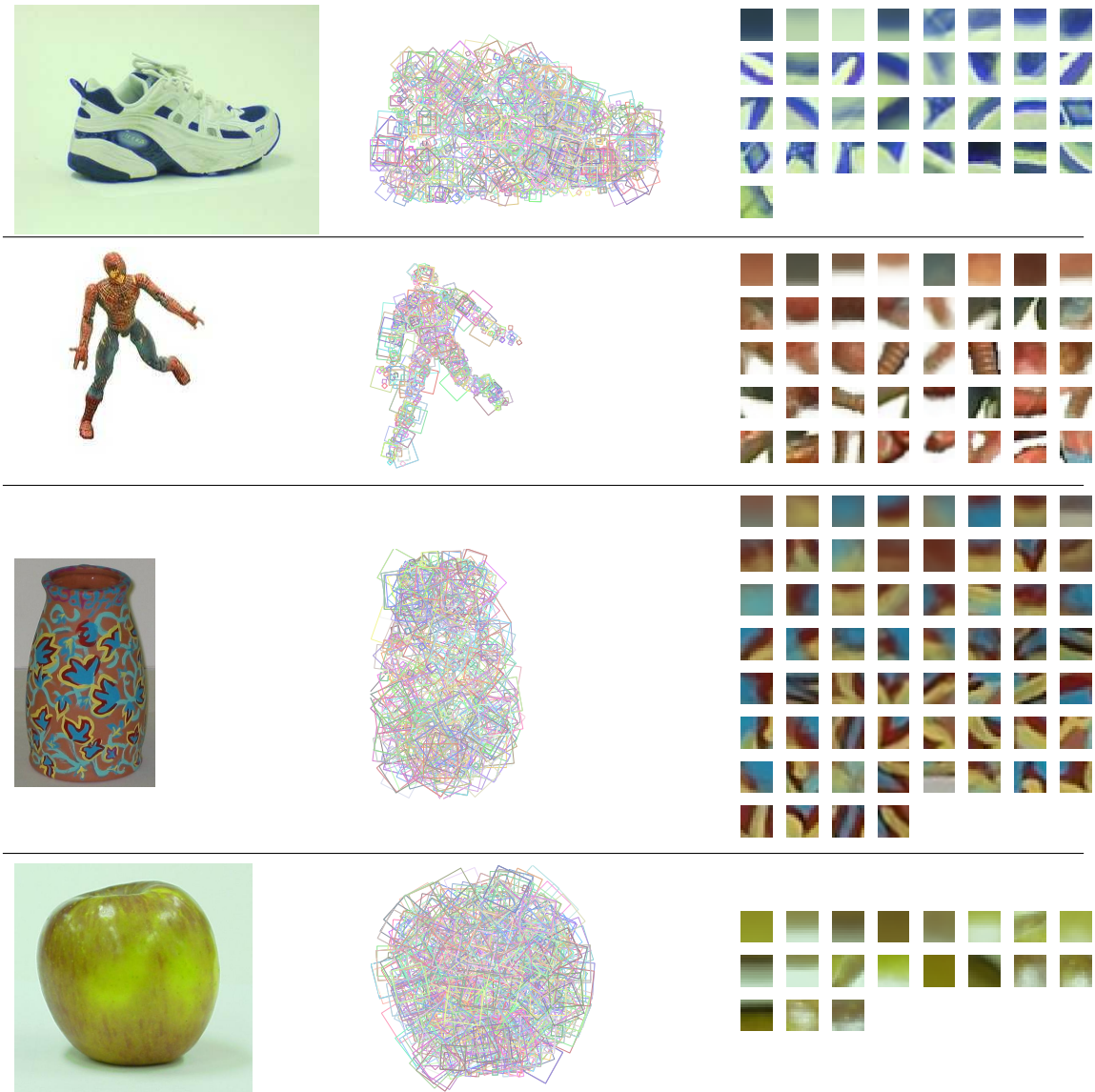


Figure 7.14: For each object class of the Ponce’s object recognition dataset, a training image is shown on the left. The random patches extracted on this image are illustrated in the center. The visual codebook learned on the all training set is illustrated on the right column. Patches are sorted with respect to the number of observations falling into that class.



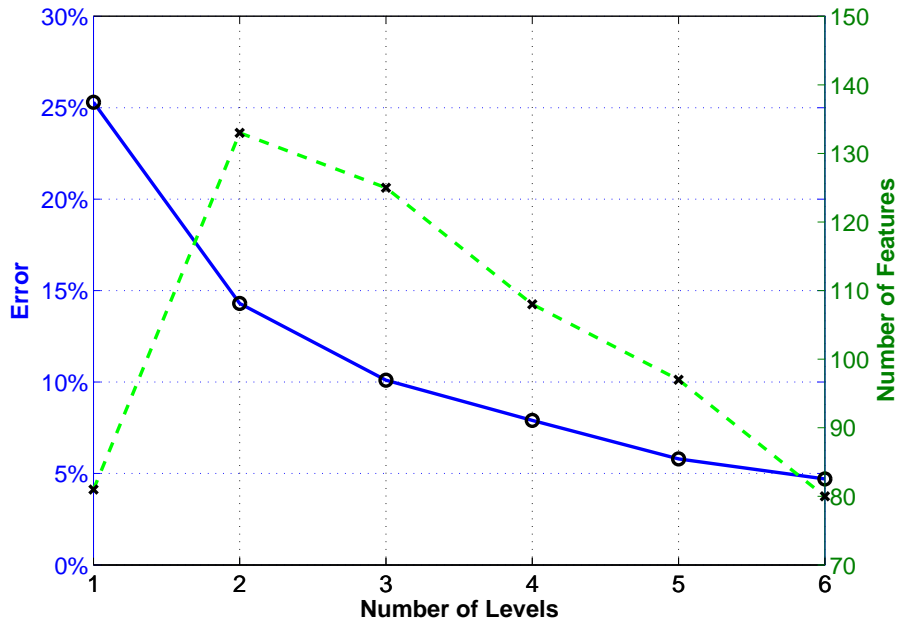


Figure 7.15: Classification error on COIL-100 [NNM96] versus the mean number of feature classes at the top level.

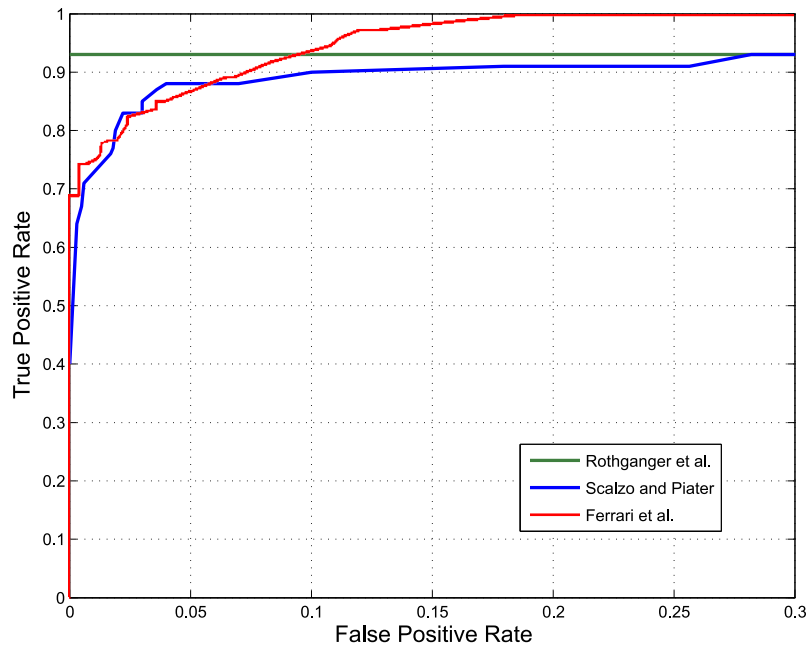


Figure 7.16: ROC curves obtained by our hierarchical framework and two of the best state-of-the-art methods.



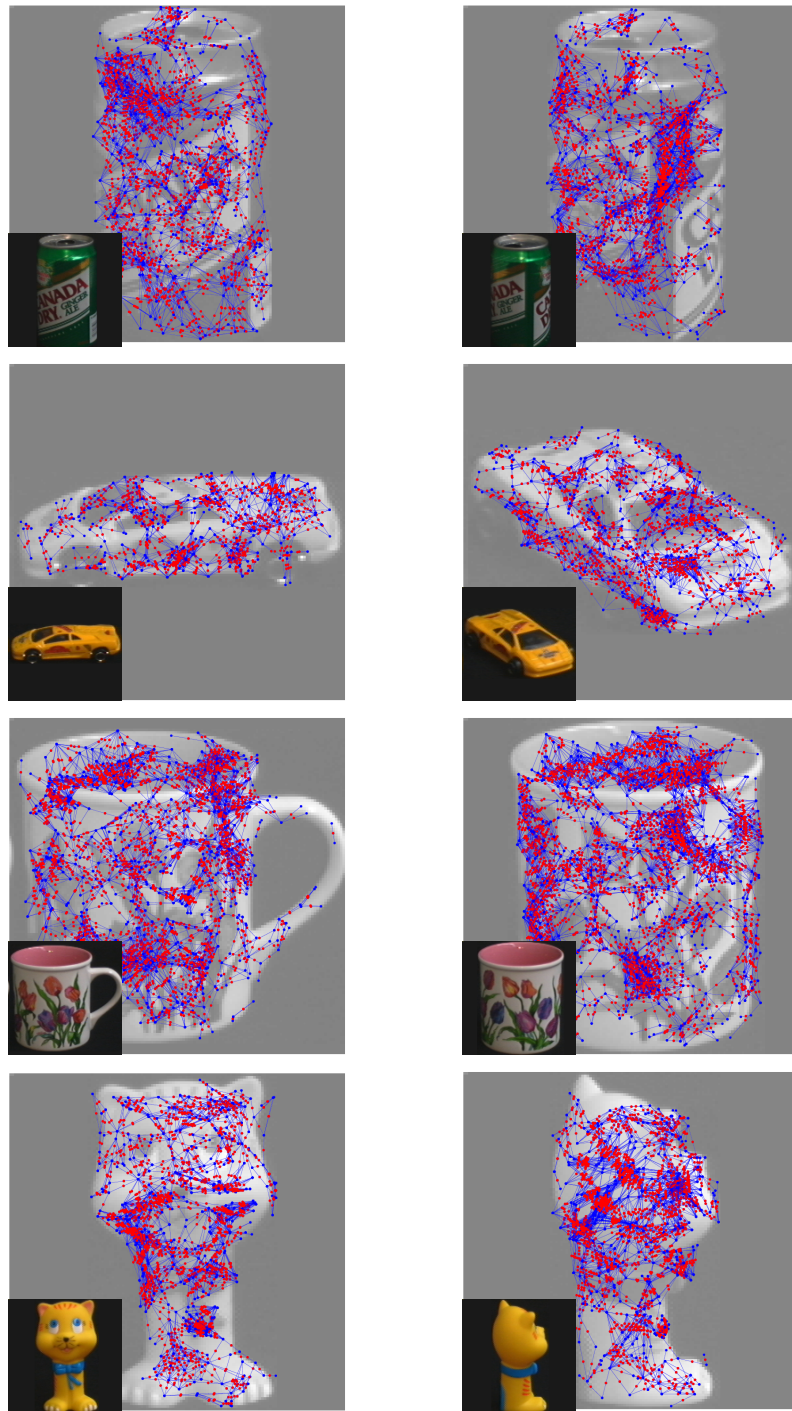


Figure 7.17: Illustration of spatial relations identified at the first level.



Figure 7.18: Examples of recognition of two objects in Ponce's object database [RLSP06]. The kernel density corresponding to the final belief (posterior marginals) after NBP are shown for the best features.

## 7.4 Object Class Recognition

The previous sections focused on the recognition of specific objects where the variations in appearance were mainly due to the pose and contextual factors. In this section, we aim to evaluate our hierarchies on the recognition of object classes. This task is more challenging for both learning and detection aspects because there exist much larger intra-class variations. In other words, appearance and shape between objects of a same class may vary.

Through the set of experiments provided below, the generalization power of our method is evaluated. To be effective it should be able to recognize the class membership of previously unseen object instances. The learning of object classes and their recognition is made possible by combining our statistical learning algorithm of feature hierarchies with a discriminant layer taking the form of a SVM classifier (see Chapter 6).

After presenting the experimental protocol in Section 7.4.1, we give some details about the structure of the framework and its parameters in Section 7.4.2. Evaluation is presented in Section 7.4.4 by using two object class datasets, Butterflies [LSP04] and Soccer [VdWS06]. Our method is compared with some of the best available methods and a bag-of-feature system of our own (Section 7.4.3). We also measure the effect of our adaptive patch features on the overall recognition performance.

### 7.4.1 Experimental Protocol

In these experiments, we evaluate our method against two challenging applications of object class recognition, the *Soccer* [VdWS06] and *Butterflies* [LSP04] image databases. The *Soccer* dataset contains 315 images, including 140 for training; the task is to recognize the team membership of soccer players. The *Butterflies* dataset is composed of 619 images, 182 of them for training, acquired from the Internet. Here the objective is to identify the butterfly species. Both datasets comprise seven classes.

During training, the system first exploits our co-occurrence based learning strategy to build a graphical model (PMRF) separately for each object class. Then a feature selection step based on the Fisher score extracts the most discriminant features. Finally, a multi-class SVM classifier is trained on the output activations of the selected features together with their class labels membership.

In recognition, the system is presented previously unseen images. Each image

contains one learned object class but may contain several instances of it. The class membership is obtained by first processing the image using NBP on all the seven graphical models, and then evaluating the SVM classifier prediction for these measurements.

## 7.4.2 Parameters and Implementation

In comparison with previous experiments, a key structural difference of the system resides in the use of adaptive patch features to represent the appearance models of higher level features (Chapter 6). During learning, our co-occurrence based learning strategy is exploited to identify pairs of features and to combine them in a bottom-up fashion. Instead of using a single region detector to produce our primitives, different kinds of detectors are used to extract image regions of potential interest. For detection, evidence is propagated using Nonparametric Belief Propagation (NBP). However, since many feature classes are not discriminant and therefore not useful for classification, we add a discriminative layer that is learned on the maximum belief activations of the graph. These discriminative models are learned by combining a SVM classifier with feature selection based on the Fisher score (Chapter 6). We successively provide in the next paragraphs detailed explanations about the parameters and design strategies involved in these techniques.

### Local Feature Extraction

As it was illustrated in Chapter 6, our local primitives are extracted using different feature detectors. In these experiments, we exploit simultaneously five region detectors: MSER, Hessian-Laplace, Hessian-Affine, Harris-Affine, and Randomized Grid (chapter 3). For description, each region is normalized to a canonical window consisting of  $13 \times 13 \times 3$  pixel values and converted to the HSV colorspace. This normalization is done by mapping the elliptic region obtained by the detector to the inner circle of the reference window [Mik02, MTS<sup>+</sup>05].

One might ask what is the motivation behind the use of different detectors, since at the end the features extracted along the randomized grid should approximately contain all of them. The reason is quite simple: learning spatial relations from random locations can be more challenging since the large number of noisy features (signal-to-noise ratio) may interfere with the learning process. Local features extracted on the basis of the image signal are often more suitable of finding reliable co-occurrences. Another reason in favor of using multiple detectors is that we do not

apply an affine normalization to the regions detected along the randomized grid. Therefore, the descriptors computed at the same location may differ because of this affine factor. As is has been discussed in Chapter 4, each detector has potential weaknesses and is more sensitive to specific changes in the image. Figure 7.19 illustrates the regions obtained from different detectors.

### High-level Appearance Models as Adaptive Patch Features

In previous experiments, high-level features were completely defined by a spatial configuration of lower features. The object model now associates to them an appearance model that is defined over a region of shape  $\mathcal{X}_{ij}$ . To estimate these appearance models, we exploit the *Adaptive Patch Features* (Chapter 6). In these experiments, the relative factors vary between 0.1 and 2.0 in the two dimensions. During detection, each adaptive patch is normalized to a canonical frame of  $13 \times 13$  pixels. Some adaptive patch features resulting from the learning process are illustrated in Figure 7.20.

The shape estimated from our adaptive patch features appears to be intuitive in most of our experiments. By analyzing adaptive patch characteristics, we observed that the patch deformations often fall in two main categories:

1. In the most common case, the region to be estimated contains variations that can visually be recognized as edges, corners, *etc.* Intuitively, whenever the patch is located close to an intensity transition in one direction (*i.e.* an edge), the selected shape deformation tends to minimize the variance by reducing the scale in the orthogonal direction of the gradient.
2. Less frequently, the region of interest may be located in a uniform area which is often bounded by some gradient (such as the regions detected by the MSER detector). In such a case, if the variation across the images is high, the size of the patch will tend to grow to cover the maximum part of the uniform region. Conversely, if the neighborhood is very stable, the scale will decrease.

Adaptive patch features are sensitive to the number of training examples: if the training set is not large enough, the patches are not accurate. Moreover, adaptive patches have some difficulties to estimate unstable texture regions (note that this is a common problem for MSER features too). Therefore, to compute an adaptive patch we require at least 50 different locations of the patch across the training set. If this is not the case or whenever the total variance (of the best solution) over

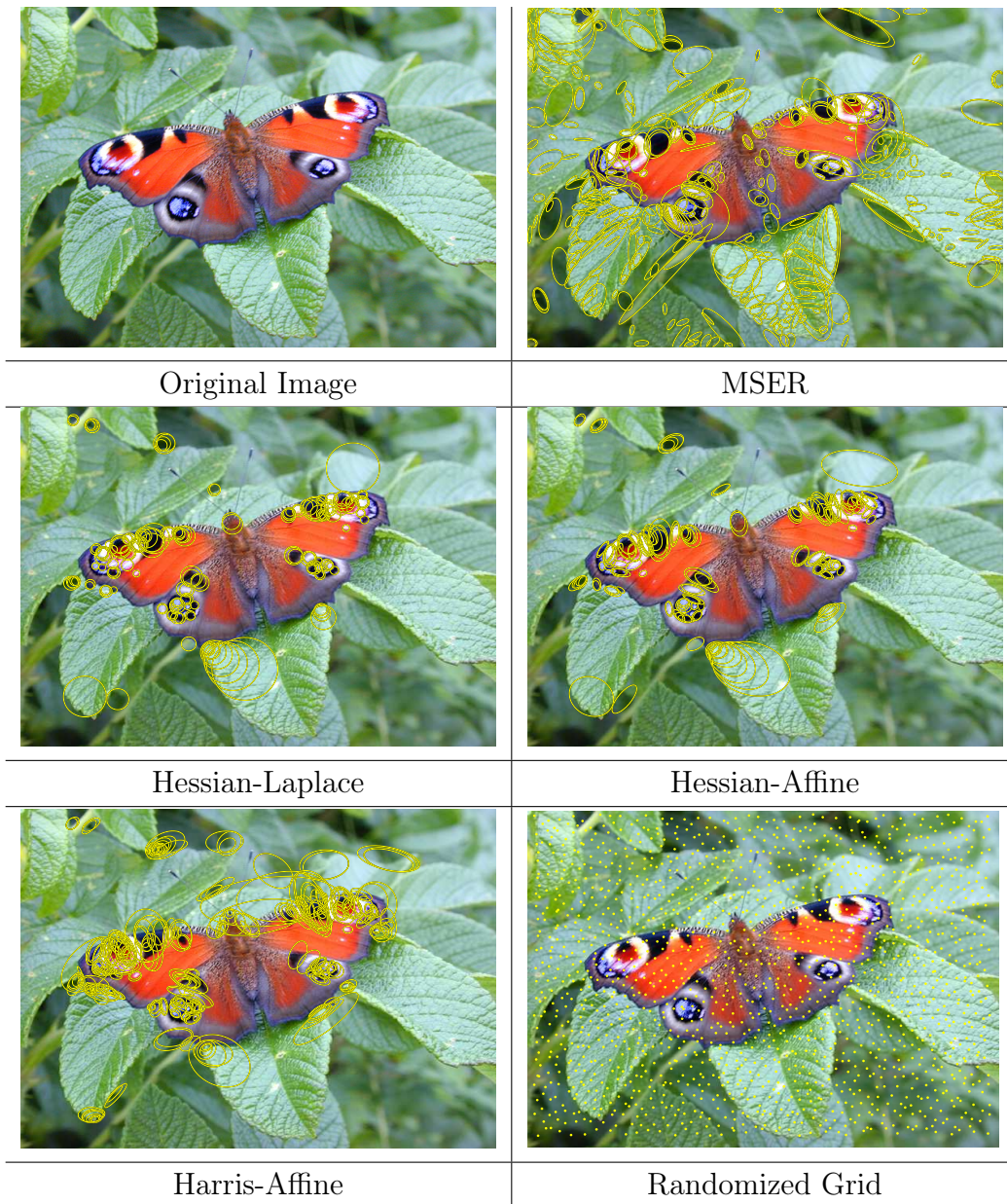


Figure 7.19: Local regions extracted from multiple feature detectors on a Peacock butterfly [LSP04]. Feature detectors based on gradient (Hessian-Laplace, Hessian-Affine, Harris-Affine) or the intensity image functions (MSER) precisely identify patterns in the image. However they tend to fail to cover uniform regions although they may be very representative. The redness of wings constitutes a representative pattern for Peacock species that is best extracted by our randomized grid detector.

the training set is above a given threshold, the appearance and scale are set to the original scale [1.0, 1.0[.

The learning of our adaptive patch features is computationally hard. A very large number of patches have to be extracted at the different scale factors ( $18 \times 18 = 324$ ) in the training images. For more efficiency, we used the Intel Integrated Performance Primitives (Intel IPP) [Ste04]<sup>5</sup>. This is an extensive library of multi-core-ready, highly optimized software functions for multimedia and data processing applications. Typically, it offers a speed-up superior to 25%.

### Support Vector Machines

Support Vector Machines (SVM) are used to produce a classifier from the output beliefs of the graphical model. We used the SVM implementation provided in the LIBSVM library [CL01] to learn a multi-class classifier.

To avoid any effect of numeric ranges attributes and numerical difficulties during the calculation, LIBSVM rescales automatically each feature vector component  $x_i$  to the interval  $[0, 1[$  via the transformation:

$$x'_i = \frac{x_i - \min_i}{\max_i - \min_i} \quad (7.1)$$

where  $\min_i$  and  $\max_i$  are respectively the minimum and maximum values of the  $i$ th feature. These values are obtained from the training samples.

The discriminant function for a general SVM classifier is given by

$$f(x) = \sum_{i=1}^n y_i \alpha_i K(x, x_i) + b \quad (7.2)$$

which consists of a weighted sum of the distances between a novel vector  $x$  and each of the training vectors  $x_i$  with label  $y_i$ . The distance is computed via kernel function  $K(x, x_i)$  that may take different forms depending on the application. Note that training vectors  $x_i$  corresponding to nonzero coefficients  $\alpha_i$  are called support vectors of the optimal separating hyperplane.

Similarly to other work [NC04, BG05], we have found that a radial basis function (RBF) kernel appears to work fine. The RBF kernel non-linearly maps samples into a higher dimensional space:

$$K(x, x_i) = e^{-\gamma \|x - x_i\|^2}, \gamma > 0 \quad (7.3)$$

---

<sup>5</sup>The Intel IPP library is available at <http://www.intel.com/>. SEBASTIEN JODOGNE has largely contributed to developing a convenient C++ interface during his PhD [Jod06].

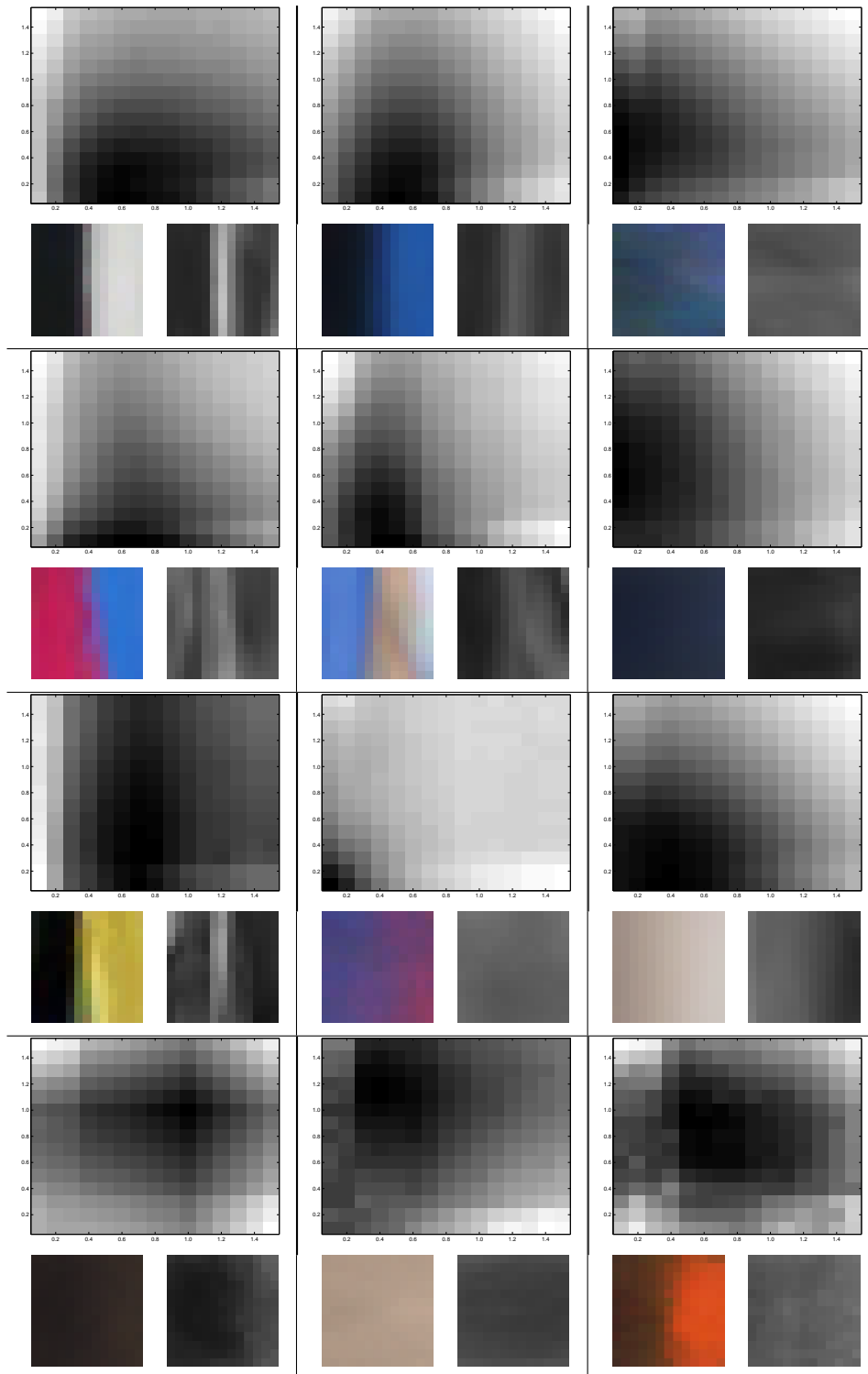


Figure 7.20: Adaptive patch features for different spatial relations. A variance map (over the training set) is shown for each adaptive patch as a function of its shape. The maximum is selected to produce means and variances that are shown on the bottom panels. A fast way to estimate the goodness of a feature is to look at its variance.



The use of the RBF kernel involves two parameters that need to be specified prior to training: the kernel parameter  $\gamma$ , and the penalty parameter  $C$  that controls the amount of penalty on the error term [CL01]. Each combination of the design parameter values corresponds to a different SVM model and may have a large impact on the classifier performances. The SVM implementation that we use [CL01] provides a model selection tool that performs a 2D grid-search for the pair of values which minimizes the mis-classification rate using cross validation.

### 7.4.3 Bag-of-Features

For evaluation purposes, we implemented a bag-of-features recognition system similar to the one recently proposed by NOWAK *et al.* [NJT06]. The main objective is to compare our feature hierarchies with such a geometry-free model. Moreover by incorporating our adaptive patch features into the bag-of-features model, we will be able to measure how useful they can be on recognition tasks.

In this bag-of-features framework [NJT06], a visual codebook is produced by applying k-means algorithm on a set of random patches extracted from the training images. Similarly to our framework, Euclidean distance is used as a distance metric for comparing descriptors during clustering. Then the main idea is to count the number of occurrences of each visual word in the codebook for a given image. This yields a histogram of codeword counts for each image.

To improve the performance of the system during recognition, we exploit an adaptive threshold selection based on the mutual information (MI). By maximizing the mutual information between the feature count and the class label over the training set, the threshold can be adjusted separately for each visual word (as shown in Figure 7.21). The mutual information of two discrete random variables  $X$  and  $Y$  is expressed as:

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x) p(y)} \quad (7.4)$$

where  $p(x, y)$  is the joint distribution of  $X$  (count) and  $Y$  (label), and  $p(x)$  and  $p(y)$  are the marginal probability distribution functions of  $X$  and  $Y$  respectively.

A multi-class SVM classifier is trained on the basis of the MI optimized histograms together with their class label. The SVM parameters are estimated in a similar fashion that was explained in the previous section.

In the standard version of the bag-of-features framework, denoted  $\mathbf{B}^-$ , we use  $13 \times 13 \times 3$  HSV color pixels descriptors. Each of them computed at a normalized

scale (using Laplacian) and orientation (using gradient direction).

A more sophisticated system  $\mathbf{B}^+$  is constructed by enriching the visual codebook with our adaptive patch features, previously estimated in our hierarchies. For the purpose of these experiments, the adaptive patches are extracted at random locations and sizes, using the learned relative scales in the two dimensions.

#### 7.4.4 Evaluation

In this section, we propose some evaluation experiments on two challenging datasets; Butterflies [LSP04] and Soccer [VdWS06]. We aim at answering the following questions;

- What is the rank of our feature hierarchies in comparison with the best state-of-the-art methods?
- What is the recognition performance of the feature hierarchies in comparison with a bag-of-features system?
- What is the impact of the adaptive patch features on the recognition performances, both in the case of feature hierarchies and bag-of-features models?

To address these questions, we evaluate the performance of our hierarchies constructed with  $\mathbf{H}^+$  or without  $\mathbf{H}^-$  the adaptive patch features. In addition, we also evaluate a bag-of-features system  $\mathbf{B}^-$  (previous section) and the effectiveness of the adaptive features on such a system  $\mathbf{B}^+$ .

The results presented in Table 7.1 and 7.2 are shown in terms of recognition rate of the final classifier on the test set. The results obtained by our hierarchical frameworks  $\mathbf{H}^-$ ,  $\mathbf{H}^+$ , are compared to the two bag-of-features systems  $\mathbf{B}^-$ ,  $\mathbf{B}^+$  together with three, recently published, state-of-the-art methods; semi-local affine frames [LSP04], extremely randomized decision trees learned on random subwindows [MGPW05c] and a bag-of-features framework that exploits efficient color features [VdWS06].

On the *Soccer* database (Table 7.1), the results indicate that our hierarchical system  $\mathbf{H}^-$  outperforms existing approaches by 1% and the use of adaptive patch features  $\mathbf{H}^+$  gives an additional 3% improvements on the overall recognition rate. As discussed with MARÉE, these results obtained by [MGPW05c] can possibly be improved by using a much larger number of subwindows.

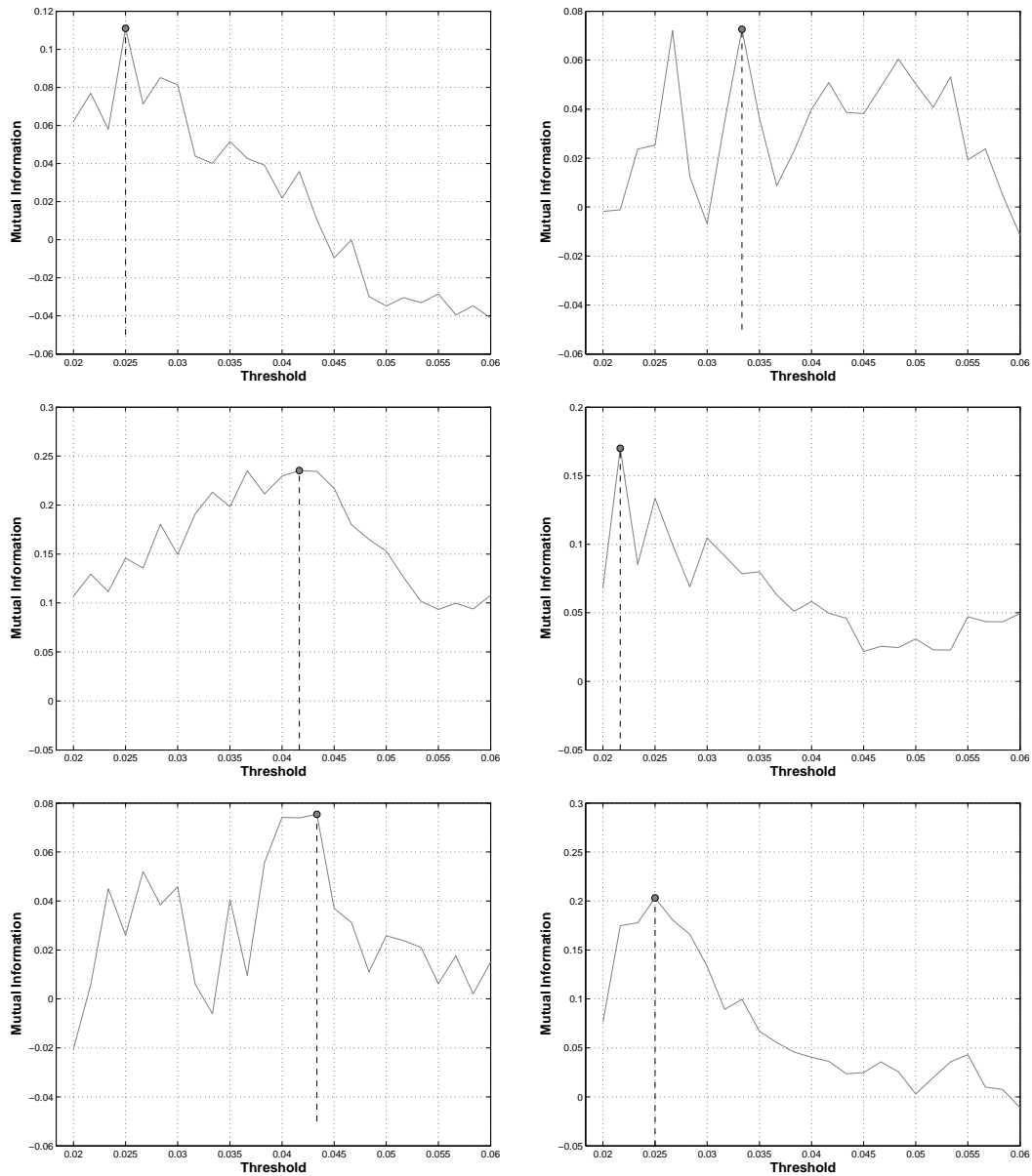


Figure 7.21: Mutual Information: threshold selection for six visual class of the codebook. The threshold used in our bag-of-features scheme is selected on the basis of the mutual information. Its value correspond to the first maximum. It is depicted by a black red and a vertical dashed line.

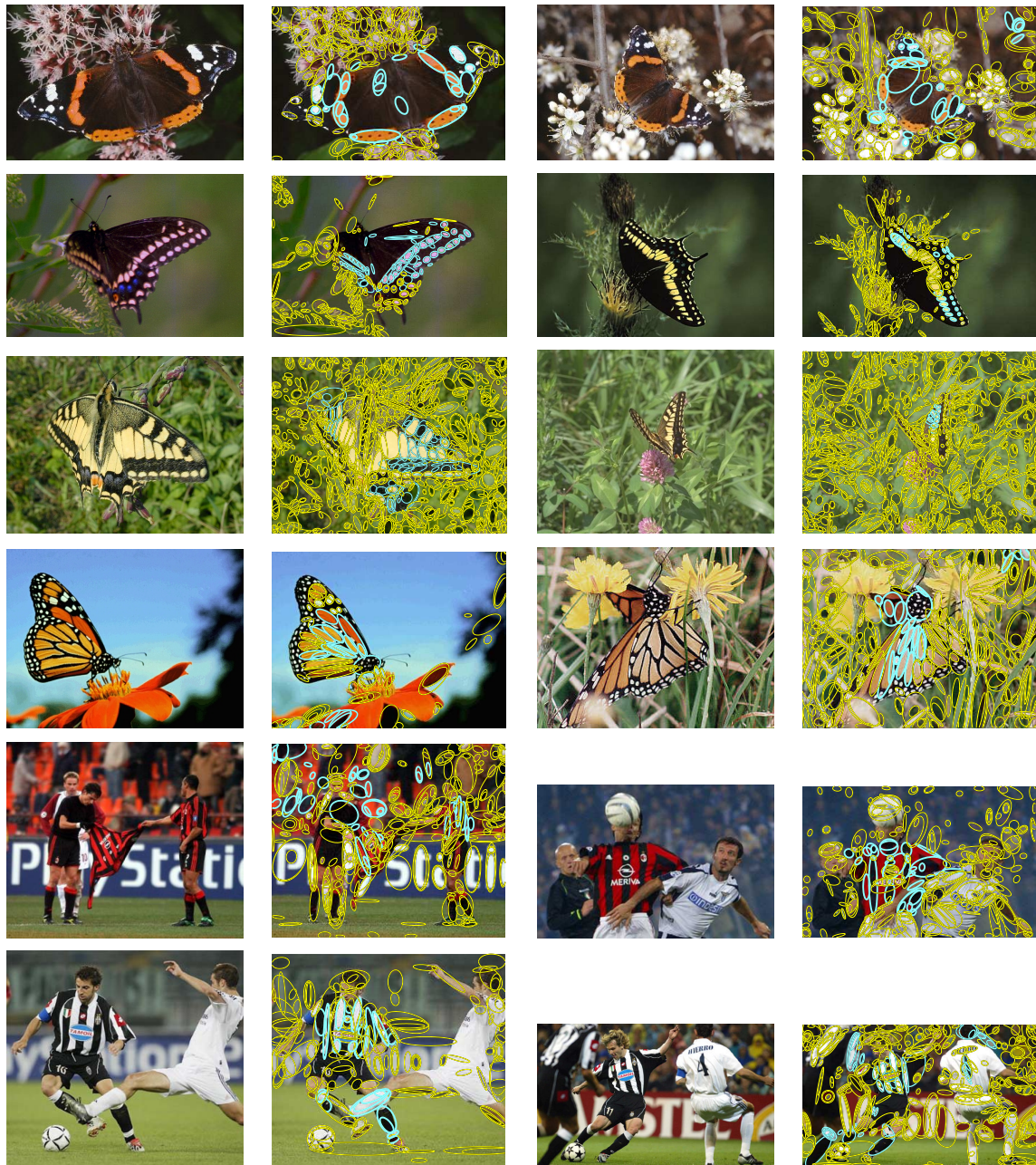


Figure 7.22: Illustration of the “most useful” low-level features during detection. Each row corresponds to a different visual class obtained either from Butterflies [LSP04] or Soccer [VdWS06] dataset. Original MSER features used during the detection process are depicted by ellipses. After evaluating the posterior marginal of each of these regions considering the best visual classes to which it has been assigned, we highlight the best features by a turquoise ellipse.

Soccer Class	$H^+$	$H^-$	$B^+$	$B^-$	[MGPW05c]	[VdWS06]
AC Milan	80%	80%	73%	67%	73%	-
Barcelona	93%	93%	93%	87%	93%	-
Chelsea	67%	67%	53%	73%	87%	-
Juventus	93%	87%	93%	80%	67%	-
Liverpool	87%	80%	87%	73%	87%	-
Madrid	87%	87%	87%	80%	93%	-
PSV	67%	60%	60%	60%	47%	-
Total	82%	79%	78%	74%	78%	73%

Table 7.1: Classification results for the Soccer dataset. We can observe the results for our feature hierarchy  $H$  and Bag-of-Features  $B$  systems with  $H^+$ ,  $B^+$  or without  $H^-$ ,  $B^-$  adaptive patch features. These are compared to methods based on random subwindows [MGPW05c] and efficient color features [VdWS06].

On the Butterflies dataset a much larger number of images is available both for training and testing. The results presented in Table 7.2 indicates that our feature hierarchies are comparable to the local affine frames [LSP04]. These results are remarkable considering that the local affine frames are clearly tuned for this dataset. To the best of my knowledge, no other object recognition system has been evaluated on this dataset.

We can observe that the use of adaptive patch features still improves the recognition rate. Interestingly, both hierarchies  $H^-$  and  $H^+$  outperform bag-of-feature models.

In Figure 7.22, we show the feature locations falling in a neighborhood that has a high posterior probability. Such a high probability occurs when several observations, obtained from different nodes, concord. In other words, this means that their relative positions and orientations were in a configuration that has been previously learned and represented in the models. Intuitively, we can say that the highlighted features were probably the most useful during the detection process to predict the position of higher level features.

Figure 7.23 and 7.24 illustrate the detection using our hierarchical models ( $H^+$ ). The first column shows the local regions obtained from a feature detector and available at an observable node  $y_i$  of the first level. Each subsequent column to the right shows the final belief of a higher-level node as a set of samples. Each of them

Butterfly Class	H <sup>+</sup>	H <sup>-</sup>	B <sup>+</sup>	B <sup>-</sup>	[LSP04]
Admiral	91%	81%	59%	73%	87%
Swallowtail	81%	75%	81%	94%	75%
Machaon	95%	84%	72%	67%	96%
Monarch 1	67%	65%	73%	65%	73%
Monarch 2	84%	79%	85%	69%	91%
Peacock	98%	94%	76%	68%	100%
Zebra	92%	83%	63%	55%	89%
Total	89.4%	83%	71%	68%	90.3%

Table 7.2: Classification results for the Butterflies dataset. Results are compared to the Local Affine Frames [LSP04].

depicts different visual aspects of the object class. The four features shown were chosen such that there exists a path linking them in the hierarchy, *i.e.*, each level- $i$  feature is a child of the level- $i + 1$  feature shown in the column to its right.

### 7.4.5 Discussion

The experimental results provided in this section show that our feature hierarchies are on par with or exceed the best published results, and highlight the contribution of our adaptive patch features. In all cases, our hierarchies outperform the bag-of-feature approaches. This can be explained by the fact that, contrary to bag-of-features systems, explicit spatial relations between features can be efficiently represented and exploited.

The challenging nature of the images used to build the models demonstrates the robustness of the learning approach. This was applied on images containing large contextual variations (lighting, image quality), pose (scale and orientation) but also intra-class variations.

However, our learning method cannot be applied directly to categorization purposes. In object categorization, objects of a same category may not necessarily share visual similarities. Therefore, the learning moves from a visual to a semantic learning process. Since our learning strategy requires repeatable patterns to be identified in the image (even if they are related by large scale, viewpoint variations). Without these reliable co-occurrences, our system will fail to learn relevant features. In object categories, much larger degree of supervision is often presented to the

learning process (object boundaries, position in the image, pose). Some systems do not require such information but assume that the pose of the object remains constant. The increasing use of new challenging databases [EZWVG, GHP07] leads us to mention in the last chapter other strategies for learning our visual feature hierarchies.

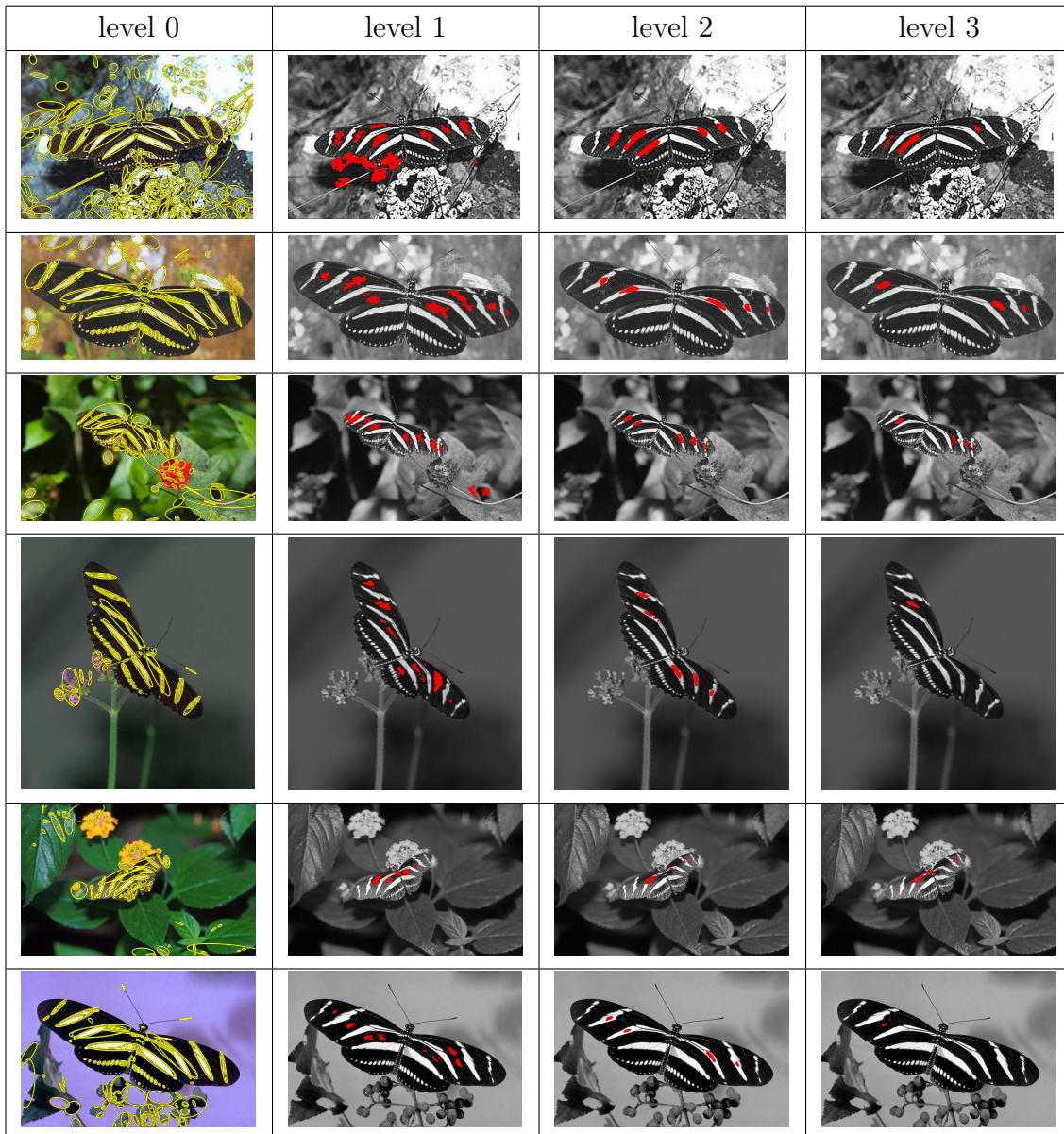







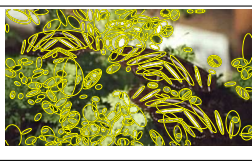
















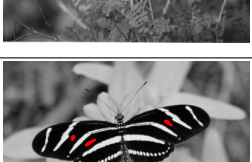
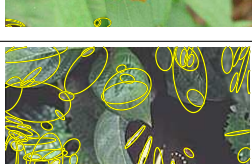
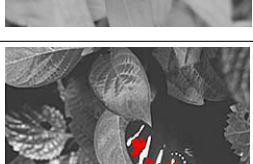
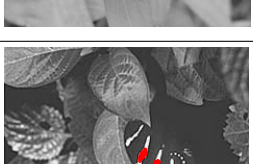
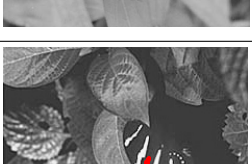


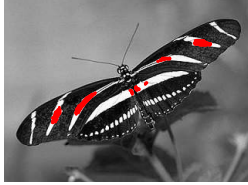





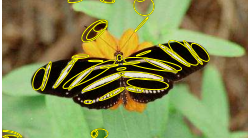







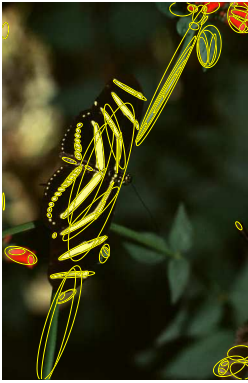






































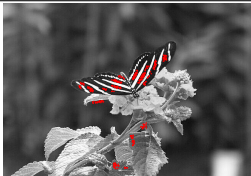
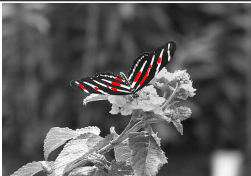
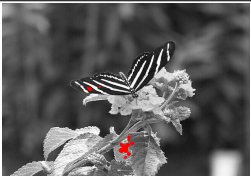

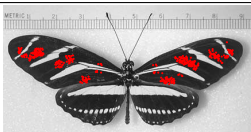
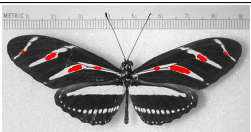
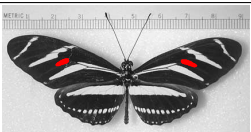

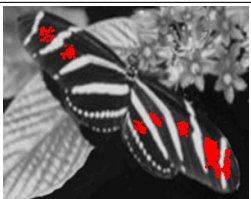

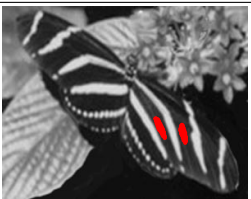
Figure 7.23: Detection of *Zebra* butterflies using our hierarchical model ( $\mathbf{H}^+$ ). The first column shows the local regions obtained from a feature detector and available at an observable node  $y_i$  of the first level. Each subsequent column to the right shows the final belief of a higher-level node as a kernel density estimate. Each of them depicts different visual aspects of the object class. The four features shown were chosen such that there exists a path linking them in the hierarchy, i.e., each level- $i$  feature is a child of the level- $i + 1$  feature shown in the column to its right.




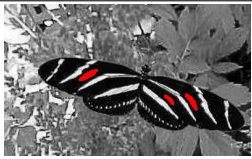


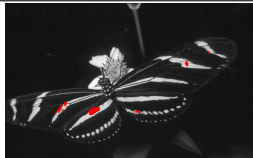


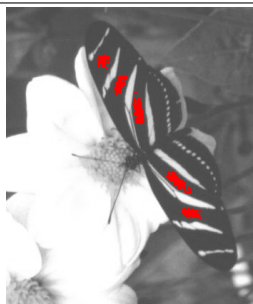

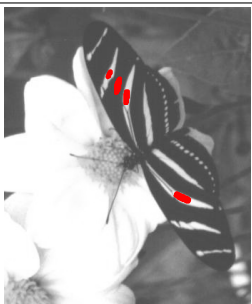




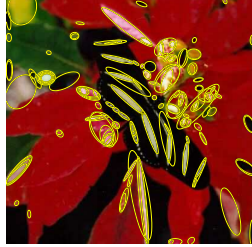



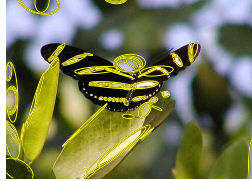


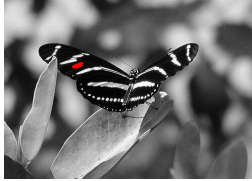






level 0	level 1	level 2	level 3
			
			
			
			
			
			
			
			

level 0	level 1	level 2	level 3
			
			
			
			
			
			
			
			

Chapter 7. Experimental Evaluation

level 0	level 1	level 2	level 3
			
			
			
			
			
			
			
			

level 0	level 1	level 2	level 3
			
			
			
			
			
			
			

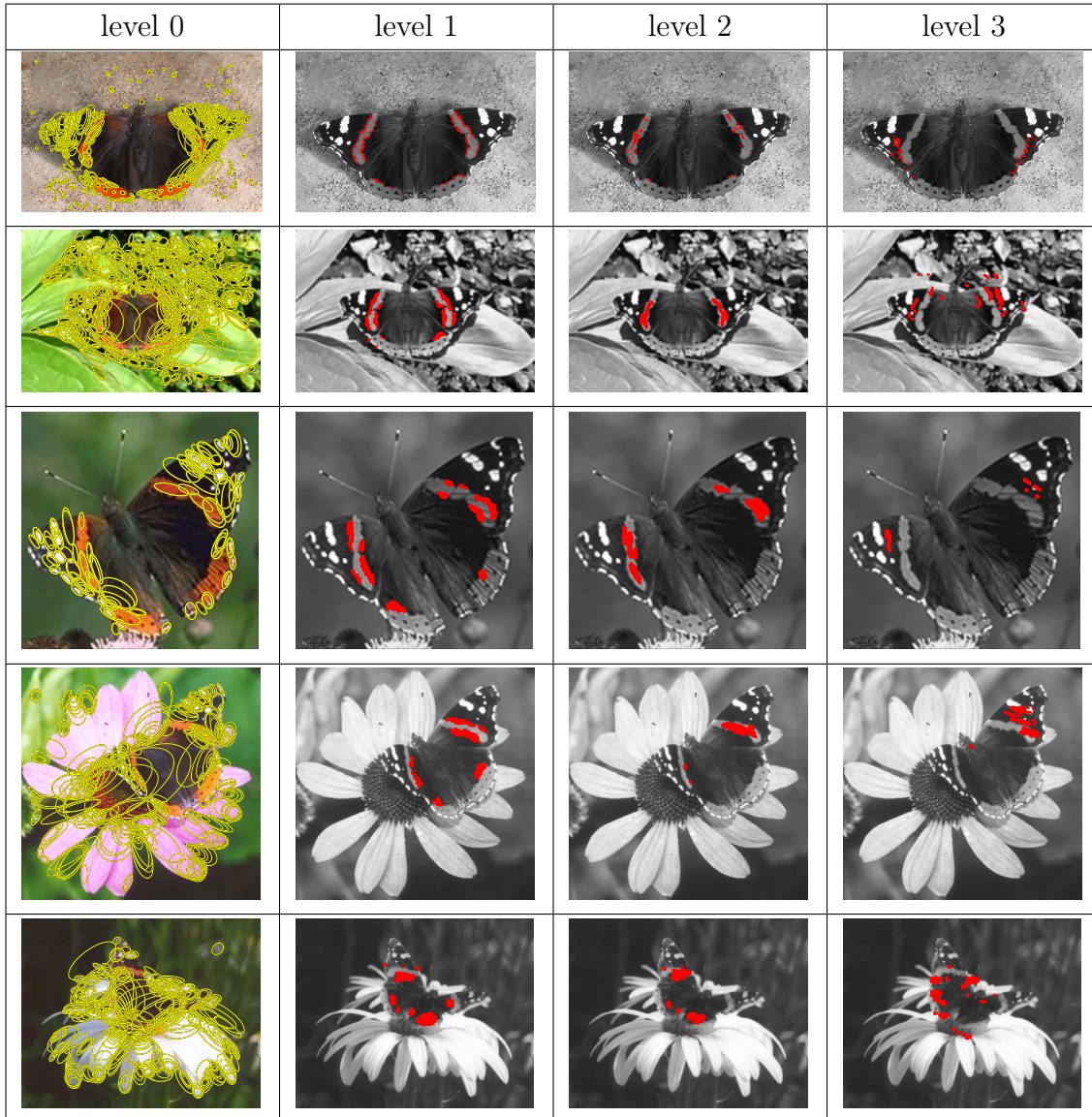
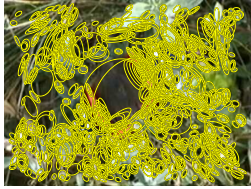



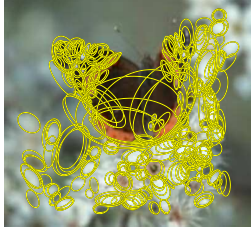
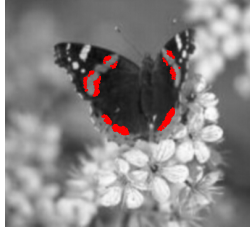
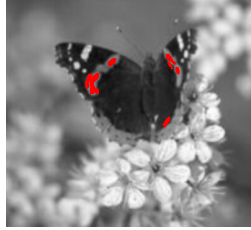


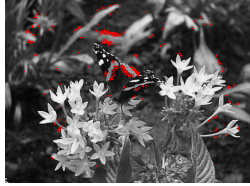
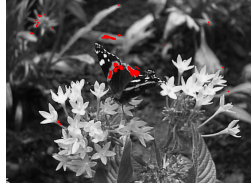
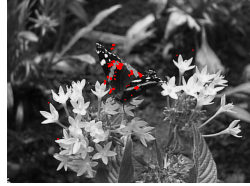


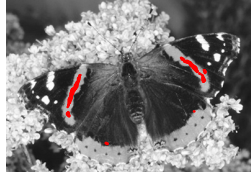
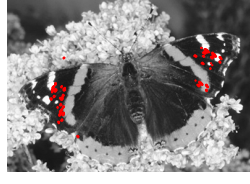




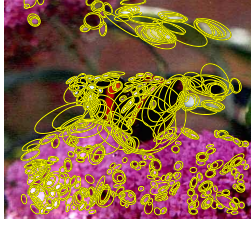
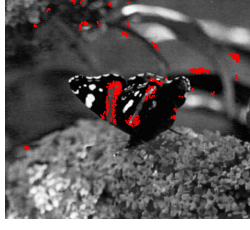
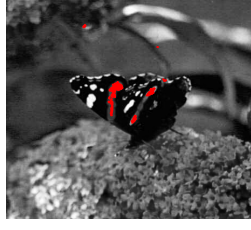
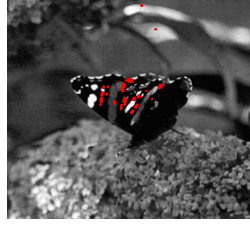

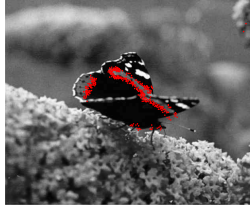
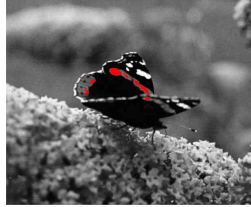
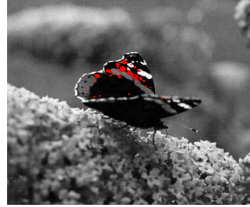

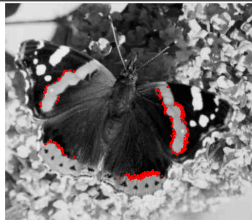
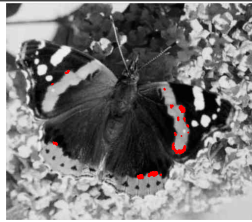





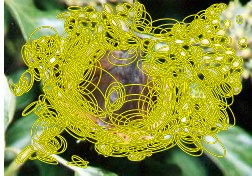








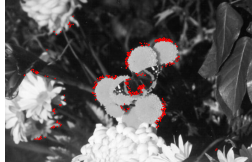

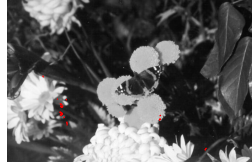









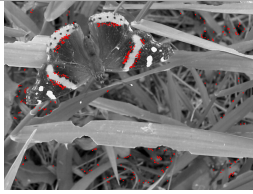
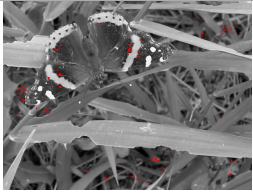
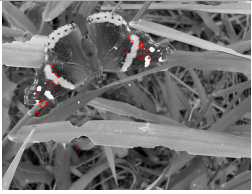

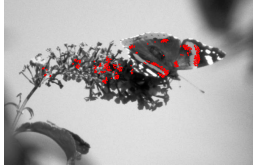
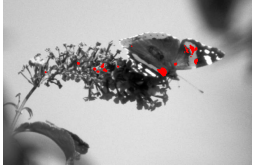
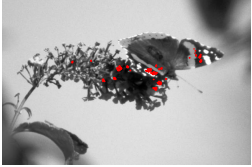
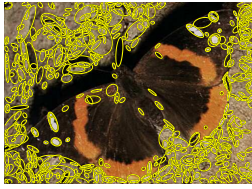


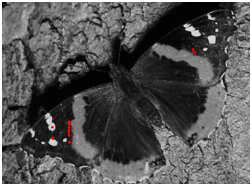
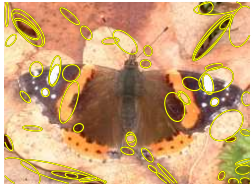

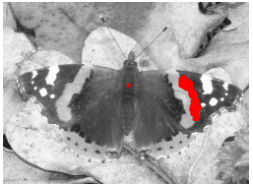
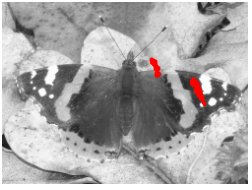
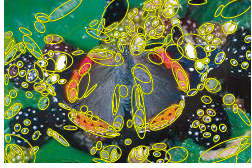




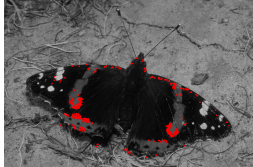


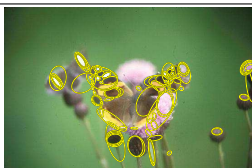

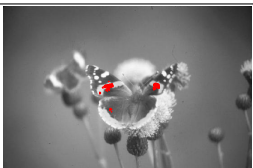














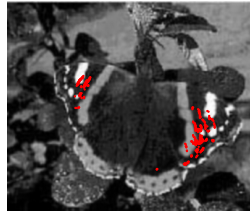




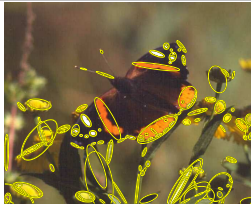
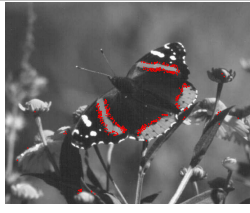
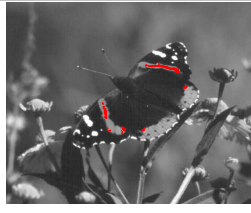
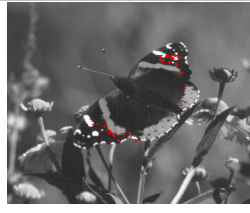




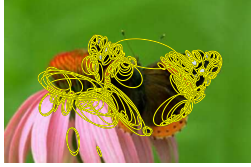







Figure 7.24: Detection of *Admiral* butterflies using our hierarchical model ( $\mathbf{H}^+$ ). The first column shows the local regions obtained from a feature detector and available at an observable node  $y_i$  of the first level. Each subsequent column to the right shows the final belief of a higher-level node as a kernel density estimate. Each of them depicts different visual aspects of the object class. The four features shown were chosen such that there exists a path linking them in the hierarchy, i.e., each level- $i$  feature is a child of the level- $i + 1$  feature shown in the column to its right.

level 0	level 1	level 2	level 3
			
			
			
			
			
			
			

level 0	level 1	level 2	level 3
			
			
			
			
			
			
			

level 0	level 1	level 2	level 3
			
			
			
			
			
			
			
			



level 0	level 1	level 2	level 3
			
			
			
			
			
			
			



# Conclusions

---

In this dissertation, we have developed a framework for representing, learning and detecting visual features hierarchies in images. The effectiveness of this method has been tested on several increasingly challenging experiments.

This chapter first presents in Section 8.1 a high-level summary of the main contributions that have been proposed to fulfill our initial aims. Object recognition has been the central task considered in this thesis. However, an interesting property of the feature representation developed in this work is to be generic. Many open problems in computer vision (tracking, 3D object recognition, segmentation, ...) could be addressed using a very similar representation. As it will be explained in Section 8.2, a few extensions have recently been made to our framework. Finally, Section 8.3 provides a few promising directions that could be followed for future research.

## 8.1 Summary of the Contributions

The main contribution of this thesis consists in the introduction of a new framework to represent visual features at various levels of complexity. An interesting property is to bring together hierarchical and structural (*i.e.* geometrical) aspects of high-level features. The model essentially combines several key concepts that have been developed the last couple of years in computer vision, machine learning and computational neurosciences: spatial relations between local visual features [Sch96, Pia01], graphical models [Pea88, PFZ03], and hierarchies of complex cells [FMI83, RP99]. This results in a hierarchical Pairwise Markov Random Field (PMRF) representation

of increasingly complex visual features [SP05, SP06].

An essential property coming from the graphical model formalism is that it allows us to pose detection as an inference problem. However, if performed exactly, such a probabilistic inference quickly becomes untractable. To solve this problem, the most popular methods [PFZ03, CFH05] often add artificial restrictions (*e.g.* number of parts, viewpoint, rotation, ...) to their models. In contrast with these approaches, the proposed scheme avoids these restrictions by exploiting the recently developed Nonparametric Belief Propagation (NBP) which provides a more efficient way to perform inference. To the best of my knowledge, the research presented in this thesis has been the first to exploit these mechanisms for recognizing objects. Note that different methods similar in spirit have been developed afterwards [OB06]. The results obtained by these methods strengthened the idea of representing visual features in a hierarchical way.

The representation of visual features is one thing, learning is another. In many current object recognition frameworks, the structure of the model is defined a priori. This greatly simplifies the learning task. Contrasting with these methods [PFZ03], we have shown that the probabilistic structure of the object model itself can be learned in an iterative manner by analyzing the co-occurrence statistics of local features.

This dissertation has introduced the following contributions:

- An extensive study of state-of-the-art feature detection and description methods. Their performances in terms of repeatability were evaluated across different image variations (illumination, blur, viewpoint change).
- A novel hierarchical representation of visual features. It uses Pairwise Markov Random Fields to factorize the object model in terms of local spatial relations between features. Such a model also provides a natural way to represent appearance and shape separately (by nodes and edges).
- A co-occurrence method to learn the structure of the graphical model in a bottom-up fashion.
- The integration of Nonparametric Belief Propagation (NBP) to detect feature hierarchies in images.
- A new kind of Adaptive Patch Features to represent the appearance of high-level features. Their width and height are automatically determined using an optimality criterion based on minimum-variance analysis.

- A way to exploit feature hierarchies to perform object recognition.

## 8.2 Extensions

The extent of the possible applications of the proposed scheme goes well beyond object recognition tasks. Many problems in computer vision require the recognition of high-level visual features. In this section, we present three extensions that have been developed recently, namely *Multidimensional Feature Hierarchies*, *Reinforcement Learning*, and *Tracking with Feature Hierarchies*. We found preferable not to include a detailed discussion about these topics. More informations can be found in the corresponding references.

### 8.2.1 Multidimensional Feature Hierarchies

The most straightforward way to extend the current work is to generalize the feature locations and their relationships to higher dimensions. A first attempt was explored by DETHIER [Det05] and has been developed recently by DETRY AND PIATER [Det06, DP07]. In their research, they extend the current method to be able to detect features located in a 3D space. For this purpose, our 2D low-level features are replaced by oriented feature patches in 3-space, annotated by various appearance characteristics [KW04]. The beauty of this extension relies on its general definition of both feature location and spatial relations. The distributions estimated on these high-dimensional spaces are defined in a fully nonparametric way.

Similarly to this thesis, a Pairwise Markov Random Field (PMRF) is exploited to factorize the statistical dependence between the visual features of the object. Nonparametric Belief Propagation (NBP) is a natural choice for detection in such a graphical model. NBP is used to infer the presence of a 3D object model from a scene represented as a set of observed 3D features. To learn an object representation, sets of 3D features are constructed using structure-from-motion techniques. Similarly to our model, a hierarchical object representation is then iteratively constructed by combining stable 3D configurations.

That work is illustrated on the application of object pose estimation. Object models are learned from a given world reference frame, within which the object is placed in a reference pose. Comparing an instance of the model in an unknown scene with an instance in the learned scene allows to estimate the object pose parameters

in the unknown scene. In addition, promising results have demonstrated that non-visual features (grasping strategy) can naturally be included in the hierarchies and inferred in presence of images not previously seen.

## 8.2.2 Reinforcement Learning

In the previous chapters, we demonstrated that visual feature hierarchies can be learned from co-occurrence statistics. A strong motivation behind co-occurrence learning originates from neuroscience findings. Alternatively, it has also been shown that human beings learn to extract useful information in an *interactive fashion* [GS83]. By evaluating the consequence of certain actions on the environment, we learn to focus our attention to visual features that are behaviorally relevant for solving a given task. This way, as we interact with the outside world, we gain more and more expertise on our tasks [TC03].

Inspired by these observations, *Reinforcement Learning* (RL) is a formal method that models the behavior of an artificial agent that learns how to perform a task through its interactions with the environment [BT96, SB98]. In RL, the agent learns to connect its sensory inputs to the appropriate actions. Contrary to supervised learning, it is not told what action it should take; rather, when it does a good or a bad action, it only receives a reward, the *reinforcement signal*.

JODOGNE *et al.* have recently proposed the *Reinforcement Learning of Visual Classes* (RLVC) Algorithm [JP05c] to apply RL to visual problems where a visual perception-action mapping has to be estimated. In RLVC, an image classifier is presented to a classical RL algorithm. This classifier partitions the perceptual space into a finite set of distinct regions according to local features, by focusing the attention of the agent on highly distinctive visual features. RLVC iteratively refines an image classifier by successively selecting new visual features.

In joint work with JODOGNE [JSP05], we have shown how visual feature hierarchies can be exploited to perform a visual task using only reinforcement feedback. The RLVC algorithm that originally makes use of individual features was modified to use a simplified version of our hierarchies. Spatial combinations of visual features were only defined by a distance between features. We demonstrated the efficacy of our algorithm on a version of the classical “Car on the Hill” control problem where position and velocity are presented to the agent visually, in a way that the task is unsolvable using individual point features.

As it has been mentioned [Jod06], it would be very interesting to take the rela-

tive orientations between pairs of lower-level visual features into account in composite features, instead of the distance alone. More detailed informations about RLVC can be found in the following publications [JP05a, JP05b, JP05c, JP06, JBP06].

### 8.2.3 Tracking with Feature Hierarchies

The development of affordable digital video camera has recently boosted the interest in video analysis. Automatic processing and interpretation of those data is currently still very challenging for state-of-the-art systems. Among the possible applications are surveillance, medical assistance, traffic management, and interactive environments. The interpretation of the video in these high-level applications generally relies on object tracking.

Similarly to the object recognition problem, object tracking can be posed as inference in a graphical model. The main difference is to add temporal constraints in the model to be able to efficiently locate the object of interest across the frames of the video sequence.

Tracking applications often have to deal with issues common to object recognition such as occlusions, clutter, as well as viewpoint and appearance variations. Therefore, it is not surprising that there is a large overlap between tracking and recognition techniques. For instance, to be robust enough to deal with common variations and occlusions, a possible strategy for tracking algorithms is to combine different local features. In parallel to our framework, similar feature hierarchies have been used to track objects.

The object models can be defined offline using a initial video sequence or manually pre-defined parameters. SIGAL *et al.* [SISB03, SBR<sup>+</sup>04, SZCB04, BSIB04] have developed a tracking framework that uses Nonparametric Belief Propagation to perform 3D human body tracking. In their model, the human body structure is defined by spatial relations between limbs (arms, legs and head).

In a similar direction, DU AND PIATER [DP06b, DP06c] have used sequential belief propagation [HW04] to fuse information in a simple hierarchical model. Real-time computational requirements of tracking systems does not currently allows to track many features at the same time. Most often the models are made of only a couple of features. Nevertheless hierarchical organizations of these local features have demonstrated their accuracy.

Tracking results can be exploit by online learning methods to improve the object model. DECLERCQ *et al.* [DP06a] have recently developed a computational model

to estimate the parameters in an incremental fashion.

## 8.3 Suggestions for Future Research

We conclude by presenting promising lines of research suggested by our visual feature hierarchies. The three main topics considered here are the learning strategy (Section 8.3.1), the representation (Section 8.3.2), and the ideas related to the appearance of the features (Section 8.3.3). Each topic is explored through open questions that have arose in this thesis. In addition, we briefly give a few suggestions for how they might be addressed.

### 8.3.1 Learning Pertinent Combinations

As we already mentioned, one of the main contributions of this thesis was the introduction of a new hierarchical feature representation. The automatic construction of such a hierarchy opens a large number of questions that remain to be addressed; What is the best strategy to learn the structure of the hierarchy? How dense should the models be? Which criterion should be used to score the feature combinations? How can we overcome the limitations of the current incremental learning strategy?

The automatic and incremental learning of new efficient feature combinations has already received some attention and appears to be a very promising directions [AG99, PG00b, Ope06, FBL06]. It allows to produce features that are not directly observable in the image and offer more distinctiveness. However, such a learning still remains a very challenging problem in computer vision.

If we turn to humans, it is now widely accepted that we are able to learn new features when we face with a novel recognition task. Although, the underlying mechanisms of this automatic learning remain unclear, different factors can be identified.

On the one hand, it has been shown that co-occurrences play an important role for the learning of new features. On the other hand, in order to perform given tasks, humans can identify features even if they appear only once.

We observed that co-occurrence based learning strategy is appropriate when repeatable features can be identified. However for more challenging tasks such as the recognition of object categories, it is often more difficult to identify useful feature combinations only from co-occurrences.

Instead of using such a criterion, a discriminant measure could be envisaged to select the feature combinations. Those features can be used to improve the



recognition performance. Unimportant features will be eliminated. Among them we can mention *Fisher score*, *Chi-Square*, *Likelihood*, and *Mutual Information* [Dor06]. It would be interesting to evaluate the performance of different strategies to combine co-occurrence based and discriminative approaches.

The structural learning is not the only task the system should perform. The parameters of the hierarchies have also to be estimated. Here also discriminative estimation of the parameters can be envisaged. Discriminative measures can not only be used to find good feature compositions, but they can also be exploited to estimate the parameters of the potentials.

### 8.3.2 Improving Feature Hierarchies

The proposed representation has been the first attempt to exploit a graphical model formalism for visual feature hierarchies. The model still remains basic. To improve the effectiveness of the model, a few usefull directions could be followed.

In a graphical model, the performance during inference essentially originates from two sources: the priors, and the observations. Priors are previously set by hand or learned, and are represented by the probabilistic structure of the model through the presence or absence of potential functions (*i.e.* edges) in the graph. The other factors come from the ability of the system to exploit the observations or measurements through likelihood functions.

Our model sees the image as a set of local features of which the appearance is exploited through the likelihood to add evidence in the graph. As other researchers [Ope06], we observed that the simultaneous use of different kind of features improve the performances of the system during recognition. Therefore, it seems logical to evaluate the strength of other type of features such as contour segments [FJS07] and Gabor filters [SWP05].

Another way to improve the effectiveness of the models would be to exploit direct measurements of spatial relations between features. More precisely, this can be achieved by introducing an observable term  $y_{ij}$  in the spatial relation potentials  $\psi_{ij}(x_i, x_j, y_{ij})$ . Such a topology has been recently used in stereo vision [SP07]. The message during Belief Propagation (BP) would be expressed as

$$m_{i,j}(x_j) \leftarrow \int \psi_{i,j}(x_i, x_j, y_{ij}) \phi_i(x_i, y_i) \prod_{k \in \mathcal{N}_i \setminus j} m_{k,i}(x_i) dx_i \quad (8.1)$$

where  $\mathcal{N}_i$  is the set of neighbors of node  $i$ ,  $\psi_{i,j}(x_i, x_j, y_{ij})$  is the pairwise potential

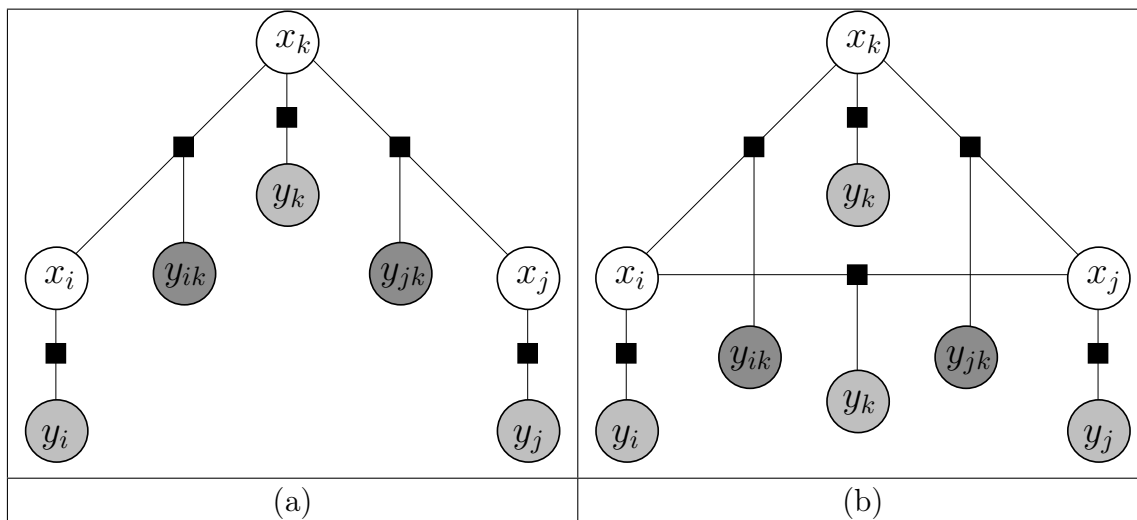


Figure 8.1: Two extensions of the hierarchical Model. (a) One way to improve the current hierarchy is to make the pairwise potential function observable  $\psi_{i,j}(x_i, x_j, y_{ij})$ . (b) Spatial relations between features of the same level might also indirectly enhance the detection of higher-level features.

between nodes  $i, j$ , and  $\phi_i(x_i, y_i)$  is the local observation potential of the current feature.

Figure 8.1 illustrates two different extensions of our hierarchical model using this new kind of pairwise potential. The first graph (a) shows a straightforward modification where each pairwise potential  $\psi_{i,j}(x_i, x_j)$  is replaced by its observable version  $\psi_{i,j}(x_i, x_j, y_{ij})$ . We could also envisaged to add edges between nodes at the same level. This would probably fasten the convergence.

In more practical terms, an easy way to boost the performance of the system would be to extent our pairwise spatial relations to  $n$ -tuple relations of features. We could also imagine to combine features from different levels in the hierarchy. These extensions are shown in Figure 8.2.

Despite the several advantages offered by Pairwise Markov Random Fields, they are not the best choice for every applications. For instance, Conditional Random Fields (CRF) [QCD04, WQMD06] or Discriminant Random Fields (DRF) [KH03] might be more attractive for supervised learning scenarios. Exploring this direction, KUMAR *et al.* [KH05, Kum05] have recently proposed a promising scene interpretation framework. It would be interesting to compare the different properties between CFR, DRF, and PMRF.

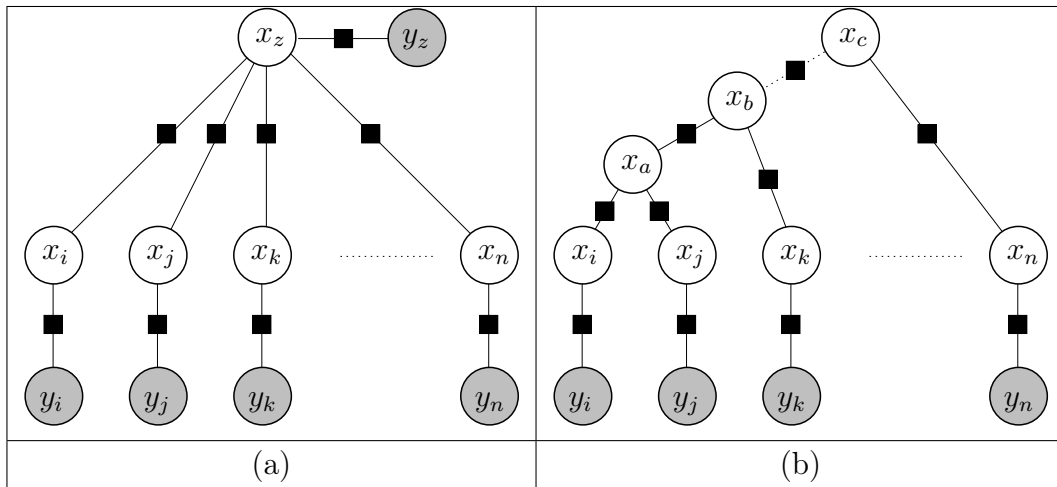


Figure 8.2: Two extensions for the topology of a hierarchy. (a) Combinations made from more than pairs of features would certainly give more selectivity. In (b), features from different levels are combined.

### 8.3.3 Appearance Model

The main weakness of our learning method is that the initial set of feature classes is fixed using a clustering method. Therefore the appearance variability that the model can handle is rather limited. A better solution would be to learn the specific variations in appearance for each composition.

Moreover there exists a relation between the appearance of the features and their spatial relations. The system could try to learn and represent these dependencies in probabilistic terms through more complex potential functions. For instance, by knowing these relations, the system could predict self-shadowing effects that often occurs on objects from the position and pose of given features.

Object recognition is a field that is currently moving very fast in many directions. It receives a large attention from the computer vision community. We can observe that hierarchical approaches [OB06, EU07] have recently become popular. Although they are still outperformed on categorization tasks by brute force approaches [ZMLS07], such as bag-of-features, dedicated to obtain the best short term results. I believe that the development of hierarchical models together with generic learning method is the most promising long term approach. It will help to develop biologically motivated models of recognition and to understand the functioning of biological system.



---

---

## Bibliography

---

- [AG99] Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999. 65, 196
- [Agi72] G. Agin. *Representation and description of curved objects*. PhD thesis, Stanford University, San Francisco, CA, USA, 1972. 59
- [AR02] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *European Conference on Computer Vision (ECCV)*, volume 4, pages 113–130, 2002. 64
- [AT06] A. Agarwal and B. Triggs. Hyperfeatures - multilevel local coding for visual recognition. In *European Conference on Computer Vision (ECCV)*, pages 30–43, 2006. 67
- [Bar89] H. B. Barlow. Unsupervised learning. *Neural Computation*, 1:295–311, 1989. 115
- [Bar94] H. B. Barlow. What is the computational goal of the neocortex? In *Large Scale Neuronal Theories of the Brain*, pages 1–22. MIT Press, Cambridge, MA, 1994. 115
- [Bau00] A. Baumberg. Reliable feature matching across widely separated views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1774–1781, 2000. 34, 46
- [BBM05] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26–33, 2005. 69

- [Bea78] P. R. Beaudet. Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, pages 579–583, 1978. 27, 35
- [BG05] C. BenAbdelkader and P. Griffin. A local region-based approach to gender classification from face images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, page 52, 2005. 171
- [BGV92] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learning Theory*, pages 144–152, 1992. 109, 137
- [Bin71] T. O. Binford. Visual perception by computer. In *IEEE Computer Society Conference on Systems and Control*, 1971. 59
- [Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, November 1995. 62
- [Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Secaucus, NJ, USA, 2006. 8
- [BL02] M. Brown and D. G. Lowe. Invariant features from interest point groups. In *British Machine Vision Conference (BMVC)*, pages 656–665, 2002. 35, 54
- [Blu67] H. Blum. A Transformation for Extracting New Descriptors of Shape. In *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967. 39
- [BMP77] L. Breiman, W. Meisel, and E. Purcell. Variable kernel estimates of multivariate densities. In *Technometrics*, volume 19, pages 135–144, 1977. 237, 239
- [BMP00] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in Neural Information Processing Systems (NIPS)*, pages 831–837, 2000. 49, 78

## BIBLIOGRAPHY

---

- [BMP01] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, pages 454–463, 2001. 50
- [BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. 50
- [BNJ03] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. 64
- [Bou05] G. Bouchard. *Generative models in supervised statistical learning with applications to digital image categorization and structural reliability*. PhD thesis, Institut National Polytechnique de Grenoble (INPG), INRIA Rhône–Alpes, France, january 2005. 67
- [BP93] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(10):1042–1052, 1993. 65
- [BP96] M. C. Burl and P. Perona. Recognition of planar object classes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, page 223, 1996. 65, 84
- [Bro05] M. Brown. *Multi-Image Matching Using Invariant Features*. PhD thesis, University of British Columbia, Vancouver, Canada, 2005. 54
- [BSIB04] S. Bhatia, L. Sigal, M. Isard, and M. J. Black. 3D human limb detection using space carving and multi-view eigen models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, page 17, 2004. 195
- [BSW05] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–517, 2005. 53
- [BT96] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996. 194

- [BT05] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 710–715, 2005. 67, 68, 112
- [BTVG06] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision (ECCV)*, pages 404–417, May 2006. 52
- [Bur96] M. C. Burl. *Recognition of visual object classes*. PhD thesis, California Institute of Technology, Pasadena, California, USA, 1996. 65
- [BWL02] J. W. Buzydlowski, H. D. White, and X. Lin. Term co-occurrence analysis as an interface for digital libraries. In *JCDL Workshop at Visual Interfaces to Digital Libraries*, pages 133–144, 2002. 115
- [BWP98] M. C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 628–641, 1998. 65
- [Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6):679–698, 1986. 39, 50
- [Car04] G. Carneiro. *Image Pattern Recognition Using Phase-Based Local Features And Their Flexible Spatial Configurations*. PhD thesis, University of Toronto, Canada, 2004. 47
- [Cat04] K. Cater. *Detail to Attention: Exploiting Limits of the Human Visual System for Selective Rendering*. PhD thesis, University of Bristol, Bristol, England, October 2004. 2
- [CCH01] C. C. Chang, C. S. Chan, and Ju Yuan Hsiao. A color image retrieval method based on local histogram. In *IEEE Pacific Rim Conference on Multimedia*, pages 831–836, 2001. 48
- [CdVC98] V. Colin de Verdière and J. L. Crowley. Visual recognition using local appearance. In *European Conference on Computer Vision (ECCV)*, volume 1, pages 640–654, June 1998. 56



## BIBLIOGRAPHY

---

- [CFH05] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10–17, 2005. 68, 112, 144, 192
- [Cha02] S. K. Chalup. Incremental learning in biological and machine learning systems. *International Journal of Neural Systems*, 12(6):447–465, 2002. 113
- [Cho57] C. K. Chow. An optimum character recognition system using decision functions. In *IRE Transactions on Electronic Computers*, volume EC6, pages 247–254, 1957. 58
- [CJ02] G. Carneiro and A. D. Jepson. Local phase-based features. In *European Conference on Computer Vision (ECCV)*, volume 1, pages 282 – 296, 2002. 46
- [CJ03] G. Carneiro and A. D. Jepson. Multi-scale local phase-based features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 414–420, 2003. 46
- [CL01] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. 171, 173
- [CL06] Y. Chen and C. Lin. Combining SVMs with various feature selection strategies. In *Feature extraction, foundations and applications*. Springer, 2006. 137, 139
- [Cli90] P. Clifford. Markov random fields in statistics. In *Disorder in Physical Systems. A Volume in Honour of John M. Hammersley*, pages 19–32. Oxford University Press, 1990. 11, 12
- [CLS05] S. Chen, B. Lovell, and T. Shan. Combining generative and discriminative learning for face recognition. In *Proceedings of the Digital Image Computing on Techniques and Applications (DICTA)*, page 5, 2005. 110
- [CMO03] O. Chum, J. Matas, and S. Obdržálek. Epipolar geometry from three correspondences. In *Computer Vision Winter Workshop (CVWW)*, pages 83–88, February 2003. 36

- [Coo89] P. R. Cooper. *Parallel object recognition from structure (the tinkertoy project)*. PhD thesis, University of Rochester, Rochester, New York, 1989. 67
- [CTB<sup>+</sup>99] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *International Conference on Visual Information Systems*, volume 1614, pages 509–516, 1999. 48
- [CV01] F. Camastra and A. Vinciarelli. Intrinsic dimension estimation of data: An approach based on grassberger procaccia’s algorithm. *Neural Processing Letters*, 14(1):27–34, 2001. 57
- [CWN04] B. Caputo, C. Wallraven, and M. E. Nilsback. Object categorization via local kernels. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 132–135, 2004. 63
- [DDF<sup>+</sup>90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. In *Journal of the American Society For Information Science*, volume 41, pages 391–407, 1990. 64
- [Det05] G. Dethier. *Apprentissage de Caractéristiques Visuelles en 3D*. Mémoire de DEA, University of Liège, Belgium, May 2005. 55, 193
- [Det06] R. Detry. *Learning Multidimensional Feature Hierarchies*. Master thesis, University of Liège, Belgium, May 2006. 102, 193
- [DFL<sup>+</sup>88] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285, New York, NY, USA, 1988. ACM Press. 64
- [DFP97] M. Fleck D. Forsyth, J. Malik and J. Ponce. Primitives, perceptual organization and object recognition. In *Vision Research (special issue on Models of Recognition)*, February 1997. 59
- [DGM98] R. Deriche, V. Gouet, and P. Montesinos. Differential invariants for color images. In *International Conference on Pattern Recognition (ICPR)*, page 21, 1998. 30

## BIBLIOGRAPHY

---

- [Din55] G. P. Dinneen. Programming pattern recognition. In *Western Joint Computer Conference*, pages 94–100, 1955. 58
- [DJNM98] C. L. Blake D. J. Newman, S. Hettich and C. J. Merz. UCI repository of machine learning databases, 1998. 121
- [Dor06] G. Dorkó. *Selection of Discriminative Regions and Local Descriptors for Generic Object Class Recognition*. PhD thesis, Institut National Polytechnique de Grenoble (INPG), INRIA Rhône–Alpes, France, june 2006. 197
- [DP06a] A. Declercq and J. H. Piater. On-line simultaneous learning and tracking of visual feature graphs. In *Workshop at IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2006. 195
- [DP06b] W. Du and J. H. Piater. Data fusion by belief propagation for multi-camera tracking. In *International Conference on Information Fusion (ICIF)*, pages 1–8, 2006. 195
- [DP06c] W. Du and J. H. Piater. Multi-view object tracking using sequential belief propagation. In *Asian Conference on Computer Vision (ACCV)*, pages 684–693, 2006. 195
- [DP07] R. Detry and J. H. Piater. Hierarchical integration of local 3d features for probabilistic pose recovery. In *Robot Manipulation: Sensing and Adapting to the Real World (Workshop at Robotics, Science and Systems)*, 2007. 88, 193
- [DS05] G. Dorkó and C. Schmid. Object class recognition using discriminative local features. Rapport de recherche RR-5497, INRIA - Rhone-Alpes, February 2005. 62
- [DSD<sup>+</sup>04] M. Demirci, A. Shokoufandeh, S. Dickinson, Y. Keselman, and L. Bretzner. Many-to-many feature matching using spherical coding of directed graphs. In *European Conference on Computer Vision (ECCV)*, May 2004. 70
- [DSK<sup>+</sup>05] S. J. Dickinson, A. Shokoufandeh, Y. Keselman, M. F. Demirci, and D. Macrini. Object categorization and the need for many-to-many matching. In *DAGM-Symposium*, pages 501–510, 2005. 70

- [DWF<sup>+</sup>04] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004. 62
- [EC04] J. Eichhorn and O. Chapelle. Object categorization with SVM: Kernels for local features. Technical Report 137, Max Planck Institute for Biological Cybernetics, 07 2004. 53
- [EG98] J. H. Elder and R. M. Goldberg. Inferential reliability of contour grouping cues in natural images. *Perception*, 27(11), 1998. 114
- [EHYI01] S. Edelman, B. P. Hiles, H. Yang, and N. Intrator. Probabilistic principles in unsupervised learning of visual structure: human data and a model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 19–26, 2001. 115
- [Ett88] G. J. Ettinger. Hierarchical object recognition using libraries of parameterized model sub-parts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, page 32, 1988. 67
- [EU05a] B. Epshtein and S. Ullman. Feature hierarchies for object classification. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, pages 220–227, 2005. 67, 68, 69
- [EU05b] B. Epshtein and S. Ullman. Identifying semantically equivalent object fragments. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2–9, 2005. 69
- [EU07] B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2007. 199
- [EV01] S. Edelman and L. M. Vaina. *David Marr (a short biography)*. International Encyclopaedia of Social and Behavioral Sciences, Pergamon, 2001. 60
- [EZWVG] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. <http://www.pascal-network.org/challenges/VOC/>. 179

## BIBLIOGRAPHY

---

- [FA91] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(9):891–906, September 1991. 45, 78
- [FA01] J. Fiser and R. N. Aslin. Unsupervised statistical learning of higher-order spatial structures from visual scenes. *Psychological Science*, 12(6):499–504, November 2001. 114
- [FA02] J. Fiser and R. N. Aslin. Statistical learning of new visual feature combinations by infants. *National Academy of Sciences USA*, 99(24):15822–15826, November 2002. 114
- [FA05] J. Fiser and R. N. Aslin. Encoding multielement scenes: statistical learning of visual feature hierarchies. *Journal of Experimental Psychology General*, 4(134):521–537, November 2005. 114
- [FBL06] S. Fidler, G. Berginc, and A. Leonardis. Hierarchical statistical learning of generic parts of object structure. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 182–189, June 2006. 67, 196
- [FE73] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973. 65
- [Fer05] R. Fergus. *Visual Object Category Recognition*. PhD thesis, University of Oxford, England, December 2005. 59, 65, 157
- [FH05] P. F. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005. 66
- [Fis89] R. B. Fisher. *From surfaces to objects: computer vision and three dimensional scene analysis*. John Wiley & Sons, Inc., New York, NY, USA, 1989. 85
- [FJS07] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007. 197

- [FK03] M. Felsberg and N. Krüger. A probabilistic definition of intrinsic dimensionality for images. In *DAGM Symposium Mustererkennung, Magdeburg*, volume 2781 of *LNCS*, pages 140–147. Springer, Heidelberg, 2003. 57
- [FM98] B. J. Frey and D. J. C. MacKay. A revolution: Belief propagation in graphs with cycles. In *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998. 18
- [FMI83] K. Fukushima, S. Miyake, and T. Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 13:826–834, 1983. 86, 191
- [För86] W. Förstner. A feature based correspondence algorithm for image matching. In *International Architecture Photogrammetry and Remote Sensing*, volume 24, pages 160–166, 1986. 27
- [FP05] L. FeiFei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 524–531, 2005. 40, 64
- [FPZ03] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 264–271, June 2003. 65, 68, 70, 84, 91, 102, 112, 144
- [FPZ05] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 380–387, 2005. 66, 68
- [FR98] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998. 121
- [FS93] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26(1):167–174, 1993. 56, 78

## BIBLIOGRAPHY

---

- [FS97] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. 63
- [FTG04] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *European Conference on Computer Vision (ECCV)*, pages 40–54, May 2004. 159, 160
- [FTVG06] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *Int. J. Comput. Vision*, 67(2):159–188, 2006. 146
- [Fuk80] K. Fukushima. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. In *Biological Cybernetics*, volume 36, pages 193–202, 1980. 70
- [GD05] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, pages 1458–1465, 2005. 63
- [GEW06] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006. 63
- [GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6:721–741, 1984. 21
- [GH03] C. Elkan G. Hamerly. Learning the k in k-means. *International Conference on Machine Learning (ICML)*, 2003. 121
- [GHP07] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. 179
- [GL96] J. Gårding and T. Lindeberg. Direct computation of shape cues using scale-adapted spatial derivative operators. *International Journal of Computer Vision (IJCV)*, 17(2):163–191, 1996. 28

- [GM04] G. H. Granlund and A. Moe. Unrestricted recognition of 3-D objects for robotics using multi-level triplet invariants. *Artificial Intelligence Magazine*, 25(2):51–67, 2004. 54, 55
- [GMDP00] V. Gouet, P. Montesinos, R. Deriche, and D. Pelé. Evaluation de détecteurs de points d'intérêts pour la couleur. In *RFIA*, volume 2, pages 257–266, Paris, France, 2000. 30
- [GMP98] V. Gouet, P. Montesinos, and D. Pelé. Stereo matching of color images using differential invariants. In *Proceedings of the IEEE International Conference on Image Processing*, pages 152–156, Chicago, Etats-Unis, 1998. 44
- [GMU96] L. Van Gool, T. Moons, and D. Ungureanu. Affine/ photometric invariants for planar intensity patterns. In *European Conference on Computer Vision (ECCV)*, pages 642–651, 1996. 56
- [Gou00] V. Gouet. *Mise en correspondance d'images en couleur - Application à la synthèse de vues intermédiaires*. PhD thesis, Université de Montpellier II, Montpellier, France, october 2000. 30
- [GPSG01] W. S. Geisler, J. S. Perry, Boaz J. Super, and D. P. Gallogly. Edge co-occurrence in natural images predicts contour grouping performance. *Vision Research*, 41:711–724, 2001. 114, 115
- [Gra06] K. Grauman. *Matching Sets of Features for Efficient Retrieval and Recognition*. PhD thesis, Massachusetts Institute of Technology (MIT), Boston, MA , USA, September 2006. 157
- [GS83] E. J. Gibson and E. S. Spelke. The development of perception. In John H. Flavell and Ellen M. Markman, editors, *Handbook of Child Psychology Vol. III: Cognitive Development*, chapter 1, pages 2–76. Wiley, 4th edition, 1983. 194
- [GS99] N. Gagvani and D. Silver. Parameter-controlled volume thinning. *CVGIP: Graph. Models Image Process.*, 61(3):149–164, 1999. 39
- [GSTK58] R. L. Grimsdale, F. H. Sumner, C. J. Tunis, and T. Kilburn. A system for the automatic recognition of patterns. *Proceedings of the Institution of Electrical Engineers*, B: 106:210–221, 1958. 58



## BIBLIOGRAPHY

---

- [GSvdB01] J. M. Geusebroek, A. W. M. Smeulders, and R. van den Boomgaard. Measurement of color invariants. In A. M. Vossepoel and F. M. Vos, editors, *Fourth Quinquennial Review 1996-2000*, pages 129–138. Dutch Society for Pattern Recognition and Image Processing, 2001. 44
- [GT97] I. Gauthier and M. J. Tarr. Becoming a "greeble" expert: exploring mechanisms for face recognition. In *Vision Research*, volume 37(12), pages 1673–1682, 1997. 55
- [Hal01] D. Hall. *Viewpoint independent recognition of objects from local appearance*. PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, October 2001. 44
- [Har83] J. A. Hartigan. *Bayes Theory*. Springer, New York, 1983. 14
- [Hof99] T. Hofmann. Probabilistic latent semantic indexing. In ACM Press, editor, *Annual International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 50–57, 1999. 64
- [Hof01] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Journal of Machine Learning Research*, 42(1-2):177–196, 2001. 64
- [HP98] T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1998. 115
- [HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of the 4th Alvey Vision Conference*, pages 147–151, University of Manchester (UK), August 1988. 27, 29, 30, 31, 35, 151
- [HSV90] R. Horaud, T. Skordas, and F. Veillon. Finding geometric and relational structures in an image. In *European Conference on Computer Vision (ECCV)*, pages 374–384, April 1990. 39
- [HW79] J. A. Hartigan and M. A. Wong. Statistical algorithms: A k-means clustering algorithm. *Journal of Applied Statistics*, 28(1):100–108, March 1979. 62, 116, 120

- [HW04] G. Hua and Y. Wu. Multi-scale visual tracking by sequential belief propagation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 826–833, 2004. 19, 195
- [HWP05] A. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object recognition. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, volume 1, pages 136–143, 2005. 110
- [HYW05] G. Hua, M. Yang, and Y. Wu. Learning to estimate human pose with data driven belief propagation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 747–754, 2005. 19
- [HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 74
- [IFMW04] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky. Nonparametric belief propagation for self-calibration in sensor networks. In *Proceedings of the International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 225–233, 2004. 23, 239
- [Ihl05] A. T. Ihler. *Inference in Sensor Networks: Graphical Models and Particle Methods*. PhD thesis, Massachusetts Institute of Technology (MIT), Boston, MA , USA, June 2005. 21, 239
- [Isa02] M. Isaksson. *Face Detection and Pose Estimation using Triplet Invariants*. Master thesis, Linköping University, Sweden, February 2002. 54
- [Isa03] M. Isard. Pampas: Real-valued graphical models for computer vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 613–620, 2003. 19, 20, 21, 98
- [JBP06] S. Jodogne, C. Briquet, and J. H. Piater. Approximate policy iteration for closed-loop learning of visual tasks. In *Proc. of the 17th European Conference on Machine Learning (ECML)*, September 2006. Accepted for publication. 195

## BIBLIOGRAPHY

---

- [JH99] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(5):433 – 449, May 1999. 48
- [JM05] B. Johansson and A. Moe. Patch-duplets for object recognition and pose estimation. In *Canadian conference on Computer and Robot Vision (CRV)*, pages 9–16, 2005. 55
- [Joa98] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *European Conference on Machine Learning (ECML)*, pages 137–142. Springer, 1998. 62
- [Jod06] S. Jodogne. *Closed-Loop Learning of Visual Control Policies*. PhD thesis, University of Liège, Liège, Belgium, December 2006. 29, 32, 74, 171, 194
- [Jor99] M. I. Jordan, editor. *Learning in graphical models*. MIT Press, Cambridge, MA, USA, 1999. 8
- [JP05a] S. Jodogne and J. H. Piater. Interactive learning of mappings from visual percepts to actions. In *International Conference on Machine Learning (ICML)*, pages 393–400, August 2005. 195
- [JP05b] S. Jodogne and J. H. Piater. Learning, then compacting visual policies. In *Proc. of the 7th European Workshop on Reinforcement Learning (EWRL)*, pages 8–10, Napoli (Italy), October 2005. 74, 76, 195
- [JP05c] S. Jodogne and J. H. Piater. Reinforcement learning of perceptual classes using  $Q$ -learning updates. In M.H. Hamza, editor, *Proc. of the 23rd IASTED International Multi-Conference on Artificial Intelligence and Applications*, pages 445–450, Innsbruck (Austria), February 2005. Acta Press. 194, 195
- [JP06] S. Jodogne and J. H. Piater. Task-driven discretization of the joint space of visual percepts and continuous actions. In *Proc. of the 17th European Conference on Machine Learning (ECML)*, September 2006. Accepted for publication. 195

- [JSP05] S. Jodogne, F. Scalzo, and J. H. Piater. Task-driven learning of spatial combinations of visual features. In *Workshop on Learning at IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005. 112, 194
- [JV96] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996. 48
- [Kan79] E. R. Kandel. Small systems of neurons. *Scientific American*, 241:67–76, September 1979. 113
- [KB01] A. Kadir, T. and M. Brady. Scale, saliency and image description. In *Proc. of the International Journal of Computer Vision*, pages 83–105, november 2001. 37
- [KD05] Y. Keselman and S. Dickinson. Generic model abstraction from examples. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1141–1156, 2005. 85
- [KFL01] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, feb 2001. 13
- [KFW04] N. Krüger, M. Felsberg, and F. Wörgötter. Processing multi-modal primitives from image sequences. In *International ICSE Symposium on Engineering of Intelligent Systems (EIS)*, 2004. 56
- [KGA02] S. Krempp, D. Geman, and Y. Amit. Sequential learning of reusable parts for object detection, 2002. 113
- [KH03] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 2003. 198
- [KH05] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, volume 2, pages 1284–1291, October 2005. 198

## BIBLIOGRAPHY

---

- [Kir03] K. Kirk. Spatial sampling and interpolation methods: Comparative experiments using simulated data. Technical report, Aalborg University, Copenhagen, September 2003. 41
- [KMN<sup>+</sup>02] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transaction on Pattern Analysis and Machin Intelligence (PAMI)*, 24(7):881–892, 2002. 151
- [KMN<sup>+</sup>04] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom. Theory Appl.*, 28(2-3):89–112, 2004. 151
- [Köh47] W. Köhler. *Gestalt Psychology*. Mentor and Liveright Publishing Corporation, New American Library, New York, USA, 1947. 61, 113
- [KPM06] M. Kelm, C. Pal, and A. McCallum. Combining generative and discriminative methods for pixel classification with multi-conditional learning. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 828–832, 2006. 110
- [KR93] R. E. Kass and A. E. Raftery. Bayes factors and model uncertainty. *Technical Report 571*, 1993. 121
- [Krü98] N. Krüger. Collinearity and parallelism are statistically significant second-order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129, 1998. 114, 115
- [KS80] R. Kinderman and J. Snell. *Markov Random Fields and their Applications*. American Mathematical Society, Providence, RI, USA, 1980. 11
- [KS04] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 506–513, June 2004. 52, 56, 78, 80
- [KSDD03] Y. Keselman, A. Shokoufandeh, M. Demirci, and S. Dickinson. Many-to-many graph matching via metric embedding. In *IEEE Com-*

- puter Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2003. 70
- [Kum05] S. Kumar. *Models for Learning Spatial Interactions in Natural Images for Context-Based Classification*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, August 2005. 198
- [KvD87] J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987. 44, 45, 78
- [KW95] R. Kass and L. Wasserman. A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association*, 90(431):928–934, 1995. 121
- [KW02] N. Krüger and F. Wörgötter. Multi-modal estimation of collinearity and parallelism in natural image sequences. *Network: Computation in Neural Systems*, 13:553–576, 2002. 115
- [KW04] N. Krüger and F. Wörgötter. Multi-modal primitives as initiators of recurrent disambiguation processes. In *ECOVISION Workshop*, Isle of Skye, 2004. 56, 193
- [KZB04] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *European Conference on Computer Vision (ECCV)*, pages 228–241, May 2004. 38
- [Lau96] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996. 8
- [Laz06] S. Lazebnik. *Local, Semi-Local And Global Models For Texture, Object And Scene Recognition*. PhD thesis, University of Illinois at Urbana-Champaign, Illinois, USA, May 2006. 48, 49, 52, 64, 157
- [LBBH98] Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 70
- [LCS06] T. N. Lal, O. Chapelle, and B. Schölkopf. *Combining a Filter Method with SVMs*, pages 441–447. Springer-Verlag, Berkeley, Southampton, Zürich, 2006. 137, 139

## BIBLIOGRAPHY

---

- [LG97] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure. *IVC*, 15(6):415–434, June 1997. 34
- [Lin98] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision (IJCV)*, 30(2):79–116, November 1998. 30, 31, 35, 119
- [LJ06] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization. In *British Machine Vision Conference*, 2006. 64
- [LM03] T. S. Lee and D. Mumford. Hierarchical bayesian inference in the visual cortex. *Journal of the Optical Society of America*, 20(7):1434–1448, July 2003. 23, 101
- [Low87] D. G. Lowe. The viewpoint consistency constraint. *International Journal of Computer Vision*, 1(1):57–72, March 1987. 59
- [Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, pages 1150–1157, September 1999. 35, 50, 56, 78
- [Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 50
- [LP99] Y. S. Lo and S. C. Pei. Color image segmentation using local histogram and self-organization of kohonen feature map. In *IEEE International Conference on Image Processing (ICIP)*, pages 232–235, 1999. 48
- [LS88] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 50(2):157–224, 1988. 16
- [LS99] E. Loupias and N. Sebe. Wavelet-based salient points for image retrieval, 1999. Technical Report RR 99.11, Laboratoire Reconnaissance de Formes et Vision, INSA Lyon. 38

- [LS03] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 409–415, June 2003. 48
- [LSD05] A. Levinshtein, C. Sminchisescu, and S. J. Dickinson. Learning hierarchical shape models from examples. In *EMMCVPR*, pages 251–267, 2005. 70
- [LSP04] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *British Machine Vision Conference (BMVC)*, pages 959–968, 2004. 55, 64, 127, 144, 146, 148, 167, 170, 174, 176, 177, 178
- [LSP06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, 2006. 63
- [LTGK06] Y. Li, Y. Tsin, Y. Genc, and T. Kanade. Statistical shape models for object recognition and part localization. In *British Machine Vision Conference (BMVC)*, Edinburgh, Scotland, September 2006. 112
- [LVB<sup>+</sup>93] M. Lades, J. C. Vorbrüggen, J. M. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42(3):300–311, 1993. 65
- [Lyu04] S. Lyu. Mercer kernels for object recognition with local features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 223 – 229, October 2004. 63
- [MAF<sup>+</sup>04] K. Morita, E. Atlam, M. Fuketra, K. Tsuda, M. Oono, and J. Aoe. Word classification and hierarchy using co-occurrence word information. *Information Processing and Management: an International Journal*, 40(6):957–972, 2004. 114
- [Mah36] P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India*, 12:49–55, 1936. 100



## BIBLIOGRAPHY

---

- [Mar82] D. Marr. *Vision*. Freeman, 1982. 84
- [Mar05] R. Marée. *Classification automatique d'images par arbres de décision*. PhD thesis, University of Liège, Liège, Belgium, February 2005. 63, 148
- [MBM05] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(11):1832–1837, 2005. 50
- [MCUP02] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference (BMVC)*, pages 384–393, 2002. 36, 75
- [Mel97] B. W. Mel. Seemore: combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Computation*, 9(4):777–804, 1997. 62, 70
- [MGA89] E. Mjolsness, G. Gindi, and P. Anandan. Optimization in model matching and perceptual organization. *Neural Computation*, 1:218–229, 1989. 67
- [MGD98] P. Montesinos, V. Gouet, and R. Deriche. Differential Invariants for Color Images. In *International Conference on Pattern Recognition (ICPR)*, pages 838–840, Brisbane, Australia, 1998. 44
- [MGPW04] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. A generic approach for image classification based on decision tree ensembles and local sub-windows. In *Asian Conference on Computer Vision (ACCV)*, volume 2, pages 860–865, 2004. 63, 159
- [MGPW05a] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Biomedical image classification with random subwindows and decision trees. In *Proc. ICCV workshop on Computer Vision for Biomedical Image Applications (CVIBA)*, volume 3765 of *LNCS*, pages 220–229, oct 2005. 63
- [MGPW05b] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Decision trees and random subwindows for object recognition. In *ICML workshop on Machine Learning Techniques for Processing Multimedia Content (MLMM2005)*, 2005. 63, 145

- [MGPW05c] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 34–40, June 2005. 40, 41, 53, 63, 174, 177
- [MH03] S. Mahamud and M. Hebert. The optimal distance measure for object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 146
- [Mik02] K. Mikolajczyk. *Detection of local features invariant to affines transformations*. PhD thesis, Institut National Polytechnique de Grenoble (INPG), INRIA Rhône–Alpes, France, July 2002. 30, 32, 34, 80, 168
- [ML06] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–18, 2006. 71
- [MMP04] P. Moreels, M. Maire, and P. Perona. Recognition by probabilistic hypothesis construction. In *European Conference on Computer Vision (ECCV)*, pages 55–68, 2004. 146
- [MN78] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Biological Science*, 200(1140):269–294, February 1978. 67
- [MN95a] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision (IJCV)*, 14:5–24, 1995. 56
- [MN95b] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision (IJCV)*, 14(1):5–24, 1995. 145
- [MOC02] J. Matas, Š. Obdržálek, and O. Chum. Local affine frames for wide-baseline stereo. In *International Conference on Pattern Recognition (ICPR)*, pages 363–366, 2002. 36
- [Mor77] H. P. Moravec. Towards automatic visual obstacle avoidance. In *International Joint Conference on Artificial Intelligence*, 1977. 27

## BIBLIOGRAPHY

---

- [MP77] D. Marr and T. Poggio. From understanding computation to understanding neural circuitry. In *Neurosciences Research Progress Bulletin*, volume 15, pages 470–488, 1977. 59, 60
- [MS01] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 525–531, 2001. 31, 32, 37, 75
- [MS02] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision (ECCV)*, pages 128–142, 2002. 28, 32, 75
- [MS03] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 257–263, June 2003. 46
- [MS04] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision (IJCV)*, 60(1):63–86, 2004. 32, 119
- [MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630, 2005. 52, 78, 80
- [MSTC94] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell. *Machine learning, neural and statistical classification*. Ellis Horwood, Upper Saddle River, NJ, USA, 1994. 121
- [MTS<sup>+</sup>05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(7):43–72, November 2005. 75, 168
- [Mun98] J. L. Mundy. Object recognition based on geometry: progress over three decades. In Physical Series A (Mathematical and Engineering Sciences), editors, *Philosophical Transactions of the Royal Society London*, volume 356, pages 1213–1231, 1998. 59, 60, 71, 72

- [Mun03] J. L. Mundy. Object recognition in the geometric era: a retrospective. In *Physical Series A (Mathematical and Engineering Sciences)*, editors, *Designing Tomorrow's Category-Level 3D Object Recognition Systems: An International Workshop*, September 2003. 59, 72
- [MWJ99] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the fifteenth conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999. 18
- [NC04] M. Nilsback and B. Caputo. Cue integration through discriminative accumulation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 578–585, 2004. 171
- [NJT06] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision (ECCV)*, pages 490–503, 2006. 173
- [NMN94] S. K. Nayar, H. Murase, and S. A. Nene. Learning, positioning, and tracking visual appearance. In *Proc. of the International Conference on Robotics and Automation*, May 1994. 56
- [NNM96] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical Report CUCS-005-96, 1996. xiii, 145, 146, 150, 157, 161, 164
- [OB06] B. Ommer and J. M. Buhmann. Learning compositional categorization models. In *European Conference on Computer Vision (ECCV)*, volume 3, pages 316–329, 2006. 67, 192, 199
- [OD04] G. C. Oana and K. J. Dana. 3D texture recognition using bidirectional feature histograms. *International Journal of Computer Vision (IJCV)*, 59(1):33–60, 2004. 48
- [OFPA04] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *European Conference on Computer Vision (ECCV)*, pages 71–84, 2004. 63, 78, 144

## BIBLIOGRAPHY

---

- [OI97] K. Ohba and K. Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):1043–1048, 1997. 56
- [OM02a] Š. Obdržálek and J. Matas. Local affine frames for image retrieval. In *Proc. of the International Conference on Image and Video Retrieval*, pages 318–327, 2002. 36, 119
- [OM02b] Š. Obdržálek and J. Matas. Object recognition using local affine frames on distinguished regions. In *British Machine Vision Conference (BMVC)*, 2002. 145
- [OP05] A. Opelt and A. Pinz. Object localization with boosting and weak supervision for generic object recognition. In *SCIA*, pages 862–871, 2005. 63, 78
- [Ope06] A. Opelt. *Generic Object Recognition*. PhD thesis, Graz University of Technology, Graz, Austria, March 2006. 63, 64, 71, 108, 157, 196, 197
- [OPFA06] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. In *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, volume 28, page 3, March 2006. 118
- [OPZ06] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3–10, 2006. 64, 113, 120
- [Pap91] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991. 62
- [Par62] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33:1065–1076, September 1962. 237
- [Pea88] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1988. 16, 18, 19, 86, 191

- [Pen90] A. Pentland. Part segmentation for object recognition. *Neural Computation*, 1(1):82–91, 1990. 67
- [PFZ03] P. Perona, R. Fergus, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 264, June 2003. 86, 191, 192
- [PG99] J. H. Piater and R. A. Grupen. Toward learning visual discrimination strategies. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1415, 1999. 113
- [PG00a] J. H. Piater and R. A. Grupen. Constructive feature learning and the development of visual expertise. In *International Conference on Machine Learning (ICML)*, pages 751–758, 2000. 113
- [PG00b] J. H. Piater and R. A. Grupen. Distinctive features should be learned. In *International Conference on Biologically Motivated Computer Vision*, pages 52–61. Springer-Verlag, 2000. 15, 91, 196
- [PH03] D. Pritchard and W. Heidrich. Cloth motion capture. *Eurographics*, 22(3), 2003. 50
- [Pia01] J. H. Piater. *Visual Feature Learning*. PhD thesis, University of Massachusetts, Computer Science Department, Amherst (MA, USA), February 2001. 45, 67, 86, 102, 157, 191
- [Pin69] K. K. Pingle. Visual perception by a computer. In *Automatic Interpretation and Classification of Images*, pages 277–284, New York, USA, 1969. Academic Press. 39
- [PL00] A. Pope and D. Lowe. Probabilistic models of appearance for 3-D object recognition. *International Journal of Computer Vision (IJCV)*, 40(2):149–167, 2000. 146
- [PPJV01] R. Paredes, J. C. Pérez-Cortés, A. Juan, and E. Vidal. Local representations and a direct voting scheme for face recognition. In *International Workshop on Pattern Recognition in Information Systems*, pages 71–79, July 2001. 56

## BIBLIOGRAPHY

---

- [Pre74] C. J. Preston. *Gibbs States on Countable Sets*. Cambridge University Press, 1974. 11
- [PU01] B. Philip and P. Updike. Car dataset. In *SURF project*, <http://www.vision.caltech.edu/>, California Institute of Technology, 2001. 144
- [Pug06] N. Pugeault. *An Early Cognitive Framework for the Spatial Reconstruction of Visual Information*. PhD thesis, University of Stirling, Stirling, Scotland, 2006. 56
- [PV98] M. Pontil and A. Verri. Support vector machines for 3D object recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 20(6):637–646, 1998. 145
- [QCD04] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems (NIPS)*, 2004. 198
- [RFZM93] C. Rothwell, D. Forsyth, A. Zisserman, and J. Mundy. Extracting projective structure from single perspective views of 3D point sets. In *International Conference on Computer Vision (ICCV)*, pages 573–582, 1993. 59
- [RLSP06] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision (IJCV)*, 66(3):231–259, 2006. xiii, 146, 147, 157, 159, 166
- [Ros56] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27:832–837, 1956. 237
- [RP99] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. In *Nature Neuroscience*, volume 2, pages 1019–1025, 1999. 68, 70, 83, 86, 191
- [RTG00] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. 63

- [SB91] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991. 47
- [SB98] R. S. Sutton and A. G. Barto. *Reinforcement Learning, an Introduction*. MIT Press, 1998. 194
- [SB06] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2048, 2006. 98
- [SBR<sup>+</sup>04] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 421–428, 2004. 23, 98, 195
- [SC95] J. R. Smith and S-F. Chang. Single color extraction and image query. In *IEEE International Conference on Image Processing (ICIP)*, pages 528–531, October 1995. 48
- [SC96] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *European Conference on Computer Vision (ECCV)*, April 1996. 48
- [SC99] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 206–213, 1999. 114
- [SC00] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision (IJCV)*, 36(1):31–50, January 2000. 48
- [Sca04] F. Scalzo. Unsupervised learning of visual feature hierarchies, September 2004. DEA Thesis, University of Liege, Belgium. 54
- [SCGM01] M. Sigman, G. A. Cecchi, C. D. Gilbert, and M. O. Magnasco. On a common circle: Natural scenes and gestalt rules. *Proceedings of the National Academy of Sciences, USA*, 98:1935–1940, 2001. 115
- [Sch78] G. Schwartz. Estimating the dimension of a model. *The Annals of Mathematical Statistics*, 6(2):461–464, 1978. 120, 121, 151



## BIBLIOGRAPHY

---

- [Sch96] C. Schmid. *Appariement d'images par invariants locaux de niveaux de gris*. PhD thesis, Institut National Polytechnique de Grenoble (INPG), INRIA Rhône–Alpes, France, July 1996. 27, 86, 191
- [SD02] A. Shokoufandeh and S. Dickinson. Graph-theoretical methods in computer vision. In *Springer-Verlag Heidelberg Lecture Notes in Computer Science*, volume 2292, pages 148–174, 2002. 69
- [Seg88] J. Segen. Learning graph models of shape. In *International Conference on Machine Learning (ICML)*, pages 29–35, 1988. 69
- [Ser06] T. Serre. *Learning a dictionary of shape-components in visual cortex: Comparison with neurons, humans and machines*. PhD thesis, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, April 2006. 70, 71
- [SIFW03] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Non-parametric belief propagation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–612, 2003. 19, 20, 21, 22, 101
- [Sil86] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, April 1986. 237, 238, 239
- [SISB03] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Advances in Neural Information Processing Systems (NIPS)*, 2003. 23, 98, 195
- [SK87] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, 1987. 55, 56
- [SKW05] J. M. Schedl, P. Knees, and G. Widmer. A web-based approach to assessing artist similarity using co-occurrences. In *International Workshop on Content-Based Multimedia Indexing (CBMI)*, June 2005. 115
- [SL03] N. Sebe and M. S. Lew. Comparing salient point detectors. *Pattern Recognition Letters*, 24(1-3):89–96, January 2003. 38

- [SLL01] S. Se, D. Lowe, and J. Little. Local and global localization for mobile robots using visual landmarks. *IEEE International Conference on Intelligent Robots and Systems*, pages 414–420, 2001. 50
- [SLP03] C. Schmid S. Lazebnik and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 48, 49, 78
- [SM97] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997. 44
- [SMB98] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, pages 230–235, 1998. 54
- [SMFW04a] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *Advances in Neural Information Processing Systems (NIPS)*, 2004. 239
- [SMFW04b] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual hand tracking using nonparametric belief propagation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 12, page 189, 2004. 23, 239
- [SP05] F. Scalzo and J. H. Piater. Statistical learning of visual feature hierarchies. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Workshop on Learning)*, San Diego (CA, USA), June 2005. 68, 192
- [SP06] F. Scalzo and J. H. Piater. Unsupervised learning of dense hierarchical appearance representations. In *International Conference on Pattern Recognition (ICPR)*, pages 395–398, August 2006. 53, 192
- [SP07] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 197

## BIBLIOGRAPHY

---

- [Spi71] F. Spitzer. Random fields and interacting particle systems. In *Mathematical Association of America Summer Seminar*, volume 78, page 142, 1971. 11
- [SPKW06] M. Schedl, T. Pohle, P. Knees, and G. Widmer. Assigning and visualizing music genres by web-based co-occurrence analysis. In *International Conference on Music Information Retrieval (ISMIR)*, Victoria, Canada, October 2006. 115
- [SPS00] C. Spence, L. C. Parra, and P. Sajda. Hierarchical image probability (HIP) models. In *IEEE International Conference on Image Processing (ICIP)*, 2000. 67
- [SRE<sup>+</sup>05] J. Sivic, B. Russell, A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, October 2005. 64
- [SS90] G. Shafer and P. P. Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2:327–351, 1990. 16
- [SS01] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. 63
- [Ste04] E. Stewart. *Intel Integrated Performance Primitives: How to Optimize Software Applications Using Intel IPP*. Intel Press, 2004. 171
- [STFW05] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, pages 1331–1338, 2005. 64
- [STL<sup>+</sup>02] N. Sebe, Q. Tian, E. Louprias, M. S. Lew, and T. S. Huang. Evaluation of salient point techniques. In *International Conference on Image and Video Retrieval (CIVR)*, pages 367–377, 2002. 39
- [STLC97] S. Sclaroff, L. Taycher, and M. La Cascia. ImageRover: A content-based image browser for the world wide web. *IEEE Workshop on Content-Based Access Image and Video Libraries*, page 2, 1997. 48

- [Sud06] E. B. Sudderth. *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology (MIT), Boston, MA , USA, May 2006. 14, 18
- [SUS02] M. Vidal-Naquet S. Ullman and E. Sali. Visual features of intermediate complexity and their use in classification. In *Nature Neuroscience*, volume 5(7), July 2002. 70
- [SW96] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(8):831–836, August 1996. 56
- [SWP05] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 994–1000, 2005. 71, 197
- [SZ02] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *European Conference on Computer Vision (ECCV)*, volume 1, pages 414–431, 2002. 46, 78
- [SZ03] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, volume 2, pages 1470–1477, October 2003. 62, 68, 70
- [SZCB04] L. Sigal, Y. Zhu, D. Comaniciu, and M. J. Black. Tracking complex objects using graphical object models. In *International Workshop on Complex Motion*, pages 227–238, 2004. 195
- [Tam05] T. Tamminen. *Models and methods for bayesian object matching*. PhD thesis, University of Technology, Helsinki, Finland, November 2005. 21
- [Tar04] M. J. Tarr. Human object recognition, do we know more now than we did 20 years ago? In *CVPRW*, pages 5–6, 2004. 71
- [Tar06] M. J. Tarr, editor. *Object Recognition - 20 Years Later*, May 2006. VSS06 symposium on object recognition. 71, 85

## BIBLIOGRAPHY

---

- [TC03] M. J. Tarr and Y. D. Cheng. Learning to see faces and objects. *Trends in Cognitive Sciences*, 7(1):23–30, 2003. 194
- [TC04] J. Thureson and S. Carlsson. Appearance based qualitative image description for object class recognition. In *European Conference on Computer Vision (ECCV)*, pages 518–529, 2004. 55
- [TCRK01] Y. Tsin, R. Collins, V. Ramesh, and T. Kanade. Bayesian color constancy for outdoor object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, December 2001. 72
- [TG00] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference (BMVC)*, September 2000. 37, 56
- [TP91] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 56
- [Tuy00] T. Tuytelaars. *Local Invariant Features for Registration and Recognition*. PhD thesis, University of Leuven, Leuven, Belgium, December 2000. 37, 38
- [TVG04] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision (IJCV)*, 59(1):61–85, 2004. 37, 39, 56, 75
- [UB91] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, 1991. 59
- [Ung59] S. H. Unger. Pattern detection and recognition. *IRE*, 47:1737–1752, 1959. 58
- [us01] Anonymous undergraduate students. Motorbikes and airplanees (side) dataset. In *Caltech Image Dataset*, <http://www.vision.caltech.edu/>, California Institute of Technology, 2001. 144
- [VdWS06] J. Van de Weijer and C. Schmid. Coloring local feature extraction. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 334–348, 2006. 149, 167, 174, 176, 177

- [vEMT<sup>+</sup>06] M. van Eede, D. Macrini, A. Telea, C. Sminchisescu, and S. S. Dickinson. Canonical skeletons for shape matching. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 64–69, 2006. 70
- [VS91] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991. 36
- [VS04] J. Vogel and B. Schiele. Natural scene retrieval based on a semantic modeling step. In *International Conference on Image and Video Retrieval*, July 2004. 40, 91
- [WAC<sup>+</sup>04] J. Willamowski, D. Arregui, G. Csurka, C. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *ICPR Workshop Learning for Adaptable Visual Systems*, August 2004. 62, 144
- [WCM05] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *IEEE Computer Society International Conference on Computer Vision (ICCV)*, pages 1800–1807, 2005. 62
- [WCT98] K. Walker, T. Cootes, and C. Taylor. Locating salient facial features using image invariants. In *IEEE Computer Society International Conference on Face and Gesture Recognition (FG)*, pages 242–247, 1998. 44
- [Web99] M. Weber. Frontal face dataset. In *Caltech Image Dataset*, <http://www.vision.caltech.edu/>, California Institute of Technology, 1999. 144
- [Web00] M. Weber. *Unsupervised learning of models for object recognition*. PhD thesis, California Institute of Technology, Pasadena, CA, USA, 2000. 65
- [Wer23] M. Wertheimer. Untersuchungen zur lehre von der gestalt II. *Psychologische Forschung*, 4:301–350, 1923. Trans. as "Laws of Organization in Perceptual Forms," in W. Ellis, ed. *A Source Book of Gestalt Psychology*, 71-88. London: Routledge & Kegan Paul, 1938. 61, 113, 115

## BIBLIOGRAPHY

---

- [WF01] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, February 2001. 18
- [Whi90] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley, New York, 1990. 8
- [Win72] R. L. Winkler. *Introduction to Bayesian Inference and Decision*. Toronto: Holt, Rhinehart and Winston, first edition, 1972. 14
- [WJ95] M. P. Wand and M. C. Jones. *Kernel smoothing*, volume 60 of *Mono-graphs on Statistics and Applied Probability*. Chapman and Hall Ltd., London, 1995. 237
- [WJW02] M. Wainwright, T. Jaakkola, and A. Willsky. Tree-based reparameterization framework for approximate estimation on graphs with cycles. In *Neural Information Processing Systems*, volume 14. The MIT Press, 2002. 18
- [WK02] H. Wersing and E. Körner. Unsupervised learning of combination features for hierarchical recognition models. In *International Conference on Artificial Neural Networks (ICANN)*, pages 1225–1230, 2002. 71
- [WK03] H. Wersing and E. Körner. Learning optimized features for hierarchical models of invariant object recognition. *Neural Computation*, 15(7):1559–1588, 2003. 71, 145
- [WMC<sup>+</sup>00] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. In *Advances in Neural Information Processing Systems (NIPS)*, 2000. 137, 139
- [WPW00] M. Weber, P. Perona, and M. Welling. Unsupervised learning of models for recognition. In *European Conference on Computer Vision (ECCV)*, pages 18–32, 2000. 91
- [WQMD06] S. Wang, A. Quattoni, L.-P. Morency, and D. Demirdjian. Hidden conditional random fields for gesture recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1527, 2006. 198

- [WR93] M. Wettler and R. Rapp. Computation of word associations based on the co-occurrences of words in large corpora, 1993. 114
- [WvdMW06] G. Westphal, C. von der Malsburg, and R. P. Würtz. Feature-driven emergence of model graphs for object recognition and categorization. In *Organic Computing - Controlled Emergence*, number 06031 in Dagstuhl Seminar Proceedings, 2006. 145
- [Xia89] Y. Xia. Skeletonization via the realization of the fire front's propagation and extinction in digital binary shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 11:1076–1086, 1989. 39
- [YFW03] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, pages 239–269, 2003. 15, 16
- [Yui91] A. L. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1):59–70, 1991. 65
- [Zhu03] S. Zhu. Statistical modeling and conceptualization of visual patterns. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 25(6):691–712, 2003. 115
- [ZK04] Z. Zivkovic and B. Krose. An EM-like algorithm for color-histogram-based object tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 798–803, 2004. 48
- [ZMLS07] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision (IJCV)*, 73(2):213–238, 2007. 63, 199



# Kernel Density Estimation

---

An attractive statistical method for modeling general probability distributions (*i.e.* without making any assumptions on the underlying distribution) from sample points is the kernel-based density estimation (KDE), or Parzen window density estimation [Ros56, Par62, Sil86, WJ95].

In kernel density estimation, a kernel function  $K : \mathbb{R}^d \mapsto \mathbb{R}$  is used to smooth a set of observed samples into a continuous density estimate. For  $N$  samples (*i.e.* data points)  $\mu_1 \dots \mu_N$ , the density estimate can be written:

$$\hat{f}(x; \mu_{i=1\dots n}) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - \mu_i}{h}\right), x \in \mathbb{R}^d \quad (\text{A.1})$$

where  $h$  denotes the kernel size (*i.e.* bandwidth), and controls the smoothness of the resulting density estimation. In this formulation, the bandwidth parameter  $h$  is fixed; so that it is held constant across  $x$  and the  $\mu_i$ 's.

Estimating multimodal densities with a fixed bandwidth kernel may affect the quality of the estimation. Therefore, in order to better capture the distribution it is possible to define a kernel  $K : \mathbb{R}^d \mapsto \mathbb{R}$  that allows a different bandwidth at each sample point [BMP77]:

$$\hat{f}_v(x; \mu_{i=1\dots n}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h(\mu_i)} K\left(\frac{x - \mu_i}{h(\mu_i)}\right), x \in \mathbb{R}^d \quad (\text{A.2})$$

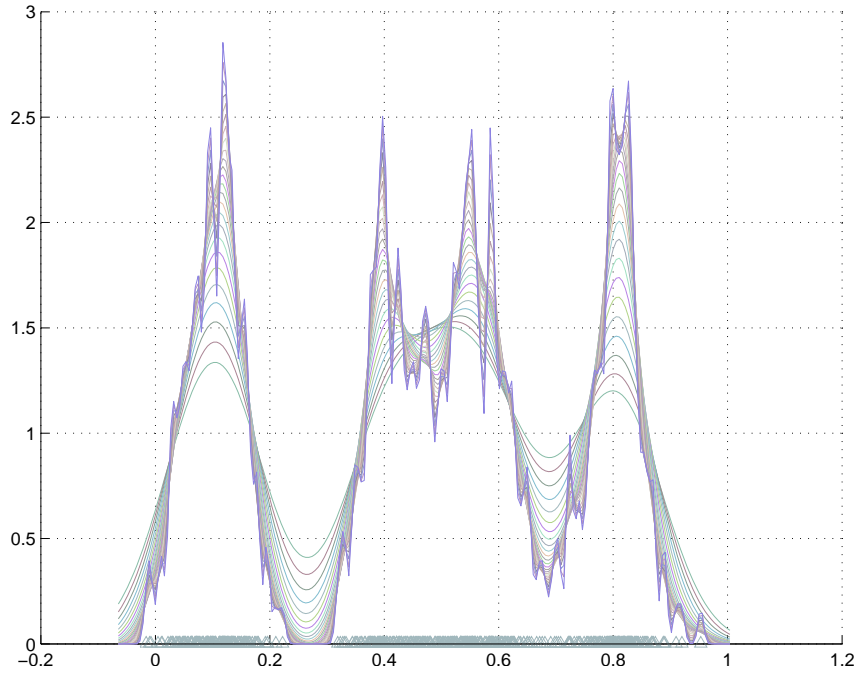


Figure A.1: Illustration of the effect of the bandwidth size on the density. Typically, when the bandwidth is too large, important features are lost; however, if it is too small, the exact values of the data begin to deteriorate the density estimate.

The most common kernel function is the isotropic Gaussian kernel  $\mathcal{G}$ , stated as follows:

$$K(x; \mu_i, h_i) = \mathcal{G}(x; \mu_i, h_i) \quad (\text{A.3})$$

$$\mathcal{G}(x; \mu_i, h_i) = \frac{1}{(2\pi)^{d/2} |h_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T h_i^{-1} (x-\mu_i)} \quad (\text{A.4})$$

where  $d$  is the dimensionality of the data,  $\mu_i$  is the mean or center of the kernel, and  $h$  is the covariance (*i.e.* bandwidth). Although the use of Gaussian Kernels is very popular, it is also possible to use other kernel shapes (Uniform, Cosinus, ...) [Sil86]. In most applications, the chosen Kernel function has only a small impact on the quality of the resulting density.

However, the selection of the kernel size (*i.e.* bandwidth)  $h$  is crucial. As it can be observed in Figure A.1, a wrong choice of bandwidth parameter may lead to over- and under- smoothing effects that may lower the quality of the density estimate.

Numerous techniques have been developed to select a reasonable value for the bandwidth automatically. We mention here three of them;

**Rule of thumb** is a simple heuristic [Sil86] that can be used to estimate the bandwidth value in the case of fixed bandwidth kernels. Specifically, it assumes that the data samples are drawn from a Gaussian distribution, and computes the optimal bandwidth for the kernel density estimate as a function of the variance of the one-dimensional data by

$$h = 1.05 \sigma^2 N^{-2/5} \quad (\text{A.5})$$

$$\text{where } \sigma = \frac{1}{N} \sum (x_i - \mu)^2 \quad \mu = \frac{1}{N} \sum x_i \quad (\text{A.6})$$

Despite its successful results in several applications [SMFW04a, SMFW04b, IFMW04], this technique has a tendency to oversmooth the distribution (see Figure A.2) and thus prefers unimodal density estimates [Sil86].

**Likelihood cross-validation** IHLER [Ihl05] proposed to use a maximum-likelihood framework to choose the bandwidth. This method is more accurate but has a higher cost in complexity.

**K-Nearest** In order to better reflect the local distribution, it is possible to use a different bandwidth  $\Sigma_i$  for each sample point  $\mu_i$  [BMP77]. A common way to compute such a covariance matrix (Equation A.8) is to use the  $k$ -nearest neighbors, where an empirical choice for the integer  $k$  is  $k = n^{1/2}$ .

$$\mathcal{G}(x; \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)} \quad (\text{A.7})$$

$$\begin{aligned} \Sigma_i &= E((X - x_i)(X - x_i)^T) \\ &= \frac{1}{n} \sum_{j=1}^k (x_j x_j^T) - \mu_i x_i^T - x_i \mu_i^T + x_i x_i^T \end{aligned} \quad (\text{A.8})$$

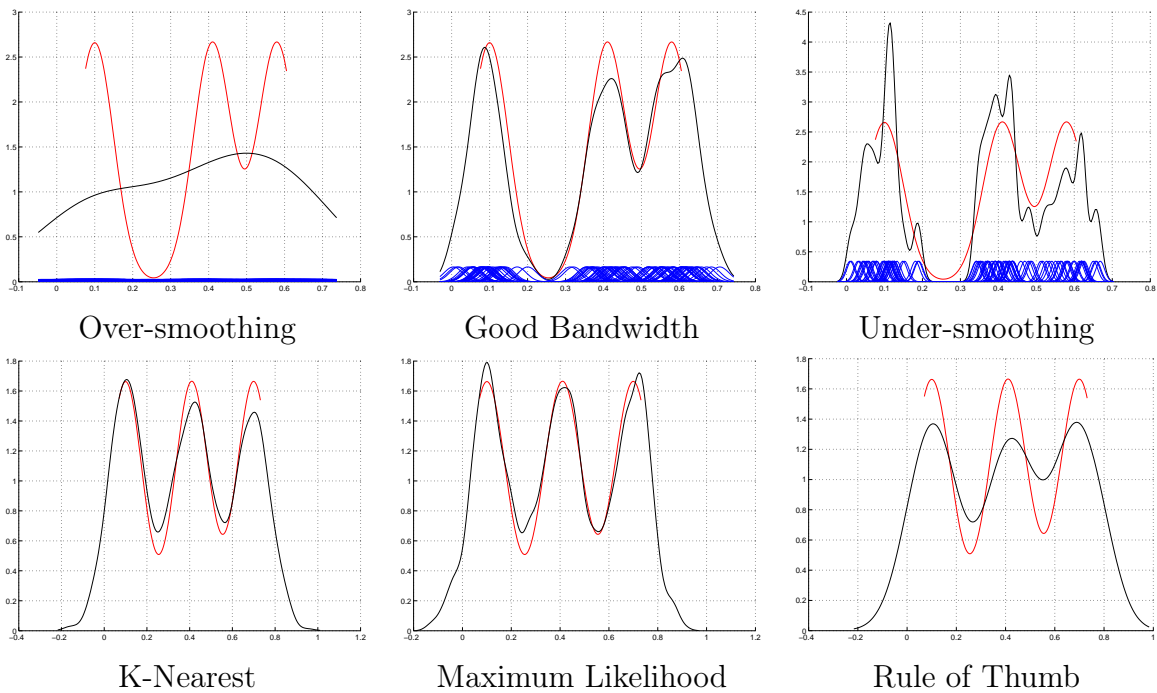


Figure A.2: The first row provides three scenarios that may occur during a KDE; over-smoothing, when the bandwidth is too large, under-smoothing when it is too small and a good bandwidth that closely matches the original distribution (red). The second row shows the resulting KDE for three different methods for selecting the bandwidth. K-Nearest and Maximum Likelihood methods provide a fair estimate but the Rule of Thumb tends to oversmooth the kernels.