



“SAPIENZA”, UNIVERSITÀ DI ROMA

DOTTORATO DI RICERCA IN INGEGNERIA INFORMATICA

XXI CICLO – 2010

Combining shape and color: a bottom-up approach to  
evaluate object similarities

Alessio Pascucci





“SAPIENZA”, UNIVERSITÀ DI ROMA

DOTTORATO DI RICERCA IN INGEGNERIA INFORMATICA

XXI CICLO - 2010

Alessio Pascucci

Combining shape and color: a bottom-up approach to  
evaluate object similarities

Thesis Committee

Prof. Fiora Pirri (Advisor)  
Prof. Christian Micheloni

Reviewers

Prof. Sven Dickinson  
Prof. Andrea Torsello

Copyright © 2010  
by Alessio Pascucci

ISBN:

AUTHOR'S ADDRESS:

Alessio Pascucci

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"

"Sapienza", Università di Roma

Via Ariosto 25, I-00185 Roma, Italy.

E-MAIL: [pascucci@dis.uniroma1.it](mailto:pascucci@dis.uniroma1.it)

WWW:

*to the women who have taught me so much:  
Elena Fazi, Laura Camponeschi, Barbara Quagliarini e Fiora Pirri*

# Abstract

The objective of the present work is to develop a *bottom-up* approach to estimate the similarity between two unknown objects. Given a set of digital images, we want to identify the main objects and to determine whether they are similar or not. In the last decades many object recognition and classification strategies, driven by higher-level activities, have been successfully developed. The peculiarity of this work, instead, is the attempt to work without any training phase nor *a priori* knowledge about the objects or their context. Indeed, if we suppose to be in an unstructured and completely unknown environment, usually we have to deal with novel objects never seen before; under these hypothesis, it would be very useful to define some kind of similarity among the instances under analysis (even if we do not know which category they belong to).

To obtain this result, we start observing that human beings use a lot of information and analyze very different aspects to achieve object recognition: shape, position, color and so on. Hence we try to reproduce part of this process, combining different methodologies (each working on a specific characteristic) to obtain a more meaningful idea of similarity. Mainly inspired by the human conception of representation, we identify two main characteristics and we called them the implicit and explicit models. The term "explicit" is used to account for the main traits of what, in the human representation, connotes a principal source of information regarding a category, a sort of a visual synecdoche (corresponding to the shape); the term "implicit", on the other hand, accounts for the object rendered by shadows and lights, colors and volumetric impression, a sort of a visual metonymy (corresponding to the chromatic characteristics).

During the work, we had to face several problems and we tried to define specific solutions. In particular, our contributions are about:

- defining a bottom-up approach for image segmentation (which does not rely on any *a priori* knowledge);
- combining different features to evaluate objects similarity (particularly focus-

ing on shape and color);

- defining a generic distance (similarity) measure between objects (without any attempt to identify the possible category they belong to);
- analyzing the consequences of using the number of modes as an estimation of the number of mixture's components (in the Expectation-Maximization algorithm).

# Acknowledgements

It is difficult to thank all the people who, for various reasons, have made this work possible. And I hope I am not forgetting anyone.

First Fiorenza Pirri, my advisor, who has been able to support (and stand) me for many years with valuable advices (not only for research). Without her I would have never got here. Thank you!

A heartfelt thanks to prof. Maurizio Lenzerini, my first doctoral coordinator, and prof. Roberto Baldoni, current coordinator: he followed and advised me wisely to complete my journey.

To the whole ALCOR LAB, for support, discussions, advice and all the good times we had together (and, most importantly, for always making me really feel part of the group, despite it all!). A special thank to Matia Pizzoli and Andrea Carbone also for valuable suggestions and support in the process of writing the thesis.

To the whole Soviet: Jasmin Martenz, Valentina Piacentini, Francesco Scialaqua, Cristina Comezzi, Lorenzo Pizzoferrato, Rachele Cardillo, Paolo Urbano and in particular to Gian Diego Tipaldi, Massimo Mastrangeli and Stefano Pellegrini for their significant reviews.

To Isotta Chimenti for her continuous (and infinite!) patience, even greater in these last months, and for the her valuable linguistic support.

Special thanks go to Edoardo, who has almost arrived: for the strong emotions that I have felt and those that will come.

To my parents, Paola and Ezio, for their continued love and for supporting me in this adventure (yet another).

To Michele "the philosopher", for the music (that gave me the right concentration).

To all the working group: Serena Borgna, Davide Campolongo, Francesco Costantini, Lorenzo Croci, Alessandro Gazzella, Matteo Luchetti, Claudio Nucci, Matteo Orlando, Mauro Porro, Marco Polverari, Simone Romagnoli, Marco Terracciano who have done all the hard work in these busy months. Thanks.

I should also make a last acknowledgment to whom, despite himself, made it possible for me to spend all this time on my PhD Thesis. But I'm not going to name him, because I know he hasn't done it on purpose.





# Contents

<b>Abstract</b>	<b>ii</b>
<b>Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>xii</b>
<b>I Introduction</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Objective . . . . .	3
1.2 Motivation . . . . .	4
1.3 Challenges . . . . .	7
1.4 Contributions . . . . .	9
1.4.1 A bottom-up approach for image segmentation . . . . .	9
1.4.2 Combining different features to evaluate objects similarity . . . . .	9
1.4.3 Shape representation and analysis . . . . .	9
1.4.4 Using the number of modes as an estimation of the number of mixtures components . . . . .	10
1.5 Methodologies and assumptions . . . . .	10
1.6 Outline . . . . .	12
<b>2 Objects and Patterns: a historical perspective</b>	<b>17</b>
2.1 Object recognition and classification . . . . .	17
2.2 Statistical approaches to classification . . . . .	18
2.2.1 Fisher Linear Discriminant analysis . . . . .	19
2.2.2 Kernel methods and Support Vector Machines . . . . .	20
2.2.3 Markov Random Fields . . . . .	22
2.3 A perceptual organization . . . . .	25

<b>II</b>	<b>An approach to implicit and explicit analysis</b>	<b>29</b>
<b>3</b>	<b>Bottom-up Image Segmentation</b>	<b>31</b>
3.1	Introduction . . . . .	31
3.2	A brief overview on the existing methodologies . . . . .	32
3.2.1	Bottom-up approaches . . . . .	33
3.2.2	Top-down approaches . . . . .	36
3.2.3	Combining bottom-up and top-down methodologies . . . . .	38
3.2.4	Autonomous agent scenarios . . . . .	39
3.3	Image segmentation: a bottom-up approach . . . . .	40
3.3.1	Noise removing . . . . .	41
3.3.2	Lightening analysis - finding the starting clusters . . . . .	44
3.3.3	Chromatic analysis - finding the probability map . . . . .	45
3.3.4	Positional analysis - refining the probability map . . . . .	49
3.3.5	Neighborhood analysis - obtaining the final cluster . . . . .	52
3.3.6	Final analysis . . . . .	57
3.4	Experimental results . . . . .	59
3.5	Open problems . . . . .	63
<b>4</b>	<b>IEA: Implicit and Explicit Analysis</b>	<b>65</b>
4.1	Introduction . . . . .	65
4.2	Methodology . . . . .	66
4.3	Explicit Analysis . . . . .	67
4.3.1	Defining the explicit model . . . . .	67
4.3.2	Shapes . . . . .	69
4.3.3	Landmarks . . . . .	71
4.3.4	Medial axis . . . . .	73
4.3.5	Critical Points . . . . .	74
4.3.6	Procrustes Analysis . . . . .	76
4.3.7	Explicit distance . . . . .	81
4.3.8	Explicit experiments . . . . .	83
4.4	Implicit Analysis . . . . .	85
4.4.1	Introduction . . . . .	85
4.4.2	Bag of tuples and probability density functions . . . . .	86
4.4.3	On the estimation of the components number . . . . .	88
4.4.4	Implicit distance . . . . .	91
4.4.5	Implicit experiments . . . . .	92
4.5	Combining the distances . . . . .	94
4.6	Open problems . . . . .	97

<b>5</b>	<b>Shape Description and Analysis</b>	<b>99</b>
5.1	Introduction	99
5.1.1	Shape description	100
5.1.2	Framework	105
5.2	Shape features representation	106
5.2.1	Shape pattern	106
5.2.2	Shape Description	110
5.3	Properties of Shape Decriptors	111
5.4	Similarity and Distance	113
<b>6</b>	<b>Implicit Analysis: evaluating the distance between two general sets of points</b>	<b>119</b>
6.1	Introduction	119
6.2	The data: our source of information	120
6.3	Sampling and fitting	121
6.4	Comparing two datasets	122
6.5	Searching a pdf	123
6.5.1	Non-parametric approach	123
6.5.2	Parametric approach	126
6.5.3	Mixture of Gaussians	128
6.6	On the underlying model	130
6.7	Our approach	131
6.7.1	Randomly generated mixtures	133
6.7.2	Sampling	133
6.7.3	Modes	133
6.7.4	Mean Shift	135
6.7.5	Distance measure	140
6.7.6	Our distance measure: Normalized Mutual Energy	144
6.7.7	Experimental Results	145
<b>7</b>	<b>Final experiments</b>	<b>153</b>
7.1	Introduction	153
7.2	Explicit analysis: topological-Fréchet distance	153
7.3	Implicit analysis: number of modes for number of components	156
7.4	Merging the distances	157
7.5	A synthesis of the whole process	158
7.5.1	Extracting the implicit and the explicit models	159
7.5.2	Similarity measure	162

<b>III</b>	<b>Conclusions</b>	<b>165</b>
<b>8</b>	<b>Conclusions and Future Works</b>	<b>167</b>
8.1	Introduction . . . . .	167
8.2	Summary of the work . . . . .	167
8.2.1	A bottom-up approach for image segmentation . . . . .	167
8.2.2	A similarity analysis based on shape and color . . . . .	168
8.2.3	Explicit analysis . . . . .	168
8.2.4	Implicit analysis . . . . .	168
8.2.5	Number of modes as component estimation . . . . .	169
8.3	Open problems and future works . . . . .	169
<b>IV</b>	<b>Appendix</b>	<b>173</b>
<b>A</b>	<b>Appendix</b>	<b>175</b>
A.1	A brief overview of the probability density distance measures . . . . .	175
A.1.1	Kullback-Leibler distance . . . . .	175
A.1.2	J divergence . . . . .	177
A.1.3	K divergence . . . . .	177
A.1.4	L divergence . . . . .	178
A.1.5	Jensen-Shannon divergence . . . . .	180
A.1.6	Other distance measures . . . . .	181
A.1.7	General inequalities . . . . .	182
	<b>Bibliography</b>	<b>183</b>

# List of Figures

1.1	An example of scenario with multiple instances of different classes of objects (picture taken from (95), pag. 3 fig. 1). . . . .	6
1.2	Sean Connery at different ages. . . . .	7
1.3	Some examples from the RGB-JPEG database. . . . .	11
1.4	Some logical images from the log-JPEG database with, on the left, the original image. . . . .	12
1.5	Some examples from the ALOI database. . . . .	13
1.6	Some examples of the used images from the Berkeley database. . . . .	14
1.7	On the left: few primitive traits of the Michelangelo's David; on the right, a part of a studio of drapery by Leonardo. . . . .	15
2.1	Two sets of 2d data points and the corresponding projection based on Fisher linear discriminant (taken from (27), pag. 188 fig 4.6). . . . .	20
2.2	A graphic example of how a Kernel method works. . . . .	21
2.3	A subset of the famous Biedermann's geons. . . . .	26
3.1	A block diagram to summarize the main steps of the segmentation algorithm. . . . .	41
3.2	An image of a toy airplane taken from the RGB-JPEG database. . . . .	42
3.3	The starting image after the local thresholding. . . . .	43
3.4	The image with the starting clusters (in different colors). The background is represented in black. . . . .	45
3.5	The image shows one starting block, the random point (red point) and the squared-shaped subset used for chromatic analysis (red rectangle). . . . .	46
3.6	The (normalized) probability map obtained evaluating the whole image with the mixture obtained from the squared-shaped set defined in figure 3.5. . . . .	49
3.7	Chromatic probability map with block enhancing. . . . .	50
3.8	The Gaussian filter for the image under analysis. . . . .	51
3.9	The probability map after the Gaussian distance filtering. . . . .	52

3.10	A schematic representation of the $s - t$ graph obtained starting from the image under analysis. From each pixel of the image start two $t$ -links: the first (in azure) toward the source $s$ with weight $D_q(1)$ , the second (in orange) toward the sink $t$ with weight $D_q(0)$ . . . . .	55
3.11	8-neighborhood system. . . . .	56
3.12	Our image after the last step: the toy airplane is isolated from the background but some parts are still missing. . . . .	58
3.13	Some examples of our segmentation technique. . . . .	59
3.14	A table containing some significant values about the images segmented (from left to right): weight of the picture, height of the picture, number of pixels of the picture, number of pixels of the object (calculated by hand), number of pixel returned by the segmentation procedure, false negatives (number of pixels of the object not recognized by the segmentation), false positives (number of pixels not belonging to the object but recognized by the segmentation), false negatives/object size ratio, false positives/object size ratio. . . . .	60
3.15	A graph showing how the object/picture size ratio influences the performance of the procedure. . . . .	62
3.16	A graph showing how the performances of the procedure are influenced by $\psi$ . . . . .	63
4.1	Images from our log-JPEG database with the implicit (on the left) and the explicit (on the right) descriptions. . . . .	66
4.2	An image of an airplane and the contour obtained by using the Canny algorithm. . . . .	68
4.3	Two objects with the same form: an elephant. . . . .	70
4.4	A rabbit-shaped cloud. . . . .	71
4.5	A logical shape (on the left) and its extracted boundary (on the right). . . . .	72
4.6	An elephant shape with 4 landmarks (A, B, C, D) highlighted. . . . .	73
4.7	A shape from our repository and the corresponding Blum's medial axis. . . . .	74
4.8	Contours of two different objects of our dataset with highlighted critical points (in red). . . . .	76
4.9	An example of Procrustes analysis on two triangles (1). Three different transformations are applied: scaling (2), translation (3) and rotation (4). . . . .	78
4.10	Two airplanes isolated from the background with critical points (in red). . . . .	79
4.11	The contours of the two airplanes of the previous figure with critical points. The critical points added to make the shapes comparable are colored in black. . . . .	80
4.12	Two digital images of bottles with highlighted critical points (in red). . . . .	82

4.13	On the left, the contours of the two bottles of figure 4.12 with critical points: in blue bottle 1, in red bottle 2. On the right, in blue the contour of bottle 1 after the transformation (and in red the contour of bottle 2). . . . .	83
4.14	Intra-class Procrustes distance in airplane family. . . . .	84
4.15	Intra-class Procrustes distance in bottle family. . . . .	86
4.16	Inter-class Procrustes distance between airplanes and bottles families. . . . .	87
4.17	A hard case: instances of the same object (elephant) in different poses. . . . .	88
4.18	A 2-dimensional Mixture of Gaussians with 4 components. . . . .	89
4.19	The HS 2D histogram from a tiger image (in the upper panel) and the corresponding peak-climbing procedure in the neighborhood of the highest peak (in the lower panel). . . . .	90
4.20	Intra-class distance values of the Normalized Mutual Energy over the tigers family. . . . .	93
4.21	Inter-class distance values of the Normalized Mutual Energy between the tiger family and other objects, specifically (from left to right): airplane, truck, dog, butterfly, building). . . . .	94
4.22	On the $x$ -axes the explicit distance (Normalized Mutual Energy), on the $y$ -axes the implicit distance (Procrustes distance). The blue circles represent the element belonging to the similarity class, while the red crosses represent the ones belonging to non-similarity. Two elements are considered similar if $d_e < \pi$ and $d_i > \lambda$ , that is if the points representing their two distances falls in the blue area. . . . .	95
4.23	On the $x$ -axes the explicit distance (Normalized Mutual Energy), on the $y$ -axes the implicit distance (Procrustes distance). The blue circles represent the element belonging to the similarity class, while the red crosses represent the ones belonging to non-similarity. Two elements are considered similar if the points representing their two distances falls in the blue area, obtained by using Fisher discriminat analysis. . . . .	96
5.1	Different transformed copies of the same shape. . . . .	106
5.2	The polygonal version of the image in figure 5.1. Boundary points are colored in green, while critical points are colored in black. . . . .	107
5.3	A simple pixel-continuous and closed shape. . . . .	110
5.4	Some figures taken from our data set and representing an elephant in different poses. . . . .	115
6.1	Histogram approach to density estimation with different dimension of the bin (taken from (27), pag. 121 fig. 2.24). . . . .	125
6.2	The same set of data estimated with two nonparametric approaches: histograms (on the right) and kernels (on the left). . . . .	126



6.3	An example of a 3-dimensional mixture of Gaussians with 5 components. . . . .	129
6.4	A block diagram to summarize the main step of the mixture estimation algorithm. . . . .	131
6.5	Some mixtures of Gaussians randomly generated. . . . .	132
6.6	A sampling of a 5 components 3D mixture of Gaussians. . . . .	134
6.7	Image taken from (51) (fig. 1, pag. 3): an example of mixture of Gaussians where the number of modes (9 - marked by " $\Delta$ ") exceeds the number of components (6 - marked by "+"). . . . .	136
6.8	A 3D 3-components mixture of Gaussians. In the panels (from left to right) the mean of one of the components changes getting closer to the mean of another component. Upper panels represent the 3d view of the mixtures; lower panels represent the profiles (projected on the $(x, y)$ axis) of the corresponding mixtures with modes (red stars) and means (yellow dots) highlighted. . . . .	137
6.9	On the $x$ and $y$ axis are represented, respectively, the different values of $\lambda$ and $h$ . For each couple of values, the histograms indicate how many occurrences of the couple has returned an estimated modes number equal to the components number (upper left), how many occurrences of the couple has returned an estimated modes number differ for ! unit from the components number (upper right) and the sum of the two previous values (lower). . . . .	140
6.10	An example of mean shift algorithm: the red points represent the sampling of the mixture (superimposed), the green stars the starting point of the algorithm, the green points the trajectories' steps of the algorithm and the yellow stars the estimated modes (before thresholding). . . . .	141
6.11	Two 2D mixtures with 3 components (top right and left) and the corresponding mutual energy (bottom, in green). . . . .	144
6.12	Experiments with 1d mixtures: $mixt_N$ is closer to the original mixture than $mixt_M$ . The original mixture (in black, bottom left), $mixt_N$ (in blue, top left), $mixt_M$ (in red, top right), a graph containing the original mixture and $mixt_M$ plotted together (in black and red respectively, bottom right). . . . .	146
6.13	Two examples where the distance between $mixt_M$ and $mixt_N$ is close to the mean value (0.0232). In the top graphs (A1-A4) $d_R = 0.0238$ , in the bottom graphs (B1-B4) $d_R = 0.0226$ . . . . .	147
6.14	Experiments with 1d mixtures: $mixt_M$ is closer to the original mixture than $mixt_N$ . The original mixture (in black, lower left), $mixt_N$ (in blue, upper left), $mixt_M$ (in red, upper right), a graph containing the original mixture and $mixt_M$ plotted together (in black and red respectively, lower right). . . . .	148

6.15	Experiments with 1d mixtures: the original mixture is represented in black, the estimated one with a dashed red line and, in green, the overlapping area. Two scenarios: a mean value for $d_M = 0.202$ (on the left) and the minimum value for $d_M = 4.0241e^{-6}$ (on the right).	149
6.16	Experiments with 2d mixtures: $mixt_N$ is closer to the original mixture than $mixt_M$ . The original mixture (in green, left), $mixt_N$ (in blue, center), $mixt_M$ (in red, right).	150
6.17	Experiments with 2d mixtures: an example where the distance between $mixt_M$ and $mixt_N$ is close to the mean value (0.0215): $d_R = 0.0217$ . The original mixture (in green, left), $mixt_N$ (in blue, center), $mixt_M$ (in red, right).	150
6.18	Experiments with 2d mixtures: another example where the distance between $mixt_M$ and $mixt_N$ is close to the mean value (0.0215): $d_R = 0.0210$ . The original mixture (in green, left), $mixt_N$ (in blue, center), $mixt_M$ (in red, right).	151
6.19	Experiments with 2d mixtures: $mixt_M$ is closer to the original mixture than $mixt_N$ . The original mixture (in green, left), $mixt_N$ (in blue, center), $mixt_M$ (in red, right).	151
7.1	Logical images representing four animals in different poses	154
7.2	A graph representing the accuracy of Topological-Fréchet similarity measure with respect to $\beta$	154
7.3	A comparison between the values with the Procrustes and the Fréchet distance among the elephant family	155
7.4	Color images representing four subjects	156
7.5	A comparison between the values obtained by comparing mixtures of Gaussians by using the number of peaks of the HS histogram (in light blue) and the number of modes of the RGB space (in light red) as an estimation of the number of components	157
7.6	On the $x$ -axes the explicit distance (Normalized Mutual Energy), on the $y$ -axes the implicit distance (topological-Fréchet distance). The blue circles represent the elements belonging to the similarity class, while the red crosses represent the ones belonging to non-similarity. Two elements are considered similar if the points representing their two distances fall in the blue area, obtained by using Fisher discriminant analysis	158
7.7	An acoustic guitar.	159
7.8	A BW version of figure 7.7.	159
7.9	The starting clusters obtained from figure 7.8.	160
7.10	The probability map obtained from cluster 1 in figure 7.9.	160
7.11	The final cluster obtained by the $s - t$ cut over the graph initialized with the probability values of figure 7.10.	161

7.12	The polygonal version of figure 7.11 with critical points highlighted.	161
7.13	The results obtained comparing the original image with 24 images taken from the Internet. The first row contains values of the explicit analysis ( $\pi = 0.52$ ), the second row values of the implicit analysis ( $\lambda = 0.56$ ), and the third row contains the classification obtained merging the two previous values. . . . .	163

## **Part I**

# **Introduction**



# Chapter 1

## Introduction

### 1.1 Objective

The objective of the present work is to develop a *bottom-up* approach to estimate the similarity between two unknown objects. Given a set of digital images, we want to find the interesting elements and to identify the similar ones. In the last decades many methodologies have been presented that try to understand the category to which a specific object belongs. To obtain this result, a training phase is necessary, or at least some *a priori* knowledge. Instead a bottom-up approach has to work without knowing anything about the object under analysis. As a consequence, it will be not able to define the category of the item, but possibly to determine its degree of similarity with respect to a reference object. This is in fact the condition under which, for example, autonomous agents usually have to work. In an unstructured and completely unknown environment, a robot has to travel around and to observe everything; it usually has to deal with novel objects never seen before, and it would be very useful if the agent was able to define some kind of similarity among the instances under analysis (even if it could not know which category they belong to). To obtain this result, we define a distance measure between objects able to determine similarities. Specifically, in order to obtain a more meaningful idea of the similarity among objects, we identify two main characteristics and we called them the implicit and explicit models. The term "explicit" is used to account for the main traits of what, in the human representation, connotes a principal source of information regarding a category, a sort of a visual synecdoche (corresponding to the shape); the term "implicit", on the other hand, accounts for the object rendered by shadows and lights, colors and volumetric impression, a sort of a visual metonymy (corresponding to the chromatic characteristics).

## 1.2 Motivation

Object recognition and classification in natural and digital images are among the most important topics in computer vision. Despite the concerted efforts of researchers over the last fifty years, the goal of identifying object categories within still or moving images remains mostly unsolved. While it may be obvious that human beings are able to recognize objects under many variable conditions, it is well known that object recognition and classification are extremely difficult tasks for computers. Human beings are able to learn the appearance of never-seen-before objects and they can create highly accurate world representations. Computers instead might be able to analyze simple features and exploit those features to recognize. However, when an object has slightly changed in its form or is seen from different viewpoints, the features would appear altered and, as a consequence, the recognition task becomes harder. Equally challenging is a scenario where a computer has to discriminate between two objects that contain the same features, but with a different organization. Visual recognition for human beings is a fast and effortless process, very robust with respect to viewpoints, lighting conditions, occlusions and clutter; human beings are able to recognize the presence, the position and the typology of an object even if it is represented only by its outlines. Biederman shows (see (22)) that a person can usually recognize more than 10,000 categories of objects. Is it possible to obtain a computer able to accomplish the same task?

Obviously the applications of an autonomous object-recognition system would be infinite. In everyday life it could help in searching videos from internet TV stations, images on the web or photos in large image databases, as a response to a "semantic" query (for instance: "*search all videos containing a bottle*"). The massive use of Internet, the increasing diffusion of social networks and the availability of high-accuracy and low-cost photo and video systems have lead to an exponential growth of high-quality digital still or moving images. As a consequence the need of a semantic image classification is becoming increasingly important. And this is the simplest scenario. Object classification is very useful also in industrial automation where robots have to recognize different items before being able to interact with them.

In the medical field also a lot of images are generated every day: echographies, radiographies, TACs and so on. A reliable and reproducible system able to routinely analyze these images can provide a great help to the diagnosis of diseases. An important improvement could be obtained also in security scenarios. Nowadays there is increasing request of autonomous video-surveillance systems for complex and huge areas, as railway stations, airports, underground stops or similar. An unintelligent system requires a person screening the flow of images, while it is possible to imagine a computer which can automatically detect suspicious people and unusual events, or which can evaluate risk situation in a crowded environment. Video compression is

another area of real interest: in fact, in the case of bandwidth limitation and signal weakness, a high-quality video communication is impossible. New improvements could be obtained using compression techniques based on a clear understanding of the visual sequence (for instance, one can encode with low quality the background and with a high quality the foreground).

Moreover in the design and development of autonomous agents the availability of a visual recognition system is crucial. A robot has usually to perform tasks in unstructured and often completely unknown environments without any human guidance. Any meaningful interaction with the world requires analysis, recognition and classification of objects. An agent has to recognize objects already known, but, above all, it has to learn the identity of unknown ones. Thinking about a rescue scenario, the agent has to move in a chaotic and unexplored environment looking for victims. The ability of finding similarity among objects can greatly improve the performances.

The applications can easily be extended to every aspect of human-computer interaction. World interaction is mostly based on visual understanding. Nowadays, as stated before, computers are not able to interpret images at a higher level of representation, as humans do. Some interesting results have been achieved in specific classes (for instance to detect human faces, to recognize vehicles, to identify particular structures and so on), but it is not possible to extend these approaches to work with any category of objects. It is very important to distinguish between these two different kinds of scenario. In fact the recognition of instances of a specific category has to cope with completely different problems with respect to the recognition of the class to which a general object belongs. In the former case we have to decide whether an image represents an instance of a particular category. If the system does not recognize the object, this is discarded and no action is expected: figure 1.1 shows an example of this scenario taken from (95) (pag. 3 fig. 1). The images can be taken from different points of view or inserted in different backgrounds, but we are working with the same class. For the detection it is clearly possible to use well-known information about shape, geometrical relationships, structure, color, texture, usual context and so on. Completely different is the scenario of a general object recognition system, where one looks instead for a visual consistency that must exist between the examples of different classes, and where one can also suppose that the system will eventually have to deal with a new (and consequently unknown) category of objects. The system analyzes the items included in the image and tries to classify them. Supposing the existence of a repository with a finite number of classes, it is possible to simply consider this scenario as a complex generalization of the previous one: every class has a specific algorithm able to recognize an object and the system has only to perform all these algorithms over the current instance. Which decision will the system take when it tries to analyze an object not belonging to any category? Many experiments have shown us that the strategy of linking the object to the *closest*



group could not have any meaning.

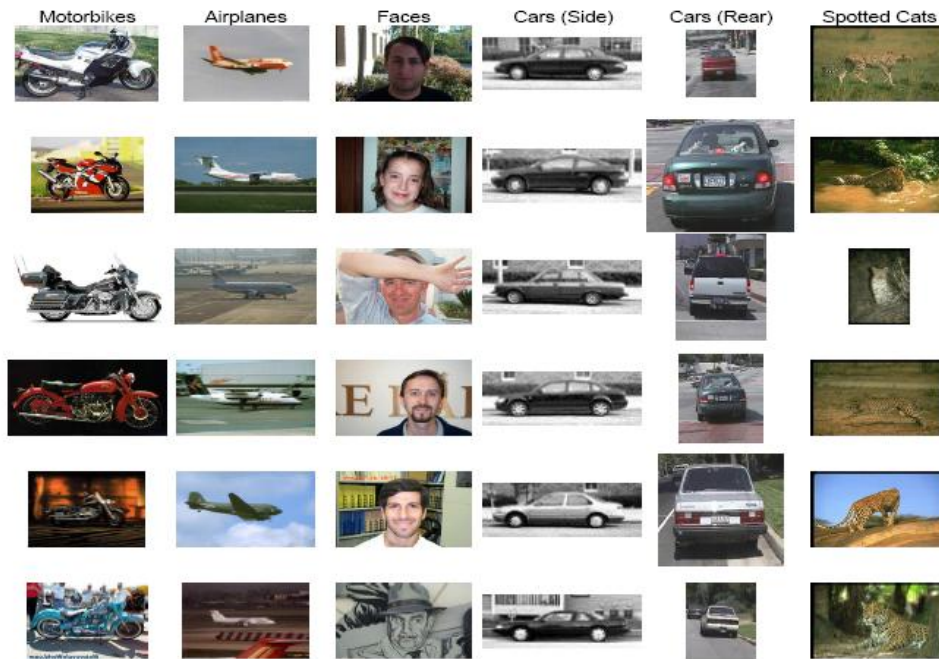


Figure 1.1: An example of scenario with multiple instances of different classes of objects (picture taken from (95), pag. 3 fig. 1).

There exist nowadays systems able to classify a large set of objects starting with a training phase. This is currently a research field of great interest. In fact, the Pascal Visual Object Classes challenge (90) is gaining increasing prestige, and can be considered as the main reference in nowadays specialized research area. It is an international competition, aiming at developing optimal object category recognition and detection strategies. It is not surprising that in all scenarios established for the competition, a training phase is always included. But how can we obtain interesting results in a purely bottom-up approach? We can not know anything about the belonging categories, so we have to define a procedure able to understand somehow whether two objects are "similar" or not. Obviously, a distance measure, which can give an evaluation of how close two generic objects are, could provide a helpful support. The aim of the present work is to make a step forward in the direction of defining this measure.

### 1.3 Challenges

Even if in the last 15 years a lot of interesting approaches for object classification have been presented, the main problems are not solved yet and still need to be addressed. According to (94), we present a first list of the most common ones:

- changes of aspect;
- illumination differences;
- changes in viewpoint;
- background clutter;
- occlusion;
- intra-class variation.

Human beings are able to easily overcome these difficulties. In fact they can quickly identify a never-seen-before object even if it has unusual characteristics, as well as recognize a person even if age has changed his/her traits (see figure 1.2). Computers can not do this, so far.

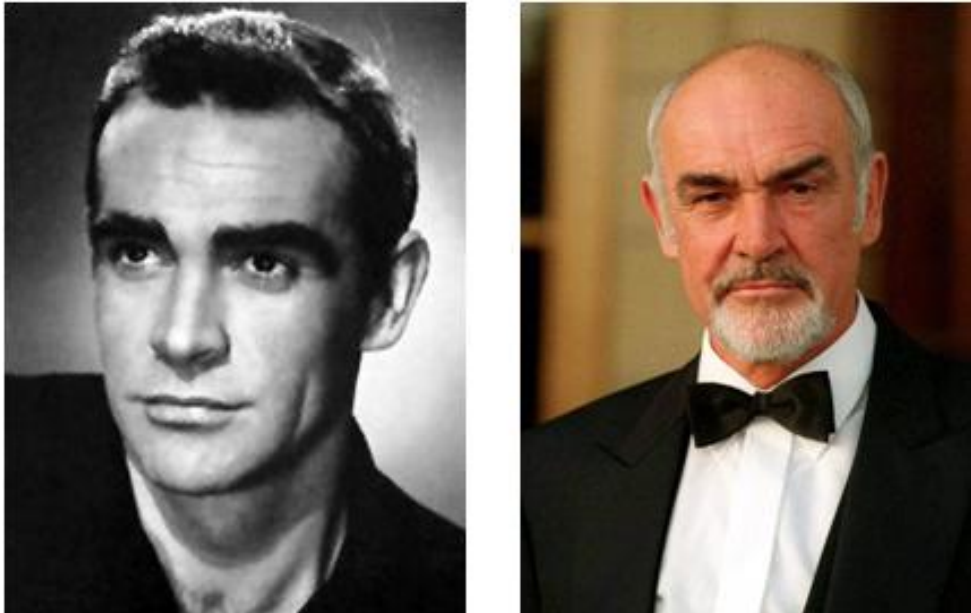


Figure 1.2: Sean Connery at different ages.

Taking into account the changes in illumination is another trivial task for people, but very hard for an autonomous system; most of the information is usually taken from the intensity version of an image and it heavily depends on light conditions. It is also important to consider that the distance between the observer and the object can completely modify the size in the image: the farther the object from the camera, the smaller its apparent size, and so a car in front of us will appear very different compared to a car far away from the observer. An object would also appear to be translated and rotated in the image depending on the location and orientation of the camera: the shape of an object, in fact, undergoes perspective effects in the imaging process. Moreover, it is necessary to consider that natural scenes rarely contain a single isolated object and, generally, this causes strong limitation to the visual process; if we glance around a room, we will quickly realize that clutter is the norm rather than the exception. An object could be partially occluded by other ones, or the reflective properties of its surface could result in part of it being invisible. Another hot topic is intra-class variation: most of the variations among many common object classes do not afford precise definitions. So the first challenge, for every kind of analysis, is to find some cues or measurements that remain unchanged or invariants, under the previously presented transformations.

Obviously the most difficult problem is the lack of a clear idea of object. We are not able to say what an object exactly is. It seems to be impossible to give a precise definition without taking into account purpose and context. Examples of distinguishing properties of objects are physical continuity (i.e. an object may be moved around in one piece), having a common cause or origin, having well defined physical limits with respect to the surrounding environment, or being made of a well defined substance. In principle, a single image taken in an unconstrained environment is not sufficient to allow a computer algorithm, or a human being, to decide where an object starts and another object ends. However, a number of cues which are based on the statistics of our everyday visual world are useful to guide this decision. The fact that objects are mostly opaque and often homogeneous in appearance makes it likely that areas of high contrast (in texture, color, brightness) will be associated with their boundaries. Similarly, in everyday life, people do not have a formal definition for a particular shape, but describe it by referring to well-known stereotyped shapes that are considered to be primitive concept. For example, describing a cloud, someone can say that it has the shape of a rabbit, being sure that other people can understand.

These last concepts give more and more credit to the rationale of planning a system able to evaluate objects similarities, not considering the category they belong to.

## 1.4 Contributions

The main aim of our work is to define a bottom-up methodology to evaluate the distance between two objects. During our research we had to face several problems and we tried to define solutions. Specifically our research contributions are listed as follows.

### 1.4.1 A bottom-up approach for image segmentation

First of all, we had to define a methodology to individuate objects in a scene without any *a priori* knowledge about them. To obtain this result our system combines light, color and positional analysis with statistical methodologies to individuate some meaningful subparts in the image. Then the system decides how to merge or to divide the blocks to isolate the interesting objects. The methodology proposed heavily relies on mixtures of Gaussians, Expectation-Maximization, Mean Shift and graph cuts.

### 1.4.2 Combining different features to evaluate objects similarity

The central idea of the work is the combination of different features (specifically shape and color information) to evaluate a distance measure between two objects (possibly completely unknown). The system describes an object by using two different models: an explicit one (that contains information about the shape) and an implicit one (that contains information about color). By defining two different distance measures (one for each model) and combining the results, the system returns an estimate of the similarity between two objects. We present a first procedure based on edge landmarks and Procrustes distance (for the explicit model), color histogram peaks and mixture of Gaussians (for the implicit one) and Fisher discriminant analysis (to merge the two distance measures).

### 1.4.3 Shape representation and analysis

The way we represent shapes is strictly connected with the definition of a similarity measure between them. So we introduce a synthetic shape representation which relies on specific critical points and the straight lines between them. The critical points are obtained considering the sudden variations in the direction of the edge of the object. The representation obtained is a compact and meaningful description of the main characteristics of the shape. We also introduce a measure of similarity, and we show how this distance is able to partially resolve some of the problems left open by the previous approach (based on Procrustes distance).

#### 1.4.4 Using the number of modes as an estimation of the number of mixtures components

We use mixtures of Gaussians several times in this work, both in the image segmentation phase and in implicit object models comparison. In both cases, we have to evaluate the best-fitting mixture given a set of points. This is a well-known and widely studied problem. The Expectation-Maximization algorithm can return the best-fitting function, but it requires to know in advance the number of components of the mixture. We use the number of modes (evaluated by a Mean Shift algorithm) as input parameters for the EM. It is already known that the number of modes could differ from the number of components. We perform a comparative experimental study about the error generated by this choice and we show that it is irrelevant in the cases of interest.

### 1.5 Methodologies and assumptions

We work on *Intel*®*core*<sup>TM</sup>2 Duo processor T5450 (1.66 GHz, 667 MHz FSB, 2 MB L2 cache) with 2 GB DDR2. We use *MATLAB*®7.0 on a *Windows*®XP environment. All the results presented (and the correlated performances) are evaluated on this platform. For our experiments we used multiple image databases, listed below.

#### RGB-JPEG database

RGB-JPEG database includes 830 JPEG RGB images of objects divided in 6 categories: airplanes, vehicles, bottles, animals, furniture and lamps. The set has been created putting together pictures taken from the Internet and digital photos realized by us. Each digital image has a size of  $1024 \times 768$  pixels. Every image contains a single object set at the center of the image. In some photo the object is on a natural background, in other ones it is set on a white or black background. Figure 1.3 shows a small subset of this database.

#### LOG-JPEG database

LOG-JPEG database includes 520 couples of JPEG RGB and logical images. The former ones are a subset of the previously presented RGB-JPEG database (containing elements from all the 6 categories); the latter are logical images (1-valued pixel on a 0-valued background) representing the edge of the starting image. As we will explain with further details in the following chapters, we use Canny algorithm with some refinements to extract the edge from the images. Figure 1.4 shows a small subset of this database.

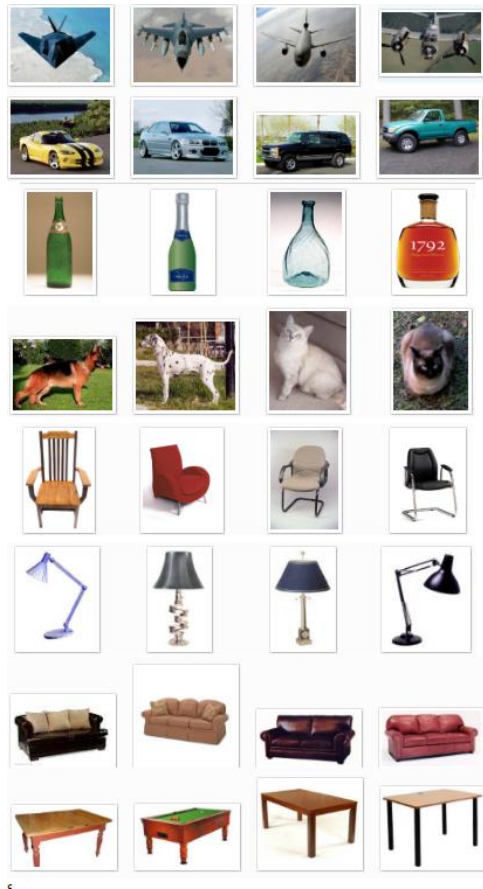


Figure 1.3: Some examples from the RGB-JPEG database.

### **ALOI database**

We take a subset of the Amsterdam Library of Object Images (ALOI, see (112)) database. It is a color images collection of one-thousand small objects. In order to capture the sensory variation in object recordings, this database contains images representing objects with variation on the viewing angle, the illumination angle and the illumination color. The database contains over a hundred images of each object, yielding a total of 110,250 images for the collection. Each image contains a single object on a black background. Figure 1.5 shows a small subset of this database.

### **BERKELEY database**

The Berkeley Segmentation Dataset and Benchmark (see (188)) was created with the goal of providing an empirical basis for research on image segmentation and bound-



Figure 1.4: Some logical images from the log-JPEG database with, on the left, the original image.

ary detection. To this end, 12,000 hand-labeled segmentations of 1,000 Corel dataset images from 30 human subjects have been collected. Half of the segmentations were obtained from presenting a color image to the subject; the other half from presenting a grayscale image. The public benchmark based on this data consists of all of the grayscale and color segmentations for 300 images. The images are divided into a training set of 200 images, and a test set of 100 images. We take a subset of the color images of this database. It consists of natural images. Figure 1.6 shows a small subset of this database.

## 1.6 Outline

The present work faces different problems strictly connected with the bottom-up object classification task, with the aim of defining a similarity measure. During our research we understood that a first step toward this direction was defining a distance measure between shapes, and that to reach this result we needed some structure able to define shapes themselves. In fact, if primitives of visual recognition were known (like phonemes in speech recognition), then we would be able to define a language for



Figure 1.5: Some examples from the ALOI database.

visual recognition based on the interpretation of these primitives and on specific laws of composition. The representation of visual perception and the perception of the visual representation of perception range over an incredible amount of symbolic traits and structures. Consider, for example, the two drawings in figure 1.7. If we look at the left sketch, few primitive traits are sufficient to denote a face, even a known face (Michelangelo's David); on the other hand, in the second one a rich representation of shadows and lights seems to be not enough to reveal the context and immediately communicate the meaning of the represented object (it is, in fact, a studio of drapery from Leonardo).

By analyzing the previous example we understand that shapes play a very important role. Indeed we put the shape-comparison problem outside of the list previously presented for its complexity and for the extreme importance it has. If the problem of defining a methodology to classify 2-dimensional shape was solved, we would have a great advance in object classification. First of all, shape analysis is very efficient (working with few data); moreover it is possible to "normalize" a shape, obtaining a universal model easier to compare. For these reasons, in this work, to evaluate object similarity, we firstly investigate shape classification. But it is not enough. By working only with shapes, we loose a great amount of information, which can really help us to discriminate among objects. So our methodology tries to perform a dual analysis: from one hand based on shapes, from the other based on color.

After a brief overview on the most known and used techniques, we present our



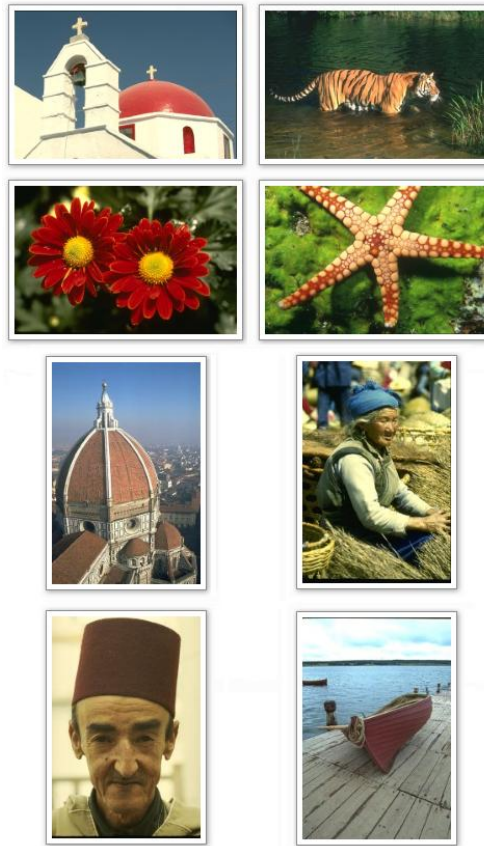


Figure 1.6: Some examples of the used images from the Berkeley database.

investigations and results. In chapter 3 we address the localization problem presenting a segmentation procedure which works without any *a priori* knowledge. Then we present our first approach to understand object similarity: Implicit and Explicit analysis (chapter 4). The idea underlying this methodology is that there are two different interesting and complementary aspects to recognize an object: shape representation (paradigmatically connoted by the traits of the David) and visual features (paradigmatically connoted by the Leonardo's drawing). The explicit description is used to account for the main traits of what, in the human representation, connotes a principal source of information regarding a category: the shape. The implicit one, on the other hand, accounts for shadows and lights, color, and *sense* of volume. While studying this approach, we had to face two different kinds of problem: how to compare different data sets (usually of different size) and how to compare shapes. In chapter 5, we introduce a specific methodology, which tries to face this latter problem: relying on a description based on critical points, we determine a similarity distance between

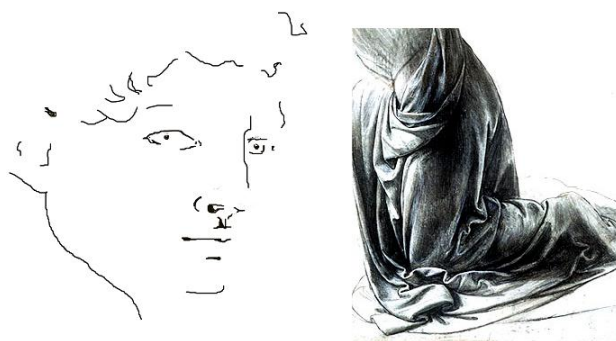


Figure 1.7: On the left: few primitive traits of the Michelangelo's David; on the right, a part of a studio of drapery by Leonardo.

shapes. At the same time, we also introduce a description able to represent all the information relevant for our analysis. Instead, in chapter 6, we try to address the classical problem of comparing two different-sized datasets with the aim of evaluating the distance between them. Starting from the idea that under every data set we can imagine a probability density function (pdf), we recall the Parzen's assumption that every pdf can be approximated to any degree by a Gaussian mixture. Then we estimate the best fitting mixtures of Gaussians for each data set and we calculate the distance between them. The measure obtained is used as an evaluation of the distance between the two starting sets. Specifically, we show experimentally how it is possible to use the number of modes to obtain some estimation of the number of components of a mixture of Gaussians. Chapter 7 integrates all the results and, finally, chapter 8 presents our conclusions and future works.



## Chapter 2

# Objects and Patterns: a historical perspective

### 2.1 Object recognition and classification

Object recognition is one of the central research topics in the Artificial Intelligence field. It is also one of the most actively studied subjects with many real applications in the same areas of object classification: security, medicine, document analysis, image and video retrieval and others. The aim is to detect and to analyze arbitrary objects in a still or moving image. Nowadays, even if a large number of different methodologies exist, it has not been developed an optimal approach. It is possible to find very efficient solutions in some specific scenarios (i.e. faces, vehicles, pedestrians), but a general solution is unknown. In (139) the authors state that even if object recognition, also in a free-form three dimensional scenario, can be considered a well-understood problem with many successful approaches (47; 62; 91; 150; 272; 305), object classification (even in the simplest 2D form), where a previously unseen object must be assigned to a generic object class, is still an open problem. In recent years there have been a lot of studies in the field of object recognition and classification. As already said, these different scenarios present similar problems. So usually the results found in one of them are also useful for the others.

To evaluate similarity it is necessary to find some cues or measurements that remain unchanged, invariants, under a general viewing transformations. The use of pattern recognition techniques can really help to solve some aspects of these problems. This is the reason why, in the following sections, we first present a brief overview on the most used pattern recognition methodologies, a theoretical and experimental basis on which relying novel approaches. Obviously we mainly focus our attention

on the techniques used in the present work.

## 2.2 Statistical approaches to classification

In the following sections we briefly analyze some of the pattern recognition methodologies used for multi-class categorization problem; other statistical approaches for image interpretation are introduced in chapter 4. In the classical scenario there is a predictor  $y(\mathbf{x})$  which takes as input a vector  $\mathbf{x}$  and tries to assign it to one of  $K$  discrete classes  $C_k$ , where  $k = 1, 2, \dots, K$ . It is possible to divide the input space into a collection of regions labeled according to the  $y$ -classification. In the most common scenario, these  $K$  classes are taken to be disjoint. The decision boundaries between the regions can be of different shape (rough or smooth). Having  $K$  different classes, one can also introduce a  $K$ -length vector  $\mathbf{t}$  (known as target vector), which represents the belonging class; specifically if the  $i$ -th element is in class  $j$ , then all element of  $\mathbf{t}_i$  are zero, except the  $j$ -th element (set to 1):

$$\mathbf{t}_i = (0, 0, \dots, 1, \dots, 0)$$

Usually a discriminative function  $y(\mathbf{x})$ , able to return the right class, does not exist, but one has to work in a probabilistic framework. In such a scenario, it is possible to rely on a conditional probability distribution

$$p(C_i|\mathbf{x})$$

modeled in an inference stage, which can help to take the optimal decision. A general model for  $y(\mathbf{x})$  can be obtained by considering a non-linear function  $f(\cdot)$  (known as activation function), which takes as input a linear combination of the element of  $\mathbf{x}$ :

$$y(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x} + w_0)$$

where  $\mathbf{w}$  is a parameter vector. We can distinguish two different scenarios. In a supervised approach we have a training set containing  $N$  couples  $(\mathbf{x}_i, \mathbf{t}_i)$  where  $\mathbf{t}_i \in [0, 1]^K$  is the label assigned to the input value  $\mathbf{x}_i$ . In an unsupervised scenario, instead, we do not know in advance the belonging classes. A new value arrives and we want to decide to which class it belongs or how to divide the complete set of samples. Many different techniques exist that try to solve this problem. We will give a brief description of some of them.

### 2.2.1 Fisher Linear Discriminant analysis

Fisher Linear Discriminant analysis (or LDA) is a linear classification technique, that is the decision surfaces are hyperplanes. It performs a dimensionality reduction of the  $K$  original classes to  $K - 1$ . This methodology, once defined a projection matrix  $D$ , multiplies each sample for it. Obviously the projection leads to a considerable loss of information and classes that are well separated in the original space may become strongly overlapping. LDA allows to define a matrix  $D$  which minimizes the distance between samples of the same class and maximizes the distance among different classes. Without loss of generality we consider the binary case ( $K = 2$ ): class  $C_1$  with  $N_1$  points and class  $C_2$  with  $N_2$  points. By using

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

the datapoints are projected into a labeled set in the one-dimensional space  $y$ . Now it is necessary to evaluate the intra- and inter-class distances. The mean vectors of the two classes are easily given by:

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{i=1}^{|C_i|} x_i$$

The separation of classes can be estimated as the separation of the projected class means. The mean of the projected data from class  $C_k$  is represented by  $m_k = \mathbf{w}^T \mathbf{m}_k$ . So we have to maximize:

$$m_2 - m_1 = \mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1)$$

To avoid that this expression becomes arbitrarily large by increasing the magnitude of  $\mathbf{w}$ , its length is constrained to be equal to 1.

Instead the within-class variance of the projected data of class  $k$  is given by:

$$s_k^2 = \sum_{i=1}^{|C_k|} (y_i - m_k)^2$$

where  $y_i = \mathbf{w}^T \mathbf{x}_i$ . So the total within-class variance for the whole dataset is simply  $s_1^2 + s_2^2$ . The Fisher criterion is defined to be the ratio of the between-class variance to the within-class variance:

$$F(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

Now it is possible to introduce the between-class matrix

$$S_B = (\mathbf{m}_2 - \mathbf{m}_1) (\mathbf{m}_2 - \mathbf{m}_1)^T$$

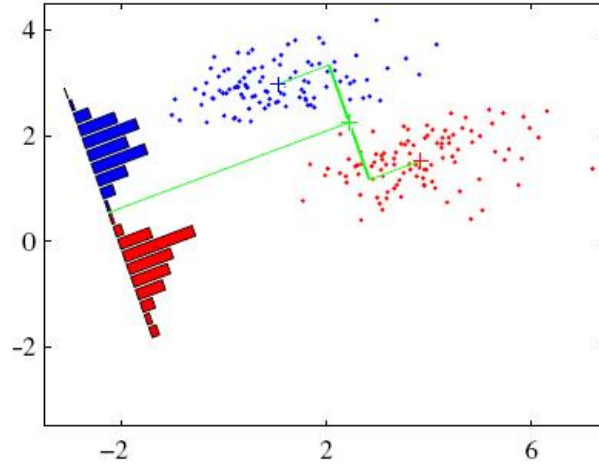


Figure 2.1: Two sets of 2d data points and the corresponding projection based on Fisher linear discriminant (taken from (27), pag. 188 fig 4.6).

and the within-class covariance matrix

$$S_W = \sum_{i=1}^{|\mathcal{C}_1|} (\mathbf{x}_i - \mathbf{m}_1) (\mathbf{x}_i - \mathbf{m}_1)^T + \sum_{i=1}^{|\mathcal{C}_2|} (\mathbf{x}_i - \mathbf{m}_2) (\mathbf{x}_i - \mathbf{m}_2)^T$$

so we can rewrite the Fisher criterion as

$$F(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

Differentiating with respect to  $\mathbf{w}$ , we find that  $F(\mathbf{w})$  is maximized when

$$\left( \mathbf{w}^T S_B \mathbf{w} \right) S_W \mathbf{w} = \left( \mathbf{w}^T S_W \mathbf{w} \right) S_B \mathbf{w}$$

from which

$$\mathbf{w} \propto S_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$$

known as Fisher's linear discriminant (figure 2.1 shows an example). Obviously the procedure can be easily extended to scenarios with  $K > 2$ .

### 2.2.2 Kernel methods and Support Vector Machines

Kernel methods received great attention in recent years. Introduced into the field of pattern recognition by Aizerman et al. (2), neglected for many years, they were re-

introduced to face non-linearity within the Support Vector Machines method. The idea behind them is simple, but very powerful: to classify two datasets which are not linearly separable, a mapping function  $\phi(\mathbf{x})$  is introduced and then one works in the transformed space, at the cost of the transformation (see figure 2.2). The concept of a kernel formulated as an inner product in a feature Hilbert space brought to extensions of many well-known algorithms by using some kernel substitutions. In fact, in a procedure where the input vector  $\mathbf{x}$  enters only in the form of scalar products, we can easily replace that product with some other choice of kernel. Interesting examples of such substitutions include nearest-neighbour classifiers and kernel Fisher discriminant (198; 243; 12) and the non-linear variant of PCA (249). In other words given an input space  $S$  and a mapping function  $\phi$ :

$$\phi(\mathbf{x}) : S \rightarrow H$$

a kernel  $K$  is a function

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_H$$

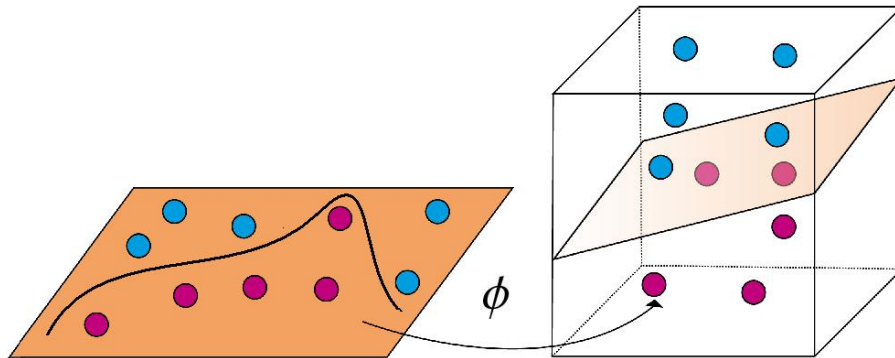


Figure 2.2: A graphic example of how a Kernel method works.

More generally one can define a valid kernel, also without knowing the function  $\phi(\mathbf{x})$ . Admissible kernels can be specified with the finite positive definite property: a necessary and sufficient condition for a function  $K$  to be a valid kernel is that:

$$\sum_{i,j=1}^n c_i c_j K(x_i, x_j) \geq 0$$

for any  $n \in \mathbb{N}$ , any subset  $\{x_1, \dots, x_n\}$  of the input space and any choice of real numbers  $c_1, \dots, c_n$ . So there can be a big set of admissible functions. There



are numerous forms of kernel functions in common use; many have the property of being functions only of the difference between the arguments, so that  $K(\mathbf{x}, \mathbf{y}) = K(\mathbf{x} - \mathbf{y})$ , which are known as stationary kernels. A further specialization involves homogeneous kernels (or radial basis functions) which depend only on the magnitude of the distance between the arguments so that  $K(\mathbf{x}, \mathbf{y}) = K(\|\mathbf{x} - \mathbf{y}\|)$ . We address the interested reader to (135; 250; 258) for a more exhaustive analysis on kernel methods.

Support Vector Machines (SVM) are a group of supervised learning algorithms that can be applied to regression, classification and novelty detection. In the mid 90s using the first methodologies for incorporating prior knowledge (248), SVMs became competitive (in particular their application was in the handwritten digit classification task). In a short period of time, SVMs have found numerous applications in a wide range of fields. The SVM algorithm is based on the statistical learning theory and the VapnikChervonenkis (VC) dimension (290). SVM models were originally defined for the classification of linearly separable classes of objects: they are able to find the unique hyperplane having the maximum margin in a two-class dataset. Most interesting is the use of SVM to separate classes that cannot be separated with a linear classifier. In such cases, the coordinates of the objects are mapped into a feature space using nonlinear feature function. The feature space is a high-dimensional space in which the classes can be separated with a linear classifier. The nonlinear mapping by the feature functions is computed with special nonlinear functions called kernels. Additional information on SVM can be found in (257; 258; 250; 19; 204).

### 2.2.3 Markov Random Fields

From their first introduction in image analysis (by Geman and Geman in (110)), Markov Random Fields (MRF in the following) have been extensively used to face computer vision problems and, specifically, in segmentation and classification tasks (see, for example, (236; 201; 85; 168)). To understand MRF we first need to briefly recall some notations about graphs and graph cuts.

A graph  $G = \langle V, E \rangle$  is defined as a set of nodes or vertices  $V$  and a set of edges  $E \subseteq V \times V$  connecting neighboring nodes. The order of a graph is  $|V|$  (the number of vertices), the graph's size is  $|E|$  (the number of edges) and we define the degree of a vertex  $v$  the number of edges that connect to it (where an edge that connects to the vertex at both ends is counted twice). We now focus our attention specifically on undirected graphs, where each pair of connected nodes is described by a single edge  $e = \{p, q\} \in E$ . In our case of interest, the node set  $V$  contains two specially designated *terminal* nodes  $s$  (source) and  $t$  (sink) and a set of *non-terminal* nodes  $Q$ :  $V = \{s, t\} \cup Q$ . Edges between nodes in  $Q$  are called  $n$ -links where  $n$  stands for

"neighbor", while the so called  $t$ -links are used to connect nodes of  $Q$  to terminals. All graph edges  $(p, q) \in E$ , including  $n$ -links and  $t$ -links, are assigned some non-negative weight (cost)  $w(p, q)$ . We will denote with  $N$  the set of all  $n$ -links. An  $s - t$  cut (in the following sometimes just called cut) defines a partitioning of the nodes in the graph into two disjoint sets  $S$  and  $T$ , such that the source  $s$  is in  $S$  and the sink  $t$  is in  $T$ . We represent a cut  $C$  either by the two subsets  $S$  and  $T$  ( $C = \{S, T\}$ ) or by the subset of the edges in  $E$  severed to obtain the partitioning. The cost of a cut  $C$  is the sum of the weights of "boundary" edges  $(p, q)$  with  $p \in S$  and  $q \in T$ :

$$|C| = \sum_{(p,q) \in C} w(p, q)$$

In recent years combinational min-cut algorithms on graphs have emerged as an increasingly useful tool for problem solving in vision. The 5<sup>th</sup> chapter (by Boykov and Veksler) of (219) presents an interesting overview about graph cuts in vision. As the authors show, there exist a lot of interesting links connecting graph cuts with other combinational algorithms (dynamic programming, shortest paths (39; 161)), statistical physics, simulated annealing, and other regularization techniques (123; 43; 142), sub-modular functions (164), random walks and electric circuit theory (121; 122), Bayesian networks and belief propagation (276), integral/differential geometry, anisotropic diffusion, level sets and other variational methods (273; 41; 6).

Now we can introduce MRFs. A Markov Random Field consists of a set  $P$  of sites  $p$  (usually in computer vision  $P$  represents the set of the pixels of an image  $I$ ), a neighborhood system  $N = \{N_p | p \in P\}$  (where each  $N_p$  is a subset of the elements in  $P$  which describes the neighbors of  $p$ ) and a field of random variable  $F = \{F_p | p \in P\}$ . Each random variable  $F_p$  takes a value  $f_p$  in some set  $L = \{l_1, l_2, \dots, l_k\}$  of the different possible labels. We define an assignment  $f = (f_1, f_2, \dots, f_{|P|})$ , which correspond to the joint event  $\{F_1 = f_1, F_2 = f_2, \dots, F_{|P|} = f_{|P|}\}$  and so to a specific realization of the field.

In order to be a MRF,  $F$  must satisfy

$$P(f_p | f_{P \setminus \{p\}}) = P(f_p | f_{N_p})$$

and

$$P(f_p) > 0$$

both for all  $p \in P$ , namely each variable depends on other random variables only through its neighbors in  $f_{N_p}$ . One of the most important result about MRF is the Equivalence Theorem proved by Hammersley and Clifford (see (128)) that associates the probability of a specific assignment  $f$  with a sum over all the cliques in the neighborhood system  $N$ . Specifically:

$$P(f) \propto \exp\left(-\sum_C V_C(f)\right)$$

where  $V_D$  is a clique potential which describes the prior probability of a particular realization of the element of the clique  $D$ . Considering MRFs whose clique potentials involve only pairs of neighbors, we can rewrite the previous formula as:

$$P(f) \propto \exp\left(-\sum_{p \in P} \sum_{q \in N_p} V_{p,q}(f_p, f_q)\right)$$

We have to consider that, in common case, we can not observe directly  $F$  and we have to estimate it starting from an observation  $O$ . Considering

$$P(O|f) = \prod_{p \in P} P(O_p = o_p | F_p = f_p)$$

where  $O_p$  is the observable label for pixel  $p$ . We want to obtain the association  $f \in L^m$  which maximizes the posterior probability  $P(f|O)$ , knowing (from Bayes' law) that  $P(f|O) \propto P(O|f)P(f)$ . Under these considerations, our maximum a posteriori (MAP) estimate  $f$  should minimize the posterior energy function

$$E(f) = -\sum_{p \in P} \sum_{q \in N_p} V_{p,q}(f_p, f_q) - \sum_{p \in P} \ln(P(O_p = o_p | F_p = f_p))$$

Let  $\delta(\cdot)$  be the unit impulse function, we define a Generalized Potts Model MRF (GPM-MRF) as an MRF having a clique potential for any pair of neighboring pixels  $p$  and  $q$  given by

$$V(f_p, f_q) = u_{\{p,q\}} \cdot (1 - \delta(f_p, f_q))$$

This MRF is also isotropic if we do not take into account the order of the couple (i.e.  $\{p, q\}$  is a set and not a couple). In (44) the authors define the posterior energy function of a GPM-MRF, under the conditions presented above. Afterwards they state (and demonstrate) that minimizing that energy function  $E(f)$  over  $f \in L^m$  is equivalent to solve a multiway cut problem on a specific graph. Their demonstration is in a multilabel framework, but it clearly holds also in our simpler case of  $m = 2$ . From a complexity point of view, this result is extremely important. In fact even if a general multiway minimum cut problem is still NP-complete, there are a lot of provably good approximations with near linear running time (see (72)). We will use these results in our segmentation procedure.

## 2.3 A perceptual organization

In a fundamental paper Chen (53) classifies the study of the primitives of visual perception into a *Great Divide* between the “early feature analysis: local-to-global” holding the viewpoint that perceptual processing is from local to global, and “early holistic registration: global-to-local”, i.e. the perceptual process coding the wholes prior to perceptual analysis of their separable properties or parts. Chen emphasizes that “physically or computationally simple does not necessarily mean psychologically simple or perceptual primitive”. And indeed, as Chen notes, from an opposite different perspective, because psychologically plausible does not necessarily mean computationally plausible, there is the need to determine the primitives of a computationally feasible language for recognition. In his seminal paper on *visual perceptual organization* (225) Pentland has pointed out that perception is successful because of an inner structuring of our environment, and because of the human ability to identify the connections between these environmental regularities and primitive elements of cognition. The model-based approaches to perception have been strongly influenced by this view (see (86; 303; 233)). Among the model-based approaches, the constructive approach, known as recognition by components (RBC), was pioneered by (186; 254; 225), and especially by (23), and finally by (303; 241). Biedermann’s (see (24) for a review) proposal has been one of the first pointing out the need of primitives in object representation in order to deal with a categorical description of objects, as composed by common and similar parts. However a criticism towards the 36 qualitative geons (“geometric icons”) introduced by Biedermann (see figure 2.3) was that they capture no metric shape information, being such information essential for interacting with the object and for distinguishing between subclasses of an object (see (79)). Another apparent drawback with the RBC approach is that it supports a viewpoint-invariant notion of objects’ visual representation, while from psychological experiments (see e.g. (132)) it has been argued that recognition performance might be affected by relatively small perturbations in viewpoint. For example, experiments in (208) show that view-invariant recognition can be achieved by a viewer-centered system that interpolates between a small number of stored views.

To face the different views, (163) has introduced the concept of aspect graph, a structured graph of the set of aspects of an object, where the edges of the graph are the transitions between two neighboring stable views and a change between aspects is called a visual event. Further more, (80) has suggested a 3D modeling of an object via a hierarchical aspect representation based on the projected surfaces of the primitives. They present a new grouping concept based on aspects recovered in the sensed image: the aspect hierarchy is used to infer a set of volumetric primitives and their connectivity. Similarly in (229; 230) objects categories are represented through their common parts, which are recognized according to a decomposition into primitives, and recomposition is achieved according to an algebra of figures and a Bayes aspect

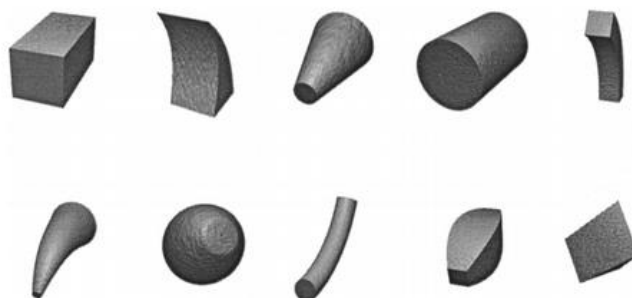


Figure 2.3: A subset of the famous Biedermann's geons.

graph. Recently (see e.g (5; 300; 35; 95; 96)), in the stream of object representation approaches based on categories gathering similar parts, the problem has been faced in new terms, considering features and appearances of parts, so as to overcome all the occlusion and view-point problems raised in the model-based approaches. In particular, Perona and colleagues (see (46; 95; 96)) have proposed to model objects as constellation of parts, proposing a successful method to learn object categories from cluttered data, with unsupervised labeling, in so relieving from the burden of manually labeling the images. In (96) features are found using the detector described in (151; 152), and features are represented in an appearance space, where each part composing an object has a Gaussian density. Analogously both shapes and relative scales are represented by a joint Gaussian, and thus the recognition model is based on maximum likelihood estimation of the parameters composition. A difficulty with this approach, at least from a cognitive point of view, is to understand which features one is akin to give up, while retaining the recognition of an element in the scene. For example, eyes without a face (or viceversa) might still lead to the recognition of a person, which is not intuitive. Indeed the model lacks a hierarchy or a preference/causal criterion on the features to be preferred and kept at last. However the approaches of (46; 95; 96) are extremely interesting because they address a non obvious compositional aspect of recognition, based on the source of information, or, as they call them, constellation of parts.

In this direction it is also important to cite the work of Papageorgiou and Poggio (218): a system that represents object classes in terms of local oriented multi-scale intensity differences between adjacent regions in the images, trained using a support vector machine (SVM) classifier. This latter approach uses generic features; in contrast in (134) a component-based face detection system is described that uses class-specific features. The system automatically learns components by growing image parts from initial seed regions until error in detection is minimized. From these image parts, components are chosen to represent faces. In this system, the image

parts and their geometric arrangement are used to train a two-level SVM. Another object recognition system that uses fragments from images is described in (285). The authors choose fragments from training images that maximize the mutual information between the fragment and the class it represents. Extremely interesting is the work with volumetric representation (166; 195; 224) and in particular the work by Xing, Liu and Yuan (306). In this area there has been much research work focused on object recognition with superquadrics or geons (37; 81; 166; 224; 237). In (37), superquadrics and geons were especially used for volumetric representation, and interpretation tree (124) was implemented for 3D recognition. Other interesting results have been reached using invariant moments (particularly with respect to shape analysis). Moment invariant techniques were first introduced by Hu in 1962 (137), and were more recently used by Maitra (see (185)). Hu's moments are defined as the projection of the image function  $f(x, y)$  onto the (non-orthogonal) monomial basis set. The reconstruction of the image is therefore computationally expensive due to the redundant information present in these moments. (277) suggested the use of orthogonal moments to overcome the problems associated with regular moments. The orthogonality property of the basis ensures the linear independence of orthogonal moments; various orthogonal moment orders correspond to different characteristics of the image. Other moment-based techniques used in object recognition include Legendre (277; 174; 222), pseudo-Zernike (278), rotational (38; 266) and complex (1) moments. We address the interested reader to (235) and (283).

Features detection and analysis represent another interesting research field in object recognition. The most famous detector is the Harris corner detector (130), based on the eigenvalues of the second-moment matrix. Then, starting from the observation that Harris corners are not scale-invariant, in (178) the author introduced the concept of automatic scale selection; his approach detects interesting points with their scale characteristic. Harris-Laplace detectors (199) can be considered a subsequent robust refinement, characterized by an high repeatability. Indeed, during the last decades, several methodologies have been developed in this direction. Among them it is important to cite complex features (13), steerable filters (102), phase-based local features (49) and Gaussian derivatives (98). A particular attention has to be focused toward SIFT and SURF. SIFT has been introduced by Lowe in (182). It represents the distribution of smaller-scale features within the interest point neighborhood. The difference of Gaussian function provides a close approximation to the scale normalized Laplacian of Gaussian

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1) \sigma^2 \nabla^2 G$$

whose maxima and minima produce the most stable image features compared to gradient, Hessian or Harris corners. (182) introduces an efficient way of computing  $D(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma)$  based on pyramids: incremental convolu-

tion is performed starting from the given image to produce images which in the scale space are separated by a constant factor  $k$ ; every octave in the scale space (i.e. every interval in which the  $\sigma$  value doubles) is divided in  $s$  intervals, so  $k = 2^{1/s}$ ; adjacent images are subtracted to produce difference of Gaussians; once an octave is complete, the image that is characterized by the double of the initial  $\sigma$  is sub-sampled by a factor of 2; the procedure iterates until the desired number of octaves is reached. SIFT have been used in a wide range of applications: 3D reconstruction, motion tracking and segmentation, robot localization, image panorama stitching, epipolar calibration and obviously recognition of particular object categories in 2D images. Recently Bay, Tuytelaars and Van Gool in (14) introduced another powerful tool for local features: SURF (Speeded Up Robust Features). SURF is a scale- and rotational-invariant interest point detector and descriptor. While SIFT which approximates Laplacian of Gaussian with Difference of Gaussians, SURF approximates second order Gaussian derivatives with box filters. Image convolutions with these box filters can be computed rapidly by using integral images (294). As a consequence, the standard version of SURF is faster than SIFT and moreover it seems to be more robust against different image transformations. Indeed SURF outperforms the state-of-the-art both in speed and in accuracy.

We will introduce other methodologies in chapter 5.

## **Part II**

# **An approach to implicit and explicit analysis**





## Chapter 3

# Bottom-up Image Segmentation

### 3.1 Introduction

Image segmentation and grouping is one of the most important challenge for computer vision. Image segmentation is obviously the first step of every higher level visual analysis. In fact, before starting to analyze and to interpret the sub-parts of a digital image, the system has to individuate some coherent and consistent blobs of pixels. The aim is to segment the scene into regions with particular semantic content, i.e. the constituent objects. It is a very hard task. And so, even if in the last three decades have been developed a lot of different methodologies to face with segmentation, nowadays there is no a general and completely satisfactory solution. An image segmentation technique has to take into account position, texture, color and much more information to decide whether some pixels have to be grouped into the same cluster. But the system has to look for clusters which can represent meaningful objects, without having any precise definition of what an object is. If we are working on a digital photo of a car, what do we expect from a well-defined segmentation technique? Has it to return a big block containing the whole vehicle or do the different parts of the car have to be analyzed singly? Consider that structured objects are almost always composed of different parts, each with different color and texture characteristics. And how can we take the right decision?

Most existing segmentation algorithms are designed simply to detect homogeneous units in images. In real scenarios they correspond only approximately to objects or to object parts. So, usually, those methods generate an over-segmentation: objects, that the system wants to individuate as single units, are (over-)segmented in multiple parts. To take meaningful decision, the system may have in advance information about the objects, and it has to use it to understand how to combine the single parts. By knowing something about the semantic associated to the blocks of pixels,

the system could try to group them together. So, to achieve this goal, the system has to recognize the objects (or its constituent parts). But object recognition (or classification) usually has to rely on image segmentation results: we first have to individuate the units that have to be studied and, only after that, we can start to analyze and recognize them; this is a classical *chicken-and-egg* problem. Image segmentation seems to be a completely low-level problem but usually, to improve the performances or simply to obtain meaningful results, the execution of the process should be task driven, i.e. supported by independent high-level information. In literature there exist a lot of segmentation techniques that use both low-level and high-level cues relying very often on semantic information to drive the process. If we can access some *a priori* knowledge about the system, we can obtain a robust and compact model. Otherwise we can rely on a very flexible and completely based on low-level information one. In the present work, considering the scenario and the motivations presented in chapter 1, we always try to favor a purely low-level methodology. Comaniciu and Meer (see (65)) rightly assert that methods which rely upon *a priori* knowledge, as well as methods which implicitly assume something about the structure of the output, could not handle the complexity of a real feature space.

In this chapter we present the segmentation problem: given a digital image, the system has to locate and isolate the possible objects inside the scene. First of all we present a brief overview about segmentation methodologies already existing. We divide the literature in bottom-up and top-down approaches. The difference between them relies in the fact that these latter use high-level information (usually thanks to a first training step) to accomplish the task. Working under the completely-unknown environment hypothesis, we obviously focus our attention on bottom-up techniques. We present our approach which consists of subsequent steps of a digital image processing. We first filter the image to remove (Gaussian) noise. Then we combine color, intensity and position analysis trying to group together the similar pixels. We present our experimental results and discuss the problems still open.

## 3.2 A brief overview on the existing methodologies

At the beginning of 20<sup>th</sup> century, Gestalt movement introduced the idea that perceptual grouping plays a fundamental role in all the human visual processes. And in fact, as said before, almost every mid- or high-level vision problems has to rely heavily on segmented images. The goal of image segmentation is to cluster pixels into salient regions corresponding to individual objects or also to surfaces or natural parts of objects. Wertheimer (in (301)) pointed out the importance of perceptual grouping and organization in vision and also listed several key factors (such as similarity, proximity and good continuation) which lead to visual grouping. Image segmentation is a

longstanding but currently largely unachieved goal in Artificial Intelligence research. And it can be considered probably the most studied problem of computer vision. According to the ideas presented in the previous section, the existing methodologies can be divided into two big families: top-down and bottom-up approaches.

As stated before, given a digital still image, it could be not clear what is the expected output of a segmentation algorithm, and, in fact, segmentations performed (by hand) by human beings often return incoherent responses. Imagine to analyze a picture representing a man wearing colored cloths. We can isolate the different cloths (shirt, pants), but also group together all the blocks of the person. There may not be a correct answer in the partitioning of an image and we have to work with different hypothesis. The more mid- and high- level knowledge (about symmetries, object model, scene, ...) we use, the more different grouping of the low-level cues (such coherence, color, brightness, texture, motion) can be performed.

### 3.2.1 Bottom-up approaches

The early computer vision research has focused on developing models for bottom-up segmentation processes. Bottom-up approaches solve the chicken-and-egg problem of using object information to individuate the (same) object location in a digital image. Starting with the assumption that we cannot use semantic information to guide the object segmentation, the system works directly on the input data. Moreover, this approach has an interesting theoretical basis in psychology: the idea of Fodor (see (99)), in fact, about information encapsulation in the peripheral or input systems is consistent with a low-level approach. Although Wertheimer (at the very beginning of Gestalt movement) suggested that familiarity might influence perceptual organization, the work of Gestalt psychologists (including Wertheimer) is typically viewed as an attempt to identify bottom-up heuristics for organizing the visual field.

There exist many bottom-up methodologies. For example in Marr's model of visual processing, the grouping of features represented in the raw primal sketch corresponds to the full primal sketch. And this first phase is completed without receiving top-down input from the object recognition stage (see (187)). Also Ullman in (284) has proposed a model-based system for visual recognition in which object knowledge is used to guide the search for and the interpretation of features in the visual field. However, his system does not use this stored object knowledge to guide any earlier image-segmentation process. Among the first bottom-up works, it is also interesting to recall the connectionists models of image segmentation. Mozer et al. (see (202; 203)) trained a connectionist network to segment images consisting of two overlapping objects. The significant contribution of this work is that it exploits the ability of connectionist networks to learn and to discover grouping principles without

needing heuristics built in by the programmer.

In the following, according to Cheng (see (55)) we group the bottom-up approaches into two main families: region-based and contour-based methods.

### **Region-based methods**

These methodologies try to solve the segmentation problem, searching for consistent regions in the image. The researchers presented, mainly in the beginning, methods which heavily relied on graph theory. The earliest graph-based methods use fixed thresholds and local measures in computing a segmentation. The work of Zahn (309) presents a segmentation method based on the minimum spanning tree of the graph. This method, indeed, has been applied both to point clustering and to image segmentation. In this latter case the edge weights in the graph are based on the differences between pixel intensities. The segmentation criterion of Zahn seemed to be inadequate: differences between pixels within a high variability region could be larger than those between the pixels belonging to two different uniform areas with a little intensity difference. Urquhart in (287) attempted to address this shortcoming by normalizing the weight of an edge using the smallest weight incident on the vertices touching that edge. When applied to image segmentation problems, however, this is not enough to provide a reasonable adaptive segmentation criterion.

One of the most important contributions among the graph-based methodologies was given by Wu and Leahy in (304). They introduced a novel theoretic approach for data clustering, easily extensible to image segmentation. The data are represented by an undirected adjacent graph with arc capacity assigned to reflect the similarity between the linked vertices. The idea is to remove arcs from the graph to obtain mutually exclusive subgraphs such that the largest inter-subgraph's maximum flow is minimized. Wu and Leahy were the first to introduce a minimum cut criterion in graphs. Their methodology is very interesting and produces meaningful results (in their paper they showed how to successfully segment an MR image of a brain), but it favored small components. To overcome this problem Shi and Malik (see (259)), proposed the normalization cut criterion. They worked, as the previous ones, with graph cuts but including some elements to evaluate also the global structure. Normalized cut is in fact an unbiased measure of dissociation between subgroups of a graph, and it has the nice property that minimizing normalized cuts leads directly to maximizing the normalized association, which is an unbiased measure for total association within the subgroups. The authors, instead of looking at the value of total edge weight connecting the two partitions, as the previously presented methodology did, compute the cut cost as a fraction of the total edge connections to all the nodes in the graph. The drawback is that it yields a NP-hard problem. Even if the authors

presented several approximation methods to solve the problem quickly, the cost of those approximations is still too high for many common applications and moreover the error generated is not well understood.

In (93) Felzenswalb and Huttenlocher presented another interesting method based on graph analysis. Starting from a digital image, the authors define an undirected graph with vertices corresponding to the pixels of the image, and the edges corresponding to a measure of dissimilarity between them. The segmentation is induced trying to capture the global image characteristics. They adaptively adjust the segmentation criterion based on the degree of variability in neighboring regions of the image. A boundary is detected when the degree of variability across the boundary of two regions is large relative to the degree of variability inside at least one of the regions. The methodology presented is very efficient and it gives interesting results in many scenarios. Very interesting (and with a lot of common elements with our work) is the approach introduced by Comaniciu and Peter (see (65)). They defined a general nonparametric technique for the analysis of multimodal feature space and they used this methodology to obtain image segmentation. Their procedure relies on an iterative well-known mode detection and clustering procedure: mean shift, proposed firstly by Fukunaga and Hostetler in 1975 (see (105)), a technique that we widely use, in different scenarios, in the present work. We briefly recall (see the section 6.7.4 for further details) that given a set of points, the mean shift procedure defines, for each point of the set, a vector proportional to the normalized density gradient estimate, and so a vector which always points toward the direction of maximum increase in the density. In this scenario, each pixel is associated with a significant mode of the joint domain density located in its neighborhood.

### Contour-based methods

The Gestalt movement has emphasized the importance of contour closure in human vision (see (153; 89; 88)) as one of the most significant grouping factors. Among the first attempts on contour-based grouping we have Parent and Zucker's work (see (220)) using relaxation methods, Shashua and Ullman's work (see (256)) on saliency networks and Guy and Medioni's work (see (127)) with voting schemes. We want to recall also Mumford, who first pointed out the connection between minimal energy curves and stochastic processes (see (205)) and discussed its application to computer vision. Zhu and Yuille (see (313)) used both boundary and region information within an energy optimization model. They assumed an image consisting of a set of homogeneous regions. The homogeneity was defined on some low-level properties such as intensity, color or texture. The drawback is that their method, in order to achieve good performances, needs a set of initial seeds to be placed correctly inside each region. In addition, they use the gradient descent algorithm to compute minimum

energy, which in many cases can only find the local minima. In a similar direction it is important to recall the work of Jermyn and Ishikawa (see (149)). They proposed a new form of energy function on edges that takes the form of a ratio of two integrals around the boundary. The numerator of the energy is a measure of the flow of some quantity, while the denominator is a generalized measure of the length of the boundary. The main contribution of this form of energy is that the general types of region information are allowed to be incorporated into the energy function.

Very interesting are also the works on active contours and deformable models. Active contours models, also known as snakes, are a framework for delineating an object outline. It attempts to minimize an energy associated to the current contour as a sum of the internal and external energy. The external energy is supposed to be minimal when the snake has a shape which is supposed to be relevant considering the shape of the sought object, while the internal when the snake is at the object boundary position. The seminal works in this area are (154) by Kass, Witkin and Terzopoulos and (28) by Blake and Zisserman. In the last decade active contours obtained new attention thanks to the work of Tony Chan (see (52)). This approach can detect objects whose boundaries are not necessarily defined by gradient, by minimizing an energy which can be seen as a particular case of the minimal partition problem. The stopping term does not depend on the gradient of the image, as in the classical active contour models, but is instead related to a particular segmentation of the image.

### 3.2.2 Top-down approaches

As said before, we are interested in a bottom-up image segmentation approach. It is very useful, however, to give a look to the most popular top-down methodologies developed in the last years. It is easy to understand that low-local cues are not enough to get a high-level segmentation. If we want the system to return complex objects (by grouping together parts having very different surface characteristics), the use of some *a priori* knowledge is unavoidable. And, according to much empirical and theoretical research, also in human beings prior knowledge partly guides visual processing. Following the interesting overview of Vecera and Farah (291), we present some works which can be considered the basis of top-down segmentation. (240; 302) for instance introduced the word-superiority effect: the perception of an individual letter is really improved when it occurs in the context of a meaningful word as compared to when it appears either in a non-word or in isolation. McClelland and Rumelhart (see (192)) provided a mechanism which can explain these results, suggesting that they could be due to partial activation of visual word representations interacting with intermediate letter representations. Interactive visual processing has also been demonstrated by Prinzmetal and Millis-Wright in (234). They found that visual feature integration is influenced by cognitive and linguistic factors. Interactive processing in higher-level

vision is also suggested by context effects in object naming (217). Objects occurring in their correct context were named more accurately than visually similar objects occurring in an improper context. We want also to recall Biedermann's work (see (25)). His contribution demonstrated that scheme-level information-influenced object recognition is consistent with interactive processing. We address the interested reader also to other works which present evidences from human vision that indicate that high-level, class-based criteria play a crucial role in the ability to segment images in a meaningful manner: (210; 211; 226; 227).

In computer vision, starting from these considerations, a lot of methodologies have been developed in the last years. Recently great interest has been directed toward the so called class segmentation methods. The general thrust behind these approaches is to use known shape characteristics of objects within a given class to guide the segmentation process. The main difficulty stems from the large variability of shapes within a given class of objects. One interesting solution is proposed by Borenstein and Ullman (see (35)). They presented an approach which uses a fragment-based representation of object classes (a methodology already used also in object recognition). Given an image containing a specific object, they use fragments previously extracted from images of the same object class to produce a consistent cover of the novel object. This cover defines a figure-ground map that associates each pixel in the input image with the likelihood of belonging to an object or background. A common strategy is to start with the easiest pieces and proceed by connecting additional pieces that match in shape, color, edges, texture, etc. In some cases, as information accumulates along this process, pieces must be replaced: locally these pieces provide good matches, but the global structure adds constraints that reject the local matches. Working with horse images, they obtained encouraging experimental results. Shotton et al. (see (261)) assign a class label to a pixel based on a joint appearance, shape and context model. Gould et al. (see (118)) proposed a super-pixel based conditional random field (CRF) to learn the relative location offsets of categories. Even if boundary detection methods based on statistical learning should seem to be closer to edge detection problem, their results represent a great improvement also in image segmentation. In this direction we want to present the work of Martin, Fowlkes and Malik (see (189)). They start with a large set of natural images, manually segmented by multiple human subjects. The output of this first step provides the ground truth label for each pixel as being on- or off-boundary. They model the boundary-probability of each pixel combining the posterior probability learned with some local measurement, such as brightness, color, texture and position.

Dollar et al. in (82) proposed a supervised learning algorithm (which they called Boosted Edge Learning or BEL) for edge and object boundary detection. In the learning stage the algorithm selects and combines a set of features out of a pool with tens of thousands of generic, efficient Haar wavelets in order to learn a discrimi-



native model. The generic features include gradients, difference of offset Gaussian and so on at multiple scales and locations. A very large aperture is used providing significant context for each decision, and some knowledge of Gestalt laws are also implicitly incorporated into the model. Very interesting results are in the work of Heiler, Keuchel and Schnorr (see (133)). They introduced a method for clustering and segmentation based on a semi-definite relaxation of the well-known minimal cut problem on graphs. The advantage of their approach is that it allows incorporating *a priori* knowledge in the clustering process without changing the target function. Instead, available equivalence information is modeled by additional constraints on the optimization problem. This simplifies the interpretation of the results and ensures that different constraints can be combined arbitrarily.

### 3.2.3 Combining bottom-up and top-down methodologies

Some recent methodologies try to combine bottom-up and top-down approaches with the aim to overcome the difficulties we usually have to face when relying only on one of them. We briefly present here some methodologies. In the approach presented by Borenstein, Sharon and Ullman in (34), the top-down approach uses object representation learned from examples to detect an object in a given input image and provide an approximation to its figure-ground segmentation. The bottom-up approach, instead, uses image-based criteria to define coherent groups of pixels that are likely to belong together to either the figure or the background part. The combination provides a final segmentation that draws the relative merits of both approaches: the result is as close as possible to the top-down approximation, but is also constrained by the bottom-up process to be consistent with significant image discontinuities. This work is very interesting because, in contrast with previous approaches (see (54; 180; 308)), it presents a very general combination which can be applied to combine a variety of top-down and bottom-up algorithms. Moreover it is fast (linear in the number of pixels) and takes into account image measurements at multiple scales, converging to a global optimum in just one pass. The methodology is very simple: the authors construct a global cost function that represents the top-down and bottom-up requirements. In the paper they show how the global minimum of this function can be efficiently found by applying the sum-product algorithm, which also provides a confidence map that can be used to identify image regions where additional top-down or bottom-up information may further improve the segmentation. However, like in other works, the training of the bottom-up and top-down modules is performed independently. Specifically for the top-down module it consists of choosing a set of fragments from a huge set of possible image fragments without taking into account any low-level cue.

Levin and Weiss (see (171)) try to overcome the disadvantages of such a choice.

Their algorithm, at run-time, scans a novel image with an object detector which tries all possible subimages until it finds a subimage that is likely to contain the object. Within that subimage they search for object parts by performing normalized correlation with a set of fragments. The location of a fragment gives rise to a local bias term for an energy function. In addition to the local bias, the energy function rewards segmentation boundaries occurring at image discontinuities. The final segmentation is obtained by finding the global minimum of the energy function. While this algorithm is similar at run-time to previous ones, the training method is unique in that it simultaneously takes into account low-level and high-level cues. This problem can be formulated in the context of Conditional Random Fields which leads to a convex cost function for simultaneous training of both the low-level and the high-level segmenter. It is moreover important to highlight that whereas pure top-down algorithms often require hundreds of fragments, this simultaneous learning procedure yields algorithms with a handful of fragments that are combined with low-level cues to efficiently compute high quality segmentations.

#### 3.2.4 Autonomous agent scenarios

If segmentation (and object recognition) on a digital still image is a complex problem, really harder is the attempt of realizing a system which can perform the same tasks on an autonomous mobile agent. Nowadays very few robot systems have demonstrated the ability to individuate objects, inside a realistic environment (such as an office, a kitchen and so on). Obviously there are significant challenges in applying an object recognition approach successfully on a physical platform. In fact even if this scenario could theoretically offer the possibility to obtain on-going some specific additional information which can help the system (a robot could, for example, drive the acquisition of more information about an unknown object), the real-time constraints and the low quality resolution of the images make the problem very challenging. In the following we present some examples of image segmentation systems for mobile robots.

In recent years, also thanks to some international robot competitions (like Semantic Robot Vision Challenge (SRVC) (215), RoboCup@Home (214), RoboCup Rescue (213)) which require participants to design platforms able to autonomously explore an unstructured and completely unknown environment locating specific objects, some interesting results have been obtained. To analyze problems and solutions of this area, we present briefly Curious George, a system developed by the team of UBC LCI Robotics from University of British Columbia (see (196)), which came first in the robot league of the SRVC for both 2007 and 2008 and first in the software league for 2009. Curious George collects low-resolution visual imagery and employs an attention scheme that allows it to identify interesting regions, corresponding to potential objects in the world. These regions will be focused on by the foveal camera,

to obtain high-resolution information, which may be of sufficient quality to allow the objects contained in the region to be recognized. The system uses top-down information to rank the potential objects it has identified, and proceeds to actively collect images of these objects from different viewpoints. The potential objects are selected in the peripheral cameras based on depth from stereo and spectral residual saliency. Spectral residual saliency returns an output that is similar to state-of-art multi-scale saliency measures. The system computes saliency on intensity, red-green and yellow-blue channels. The saliency maps so obtained are summed and regions are detected in their sum using the Maximally Stable External Region (MSER) detector (see (191)). MSER outputs nested regions and these latter come in a wide range of size. The different region sizes map nicely to different zoom setting on the robot camera. The detected MSERs are additionally pruned by requiring that they have a depth different from that of the ground plane. Once identified a potential object, Curious George tries to locate it on a geometric map, associating the object(s) with a particular region of the map. Once the environment has been fully covered, numerous potential objects will likely have been identified and the system starts to perform recognition to identify the true semantic labels for each of these. There is a fundamental difference between this methodology and ours: Curious George needs to be previously trained with known (and so labeled) images, while our system does not have any *a priori* knowledge neither for the segmentation step, nor for the recognition one.

### 3.3 Image segmentation: a bottom-up approach

In this section we present a bottom-up methodology for image segmentation. We work under an unstructured and completely unknown environment hypothesis. We also suppose to have still RGB digital images and not to know anything about the objects we are looking for. Our aim is to identify the "interesting" elements within the image. It is obviously necessary to define the exact details of what we consider "interesting". In order to do so, let's consider our work scenario. We do not possess any information about the objects that will be found, neither about their structure. We only have to select the "possible" objects within the environment, not considering what they represent. Obviously, under this hypothesis, the algorithm presented can only look for coherent pixel sets which are grouped together, relying on specific low-level features: light, color and position.

After a noise removing phase, by considering light information, the algorithm derives the starting blocks, which will be refined working on chromatic and positional elements. These first steps define a probability map which assigns to each single pixel of the starting image an "object-probability". By using a  $s - t$  graph cut approach, we will determine the final clusters.

The block diagram in figure 3.1 summarizes the image segmentation algorithm.

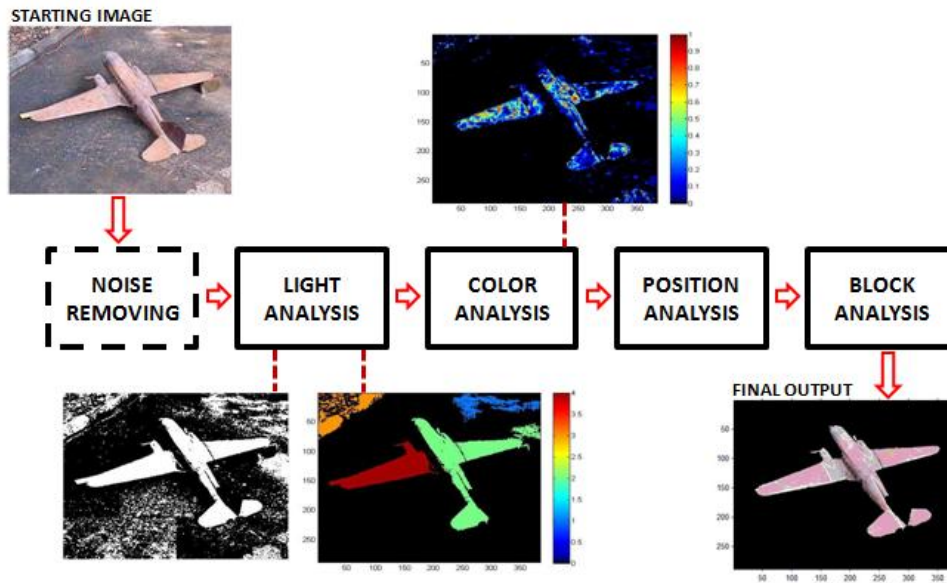


Figure 3.1: A block diagram to summarize the main steps of the segmentation algorithm.

In the following we briefly describe each step, presenting also examples with respect to figure 3.2. At the end of the chapter we will discuss the open problems.

### 3.3.1 Noise removing

We work with images taken from different sources (see section 1.5). Some images need a filtering step before processing. We suppose the presence of a white Gaussian noise and we implement the technique proposed by Vijaykumar et al. (see (292)) to remove it. The authors presented a fast and efficient algorithm. First, the amount of noise corruption in the noisy image is estimated. Then the center pixel of a given window is replaced by the mean value of some of the surrounding pixels based on a threshold value. We adopt this procedure not only because its computational complexity is very low, but above all because it does not affect edge consistency (a usual drawback of noise filtering). In the following we briefly give some details of the method. We address the interested reader to (292) for further details.

Noise having Gaussian-like distribution is very often encountered in acquired data. In such scenario, to each pixel of the original image is added a value from a



Figure 3.2: An image of a toy airplane taken from the RGB-JPEG database.

zero-mean Gaussian distribution. This condition allows to remove that noise simply by locally averaging pixel values (see (143)). Given a  $N \times M$  image  $I$ , for every pixel  $P_{i,j}$  of  $I$  (with  $0 \leq i \leq N$  and  $0 \leq j \leq M$ ), adding a white Gaussian noise we obtain a new pixel  $R$ :

$$R_{i,j} = P_{i,j} + G_{i,j}$$

where  $G_{i,j}$  is a noise value drawn from a zero-mean Gaussian distribution. First of all, by using the Immerkaer's fast method (see (108)), the noise standard deviation of the image is estimated. The absolute difference between the centered pixel  $C$  and the surrounding pixels in the filtering window is obtained by subtracting  $C$  to each element in the filtering window. This difference is compared with a threshold, calculated as the product of a smoothing factor and noise standard deviation. If the smoothing value is high, then the noise removal is better at the cost of loss of image details. We are mainly interested in preserving details and so, knowing that our acquired images are not affected by a high noise, we set the smoothing factor to 1.5 (instead of 2, the value proposed by the authors). At this point, if the absolute difference above introduced is within the threshold, the corresponding pixel values are taken for further processing. The number of pixels taken into account in a filtering window should be at least 5 (as the authors suggested). If it fails to satisfy the above condition, the window size is increased and the above mentioned procedure is repeated until the number of pixels under consideration in a filtering window is at least 5. Then the center pixel is replaced by the mean of those pixels that are considered. As the authors explained, the procedure can be summarized as follows. Let  $I$  be a  $N \times M$  image and  $P_{i,j}$  a pixel of  $I$ ; let  $S_{i,j}$  be the filtering window of size  $(2L + 1) \times (2L + 1)$  centered at

$P_{i,j}$ . The elements of this window are  $S_{i,j} = \{P_{i-u,j-v}, -L \leq u, v \leq L\}$  (where  $L$  depends on the windows size):

1. the noisy image is taken as  $P$ ;
2. the noise standard deviation is found out using Immerkar's fast method;
3. a 2D filtering window  $S_{i,j}$  of size  $3 \times 3$  is selected from the noisy image and let its center pixel be  $P_{i,j}$ . In the window, the center pixel is subtracted from each element and the absolute value of difference is calculated as  $AD = |S_{i,j} - P_{i,j}|$ ;
4. if the absolute difference  $AD < SF \times SD$  (where  $SF$  is the smoothing factor and  $SD$  the standard deviation), store the corresponding pixels in a one-dimensional array as  $DA(x)$ ;
5. if the number of elements in the  $DA(x)$  is at least  $2W - 1$  (where  $W$  is chosen to be 3 for a  $3 \times 3$  window) then the mean of  $DA(x)$  is calculated and at the center pixel  $P_{i,j}$  of the window is replaced with it;
6. otherwise the window size is increased and the same process is repeated;
7. steps 3 – 6 are repeated until the processing is completed for the entire image.



Figure 3.3: The starting image after the local thresholding.



### 3.3.2 Lightening analysis - finding the starting clusters

We have to individuate some starting clusters of pixels to be refined in the following steps, and we only want to rely on light, color and positional information. The first partition could be not very accurate. Indeed it has simply to be a starting point. So we start with an intensity analysis. The intensity version of an RGB image, in fact, contains few information (one third of the whole amount) and so its analysis is computationally very simple. Specifically we want to group together sets of adjacent pixels having similar lightening characteristics. First of all, we convert the RGB image into its corresponding intensity image. For each pixel  $P_{i,j}$ , whose RGB values are  $R_{i,j}, G_{i,j}, B_{i,j}$ , the intensity (gray scale) value is:

$$Y_{i,j} = 0,299 * R_{i,j} + 0,587 * G_{i,j} + 0,114 * B_{i,j}$$

Once the intensity image is to hand, we can define a threshold  $th$  and group together on one side all the pixels having intensity values over  $th$  (setting them to 1) and on the other side the remaining ones (setting them to 0). This is exactly what happens when we convert an intensity image to its corresponding BW version. The *naive* approach to define  $th$  is to consider the mean intensity value of the whole image. Obviously, if the image presents very dark or very light areas (as it usually happens in real scenarios), by using a global value we waste a lot of important details. To overcome this disadvantage we perform a locally normalized analysis: we divide the image in equal-sized blocks and we calculate a threshold for each of them (still considering the local mean value). Figure 3.3 shows the result of this process performed on our image.

Now we have all the pixels set to 1 or 0. Then we consider all the blocks of adjacent identical pixels, whose size is over a specific threshold, and we collect them in an array. We set the threshold to 0.3% of the whole image (i.e. working with  $600 \times 800$  images we will discard blocks having size less than 1,600 pixels). To individuate the blocks we use simple morphological operations, considering an 8-connected neighborhood. We label each pixel of the image belonging to a block with the same (progressive) integer, setting to 0 all the pixels which do not belong to any group (background). In figure 3.4 it is possible to observe the previously described procedure applied to the image of figure 3.2. As we can see, there are some blocks containing sparse sets of similar pixels which do not represent any meaningful object. Moreover the main object of the scene (the toy airplane) is subdivided into two different blocks. It is a very common scenario: in fact this first segmentation only based on intensity values is used to obtain what we call the starting clusters, that need a refinement to become the interesting objects of the scene.

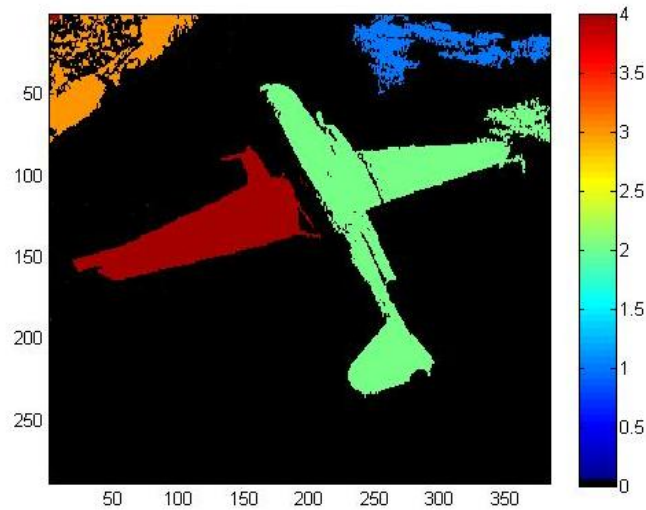


Figure 3.4: The image with the starting clusters (in different colors). The background is represented in black.

### 3.3.3 Chromatic analysis - finding the probability map

In a bottom-up approach we must use only the physical characteristics of the objects to segment the image. We have already performed a first coarse clusterization based on intensity values, grouping together pixels of the image only relying on light intensity. Now we have to combine those results with information taken from color. In fact, it is possible that, as shown, after this first analysis, some sets of pixels are over- or under-segmented. The aim of the present step is to group together adjacent pixels with similar chromatic characteristics.

We obviously have to start from the blocks identified previously. Due to the low accuracy, it is plausible that blocks contain different chromatic areas. Our idea is to individuate a set of adjacent pixels and to perform a probability function estimation to represent its color-range. This function will help us to discover in the image other pixels with the same chromatic characteristics possibly belonging to different starting clusters. Obviously considering the whole block of a starting cluster does not help us. Conversely we select a subset of the block to perform the analysis. How can we choose a meaningful subset? Is there a right size? If we could rely on some higher-level information, the decision would be easier, but in a bottom-up approach this is not available. Therefore it will be necessary to decide on the basis of other considerations. First, it is fair to say that, as in the analysis performed by the human visual system, color information must have a great importance. The choice of the



size of the subset is closely related to the weight that the color analysis will have compared to the first clustering process: the smaller the size of the subset, the greater the influence of this second step. As a drawback, if we considered sets that are too small, the outputs could be meaningless. Experimental results suggest to chose a dimension between 30% and 40% of the starting cluster. Once the size is to hand, we have also to decide how to select those points. As said previously, the starting clusters contain areas with similar intensity characteristics, but we have no assurance of chromatic uniformity. Obviously the chromatic differences inside a single starting cluster can not be very high. In the natural objects and artifacts is also easy to observe some color consistency: given a point, its surroundings has generally similar colors or, at most, different shades of the same hue. Sudden variations are unusual and even when this occurs (e.g. in the border zone between two areas with different color), does not occur in all directions. Under these assumptions, wanting to maintain a bottom-up approach, without any additional information, we randomly chose a point of the block and we selected a square-shaped subset centered on it (see figure ref fig: *air<sub>r</sub>ectangle*).

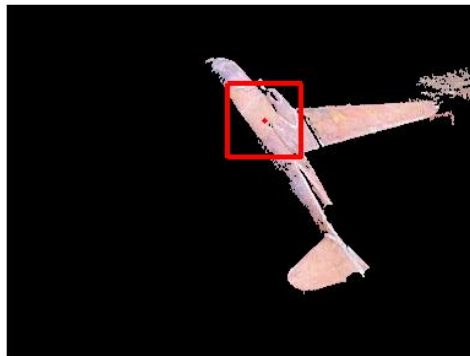


Figure 3.5: The image shows one starting block, the random point (red point) and the squared-shaped subset used for chromatic analysis (red rectangle).

To estimate its own chromatic characteristics, we need a probability function able to describe the color distribution in a meaningful and compact way. We use a mixture of Gaussians. A mixture of Gaussians can be seen as a weighted sum of  $M$  Gaussians, as in the following:

$$p(x) = \sum_{m=1}^M \alpha_m G(x; \mu_m, \Sigma_m)$$

where

$$G(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

and where  $\alpha_1, \dots, \alpha_M$  are the mixing coefficients, and  $\mu_1, \dots, \mu_M$  and  $\Sigma_1, \dots, \Sigma_M$  the means and covariances of each component.

In this work we will widely use mixtures of Gaussians and in chapter 6 we will describe in details the reasons of this choice. Now we simply want to point out that we do not know anything in advance about the characteristics of the function and, as Parzen explained (see (221)), a mixture of Gaussians can describe, with any degree of approximation, every other probability density function. Moreover there exists an efficient and very reliable iterative procedure to define a mixture of Gaussians starting from the set of values: the Expectation-Maximization algorithm.

### The Expectation-Maximization approach

A mixture of Gaussians (see figure 4.18) can be obtained as weighted sum of  $M$  Gaussians. We want to model a set of samples (considered as an i.i.d.) by using this pdf. A mixture of Gaussian is a function in the following form:

$$p(x) = \sum_{k=1}^M \alpha_k G(x; \mu_k, \Sigma_k)$$

where

$$G(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

The parameters in a Gaussian mixture (the mixing coefficients  $\alpha_1, \dots, \alpha_M$ , the mean and covariance of each component  $\mu_1, \dots, \mu_M$  and  $\Sigma_1, \dots, \Sigma_M$ ) can be estimated very efficiently from sets of samples with the Expectation-Maximization (EM) algorithm (26), given that the number of components  $M$  is known. Starting from initial values of these parameters, the EM algorithm proceeds by executing the following steps until convergence,

$$p_k(i) = \frac{\alpha_k G(x_i; \mu_k, \Sigma_k)}{\sum_{j=1}^M \alpha_j G(x_i; \mu_j, \Sigma_j)}$$

for  $i = 1, \dots, N, k = 1, \dots, M$

$$\alpha_k = \frac{1}{N} \sum_{i=1}^N p_k(i)$$

for  $k = 1, \dots, M$

$$\mu_k = \frac{\sum_{i=1}^N x_i p_k(i)}{\sum_{i=1}^N p_k(i)}$$

for  $k = 1, \dots, M$

$$\Sigma_k = \frac{\sum_{i=1}^N (x_i - \mu_k)(x_i - \mu_k)^T p_k(i)}{\sum_{i=1}^N p_k(i)}$$

for  $k = 1, \dots, M$

The EM algorithm guarantees to converge within finite steps to a local maximum of the log-likelihood function of the parameters, given the set of samples. More details of the EM estimation for Gaussian mixtures can be found in (26).

As previously said EM is an iterative procedure that, given the number of components  $M$  and starting from a set of values, proceeds by executing simple steps until convergence and determines the parameters  $(\alpha_i, \mu_i, \Sigma_i)$ , for the best-fitting  $M$ -components mixture. The knowledge about the number of components  $M$  is necessary. Many techniques have been proposed in the literature to face this problem. We use the estimated number of modes obtained by the Mean Shift algorithm. Mean Shift is a very simple non-parametric iterative procedure that shifts each data point to the average of data points in its neighborhood for seeking the mode of a density function represented by a set of samples. In chapter 6 we will analyze in details the use of modes as input parameters of the EM, and the use of Mean Shift to estimate them.

We want to summarize the procedure:

- choose randomly a point  $P$  inside the block;
- define a square-sized set of pixels (say  $S$ ) around  $P$  (whose size is between 30% and 40% of the whole block size);
- use Mean Shift to estimate the number of modes of  $S$  (say  $M$ );
- by using the EM (initialized with  $M$  as the number of components), estimate the parameters of the best-fitting mixture of Gaussians.

Once the probability density function is to hand, we can determine the pixels of the image having the same chromatic characteristics of  $S$ . After a normalization step, we obtain a mapping from each pixel of the starting image to a probability value in the range  $[0, 1]$ . In figure 3.6 we can observe the probability map obtained starting

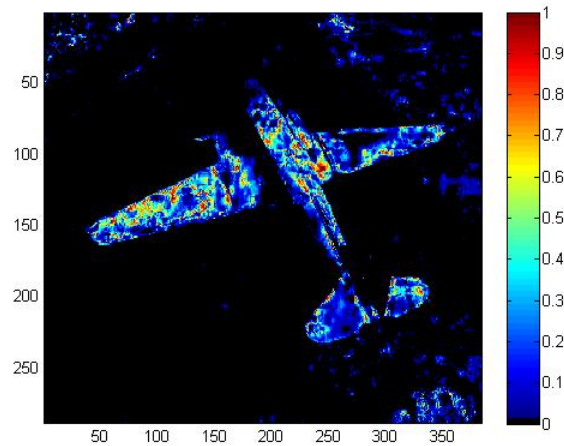


Figure 3.6: The (normalized) probability map obtained evaluating the whole image with the mixture obtained from the squared-shaped set defined in figure 3.5.

from the squared-shaped set showed in figure 3.5.

The previous step has already provided some information about the structure of the image. In particular, it has grouped together adjacent points with similar lighting conditions. We do not want to miss this information and thus it is necessary to define some procedures which allow us to increase the probability of the pixels belonging to the same block with respect to the others belonging to a different one. To this end, we multiply the probability of all points belonging to the same starting block by a factor  $k_l$  equal to 1.2. This last operation could result in probability values out of the range  $[0, 1]$ , so we have to perform a further normalization step. In figure 3.7 we present the result of this last step. As we can see, the probability of the points on the left wing (which was not part of the starting block under analysis) are proportionally reduced with respect to the points belonging to the airplane's starting set.

### 3.3.4 Positional analysis - refining the probability map

At this point, we have an analysis based on light and color values. Obviously, it is necessary to refine the probability map by adding some information about the relative position of the pixels within the image. Proximity is one of the criteria for identifying objects on which usually systems rely more. Besides the fact that proximity was one of the first laws of Gestalt submitted, everyday experience teaches us that we tend to group together parts of the image having similar color characteristics, only if they are adjacent to each other. We have a probability map over the whole image, which

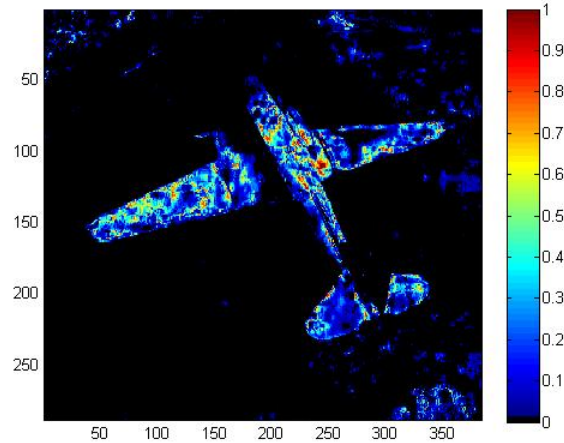


Figure 3.7: Chromatic probability map with block enhancing.

returns the reliability of each pixel to have similar chromatic characteristics to the subset of the starting cluster. Now we want to analyze also their relative position: the closer the pixels, the greater the probability to group them together.

We perform two different approaches to consider the relative position of each pixel. In the first case, without any regard to the values already obtained in the previous steps and without any additional consideration, we simply decrease the probability of a point in function only of the distance from  $E$  (the random point selected at the beginning of the chromatic analysis). By using an isotropic filter, we did not take into account the relative direction between  $E$  and the point under analysis. To obtain this result we define a Gaussian isotropic filter with mean set to  $E$  and variance defined with respect to the position of  $E$  in the picture. The probability map is then multiplied by the so defined filter. Figure 3.8 shows an example of this kind of filter (already centered in  $E$ ), while figure 3.9 shows the result of the product between it and the image under analysis.

The previous approach is very efficient and allows us to discard some noise in the peripheral area of the picture, but, not taking into account any information about the shape of the possible object, it is not completely satisfactory. In fact independently from the direction, we obtain a value influenced only by the distance, and by the distance evaluated from a reference point  $E$ , that we know has been chosen randomly inside the patch. To obtain better results, we start from the idea that, to calculate the decreasing of the value of a generic point  $x'$ , we have to consider two aspects: the relative position of  $x'$  with respect to the center of the probable object and the spatial

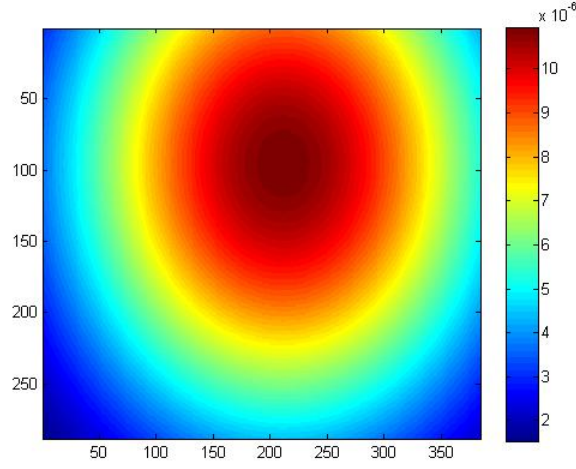


Figure 3.8: The Gaussian filter for the image under analysis.

distribution of the object main points  $\{x''_i\}_{i=1..k}$  (that is the shape). So, if we consider the probability of  $x''$  given the characteristics of the object, we can define:

$$P(x' | \theta) = f(x | \mu, \Sigma)$$

where  $\mu = \frac{1}{k} \sum_{i=1..k} x''_i$  is the center of mass of the sampled points cloud and  $\Sigma$  is the sample variance of the set. Under this assumption, the Mahalanobis distance ( $M_d$ ) can help us to evaluate the probability of  $x'$ . In fact we recall that

$$M_d(x') = (x' - \mu)^T \Sigma^{-1} (x' - \mu)$$

Obviously this second approach has the advantage to take into account the spatial disposition of the object points when we decrease the probability of a pixel with respect to the distance. But, firstly, we have to define the value of  $\Sigma$ . We use a Bayesian approach. Given a digital image, we indicate with  $X$ , the area of  $I$  representing the object. We considered 4 factors which can influence the value of  $\Sigma$ :

1. the density of  $X$ ;
2. the position of  $X$  with respect to the whole image;
3. the distance of the point of  $X$  from  $\mu$ ;
4. the distance of the centers of the other clusters from  $\mu$ .

Generally, as we can see from the figures, in this phase most of the "noise" generated from the chromatic analysis previously performed (as a consequence of the

presence of distant points with similar color), is filtered out.

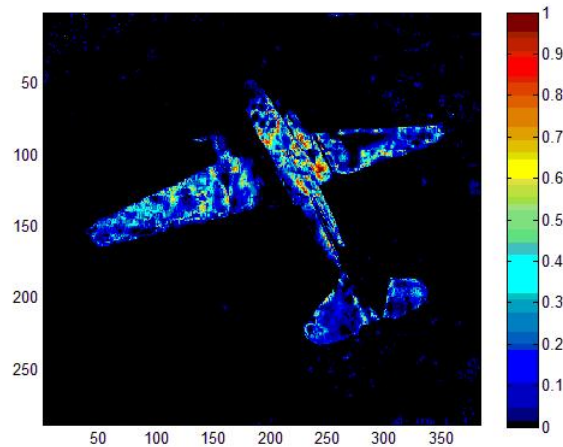


Figure 3.9: The probability map after the Gaussian distance filtering.

### 3.3.5 Neighborhood analysis - obtaining the final cluster

At this point we have a probability map that takes into account the information about light, color and position, but we do not yet have the final block. With a very naive approach, we could define a certain threshold on the probability values and consider "interesting" only those pixels that have a value above that threshold. This would result in numerous problems. First, we should think of some ways to define the threshold value (and this alone would be enough to make it difficult to generalize the method in a completely unsupervised scenario). Furthermore, as can be seen also in figure 3.9, the final image would contain false positives and false negatives. In fact, pixels not belonging to the object (false positives), which appear to be not far from the block under analysis because of their location and color characteristics, would be included as well. At the same time some parts of the object may have color characteristics greatly altered by lighting conditions, and therefore may be discarded. In the scene presented, for example, there is an area of pixels with very low probability that separates the left wing and the fuselage of the plane, creating an unwanted gap. Thus, a function must be defined in some way to be able to take into account the probability of each pixel to belong to the image, but also the connectivity. When choosing a pixel, the system must also consider its "neighbors" and their values. Therefore some representation is needed that allows us to incorporate the idea of "closeness". The Markov Random Field framework can really help us to solve the problem (for further

details on MRF see section 2.2.3). As we stated earlier, solving a MRF is equivalent to solve a multiway cut problem on a specific graph (44). This result is extremely important: even if a general minimum cut problem is still NP-complete, there exist a lot of provably good approximations with near linear running time (72). Specifically we will use a  $s - t$  cut graph approach. By using graphs, we can represent the image pixels as nodes and we can use the edges to define the relationship of adjacency. Starting from the graph, we will try to define a min-cut that separates the pixels of the object from the background (42; 40).

In the following we will use the same notation and terminology introduced in section 2.2.3. We recall that we have an undirected graph  $G = \langle V, E \rangle$ , where the node set  $V$  contains two special nodes called terminal nodes:  $V = \{s, t\} \cup Q$ ; edges between nodes in  $Q$  are called  $n$ -links where  $n$  stands for "neighbor", edges connecting nodes of  $Q$  to terminals are called  $t$ -links. The cut will partition  $V$  into two disjoint sets  $S$  and  $T$ , such that the source  $s$  is in  $S$  and the sink  $t$  is in  $T$ . Nodes  $Q$  represent image pixels (with a one-to-one correspondence), while node  $s$  represents the object and node  $t$  represents the background. The nodes are interconnected by edges in  $E$ . Typically, neighboring pixels are interconnected in a regular grid-like fashion. It is important to note that a neighborhood system can be arbitrary and may include also diagonal or any other kind of  $n$ -links. At the beginning of the procedure, every node of  $Q$ , other than to  $n$ -links, is connected with  $s$  and  $t$  by two  $t$ -links. Now we have to determine the cut.

The minimum cut problem looks for a cut which has the minimum cost among all possible ones. A very interesting result for defining the minimum cut of a graph, is the max-flow/min-cut theorem by Ford and Fulkerson (see (100)) which states that, in a flow network, the maximum amount of flow passing from the source to the sink is equal to the minimum capacity which, when removed in a specific way from the network, causes the situation that no flow can pass from the source to the sink. In other words the two problems are equivalent and to determine the min-cut it is sufficient to calculate the max-flow of the graph. There are a lot of polynomial time algorithms for min-cut/max-flow, which can be divided into two groups: Goldberg-Tarjan style push-relabel methods (see (113)) and Ford-Fulkerson style "augmenting paths" (see (100)). Even if the formers perform better on general graphs, for the kind of graphs of our interest (two dimensional grids) the latter seem to work in a more meaningful way. In (42) the authors presented a fast augmenting path algorithm with observed linear running time, which presents very interesting performances. We briefly describe how we can formulate the object/background segmentation as an energy minimization problem by using graph cuts.

We have to group together the pixels of the image into two sets, represented by the source  $s$  and the sink  $t$ . Given some neighborhood system (represented by  $N$ )



of all (unordered) pairs  $\{p, q\}$  of elements in  $Q$ , we can define an assignment  $f = (f_1, f_2, \dots, f_{|Q|})$  that is a binary vector whose components  $f_q$  specify assignments to pixels  $q$  in  $Q$ . Each  $f_q$  can be either 1 (for object) or 0 (for background). Vector  $f$  defines a segmentation and so it defines a cut. We can associate an energy (i.e. a cost) to  $f$ . Let  $D_q(l)$  with  $l \in \{0, 1\}$  be a fixed penalty for assigning to pixel  $q$  a specific value (knowing its starting probability). As a consequence of the way we have defined the probability values of  $q$ , it is clear that  $I(q)$  represents our degree of trust that pixel  $q$  belongs to the object under analysis. Thus we can define  $D_q(l)$  as in the following:

$$D_q(l) = \begin{cases} I(q) & \text{if } l = 1 \\ 1 - I(q) & \text{if } l = 0 \end{cases}$$

and, more simply,  $D_q(l) = |I(q) - l|$ . To encode these constraints, we create two  $t$ -links for each pixel node  $q$ , one from  $q$  to  $s$  with weight  $D_q(1)$  and the other from  $q$  to  $t$  with weight  $D_q(0)$ . Figure 3.10 shows a schematic representation of the graph (starting from the image under analysis). Now it is important to introduce some constraints able to represent the coherence between neighboring pixels. Specifically, we use a 8-neighborhood system as in figure 3.11.

So we create  $n$ -links between nodes with that configuration. The weight of these links is set to a smoothing parameter  $\nu \geq 0$ , which discourages the separation of adjacent pixels. Now the cost of an arbitrary cut  $C$  includes weights of two types of edges: severed  $t$ -links and severed  $n$ -links. It is important to note that a cut severs exactly one  $t$ -link per pixel: it will sever  $t$ -link  $(s, q)$  if the pixel will be in component  $T$  (it belongs to background), it severs  $t$ -link  $(q, t)$  otherwise. There will be severed also the  $n$ -links between adjacent pixels that are collocated in different sides. Therefore:

$$|C| = \sum_{q \in Q} D_q(f_q) + \sum_{(q,p) \in N, q \in S, p \in T} w(q,p)$$

The energy of the labeling  $f$  is correlated to the cost of  $C$ :

$$E(f) = |C| = \sum_{q \in Q} |I_q - f_q| + \nu \cdot \sum_{(q,p) \in N} V(f_q \neq f_p)$$

where  $V(f_q \neq f_p)$  returns 1 if its argument is true, 0 otherwise. With these assumptions, a minimum cut of  $G$  gives labeling  $f$  that minimizes energy  $E(f)$ . Note that parameter  $\nu$  controls the relative importance of the data constraints versus the regularizing constraints. Note that if it is very small, an optimal labeling assigns each pixel  $q$  a label  $f_q$  that minimizes its own data cost  $D_q(f_q)$ . In this case, each pixel chooses its own label independently from the other pixels. If  $\nu$  is big, then all pixels must choose one label that has a smaller average data cost. For intermediate values

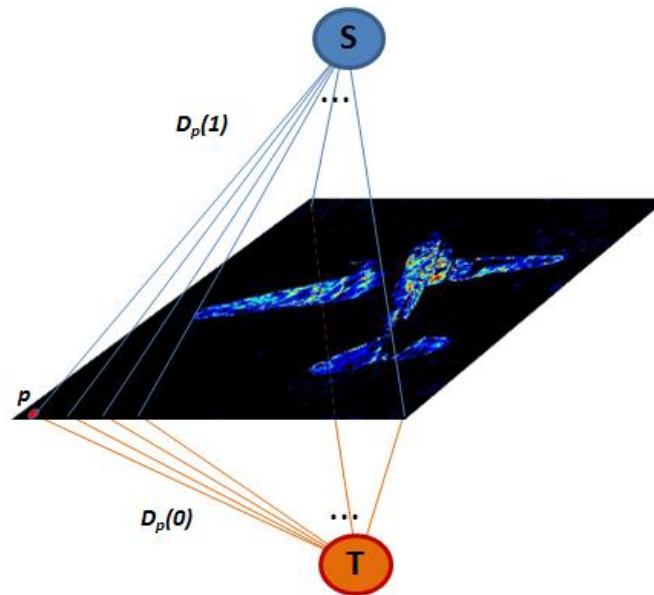


Figure 3.10: A schematic representation of the  $s - t$  graph obtained starting from the image under analysis. From each pixel of the image start two  $t$ -links: the first (in azure) toward the source  $s$  with weight  $D_q(1)$ , the second (in orange) toward the sink  $t$  with weight  $D_q(0)$ .

of  $\nu$ , an optimal labeling  $f$  should correspond to a balanced solution with compact spatially coherent clusters of pixels who generally like the same label. We set  $\nu = 2$ . Now we have to define a low-complexity implementation of this procedure.

Greig et al. in (123) constructed a two terminal graph such that the minimum cost cut of the graph gives a globally optimal binary labeling  $f$ . Before their work, exact minimization of energies like  $E(f)$  was not possible and such energies were approached mainly with iterative algorithms like simulated annealing. One of the most important contributions in (123) is showing that, in practice, simulated annealing reaches solutions very far from the global minimum even in simple binary cases. Even if, as we said before, in computer vision scenario path augmenting algorithms seem to return interesting results, they present a high complexity. Those methodologies, in fact, need to repeat continuously a breadth-first search for paths from  $s$  to  $t$ . In (42) the authors note that in the analysis of real images, building a breadth-first search tree typically involves scanning the majority of pixels and it could be a very expensive operation if it has to be performed too often. So they presented a novel approach which really improves the performances. The methodology they presented

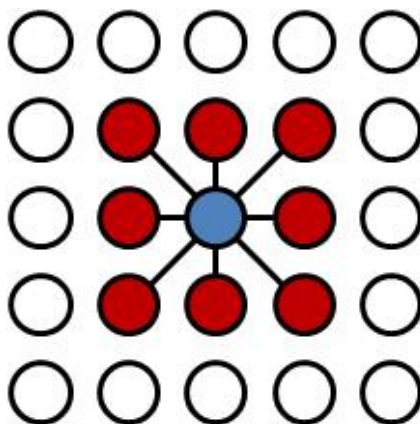


Figure 3.11: 8-neighborhood system.

maintain two non-overlapping search trees  $S$  and  $T$  with root at source  $s$  and at the sink  $t$  respectively. In the following we present the main steps of the process as the authors described in the original paper. For further details we address the interested reader to their work (42).

In the tree  $S$  all edges from each parent node to the children are non-saturated, while in tree  $T$  edges from children to their parents are non-saturated. The nodes that are not in  $S$  or  $T$  are called "free". The others can be either "active" or "passive". The active ones represent the outer border in each tree while the passive ones are internal. The point is that active nodes allow trees to "grow" by acquiring new children (along non-saturated edges) from a set of free nodes. The passive nodes can not grow as they are completely blocked by other nodes from the same tree. It is also important that active nodes may come in contact with the nodes from the other tree. An augmenting path is found as soon as an active node in one of the trees detects a neighboring node that belongs to the other tree. The algorithm iteratively repeats the following three stages:

- "growth" stage: search trees  $S$  and  $T$  grow until they touch giving a path from  $s$  to  $t$ ;
- "augmentation" stage: the found path is augmented, search tree(s) break into forest(s);
- "adoption" stage: trees  $S$  and  $T$  are restored.

At the growth stage the search trees expand. The active nodes explore adjacent non-saturated edges and acquire new children from a set of free nodes. The newly

acquired nodes become active members of the corresponding search trees. As soon as all neighbors of a given active node are explored the active node becomes passive. The growth stage terminates if an active node encounters a neighboring node that belongs to the opposite tree. The augmentation stage augments the path found at the growth stage. Since we push through the largest flow possible some edge(s) in the path become saturated. Thus, some of the nodes in the trees  $S$  and  $T$  may become "orphans", that is, the edges linking them to their parents are no longer valid (they are saturated). In fact, the augmentation phase may split the search trees  $S$  and  $T$  into forests. The source  $s$  and the sink  $t$  are still roots of two of the trees while orphans form roots of all other trees. The goal of the adoption stage is to restore single-tree structure of sets  $S$  and  $T$  with roots in the source and the sink. At this stage we try to find a new valid parent for each orphan. A new parent should belong to the same set,  $S$  or  $T$ , as the orphan. A parent should also be connected through a non-saturated edge. If there is no qualifying parent we remove the orphan from  $S$  or  $T$  and make it a free node. We also declare all its former children as orphans. The stage terminates when no orphans are left and, thus, the search tree structures of  $S$  and  $T$  are restored. Since some orphan nodes in  $S$  and  $T$  may become free, the adoption stage results in contraction of these sets. After the adoption stage is completed the algorithm returns to the growth stage. The algorithm terminates when the search trees  $S$  and  $T$  can not grow (no active nodes) and the trees are separated by saturated edges. This implies that a maximum flow is achieved. The corresponding minimum cut can be determined by  $C = \{S, T\}$ . In the tests of the algorithm with image segmentation (the case of our interest), the authors considered the use of some "seeds" introduced interactively by the user. Obviously this initialization step has been replaced in our case by the previously described probability map. It is important to note that even if the authors did not give a polynomial bound for their algorithm, they experimentally showed that in grid graphs for image analysis it performs better than the other known methods. Our experiments confirmed this finding in specific scenarios supporting our choice.

Indeed this approach helps us to solve the connectivity problem and to include, in the final cluster, also those parts of the image that, for chromatic and lighting characteristics, have not been included in the previous steps. Obviously we have no guarantees that the whole object is considered. Figure 3.12 shows the final result of the procedure on the selected image: as we can see, in the rear of the airplane the vertical stabilizer and the rudder are completely ignored.

### 3.3.6 Final analysis

Obviously the procedure has to be repeated at least once for each starting block. In fact, as said before, the first coarse clusterization could have grouped together pixels

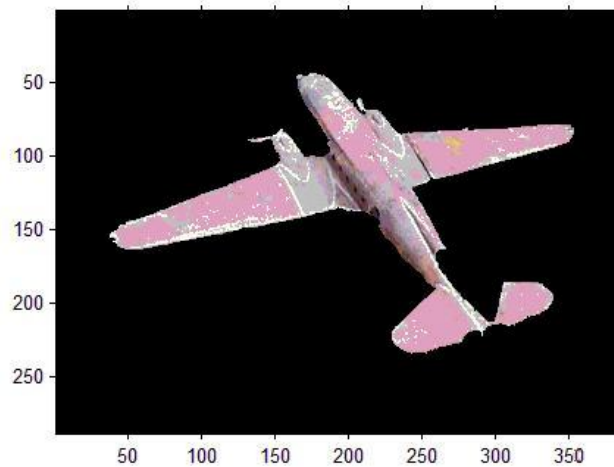


Figure 3.12: Our image after the last step: the toy airplane is isolated from the background but some parts are still missing.

with very different RGB values. Given a starting block, we have defined  $S$  (which is a strict subset of the block). After the first iteration of the procedure above described, we might have two different scenarios about the remaining pixels (the ones belonging to the starting block, but not to  $S$ ):

- if the pixels subset whose estimated probability value is below a certain threshold  $th_c$  has a size of at least 30% of the starting block, we repeat the procedure over this subset (i.e. we choose a random point and we define a square-sized subset whose size is between 30% and 40% of the block size);
- otherwise the remaining pixels are not considered and we analyze the next starting block.

We experimentally determined this last threshold to be 0.2.

As we can observe from the examples presented (more specifically from figure 3.4), it is possible that in the starting clusters (both at the beginning of the process and after some iterations), there are some subsets consisting of pixels that can not become a meaningful object (or part of it), independently from the complexity of next analysis. For example it can happen that adjacent pixels with similar intensity characteristics are grouped together in an extremely sparse (eventually connected) set. Consider, for example, the second and third blocks of figure 3.4. So, before any further processing, the system has to evaluate, relying on graph operators, the

consistency of the block and the possibility that it can lead to an "interesting" cluster. To perform this analysis, we simply determine the Laplacian matrix associated with the cluster. We recall that the element  $\{i, j\}$  of the Laplacian matrix  $M$  (also called admittance or Kirchoff matrix) associated to a graph  $G$  can be defined as:

$$M_{i,j} = \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$$

where  $\deg(v_i)$  represents the degree of node  $v_i$ . Afterwards we calculate the algebraic connectivity (see (31; 212)), which corresponds to the second smallest eigenvalue of  $M$ . This value gives information about the connectivity of the graph: if it is connected (with a measure of the connectivity) or not. It is interesting to note that at the beginning of the process, we can be sure that the starting clusters are connected, but, after the iterations of the procedure, we can not have any assurance.

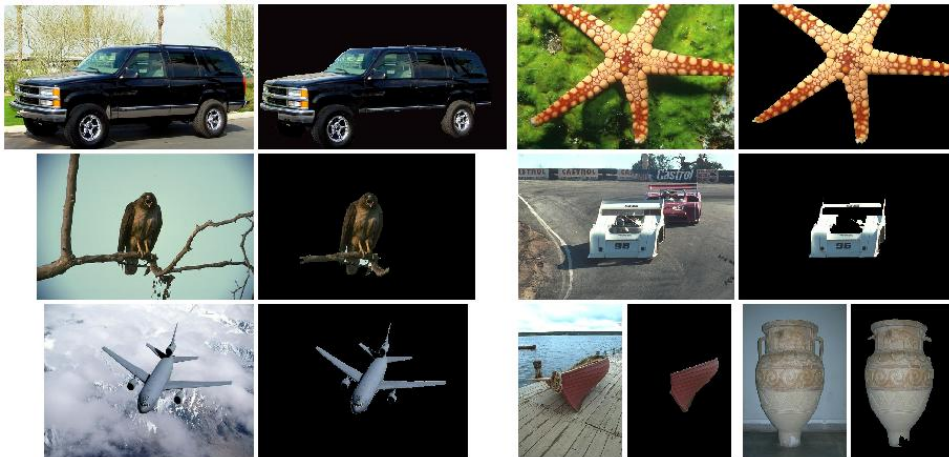


Figure 3.13: Some examples of our segmentation technique.

### 3.4 Experimental results

In this section we present the results from our experiments. Our method tries to individuate, with a bottom-up procedure, the main object(s) appearing in a digital picture. Once obtained this result (if it is possible), there is no further analysis and the remaining part of the image is simply considered background (even if there can be other items). We perform experiments with 5,000 different images taken from our databases. Figure 3.13 shows some meaningful examples. For many kinds of objects the methodology proposed can group together the interesting pixels of the

whole element or the main portion of it. It is worth noting that the segmentation step is performed without requiring any previous recognition. Our method obviously works worst with fairly articulated objects having subparts with different chromatic characteristics. In fact grouping is performed taking into account intensity, color and position. However for objects having a complex structure, even if it may not be able to isolate the whole element, the segmentation step can individuate one of the more semantically meaningful parts (or a connected subset of them). In such a scenario, it is not possible to perform better without any higher-level information.








	WEIGHT	HEIGHT	IMAGE PIXELS	OBJECT PIXELS	PIXELS GROUPED	FALSE NEGATIVES	FALSE POSITIVES	NEG/OBJ PIX	POS/OBJ PIX
	481	320	153.920	23.310	24.095	1.035	1.820	4,44%	7,81%
	1024	680	696.320	187.104	179.604	13.632	6.132	7,29%	3,28%
	300	455	136.500	38.818	38.896	672	750	1,73%	1,93%
	640	480	307.200	104.207	103.303	2.808	1.904	2,69%	1,83%
	320	214	68.480	32.840	35.516	1.608	4.284	4,90%	13,05%
	380	274	104.120	37.635	30.699	10.457	3.521	27,79%	9,36%
	525	679	356.475	130.025	129.329	4.896	4.200	3,77%	3,23%

Figure 3.14: A table containing some significant values about the images segmented (from left to right): weight of the picture, height of the picture, number of pixels of the picture, number of pixels of the object (calculated by hand), number of pixel returned by the segmentation procedure, false negatives (number of pixels of the object not recognized by the segmentation), false positives (number of pixels not belonging to the object but recognized by the segmentation), false negatives/object size ratio, false positives/object size ratio.

If we look at figure 3.13, we can observe different examples. When the object presents similar chromatic characteristics in the whole structure (as in the airplane



and the starfish images), even if the color is not uniform, if the object is complex and articulated or if it is partially occluded, we are able to individuate it and to clearly separate it from the background. It is interesting also to point out that while the starfish has colors that strongly differ from the one of the background, in the picture with the airplane the difference is very mild, but the segmentation returns anyway the whole object. When the object consists of subparts with different chromatic and lighting characteristics, the segmentation may return only one of them (or all of them but belonging to different clusters), as in the picture of the boat: the red hull is individuated, but it is completely separated from the rest. Instead when the subparts present different appearance, but their dimensions are small if compared with the size of the whole object and/or such subparts are included in the boundary of one specific part of the object, the method is able to group them together; for example if we look at the picture of the truck, we can see that the rims, the windows, the lights and the indicators are considered also if they strongly differ (for intensity and color) from the main block. Conversely if we look attentively to the picture of the racing car, we can note that the passenger compartment is not included; firstly the subpart has a big size, but, above all, it is important to observe that the upper right part of the spoiler (which is adjacent to the passenger compartment) presents similar chromatic characteristics, defining a darker area which does not allow us to consider the compartment completely surrounded by the main block. The amphora picture presents a similar scenario: in the lower left part we can find a lighter group of pixels in the peripheral area adjacent to the boundary, which is not included in the cluster. Finally it is important to note that if in the picture there are other adjacent elements or the background near the object under analysis presents similar chromatic characteristics, we may obtain an under-segmentation with blocks containing elements from semantically separated objects; this is the case of the picture of the hawk, where the bird of prey is grouped together with the branch of the tree it is perched on.

Figure 3.14 presents a table containing values which can help us to evaluate the performances of the methodology presented. Given a picture, we perform the segmentation and, by a human inspection, we extract the number of pixels of the main object in the scene; then we calculate the number of pixels of the object discarded by the algorithm (false negatives) and the number of pixels of the scene not belonging to the object but grouped in the main cluster by the algorithm (false positives). Clearly, as stated in the first sections of this chapter, also the segmentation performed by human beings could return different results (especially about the grouping of different subparts into a single cluster). Knowing this limitation, we considered the characteristics of the procedure also in the extraction by hand. Specifically, if we have a picture containing a person wearing clothes of very different colors, we do not group them together. The graph in figure 3.15 shows how the ratio between the size of the main object and the dimensions of the image influences the performances of this methodology (in particular the presence of false negatives). As we can expect when



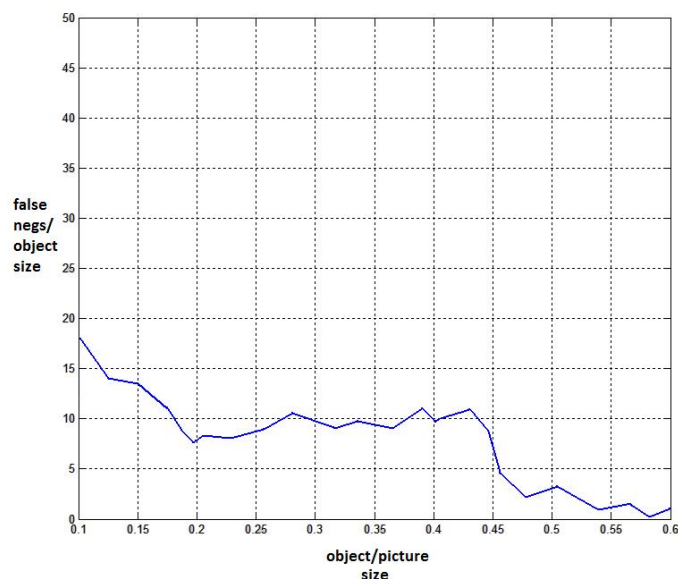


Figure 3.15: A graph showing how the object/picture size ratio influences the performance of the procedure.

the dimension of the object increases with respect to the dimension of the picture, we can observe fewer false negatives.

It is important to evaluate the performance of the algorithm with respect to the "complexity" of the object under analysis. In fact, given the details of the procedure, it is reasonable to expect a decrease in performance with increasing polychromatism and structural complexity of the object. Even looking at the examples shown above, there are clues in this direction (we consider in this regard images of the amphora and the race car). In the literature there is no specific performance index that will allow us to evaluate features of interest. We have therefore introduced a new parameter that could partly solve the problem. This value, called  $\psi$ , will have to carry multiple information. Specifically, it will grow significantly with the number of subparts of the object and their chromatic distance. The relevance of a subset will depend on two factors: the relationship between its size and the entire object (expressed in terms of pixels number) and the chromatic distance from the entire object. This value will be established with an average color value for each subpart and for the entire object. Let  $O$  be an object contained in a picture, whose size, expressed in number of pixels, is represented by  $H$ . Suppose that the object consists of  $M$  different components, each with size  $s_i$  with  $i = 1, 2, \dots, M$ .  $\psi$  has to be higher when the number of components grows, when the chromatic variation within each single component is

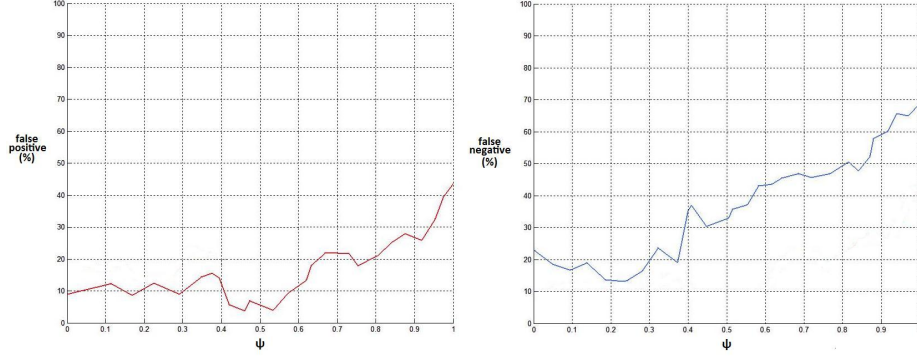


Figure 3.16: A graph showing how the performances of the procedure are influenced by  $\psi$ .

high and, moreover, when the chromatic characteristics of the components differ each other. We indicate with  $r_i$  the normalized RGB triplet (with values in  $[0, 1]^3$ ) of pixel  $i$ , with  $c_{k,k}$  the chromatic variance of the element  $k$  and with  $c_{k,h}$  a value which represents the chromatic distance between component  $k$  and component  $h$ .  $c_{k,k}$  and  $c_{k,h}$  are defined as in the following:

$$c_{k,k} = \frac{1}{N_k} \sum_{i=1}^{N_k} (r_i^{(k)} - m^{(k)})^T (r_i^{(k)} - m^{(k)})$$

and

$$c_{k,h} = \left\| m^{(k)} - m^{(h)} \right\|$$

where  $m^{(k)} = \sum_{i=1}^{N_k} r_i^{(k)}$ . Now we can define  $\chi$ :

$$\psi = \frac{1}{H^2} \sum_{i=1}^M \sum_{j=1}^M c_{i,j} s_i s_j = \frac{1}{H^2} \left( \sum_{i=1}^M c_{i,i} s_i^2 + 2 \sum_{i=1}^M \sum_{j=i+1}^M c_{i,j} s_i s_j \right)$$

As we can observe in figure 3.16, the performance of the our segmentation procedure decreases as  $\psi$  increases.

### 3.5 Open problems

By using the procedure previously presented, starting from an unseen picture, with the procedure previously described, it is possible to isolate one or more "probable"

objects. In the approach there still are some open problems concerning both the effectiveness and the efficiency.

About this latter aspect, it is worth noting that, in the previous description, we did not completely consider the run-time complexity. In real scenarios decision time is a very critical point. Unfortunately, even if our methodology can mostly return the main object of a scene very quickly, considering that image segmentation is only the first step of a more complex image analysis, in the worst case the time complexity is not feasible for real applications. However it is possible to define some approximations; the fact that the procedure works directly with every single pixel of the image causes a high cost. So we can obtain a great reduction, grouping together adjacent pixels before the first step of the algorithm and working with these sets. A similar choice is performed, with very interesting results, in (3): the image is pre-processed subdividing it into a grid of regular  $N \times N$  pixels cells. Obviously with this resolution reduction, the number of elements heavily decreases (of a factor equal to  $N^2$ ). Increasing  $N$ , with the time complexity reduction, also the performances of the algorithm became worst. It can be useful to perform a comparative study to determine the optimal  $N$  value. In the same work, we also find another interesting "trick" to reduce complexity: instead of working with the complete RGB triplets, the authors convert each image in  $I_1 I_2 I_3$ . This specific space can help because the three components are statistically-uncorrelated color features, independent of intensity changes. (206) works only on  $I_2$  and  $I_3$  for segmentation and classification, while the authors of (3) present interesting results working only with the  $I_2$  component. We are performing experiments with other color spaces to understand how this choice can influence our results.

From a completely different point of view, we want to evaluate whether the procedure could be more effective. Indeed, for complexity considerations, in the analysis of the final clusters, we performed a  $s - t$  graph cut, isolating each single possible object at time. Then, individuated the block, the belonging pixels were discarded and the analysis started again on the remaining ones. If we could be able to perform a multi-label analysis, the obtained results really would be improved. Once estimated, for each single cluster, the probability map over the whole image, by using MRF we could define a function able to maximize the sum of the probability of each single pixel, maintaining, at the same time, the cluster connected. This implementation, obviously, needs more space and time resources. We are trying to perform a multi-label image analysis under certain assumptions which can reduce the complexity.

## Chapter 4

# IEA: Implicit and Explicit Analysis

### 4.1 Introduction

The procedure presented in the previous chapter returns connected subparts of the original image containing "interesting" features. The aim of the present work is to define a similarity measure between objects. We start observing that human beings use a lot of information and analyze very different aspects to achieve the goal of classifying objects: shape, position, color and so on. The main idea that inspired this work is to combine different methodologies (each working on a specific characteristic) to obtain a more meaningful idea of the distance between objects. In fact in the Implicit and Explicit Analysis (in the following also called IEA) we try to analyse two important aspects, paradigmatically connoted by the traits of the David and Leonardo's drawings shown in figure 1.7. Mainly inspired by the human conception of representation, we identify two main characteristics of object representation and we called them the implicit and explicit models. The term "explicit" is used to account for the main traits of what, in the human representation, connotes a principal source of information regarding a category, sort of visual synecdoche; the term "implicit", on the other hand, accounts for the object rendered by shadows and lights, colors and volumetric impression, sort of visual metonymy in which specific features account for the whole object.

In this chapter we will firstly show how, once isolated an object from the scene, to obtain the two representations; then we will define a specific distance measure for both of them, showing the results obtained by individually using each of these two approaches, without any interaction with the other. At the end we will introduce a

specific methodology to combine these measures to obtain a more general object similarity distance. As we will show in a more detailed way in next sections, we choose a logical matrix to describe the contour and a set of real tuples for the interior. We will use Procrustes analysis to compare logical shapes, and Kernel methods for the other characteristics. The chapter ends with our experimental results and with the analysis of the open problems of the methodology.

## 4.2 Methodology

The representation of every object is given as a pair  $\langle E, I \rangle$  where  $E$  is the "drawing", a necessarily connected (maybe meaningless) contour of an object or of its part (the so-called explicit representation), and  $I$  is the "painting", the area inside the boundary (the so-called implicit representation). Some examples are given in figure 4.1. We work with different kinds of images. Some of them are taken from the log-JPEG database (see section 1.5), which contains images which do not require any further processing, being already in the format described. Moreover we work also with objects isolated by using the segmentation technique presented in the previous chapter starting from generic digital images taken from the other databases.



Figure 4.1: Images from our log-JPEG database with the implicit (on the left) and the explicit (on the right) descriptions.

This first scenario is very simple: we have a pair  $\langle E, I \rangle$ , that describes the object

under analysis, and we want to assign it to the proper category. In this approach, in contrast with the rest of the work, we assume the existence of a repository. In these first experiments, the existence of the repository can give a great help. In fact we can evaluate the performances of the methodologies presented working on images containing known subjects. Once obtained an estimation of the effectiveness of the algorithms, we can eliminate the repository and compare directly two completely unknown objects extracted from the scene. Our knowledge base consists of a set of reference classes. Each class is represented by some objects described by an implicit-explicit model. The explicit denotation is a  $60 \times 60$  logical matrix; it will be compared with the extracted shape via generalized Procrustes analysis, as illustrated in section 4.3. The model of the implicit denotation, instead, is a learned mixture describing the chromatic characteristics of the set of tuples which represent the interior of the object; it will be compared with the implicit description of the current object by using kernel methods, as illustrated in section 4.4. It is clear from the figures that while the texture representation gives a very detailed description of the part analyzed, the shape can only give an approximation. Despite all this, the importance of shape in object classification is very significant, as we will explain thoroughly in the next sections. The experiments will show that even if each object can be analyzed and compared for similarities with prior shapes and parameters, suitably classified in the knowledge base, each of the two data sets by itself does not give enough information, while, once combined, they can lead to further selection of the correlated objects. Obviously we will need also to define precise methodology for combining the two distances in a meaningful way.

## 4.3 Explicit Analysis

### 4.3.1 Defining the explicit model

The explicit model for each object consists of a logical  $60 \times 60$  matrix: the pixels of value 1 are on a 0 values background. We extracted the contour by using the widely known Canny method (see (48)). The Canny algorithm uses an optimal edge detector based on a set of criteria which include finding most of the edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization, and marking edges only once when a single edge exists for minimal response. The methodology was first introduced by John Canny for his Masters thesis at MIT in 1983, and still outperforms many of the more recent algorithms that have been developed. The solution to the edge detection problem was presented as a rather complex exponential function, but Canny found several ways to approximate and optimize the edge-searching problem. His procedure consists of four main steps:

- the image is smoothed by two one-dimensional Gaussians, one in the  $x$  direction and the other in the  $y$  direction (2D Gaussian filter is separable);
- the gradient of the image is calculated (again working in the  $x$  and  $y$  directions separately);
- each non-maximal pixel is suppressed: in this step only the local maxima pixels in the direction of the gradient are preserved (with the consequent suppression of all others);
- each pixel is evaluated by using two thresholds (hysteresis): if a pixel has a value above the higher threshold, it is set as an edge pixel; otherwise if it has a value between the two thresholds there are two cases: if it is the neighbor of an edge pixel, it is set as an edge pixel as well, otherwise not. A pixel having a value below the lower threshold is never set as an edge pixel.



Figure 4.2: An image of an airplane and the contour obtained by using the Canny algorithm.

In figure 4.2 we can see an example of the Canny algorithm applied on one of our images: as we can see, the previous segmentation step really improves the performance of the edge detection. We opted for Canny for its performances but, above all, because it produces single pixel thick, continuous edges, very useful for our analysis. As we said previously, we need to resize the contours to obtain a  $60 \times 60$  matrix; considering the very low complexity of the figure, we simply adopt a nearest neighbor interpolation. The choice of the matrix size is of extreme importance. In one of the next steps we have to extract from the contour some critical points. If the matrix is too small, the risk is having *false negatives*, and so losing some important information. On the other hand, if it is too big, the risk is having *false positives*. We experimentally found that the optimal size is between 53 and 72. We chose arbitrarily a size of 60.

We now need to determine a way to compare two explicit models, i.e. two  $60 \times 60$  logical matrices representing two shapes. Many different methodologies exist in this direction. The next chapter contains a brief overview about the most used methodologies for shape description and comparison.

### 4.3.2 Shapes

The choice of a shape descriptor can heavily influence the performances of the operations on the shape itself. Before starting describing a shape, we have to give some definitions. But how can we define a shape? In our work we recall the results of D.G. Kendall (in (159)):

#### Definition 1: Shape

A shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.

According to this definition, shape - in other words - is something invariant to Euclidean similarity transformations. Now we have to ask how we can compare two shapes and under which conditions they can be considered similar. We start from the interesting idea that they can be considered *congruent* if they differ by a rigid body transformation. In fact a shape is what we obtain if we remove location, scaling and orientation from a 2D figure, while we define *form* (see figure 4.3) what we obtain if we remove only location and orientation (i.e. a form is a shape at different scales). We can express mathematically the previous definitions.

More specifically two figures  $\mathbf{A} : N \times K$  and  $\mathbf{B} : N \times K$  have the same shape, i.e. they belong to the same shape equivalence class  $[\mathbf{A}] = [\mathbf{B}]$ , if:

$$\mathbf{B} = \mathbf{A}\Gamma + \mathbf{1}_N\gamma^\top$$

while they have the same form if:

$$\mathbf{B} = \beta\mathbf{A}\Gamma + \mathbf{1}_N\gamma^\top$$

where  $\beta > 0$  is a scalar which makes the two figures of the same size,  $\Gamma : K \times K$  is a rotation ( $|\Gamma| = 1$ ),  $\mathbf{1}_N$  is a vector of ones and  $\gamma : K \times 1$  is a translation.

There is clearly a strong connection between shapes, their descriptors and the methodology to calculate a distance between them. Think about everyday life: how can we describe or define a shape? It is clear that we do not have a formal and analytic description, but usually, when we have to describe how an object appears, we use references to known shapes (e.g. a cloud seems to have the shape of a rabbit,



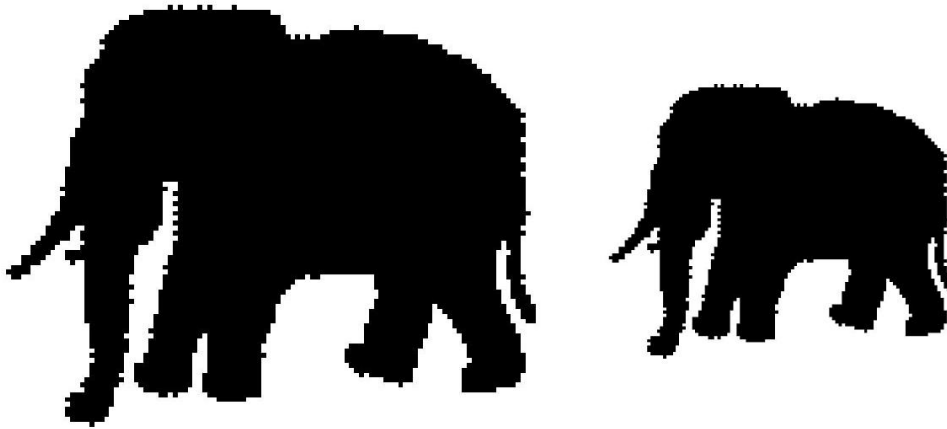


Figure 4.3: Two objects with the same form: an elephant.

see figure 4.4). Obviously such descriptions can not easily be used in an algorithmic framework. In the following sections we will describe a shape by locating a finite number of points on its outline. We refer to those points as landmarks (see (83)):

**Definition 2: Landmark**

A landmark is a point of correspondence on each object that matches between and within populations.

In (83) the authors discriminate landmarks into three subgroups:

- Anatomical landmarks: points assigned by an expert that corresponds between organisms in some biologically meaningful way.
- Mathematical landmarks: points located on an object according to some mathematical or geometrical property, i.e. high curvature or an extremum point.
- Pseudo-landmarks: constructed points on an object either on the outline or between landmarks.

A mathematical representation of an  $n$ -point shape in  $k$  dimensions could be obtained by concatenating each dimension into a  $kn$ -vector. The vector representation for planar shapes (i.e.  $k = 2$ ) would then be:

$$\mathbf{x} = [x_1, y_1; x_2, y_2; \dots; x_n, y_n]^T$$

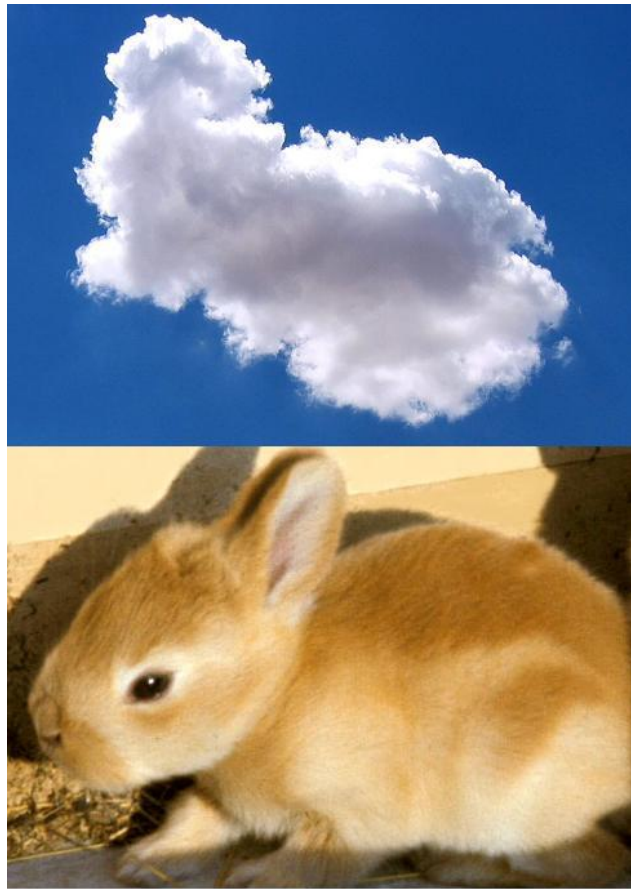


Figure 4.4: A rabbit-shaped cloud.

### 4.3.3 Landmarks

We stated that it is possible to represent a shape by using only a subset of the boundary points, extracting from an image contour all the relevant information and discarding the redundant ones. Now we need to decide which points of the contour can be discarded and which ones have instead to be considered important. We will call these latter "critical points". We hereby start making some assumptions and introducing some definitions, concerning the object image:

1. the image is formed by pixels in the set  $\{1, 0\}$  in a  $60 \times 60$  logical matrix, such that the contour of the represented shape is 1s on a background of 0s;
2. each point on the boundary is specified by a single pixel. The boundary is pixel-continuous (i.e. no discontinuity is allowed between two pixels) and closed.

In figure 4.5 it is shown a logical shape from our data set and its extracted boundary (obtained by using Canny algorithm).

**Definition 3: Configuration**

A configuration is a finite set of landmarks (i.e. the  $xy$  coordinates) on a specimen. The configuration matrix  $X$  for planar shapes is the  $k \times 2$  matrix of Cartesian coordinates of  $k$  landmarks in 2 dimensions. The coordinates of the points are calculated starting from the upper left corner of the image.

**Definition 4: Topologically Ordered Configuration**

A topologically ordered configuration (TOC) is a configuration in which the points in the rows of  $X$  respect a cyclic order (invariant for circular shifting of matrices rows) induced by any parameterization of the contour of the shape.

**Definition 5: Clockwise/Counterclockwise TOC**

A clockwise/counterclockwise topologically ordered configuration is a topologically ordered configuration in which the points are ordered clockwise/counterclockwise with respect to the contour of the shape.

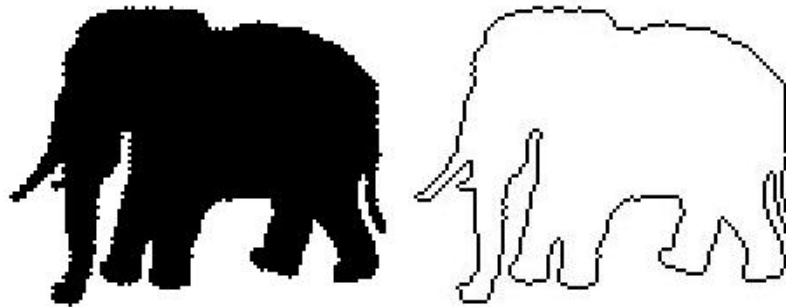


Figure 4.5: A logical shape (on the left) and its extracted boundary (on the right).

Given the landmarks  $\langle A, B, C, D \rangle$  in figure 4.6, for any parameterization of the contour, the points  $\langle A, B, C, D \rangle$  can appear in any configuration which is a circular shifting of this sequence (i.e.  $\langle B, C, D, A \rangle$ ,  $\langle C, D, A, B \rangle$ ,  $\langle D, A, B, C \rangle$ ), and these are all part of a topologically ordered configuration. Instead a sequence  $\langle A, C, B, D \rangle$  violates the topological order. Moreover the sequence  $\langle D, C, B, A \rangle$ , even if it does not violate the topological order, it is not a clockwise TOC, but counterclockwise. Note also that the conditions on the continuity and closure of the boundary hold.

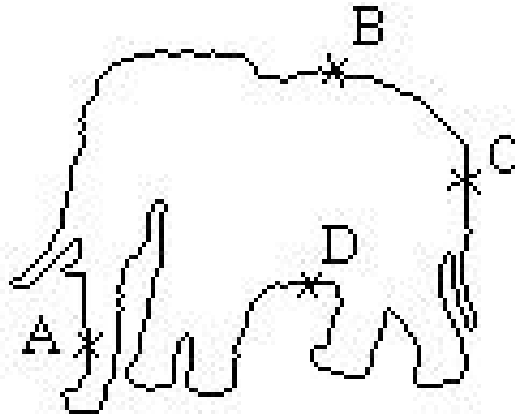


Figure 4.6: An elephant shape with 4 landmarks (A, B, C, D) highlighted.

But we still have to decide which points can be considered critical. We start from the idea that the variation points of a boundary bring the information needed to analyze a shape. If we think to the polygons (which can be considered the most simple and regular shapes) and we want to extract the minimum number of pixels to preserve all the information, the only choice is to maintain the vertexes. If we take fewer points it will be impossible to reconstruct the original figure and, at the same time, every point we take in addition is redundant. Our description is not already complete and we need to introduce a last element to define mathematically the critical points: the medial axis.

#### 4.3.4 Medial axis

Let  $\Omega$  be a domain in  $\mathbb{R}^2$ . Let  $B_r(p)$  denote the closed disk of radius  $r$  centered at  $p$ . (60) defines  $D(\Omega)$  as the ordered set of all closed disks contained in  $\Omega$ :

$$D(\Omega) = \{B_r(p) \mid B_r(p) \subset \Omega\}$$

which is ordered by the set inclusion. The *core* of a domain  $\Omega$  is consequently defined as the set of all maximal elements in  $D(\Omega)$ , i.e.:

$$CORE(\Omega) = \{B_r(p) \in D(\Omega) \mid B_s(q) \in D(\Omega) \text{ and}$$

$$B_r(p) \subset B_s(q) \implies B_r(p) = B_s(q)\}.$$

The *medial axis* of a domain  $\Omega$  is the set (i.e. the geometrical *locus*) of the centers of the disks in  $CORE(\Omega)$ , that is:

$$MA(\Omega) = \{p \in \Omega \mid B_r(p) \in CORE(\Omega)\}.$$

Figure 4.7 shows a shape taken from our repository and its medial axis. The description presented here allows us to obtain the so called Blum's medial axis (see (29)), from the name of the researcher who first introduced the idea. In the literature there exist more complex, accurate and meaningful structures used to determine the medial axis (briefly introduced in chapter 5 and also object of our current research).

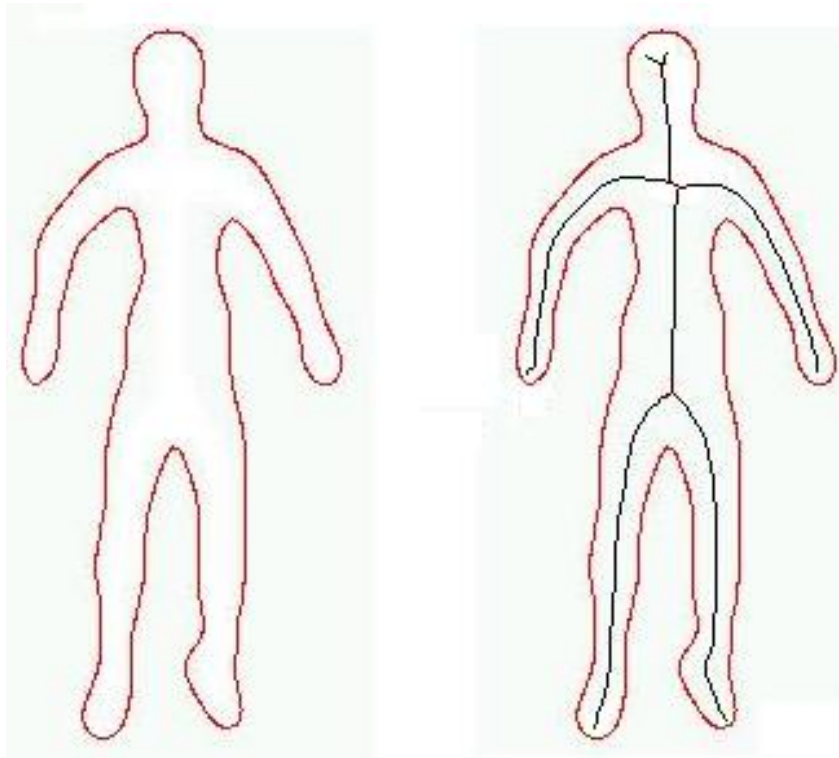


Figure 4.7: A shape from our repository and the corresponding Blum's medial axis.

### 4.3.5 Critical Points

Now we have a meaningful background to define our landmarks. The introduction of the medial axis, in fact, allows us to introduce an analytical procedure to determine the critical points. The vertices of a polygon, as well as other significant points of a shape, correspond, as mentioned, to the points where the boundary presents rapid changes in its direction. The medial axis is a skeleton around which the boundary

develops and follows the profile. If we imagine to unroll the skeleton of a shape, represent it as a straight line, and, therefore, to indicate the boundary of the shape from this new reference system, we will notice a series of profiles represented by line segments on which peaks and valleys our points of interest would be located. That is, if we defined a measure able to associate each point of the boundary to the value of the distance from the corresponding skeleton point, the points of interest would be easily associated to the minimum and maximum of that function. This allows us to introduce the following definition:

**Definition 6: Critical Point**

Given a shape boundary  $S$ , we indicate with  $C$  the subset of points in  $S$  over which there are sudden variations in the direction of  $S$ . The elements of  $C$  are called critical points.

Now we will give a mathematical formalization for the critical points:

Let  $C$  be a pixel-continuous and closed shape contour of  $n$  pixels; let  $L$  be a clockwise topologically ordered configuration of a set of landmarks for each point location of  $C$  ( $|L| = n$ ); let  $\Omega_C$  be the set of points in  $\mathbb{R}^2$  contained inside  $C$  and  $MA(\Omega_C)$  the medial axis of  $C$ ; let  $h : L \mapsto \mathbb{R}$  a function that, taken a landmark of  $L$ , returns the distance of the corresponding point in  $MA(\Omega_C)$ ; define

$$T = \left\{ l \in L : \frac{\partial h(l)}{\partial l} = 0 \right\}$$

then the points  $c \in C$  such that the corresponding landmark  $l$  is in  $T$  are the critical points of the shape.

In other words each point of the contour which represents a local maximum or a local minimum of the distance from the corresponding point in the medial axis can be considered a critical point. More simply, every point of the contour in which there is a sudden variation in the direction of the contour itself represents a critical point. According to the classification presented in section 4.3.2 our critical points are mathematical landmarks.

Now we have to locate the critical points. Our first approach tried to solve the problem analytically, calculating the medial axis of the figure and the derivative of the function which maps the points of the axis to their distance from the contour (i.e. following exactly the previous definition). But the results were not encouraging and we opted for a heuristic approach. It is extremely important in fact to highlight that, once resized and digitalized, our image is affected by pixelization and, as a consequence, there might be a lot of false positive. In real scenarios, the set of critical

points defined above is not the minimum one: there is a little redundancy. However this redundancy does not affect the results. Only if the number of false positives becomes too high, the following steps of the analysis fail. Our heuristic approach seems to partially solve this problem. We define a set of

*superimposed*

$n \times n$  windows (with  $n = \{3, 5, 7, 9\}$ ). On each contour pixel we center all these windows. For each window we estimate the angles described by the central and the two external points. If the difference between the four angles is less the 40 degree, we consider only the minimum angle (i.e. the angle with the lowest value), say  $\alpha$ . If  $\alpha < 144$  (a value found experimentally), then we have a critical point, otherwise there is none. To face the consequences of the pixelization, when we find a critical point we shift the windows of 9 contour pixels. See figure 4.8 for some examples of shapes with critical points.

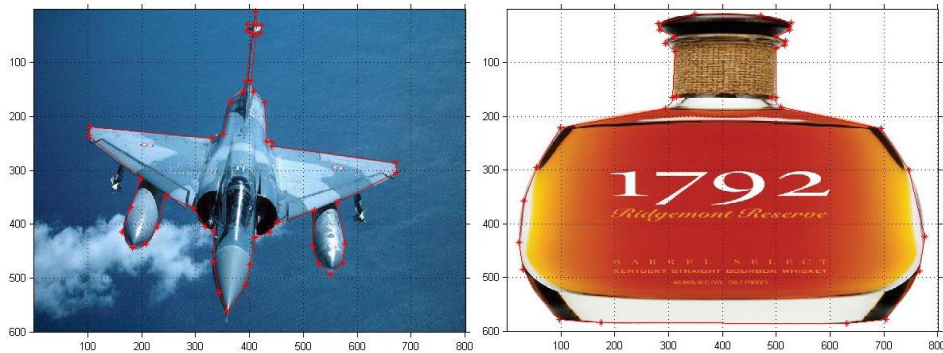


Figure 4.8: Contours of two different objects of our dataset with highlighted critical points (in red).

### 4.3.6 Procrustes Analysis

Once the critical points are to hand, we need to define some procedure to compare two shapes and to determine the distance between them, possibly taking the most advantage of the shape description adopted. Procrustes analysis seems to be an interesting answer: it is a powerful tool that, while determining the rigid body transformation (in case it exists) between two shapes, can return an evaluation of the distance between them. This distance is calculated as a measure of the complexity of the transformation itself. More specifically the Procrustes distance is a least-squares type shape-metric that requires two aligned shapes with one-to-one point correspondence. The alignment part involves four steps (see figure 4.9):

- compute the centroid of each shape;
- re-scale each shape to have equal size;
- align w.r.t. position the two shapes at their centroids;
- align w.r.t. orientation by rotation.

Procrustes analysis theory is a set of mathematical tools to directly estimate and perform simultaneous similarity transformations among the model point coordinates matrices up to their maximal agreement. It avoids the definition and solution of the classical normal equation systems. No prior information is requested for the geometrical relationship existing among the different model objects components. By this approach, the transformation parameters are computed in a direct and efficient way based on a selected set of corresponding point coordinates (see (15)). Akra in (78) describes the birth and the evolution of this methodology. The method was explained and named as Orthogonal Procrustes problem by Schoenemann in 1966 (see (246)), a scientist in the Quantitative Psychology area. In this publication, Schoenemann gave the direct least-squares solution of the problem, that is to transform a given matrix  $A$  into a given matrix  $B$  by an orthogonal transformation matrix  $T$  in such a way to minimize the sum of squares of the residual matrix  $E = AT - B$ . The first generalization to the Schoenemann orthogonal Procrustes problem was given by Schoenemann and Carroll in 1970 (see (247)), when a least squares method for fitting a given matrix  $A$  to another given matrix  $B$  under the choice of an unknown rotation, an unknown translation and an unknown scale factor, was presented. This method is often identified in statistics and psychometry as Extended Orthogonal Procrustes problem.

After Schoenemann, similar methods were proposed in computer vision and robotics area (see (8; 136)). The solution of the Generalized Orthogonal Procrustes problem to a set of more than two matrices was reported (119; 279). Further generalization in the stochastic model is called Weighted Procrustes Analysis, which can be different weighting across columns (179) or across rows (165) of a matrix configuration. An approach that can differently weight the homologous points coordinates was given in a important paper by Goodall (see (117)). A method that can take into account the stochastic properties of the coordinate axes was given by Beinat and Crosilla (70).

Geodetic Sciences were given by Crosilla and Beinat (70): photogrammetric block adjustment by independent models, and registration of laser scanner point clouds. The reader can also find a detailed survey of the Procrustes analysis and some of its possible applications in the Geodetic Sciences in (69).

Procrustes analysis starts pre-multiplying the shape  $\mathbf{A}$  by an  $(N-1) \times N$  Helmert matrix  $\mathbf{H}$ .  $\mathbf{H}$  has orthonormal rows, each orthogonal to the unit vector  $\mathbf{1}_N/\sqrt{N}$ . The rows of the matrix  $\mathbf{A}_H = \mathbf{H}\mathbf{A}$  are the coordinates of the derived landmarks. And



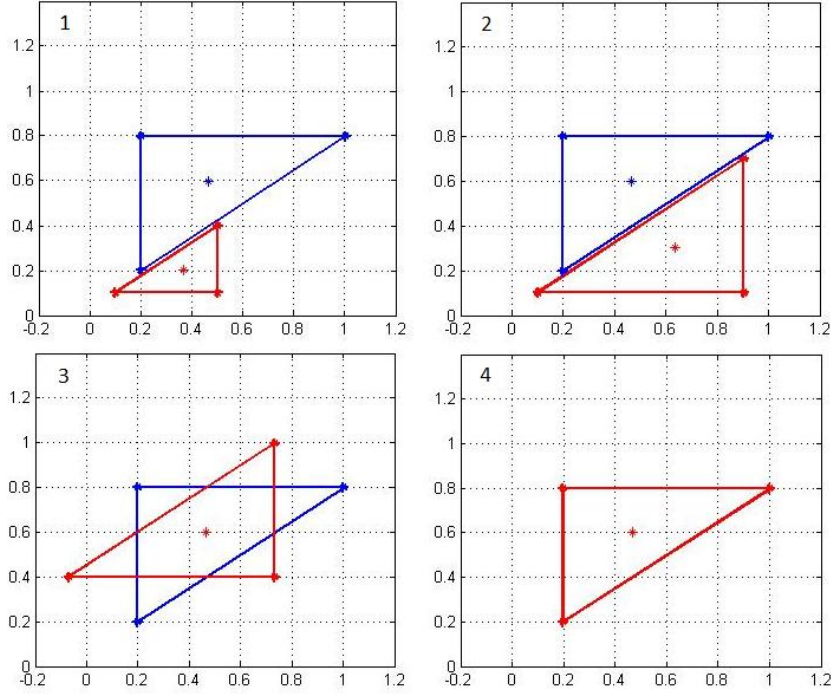


Figure 4.9: An example of Procrustes analysis on two triangles (1). Three different transformations are applied: scaling (2), translation (3) and rotation (4).

the centered landmarks, i.e. with the location removed, are obtained by the derived landmarks as

$$\mathbf{A}_C = \mathbf{H}^\top \mathbf{A}_H$$

The matrix of the derived landmarks is said to be in preform space  $\mathbb{R}^{(N-1)K}$ , while the original figure is in figure space  $\mathbb{R}^{(N-1)K}$  (Goodall in (117) notes that any statistical model for the matrix in the preform space can be derived from the figure matrix in the figure space). The derived landmarks are centered and scaled by:

$$\mathbf{Z}_A = \mathbf{H}^\top \frac{\mathbf{H}\mathbf{A}}{\|\mathbf{H}\mathbf{A}\|}$$

the Procrustes distance between  $\mathbf{A}$  and  $\mathbf{B}$  is thus

$$d(\mathbf{A}, \mathbf{B}) = \inf_{\Gamma, \beta} \|\mathbf{Z}_B - \beta \mathbf{Z}_A \Gamma\|$$

where  $\Gamma = \mathbf{U}\mathbf{V}^\top$ , with  $\mathbf{U}$  and  $\mathbf{V}$  obtained by the singular values decomposition of  $\mathbf{Z}_A^\top \mathbf{Z}_B$  and:

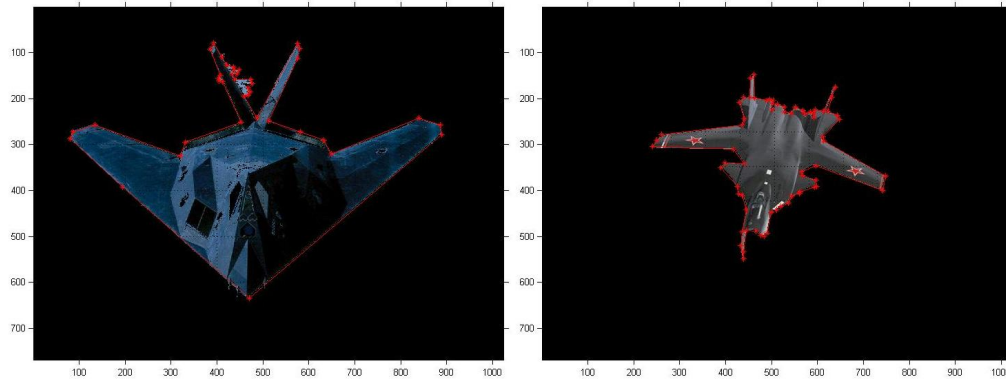


Figure 4.10: Two airplanes isolated from the background with critical points (in red).

$$\beta = \sum_i^k \lambda_i, \quad \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k, \text{ the singular values}$$

Procrustes analysis has many variations and forms. Of these forms, the generalized orthogonal Procrustes analysis (GPA) is the most useful in shape correspondence, because of the orthogonal nature of the rotation matrix, and because  $k$  sets can be aligned to one target shape or aligned to each other. Ross in *Procrustes Analysis* presents an interesting description of GPA. We summarize briefly the algorithm (we address the interested reader to (119; 279; 117; 83; 36)):

- select one shape to be the approximate mean shape (i.e. the first shape in the set);
- align the shapes to the approximate mean shape:
  - calculate the centroid of each shape (or set of landmarks);
  - align all shapes' centroids to the origin;
  - normalize each shape's centroid size;
  - rotate each shape to align with the newest approximate mean.
- calculate the new approximate mean from the aligned shapes;
- if the approximate means from steps 2 and 3 are different, then return to step 2, otherwise we have found the true mean shape of the set.

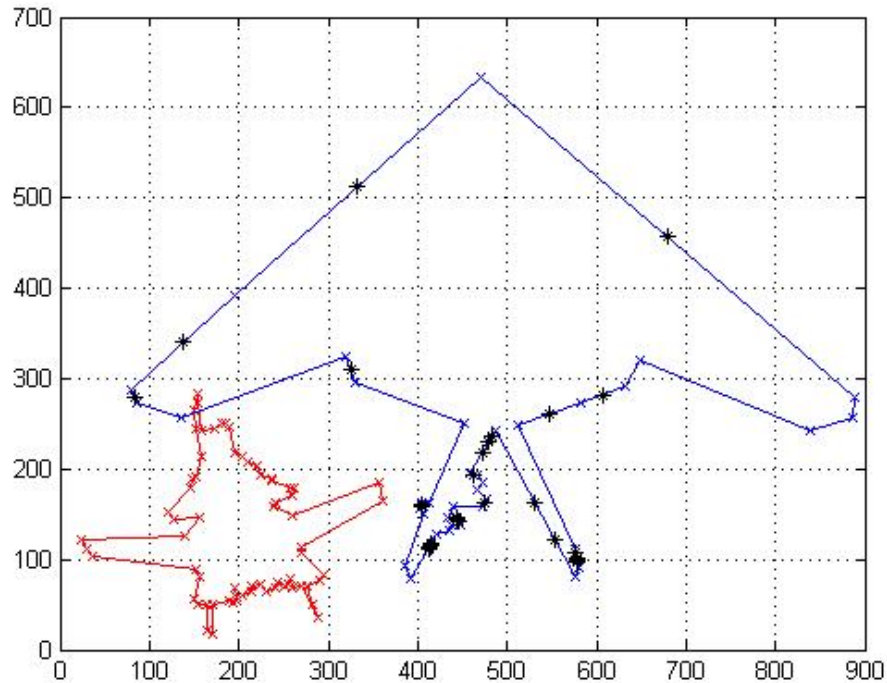


Figure 4.11: The contours of the two airplanes of the previous figure with critical points. The critical points added to make the shapes comparable are colored in black.

Procrustes analysis has several advantages. First of all it is a fairly straightforward approach to shape correspondence. The algorithm's low complexity allows for easy implementation. And these are the reasons why we used it in our analysis. This methodology does however have some disadvantages as well (see (120)). First, it is a rigid evaluation and requires a one-to-one landmark correspondence, both of which limit its correspondence capabilities. Also, the convergence of means is not guaranteed, and therefore convergence is then signified when there is not a significant change in the mean. The major issue for comparing two figures using the Procrustes methodology is the representation of data. This is not a minor problem because the Procrustes methodology, as stated, assumes that figures are denoted by their coordinates in  $\mathbb{R}^k$  which are the figure landmarks. Furthermore it is assumed that the two mentioned figures have the same number of landmarks. Usually when using Procrustes it is necessary to face with different (but strictly connected) problems. First of all to determine what a landmark is and which points of the representation (even

the whole set, if necessary) can be considered belonging to this category. Moreover, assuming that the two figures would be unusually described by the same number of landmarks, we have to define a methodology to "normalize" the two sets, making them comparable. We already addressed the first problem by introducing our critical points representation of the shape. In the following we will describe how to face with the other one.

### 4.3.7 Explicit distance

When a shape  $\mathbf{A}$  is extracted from a patch, it is a  $M \times K$  logical matrix. The contour  $S_A$  extracted, once vectorized, is a matrix  $S_A$ ,  $H \times 2$  of points. The goal is to compare, by the Procrustes analysis,  $S_A$  with the shapes in the repository and to find the relevant category. Herewith we define a threshold: if the distance from the closest class is less than the given threshold, the object belongs to that category; otherwise we have a non-classified object. Procrustes needs the two shapes to be described by the same number of points. We have the contour for each image in the repository and an efficient method to extract this information online. Let  $\mathbf{A}$  be the image extracted from the patch, this is initially resized, by nearest neighbor interpolation, to a logical matrix of  $60 \times 60$ . It is worth noting that the result obtained by the vectorization of the contour matrix  $S_A$  is not suitable, because it is described column-wise, i.e.  $(x_1, y_1), (x_2, y_2), \dots, (x_H, y_H)$  are sorted according to the column indexing, therefore it is necessary to recover the *continuous* function  $f(S_A)$  describing the shape contour and its critical points. The critical points  $\mathbf{Cp}_A$  of  $f(S_A)$  shall constitute the core landmarks. Obviously there might still be no complete correspondence in the number of critical points, even in the case of similar shapes. For example some critical points may be induced by noise or by the steps of the pixels. During the comparisons  $f(S_A)$  needs to be adapted to the target contour  $f(S_X)$  for each element  $X$  in the repository. To obtain this result we follow a simple procedure. Let  $\mathbf{Cp}_A$  be the  $J \times 2$  matrix of the critical points of  $f(S_A)$  and  $\mathbf{Cp}_X$  the  $N \times 2$  (with  $N > J$ ) matrix of the critical points of the target contour  $f(S_X)$ . Then  $\mathbf{L}_A$  is the  $N \times 2$  matrix of the new set of critical points obtained by adding to  $\mathbf{Cp}_A$  new elements, so as to match with  $\mathbf{Cp}_X$ . We are assuming, without loss of generality, that the image under analysis has fewer critical points than the repository image.  $\mathbf{L}_A$  is initially set equal to  $\mathbf{Cp}_A$ ; let  $C_1$  and  $C_2$  be two critical points at maximal distance in  $f(S_A)$ ; then  $p_m$ , a point randomly extracted on the path between  $C_1$  and  $C_2$ , is added to  $\mathbf{L}_A$ . This is repeated until the new set has the same size of  $\mathbf{Cp}_X$ . Obviously if  $|\mathbf{Cp}_A| > |\mathbf{Cp}_X|$ , we follow the same procedure adding points to  $\mathbf{Cp}_X$ .

Given a threshold  $\pi$ , we say that two instances  $A = \{x_i, y_i\}_{i=1..n}$  and  $B = \{x_i, y_i\}_{i=1..m}$  are similar if:

1. there exist approximations  $t_1(A)$  of instance  $A$  and  $t_2(B)$  of instance  $B$  such that  $d_e(\Sigma_X, \Sigma_{t_i(X)}) = 0$ , for  $i = 1, 2$  and  $X \in \{A, B\}$ , where  $d_e$  is the Procrustes distance,  $\Sigma_X$  is the empirical variance-covariance matrix of  $X$  and  $t_i(X) = \{x_h, y_h\}_{h=1..k}$  with  $k = \max(n, m)$ ;
2.  $d_e(t_1(A), t_2(B)) \leq \pi$ , with  $t_i(X)$  defined as in the previous item and  $d$  representing the Procrustes distance.

To satisfy the first condition, it is really important to note that the new contour  $\mathbf{L}_A$  has the same spherical variance and mean as the contour  $f(S_A)$ . Hence:

$$P(p_m | \theta_{f(S_A)}) \approx P(p_m | \theta_{L_A})$$

where  $\theta_X = (\mu_X, \Sigma_X)$ . Therefore the transformation has not affected the properties of the original contour, as far as the critical points remain untouched. In the worst case, the extension of  $\mathbf{Cp}_A$  to  $\mathbf{L}_A$  has to be performed for each comparison. Fortunately the complexity of this step is  $O(h)$ , with  $h = \|\mathbf{Cp}_A\| - \|\mathbf{Cp}_X\|$ .



Figure 4.12: Two digital images of bottles with highlighted critical points (in red).

Figures 4.10 and 4.11 show the above mentioned steps. In the first figure, we can observe the image of two airplanes isolated from the background with the critical

points highlighted, whose number is different between the two airplanes. Instead the second figure shows the contours of the starting objects with critical points, plotted on the same graph. In black are marked the fictitious points added to make them comparable. As we can note, to make the figure clearer, we do not perform the resizing of the original image.

### 4.3.8 Explicit experiments

In this section we present the results from our experiments. We set  $\pi = 0.52$  (a value found experimentally). Consider the two bottles of figure 4.12: after the resizing, the algorithm highlighted the critical points (in red in the figure). Figure 4.13 shows the two original contours of the bottles (drawn on the same graph) and in black, the critical points added to make the sets comparable. On the left side is also shown the transformation of the first contour performed by the Procrustes analysis to fit the other one.

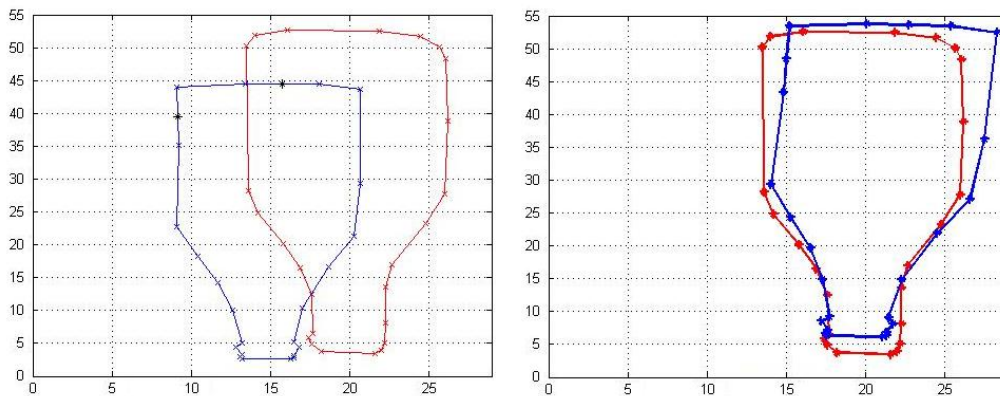


Figure 4.13: On the left, the contours of the two bottles of figure 4.12 with critical points: in blue bottle 1, in red bottle 2. On the right, in blue the contour of bottle 1 after the transformation (and in red the contour of bottle 2).

In our experiment we have considered 1,200 images, among which 500 have been extracted by segmentation, 500 are from databases and 200 have been drawn by hand with a suitable interface. We consider different classes of objects, evaluating the Procrustes distance between instances of the same class and, obviously, between instances belonging to different classes. Consider two families of objects having very different shape characteristics: airplanes and bottles. Let's start with the airplane

family. Figure 4.14 shows our results. The distance values between the instances from 2 to 9 are all below the threshold. Instead the values between the first instance and the other ones are all above it. This fact must not surprise: even if the first element belongs to the airplane family, its shape is different from the others; specifically, the wings are strictly connected with the fuselage and so, differently from the other airplanes, it seems like it has only a big central part. When the algorithm tries to fit this element with another one, the difference in the structure leads to more complex transformation and consequently to a higher distance value. Surprisingly, instead, the seventh airplane has a shape which is slightly different from the other ones; despite this, its distance values are under the threshold and only when it is compared with the fifth airplane, the distance returns a value very close to  $\pi$ ).

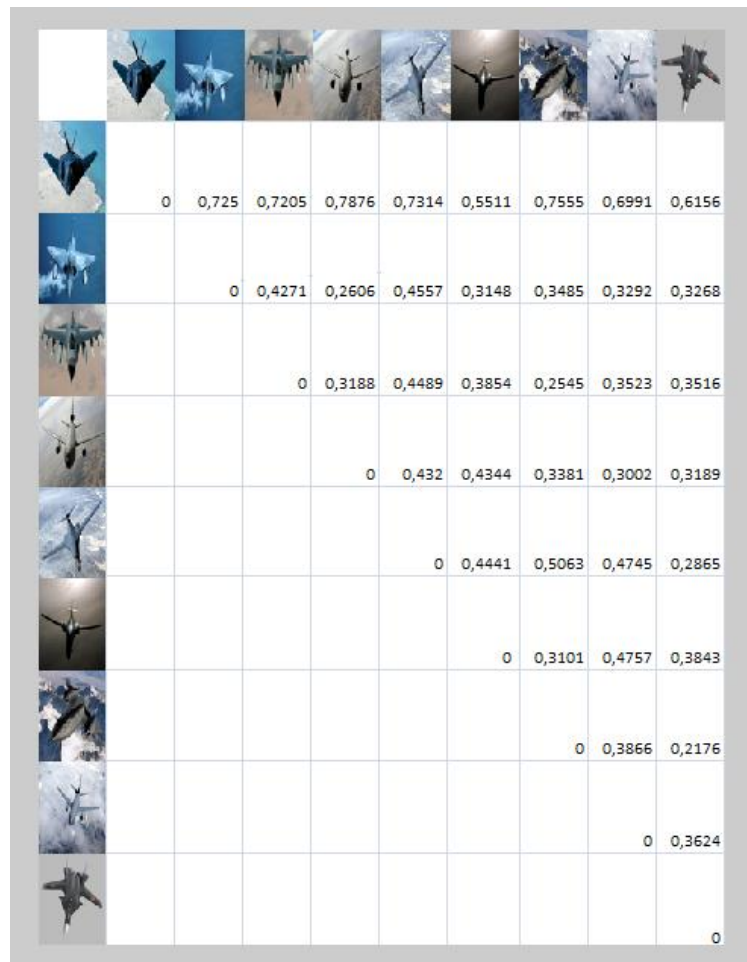


Figure 4.14: Intra-class Procrustes distance in airplane family.



Now we consider the bottles. From figure 4.15 we can observe that almost all the distances are below the threshold  $\pi$ . However, generally, the values are very low. This happens because these shapes are simpler than the ones of the airplanes having less critical points and less protrusions. Only the seventh bottle, whose size is too big to exceed the frame dimensions, has a different behavior: it presents some values above  $\pi$  (with the instances 3 and 5). The fact that the intra-class values between bottles are lower with respect to those between airplanes must not be surprising: the shape complexity of the airplanes, due to the presence of different subparts (which generate boundary local variations), makes it much more complex to describe the shape. The number of critical points is higher and more difficult the computation. Consider that on average we needed 22 points to describe a bottle and 92 to describe an airplane.

It is interesting also to observe the distance values between objects belonging to different classes. Figure 4.16 shows the Procrustes distances evaluated between five bottles and five airplanes. As we can expect, all the distances calculated are mainly over the intra-class distances and they are all over the threshold  $\pi$ .

Instead figure 4.17 shows some results obtained from experiments with instances of a specific class (elephants) in different poses. As we can expect, Procrustes analysis does not perform well in such scenario. Looking at the table, we can observe that generally the distance values are above  $\pi$ .

## 4.4 Implicit Analysis

### 4.4.1 Introduction

It is clear from many experiments on recognition that shape analysis can provide a meaningful support to recognition. Beside being certainly interesting by itself in several applications, it needs however to be combined with dimensional and featural information (extracted from the image), to provide the set of hypotheses necessary to infer some suitable conclusions on the current extracted patch. In fact, as the previous sections have shown, the Procrustes analysis can be seen as a first important step in the classification task, but it cannot be considered a general solution to the problem (see (279)). We introduce now a notion of distance complementary to the Procrustean, taking into account other features. For each object the implicit model consists of a probability density function estimated starting from a set (with no fixed size) of tuples: the chromatic characteristics of every pixel contained inside the boundary. Differently from the explicit approach, here we do not need any procedure to extract this information, nor to adapt the size of the set. The operations for





Figure 4.15: Intra-class Procrustes distance in bottle family.

obtaining the interesting values can be performed very quickly and we only need to estimate an underlying synthetic structure. But, as in the explicit scenario, we need a methodology to compare the two models. For their effectiveness and efficiency, we will use kernel methods.

#### 4.4.2 Bag of tuples and probability density functions

As stated before, the repository contains, for each element, an explicit and an implicit descriptions. The former consists of a figure contour and a set of critical points, constituting the basis for the Procrustes analysis (see section 4.3). The latter (the


					
	0,93385	0,96509	0,94144	0,91631	0,8286
	0,60712	0,66574	0,64268	0,60016	0,73414
	0,65455	0,67073	0,67277	0,60845	0,68691
	0,6334	0,64827	0,67258	0,64986	0,68181
	0,79157	0,64175	0,77768	0,63698	0,57311

Figure 4.16: Inter-class Procrustes distance between airplanes and bottles families.

implicit one) needs to be described in detail. We can treat the patches as a "bug of tuples" (see (146)): every tuple contains the intensity in terms of RGB values. The number of tuples depends on the dimensions of the shape. As already stated, the sets can have different sizes. Obviously this representation is invariant with respect to the permutations of the tuples. But how can we meaningfully compare these models? It is necessary to define a synthetic representation of the set. If we consider these "clouds of pixels" as collections of i.i.d. samples from some unknown probability distribution, we can evaluate these pdfs and use them as synthetic descriptions.

The idea of using a pdf to describe the characteristics of a set of data is very old and in the literature a lot of different approaches exist. In this first experience we chose the Gaussian mixture model. There are a lot of reasons for this choice. First

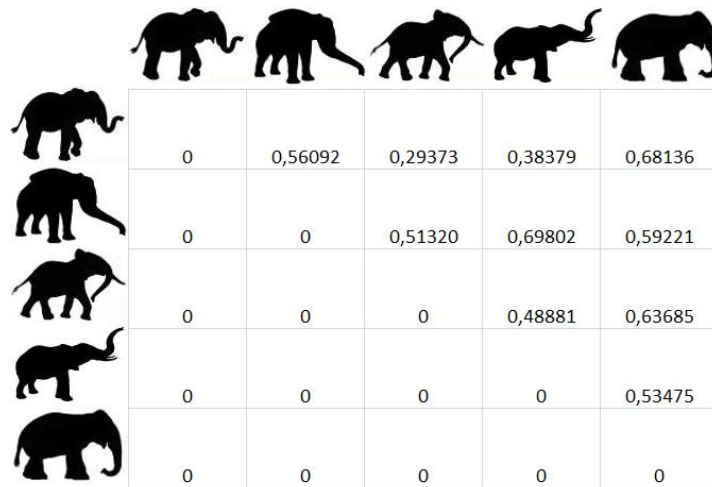


Figure 4.17: A hard case: instances of the same object (elephant) in different poses.

of all, it is a very well-known model. Moreover this choice was supported by the idea that any probability density function can be approximated by a Gaussian mixture (see (221)). We will discuss this subject in a more detailed way in chapter 6. So we have in our depository a set of tuples and an associated Gaussian mixture for every model. And we have to perform the same estimation on the patch under analysis. How can we proceed? If the number of components (say  $N$ ) is known, starting from a set of values and considering them as i.i.d. samples, the underlying mixture of Gaussians can be easily estimated with the Expectation-Maximization technique (77), already presented in the previous chapter. But, as said before, there is a great disadvantage for the EM algorithm: it requires knowing the number of components ( $M$ ) as input parameter. Given  $M$ , EM evaluates, in a finite number of steps, the best fitting  $M$ -components Gaussian mixture for the dataset. But usually we do not have any information about  $M$ . We now present our first approach to estimate  $M$ . For classical methods we address the interested reader to (193; 73; 68; 295).

#### 4.4.3 On the estimation of the components number

In our scenario, the estimation step is performed off-line for all the models in the repository. When we encounter a "new" patch we have to elaborate it on-line to define its own explicit and implicit characteristics. First of all we have to decide (both for the object in the repository and for the analyzed one)  $M$ , the number of components. To obtain this result, we will define a chromatic histogram: the number of peaks will be used as an estimation of  $M$ . If we worked with RGB values, we would

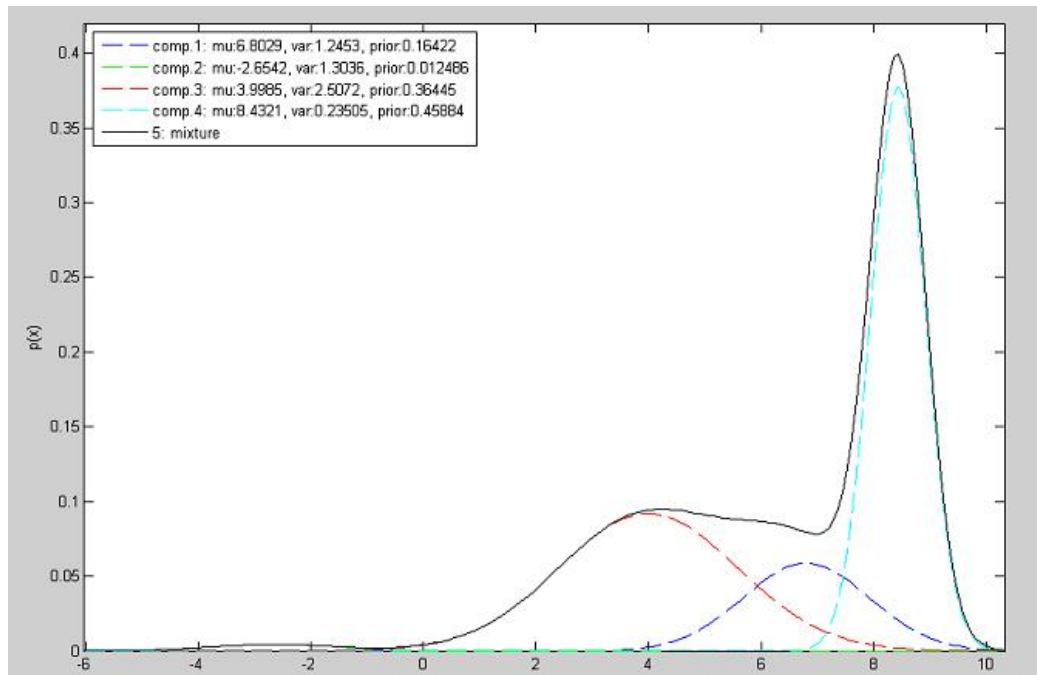


Figure 4.18: A 2-dimensional Mixture of Gaussians with 4 components.

obtain a 3D histogram, over which performing the peak search. In a  $N$ -dimensional structure, the "peaks" correspond to the bins with maximal accumulation. Obviously in a 3D histogram, peak search presents a lot of problems due to the data sparseness on one side and to the increasing complexity of the search algorithm on the other. As shown in (175; 297), a suboptimal implementation of multiband images consists in processing each band separately and then fusing together the different obtained results. Obviously in this way the paucity of data is partly overcome and also the search effort is smaller.

In our specific case, starting from an RGB dataset, we could define 3 two-band analysis: RG, RB and BG band-pairs (as in the methodology presented in (169)). Indeed it is possible to follow another approach. Working in the HSV color-space, we can discard the V component, which refers to the lightness and so brings less information about the color. Afterwards we can calculate the histogram only on the first two components and analyze it. Now there are two problems to face: how to define the size of the histogram bins and how to determine the number of peaks. Considering the resolution that we want on the graph, we decided to divide the interval of H and S (which assume values in  $[0, 1]$ ) into 15 equal-sized sub intervals. Instead,

the estimation of the peaks number is more complex. We adopted the peak-climbing approach developed by Knoontz et al. in (162). We briefly summarize the procedure. Considering an 4-connection neighboring system, for each bin only the neighbors having a higher value are considered. Among them, the bin is connected with the one having the highest value, choosing arbitrarily if there are more cells with this characteristic. So we have defined a parent-childhood relation between bins: each cell can be linked to one parent, but can be parent of more than one cell.

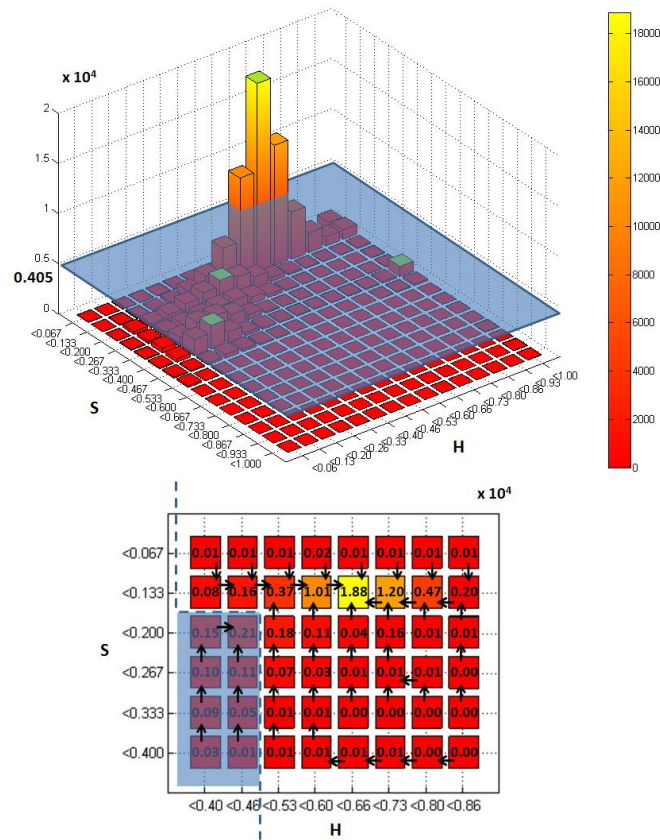


Figure 4.19: The HS 2D histogram from a tiger image (in the upper panel) and the corresponding peak-climbing procedure in the neighborhood of the highest peak (in the lower panel).

Every bin with the largest density in the neighborhood (i.e. a cell with no parent) is a peak. It is possible that the algorithm might bring to scenarios where there are two or more peaks neighbors of each other. If it happens, the algorithm chooses one peak arbitrarily that becomes parent of the other (the assumption is that two or more

neighboring peaks belong to the same cluster). The method is very efficient, in fact in a  $K$ -dimensional histogram we have to consider  $(3^K - 1)$  elements for each cell (so only  $3^2 - 1 = 8$  cells in our case with  $K = 2$ ). The procedure is not iterative and does not require the number of peaks to be specified *a priori*. Considering the wide amount of data in our experiments, we slightly modified the presented procedure, discarding all the peaks having a value below a certain threshold: once the histogram is to hand, we calculate the mean value of the bins (say  $m$ ) and we set the threshold to this value. Figure 4.19 shows an example of the procedure. In the upper part of the image, it is represented the 2D histogram of a set of pixel taken from a picture of a tiger. In green are highlighted the peaks selected with the algorithm. Three of them will be discarded because their values are below  $m = 0.405 \times 10^4$  (represented by the blue plane in the graph). In the lower part of the image, it is shown the peak climbing algorithm in the area around the highest peak. As we can see, all the cells are directed toward a neighboring cell with the highest value.

#### 4.4.4 Implicit distance

To determine the distance between two mixtures of Gaussians there exists a closed form solution introduced by Lyu in (183). It uses kernel analysis (see section 2.2.2), but avoiding the computationally very expensive integral operation. The closed form presented in (183) allows to calculate the Expected Likelihood Kernel, introduced by Jebara and Kondor in (144; 145). Expected Likelihood kernel ( $K_{EL}$ ) is a special case of the Probability Product kernel:

$$K_{\rho}(p, p') = \int p(x)^{\rho} p'(x)^{\rho} dx$$

with  $\rho = 1$ :

$$K_{EL}(p, p') = \int p(x) p'(x) dx$$

Essentially, the Probability Product kernels act as a measure of the degree of similarity between two distributions. Specifically, given two  $d$ -dimensional Gaussian mixtures  $p$  and  $p'$ ,

$$p(x) = \sum_{m=1}^{N_1} \alpha_m G(x; \mu_m, \Sigma_m)$$

and

$$p'(x) = \sum_{m=1}^{N_2} \alpha'_m G(x; \mu'_m, \Sigma'_m)$$

the *Expected Likelihood Kernel* can be computed in a closed form as

$$K_{EL}(p, p') = (2\pi)^{-\frac{d}{2}} \beta^T \Gamma \gamma$$

where  $p$  has  $N_1$  components,  $p'$  has  $N_2$  components,  $\beta = (\alpha_1^{(1)}, \dots, \alpha_{N_1}^{(1)})^T$ ,  $\gamma = (\alpha_1^{(2)}, \dots, \alpha_{N_2}^{(2)})^T$  are the mixing coefficients.  $\Gamma$ , instead, is a function that correlates the different components of the two distributions; it is a  $N_1 \times N_2$  matrix which is formed as  $\Gamma(i, j) = g(\mu_i, \Sigma_i, \mu_j, \Sigma_j)$ , where function  $g$  is defined as:

$$g(\mu_1, \Sigma_1, \mu_2, \Sigma_2) = \frac{|\Sigma|^{\frac{1}{2}} \exp\left(\frac{1}{2} \mu^T \Sigma \mu\right)}{\prod_{i=1}^2 |\Sigma_i|^{\frac{1}{2}} \exp\left(\frac{1}{2} \mu_i^T \Sigma_i^{-1} \mu_i\right)}$$

with  $\mu = \Sigma_1^{-1} \mu_1 + \Sigma_2^{-1} \mu_2$  and  $\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$ . Moreover, as Lyu shows, this measure can also be made independent of the dimensionality of data. This normalization (that we call Normalized Mutual Energy), can be obtained dividing  $K_{EL}(p, p')$  by the product of the energies of  $p$  and  $p'$ . Given a distribution  $p$ , its energy can be calculated as the Expected Likelihood kernel between  $p$  and  $i$ ; that is:

$$M_N(p, p') = \frac{K_{EL}(p, p')}{K_{EL}(p, p) K_{EL}(p', p')}$$

where  $M_N$  is the Normalized Mutual Energy; we can note that its value is in  $[0, 1]$ .  $M_N$  can be used to determine the similarity between two elements. Specifically, given a threshold  $\lambda$ , we say that two datasets  $A = \{h_i, s_i, v_i\}_{i=1 \dots n}$  and  $B = \{h_i, s_i, v_i\}_{i=1 \dots m}$  are similar if

1. there exist probability density functions  $f_A(x)$  and  $f_B(x)$ , such that a set  $X$  can be seen as a sampling of  $f_X(x)$  for  $X \in \{A, B\}$ ;
2.  $d_i(f_A(x), f_B(x)) \geq \lambda$  with  $f_X(x)$  defined as in the previous item and  $d_i$  representing the Normalized Mutual Energy.

It is important to note that the computational cost of this distance is  $O(k_1 k_2)$ , where  $k_i$  is the number of the parameters necessary to describe  $f_X(x)$  (with  $X \in \{A, B\}$ ). Independently from the size of the dataset, the corresponding mixture of Gaussians can be completely described by  $3M$  parameters (where  $M$  is the number of components). So we obtain that the complexity is  $O(M_1 M_2)$ .

#### 4.4.5 Implicit experiments

In this section we present the experimental results obtained by using the implicit methodology previously described. We set  $\lambda = 0,56$  (a value found experimentally).



Figure 4.20: Intra-class distance values of the Normalized Mutual Energy over the tigers family.

In our experiments, we have considered 860 images. Consider a set of tiger pictures, once isolated the interesting pixels, we convert their RGB triplets in HSV values. Then we define the 2D histogram for the H and S components. By using the peak-finding approach earlier presented, we obtain the number of peaks and we use this value as input parameter of the Expectation Maximization. At the end of this procedure, we have the mixture of Gaussians and we are able to determine the Normalized Mutual Energy.

Figure 4.20 shows the intra-class distance values between 7 tiger pictures. As we can see, with few exceptions, the values are all over  $\lambda$ . Only the sixth instance presents 2 values under the threshold. It is also interesting to show some inter-class results. Figure 4.21 shows the distances between tigers and instances belonging to different classes (airplane, truck, dog, butterfly, building); as we can see, mainly the values are below  $\lambda$ . Note that the butterfly presents all distance values over the threshold; this is not surprising: in fact its chromatic characteristics (mainly yellow, green and black) are not so different from the ones of the tigers. Moreover also the distances between the dog and two specific instances of the tiger family are over  $\lambda$ . This can be explained considering that the dog is almost completely white and the colors of those tigers, due to the environment characteristics (snowy landscape), are very light.



					
					
	0,383	0,522	0,459	0,605	0,006
					
	0,505	0,384	0,482	0,614	0,002
					
	0,338	0,523	0,610	0,653	0,059
					
	0,485	0,485	0,411	0,627	0,003
					
	0,448	0,533	0,630	0,675	0,061

Figure 4.21: Inter-class distance values of the Normalized Mutual Energy between the tiger family and other objects, specifically (from left to right): airplane, truck, dog, butterfly, building).

## 4.5 Combining the distances

The explicit description (consisting of the shapes extracted from each patch and the correlated critical points) and the implicit one (consisting of a mixture of Gaussian) conveyed two distance measures: the Procrustes ( $d_e$  as explicit) and the Normalized Mutual Energy ( $d_i$  as implicit). The two distances assume values in the range  $[0, 1]$ , where 1 represents a perfect matching for  $d_i$  and totally wrong for  $d_e$ , while 0 represents a perfect matching for  $d_e$  and totally wrong for  $d_i$ . Now, using these pairs of values, we have to classify every patch, i.e. we have to estimate the class that each of them belongs to. As stated before, our system can rely on a repository containing the descriptions (implicit and explicit) of several models. They are obtained through an off-line analysis of images. To classify a patch, we start evaluating its distance from every stored model.

To combine  $d_i$  and  $d_e$ , at the beginning we tried a very simple approach; given a pair of objects,  $A$  and  $B$ , we considered them similar only if they were similar with respect to both the distance measures simultaneously. That is:

$$A =_S B, \text{ iff } d_e < \pi, d_i > \lambda$$

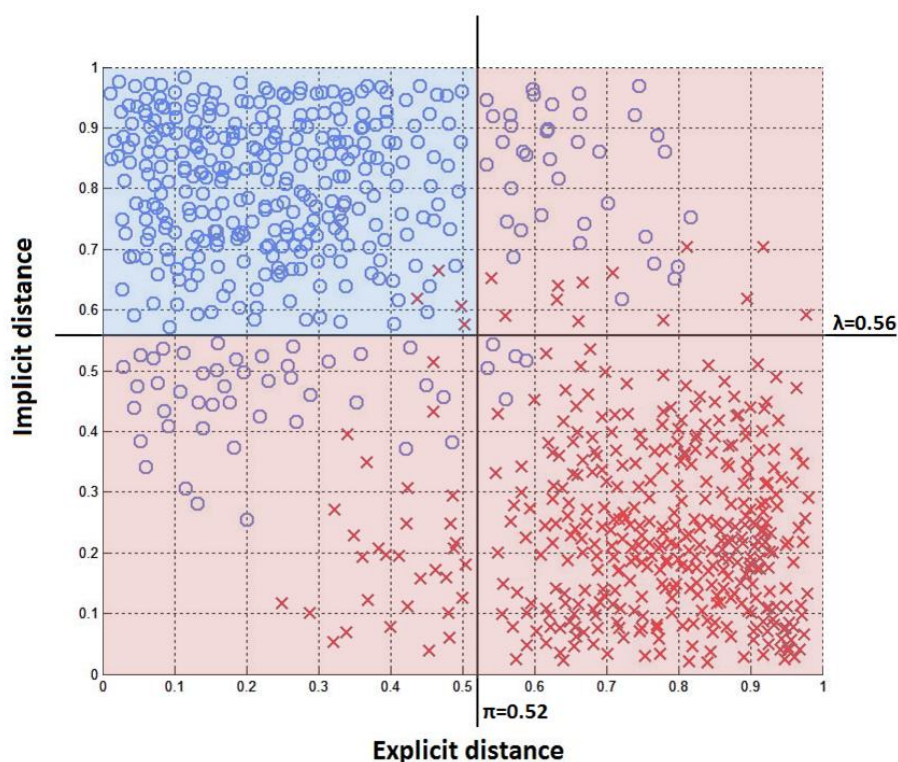


Figure 4.22: On the  $x$ -axes the explicit distance (Normalized Mutual Energy), on the  $y$ -axes the implicit distance (Procrustes distance). The blue circles represent the element belonging to the similarity class, while the red crosses represent the ones belonging to non-similarity. Two elements are considered similar if  $d_e < \pi$  and  $d_i > \lambda$ , that is if the points representing their two distances falls in the blue area.

We consider a set  $S$  consisting of several images belonging to different classes. From the repository, we determine a set  $T$  consisting of two specific instances taken from each category. Then we calculate both implicit and explicit distances between the objects in  $S$  and the elements in  $T$ . We plot the results on a graph having on the  $x$ -axes the explicit distance and on the  $y$ -axes the implicit one. We classify all the instances in two categories: similarity (blue circles) and non-similarity (red crosses). In figure 4.22 is shown the scenario described above. We consider similar only instances with  $d_e < \pi$  and  $d_i > \lambda$ . The blue rectangle represents the selected region. As we can see, there are a lot of false negatives in the upper-right and lower-left areas of the graph (i.e. where only one of the two distance measures holds), but very few false positives.

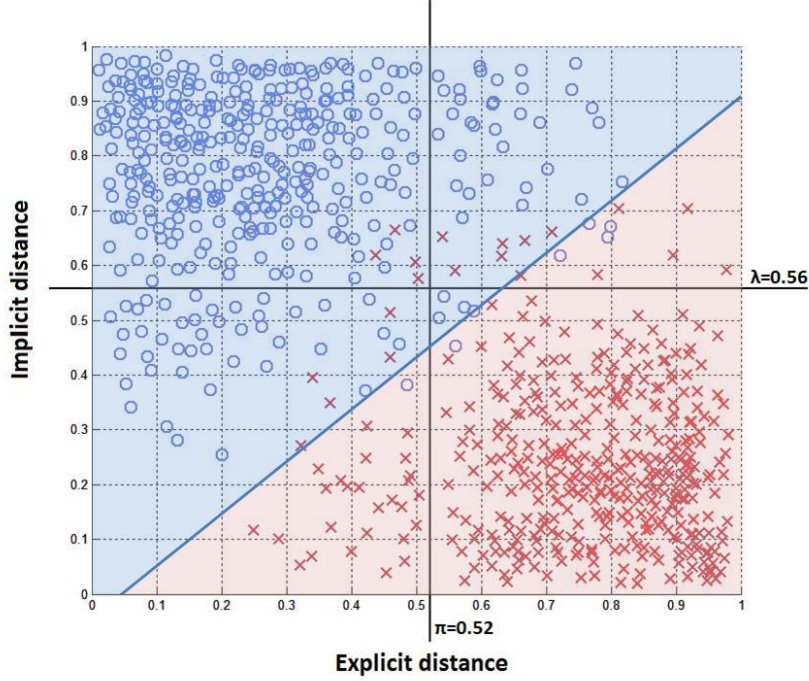


Figure 4.23: On the  $x$ -axes the explicit distance (Normalized Mutual Energy), on the  $y$ -axes the implicit distance (Procrustes distance). The blue circles represent the element belonging to the similarity class, while the red crosses represent the ones belonging to non-similarity. Two elements are considered similar if the points representing their two distances falls in the blue area, obtained by using Fisher discriminant analysis.

We obtain more interesting results by using Fisher Linear Discriminant Analysis (see section 2.2.1). Firstly we assume the sets to be linearly separable:

$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$$

where  $\mathbf{w}^\top$  is the weight and  $w_0$  is the threshold. Therefore a pair  $\mathbf{x} = \langle d_i, d_e \rangle$  is classified *similar*, to the current target, if  $\mathbf{w}^\top (\mathbf{x} - \mu) > 0$  and *non-similar* otherwise.

Using the values obtained on couples of known images, we partition the test samples in  $S = \text{similar}$  and  $N = \text{non-similar}$  with  $\mu_S$  and  $\mu_N$  their mean and

scatter matrix given by:

$$\Sigma_w = \sum_{\mathbf{x} \in S} (\mathbf{x} - \mu_S)^\top (\mathbf{x} - \mu_S) + \sum_{\mathbf{x} \in N} (\mathbf{x} - \mu_N)^\top (\mathbf{x} - \mu_N) \quad (4.1)$$

So that learned weight  $\mathbf{w}$  can be obtained as  $\mathbf{w} = \alpha n \Sigma_w^{-1} (\mu_S - \mu_N)$ , (see (84) for details). We have selected several different images belonging to different classes. We have evaluated the pair of distances between all of them and we have represented these values on the  $d_i/d_e$  plan. We distinguish between the "valid distances" (distances between elements of the same class) and "invalid distances" (distances between element of different classes).

As we can see in figure , the discrimination model, however, is dichotomic, it does not account for *non-classifiable*. On the other hand, if all the measures fall in the invalid side of the discriminative plane, then we consider the patch as not belonging to any category.

Figure 4.23 shows that the combined use of the two distances improves the results obtained: even if there are more false positives, the number of false negatives greatly decreases.

## 4.6 Open problems

The key idea of this approach is the use of two different models to describe objects and to evaluate their similarity: the first tries to capture some explicit information about the boundary, defined by the critical points of the shape and their disposition on the edge; the second one tries to extract implicit information about the chromatic characteristics of the whole set of points in the interior of the boundary. For both descriptions, we introduced a specific distance measure: Procrustes for the explicit, Normalized Mutual Energy for the implicit. The use of two different distance measures allows to estimate how the two objects have similar features and consequently, how much they are similar. In fact it is possible that two objects have the same shape but different chromatic features, and viceversa.

Obviously there are still some open problems both in the general methodology and in the two single procedures. We start observing that the approach presented needs the existence of a repository. As we discussed earlier, this choice causes some problems: first of all, it is necessary to spend some time to initialize the system; moreover the performances of the classification task can rely on the characteristics of the repository itself (number of objects, quality of the images, . . .). Our aim is to obtain a similarity measure that, given two objects, can return a distance between

them, even if the system does not know anything about the "semantic" of the two instances. Obviously, if it was possible, we also would be able to define a meaningful behavior for the scenario in which the system has to analyze unknown object not in the repository.

Concerning the explicit procedure, there are some critical aspects. First of all, we did not introduce a meaningful and compact shape description able to capture all the salient features of the boundary; moreover we opted for the Procrustes analysis, a very powerful tool, but which requires the sets of landmarks used to define the boundary to have the same size. As shown, this condition usually does not hold; even if we introduced a procedure to make two general sets comparable, the solution adopted is not completely satisfying. Finally we want to highlight that Procrustes analysis fails when we have to work with pictures representing the same object in different poses: it considers the global spatial arrangement of the points over the boundary, but it is not able to manage information about the structure of the shape analyzed. In the next chapter we will present some improvements which try to face some of these problems.

Instead, in the implicit analysis, there is mainly one single problem to face: the use of EM requires knowledge about the supposed number of components of the mixture. Obviously different choices about this number lead to very different results. In our first experiments, we try to determine  $M$  by a peak-searching over the HS histogram. The results obtained are not satisfying and it is necessary to explore other approaches to evaluate  $M$ . Chapter 6 will show how the estimation of the modes can help us in this direction.

## Chapter 5

# Shape Description and Analysis

### 5.1 Introduction

In the explicit approach of IEA (see chapter 4) we had to evaluate the distance between two shapes. While analyzing the boundaries, we tried to identify some landmarks to extract the most significant aspects of the shape. We called "critical" those reference points. Starting from them we used the Procrustes distance measure to compare the shapes. The approach presented some limitations; above all, Procrustes analysis requires that the two compared datasets have the same size, a very unusual scenario. We introduced an empiric methodology to get round the problem and make the two sets comparable; specifically we added fictitious points to the set with less elements, demonstrating that also after this transformation its statistical proprieties were preserved. Moreover Procrustes analysis does not return a meaningful comparison between the whole structure of the objects. If we have to compare two shapes representing the same subject in different poses, we can not use Procrustes.

In this chapter we present a more detailed analysis of shape distance estimation. Starting from the idea, earlier presented, that a distance measure for shape heavily depends on the description of the shape itself, at the beginning we define a novel model for 2D shapes, demonstrating some interesting properties about it. Our model is compact, complete, easily computable and, above all, robust to the Euclidean transformations. It will help us in local analysis of a shape (nevertheless without giving a powerful tool for global analysis). For this reason we will subsequently focus on a similarity measure derived from the Fréchet distance and we will show that it can improve shape analysis when we deal with different poses of the same object.



### 5.1.1 Shape description

As Pizlo already explained in his fundamental book on shapes (see (232)), "object's shape is a unique perceptual property of the object in the sense that is the only perceptual property that has sufficient complexity to allow an object to be independent" and so it needs a specific and very detailed analysis. And in fact also in (194), the authors stated that in the last years the problem of classifying objects extracted from an image is often most intuitively formulated as a shape classification task. This is the reason why we focused our attention on shape and shape description. In fact, the way we use to describe a shape heavily influences the study we can perform on it. And it is very hard to obtain meaningful results: shape is corrupted by noise, distortions and occlusions. In the modeling of a shape, a plethora of different features have been considered in literature to face these problems. We can recall shape invariants, shape context, shape signature, shape histogram, moments, curvature and so on. In the last decades have been introduced a lot of different (often very different) descriptions. The work of Pizlo mainly concerned with 3D shapes, while, in this section, we work with 2D elements and so we will point out our attention on 2D. In fact it is worth noting that when a 3D object is projected onto the visual plane, it is perceived as a plane figure, so the capability of understanding 2D shapes (a problem still unsolved) is the first important step toward the analysis of a real object.

Shape descriptors can be classified in several ways. First of all, it is possible to distinguish between global and local descriptions. By using a global descriptor, we have some problems with articulated objects. In fact, even if the pose of the object undergoes little changes, the global structure may be deeply affected (a scenario that can not be handle neither by critical points and Procrustes distance, as we have already shown in chapter 4). On the other side, if we use local descriptors, it is not possible to take into account global information. But we start referring to a review about shape representation and description techniques (310), where the authors, Zhang and Lu, present a meaningful taxonomy of the existing techniques. They divide shape descriptors into two big family: contour-based and region-based. Each of these "families" is subsequently subdivided into structural and global methodologies. In the following we first of all briefly recall the most used contour-based methodologies presented by the authors with a particular attention towards those approaches which influenced our work. Obviously this overview has not to be considered exhaustive and we address the interested reader to the original paper (310) for further details.

**Contour-based structural methods** In this group we can find methodologies which break the shape boundary into segments called primitives. Usually the final result can be represented as a string  $S = s_1, s_2, \dots, s_n$ , where  $s_i$  is a single element and  $S$  represent the boundary. We start considering this family, mainly because it heavily

influenced our shape representation. Chain code (101) is an example: the shape is represented by a set of unit-size segments with the relative orientation; there exist numerous variations of chain code, but they all are sensitive to noise and requires high dimensions. Commonly it is used for a higher level analysis. Another interesting structural method is the so called polygon decomposition: the boundary is broken down into segments by polygon approximation and the vertices are used as primitives (125; 126), described by internal angle, distance from the previous (or next) vertex and spatial coordinates. Given this representation, once selected a subset of the sharpest angles, the distance between shapes is expressed by the editing distance of the two features string. Obviously, given the dependency on polygons, this approach works better with manmade rather than natural objects. In (20) the authors present a methodology which relies on a primitive called token: it consists of the maximum curvature and the orientation of a coherent continue subpart of the boundary and a weighted Euclidean distance is used to evaluate the similarity between two of them. A particular attention has to be pointed out on the so called syntactic analysis. The underlying idea is that, as in the speech recognition, it is possible to define some "basic" phonemes which can be combined together to obtain words and subsequently also very complex sentences (103). Once the shape is described by a string, the matching can be performed by finding the minimal number of edit operations necessary to convert one string into the other. There exist interesting more articulated variations which work with specific grammar, able to describe also the phonemes composition laws. Also shape invariants can be considered structural contour-based methods. Generally with invariants it is specified a group of methodologies based on specific properties of the boundary, which remain unchanged under some transformations (and, obviously, the class of transformations of interest). Cross-ratio, length-ratio, distance-ratio, angle, area, determinant, eigenvalues, curvature (173; 138; 269) are all examples of invariants (geometrical, algebraic or differential).

**Contour-based global methods** These techniques compute a multi-dimensional numeric feature vector from the shape boundary information. The matching between shapes is a straightforward process, usually obtained by using a metric distance. First of all we can consider the most simple global descriptors such as area, circularity, eccentricity, axis orientation and bending energy (307) or other ones more complex as convexity, axis ratio, circular variance and elliptical variance (228). These descriptors usually can return only very coarse information allowing to discriminate between shapes having large differences. Another interesting instance of this group is the Hausdorff distance: represented the shapes by two sets of points  $S_1 = \{s_1^1, \dots, s_n^1\}$  and  $S_2 = \{s_1^2, \dots, s_m^2\}$ , the Hausdorff distance ( $H$ ) is defined as:

$$H(S_1, S_2) = \max(h(S_1, S_2), h(S_2, S_1))$$

where

$$h(S_1, S_2) = \max_{s_i \in S_1} \min_{s_j \in S_2} \|s_i - s_j\|$$



and  $\|\cdot\|$  is the underlying norm of points of  $S_1$  and  $S_2$ . Even if this distance measure is not invariant to translation, scale and rotation and, at the same time, is too sensitive to noise or outlier, in literature there exist a lot of implementations relying on it (59; 140; 197). Among the numerous variations proposed, we want to recall that Rucklidge extended Hausdorff distance matching into affine invariant matching (244). Also shape matching using shape context (16) can be seen as an improvement to the traditional Hausdorff distance methods: it extracts a global feature (the so called shape context) for each corresponding point. The matching between the context features represents the matching between shape points. Instead shape signature represents a shape by a one dimensional function derived from the boundary points. We can recall centroidal profile, complex coordinates, centroid distance, tangent angle, cumulative angle, curvature, and chord length (75; 288; 311). Even if they can be derived simply (and with a very low cost), the matching techniques are expensive and, generally, small modifications on the boundary can lead to great difference in the representation of the shapes. Assuming that the shape boundary has been represented by shape signature, it is also possible to derive boundary moments. Specifically, having a shape signature  $z(i)$ , the  $r$ -th moment  $m_r$  and central moment  $\mu_r$  can be estimated as:

$$m_r = \frac{1}{N} \sum_{i=1}^N [z(i)]^r$$

and

$$\mu_r = \frac{1}{N} \sum_{i=1}^N [z(i) - m_1]^r$$

where  $N$  is the number of boundary points. The normalized moments  $\bar{m}_r = m_r / (\mu_2)^{r/2}$  and  $\bar{\mu}_r = \mu_r / (\mu_2)^{r/2}$  are invariant to shape translation, rotation and scaling (267). Another interesting approach is represented by the so called elastic matching, proposed firstly by Bimbo and Pala (76); a deformed template is generated as the sum of the original template  $\tau(s)$  and a warping deformation  $\theta(s)$ :

$$\phi(s) = \tau(s) + \theta(s)$$

where  $\tau = (\tau_x, \tau_y)$  is a second order spline and  $\theta = (\theta_x, \theta_y)$  is the deformation. This description is associated to a specific function able to return a similarity measure. We can also cite stochastic methods like autoregressive modeling: considering a function  $f$  describing the boundary, a linear autoregressive model can express the value of  $f$  as the linear combination of a certain number of preceding values. Given a set of observation, the model predicts the value of the next one and, specifically, it defines the current value of the property considered (i.e. radius) by a linear combination of the values already observed, with a clear reduction of the complexity.

Golland, in her PhD thesis on statistical shape analysis of anatomical structures (114), gives another interesting classification of shape descriptors which allows us to introduce other interesting methodologies:

**Parametric descriptors** The methods in this family fit a parametric model to a boundary surface in a 3D image, or an outline curve in a 2D image, and use the model parameters as feature vector components. The best known parametric shape descriptors are based on decomposition of the object using a particular functional basis, such as the Fourier series (271; 275) or the harmonic functions (45; 157). The model parameters are typically extracted from segmented images.

**Deformation fields** Descriptors in this class are based on non-rigid matching of a template to an input image. Additional constraints on the resulting deformation field stabilize the inherently under-constrained matching problem. Examples of regularization models that ensure smoothness of the field include thin plate splines (32; 115), elasticity constraints (74; 184; 190) and viscous fluid models (61; 71).

**Distance transforms** A distance transform, or distance map, is a function that, for each point in the image is equal to the distance from that point to the boundary of the object. The boundary is modeled implicitly as a zero level-set of the distance transform. A signed variant of the distance transform, which negates the values of the distance transform outside the object, eliminates the singularity at the object outline, changing linearly as we cross the boundary. Distance transforms have been used in computer vision for medial axis extraction (115; 172), and more recently, in medical image analysis for shape description (115; 170). The distance transform is computed from a binary segmentation of the object.

A particular attention has to be given to the landmarks based methods, strictly connected to our approach. A landmark is presented as a point on the object boundary, a surface in a 3D image, a curve in a 2D image, that can be reliably estimated from an image. Landmarks can be placed manually by the users who employ their knowledge of anatomy to identify special locations (32; 67; 66), or detected automatically using geometric properties of the outline surface, such as curvature (223). A special attention has to be given to the use of these points to describe a shape. In this direction a very interesting and efficient approach is presented in (194). The authors introduced a new description of shape boundaries using a set of equally spaced points (avoiding the need to extract specific landmark points). Much of the work in this area uses a finite set of points taken from the object's boundary as the shape representation. Very different approaches exist for the choice of the reference points. They can

be selected on the basis of maximal curvature (274), distance from the centroid (312) or any criteria deemed suitable to the class of shapes involved. More sophisticated approaches parameterize the boundary as a closed curve and slide points along the outline to minimize an objective function (298). Another interesting approach is to simply place points at roughly equal intervals along the boundary (17). As we show hereby, in this work we preferred selecting the landmarks in the critical points (i.e. where the contour has a sudden variation in the direction).

There exist a lot of other methodologies, not included in the previous lists. Among the approaches based on shapes, there exist other procedures relying on boundary (see e.g. (9; 159; 30; 33) for early works), other ones based on fill-in features (e.g. (96)) or on skeleton representation of the shape and their shock graphs (we will give more details about these methodologies in the following). Coordinate systems (Cartesian, polar, tangential, etc.), length and curvature representation (which represents every contour segment by its length and its angle with the axis), B-spline, statistical moments, fractals belong to the so called contour-based methods. Other approaches use the Fourier transformation of the boundaries or the estimation of the tolerance interval. On the other side there are region-based shape representations; specifically there are some heuristic approaches which allow to extract important features from simple shapes. These techniques can be applied to the analysis of more complicated shapes, by decomposing these latter in smaller and simpler sub-regions, and by describing them separately. Some of the features extracted from shapes are: area (number of pixels inside the contour), Euler's number (difference between the number of contiguous parts and the number of holes in an object), vertical and horizontal projections, elongatedness (a ratio between the length and width of the region boundary rectangle), rectangularity (maximum ratio of the region area and the area of a variable-direction bounding rectangle), direction (the direction of the longer size of a minimum bounding rectangle) and compactness (ratio between the region border length and the area). Another interesting approach calculates the convex hull of a shape, i.e. the smallest convex region which contains all the points of the shape (for an exhaustive description of all the methodologies cited above, we address the interested reader to (268; 231; 286; 245)). Finally we want to cite (10): the authors present an interesting use of the Spherical Harmonic Descriptor (SHD, see (156; 155) which has been shown empirically to perform well for the task of shape matching (see for example the applications of SHD in (107; 200; 260).

A specific attention has to be given towards the so called skeleton approaches (briefly already introduced in chapter 4). Medial axis was firstly introduced in the 70's by Blum (30) as a tool for biological shape recognition. In the last years many researchers point out a renewed attention toward this topic which has allowed to obtain a richer (and compact) description of a shape boundary. In (160), the authors firstly introduced the idea of characterized boundary shapes using the differential

singularities of the reaction equation: in fact, by using the reaction-diffusion equation, the boundary evolves in the skeleton, which represents the singularities in the curve evolution (the locus where there are the collisions between the evolving inward moving boundaries). There exist a lot of methodologies to calculate the skeleton (see (7; 216)). Recently Torsello in his PhD thesis (282) presents a variation of the classical method introduced by Siddiqi et al. (262), who showed an interesting approach which relies on the Hamilton-Jacobi formalism to solve the eikonal equations. The new procedure seems to be more stable with respect to local variations. In chapter 8 we will describe how we are attempting to merge these methodologies with our approach.

### 5.1.2 Framework

A meaningful formal shape description has to capture all the salient features. According to (109), we are interested in a representation that is:

- **complete**, i.e. the whole shape information is stored in the representation;
- **simple, meaningful** and **compact**, i.e. redundancies are removed by capturing symmetries, region information, articulations (local deformations), and dividing the object into "object parts" with as few parts as needed for any level of specified details;
- **stable**, i.e. robust under small variations of the shape;
- **easily computable**, i.e. polynomial time algorithms (on the size of the shape) exist to find its representation.

We also add another important characteristic to this list:

- **comparable**, i.e. it is easy to define a suitable notion of distance between two different shape descriptions.

The distance, in fact, has to capture a qualitative concept of similarity between objects and the class they belong to. Analyzing a set of object images, obviously there might be a large degree of variation in position, viewing direction, illumination and other aspects. Ideally we want to obtain a representation capable to overcome all these aspect changes. Given the plethora of different approaches and methodologies to represent and to analyze a shape, we first need to define a framework to establish how well a representation works. (268) shows how description methods can be characterized from different points of view:

- input representation form;

- object reconstruction ability;
- incomplete shape recognition ability;
- local/global description character;
- statistical or syntactic object description;

and, above all,

- robustness to translation, rotation and scale transformation.

These are crucial aspects to understand how good a measure is. In particular, we want to focus our attention on the latter aspect: the robustness to rigid-body transformation. We recall again that according to Kendall's definition (see (158)), shape has to be invariant to Euclidean similarity transformations. Kendall defines also the idea of *pre-shape*, which is the last step towards shape, in which rotational effects still need to be filtered out. In the following we recall this idea of pre-shape with some refinements. By using all these techniques, several significant results have been established in the literature (e.g. see (9; 280; 263; 11; 83; 18; 253; 255; 252; 270)), but, as said, a satisfactory methodology is not available yet.

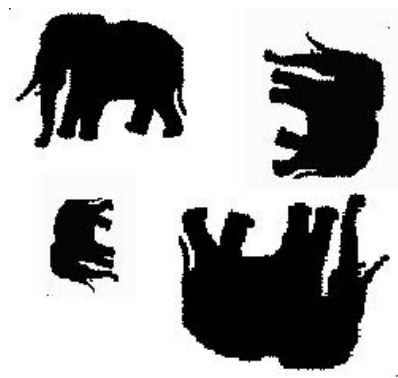


Figure 5.1: Different transformed copies of the same shape.

## 5.2 Shape features representation

### 5.2.1 Shape pattern

Once found the critical points (see section 4.3.5), we construct a new image (called *polygonal image*). The new image is an abstraction of the original one. It is a closed

boundary obtained connecting by straight lines the contiguous critical points. This step is necessary to overcome the pixelization problem. Figure 5.2 shows an example of a polygonal image, obtained starting from the shape of figure 5.1.

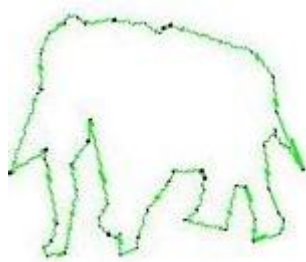


Figure 5.2: The polygonal version of the image in figure 5.1. Boundary points are colored in green, while critical points are colored in black.

Now it may be interesting to obtain additional information about the local characteristics of the boundary around the critical points. To obtain this result, we introduce the shape patterns:

**Definition 7: Shape Pattern**

A shape pattern  $L_n$  (with  $n$  odd) is a  $n \times n$  matrix with all 0s and  $n$  1s. The 1s describe the configuration of  $n$  consecutive contour pixels, considering one of them as the central pixel. Every shape pattern is centered on this specific point (a single pixel) so the central element of the matrix, which represents this pixel, is always set to 1.

Under the hypothesis of pixel-continuous single-thick boundary, we can obtain a specific shape pattern  $L_n$ , simply centering a  $n \times n$  window over one of the boundary pixels and considering the other pixels which fall into the window. As stated before, once we have found the critical points, the shape patterns can help us to observe the local variations of the boundary. Obviously, adjusting the value of  $n$ , we can decide the dimension of the boundary segment to analyze. Our experiments have shown that also considering the smallest set (i.e.  $n = 3$ ), we can obtain relevant information about the local variation. So in the following we start pointing our attention on the  $L_3$  shape patterns. It is easy to show that only 28 of these patterns exist; in fact, putting the first 1 in the center of the  $3 \times 3$  matrix, we have 8 different positions for the second 1 and for the third 1 we can chose one of the 7 left positions. So:

$$28 = \frac{8 \times 7}{2}$$

We use  $Q^3$  to represent the set of all  $L_3$  shape patterns. The element of  $Q^3$  will be used as basic elements to describe a shape. In the following we show four of these patterns, denoted by  $A, B, C$  and  $D$ :

$$\begin{array}{cc} A & B \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ C & D \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

In this work the coordinates of the points in the  $3 \times 3$  matrices are calculated with respect to the central point. So, the coordinates of the center point are  $(0, 0)$  and the other 8 positions are described using  $\{-1, 0, 1\}$ , as shown in the following:

$$\begin{bmatrix} (-1, -1) & (-1, 0) & (-1, +1) \\ (0, -1) & (0, 0) & (0, +1) \\ (+1, -1) & (+1, 0) & (+1, +1) \end{bmatrix}$$

Using this notation, for example the pattern  $A$  is uniquely described by

$$\{(-1, -1), (0, 0), (-1, 0)\}$$

Obviously the order of the couples does not matter. We can now define a function which, given a shape, returns the corresponding shape patterns.

### Definition 8: Pattern Mapping Function

Let  $C$  be a pixel-continuous and closed-shape contour of  $n$  pixels; let  $L$  be a clockwise topological ordered configuration of a set of landmarks for each point location of  $C$  ( $|L| = n$ ) and  $Q_k^3$  ( $k \in \{1, \dots, 28\}$ ) a  $L_3$  shape-pattern. Let  $X = \langle (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}) \rangle$  be a set of three consecutive points in  $L$  (chosen with respect to the clockwise topologically order), and let  $l : Q^3 \mapsto (\{-1, 0, +1\} \times \{-1, 0, +1\})^3$  be the function which maps each pattern  $Q_k^3$  in its own set of landmarks. Then it is possible to define a mapping function  $f : X \mapsto l(Q)$ , where  $f$  returns the landmarks of the pattern corresponding to  $X$ .  $f$  is defined as follows:

$$\begin{aligned} f(x_{i-1}, y_{i-1}) &= (x_{i-1} - x_i, y_{i-1} - y_i); \\ f(x_i, y_i) &= (0, 0); \\ f(x_{i+1}, y_{i+1}) &= (x_{i+1} - x_i, y_{i+1} - y_i) \end{aligned}$$

**Lemma 1**

Let  $L$  be a finite clockwise topologically ordered set of  $n$  landmarks, each corresponding to a point location of a pixel-continuous and closed shape. Then  $L$  can be factored uniquely into a sequence of patterns in  $Q^3$ . We will obtain a pattern for each landmark in  $L$ . It is possible to define a function  $F : L \mapsto (Q^3)^n$  which takes  $L$  as input and returns the ordered sequence of  $n$   $L_3$  patterns corresponding to the points of  $L$ .

**Proof**

$L$  contains  $n$  points. It is possible to define  $n$  triplets. For each point  $i$  of  $L$  we take the previous and the following points (according to the clockwise topological order). The  $i$ -th triplet is then  $\langle (i-1) \bmod n, i, (i+1) \bmod n \rangle$ . The function  $f$  previously introduced maps each triplet in one of the 28 shape patterns. So we obtain  $n$   $L_3$  patterns.

□

Consider for example the set of landmarks  $L = \{(2, 3), (3, 4), (4, 4), (4, 3), (3, 2)\}$  of the pixel-continuous and closed shape shown in figure 5.3.  $L$  is a clockwise topologically ordered configuration. The factorization obtained by calculating  $F$  on  $L$  is:  $E, M, C, O, X$  where:

$$\begin{array}{ccc} E & M & C \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\ \\ O & X & \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} & \end{array}$$

For the point  $(3, 4)$ , for example, in fact we obtain:

$$f((2, 3), (3, 4), (4, 4)) = \{(2, 3) + (3, 4), (0, 0), (4, 4) - (3, 4)\} = \{(-1, -1), (0, 0), (1, 0)\}$$

and so for the other points.

The 28 patterns can be grouped in 4 classes:  $C_1, C_2, C_3$  and  $C_4$ . Each category contains a basic shape pattern and other 7 elements which are simply rotations of the basic pattern. Obviously the fourth category ( $C_4$ ) has only 4 shape patterns (the others are identical to these, two by two). The four basic patterns for  $C_1, C_2, C_3$  and  $C_4$



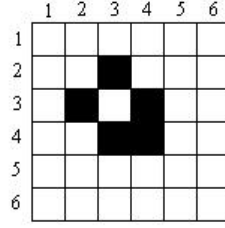


Figure 5.3: A simple pixel-continuous and closed shape.

are, respectively, the  $A$ ,  $B$ ,  $C$  and  $D$  patterns previously showed.

#### Notational convention

Let  $Y$  be the basic pattern for a specific class ( $Y \in \{A, B, C, D\}$ ), we indicate with  $Y^\alpha$  the pattern obtained by a clockwise rotation of  $\alpha \times \frac{\pi}{4}$  degrees starting from  $Y$ :

$$\begin{matrix} B & I = B^1 \\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (5.1)$$

We also represent a segment  $s$  of length  $l$  and angle with y-axis  $\beta$  with  $l^\beta$ . It is simple to observe that  $s$  is uniquely defined by  $l^\beta$ .

## 5.2.2 Shape Description

With all the elements previously defined to hand, we can introduce our shape description:

### Definition 9: Shape Description

Let  $X$  be a  $k \times 2$  matrix representing a clockwise topologically ordered configuration of  $k$  points describing a close and continuous contour of a shape  $S$  having  $n$  critical points. We use the following notation to describe  $S$ :

$$S = \langle \langle C_1^{\alpha_1}, C_2^{\alpha_2}, \dots, C_n^{\alpha_n} \rangle, \langle l_{N1}^{\beta_1}, l_{N2}^{\beta_2}, \dots, l_{Nn}^{\beta_n} \rangle \rangle$$

where  $\langle C_1^{\alpha_1}, \dots, C_n^{\alpha_n} \rangle$  and  $\langle l_{N1}^{\beta_1}, l_{N2}^{\beta_2}, \dots, l_{Nn}^{\beta_n} \rangle$  are clockwise topologically ordered configurations,  $C_i \in \{A, B, C, D\}$ ,  $\alpha_i \in \{1, 2, \dots, 8\}$ ,  $l_i \in \mathfrak{R}^+$ ,  $\beta_i \in [0, \pi[$  and  $C_i^{\alpha_i}$  represents the pattern centered on the  $i$ -th (with respect to the clockwise topological ordering) critical point.  $l_{Ni}^{\beta_i}$  represents the  $i$ -th straight segment connecting the

$i$  -  $th$  and the  $(i + 1) - th$  critical points. In particular, let  $l_i (\forall i \in \{1, \dots, n\})$  be the length of the  $i - th$  segment and  $l_{MAX} = \max_{i=1, \dots, n} (l_i)$ , then  $l_{Ni} = \frac{l_i}{l_{MAX}}$ . Obviously  $l_{Ni} \in [0, 1]$ .

### 5.3 Properties of Shape Descriptors

First of all we want to show the properties of our shape representation with respect to the lists presented in the previous section.

Our representation is:

- **simple**: we describe a shape by using very few information;
- **meaningful**: we preserve information about the number, the relative position and the main characteristics of the critical points;
- **compact**: having a boundary of  $n$  pixels with  $m$  critical points (with usual  $m \ll n$ ), we need only  $4m$  values to represent the boundary;
- **stable**: small variations of the shape do not affect the description (that also solves the pixelization problem);
- **easy to compute**: once obtained the critical points of the boundary ( $O(n)$ , where  $n$  is the size of the boundary), it is very simple to define  $C_i, \alpha_i, l_{Ni}$  and  $\beta_i$ .

Above all we want to demonstrate that the shape description defined is robust versus translation, rotation and scale transformations. In the following we present these important results.

#### Lemma 2

The representation presented is invariant with respect to translations.

#### Proof

Let  $S$  be a shape and  $S'$  a 2D-translation  $(\Delta_x, \Delta_y)$  of  $S$ . The length ( $l_{Ni}$ ) and the angle ( $\beta_i$ ) of the segments do not change after the translation. Moreover the critical points of  $S'$  ( $C_i^{\alpha_i}$ ) are the same of  $S$  (for the definition of critical points). The patterns in the critical points are obtained using the mapping function  $f$  previously introduced. Given two critical points  $a$  and  $a'$  where  $a$  is taken from  $S$  and  $a'$  is the corresponding critical point in  $S'$ ,  $f$  returns the same pattern on  $a$  and  $a'$ . In fact:

$$\begin{aligned} f(x_{i-1} + \Delta_x, y_{i-1} + \Delta_y) &= (x_{i-1} - x_i, y_{i-1} - y_i) = \\ &= f(x_{i-1}, y_{i-1}); \end{aligned}$$

$$\begin{aligned} f(x_i + \Delta_x, y_i + \Delta_y) &= (0, 0) = \\ &= f(x_i, y_i); \end{aligned}$$

$$\begin{aligned} f(x_{i+1} + \Delta_x, y_{i+1} + \Delta_y) &= (x_{i+1} - x_i, y_{i+1} - y_i) = \\ &= f(x_{i+1}, y_{i+1}). \end{aligned}$$

□

**Lemma 3**

The representation presented is invariant with respect to scaling.

**Proof**

Let  $S$  be a shape and  $S'$  the shape obtained by a  $k$ -factor scaling of  $S$ . Note that orientation and angles are the same, thus the patterns over the critical points are unaffected, hence  $\langle C_1^{\alpha_1}, \dots, C_n^{\alpha_n} \rangle = \langle C_1^{\alpha'_1}, \dots, C_n^{\alpha'_n} \rangle$ . Also the angle ( $\beta_i$ ) of each segment  $l_{Ni}$  is unaffected. The length of each segment will be multiplied for the scaling factor  $k$ . But the values  $l_{Ni}$  are normalized with respect to  $l_{MAX}$ , so they will be unchanged. In fact ( $\forall i \in \{1, \dots, n\}$ ):

$$l'_{Ni} = \frac{l'_i}{l'_{MAX}} = \frac{k \times l_i}{k \times l_{MAX}} = \frac{l_i}{l_{MAX}} = l_{Ni}.$$

□

The representation is not invariant to rotations, but it is very simple to update when a rotation occurs. In the following we show how to update the structure:

**Lemma 4**

Given a shape  $S = \langle \langle C_1^{\alpha_1}, \dots, C_n^{\alpha_n} \rangle, \langle l_{N1}^{\beta_1}, \dots, l_{Nn}^{\beta_n} \rangle \rangle$ , the new shape  $S'$  obtained by a clockwise  $\gamma$ -degree rotation of  $S$  has the following structure:

$$S' = \langle \langle C_1^{\alpha'_1}, \dots, C_n^{\alpha'_n} \rangle, \langle l_1^{\beta'_1}, \dots, l_n^{\beta'_n} \rangle \rangle$$

where  $\forall i \in \{1, \dots, n\}$ :

$C'_i = C_i$  (i.e. the basic patterns  $C_i$  remain the same),

$\alpha'_i = (\alpha_i + [\gamma \times \pi/4]) \text{ mod } 8$ ,

$l'_i = l_i$ ,

$$\beta'_i = (\beta_i + \gamma) \bmod (2\pi).$$

where  $[x]$  represents the nearest integer to  $x$ .

**Proof**

$S'$  is the shape obtained by a  $\gamma$ -degree rotation of  $S$ . The rotation does not change the length of the segments ( $l_{Ni}$ ) neither the basic patterns overimposed to the critical points ( $C_i$ ).

□

So, recalling the definitions given previously, our representation is not feasible for shape, but for pre-shape. Our shape description allows us to obtain quickly the most significant characteristics (including the critical points) but does not suggest a way to compare them.

## 5.4 Similarity and Distance

In this section we discuss a methodology for evaluating the similarity between two-dimensional shapes, each one represented by a set of topologically ordered critical points. In this framework we use a different notation, representing the boundary points by a vector  $Z = \langle Z_1, Z_2, \dots, Z_k \rangle$  of complex numbers. The transformation from Euclidean to complex coordinates is the following:

$$Z_j = x_j + iy_j$$

for  $j = 1, 2, \dots, k$ . Consider the Fréchet distance:

**Definition 10: Fréchet distance**

Given two parametric curves  $f : [a, b] \rightarrow V$  and  $g : [a, b] \rightarrow V$ , the Fréchet distance between them is

$$d_F(f, g) = \min_{\alpha, \beta} \max_{t \in [0, 1]} d(f(\alpha(t)), g(\beta(t)))$$

where  $\alpha$  and  $\beta$  are continuous parameterization of  $f$  and  $g$  respectively.

Fréchet distance is easy to understand thinking to a man that walks his dog. Each of them walks on a different curve and  $d_F$  is the minimum length of the leash that traverse them. Configuration points can be seen as the edges of a polygonal approximation of shape contours, so we introduce a proper version of Fréchet distance illustrated in (87). Given a polygonal curve  $z$ , we denote the sequence of its end points as  $\sigma(z) = (z_1, \dots, z_k)$ .

**Definition 11: Coupling**

Let  $Z$  and  $Y$  be two polygonal curves and  $\sigma(Z) = (Z_1, \dots, Z_p)$  and  $\sigma(Y) = (Y_1, \dots, Y_q)$ , a coupling  $\Lambda$  is a sequence

$$\Lambda = c(Z, Y) = (Z_{u_1}, Y_{v_1}), \dots, (Z_{u_m}, Y_{v_m})$$

where  $u_1 = v_1 = 1$ ,  $u_m = p$ ,  $v_m = q$ ,  $m = \min(p, q)$ .

The length of a coupling  $\Lambda$  is defined as:

$$|\Lambda| = \max_{i=1 \dots m} d(Z_{u_i}, Y_{v_i})$$

We also introduce the discrete Fréchet distance:

**Definition 12: Discrete Fréchet distance**

Given two polygonal curves  $Z$  and  $Y$ , the discrete Fréchet distance between them is

$$d_{dF}(Z, Y) = \min \{ |\Lambda| : \Lambda = c(Z, Y) \}$$

where  $c(Z, Y)$  is a coupling between  $Z$  and  $Y$ .

The Fréchet distance is independent of shape configurations, but before computing it, we have to find a *significant* alignment between the configurations.

**Definition 13: Sub-Configuration Set**

Given a configuration  $Y$  such that  $|Y| = q \geq p$ , then

$$S_p^T(Y) = \left\{ Y' \subseteq Y : |Y'| = p \text{ and } Y' \text{ is topologically ordered} \right\}$$

is the set of topologically ordered sub-configurations of cardinality  $p$  that can be derived from  $Y$ . For any  $q \geq p$ , we have  $|S_p^T(Y)| = \binom{q}{p}$ .

**Definition 14: Common Structure Registration (CSR)**

Let  $Z$  and  $Y$  be two configurations (without loss of generality suppose  $|Z| = p \leq q = |Y|$ ). A common structure registration is a matching between  $Z$  and  $Y^S$  such that

$$Y^S = \operatorname{argmin}_{Y' \in S_p^T(Y)} d_P(Z, Y')$$

The hypothesis that undergoes the CSR is that if two configurations represent objects of the same class, for example two elephants in different poses (see figure 5.4), then they have to differ only for some details. In fact, objects of the same class have a similar structure, which is supposed to be more evident in the configuration with fewer points. So the CSR finds the subset of the points of  $Y$  which is the configuration more similar to  $Z$ . Since  $|Y_p^T| = \binom{q}{p}$ , the cost of finding a CSR between two configuration is  $O\left(\binom{q}{p}\right)$ . Obviously if a common structure between the two configurations does not exist, because they represent different objects, there will be a value of the objective function higher than a fixed threshold  $\xi$ .

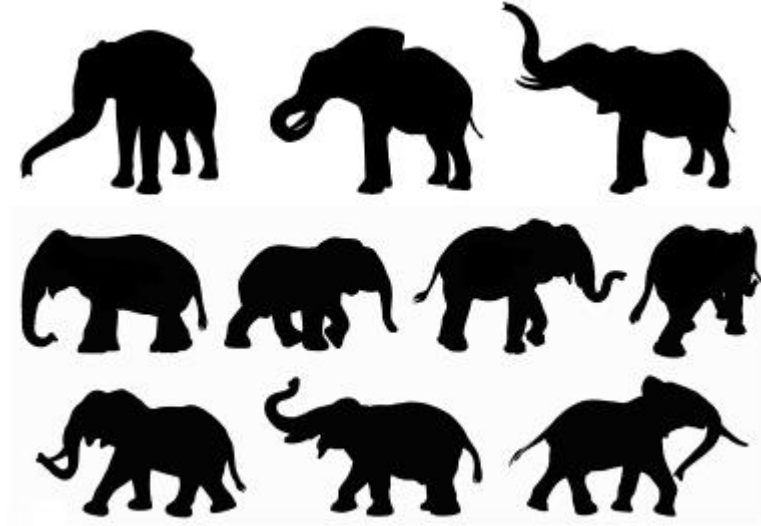


Figure 5.4: Some figures taken from our data set and representing an elephant in different poses.

However, once  $Y^S$  has been found, we calculate the affine transformation that superimposes one shape to the other. Subsequently, being  $Z$  and  $Y$  superimposed, we can compute the discrete Fréchet distance between them. Having aligned the two starting shapes, it is easy to compute this distance. First of all circularly shift  $Y$  so that  $Y_1 = Y_1^S$ , then let  $\bar{Z} = [ZZ_p]$  and  $\bar{Y}^S = [Y^S Y_q]$ . Let  $I_j$  be the set of point of  $Y$  included by  $\bar{Y}_j^S$  and  $\bar{Y}_{j+1}^S$ , and let

$$\partial(I_j) = \max\left(\max_{p \in I_j} \left(d\left(p, \bar{Z}_j\right)\right), \max_{p \in I_j} \left(d\left(p, \bar{Z}_{j+1}\right)\right)\right)$$

then we have that

$$d_{dF} = \max \left( \max_{i=1 \dots p} \left( d \left( Y_i^S, Z_i \right) \right), \max_{j=1 \dots p} \left( \partial(I_j) \right) \right)$$

Since we have some information about the contour of configuration points, we can use it for deriving a new notion of distance. To obtain this result first we have to define the distance between critical points. Given two critical points  $C_i^{\alpha_i}$  and  $C_j^{\alpha_j}$ , the distance between them is defined as

$$d_C \left( C_i^{\alpha_i}, C_j^{\alpha_j} \right) = \frac{1}{4} \left( |D_i - D_j| \bmod 4 \right) \cdot \frac{1}{8} \left( |\alpha_i - \alpha_j| \bmod 8 \right)$$

where

$$D_k = \begin{cases} 1 & \text{if } C_k = A \\ 2 & \text{if } C_k = B \\ 3 & \text{if } C_k = C \\ 4 & \text{if } C_k = D \end{cases}$$

Note that  $d_C$  returns value in  $[0, 1]$ . Now we can introduce the topological distance:

**Definition 15: Topological distance**

Let  $Z$  and  $Y$  be the vectors of two configurations, both of cardinality  $k$ . The topological distance between them is

$$d_T(Z, Y) = \sum_{j=1}^k d_C \left( C_j^{\alpha_j}, C_j^{\alpha'_j} \right)$$

where  $C_i^{\alpha_i}$  for  $i = 1, \dots, k$  are the critical points of  $Z$  and  $C_i^{\alpha'_i}$  for  $i = 1, \dots, k$  are the critical points of  $Y$ .

At this point we can use the pair  $\left\langle d_{dF}(Z, Y), d_T(Z, Y^S) \right\rangle$  for classifying the similarity between the configurations. In fact  $d_{dF}$  quantifies how much the whole figure borders are similar and  $d_T$  suggests when there is a similar structure between them. In particular  $d_{dF}(Z, Y) = 0$  when points of both configurations are perfectly superimposed. When the Fréchet distance increases two cases might occur: the shapes are of the same class but differs for some details, or they belong to different classes. Fréchet distance is not sufficient for deciding between this two cases since it can lead to a non significant distance even between objects of the same class. Thus we introduce the following similarity measure (defined topological-Fréchet distance):

$$S_{TF}(Z, Y) = \beta \frac{d_T(Z, Y^S)}{t_T} + (1 - \beta) \frac{d_{dF}(Z, Y)}{t_F}$$

where  $\beta \in [0, 1]$  is a weight balancing the topological and Fréchet components and  $t_T$  and  $t_F$  are empirically derived thresholds equal to the mean value of  $d_T$  and  $d_{dF}$  respectively, that we obtain comparing objects of different classes. When the value of  $S$  is greater than a given threshold  $\pi_s$ , shapes belong to different classes, otherwise they are similar. Note that  $S_{TF}$  is not a distance measure since the triangular inequality does not hold for it.

In chapter 7, we will compare this new approach with the Procrustes distance previously introduced.





## Chapter 6

# Implicit Analysis: evaluating the distance between two general sets of points

### 6.1 Introduction

In chapter 4, we tried to estimate the distance between two objects by integrating information about shape (explicit model) and about color (implicit model). The previous chapter introduced some improvements on the explicit analysis; now we point out our attention on the implicit one. The comparison of two sets of samples of different size with the aim of defining their distance is one of the most studied problems not only in Artificial Intelligence, but more generally in statistical literature. As we have already shown, to obtain this result we supposed the existence of an underlying model represented by a probability density function. Starting from the datasets, we tried to estimate the function. Afterwards we evaluated the distance between the two pdfs and we considered that value as a distance measure between the two starting sets. Obviously there were a lot of problems to solve. First of all we had to choose which function can be used to describe in a meaningful way the sets (we opted for a mixture of Gaussians). Once decided the "family" of the function, we had to estimate the specific parameters suitable for our dataset. To obtain this result for mixtures of Gaussians, there exists a well-known algorithm: Expectation-Maximization. We want to remind the reader that, even if it is very efficient and effective, the EM requires to know in advance the number of components of the mixture. We estimated this parameter in a easy way: starting from the histogram, we cumulated the HS channels to find the peaks which represent our component estimation.

In this chapter we will investigate the theoretical basis which underlies this approach and the use of a mixture of Gaussians model. Moreover we want to improve our methodology to determine the number of components. Specifically we will present a different and more effective way to estimate this parameter, based on the estimation of the number of modes. Mode estimation can easily be performed by using the Mean Shift algorithm (briefly already shown in chapter 3), a convergent iterative gradient descent procedure which returns the modes of a given set. Even if it is possible to demonstrate that sometimes the number of modes differs from the number of components, nevertheless we will show that the results obtained using this approach are very interesting, since this choice produces a negligible error.

## 6.2 The data: our source of information

An inference process wanting to estimate the characteristics of an unknown model can rely only on the observable state, and a state is observable only through the data. Every cognitive process, exploiting interaction with the real world, must work with data and their manipulation. The data is all we have and also most of what we need. When we observe the world, we observe data. For example, *computer vision* deals with digital images: each color image is simply a matrix of integer triples in the range  $[0,255]$ . In the same way, considering a physical scenario, if we want to study the evolution of the state of a gas, it can be described by the values of its state variables taken in different instants. Moreover, by analyzing the performance of our Engineering Department of the University "La Sapienza", we can study the value of variables, such as the number of new students, the average students' age, the number of students who find a job within the first year after graduation, and so on. The scenario does not matter: to gather information we need techniques that allow us to extract all the information hidden in the data.

Indeed all the information is in the data, we have only to find it. But the information is spread: it lies not only in the values of variables, but, above all, in the relations between them. This idea is usually misunderstood: even if we make complex transformations on our data, we cannot add anything to the informative content, we can only make the information more clear and readable. Extracting information from data means to find relations, patterns, trends and so on to effectively understand, as stated in (131), "what data says". Patterns are regularities; they pave the way to interpretation. They allow us to learn, to compare, to generalize: patterns allow us to know. Every day human beings can recognize shapes, objects, phonemes, words. We are able to understand whether a specific sound belongs to our language, even if we have never heard it before. We are able to recognize a human face even if it has been

transformed. We know (and recognize) because we detect (and recognize) patterns. Patterns are what we have to look for over our data (chapter 2 gives a brief overview about the most popular pattern recognition techniques).

Vast amounts of data are being generated in many fields of science and industry. But what kind of data do we have to deal with? We have huge (sometimes unlimited) sets of values taken from measurements. Some are from quantitative measurements, where it is possible to state an order; others are from qualitative measurements assuming values in a finite set (see, for instance, the famous discrimination example due to Fisher in (97) on the species of Iris). The values are a subset (often very limited) of the whole information, but they are the only subset we can rely on. Recalling the previous example of a digital still image, the triples are just a sampling of the real image view. The color values are taken and evaluated only on the pixels, that is, in a discrete set of equidistant points of the observed scene. And also the intensity of each color is sampled: the light intensity (analogical in its nature) is mapped on a limited range of integers between 0 and 255. A digital image always represents a sampling in space and a sampling in a range of values. Analogously the state of a gas is described by values of variables that are taken in discrete instants and that are bound by the precision of the measurement instrument. Our data are always a sampling of the real world.

### 6.3 Sampling and fitting

Sampling is a process through which we obtain a discrete and limited description of a real phenomenon. From a statistical point of view, for instance, sampling a probability density function  $f(x)$  gives us a set of points in  $\mathbb{R}^n$  distributed over the space according to  $f(x)$  itself. Sampling is generally quite simple (under certain assumptions). But what happens if we try to do the reverse process? Estimating a function, given a set of samples, is instead a hard problem. We have no assurance of finding always a solution and, even when it is possible, the function found could not be unique. It is the same scenario of fitting.

If we make different assumptions on the class the function belongs to, on the quantities to minimize, on the relations among the points, we will obtain completely different results. But to make assumptions means to have more knowledge. This fact is not surprising if we consider (as stated before) that sampling is a very limited representation of the real world. While augmenting the number of points, the range of different possibilities decreases: in fact, after receiving more information on the world, it is possible to cut off some supposed models.

## 6.4 Comparing two datasets

Obviously the considerations that we present are feasible for every kind of data, not exclusively for digital images. For this reason the classification approach that we introduce (including the commentaries) has to be considered, as a general perspective, not limited to the Computer Vision field. But what does "classification" mean? As we have seen before, in a classical supervised approach we have a set of classes  $\{C_1, C_2, \dots, C_K\}$  and we have also a training set which contains elements associated to their own label. In an object recognition task, this means that, for every object met, the system has to decide the class (among the classes it knows) to which the object belongs. However, as said before, the need of a repository could be extremely limiting, especially in the scenarios of our interest. In fact, usually we do not know in advance the kinds of objects the system will have to deal with; so it can be more useful not to define a set of classes for the classification, but simply ask the system to recognize possible similarities between the objects. This means that the system has to decide if two objects (represented by two general sets of  $d$ -dimensional points) are similar and it has to evaluate their degree of similarity, without any further "semantic" consideration. In other words we want to compare two entities, that is to evaluate whether two sets of general  $d$ -dimensional points are different or not, whether they belong to the same category (even if we do not know exactly which is the category) and how much distant they are. We want to know whether and how much they are similar.

But what does "similarity" mean? Similarity is a way to evaluate the distance between two entities with the aim of understanding whether the two entities are different instances of the same class or whether they are completely distinct. To express mathematically this idea, we need a distance measure. It is simple to understand that the two sets of points cannot be compared directly. First of all, as seen in the Procrustes analysis (see section 4.3.6), some methodologies require that the two sets have the same size. And, usually, this condition does not hold. But also when the same size is not required, even if we normalize the points (and so we make them analytically comparable), most of the information can be completely hidden and it is necessary to extract it. Directly comparing two sets of points is the same as comparing two closed boxes without knowing anything about their content.

We supposed the existence of an underlying hidden source for each dataset. Such source can describe the data, being a compact abstraction of the set of samples. Obviously now we have to consider two different sides: the possibility of obtaining a complete description of the source with all the relevant characteristics, and the proof that the source could be a synthetic description of the whole set. We need a very flexible structure, which can be described in an implicit way using few parameters. But it is still not enough: we also have to determine a method to compare the sources

and to evaluate the distances between them. The measure obtained could be extended to the two starting sets and considered as a distance between them. We made different hypothesis about the underlying sources, but for the characteristics of the set of values, probability density functions and more specifically mixture models, represent the best solution.

## 6.5 Searching a pdf

The search of the underlying source (i.e. the underlying model) is then reduced to the search of a probability density function. To estimate a pdf, starting from a given set of samples, classically we can adopt two different approaches: parametric and non-parametric. This latter relies exclusively on data without any additional hypothesis about the structure of the function; the model will be very close to the set of points under analysis and the evaluated pdf can give a great emphasis also to little variations inside the set. The parametric approach, instead, depends on some *a priori* considerations about the final shape of the function and about the characteristics of the process.

In our scenario we consider the set of points as a set of independent identically-distributed values, taken from a continue random variable and we try to define a pdf  $f(x)$ , able to describe its distribution. Obviously the pdf allows us to know the statistical characteristics (as mean and variance) of the set, but also it gives us information about which areas of the space are dense and which are sparse and allow us to calculate the probability that the variable will take on values in a certain interval. Thus the pdf is very useful: it completely characterizes the behavior of the variable. Let's give a look to the classical approaches for pdf estimation.

### 6.5.1 Non-parametric approach

The non-parametric estimate is strictly connected to the data and tries to capture as much information as possible about them. It tries to solve the estimation problem without assuming that  $f(x)$  has some known functional form. This is obviously in contrast with parametric estimation, where the density is assumed to come from a given family, and the parameters are then estimated by various statistical methods. Early contributors to the theory of non-parametric estimation include N. V. Smirnov (265), M. Rosenblatt (242), E. Parzen (221), N. N. Chenston (58). Extensive descriptions of various approaches to non-parametric estimation, along with a comprehensive bibliography, can be found in (264) and (209), while results of experimental comparison of some widely used methods appear in (141) and (293). We address the

interested reader to those works.

Histogram is a first common example of a non-parametric density estimation. The construction of a histogram is fairly simple. We have a random sample  $X_1, X_2, \dots, X_n$  from some unknown continuous distribution. According to (129) we can define the following steps:

- select an origin  $x_0$  and divide the real line into bins of bin-width  $h$ :

$$B_j = [x_0 + (j - 1)h, x_0 + jh], j \in Z$$

- count how many observations fall into each bin. Denote the number of observations that fall into bin  $j$  by  $n_j$ ;
- for each bin divide the frequency count by the sample size  $n$  (to convert them into relative frequencies, the sample analog of probabilities), and by the bin-width  $h$  (to make sure that the area under the histogram is equal to one);
- plot the histogram by erecting a bar over each bin with height  $f_j$  and width  $h$ .

More formally the histogram is given by:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n \sum_j I(X_i \in B_j) I(x \in B_j)$$

where

$$I(X_i \in B_j) = 1 \Leftrightarrow X_i \in B_j$$

Figure 6.1 (taken from (27), pag. 121 fig. 2.24) mostra un esempio di uso degli istogrammi per fittare una funzione al variare della dimensione del bin. Although the use of histograms gives us a first non-parametric pdf estimation,  $(\hat{f}_h(x))$ , (129) shows how it is possible to make some improvements by using a kernel method estimation (see figure 6.2). In fact the histogram retains some undesirable properties:

- the histogram assigns the same estimate  $f$  to all the  $x$ s set in a bin. This seems to be overly restrictive;
- the histogram is not a continuous function, but it has gaps at the boundaries of the bins. It is not differentiable at the gaps and has zero derivative elsewhere. This leads to the ragged appearance of the histogram which is especially undesirable if we want to estimate a smooth, continuous pdf.

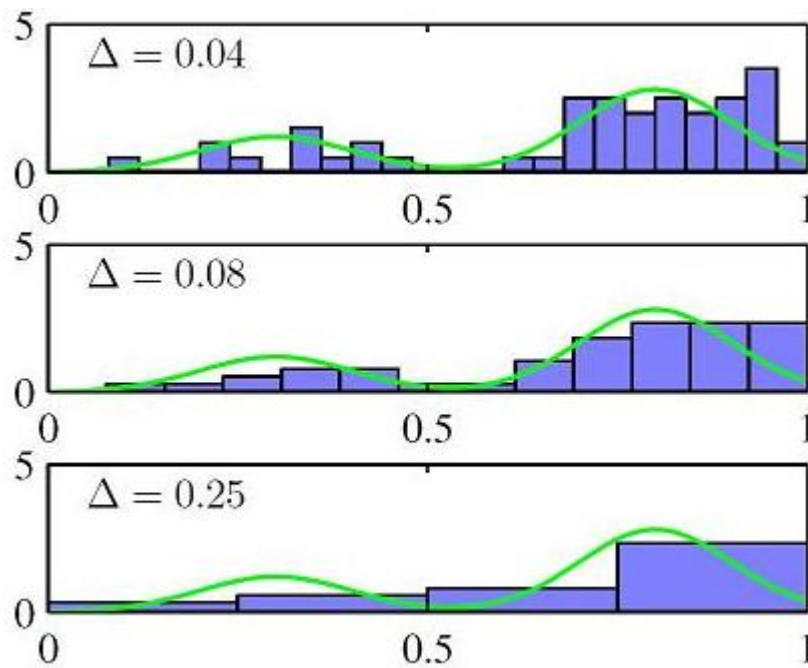


Figure 6.1: Histogram approach to density estimation with different dimension of the bin (taken from (27), pag. 121 fig. 2.24).

We recall that we use histograms in our first attempt to estimate the number of components for the mixtures (see chapter 4).

The research in the area of non-parametric estimation has grown exponentially. Various methods have been proposed for non-parametric density estimation in mathematical statistics, such as the kernels (221; 21; 299) and the orthogonal series methods (167; 251). But we want also to recall the classical ones: penalized maximum likelihood of Good and Gaskins (116), the near neighbour estimators of Loftsgaarden and Quesenberry (181), the spline methods of Wahba (296) or the histogram type estimator of Van Ryzin (289). The kernel method has been extensively studied, and it is probably nowadays the most popular scheme in practical applications. In this method the value of the density at the point  $x$  is estimated as

$$f_h(x) = \frac{1}{nh} \sum_i K\left(\frac{x - X_i}{h}\right)$$

where  $K(u)$  is a kernel function chosen from different alternative functions, which works as a weight function returning in a point  $u$  a value which is the weighted sum of the values of the neighbours of  $u$ . Kernel density estimation generalizes the



histogram method, but also gives us a clearer idea of how non-parametric methods work. We evaluate a specific kernel on each point of the set and we define the pdf in that point as the result of the "weighted sum" of the points near  $x$ .

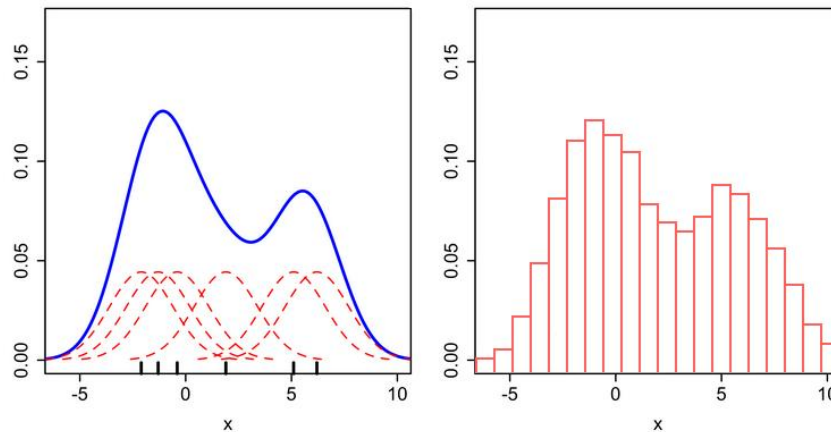


Figure 6.2: The same set of data estimated with two nonparametric approaches: histograms (on the right) and kernels (on the left).

In the orthogonal series method, instead, we have to choose a convenient orthogonal system of functions and write the pdf as the corresponding orthogonal series expansion. In order to estimate  $f$ , we first have to cut the series keeping only a finite number of terms, and then estimate the coefficients of this finite series. This method was studied among others by Schwartz (251) using the system of Hermite functions, by Kronmal and Tarter (167) using the trigonometric system of functions, and by Cencov (207) using a general orthogonal system.

Non-parametric estimate is really close to the set of points and attempts to consider the whole information contained in it. As a drawback, the function obtained is simply a sum of kernels having as many terms as the number of points of the set. We do not obtain a compact representation of data, and so this representation can not help us to make a comparison between sets.

### 6.5.2 Parametric approach

A completely different way is given by the parametric approach: given a set of samples, to estimate the underlying pdf, we start making some *a priori* assumptions on the characteristics of the underlying function. We suppose that the unknown pdf be-

longs to a particular class of functions and we search only the best-fitting instance of this class. In every class, each function can be associated to a limited set of parameters, which represents a complete description of the function itself: if we know the values of the parameters we can evaluate the function in every point. The estimate of the density  $g(x)$  is formed by substitution of the estimate of the parameters:

$$g(x) = g(x|\theta)$$

We use the notation:

$$g_{\theta}(x)$$

where  $\theta$  represents the unknown parameters (one if necessary) that govern the distribution of  $X$ . This is called a parametric model for  $X$ .

As an example, if  $X$  is supposed to have a normal distribution with mean  $\mu$  and variance  $\sigma^2$ , then  $\theta = (\mu, \sigma^2)$  and

$$g_{\theta}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(x-\mu)^2/\sigma^2}$$

There are different methods to estimate the parameters. Most of them involve minimization of residuals. To fit a parametric probability density function, the most common ways are maximum likelihood, matching moments and matching quantiles. We recall (111), which gives an interesting summary of these approaches:

**Maximum Likelihood Methods** The method of maximum likelihood involves the use of a likelihood function that comes from the joint density for a random sample. If  $g(x|\theta)$  is the underlying density, the joint density is just  $\prod_i p(x_i|\theta)$ . The likelihood is a function of the parameter  $\theta$ :

$$L(\theta; x_1, \dots, x_n) = \prod_i p(x_i|\theta)$$

Note the reversal in roles of variables and parameters. The mode of the likelihood (that is, the value of  $\theta$  for which  $L$  attains its maximum value) is the maximum likelihood estimate of  $\theta$  for the given data,  $x$ . The data, which are realizations of the variables in the density function, are considered fixed as well, and the parameters are considered as variables of the optimization problem in maximum likelihood methods.

**Fitting by Matching Moments** Since many of the interesting distributions are uniquely determined by a few of their moments, another method of estimating the density is just to estimate parameters of a given family, so that the population moments (or model moments) match the sample moments. In some distributions, the

parameter estimates derived from matching moments are the same as the maximum likelihood estimates. In general, we would expect the number of moments that can be matched exactly to be the same as the number of parameters in the family of distributions being used.

**Fitting by Matching Quantiles** The moments of distributions with infinite range may exhibit extreme variability. This is particularly true for higher-order moments. For this reason it is sometimes better to fit distributions by matching population quantiles with sample quantiles. In general, we would expect the number of quantiles that can be matched to be exactly the same as the number of parameters in the family of distributions being used. A quantile plot may be useful in assessing the goodness of the fit. The idea seems to be completely different with respect to the non-parametric approach: in this latter we "force" the function to fit the data, while here we "force" the data to fit a predetermined function. If this might seem limiting, on the other hand we obtain a concise description of the whole set and we can represent a large set of points using only a few parameters. Obviously the real disadvantage of this method is that we have to decide in advance what kind of parameters the fitting function will have. In fact the use of a parametric family for estimating an unknown density results in good estimators only if the unknown density is a member of that family. If this is not the case, the density estimator is not robust.

### 6.5.3 Mixture of Gaussians

It is clear that, for our purposes, it would be extremely powerful to describe the whole set of samples using only a few parameters. But adopting a parametric approach, we have the problem of deciding *a priori* the class of the function. Can the parametric approach solve the problem? Can the estimated pdf completely describe our samples? Can the pdf capture in few parameters the interesting features of the set? No, it cannot when we have to deal with so complex and unpredictable structures. In such scenario, ML will give an unsatisfactory approximation. We recall that our samples can be potentially distributed in any way over the space and we could not have any information about the underlying process. On the other hand, the use of a non-parametric approach does not give us the possibility to capture the most important information of the set and to discard the others. We want to combine the flexibility of non-parametric methods and the efficiency in evaluation of parametric ones. This result can be obtained by using a mixture of parametric pdfs. By using a mixture we obtain the powerful and the flexibility of a non-parametric model, and we can easily estimate the parameters of the single component either by some optimization technique, like gradient descent, or by Expectation-Maximization algorithm. Specifically

we opted for a mixture of Gaussians (see section 3.3.3).

The mixture of Gaussians is probably the most studied and widespread model in statistical literature. It seems to combine the advantages of parametric and non-parametric approaches. It combines the simplicity of a Gaussian model and the possibility of describing a pdf using only a few parameters, with the capability to deal with multi-modal distribution, which is usually (and obviously in our scenario) the case of interest. The search of a mixture of Gaussians model can be seen also as a sort of non-parametric technique, if we consider that, as Parzen has shown in (221), with a sufficient number of components, any probability density can be approximated to any degree by a Gaussian mixture.

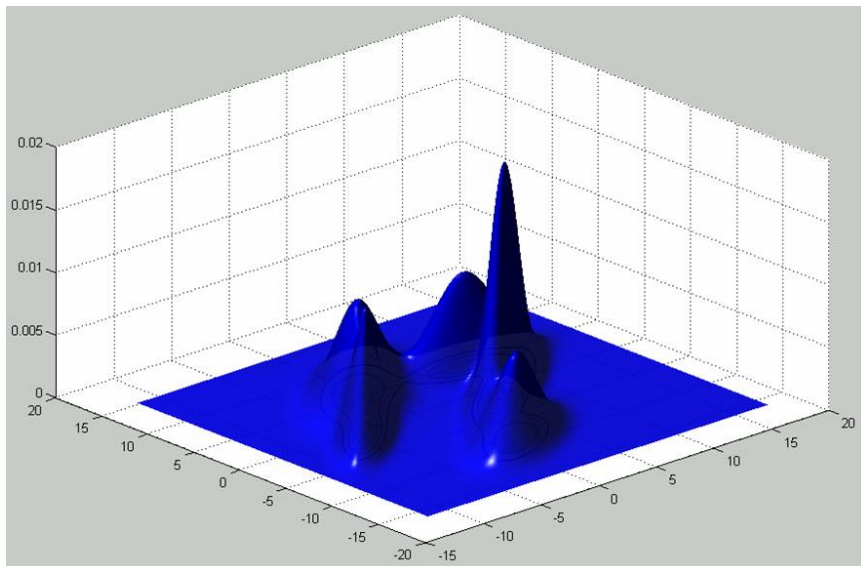


Figure 6.3: An example of a 3-dimensional mixture of Gaussians with 5 components.

This solves the problem of choosing *a priori* the family to which the pdf has to belong. However it is extremely important to note that choosing as approximating function a mixture of Gaussian, does not mean knowing the parameters, but only the class of parameters (namely  $3M - 1$  parameters where  $m$  has to be discovered) that have to be estimated. Therefore this is far different from assuming a known distribution, such as a binomial or Poisson or Gamma (or even Gaussian); the fact that a Gaussian mixture refers to a set of Gaussian distributions has an implication only with respect to the properties of the mixture, but the relation between each sample class and its component has yet to be discovered.

As stated in the same section, we can use the Expectation-Maximization algorithm to estimate the parameters. The drawback is that we have to know in advance (and to pass it as an input parameter to the algorithm) the number  $M$  of components. In chapter 4, we proposed an interesting but not completely satisfactory method to estimate  $M$ . Now we present an improvement and the connected results.

## 6.6 On the underlying model

First of all it is necessary to make some clarifications about the model underlying the set of points. We are looking for a function which can "explain" our samples. The pdf has to be seen simply as a descriptor of the collected points. Indeed we are looking for a function which can characterize our data and extract from it some important features, otherwise unknown. Than it could not have any meaning speaking about a "real" model. The pdf obtained will be for sure an approximation of the dataset. We try to "parameterize" (and thus "simplify") the set and to represent it only through a few parameters able to emphasize the main features and to discard the unimportant and redundant ones. Starting from this point of view, we can look at some classical density estimation problems in an another way. Let us go back again to our scenario: we are looking for a mixture of Gaussians to describe our data. Instead of trying to evaluate how well the chosen function can fit our data, it is of extreme interest to wonder how much information the model can extract from the data, which kind of information it is and, moreover, how this information can help us to compare the original datasets.

Given a dataset of points in  $\mathfrak{R}^n$ , we want to estimate the parameters of the best-fitting mixture of Gaussians. We have presented a powerful methodology to obtain this result; but we recall that the EM requires, as input parameter, the number of components  $M$ . This problem has been largely studied in the literature, but satisfactory solutions are not known nowadays. Even If we already have introduced an empiric approach, based on the HS histogram peaks, in the following sections, we will present a novel methodology. It is very crucial however to highlight that it is not only hard to solve this problem, but it is also very difficult to give an evaluation of the performance of different approaches: that is, starting with a set of experimental data, once estimated the supposed underlying mixture (with an hypothesis on  $M$ ), we will not have a reference function to compare with, nor a specific distance measure. So we cannot obtain an evaluation of the obtained mixture (and consequently of the estimated parameters).

We use the mean shift algorithm to estimate the number of modes of a dataset, and we use this value as the input parameter of the EM algorithm. To cope with the problems presented above, in our experiments we make this assumption: instead of

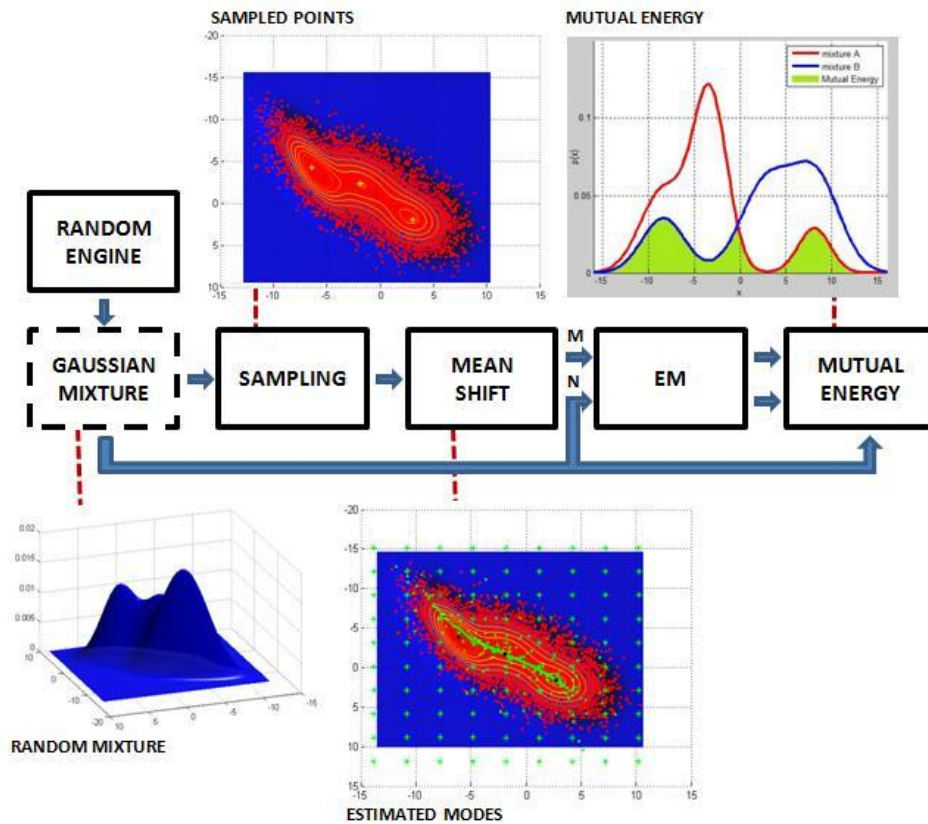


Figure 6.4: A block diagram to summarize the main step of the mixture estimation algorithm.

considering datasets taken from digital images, we start with a completely known probability density function and we sample it to obtain the data; once estimated the pdf (using the number of modes as input parameter of the EM), we can compare the two pdfs (the first one, completely known and the second one estimated) and we can obtain an evaluation of our choice on the number of components. Being the obtained results encouraging, we then integrate the implicit analysis with this approach.

## 6.7 Our approach

According to the block diagram of figure 6.4, we start from a known Mixture of Gaussians (say  $f_K(x)$  with  $N$  components). We sample from it a set of points (say  $P_{f_K}$ ). We use the mean shift algorithm to estimate the number of modes on  $P_{f_K}$  (say

$M$ ). We use the EM algorithm to estimate the best fitting Mixture of Gaussians on  $P_{f_K}$ : firstly by using  $M$  as input parameter, and then by using  $N$  (the real number of components). We obtain two estimated pdfs (say  $f_E$  and  $f_{E_R}$ ). Finally we evaluate how distant the two pdfs ( $f_E$  and  $f_{E_R}$ ) are from the starting function  $f_K$ , and so we obtain an evaluation of the methodology used. In the following we will describe in a more detailed way each single step.

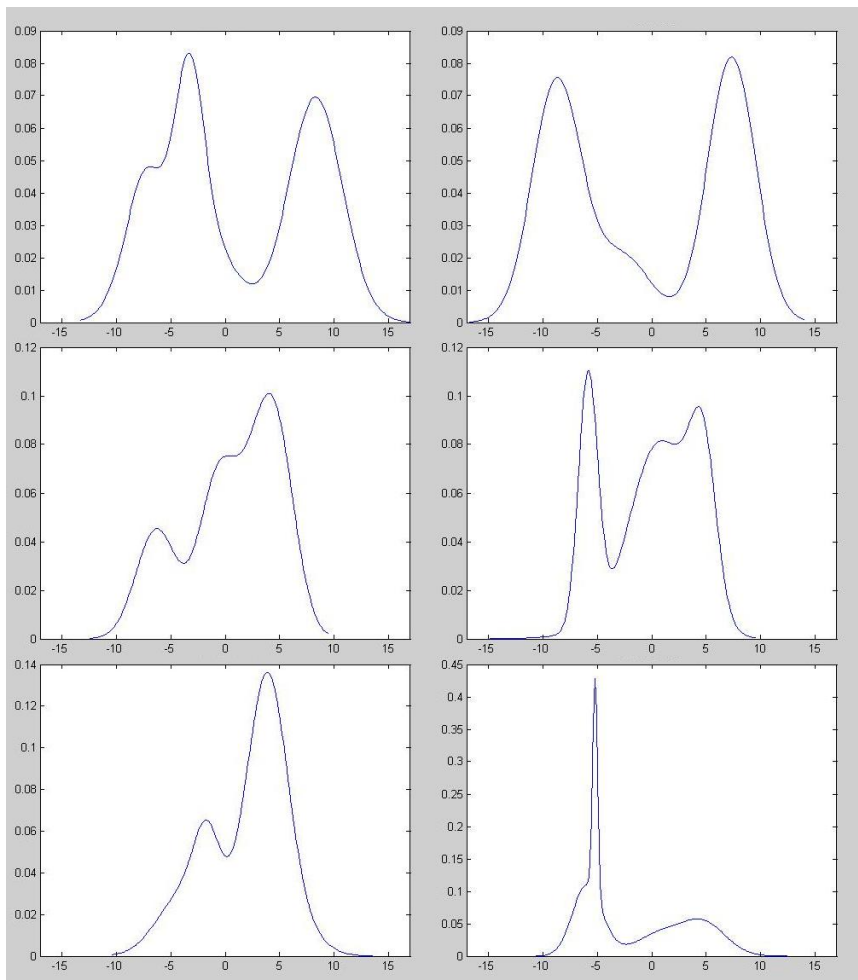


Figure 6.5: Some mixtures of Gaussians randomly generated.

### 6.7.1 Randomly generated mixtures

First of all we defined a set of mixtures. To obtain this result, we simply defined the ranges of each parameter ( $\mu$  and  $\Sigma$ ) and we randomly generated them. The parameters are taken inside the following ranges:  $[-9, +9]$  for mean values ( $\mu$ ) and  $[0, +8]$  for covariance values ( $\Sigma$ ). We worked with  $d$ -dimensional mixtures with a number of components varying from 2 to 6. Due to visualization issue, without any loss of generality, we will present only examples from experiments performed with 2D and 3D mixtures. However in one of the next sections we will show the complete results. Figure 6.5 shows some examples of randomly generated 2D mixtures.

### 6.7.2 Sampling

Now we have to define a procedure to obtain a set of samples for each generated mixture. There exist a lot of sampling techniques in literature. We needed to generate wide sets of samples (20,000 points for every pdf) so we need an efficient approach. We perform a Montecarlo sampling. In figure 6.6 it is possible to observe a 3D mixture with 5 components and the set of sampled points.

We recall that only sampling a known pdf we can have datasets directly connected with the functions and we can consider the original mixtures as "true" models of the sets. Obviously this is the only way to know all the parameters of the underlying model and also to understand how far will be the mixture estimated from the original one: we are able to estimate the mixture only starting from the sampled points and, once obtained, we can compare the results with the well-known starting mixtures. Now we have obtained our starting sets. Obviously this allows us to rely on a powerful technique to evaluate the goodness of the estimation about the number of components. However we still have the problem of defining a methodology to define it.

### 6.7.3 Modes

It is necessary to analyze some aspects about the nature of the model adopted. A mixture of Gaussians is simply a weighted sum of functions. Given a set of points, we will use the expectation-Maximization algorithm to estimate the best-fitting probability density function. To use successfully the EM procedure, it is necessary to know in advance the number of components of the mixture. We tried to solve this problem by using the peaks of the HS histogram; now we will use the number of the modes. The use of the number of modes arises from a simple idea: it is reasonable to hypothesize that each single Gaussian would be set at the center of the densest areas and that we



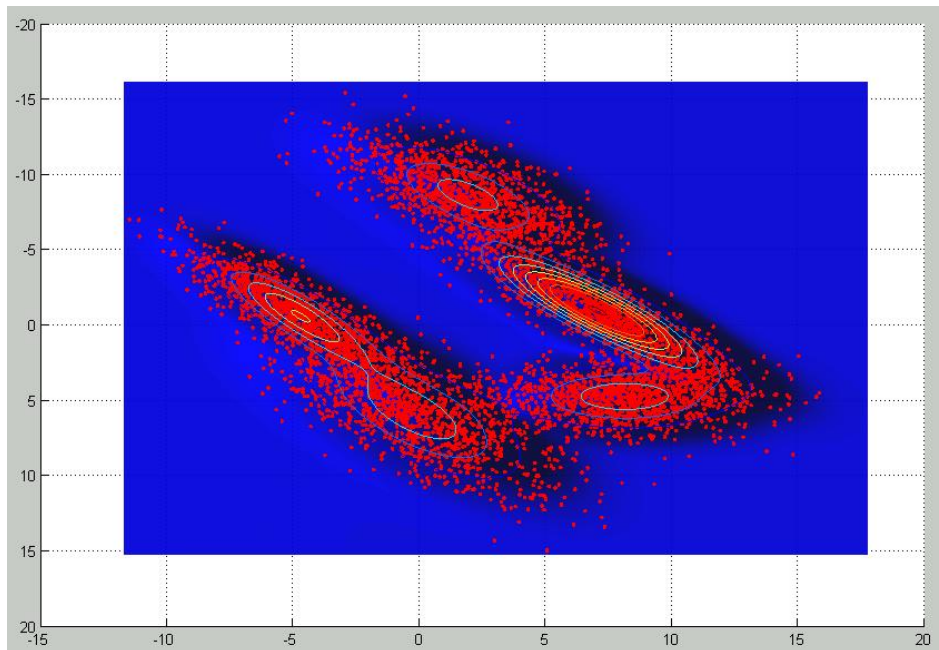


Figure 6.6: A sampling of a 5 components 3D mixture of Gaussians.

have one (and only one) component for each of them. Since that a mode is defined as the value that occurs most frequently in a data set or a probability distribution, it lies exactly in the center of those areas. The number of modes can thus represent a bound for the number of components or, in specific case, they can even coincide. Unfortunately there exist scenarios in which these two numbers have no correlation. In (50) the authors presented the following conjecture:

**Conjecture 1**

Let  $p(x)$ , be a mixture of  $M$   $d$ -variate normal distributions, then  $p(x)$  has  $M$  modes at most, all of which in the convex hull of  $\{\mu_m\}_{m=1}^M$ , if one of the following conditions holds:

1.  $d = 1$  (one-dimensional mixture);
2.  $d \geq 1$  and the covariance matrices are arbitrary but equal:  $\Sigma_m = \Sigma = 1 \dots M$  (homoscedastic mixture);
3.  $d \geq 1$  and the covariance matrices are isotropic:  $\Sigma_m = \sigma_m^2 I_d$  (isotropic mixture).

As the authors showed in (51) those conditions are necessary and several parts of the conjecture hold. Specifically they demonstrated that all modes lie in the convex

hull and that for  $d = 1$ ,  $M$  is upper bounded by the number of modes (for the proof of these assertions we address the interested reader to (51)); as a corollary of this latter theorem, the authors also showed that any 1D projection (marginal or conditional distribution) of any Gaussian mixture in  $d$  dimensions with  $M$  components, has, at most,  $M$  modes.

Obviously we can not make any specific assumption about the structure of our mixtures. On the contrary we recall that the choice of this model (mixture of Gaussian) has derived from the Parzen's assumption that such function can approximate every other pdf and from our complete lack of information about the underlying dataset. And, for sure, we know that there exist scenarios in which there is no correlation between the two values. We will show an image (see figure 6.7) taken from (51) in which there is a mixture with 6 components having different, non-isotropic covariances; this mixture has 9 modes (marked with " $\Delta$ "): 6 of them are coincident with the means  $\mu_m$  (marked with "+") and the other ones are in the overlapping areas of the components. It is interesting to note that some modes are outside the convex hull of the centroids (marked by the thick black line). So the number of modes exceeds the number of components.

At the same time we can define a scenario where the number of modes is less than  $M$ . Consider figure 6.8. There is a 3D Gaussian mixture with 3 components. The figure consists of 3 couples of graphs, where at the top a 3D graph of the mixture is represented, while at the bottom the corresponding profiles projected on the  $xy$  plane is shown. In the bottom the means of the components (yellow dots) and the modes of the functions (red stars) are also highlighted. The 3 couples of images refer to mixtures with the same values of the priors, of the covariances and of 2 of the 3 means. We only perturb the mean of the second component to reduce the Euclidean distance between it and the mean of the first component (from left to right). When the 2 component are close enough, despite there are 3 components, we observe only 2 modes.

Thus, as we have shown, there is no analytic correlation between these two values. In the following we will experimentally demonstrate that, even if the number of modes does not represent the "correct" answer, the approximation obtained by using it, leads to a small error. This methodology seems interesting also because there exists an efficient algorithm to estimate the modes of a dataset: the Mean Shift. We will explain it in the next section.

#### 6.7.4 Mean Shift

The mean shift algorithm has been firstly proposed as a method for cluster analysis (see (106), (264), (57)). A great improvement occurred when it started to be used as a

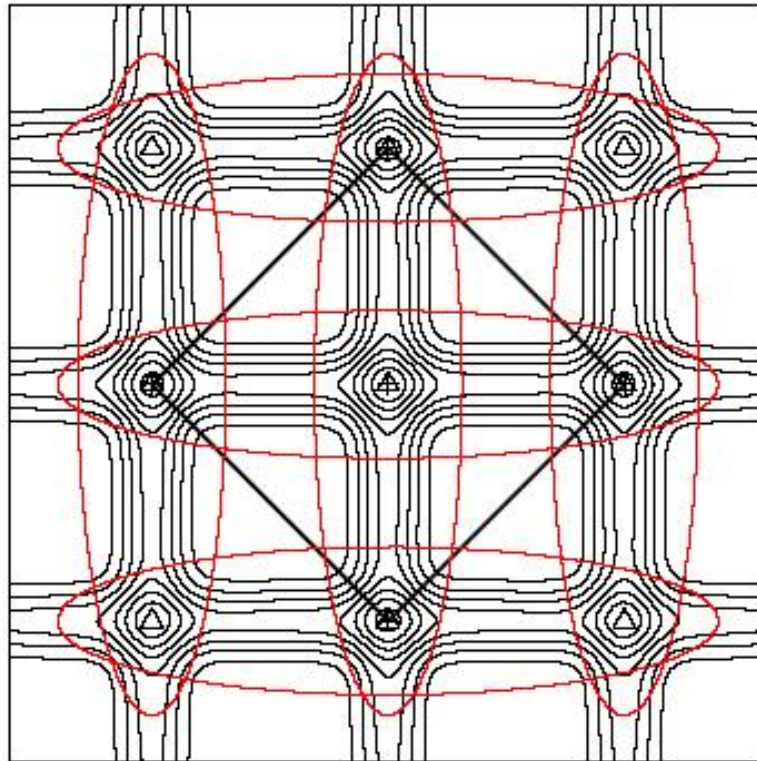


Figure 6.7: Image taken from (51) (fig. 1, pag. 3): an example of mixture of Gaussians where the number of modes (9 - marked by " $\Delta$ ") exceeds the number of components (6 - marked by "+").

gradient ascent. The Mean Shift was introduced by Fukunaga (who is also one of the authors of the peak-climbing algorithm used in chapter 4, (162)) and Hostetler (106). It is in fact a simple nonparametric iterative procedure that shifts each data point to the average of data points in its neighborhood for seeking the mode of a density function represented by a set  $S$  of samples. The procedure uses kernels, as decreasing functions of the distance from a given point  $t$  to a point  $s$  in  $S$ . For every point  $t$  in a given set  $T$ , the sample means of all points in  $S$ , weighted by a kernel at  $t$ , are computed to form a new version of  $T$ . This computation is repeated until convergence. The resulting set  $T$  contains estimates of the modes of the density underlying set  $S$ . Obviously it is very interesting because gives us a powerful iterative procedure that, given a dataset, returns the modes of the underlying pdf, and so we can use this information to estimate the number of component of the mixture of Gaussians. Cheng in (56) revisited mean shift, developing a more general formulation and demonstrating

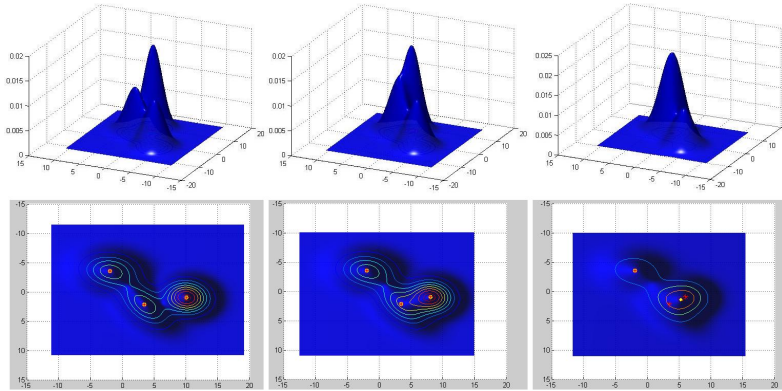


Figure 6.8: A 3D 3-components mixture of Gaussians. In the panels (from left to right) the mean of one of the components changes getting closer to the mean of another component. Upper panels represent the 3d view of the mixtures; lower panels represent the profiles (projected on the  $(x, y)$  axis) of the corresponding mixtures with modes (red stars) and means (yellow dots) highlighted.

its potential uses in clustering and global optimization.

Consider a simple flat kernel  $k(x)$ :

$$K(x) = \begin{cases} 1 & \text{if } \|x\| \leq \lambda \\ 0 & \text{otherwise} \end{cases}$$

The sample mean at  $x \in X$  is

$$m(x) = \frac{\sum_{s \in S} K(s - x) s}{\sum_{s \in S} K(s - x)}$$

The difference  $m(x) - x$  is called mean shift in (106). The repeated movement of data points to the sample means is called the mean shift algorithm (see (106), (264)). In each iteration of the algorithm,  $s \leftarrow m(s)$  is performed for all  $s \in S$  simultaneously. Cheng in (56) defines the algorithm as in the following:

**Definition 16: Mean Shift procedure**

Let  $S \subset X$  be a finite set (the "data" or "sample"). Let  $K$  be a kernel and  $w : S \mapsto (0, \infty)$  a weight function. The sample mean with kernel  $K$  at  $x \in X$  is defined as

$$m(x) = \frac{\sum_{s \in S} K(s - x) w(s) s}{\sum_{s \in S} K(s - x) w(s)}$$

Let  $T \subset X$  be a finite set (the "cluster centers"). The evolution of  $T$  in the form of iterations  $T \leftarrow m(T)$  with  $m(T) = \{m(t); t \in T\}$  is called a mean shift algorithm. For each  $t \in T$ , there is a sequence  $t, m(t), m(m(t)), \dots$ , that is called the trajectory of  $t$ . The weight  $w(s)$  can be either fixed throughout the process or re-evaluated after each iteration. It may also be a function of the current  $T$ . The algorithm halts when it reaches a fixed point ( $m(T) = T$ ).

Even if in the original formulation the mean shift procedure was strictly connected to the idea of kernels, Fashing and Tomasi in (92) defined mean shift in terms of a profile. We recall some definitions from their paper:

**Definition 17: Profile**

A profile  $k$  is a piecewise continuous, monotonically non-increasing function from a non-negative real to a non-negative real, such that the definite integral  $\int_0^{\infty} k(r) dr < \infty$ .

**Definition 18: Kernel**

A kernel  $K$  is a function from a vector  $x$  in the  $n$ -dimensional real Euclidean space,  $X$ , to a nonnegative real, such that  $K(x) = k(\|x\|^2)$  for some profile  $k$ .

**Definition 19: Mean Shift profile procedure**

Let  $X$  be an  $n$ -dimensional real Euclidean space and  $S$  a set of sample vectors in  $X$ . Let  $w$  be a weight function from a vector in  $X$  to a non-negative real. Let the sample mean  $m$  with profile  $k$  at  $x \in X$  be defined such that

$$m(x) = \frac{\sum_{s \in S} k(\|s - x\|^2) w(s) s}{\sum_{s \in S} k(\|s - x\|^2) w(s)}$$

Let  $M(T) = \{m(t) : t \in T\}$ . One iteration of mean shift is given by  $T \leftarrow M(T)$ . The full mean shift procedure iterates until it finds a fixed point  $T = M(T)$ .

Cheng also introduced the concept of shadow kernel:

**Definition 20: Shadow Kernel**

Kernel  $H$  is said to be a shadow of kernel  $K$ , if the mean shift using  $K$ ,

$$m(x) - x = \frac{\sum_{s \in S} K(s - x) w(s) s}{\sum_{s \in S} K(s - x) w(s)}$$

Is in the gradient direction at  $x$  of the density estimate using  $H$ ,

$$q(x) = \sum_{s \in S} H(s - x) w(s)$$

And he also demonstrated that kernel  $H$  is a shadow of kernel  $K$  if and only if their profiles,  $h$  and  $k$ , satisfy

$$h(r) = f(r) + c \int_r^\infty k(t) dt,$$

where  $c > 0$  is a constant and  $f$  is a piecewise constant function. It is possible to choose several kernels for the algorithm. According to (64), we use the Epanechnikov kernel:

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1 - x^T x) & , \text{ if } x^T x < 1 \\ 0 & \text{ otherwise} \end{cases}$$

where  $c_d$  is the volume of the unit  $d$ -dimensional sphere. It is very important to note that  $K_E$  is the shadow of the flat kernel. Comaniciu and Meer demonstrated that if  $f_E = \{f_k(y_k, K_E)\}_{k=1,2,\dots}$  is the sequence of density estimates obtained using  $K_E$  and computed in the points defined by the successive locations of the mean shift algorithm with uniform kernel, then the convergence of the sequence is guaranteed. Mean shift procedure has met great popularity in the computer vision community. Applications range from image segmentation and discontinuity-preserving smoothing (64), (65) to higher level tasks like appearance-based clustering (238), (239) and blob tracking (63).

We have a set of points sampled from a completely known mixture of Gaussians. Starting from this set, we will use Mean Shift algorithm to estimate the number of modes of the underlying pdf. In our experiments we tried different kernels (flat, biweight, Epanechnikov and Gaussian); the results presented are obtained with the Epanechnikov kernel. In the use of mean shift it is necessary to tune two parameters: the size of the window over which to calculate the mean in each single step (we define hyper-cubic areas con edge  $2h$ , so the volume in  $d$  dimensions is:  $(2h)^d$ ) and the threshold value which bound the kernel (say  $\lambda$ ). In literature there is not an exhaustive analysis of these values, so we perform some experiments to define  $\lambda$  and  $h$ . We generated 1,000 random mixtures and we perform mean shift analysis with 7 different values of  $\lambda$  and  $h$ :  $\{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5\}$ . We than consider the assignments which leads to a number of modes which is equal or different for 1 from the real number of components. As you can see in figure 6.9, the best results are obtained with  $\lambda = 1.5$  and  $h = 3.5$ .

In figure 6.10 instead we show an example of the complete procedure. In this case we have a 3D mixture with 3 components. The image represents a 2D projection of the mixture (with the profiles of the components). The red points represent

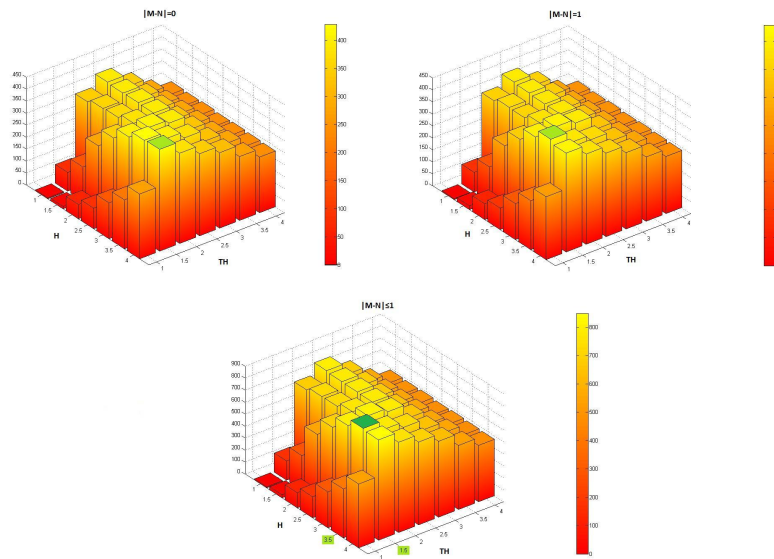


Figure 6.9: On the  $x$  and  $y$  axis are represented, respectively, the different values of  $\lambda$  and  $h$ . For each couple of values, the histograms indicate how many occurrences of the couple has returned an estimated modes number equal to the components number (upper left), how many occurrences of the couple has returned an estimated modes number differ for 1 unit from the components number (upper right) and the sum of the two previous values (lower).

the sampling of the mixture. The green stars represent the starting points and the green points represent the trajectories of the mean shift algorithm. The yellow stars represent the estimated modes. As you can see, there are some modes very close to each other in the densest areas; at the same time there are some modes also in areas with very few points. To obtain a more meaningful result, we perform a successive thresholding step which discards some hypothetical but not interesting modes.

### 6.7.5 Distance measure

Once the number of modes is to hand, it is possible to initialize the EM for the estimation of the other parameters of the mixture. As we said before, to evaluate the quality of our choice, we need a meaningful procedure able to calculate the distance between two pdfs. In this section we will derive our distance measure, which will be used both to estimate the distance between the starting pdf and the estimated one, but also, in the general procedure, to evaluate the similarity between 2 implicit models.



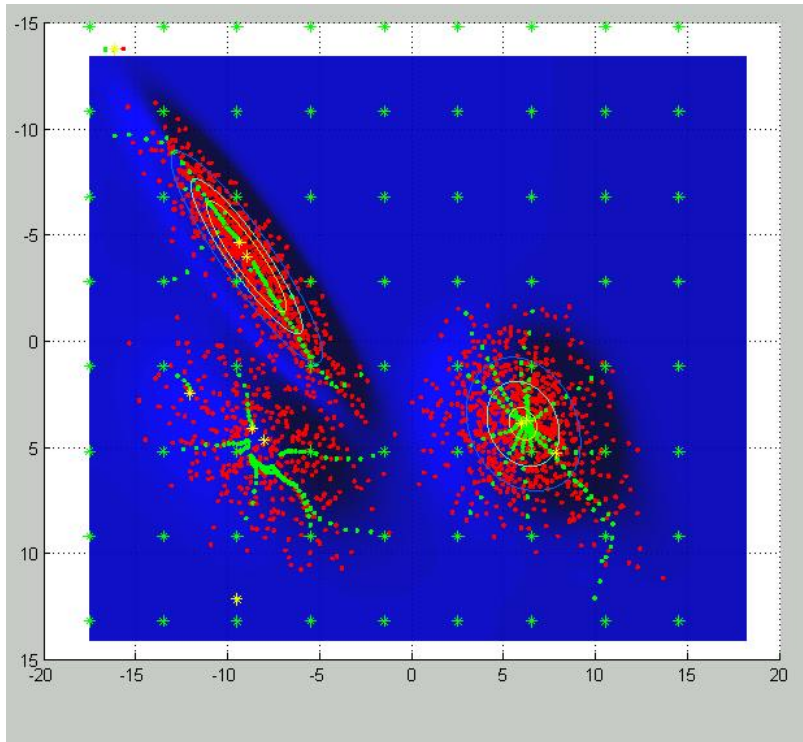


Figure 6.10: An example of mean shift algorithm: the red points represent the sampling of the mixture (superimposed), the green stars the starting point of the algorithm, the green points the trajectories' steps of the algorithm and the yellow stars the estimated modes (before thresholding).

In the literature many different distance measures have been introduced. Now we present a brief list of their most interesting properties:

- nonnegativity:

$$d(p, q) \geq 0$$

- reflexivity:

$$d(p, p) = 0$$

- isolation:

$$d(p, q) = 0 \Rightarrow p = q$$

- symmetry:

$$d(p, q) = d(q, p)$$



- triangle inequality:

$$d(p, q) + d(q, r) \geq d(p, r)$$

- relaxed triangle inequality:

$$d(p, q) + d(q, r) \geq cd(p, r)$$

$$c < 1$$

- finiteness:

$$d(p, q) < \infty$$

- boundness:

$$d(p, q) \leq c$$

$$c \in \mathfrak{R}$$

- semiboundness:

$$d(p, q) \geq d(p, p)$$

But what do we exactly expect from a distance measure? Above all we want it to be able to determine the similarity between the two analyzed elements. Similarity is an extremely important and always widely used concept; but, at the same time, it is also complex to formalize. In the last decades many different definitions of similarity have been proposed, and most of all were exactly tied for a particular scenario. Lin in (176) suggests a definition of similarity which tries to achieve two extremely important goals: universality and theoretical justification. He made the following 6 assumptions, that we consider a good starting point:

**Assumption 1** The commonality between  $A$  and  $B$  is measured by

$$I(\text{common}(A, B))$$

where  $\text{common}(A, B)$  is a proposition that states the commonalities between  $A$  and  $B$  and  $I(s)$  is the amount of information contained in a proposition  $s$ .

**Assumption 2** The differences between  $A$  and  $B$  is measured by

$$I(\text{description}(A, B)) - I(\text{common}(A, B))$$

where  $\text{description}(A, B)$  is a proposition that describes what  $A$  and  $B$  are.

**Assumption 3** The similarity between  $A$  and  $B$ ,  $\text{sim}(A, B)$  is a function of their commonalities and differences. That is

$$\text{sim}(A, B) = f(I(\text{common}(A, B)), I(\text{description}(A, B)))$$

the domain of  $f$  is  $\{(x, y) \mid x \geq 0, y > 0, y \geq x\}$

**Assumption 4** The similarity between a pair of identical objects is 1:

$$\forall x > 0, f(x, x) = 1$$

**Assumption 5** Where there is no commonalities between  $A$  and  $B$ , their similarity is 0:

$$\forall y > 0, f(0, y) = 0$$

**Assumption 6** The overall similarity of two objects is a weighted average of their similarities computed from different perspectives. The weights are the amounts of information in the description:

$$\begin{aligned} \forall x_1 \geq y_1, x_2 \geq y_2 : f(x_1 + x_2, y_1 + y_2) &= \\ &= \frac{y_1}{y_1 + y_2} f(x_1, y_1) + \frac{y_2}{y_1 + y_2} f(x_2, y_2) \end{aligned}$$

Very interesting observations (strictly connected to the scenario of our interest) can also be taken from the famous article by Ali and Silvey ((4)); they present a general class of coefficients to estimate the divergence of one distribution from another. They introduce four properties that seem reasonable to demand of a real coefficient  $d(P_1, P_2)$  whether "this coefficient is to reflect the facts that some distributions may be closer together than others and that it may be more difficult to distinguish between the distributions of one pair than between those of another". We list them in the following:

**First property** The coefficient  $d(P_1, P_2)$  should be defined for all pairs of measures  $P_1$  and  $P_2$  on the same sample space.

**Second property** Suppose that  $y = t(x)$  is a measurable transformation from  $(H, F)$  onto a measure space  $(Y, G)$ . Then we should have

$$d(P_1, P_2) \geq d(P_1 t^{-1}, P_2 t^{-1})$$

Here  $P_i t^{-1}$  denotes the induced measure on  $Y$  corresponding to  $P_i$ .

**Third property**  $d(P_1, P_2)$  should take its minimum value when  $P_1 = P_2$  and its maximum value when  $P_1$  is orthogonal to  $P_2$ .

**Fourth property** Let  $\theta$  be a real parameter and let  $\{P_\theta = \{\theta \in (a, b)\}\}$  be a family of equivalent (mutually absolutely continuous) distributions on the real line such that of the family of densities  $p_\theta(x)$  with respect to a fixed measure  $\mu$  has monotone likelihood ratio in  $x$ . Then if  $a < \theta_1 < \theta_2 < \theta_3 < b$ , we should have

$$d(P_{\theta_1}, P_{\theta_2}) \leq d(P_{\theta_1}, P_{\theta_3})$$

### 6.7.6 Our distance measure: Normalized Mutual Energy

According to the previous definitions to calculate the distance between two mixtures of Gaussians, we use the Normalized Mutual Energy (see section 4.4.4). Different observations lead us to this specific choice. Consider two  $d$ -dimensional functions:

$$f : \mathbb{R}^d \rightarrow \mathbb{R}$$

$$g : \mathbb{R}^d \rightarrow \mathbb{R}$$

From a theoretical point of view, it is worth noting that Normalized Mutual Energy returns a value which represents the  $d$ -dimensional hyper-volume on which  $f$  and  $g$  overlap. So it gives us an idea of how distant the two functions are. In figure 6.11 you can see 2 randomly generated 2D mixtures with 3 components and the corresponding Normalized Mutual Energy (below).

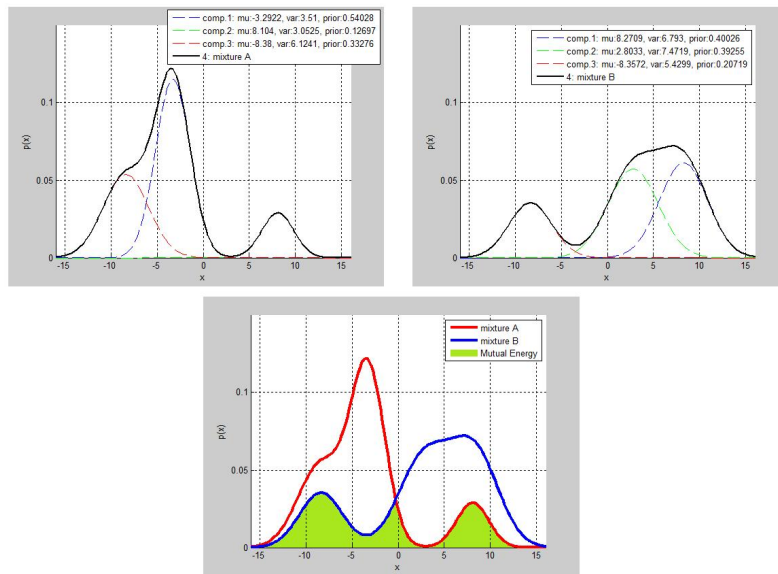


Figure 6.11: Two 2D mixtures with 3 components (top right and left) and the corresponding mutual energy (bottom, in green).

As we stated previously, this measure is very interesting also from a computational point of view. In fact, by using the method presented by Lyu in (183), it is possible to calculate this distance in a closed form. Specifically having two functions  $p$  with  $N_1$  components and  $p'$  with  $N_2$  components, the Normalized Mutual Energy ( $M_N$ ) has the following form:

$$M_N(p, p') = \frac{K_{EL}(p, p')}{K_{EL}(p, p) K_{EL}(p', p')}$$

where  $K_{EL}$  is the Expected Likelihood Kernel, which we recall can be defined in a closed form as:

$$K_{EL}(p, p') = (2\pi)^{-\frac{d}{2}} \alpha^T \Gamma \beta$$

For the description of  $\alpha$ ,  $\beta$  and  $\Gamma$  we address the reader to section 4.4.4. Let's give a look to the properties which hold for  $M_N$ :

- non-negativity:  $M_N(p, q) \geq 0$
- reflexivity:  $M_N(p, p) = 0$
- isolation:  $M_N(p, q) = 0 \Rightarrow p = q$
- symmetry:  $M_N(p, q) = M_N(q, p)$
- finiteness:  $M_N(p, q) < \inf$
- boundness:  $M_N(p, q) \leq c, c \in \mathfrak{R}$

Moreover it is interesting to highlight that the Mutual Energy is also an upper bound for the Kullback-Leibler distance. In Appendix A we present some interesting relationships between the most used pdfs distance measures.

### 6.7.7 Experimental Results

We recall the block-diagram of figure 6.4. Firstly a random mixture of Gaussians is generated. The mixture is sampled and the samples obtained are used as starting set. The samples became the input of a mean shift algorithm, which is used to determine the number of the modes of the pdf underlying the samples. This value ( $M$  in the block diagram) is used as input parameter of the Expectation-Maximization. Also the real number of components ( $N$  in the diagram) is passed as input parameter to the EM. It is important to note that we know the real number of components only because we start from a specific underlying mixture. Obviously, in real scenarios, we do not know anything about  $N$ . The EM therefore returns two different estimated mixtures (respectively  $mixt_M$  and  $mixt_N$ ).

We use the Normalized Mutual Energy to compare the distances between these two mixtures and the original one: specifically we indicate with  $d_N$  the distance between the original mixture and the one estimated by using  $N$  and with  $d_M$  the

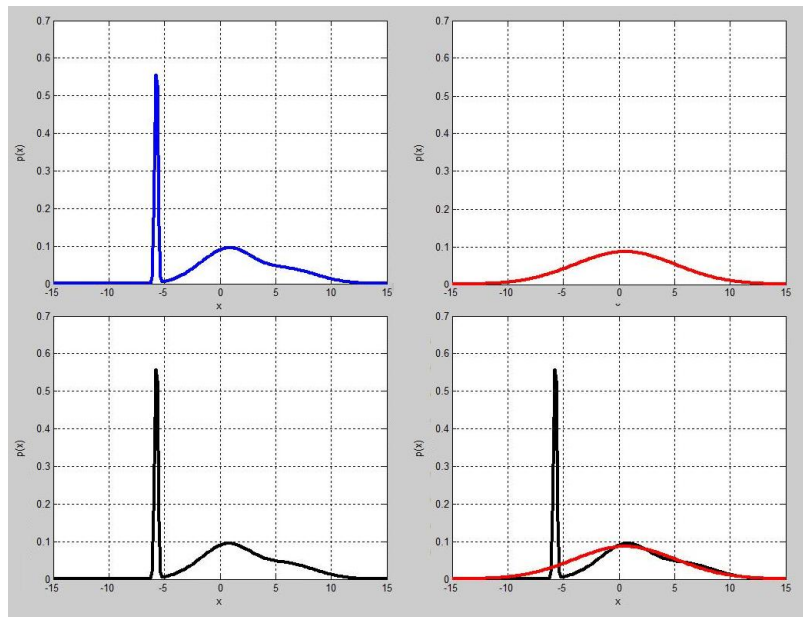


Figure 6.12: Experiments with 1d mixtures:  $mixt_N$  is closer to the original mixture than  $mixt_M$ . The original mixture (in black, bottom left),  $mixt_N$  (in blue, top left),  $mixt_M$  (in red, top right), a graph containing the original mixture and  $mixt_M$  plotted together (in black and red respectively, bottom right).

distance between the original mixture and the one estimated by using  $M$ . These distances are in the range  $[0, 1]$ , where 1 represents a perfect matching, while 0 represents two mixtures without any overlapping areas. We use the difference between  $d_N$  and  $d_M$  (indicated with  $d_R$ ) to evaluate the quality of the approximation obtained with the mean shift with respect to the mixture estimated by using the real number of components.  $d_M$  also returns an evaluation about the absolute quality of the mixtures obtained. In the following we present our experimental results. We worked with mixtures with different dimensions. We generated over 10,000 for each scenario.

### 1D mixtures

The results obtained working with 1D mixtures must be considered a special case: it is known that for one-dimensional mixtures the number of modes never exceeds the number of components (see (51)). We generated 11,768 random mixtures. In 7.25% of the cases, the number of modes estimated by the mean shift is equal to the real number of components, so the two estimated mixtures are equivalent and the  $d_R$

is equal to 0. In 82.25% of cases  $mixt_M$  has a greater distance from the original mixture than  $mixt_N$ . But the error is negligible:

- in 50.75% of cases,  $|d_N - d_M| \leq 0.01$ ;
- in 21.50% of cases,  $0.01 < |d_N - d_M| \leq 0.05$ ;
- in 5.25% of cases,  $0.05 < |d_N - d_M| \leq 0.1$ ;
- in 4.75% of cases,  $0.1 < |d_N - d_M| \leq 1$ ;

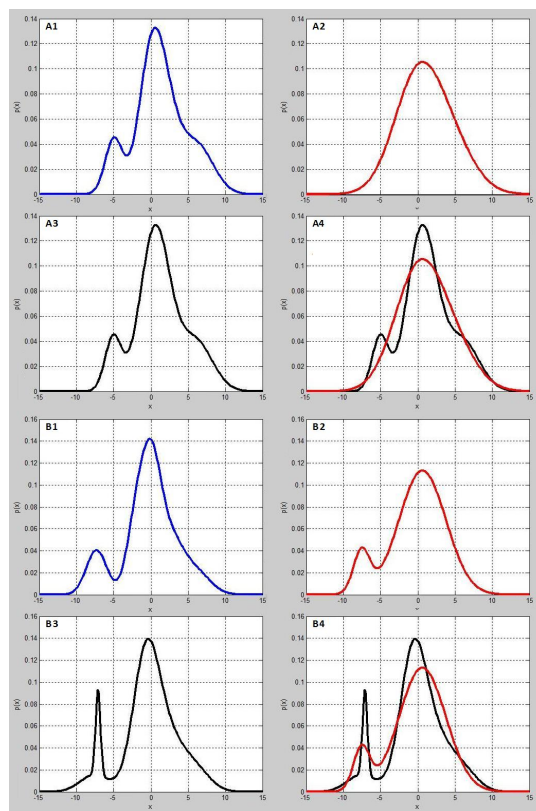


Figure 6.13: Two examples where the distance between  $mixt_M$  and  $mixt_N$  is close to the mean value (0.0232). In the top graphs (A1-A4)  $d_R = 0.0238$ , in the bottom graphs (B1-B4)  $d_R = 0.0226$ .

It is extremely important to note that the maximum distance between  $mixt_N$  and  $mixt_M$  is equal to 0.3755; figure 6.12 shows this case: as in the other figures, the original mixture is represented in black (in the lower left graph),  $mixt_N$  in blue

(in the upper left graph) and  $mixt_M$  in red (in the upper right graph). The lower right graph represents the original mixture and  $mixt_M$  plotted together. In this case  $d_N = 6.007e^{-5}$  while  $d_M = 0.3755$ .

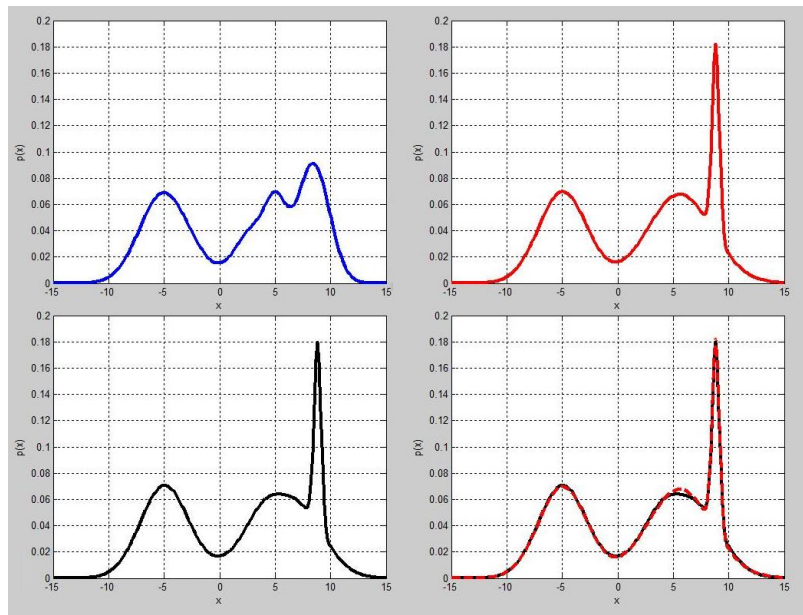


Figure 6.14: Experiments with 1d mixtures:  $mixt_M$  is closer to the original mixture than  $mixt_N$ . The original mixture (in black, lower left),  $mixt_N$  (in blue, upper left),  $mixt_M$  (in red, upper right), a graph containing the original mixture and  $mixt_M$  plotted together (in black and red respectively, lower right).

Considering all the mixtures generated, the mean value of  $d_R$  is equal to 0.0232 (in figure 6.13, we can see two cases with  $d_R$  equal to 0.0238 and 0.0226). Surprisingly in 10.5% of cases the approximation obtained by using  $M$  works better than  $mixt_N$ . Figure 6.14 shows an example. As we can see, there is a remarkable difference between the two different approximations and  $mixt_M$  overlaps almost perfectly the original mixture.

We have experimentally showed that the when we use the number of modes as an estimation of the number of components the mixture obtained has a distance from the original one comparable with the distance between this latter and the mixture obtained knowing the real number of components. Now we want to show how far  $mixt_M$  is from the original one. In our experiments the maximum value of this distance is 0.3755 (see figure 6.14) with a mean value of 0.0202 and a minimum of

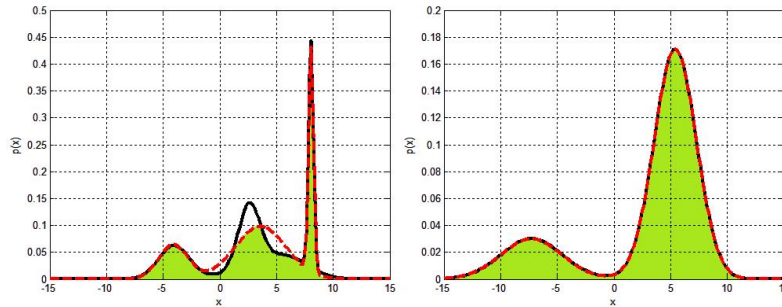


Figure 6.15: Experiments with 1d mixtures: the original mixture is represented in black, the estimated one with a dashed red line and, in green, the overlapping area. Two scenarios: a mean value for  $d_M = 0.202$  (on the left) and the minimum value for  $d_M = 4.0241e^{-6}$  (on the right).

$4.0241e^{-6}$  (see figure 6.15).

## 2D mixtures

The results obtained working with 2D confirm the trends of the previous ones. We generated 12,324 random mixtures. In 14.0% of the cases, the number of modes estimated by the mean shift is equal to the real number of components, so the two estimated mixtures are equivalent and the  $d_R$  is equal to 0. In 62.5% of cases  $mixt_M$  has a greater distance from the original mixture than  $mixt_N$ . But the error is negligible:

- in 41.75% of cases,  $|d_N - d_M| \leq 0.01$ ;
- in 13.00% of cases,  $0.01 < |d_N - d_M| \leq 0.05$ ;
- in 4.25% of cases,  $0.05 < |d_N - d_M| \leq 0.1$ ;
- in 3.50% of cases,  $0.1 < |d_N - d_M| \leq 1$ ;

It is extremely important to note that the maximum distance between  $mixt_N$  and  $mixt_M$  is equal to 0.4684; figure 6.16 shows this case: as in the other figures, the original mixture is represented in green (on the left),  $mixt_N$  in blue (in the center) and  $mixt_M$  in red (on the right). In the lower part are drawn the corresponding profiles projected on the  $xy$ -plane.

Considering all the mixtures generated, the mean value of  $d_R$  is equal to 0.0215; in figure 6.17 and 6.18, we can see two cases with  $d_R$  equal to 0.0217 and 0.0210).



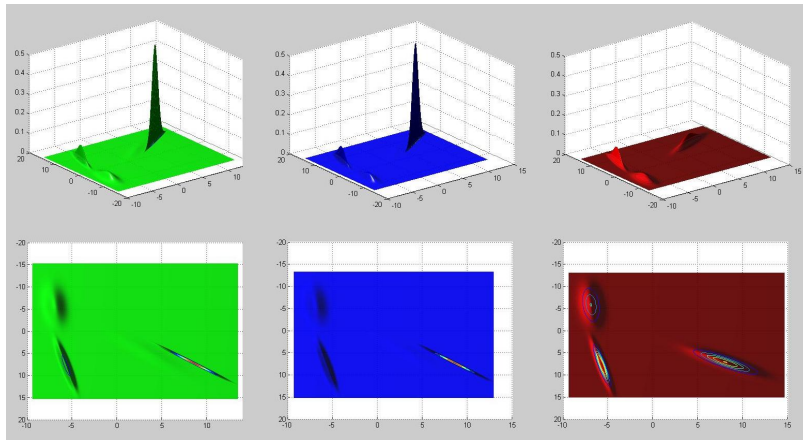


Figure 6.16: Experiments with 2d mixtures:  $mixt_N$  is closer to the original mixture than  $mixt_M$ . The original mixture (in green, left),  $mixt_N$  (in blue, center),  $mixt_M$  (in red, right).

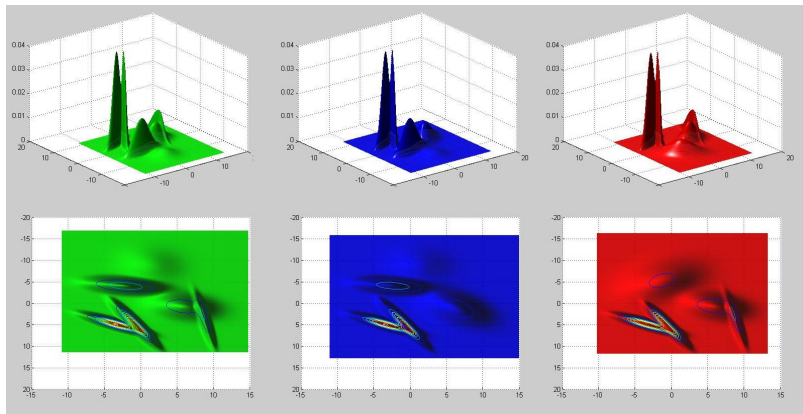


Figure 6.17: Experiments with 2d mixtures: an example where the distance between  $mixt_M$  and  $mixt_N$  is close to the mean value (0.0215):  $d_R = 0.0217$ . The original mixture (in green, left),  $mixt_N$  (in blue, center),  $mixt_M$  (in red, right).

There exist cases in which the approximation obtained by using  $M$  works better than  $mixt_N$ . Figure 6.19 shows an example. As we can see, as in the 1D scenario, there is a remarkable difference between the two different approximations and  $mixt_M$  overlaps almost perfectly the original mixture.

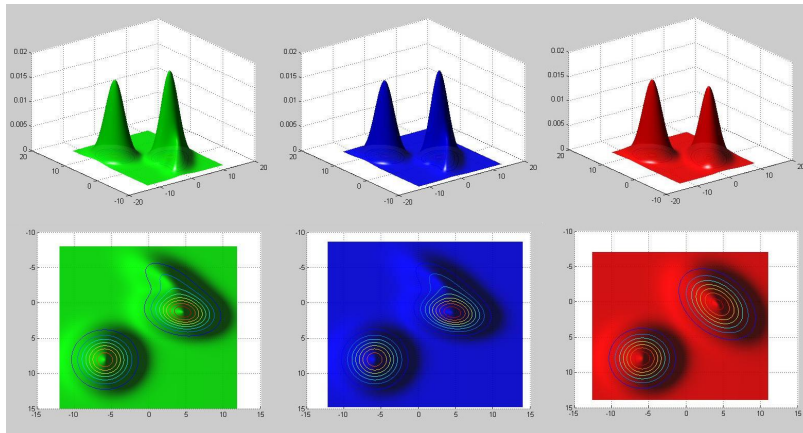


Figure 6.18: Experiments with 2d mixtures: another example where the distance between  $mixt_M$  and  $mixt_N$  is close to the mean value (0.0215):  $d_R = 0.0210$ . The original mixture (in green, left),  $mixt_N$  (in blue, center),  $mixt_M$  (in red, right).

### Higher dimensional mixtures

When we work with datasets with 3 or more dimensions, it is impossible to graphically represent the corresponding mixtures. We present in the following the result obtained. We generate 25, 342 mixtures with 3, 4 and 5 dimensions.

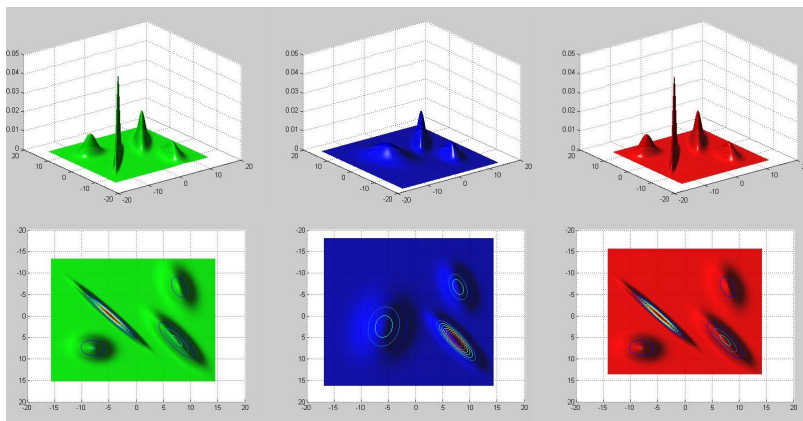


Figure 6.19: Experiments with 2d mixtures:  $mixt_M$  is closer to the original mixture than  $mixt_N$ . The original mixture (in green, left),  $mixt_N$  (in blue, center),  $mixt_M$  (in red, right).

- 5% of cases  $d_N = d_M$ ;

- 67.25% of cases  $d_N < d_M$  and specifically:
  - in 54.23% of cases,  $|d_N - d_M| \leq 0.01$ ;
  - in 11.02% of cases,  $0.01 < |d_N - d_M| \leq 0.05$ ;
  - in 1.21% of cases,  $0.05 < |d_N - d_M| \leq 0.1$ ;
  - in 0.79% of cases,  $0.1 < |d_N - d_M| \leq 1$ ;

In chapter 7, we will show how this approach can improve the performance of the Implicit Analysis.

## Chapter 7

# Final experiments

### 7.1 Introduction

Our aim is to define a similarity measure between unknown objects. To obtain this goal we perform two analysis based on different features: explicit analysis (based on shape) and implicit analysis (based on color). For the explicit analysis, we use the Procrustes distance between sets of critical points taken from the boundaries; for the implicit one, we determine the Normalized Mutual Energy between mixture of Gaussians estimated by the Expectation Maximization, using the number of peaks of the HS histogram as an estimation of the number of components. These two distance values are then merged by Fisher Linear Discriminant analysis. Subsequently we try to improve the results obtained in the two analysis. Specifically a similarity measure based on Fréchet and topological distances has been used to compare shapes instead of Procrustes, and the number of components of the mixture (for the EM) has been estimated by a Mean Shift procedure (that is by using the estimated number of modes of the datasets). In this chapter we present experimental results which show how these two new approaches can improve the performance of the procedure.

### 7.2 Explicit analysis: topological-Fréchet distance

We have already shown that even Procrustes generally returns encouraging results on similar shapes, but when it has to deal with images containing the same object in different poses, the performance becomes worst. The new measure (that we call topological-Fréchet distance) seems to partly solve this problem. For our experiments we have used logical images containing single objects belonging to a database of 35 categories of shapes. For each category, the objects are presented in different poses,

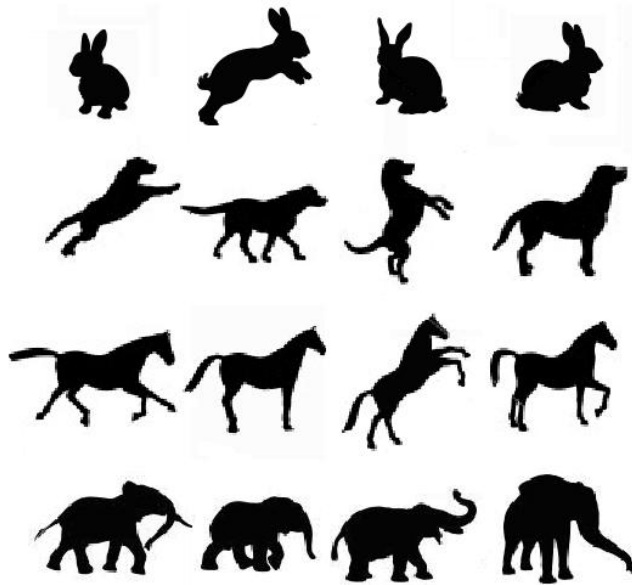


Figure 7.1: Logical images representing four animals in different poses

for a total of 3,600 shapes. In figure 7.1 some examples of shapes used in our tests are reported.

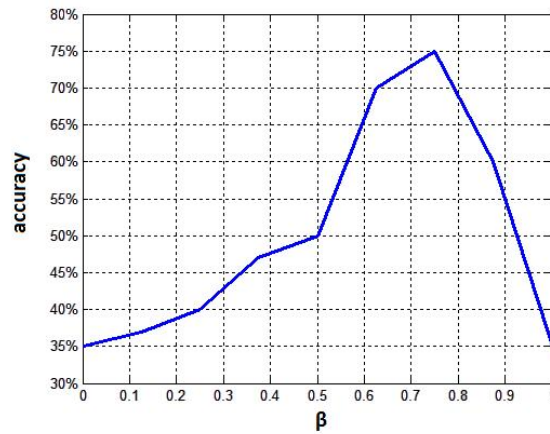
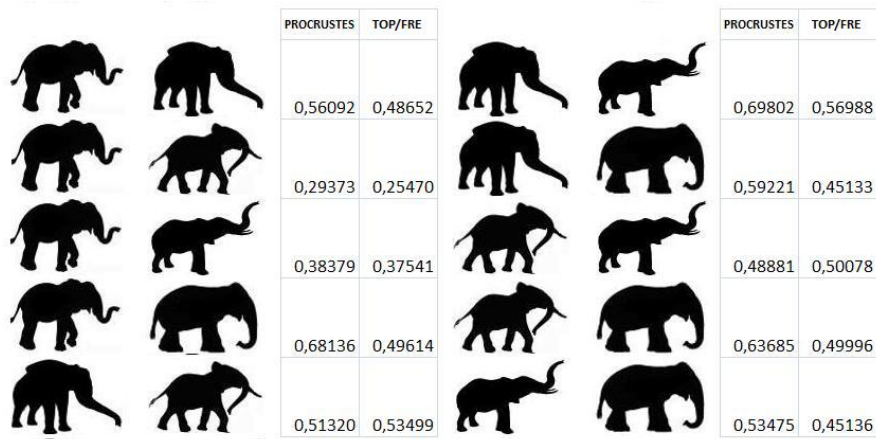


Figure 7.2: A graph representing the accuracy of Topological-Fréchet similarity measure with respect to  $\beta$

We have divided our dataset in a training set (TS) of 72 elements (2 per category) and a validation set (VS). We have first extracted the contour of TS shapes, topologically ordering their points, and then found critical points with the method described in section 4.3.5. For this set we have empirically computed the three thresholds  $t_F$ ,  $t_T$  and  $\tau_s$  in order to guarantee a classification accuracy greater than 70%. Then we have used the thresholds found for classifying the similarity between VS shapes. We have repeated the experiment 9 times using different values of  $\beta$  (see section 5.4). The results obtained are listed in figure 7.2. Varying  $\beta$  from 0 to 1, we have analyzed the influence of the topological and Fréchet distance on the accuracy in finding the similarity between objects of the same class. As we can see (and as it was supposed), a slightly higher relevance of the topological component gives the best results.



PROCRUSTES	TOP/FRE	PROCRUSTES	TOP/FRE
0,56092	0,48652	0,69802	0,56988
0,29373	0,25470	0,59221	0,45133
0,38379	0,37541	0,48881	0,50078
0,68136	0,49614	0,63685	0,49996
0,51320	0,53499	0,53475	0,45136

Figure 7.3: A comparison between the values with the Procrustes and the Fréchet distance among the elephant family

Now we show the results obtained with this similarity measure, compared with the results obtained with Procrustes. In the analysis of poses, topological-Fréchet distance presents better performance. Figure 7.3 shows some experiments: in the first column of the table the values obtained with Procrustes are listed, while in the second column are the values obtained with topological-Fréchet. We recall that we previously set the similarity threshold such that two instances can be considered similar if their distance is below  $\pi = 0.52$ . As we can observe, on average the values of Procrustes are greater than the values of Fréchet, with only two exceptions: couples 5 and 8. Note that the new values are able to return a correct classification also in the cases where Procrustes was unable (see couples 1, 4, 5, 6, 7 and 8), and that topological-Fréchet distance returns values above the threshold, whenever Procrustes does. Thus, also maintaining the value of the threshold ( $\pi = 0.52$ ), the performance



Figure 7.4: Color images representing four subjects

increases.

The selection of results presented in figure 7.3 is representative of the entire panel of our experiments.

### 7.3 Implicit analysis: number of modes for number of components

In the implicit analysis, the new methodology will return better results with respect to the previous one in all cases considered. This fact has not to be surprising: in fact, Mean Shift is able to identify more accurate results than a peak-climbing algorithm. Moreover, in the previous experiments we consider only two color channels (H and S), without any analysis on the third. Instead the new approach considers complete RGB tuples, and the mode analysis and best-fitting mixture estimation are both performed on the entire color space. For our experiments we have used RGB images containing objects belonging to a database of 24 categories. Obviously we focused our attention on classes where the color is a discriminative feature. We work with 2,400 objects. In figure 7.4 some examples of images used in our tests are reported.

Now we present some examples showing how the results obtained with the new







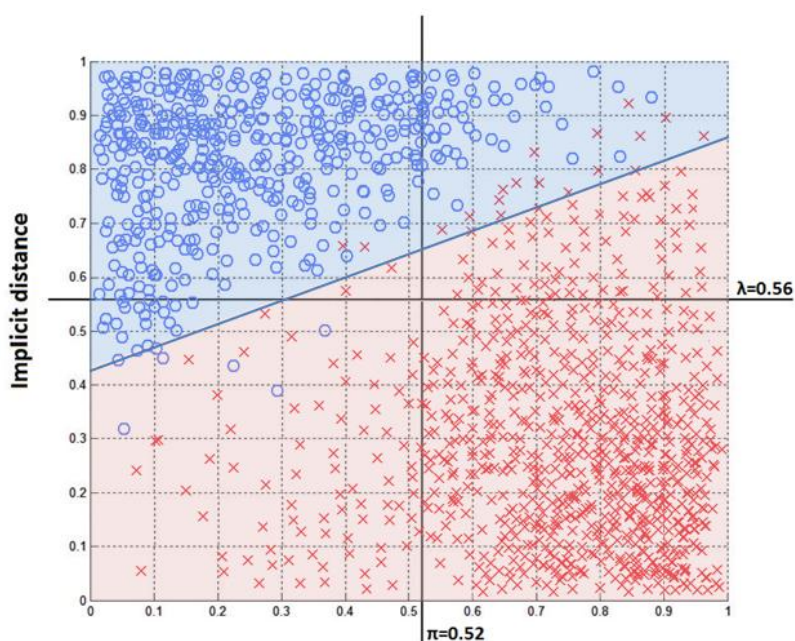


Figure 7.6: On the  $x$ -axes the explicit distance (Normalized Mutual Energy), on the  $y$ -axes the implicit distance (topological-Fréchet distance). The blue circles represent the elements belonging to the similarity class, while the red crosses represent the ones belonging to non-similarity. Two elements are considered similar if the points representing their two distances fall in the blue area, obtained by using Fisher discriminant analysis

implicit one. So the decision region is closer to the area which represents the implicit similarity than to the other.

## 7.5 A synthesis of the whole process

In this section we present a specific example to summarize the process. Starting from an image downloaded from the Internet representing an acoustic guitar, we perform the analysis to obtain the implicit and the explicit models. Subsequently, we compare the image with 24 pictures downloaded from the Internet or taken from our databases, 20 of which are chosen randomly, while the remaining 4 contain the same subject of the original 2 in different colors.



Figure 7.7: An acoustic guitar.



Figure 7.8: A BW version of figure 7.7.

### 7.5.1 Extracting the implicit and the explicit models

Figure 7.7 shows the image selected for our analysis. It is a  $600 \times 449$  JPEG color picture representing an acoustic guitar. As figure 7.8 shows, first of all, we convert the image in BW by performing a local thresholding (see section 3.3.2).

Then the starting clusters, obtained grouping together adjacent pixels having the same value, are defined (see section 3.3.2). Figure 7.9 shows the result of this step.

By combining chromatic and positional analysis, we obtain a probability map

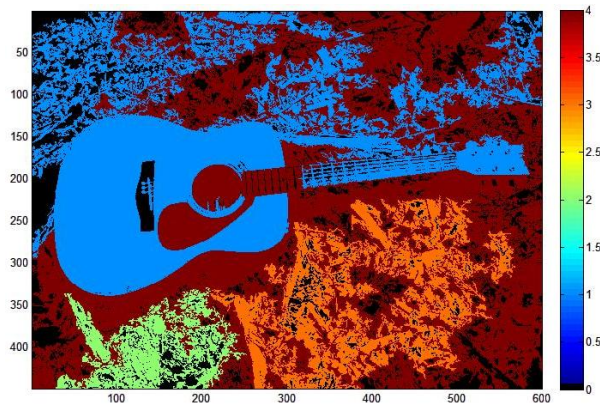


Figure 7.9: The starting clusters obtained from figure 7.8.

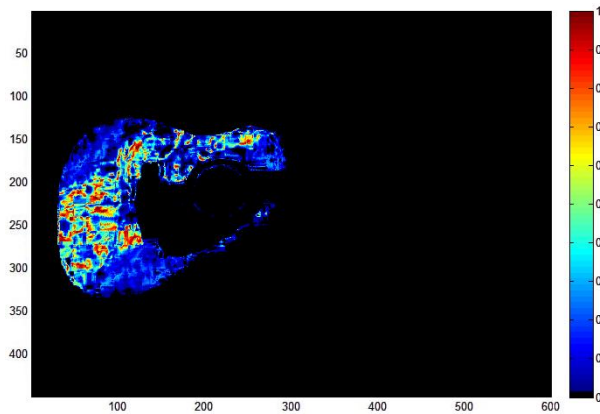


Figure 7.10: The probability map obtained from cluster 1 in figure 7.9.

which assigns a value to each pixel in the image (see sections 3.3.3 and 3.3.4). Figure 7.10 shows the result of these steps.

Subsequently we define a graph over the pixels of the image and we determine the weights of the edges starting from the values of the probability map previously obtained. We calculate the  $s - t$  cut of the graph and we obtain the final cluster (see section 3.3.5). Figure 7.11 shows the result: as we can see, the segmentation algorithm individuate only a part of the whole object.

To determine the explicit model, we resize the image to a dimension of  $60 \times 60$  pixels, we extract the edge (section 4.3.3) and we calculate the critical points (section 4.3.5). Then we define the polygonal version and we calculate the shape patterns



Figure 7.11: The final cluster obtained by the  $s - t$  cut over the graph initialized with the probability values of figure 7.10.

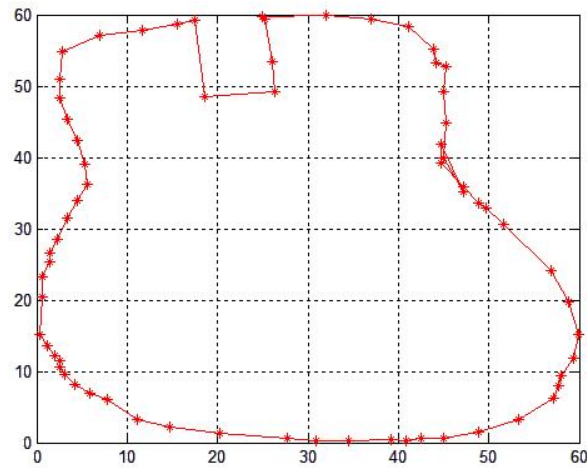


Figure 7.12: The polygonal version of figure 7.11 with critical points highlighted.

(sections 5.2.1 and 5.2.2) of the original image (as shown in figure 7.12).

To obtain the explicit model, we consider the RGB tuples of figure 7.11, we determine the number of modes by the Mean Shift (section 6.7.4) and we evaluate the best-fitting mixture by the EM. The mixture so obtained has the following parameters (note that the RGB values are in the range  $[-15, +15]$ ) :

- number of components: 3;

- priors:  $\pi_1 = 0.6374$ ,  $\pi_2 = 0.2076$ ,  $\pi_3 = 0.1549$ ;
- means:

$$\mu_1 = [12.38, 4.96, -7.26]^T, \mu_2 = [8.34, 2.34, -6.58]^T,$$

$$\mu_3 = [10.63, 6.31, -1.61]^T$$

- covariances:

$$\Sigma_1 = \begin{bmatrix} 0.73 & 0.72 & 0.31 \\ 0.72 & 0.87 & 0.50 \\ 0.31 & 0.50 & 1.60 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1.80 & 0.88 & -0.19 \\ 0.88 & 0.59 & 0.35 \\ -0.19 & 0.35 & 1.75 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 11.62 & 7.32 & -0.20 \\ 7.32 & 8.09 & 7.99 \\ -0.20 & 7.99 & 22.54 \end{bmatrix}$$

### 7.5.2 Similarity measure

Figure 7.13 shows the comparison values between the original images and the others. The first row shows the values of the explicit analysis, the second row the values of the implicit, and in the lower row there is the classification obtained merging the previous values.

As we can observe, even if the segmentation step did not extract the whole object, we obtain a good matching with the 4 pictures representing the same object. It is not surprising that one of the two instances of the same object with different colors has been misclassified. In fact the final decision is taken considering both distances and, although the value of the implicit distance is below the threshold, the classification fails. At the same time, the value of the explicit distance between the guitar and the flower is acceptable, but the classification fails due to the implicit analysis.

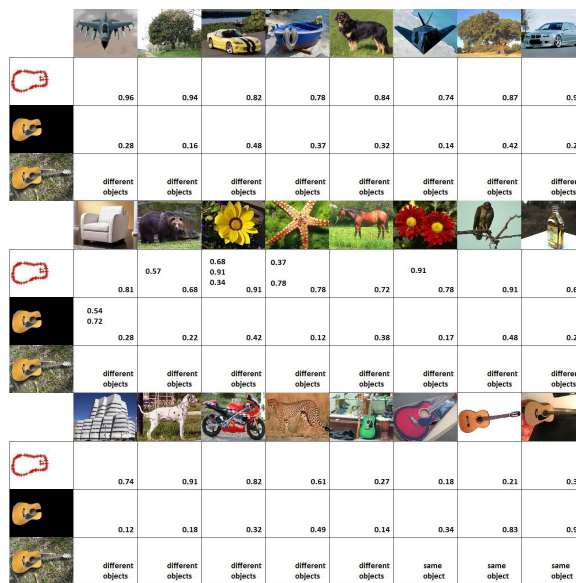


Figure 7.13: The results obtained comparing the original image with 24 images taken from the Internet. The first row contains values of the explicit analysis ( $\pi = 0.52$ ), the second row values of the implicit analysis ( $\lambda = 0.56$ ), and the third row contains the classification obtained merging the two previous values.



## **Part III**

# **Conclusions**





## Chapter 8

# Conclusions and Future Works

### 8.1 Introduction

The development of a general bottom-up object recognition and classification system is very complex. In the last three decades plethora of different methodologies have been defined. An efficient and effective solution is very far from being reached. In the present work we analyzed some of the most common problems in this specific field. The main objective was to define a similarity measure between two unknown objects extracted from a digital image.

### 8.2 Summary of the work

#### 8.2.1 A bottom-up approach for image segmentation

First we defined a procedure to extract the interesting objects from a scene in a purely bottom-up approach. To obtain this result we combined light, color and positional analysis with statistical methodologies to locate the meaningful areas. First we converted an RGB image, by using a local thresholding, in its BW version; then we grouped together adjacent pixels having the same values (obtaining the starting clusters). This coarse grouping also generated meaningless blocks. For each single cluster, we analyzed the connectivity and discard the ones that seemed not promising; then, considering the set of RGB tuples of the block, we estimated the number of modes by the Mean Shift and we used this value as an estimation of the number of components to obtain, by Expectation Maximization, the best-fitting mixture, able to describe the chromatic characteristics of the cluster. Once this pdf is to hand, we assigned to each pixel of the starting image a chromatic probability. These proba-

bilities were subsequently refined by using information about the relative position of the pixels in the image. To obtain the final segmentation, we first defined a weighted graph having a node for each pixel of the image, where the weights of the edges are initialized by using the values of the probability map. Then we perform an  $s - t$  cut and we isolated a possible object.

### 8.2.2 A similarity analysis based on shape and color

To evaluate the similarity between two (possible unknown) objects, we combined shape and chromatic analysis. The explicit model brings information about the shape, instead the implicit model about color. In the explicit analysis we extracted the boundary from the object and we determined on it some reference points (called "critical points"). On the other hand, in the implicit analysis, we represented the object by the set of RGB tuples of the area delimited by the boundary. Once defined two distance measures able to compare the models, we combined them by Fisher Linear Discriminant analysis, obtaining a similarity measure.

### 8.2.3 Explicit analysis

Our first approach to explicit distance was based on Procrustes analysis. Once individuated the critical points, we considered the Procrustean distance between them as an estimation of their similarity. This approach was not completely satisfying. First of all, it requires that the number of elements of the two datasets is the same; moreover it presents very low performance when we compare two shapes representing the same object in different poses. We tried to improve the results by introducing a different shape description (which considers also the local spatial characteristics of the boundary points around the critical ones) and a similarity measure obtained as a combination of the Fréchet and the topological distances. This approach seemed to partly solve the previous problems.

### 8.2.4 Implicit analysis

Instead, to compare the implicit models, we started from the idea of supposing the existence of a model underlying the datasets: a probability density function able to summarize the most important characteristics of the sets and to make them easily comparable. Specifically we opted for mixtures of Gaussians and we used the Expectation-Maximization algorithm to estimate the unknown parameters. To perform the estimation, EM needs to know in advance the number of the components of

the estimating mixture. First we used an empirical approach to estimate this value: we converted the RGB into HSV tuples, we calculated the histogram representing the HS values of the set, and the peaks number of the histogram obtained was our estimation of the number of components. Obviously, discarding a color space component leads to loss of information. So we tried to refine this approach considering a complete color space. Specifically we estimated the number of modes of the RGB tuples by the Mean Shift algorithm and we used this value for the EM. Once obtained the two pdfs, we used a methodology introduced by Lyu to evaluate in a closed form the distance between them.

### 8.2.5 Number of modes as component estimation

The estimation of the number of components for an unknown mixture model is one of the most studied problems in statistical literature. We perform this estimation by using the number of modes. This approach is very efficient, since a reliable iterative procedure exists able to obtain this value: the Mean Shift. Unfortunately, it is known that there exist scenarios in which the number of components and the number of modes differ. To obtain an evaluation of our approach, we perform a systematic experimental analysis with the aim of understanding the quality of the approximation. Specifically we generated random mixtures of Gaussians and we sampled them. By using the Mean Shift we calculated the number of modes of the obtained data set. Then we perform twice best-fitting mixture estimations by EM algorithm on the set: first by using the number of modes, then by using the real number of components (known from the original mixture). By using the measure between mixtures of Gaussians previously introduced, we estimated the distances between the two obtained mixtures and the original one. The experiments confirmed our hypothesis: the estimation error generated by the use of the number of modes is absolutely negligible.

## 8.3 Open problems and future works

There are still some open problems:

- the segmentation step, as already stated in section 3.5, needs further refinements. First of all, in the worst case, computational time is not feasible for real applications. It is possible to perform approximations which can reduce the cost. A great reduction can be obtained, grouping together adjacent pixels before the first step of the algorithm and working with these sets of pixels instead that with single units. Obviously, as a consequence of this resolution reduction, the performances of the algorithm became worst. We are actually

performing a comparative study to understand the optimal approximation. In the same direction, mainly inspired by (206; 3), we are also evaluating which results can be obtained working with different color spaces and discarding one or two color channels.

- To describe a shape we used critical points, defined as the points where the boundary has sudden variations. Even if we defined critical points from an analytical point of view, we used an empiric approach to individuate them. The procedure is efficient, but, also due to pixelization problems, it does not individuate the minimal set. Obviously a redundant description makes the comparison procedures less efficient and less effective. So we are studying a different approach to define critical points.
- The comparison between shapes, either with the Procrustes or the topological-Fréchet distance, does not produce satisfactory results. It is reasonable to perform first a global structure comparison between shapes and subsequently a local analysis. Some results obtained with skeletons (see (282)) seem to be really encouraging. We are performing a first experimental phase to test the possibility of integrating our methodology with the results described in (282). Specifically two characteristics are very appealing: the stability under deformations and the ability to return significant distance values between instances of the same class in different poses.
- Working with shapes, we observed that some known techniques can either easily find similarities on the general structure without giving any result in the analysis of small local variations, or they can manage the pose variations without presenting interesting results on the general structure. The proposed combination of skeleton and critical points (currently under analysis) could probably mitigate this drawback. Note that two different poses of an object are mainly represented by boundaries having many common subparts and dissimilarities only in some areas interested by the variations. Obviously it is complex to define a technique to compare two entities with these characteristics: the boundary has to be divided into subparts and similar blocks have to be correlated obtaining a global distance between the two elements. The scenario seems to be very similar to a multiple sequence alignment problem. A multiple sequence alignment (MSA) is an alignment technique used for biological sequences (generally protein, DNA, or RNA). The aim of the process is to determine whether the query sequences have an evolutionary relationship by which they share a lineage and are descended from a common ancestor. MSA also refers to the process of aligning such a sequence set. Because three or more sequences of biologically relevant length can be difficult and are almost always time-consuming to align by hand, computational algorithms are used

to produce and analyze the alignments. There are interesting analogies with our scenario and we are trying to adapt the presented methodology to shape analysis.



## **Part IV**

# **Appendix**





# Appendix A

## Appendix

### A.1 A brief overview of the probability density distance measures

in the following we present the most used pdf distance measures and their relationships. We recall that  $ME_N$  is an upper bound for the  $KL$  distance.

#### A.1.1 Kullback-Leibler distance

##### description

Also known as Information divergence (or I directed divergence).

##### formulation

$$KL(p_1||p_2) = \sum_{x \in X} p_1(x) \log \frac{p_1(x)}{p_2(x)}$$

##### note

The base of the logarithm usually is 2. Here we consider the natural logarithm, so the base is  $e$ .

##### properties

$KL$  distance is **not symmetric** and in fact generally, as it is simple to see,  $KL(p_1||p_2) \neq KL(p_2||p_1)$ .  $KL$  is **non negative** and **additive**. One important drawback with  $KL$

distance is that it is undefined for  $p_2 = 0$  and  $p_1 \neq 0$  and so  $p_1$  has to be absolutely continuous w.r.t.  $p_2$ .

### identities and inequalities

It is important to note that:

$$KL(p_1||p_2) \geq \max \{L_1(V(p_1||p_2)), L_2(V(p_1||p_2))\}$$

where  $V$  is the *variational distance* (see below) and

$$L_1(x) = \log \frac{2+x}{2-x} - \frac{2x}{2+x}$$

and

$$L_2(x) = \frac{x^2}{2} + \frac{x^4}{36} + \frac{x^6}{288}$$

Even if this interesting lower bound exists, note that there is no general upper bound for  $KL$  in terms of  $V$ .

$KL$  can be expressed in function of  $L$  divergence:

$$KL(p_1||p_2) = \sum_{\nu=0}^{\infty} 2^{\nu} L(m_{\nu}||p_2)$$

where  $m_{\nu} = 2^{-\nu}p_1 + (1 - 2^{-\nu})p_2$ . Moreover  $KL$  can be bounded by  $\Delta^*$ :

$$\frac{1}{2}\Delta^*(p_1||p_2) \leq KL(p_1||p_2) \leq \log 2 \cdot \Delta^*(p_1||p_2)$$

where  $\Delta^*$  is the quantity defined in the subsection about the *triangular discrimination of order  $\nu$* .

$KL$  can also be bounded by  $V$ :

$$KL(p_1||p_2) \geq \frac{1}{2} \sum_{\nu=0}^{\infty} 2^{\nu} \frac{1}{2} (V(m_{\nu}||p_2))^2 = \frac{1}{2} V(p_1||p_2)$$

Theorem 4 of (281) shows another interesting inequality about  $KL$ :

$$KL(p_1||p_2) \leq L(p_1||p_2) + \log \left( \frac{1}{2} (1+c) \right)$$

where  $c = \max_{x \in X} \left( \frac{p_1}{p_2} \right)$ .

**note**  $KL$  is an instance of *Ali – Silvey* class with  $c(x) = x \log x$  and  $f(x) = x$ .

### A.1.2 J divergence

#### description

This measure was introduced to overcome the non-symmetry of the previous measure. In fact it can be expressed (as you can see in the following) as the sum of the two  $KL$  distances evaluated on the couple  $p_1$  and  $p_2$ . It was firstly introduced by Jeffreys (see (147) and (148)). It is defined as the difference in the mean values of the log-likelihood ratio under the two hypotheses.

#### formulation

$$\begin{aligned} J(p_1||p_2) &= \sum_{x \in X} (p_1 - p_2) \log \frac{p_1}{p_2} \\ &= KL(p_1||p_2) + KL(p_2||p_1) \\ &= E_1[\log Li(x)] - E_2[\log Li(x)] \end{aligned}$$

where  $E_i$  is the expected value w.r.t. the distribution  $p_i$  and  $Li(x) = \frac{p_1}{p_2}$  is the likelihood ratio.

#### properties

The  $J$  divergence is **symmetric** (so it solves one of the problems of the previous measure), but it is defined only whether  $p_1$  and  $p_2$  are absolutely continuous w.r.t. each other. It satisfies **all the properties of a metric** except the **triangle inequality**.  $J$  is also additive.

#### identities and inequalities

As stated before, the  $J$  divergence can be expressed in function of  $KL$  divergence:

$$J(p_1||p_2) = KL(p_1||p_2) + KL(p_2||p_1)$$

### A.1.3 K divergence

#### description

This measure was introduced to overcome some drawbacks of the  $KL$  divergence. It can be expressed in function of  $KL$ .

**formulation**

$$K(p_1||p_2) = \sum_{x \in X} p_1(x) \log \frac{p_1(x)}{\frac{1}{2}p_1(x) + \frac{1}{2}p_2(x)}$$

**properties**

The K divergence is **non negative** but (as the  $KL$ ) it is also **not symmetric**. It is important to highlight that it is well defined for every  $x$ . In addition it is **finite**, **semibounded** and **bounded by 1**. Another very interesting characteristic is the **isolation**:  $K(p_1||p_2) = 0$  iff  $p_1 = p_2$ .

**identities and inequalities**

As stated before,  $K$  can be expressed in function of  $KL$ :

$$K(p_1||p_2) = KL\left(p_1||\frac{1}{2}p_1 + \frac{1}{2}p_2\right)$$

Theorem 1 in (177) states that:

$$K(p_1||p_2) \leq \frac{1}{2}KL(p_1||p_2)$$

Theorem 2 in (177) states that:

$$K(p_1||p_2) \geq \max\left\{L_1\left(\frac{V(p_1||p_2)}{2}\right), L_2\left(\frac{V(p_1||p_2)}{2}\right)\right\}$$

Moreover it is possible to define an upper bound from  $V$ :

$$K(p_1||p_2) \leq V(p_1||p_2)$$

**note**

$K$  divergence coincides with the  $f$ -divergence with

$$f(x) = x \log\left(\frac{2x}{1+x}\right)$$

**A.1.4 L divergence****description**

$L$  divergence is also known as *Capacitory discrimination*. It was introduced to obtain a symmetrized version of  $K$  divergence. As the  $J$  divergence can be seen as the sum of the two  $KL$  distances evaluated on the couple  $p_1$  and  $p_2$ , so the same is for the  $L$  divergence respect to the  $K$  divergence.

In some literature this measure is known as *Jensen – Shannon* divergence.

**formulation**

$$\begin{aligned} L(p_1||p_2) &= \frac{1}{2} \left( \sum_{x \in X} p_1 \log \frac{2p_1}{p_1 + p_2} + \sum_{x \in X} p_2 \log \frac{2p_2}{p_1 + p_2} \right) = \\ &= K(p_1||p_2) + K(p_2||p_1) \end{aligned}$$

**properties**

$L$  divergence is obviously **symmetric**. As the  $K$  divergence from which it derives,  $L$  divergence is **non negative, finite, semibounded and bounded** (in particular  $L(p_1||p_2) \leq 2$ ). It is of extreme importance to note that  $L$  divergence is the **square of a metric**. Being the sum of the two  $K$  divergence measure on  $p_1$  and  $p_2$ , for  $L$  divergence also holds **isolation**:  $L(p_1||p_2) = 0$  iff  $p_1 = p_2$ .

Following the notation introduced in (104), if we consider the set  $M_+^1(A)$  of probability distributions where  $A$  is a set provided with some  $\sigma$ -algebra, we can define  $L$  as a function:

$$L : M_+^1(A) \times M_+^1(A) \rightarrow [0, \infty]$$

recalling that  $L$  is the square of a metric, it is important to highlight that  $(M_+^1(A), \sqrt{L})$  is isometrically isomorphic to a subset in Hilbert space.

**identities and inequalities**

As stated before,  $L$  divergence derives from  $K$  divergence, which, in turn, derives from  $KL$ . So:

$$\begin{aligned} L(p_1||p_2) &= K(p_1||p_2) + K(p_2||p_1) \\ &= \frac{1}{2}KL \left( p_1 || \frac{1}{2}p_1 + \frac{1}{2}p_2 \right) + \frac{1}{2}KL \left( p_2 || \frac{1}{2}p_1 + \frac{1}{2}p_2 \right) \end{aligned}$$

It is simple to find a relation between this measure and the  $J$  divergence:

$$L(p_1||p_2) \leq \frac{1}{2}J(p_1||p_2)$$

Theorem 3 in (177) states that:

$$L(p_1||p_2) \leq v(p_1||p_2)$$

It is important to observe that  $L$  divergence can be expressed as a function of the *Shannon entropy function* (represented by  $H$ ):

$$L(p_1||p_2) = 2H \left( \frac{p_1 + p_2}{2} \right) - H(p_1) - H(p_2)$$

Considering the *Information Transmission Rate* we have:

$$L(p_1||p_2) = 2I\left(\frac{1}{2}, \frac{1}{2}\right)$$

$L$  can be expressed in function of the *triangular* discrimination of order  $\nu$  (Theorem 1 from (281)):

$$L(p_1||p_2) = \sum_{\nu=1}^{\infty} \frac{1}{2\nu(2\nu-1)} \Delta_{\nu}(p_1||p_2)$$

### A.1.5 Jensen-Shannon divergence

#### description

This measure allows to assign a different weight to each probability distribution. This makes *Jensen – Shannon* divergence particularly suitable for decision problems where the weights could be the prior probabilities.

#### formulation

$$JS_{\pi}(p_1||p_2) = H(\pi_1 p_1 + \pi_2 p_2) - \pi_1 H(p_1) - \pi_2 H(p_2)$$

where  $\pi_1$  and  $\pi_2$  are the weights of the two probability distributions  $p_1$  and  $p_2$ , respectively.

$$\pi_1, \pi_2 \geq 0, \pi_1 + \pi_2 = 1$$

#### properties

Since  $H$  is a concave function, according to Jensen's inequality,  $JS_{\pi}(p_1||p_2)$  is **non-negative** and for it holds **isolation**:  $JS_{\pi}(p_1||p_2) = 0$  when  $p_1 = p_2$ .  $JS$  can be generalized to provide such a measure not only for two, but for any finite number of distributions. It is important to note (see following paragraph) that  $JS$  divergence provides both lower and upper bounds to the *Bayes probability of error*; from these bounds it is possible to note that  $JS$  is also **bounded** by 1.

#### identities and inequalities

Theorem 4 and 5 in (177) state that:

$$P_e(p_1, p_2) \leq \frac{1}{2} (H(\pi_1, \pi_2) - JS_{\pi}(p_1||p_2))$$

$$P_e(p_1, p_2) \geq \frac{1}{4} (H(\pi_1, \pi_2) - JS_{\pi}(p_1||p_2))^2$$

where  $H(\pi_1, \pi_2) = -\pi_1 \log \pi_1 - \pi_2 \log \pi_2$  and  $P_e$  is the Bayes probability of error:

$$P_e(p_1, p_2) = \sum_{x \in X} \min(\pi_1 p_1, \pi_2 p_2)$$

So it is possible to derive ( $P_e$  is non negative):

$$JS_\pi(p_1 || p_2) \leq H(\pi_1, \pi_2) - 2P_e(p_1, p_2) \leq H(\pi_1, \pi_2) \leq 1$$

**note**

As stated before, this measure can be generalized for any finite number of distributions (not only two):

$$JS_\pi(p_1, p_2, \dots, p_n) = H\left(\sum_{i=1}^n \pi_i p_i(x)\right) - \sum_{i=1}^n \pi_i H(p_i(x))$$

It is possible again to provide bounds to the *Bayes probability of error* (Theorem 6 and 7 in (177)):

$$P(e) \leq \frac{1}{2} (H(\pi) - JS(p_1, p_2, \dots, p_n))$$

$$P(e) \geq \frac{1}{4(n-1)} (H(\pi) - JS(p_1, p_2, \dots, p_n))^2$$

where:

$$P(e) = \sum_{x \in X} p(x) (1 - \max(p(c_1|x), p(c_2|x), \dots, p(c_n|x)))$$

**A.1.6 Other distance measures**

We briefly present other distance measures which are interesting for the inequalities with the measures already presented.

**Triangular discrimination**

$$\Delta(p_1 || p_2) = \sum_{x \in X} \frac{|p_1(x) - p_2(x)|^2}{p_1(x) + p_2(x)}$$

**Triangular discrimination of order  $\nu$**

$$\Delta_\nu(p_1 || p_2) = \sum_{x \in X} \frac{|p_1(x) - p_2(x)|^{2\nu}}{(p_1(x) + p_2(x))^{2\nu-1}}$$



**Variational distance**

$$V(p_1||p_2) = \sum_{x \in X} |p_1(x) - p_2(x)|$$

**Kolmogorov Variational distance**

$$KV_\pi(p_1||p_2) = \sum_{x \in X} |\pi_1 p_1(x) - \pi_2 p_2(x)|$$

where  $\pi_1, \pi_2 \geq 0$  and  $\pi_1 + \pi_2 = 1$ .

**Kendall's  $\pi$** 

$$\pi(p_1||p_2) = \sum_{x, y \in X} \frac{\text{sign}[(p_1(x) - p_2(y))(p_1(x) - p_2(y))]}{2 \binom{|X|}{2}}$$

**Hellinger distance**

$$H(p_1||p_2) = \sqrt{\sum_{x \in X} \left( \sqrt{p_1(x)} \sqrt{p_2(x)} \right)^2}$$

**A.1.7 General inequalities**

In this section are presented some inequalities which relate together some of the measures introduced in the previous section.

$$\frac{1}{2}L(p_1||p_2) \leq I_{max} \leq R_{min} \leq R(M) \leq L(p_1||p_2)$$

Combining some inequalities already introduced, we can also obtain:

$$\begin{aligned} \frac{1}{8}V(p_1||p_2) &\leq \frac{1}{4}\Delta(p_1||p_2) \leq \frac{1}{2}L(p_1||p_2) \leq I_{max} \leq R_{min} \leq \\ &\leq L(p_1||p_2) \leq \log 2 \cdot \Delta(p_1||p_2) \leq \log 2 \cdot V(p_1||p_2) \end{aligned}$$

# Bibliography

- [1] Y. S. Abu-Mostafa and D. Psaltis. Image normalization by complex moments. *IEEE Transactions on PAMI*, 7(1):46–55, 1985.
- [2] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer. The probability problem of pattern recognition learning and the method of potential functions. *Automation and Remote Control*, 25:1175–1190, 1964.
- [3] M. Alencastre-Miranda, Munoz-Gomez L., R. Swain-Oropeza, and Nieto-Granda. Color-image classification using mrfs for an outdoor mobile robot. *Journal of Systemics, Cybernetics and Informatics*, 3(1):52–59, 2004.
- [4] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Society. Series B (Methodological)*, 28(1):131–142, 1966.
- [5] Yali Amit and Donald Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.
- [6] B. Appleton and H. Talbot. Globally optimal surfaces by continuous maximal flows. *Digital Image Computing: Techniques and Applications, Proc. VIIth APRS*, 1:623–632, 2003.
- [7] C. Arcelli and G. Sanniti di Baja. A width-independent fast thinning algorithm. *IEEE Trans. PAMI*, 7(4):463–474, 1985.
- [8] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.
- [9] D. H. Ballard. Generalising the hough transform to detect arbitrary shapes. *Pattern Recognition*, (13):111–122, 1981.

- 
- [10] Z. Barutcuoglu and C. DeCoro. Hierarchical shape classification using bayesian aggregation.
- [11] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1988.
- [12] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.
- [13] A. Baumberg. Reliable feature matching across widely separated views. *CVPR*, pages 774–781, 2000.
- [14] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *in the ninth European Conference on Computer Vision*, 2006.
- [15] A. Beinat and F. Crosilla. Generalised procrustes analysis for size and shape 3-d object reconstructions. *Optical 3-D Measurement Techniques*, pages 345–353, 2001.
- [16] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. *Proc. of 8th IEEE Int'l Conference on Computer Vision*, 1:454–461, 2001.
- [17] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Patt. Anal. and Machine Intell.*, 24(1):509–522, 2002.
- [18] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.
- [19] K. P. Bennett and C. Campbell. Support vector machines: hype or hallelujah? *SIGKDD Explorations*, 2(2):1–13, 2000.
- [20] S. Berretti, A.D. Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and elective indexing. *IEEE Trans. Multimedia*, 2(4):225–239, 2000.
- [21] P. J. Bickel and M. Roseblatt. On some global measures of the deviations of density function estimates. *Annals of Statistics*, pages 1071–1095, 1973.
- [22] I. Biederman. *An Invitation to Cognitive Science*, volume 2. The MIT Press, 1995.
- [23] Irving Biederman. Recognition by components - a theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

- 
- [24] Irving Biederman. Recognizing depth-rotated objects: A review of recent research and theory. *Spatial Vision*, 13:241–253, 2001.
- [25] L. Biederman. On the semantic of a glance at a scene. in *Perceptual Organization*, pages 213–263, 1981.
- [26] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-021, UC Berkeley, 1997.
- [27] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [28] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.
- [29] H. Blum. A transformation for extracting new descriptors of shape. *Proc. Symp. Models for the Perception of Speech and Visual Form*, pages 362–368, 1967.
- [30] H. Blum. Biological shape and visual science (part 1). *Journal of Theoretical Biology*, (38):205–287, 1973.
- [31] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D-U. Hwang. Complex networks : Structure and dynamics. *Phys. Rep.*, 424(4-5):175–308, 2006.
- [32] F. L. Bookstein. Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–243, 1997.
- [33] F.L. Bookstein. Size and shape spaces for landmark data in two dimensions (with discussion) statist. sci. *Journal of the Royal Statistica Society, Series B (Methodological)*, 4:181–242, 1986.
- [34] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition - Workshop on Perceptual Organization in Computer Vision*, 2004.
- [35] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. *Proc. ECCV 2002*, pages 109–122, 2002.
- [36] I. Borg and P. Groenen. Modern multidimensional scaling: theory and applications. *Springer-Verlag*, pages 337–379, 1997.
- [37] D. L. Borges and R. B. Fisher. Class-based recognition of 3d objects represented by volumetric primitives. *Image and Vision Computing*, 15(8):655–664, 1997.

- [38] J. F. Boyce and W. J. Hossak. Moment invariant for pattern recognition. *Pattern Recognition Letters*, 1:451–456, 1983.
- [39] Y. Boycov and G. Funka-Lea. Object extraction via constrained graph-cuts. *International Journal of Computer Vision*, 2005.
- [40] Y. Boycov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.
- [41] Y. Boycov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. *International Conference on Computer Vision*, 1:26–33, 2003.
- [42] Y. Boycov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on PAMI*, 26(9):1124–1137, 2004.
- [43] Y. Boycov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on PAMI*, 23:1222–1239, 2001.
- [44] J. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. *Proceeding of IEEE conference on Computer Vision and Pattern Recognition*, 1998.
- [45] C. Brechbühler, G. Gerig, and O. Kübler. Parameterization of closed surfaces for 3-d shape description. *CVGIP: Image Understanding*, 61:154–170, 1995.
- [46] Michael C. Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II*, pages 628–641, London, UK, 1998. Springer-Verlag.
- [47] R. Campbell and P. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding (CVIU)*, 81(2):166–210, 2001.
- [48] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [49] G. Carneiro and A. Jepson. Multi-scale phase-based local features. *CVPR*, 1:736–743, 2003.
- [50] M. A. Carreira-Perpinan. Continuous latent variable models for dimensionality reduction and sequential data reconstruction. *PhD Thesis - Dept. of Computer Science, University of Sheffield, UK*, 2001.

- 
- [51] M. A. Carreira-Perpinan and C. Williams. On the number of modes of a gaussian mixture. *Proc. 4th Int'l Conference on Scale-Space Theories in Computer Vision*, pages 625–640, 2003.
- [52] T. F. Chan. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2), 2001.
- [53] Lin Chen. The ological approach to perceptual organization. *Visual Cognition*, 12(4):553–637, 2005.
- [54] X. R. Chen, Z. W. Tu, A. L. Yuille, and S. C. Zhu. Image parsing: Segmentation, detection and recognition. *Proc. ICCV*, page 1825, 2003.
- [55] C. Cheng. Scene segmentation and object classification for place recognition. *PhD diss., University of Tennessee*, 2010.
- [56] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.
- [57] Y. Cheng and K.S. Fu. Conceptual clustering in knowledge organization. *IEEE Truns. Puttern Analysis und Muchine Intelligence*, 7:592–598, 1985.
- [58] N. N. Chentsov. Estimation of unknown probability density based on observations (in russian). *Dokl. Akad. Nauk SSSR*, pages 45–48, 1962.
- [59] D. Chetverikov and Y. Khenokh. Matching for shape defect detection. *Lecture Notes in Computer Science*, 1689:367–374, 1999.
- [60] H. I. Choi, S. W. Choi, and H. P. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181(1):57–88, 1997.
- [61] G. Christensen, R. D. Rabbitt, and M. I. Miller. A deformable neuroanatomy textbook based on viscous fluid mechanics. *roceedings of 27th Conference on Information Sciences and Systems*, pages 211–216, 1993.
- [62] C. S. Chua and R. Jarvis. Point signatures: a new representation for 3d object recognition. *nternational Journal of Computer Vision*, 25(1):63–85, 1997.
- [63] R. Collins. Mean-shift blob tracking through scale space. *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 234–240, 2003.
- [64] D. Comaniciu and P. Meer. Mean shift analysis and applications. *Proc. International Conference on Computer Vision*, pages 1197–1203, 1999.

- [65] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [66] T. F. Cootes, C. Beeston, G. J. Edwards, and C. J. Taylor. A unified framework for atlas matching using active appearance models. *Proceedings of IPMI'99: Information Processing in Medical Imaging*, pages 322–333, 1999.
- [67] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. *Proceedings of British Machine Vision Conference*, pages 9–18, 1992.
- [68] A. Corduneanu and C. M. Bishop. Variational bayesian model selection for mixture distributions. In *In Proc. 8-th Intern. Conf. on Artificial Intelligence and Statistics*, pages 27–34, 2001.
- [69] F. Crosilla. Procrustes analysis and geodetic sciences. *Quo vadis geodesia?. Technical Reports, Department of Geodesy and GeoInformatics*, pages 69–78, 1999.
- [70] F. Crosilla and A. Beinat. Use of generalized procrustes analysis for the photogrammetric block adjustment by independent models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 56(3):195–209, 2002.
- [71] J. G. Csernansky, S. Joshi, L. Wang, J. M. Haller, M. Gado, J. P. Miller, U. Grenander, and M. I. Miller. Hippocampal morphometry in schizophrenia by high dimensional brain mapping. *Proceedings of National Academy of Science*, 95(19):11406–11411, 1998.
- [72] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiway cuts. *ACM Symposium on Theory of Computing*, pages 241–251, 1992.
- [73] Sanjoy Dasgupta. Learning mixtures of Gaussians. In *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, page 634, Washington, DC, USA, 1999. IEEE Computer Society.
- [74] C. Davatzikos, M. Vaillant, S. Resnick, J. L. Prince, S. Letovsky, and R. N. Bryan. A computerized method for morphological analysis of the corpus callosum. *Journal of Computer Assisted Tomography*, pages 88–97, 1996.
- [75] E.R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*,. Academic Press, New York, 1997.

- [76] A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):121–132, 1997.
- [77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.
- [78] M. Devrim. Generalized procrustes analysis and its applications in photogrammetry. 2003.
- [79] S. Dickinson, R. Bergevin, I. Biederman, J. Eklundh, R. Munck-Fairwood, A. Jain, and A. Pentland. Panel report: The potential of geons for generic 3-d object recognition, 1997.
- [80] S. Dickinson, A. Pentland, and A. Rosenfeld. 3d shape recovery using distributed aspect matching. *PAMI*, 14(2):174–198, 1992.
- [81] S.J. Dickinson, R. Bergevin, I. Biederman, J. O. Eklundh, R. Munck-Fairwood, A.K. Jain, and A. Pentland. Panel report: the potential of geons for generic 3-d object recognition. *Image and Vision Computing*, 15(4):277–292, 1997.
- [82] P. Dollar, Z. W. Tu, and S. Belongie. Supervised learning of edges and object boundaries. *Proc. CVPR*, 2:1964–1971, 2006.
- [83] I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, 1998.
- [84] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons Inc., 2001.
- [85] R. Dugad and V. Desai. Image interpretation using hidden markov models. *Proceedings of International Conference on Information, Communications and Signal Processing (ICICS)*, 3:1532–1536, 1997.
- [86] S. Edelman. Computational theories of object recognition. pages 296–304, 1997.
- [87] T. Eiter and H. Mannila. Computing discrete fréchet distance. *Technical Report Technische Universität Wien*, 1(4), 1994.
- [88] J. Elder and S. W. Zucker. The effect of contour closure on the rapid discrimination of two-dimensional shapes. *Vision Resolution*, 33:981–991, 1993.
- [89] J. Elder and S. W. Zucker. A measure of closure. *Vision Resolution*, 34(24):3361–3369, 1994.



- [90] M. Everingham, L. Van Gool, Williams C. K. I., J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int'l Journal of Computer Vision*, 88:303–338, 2010.
- [91] T. Fan, G. Medioni, and R. Nevatia. Recognizing 3d objects using surface descriptions. *IEEE Trans. on Pattern Analysis and Mach. Int. (PAMI)*, 11(11):1140–1157, 1989.
- [92] M. Fashing and C. Tomasi. Mean shift is a bound optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3), 2005.
- [93] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.
- [94] R. Fergus. Visual object category recognition. Thesis, University of Oxford, 2005.
- [95] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR (2)*, pages 264–271, 2003.
- [96] Robert Fergus, Pietro Perona, and Andrew Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR (1)*, pages 380–387, 2005.
- [97] R. A. Fisher. The use of multiple measurement in taxonomic problems. *Ann. Eugenicis*, 7:179–188, 1936.
- [98] L.M.J. Florack, B.M.t. Haar Romeny, J.J. Koenderink, and M.A. Viergever. General intensity transformations and differential invariants. *JMIV*, pages 171–187, 1994.
- [99] J. A. Fodor. *The modularity of mind*. MIT Press, 1983.
- [100] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- [101] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput*, 10:260–268, 1961.
- [102] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *PAMI*, 13:891–906, 1991.
- [103] K.S. Fu. *Syntactic Methods in Pattern Recognition*. Academic Press, New York, 1974.
- [104] B. Fuglede and F. Topsøe. Jensen-shannon divergence and hilbert space embedding.
- [105] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. on Information Theory*, 21(1):32–40, 1975.

- [106] K. Fukunaga and L.D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 1975.
- [107] T. Funkhouser, P. Min, M. Kazhdan, T. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3d models. *ACM Transactions on Graphics (TOG)*, pages 83–105, 2003.
- [108] R. Garnett, T. Huegerich, and C. Chui. Universal noise removal algorithm with an impulse detector. *IEEE Trans. on Image Processing*, 14(11):1747–1754, 2005.
- [109] D. Geiger, T. Liu, and R. V. Khon. Representation and self-fimilarity of shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 25, 2003.
- [110] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on PAMI*, 6:721–741, 1984.
- [111] J. E. Gentle. *Elements of Computational Statistics*. Springer New York, 2002.
- [112] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The amsterdam library of object images. *Int'l Journal of Computer Vision*, 1(61):103–112, 2005.
- [113] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940, 1988.
- [114] P. Golland. Statistical shape analysis of anatomical structures. phd thesis. *Massachusetts Institute of Technology*, 2001.
- [115] P. Golland and W. E. L. Grimson. Fixed topology skeletons. *Proceedings of CVPR'2000*, pages 10–17, 2000.
- [116] I. J. Good and R. A. Gaskins. Density estimation atad bumphunting by the penalized likelihood method exemplified by scatering atad meteorit date. *J. Amer. Statist. Assoc.*, 75:42–73, 1980.
- [117] Colin Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society*, 53(2):285–3399, 1991.
- [118] S. Gould, J. Rogers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *IJCV*, 2008.
- [119] J.C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

- [120] J.C. Gower and G.B. Dijkstra. Procrustes problems. *Oxford University Press*, 2004.
- [121] L. Grady. Space-variant computer vision: a graph-theoretic approach. *PhD Thesis, Boston, MA*, 2004.
- [122] L. Grady and G. Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, ECCV 2004 Workshops CVAMIA and MMBIA*, pages 230–245, 2004.
- [123] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.
- [124] W. E. L. Grimson. *Object recognition by computer: the role of geometric constraints*. MIT Press, Cambridge, Massachusetts, 1990.
- [125] W.I. Groskey and R. Mehrotra. Index-based object recognition in pictorial data management. *Comput. Vision Graphics Image Process*, 52:416–436, 1990.
- [126] W.I. Groskey, P. Neo, and R. Mehrotra. A pictorial index mechanism for model-based matching. *Data Knowledge Eng.*, 8:309327, 1992.
- [127] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *Proc. Int'l Journal of Computer Vision*, 20(1-2), 1996.
- [128] J. Hammersley and P. Clifford. *Markov field on finite graph and lattices*. Preprint University of California, Berkeley, 1971.
- [129] W. Hardle, M. Muller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models, an introduction*. 2004.
- [130] C. Harris and M. Stephens. A combined corner and edge detector. *Proc. of the Alvey Vision Conference*, pages 147–151, 1988.
- [131] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of Statistical Learning - Data Mining, Inference and Prediction*. Springer, 2001.
- [132] Tarr Michael J. Hayward, William G. Testing conditions for viewpoint invariance in object recognition. *Journal of Experimental Psychology, Human Perception and Performance*, 1997.

- [133] M. Heiler, J. Keuchel, and C. Schnrr. Semidefinite clustering for image segmentation with a-priori knowledge. *Proc. of DAGM-Symposium'2005*, pages 309–317, 2005.
- [134] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, 1:657–662, 2001.
- [135] R. Herbrich. *Learning Kernel Classifiers*. MIT Press, 2002.
- [136] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed form solution of absolute orientation using orthonormal matrices. *Journal of Optical Society of America*, 5(7):1128–1135, 1988.
- [137] M. K. Hu. Visual pattern recognition by moments invariant. *IRE Transactions on Information Theory*, 8:179–187, 1962.
- [138] C.-L. Huang and D.-H. Huang. A content-based image retrieval system. *Image Vision Comput.*, 16:149–163, 1998.
- [139] D. Huber, A. Kapuria, R. Donamukkala, and M. Hebert. Parts-based 3d object classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, 2004.
- [140] D.P. Huttenlocher and W.J. Rucklidge. A multi-resolution technique for comparing images using the hausdorff distance. *Technical Report, TR-92-1321, Department of Computer Science, Cornell University*, 1991.
- [141] J. Hwang. Non-parametric multivariate density estimation: a complete study. *IEEE Trans. Signal Processing*, pages 2795–2810, 1994.
- [142] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Transactions on PAMI*, 25(10):1333–1336, 2003.
- [143] A. K. Jain. *Fundamentals of digital image processing*. Prentice Hall, 1989.
- [144] T. Jebara and K. Kondor. Bhattacharyya and expected likelihood kernels. *Conference on Learning Theory, COLT/KW*, 2003.
- [145] T. Jebara, K. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research, JMLR, Special Topic in Learning Theory*, 5:819–884, July 2004.
- [146] Tony Jebara. Image as bag of pixels. In *International Conference on Computer Vision (ICCV)*, 2003.

- [147] H. Jeffreys. An invariant form for the prior probability in estimation problems. In *Proceedings of Royal Society A*, volume 186, pages 453–461, 1946.
- [148] H. Jeffreys. *Theory of probability*. Oxford University Press, 1948.
- [149] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(10):1075–1088, 2001.
- [150] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. on Pattern Analysis and Mach. Int. (PAMI)*, 21(5):433–449, 1999.
- [151] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.
- [152] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *ECCV (1)*, pages 228–241, 2004.
- [153] G. Kaniza. Contours without gradients or cognitive contours. *Italian journal of Psychology*, 1:93–112, 1971.
- [154] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Proc. Int'l Journal of Computer Vision*, 1(4):321–331, 1988.
- [155] M. Kazhdan. Shape representations and algorithms for 3d model retrieval. phd thesis. *Princeton University*, 2004.
- [156] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. *Symposium on Geometry Processing*, pages 167–175, 2003.
- [157] A. Kelemen, G. Székely, and G. Gerig. Three-dimensional model-based segmentation. *Proceedings of IEEE International Workshop on Model Based 3D Image Analysis*, pages 87–96, 1998.
- [158] D.G. Kendall. The diffusion of shape. *Advances in applied probability*, 9:428–430, 1977.
- [159] D.G. Kendall. A survey of statistical theory of shape. *Statistical Sciences*, 4:87–120, 1989.

- [160] B.B. Kimia, A.R. Tannenbaum, and S.W. Zucker. Shapes, shocks, and deformations. *Int. J. Comput. Vision*, 15:189–224, 1995.
- [161] D. Kirsanov and S. Gortier. A discrete global minimization algorithm for continuous variational problems. *Harvard Computer Science Technical Report*, TR-14-04, 2004.
- [162] W. Knoontz, P. M. Narendra, and K. Fukunaga. A graph theoretic approach to non-parametric cluster analysis. *IEEE Trans. on Computers*, 25:936–944, 1976.
- [163] J. Koenderink and A. van Doorn. The internal representation of solid shape with reference to vision. *Biological Cybernetics*, 32:211 – 216, 1979.
- [164] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on PAMI*, 26(2):147–159, 2004.
- [165] M.A. Koschat and D.F. Swayne. A weighted procrustes criterion. *Psychometrika*, 56(2):229–239, 1989.
- [166] J.C. Krivic and F. Solina. Part-level object recognition using superquadrics. *Computer Vision and Image Understanding*, 95(1):105–126, 2004.
- [167] R. Kronmal and M. Tarter. The estimation of probability densities and cumulative by fourier series methods. *American Statist. Association J.*, pages 925–952, 1968.
- [168] K. S. Kumar and U. B. Desai. Joint segmentation and image interpretation. *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 3:853–856, 1996.
- [169] F. Kurugollu, B. Sankur, and A. E. Harmanci. Color image segmentation using histogram multithresholding and fusion. *Image and Vision Computing*, 19:915–928, 2001.
- [170] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. *Proceedings of CVPR'2000: Computer Vision and Pattern Recognition*, pages 316–323, 2000.
- [171] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. *ECCV*, pages 581–594, 2006.
- [172] Frederic Leymarie and Martin D. Levine. Simulating the grassfire transform using an active contour model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(1):56–75, 1992.

- [173] S.Z. Li. Shape matching based on invariants. *in: O. Omidvar (Ed.), Shape Analysis, Progress in Neural Networks*, 6:203–228, 1999.
- [174] S. X. Liao and M. Pawlak. On image analysis by moments. *IEEE Transactions on PAMI*, 18(3):254–266, 1996.
- [175] Y. W. Lim and S. U. Lee. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means technique. *Pattern Recognition*, 23:935–952, 1990.
- [176] D. Lin. An information-theoretic definition of similarity. *Proc. 15th International Conf. on Machine Learning*, pages 296–304, 1998.
- [177] J. Lin. Divergence measure based on the shannon entropy. *IEEE Transactions on Information theory*, 37:145–151, 1991.
- [178] T. Lindeberg. Feature detection with automatic scale selection. *Int. Journal of Computer Vision*, 60:79–116, 1998.
- [179] R.W. Lissitz, P.H. Schoenemann, and J.C. Lingoes. A solution to the weighted procrustes problem in which the transformation is in agreement with the loss function. *Psychometrika*, 41(1):547–550, 1976.
- [180] L. Liu and S. Sclaroff. Region segmentation via deformable model-guided split and merge. *ICCV*, 1:98–104, 2001.
- [181] D. O. Loetsgaarden and C. P. Quesenberry. A nonparametric estimate of a multivariate density function. *Ann. Math. Statist*, 36:1049–1051, 1965.
- [182] D. Lowe. Distinctive image features from scale-invariant keypoints, cascade filtering approach. *IJCV*, 60:91–110, 2004.
- [183] S. Lyu. Kernel for unordered sets: the gaussian mixture approach. *European Conference on Machine Learning, ECML*, 2005.
- [184] A. M. C. Machado and J. C. Gee. Atlas warping for brain morphometry. *Proceedings of SPIE*, pages 642–651, 1998.
- [185] S. Maitra. Moments invariant. *Proceedings of the IEEE*, 67(4):697–699, 1979.
- [186] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. In *Proc. R. Soc. Lond. B*, vol. 200, pages 269–294, 1978.

- [187] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman, San Francisco, 1982.
- [188] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
- [189] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.
- [190] J. Martin, A. Pentland, and R. Kikinis. Shape analysis of brain structures using physical and experimental models. *Proceedings of CVPR'94*, pages 752–755, 1994.
- [191] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *13th BMVC*, pages 384–393, 2002.
- [192] J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception: part 1. an account of basic findings. *Psychological Review*, 88:375–407, 1981.
- [193] G.J. McLachlan and D. Peel. MIXFIT: an algorithm for the automatic fitting and testing of normal mixture models. In *Proceedings of the 14th International Conference on Pattern Recognition*, pages 553–557, 1998.
- [194] G. McNeill and S. Vijayakumar. 2d shape classification and retrieval. *Proceedings of the Nineteenth IJCAI*, pages 1483–1488, 2005.
- [195] G. Medioni and A. François. 3-d structures for generic object recognition. *Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00)*, pages 1030–1037, 2000.
- [196] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe. Curious george: An attentive semantic robot. *Robotics and Autonomous Systems Journal*, 2008.
- [197] T. Meier. Segmentation for video object plane extraction and reduction of coding artifacts. *PhD Thesis, Department of Electrical and Electronic Engineering, University of Western Australia*, 1998.



- [198] S. Mika, G. Ratsch, J. Weston, and B. Scholkopf. Fisher discriminant analysis with kernels. In *Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas (Eds.), Neural Networks for Signal Processing IX*, pages 41–48, 1999.
- [199] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. *ICCV*, 1:525–531, 2001.
- [200] P. Min. A 3d model search engine. phd thesis. *Princeton University*, 2003.
- [201] J. W. Modestino and J. Zhang. Markov random field model-based approach to image interpretation. *Chellappa, R. and Jain, A. (eds): Markov Random Fields: Theory and Application. Academic Press*, pages 369–408, 1993.
- [202] M. C. Mozer, R. S. Zemel, and M. Behrmann. Discovering and using perceptual grouping principles in visual information processing. *Proc. of the 14th Annual Conference of the Cognitive Science Society*, pages 283–288, 1992.
- [203] M. C. Mozer, R. S. Zemel, M. Behrmann, and K. C. I. Williams. Learning to segment images using dynamic feature binding. *Neural Computation*, 4:650–655, 1992.
- [204] K. R. Muller, S. Mika, K. Ratsch, G. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, 2001.
- [205] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math*, 42(4), 1989.
- [206] R. Murrieta-Cid, Briot M., and N. Vandapel. Visual navigation in natural environments: from range and color data to a landmark-based model. *Autonomous Robots*, 13:143–168, 2002.
- [207] Cencov N. Evaluation of data unknown distribution density from observations. *Soviet Math.*, 3:1559–1562, 1962.
- [208] H.H. Bülthoff N. K. Logothetis, J. Pauls and T. Poggio. View-dependent object recognition by monkeys. *Current Biology*, 4:401–414, 1994.
- [209] E. A. Nadaraya. Non-parametric estimation of probability densities and regression curves. *Mathematics Applications*, 1989.
- [210] A. Needham. Object recognition and object segregation in 4.5-month-old infants. *Journal of Experimental Child Psychology*, 78:324, 2001.

- [211] A. Needham and R. Baillargeon. Effects of prior experience in 4.5-month-old infants object segregation. *Infant Behaviour and Development*, 21:124, 1998.
- [212] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 2:167–256, 2003.
- [213] Executives of RoboCup Rescue. Robocup rescue. Online; lastchecked on February 2011.
- [214] Executives of RoboCup@Home. Robocup@home. Online; lastchecked on February 2011.
- [215] Executives of the Semantic Robot Vision Challenge. Semantic robot vision challenge. Online; lastchecked on Decembre 2010.
- [216] R.L. Ogniewicz. A multiscale mat from voronoi diagrams: the skeleton-space and its application to shape description and decomposition. *Aspect of Visual Form Processing - 2nd Int'l Workshop on Visualm Form*, pages 430–439, 1994.
- [217] S. E. Palmer. The effects of contextual scenes on the identification of objects. *Memory Cognition*, 3:519–526, 1975.
- [218] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.
- [219] N. Paragios, Y. Chen, and O. Faugeras. *Handbook of Matematical Models in Computer Vision*. Springer New York, 2006.
- [220] P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(8), 1989.
- [221] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statistics*, 33:1065–1076, 1962.
- [222] M. Pawlak. On the reconstruction aspects of moment descriptors. *IEEE Transactions on Information Theory*, 38(6):1698–1708, 1992.
- [223] X. Pennec, N. Ayache, and J. P. Thirion. Landmark-based registration using features identified through differential geometry. *Handbook of Medical Imaging*, pages 499–513, 2000.
- [224] A. Pentland. Recognition by parts. *n Proceedings of the First International Conference on Computer Vision (ICCV'87)*, pages 612–620, 1987.

- [225] Alex Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(2):293–331, 1986.
- [226] M. A. Peterson. Object recognition processes can and do operate before figureground organization. *Current Directions in Psychological Science*, 3(4):105–111, 1994.
- [227] M. A. Peterson and B. S. Gibson. Shape recognition contributions to figure-ground organization in three-dimensional displays. *Cognitive Psychology*, 25:383–429, 1993.
- [228] M. Peura and J. Iivarinen. Efficiency of simple shape descriptors. *Proc. of the Third Int'l Workshop on Visual Form Comput. Graphics Image Processing*, pages 443–451, 1997.
- [229] Fiora Pirri. The usual objects: a first draft on decomposing and reassembling familiar objects images. In *Proceedings of XXVII Annual Conference of the Cognitive Science Society*, pages 1773–1778, 2005.
- [230] Fiora Pirri and Massimo Romano. 2d qualitative recognition of symgeon aspects. In *Proc. KES 2003*, volume 2774 of *Lecture Notes in Computer Science*, pages 1187–1194. Springer, 2003.
- [231] I. Pitas. *Digital image processing algorithms and applications*. Wiley-IEEE, 2000.
- [232] Z. Pizlo. *3D Shape - Its Unique Place in Visual Perception*. The MIT Press, 2008.
- [233] A.R. Pope and D.G. Lowe. Learning object recognition models from images. In *ICCV93*, pages 296–301, 1993.
- [234] W. Prinzmetal and M. Millis-Wright. Cognitive and linguistic factors affect visual feature integration. *Cognitive Psychology*, 16:305–340, 1984.
- [235] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *Computer Vision, Graphics and Image Processing*, 54(5):438–460, 1992.
- [236] M. Pujol, R. Rizo, P. Arques, P. Compan, F. Escolano, R. Molina, and F. Pujol. Application de los modelos de campos aleatorios de markov en vision artificial. *Revista Electronica de Vision por Computador*, 4:1–23, 2000.
- [237] N. Raja and A. Jain. Recognizing geons from superquadrics fitted to range data. *Image and Vision Computing*, 10(3):179–190, 1992.

- [238] D. Ramanan and D.A. Forsyth. Finding and tracking people from the bottom up. *Proc. IEEE International Conf. Computer Vision and Pattern Recognition*, pages 467–474, 2003.
- [239] D. Ramanan and D.A. Forsyth. Using temporal coherence to build animal models. *Proc. Internrtional Conf. Computer Vision*, pages 338–346, 2003.
- [240] G. M. Reicher. Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of Experimental Psychology*, 81:274–280, 1969.
- [241] Ehud Rivlin, Sven J. Dickinson, and Azriel Rosenfeld. Recognition by functional parts. *Computer Vision and Image Understanding: CVIU*, 62(2):164–176, 1995.
- [242] M. Roseblatt. Remarks on some non-parametric estimates of a density function. *Ann. Math. Statist.*, 38(1):832–837, 1956.
- [243] V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. In S. A. Solla, T. K. Leen, and K. R. Müller (Eds.), *Advances in Neural Information Processing Systems*, 12, 2000.
- [244] W.J. Rucklidge. Efficient locating objects using hausdorff distance. *Int.l Journal of Computer Vision*, 24(3):251–270, 1997.
- [245] J. C. Russ. *The image processing handbook*. CRC PRes, 2002.
- [246] P.H. Schoenemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, v(1):1–10, 1966.
- [247] P.H. Schoenemann and R. Carroll. Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika*, 35(2):245–255, 1970.
- [248] B. Scholkopf, C. Burges, and V. Vapnik. Incorporating invariances in support vector learning machines. In C. von der Malsburg, W. von Seelen, J. C. Vorbruggen and B. Sendhoff (Eds.), *Artificial neural networks - ICANN96*, 1112:47–52, 1996.
- [249] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
- [250] B. Scholkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.
- [251] S. C. Schwartz. Estimation of probability density by an orthogonal series. *Ann. Math. Statist.*, pages 1261–1265, 1967.

- [252] Stan Sclaroff and Alex Pentland. Modal matching for correspondence and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(6):545–561, 1995.
- [253] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):550–571, 2004.
- [254] L. G. Shapiro, J. D. Moriarty, R. M. Haralick, and P. G. Mulgaonkar. Matching three-dimensional objects using a relational paradigm. *Pattern Recognition*, 17(4):385–405, 1984.
- [255] E. Sharon and David Mumford. 2d-shape analysis using conformal mapping. In *CVPR* (2), pages 350–357, 2004.
- [256] A. Shashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. *Proc. Second Int'l Conference on Computer Vision*, 1988.
- [257] J. Shawe-Taylor and N. Cristianini. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, 2000.
- [258] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [259] J. Shi and J. Malik. Normalized cut and image segmentation. *IEEE Transactions on PAMI*, 22(8), 2000.
- [260] P. Shilane, M. Kazhdan, P. Min, and T. Funkhouser. The princeton shape benchmark. *Proc. Shape Modeling International*, 2004.
- [261] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout and context. *IJCV*, 81(1):2–23, 2009.
- [262] K. Siddiqi, S. Bouix, A.R. Tannenbaum, and S.W. Zucker. The hamilton-jacobi skeleton. *Proc. Int. Conf. Computer Vision*, 2:828–834, 1999.
- [263] Kaleem Siddiqi and Benjamin B. Kimia. A shock grammar for recognition. In *CVPR*, pages 507–513, 1996.
- [264] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1993.

- [265] M. V. Smirnov. On the approximation of probability densities of random variables (in russian). *Scholarly Notes of Moscow State Polytechnical Institute*, pages 69–96, 1951.
- [266] F. W. Smith and M. H. Wright. Automatic ship photo interpretation by the method of moments. *IEEE Transactions on Computers*, 20(9):1089–1095, 1971.
- [267] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman Hall, London, UK, NJ, 1993.
- [268] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and machine vision*. PWS, 1998.
- [269] D.M. Squire and T.M. Caelli. Invariance signature: characterizing contours by their departures from invariance. *Comput. Vision Image Understanding*, 77:284–316, 2000.
- [270] Anuj Srivastava, Shantanu H. Joshi, Washington Mio, and Xiuwen Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):590–602, 2005.
- [271] L. Staib and J. Duncan. Boundary finding with parametrically deformable models. *IEEE PAMI*, 14(11):1061–1075, 1992.
- [272] F. Stein and G. Medioni. Structural indexing: efficient 3d object recognition. *IEEE Trans. on Pattern Analysis and Mach. Int. (PAMI)*, 14(2):125–145, 1992.
- [273] G. Strang. Maximal flow through a domain. *Mathematical Programming*, 26:123–143, 1983.
- [274] B.J. Super. Fast correspondence-based system for shape-retrieval. *Patt. Recog. Lett.*, 25(1):217–225, 2004.
- [275] G. Székely, A. Kelemen, C. Brechbühler, and G. Gerig. Segmentation of 2d and 3d objects from mri volume data using constrained elastic deformations of flexible fourier contour and surface models. *Medical Image Analysis*, 1(1):19–34, 1996.
- [276] M. Tappen and W. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. *IEEE Int’l Conference on Computer Vision (ICCV)*, 2003.
- [277] M. R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8):920–930, 1980.

- [278] C. H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Transactions on PAMI*, 10(4):496–513, 1988.
- [279] J.M.F. Ten Berge. Orthogonal procrustes rotation for two or more matrices. *Psychometrika*, 42(2):267–276, 1977.
- [280] Demetri Terzopoulos and Kurt W. Fleischer. Deformable models. *The Visual Computer*, 4(6):306–331, 1988.
- [281] F. Topsøe. Some inequalities for information divergence and related measures of discrimination. *Journal of Inequalities in Pure and Applied Mathematics*, 2(1), 1999.
- [282] A. Torsello. Matching hierarchical structures for shape recognition. *Ph.D. Thesis, University of York*, 2004.
- [283] O. D. Trier, A. K. Jain, and T. Taxt. A trainable system for object detection feature extraction methods for character recognition. a survey. *Pattern Recognition*, 29:641–701, 1996.
- [284] S. Ullman. Aligning pictorial descriptions: an approach to object recognition. *Cognition*, 32:193–254, 1989.
- [285] S. Ullman and E. Sali. Object classification using a fragment-base representation. *Biologically Motivated Computer Vision*, pages 73–87, 2000.
- [286] S. E. Umbaugh. *Computer imaging: digital image analysis and processing*. CRC Press, 2005.
- [287] R. Urquhart. Graph theoretical clustering based on limited neighbourhood sets. *Pattern Recognition*, 15(3):173–187, 1982.
- [288] P.J. van Otterloo. *A Contour-Oriented Approach to Shape Analysis*. Prentice-Hall International (UK) Ltd, Englewood Cliffs, NJ, 1991.
- [289] I. Van Ryzin. On strong consistency of density estimates. *Ann. Math. Statist.*, 40(1):1765–1772, 1969.
- [290] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition*. 1974.
- [291] S. P. Vecera and Farah M. J. Is visual image segmentation a bottom-up or an interactive process? *Perception Psychophysics*, 59(8):1280–1296, 1997.

- [292] V. R. Vijaykumar, P. T. Vanathi, and P. Kanagasabapathy. Fast and efficient algorithm to remove gaussian noise in digital images. *Int'l Journal of Computer Science*, 37(1), 2009.
- [293] R. Vio, G. Fasano, M. Lazzarin, and O. Lessi. Probability density estimation in astronomy. *Astron. Astrophys.*, pages 640–648, 1994.
- [294] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, page 511518, 2001.
- [295] Nikos Vlassis and Aristidis Likas. A greedy EM algorithm for Gaussian mixture learning. *Neural Process. Lett.*, 15(1):77–87, 2002.
- [296] G. Wahba. Optimal convergence properties of variable knot, kernel, orthonormal series methods for density estimation. *Ann. Statist.*, 3:15–29, 1975.
- [297] S. J. Wan, P. Prusinkiewicz, and S. K. M. Wong. Variance-based color image quantization for frame buffer display. *Color Research and Application*, 15:52–58, 1990.
- [298] S. Wang, T. Kubota, and T. Richardson. Shape correspondence through landmark sliding. *IEEE Conf. on Comp. Vis. and Patt. Recog.*, 1:143–150, 2004.
- [299] G. S. Watson and M. R. Leadbetter. On the estimation of the probability density. *Ann. Math. Statist.*, pages 480–491, 1963.
- [300] Markus Weber, Max Welling, and Pietro Perona. Unsupervised learning of models for recognition. In *ECCV (1)*, pages 18–32, 2000.
- [301] M. Wertheimer. Laws of organization in perceptual forms. In W. D. Ellis, editor, *A Sourcebook of Gestalt Psychology*, pages 331–363. Harcourt, Brace and Company, 1939.
- [302] D. D. Wheeler. Processes in word recognition. *Cognitive Psychology*, 1:59–85, 1970.
- [303] Kenong Wu and Martine Levin. 3D object representation using parametric geons. Technical Report CIM-93-13, CIM, 1993.
- [304] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on PAMI*, 15(11):1101–1113, 1993.



- 
- [305] J. Wyngaerd, L. V. Gool, R. Kock, and M. Proesmans. Invariant-based registration of surface patches. *Intl. Conf. on Computer Vision (ICCV)*, pages 301–306, 1999.
- [306] W. Xing, W. Liu, and B. Yuan. 3d object classification based on volumetric parts. *International Journal of Cognitive Informatics and Natural Intelligence*, 2(1), 2008.
- [307] I. Yong, J. Walker, and J. Bowie. An analysis technique for biological shape. *Comput. Graphics Image Processing*, 25:357–370, 1974.
- [308] S. X. Yu, R. Gross, and J. Shi. Concurrent object recognition and segmentation by graph partitioning. *NIPS*, 2002.
- [309] C. T. Zahn. Graph-theoretic methods for detecting and describing gestalt clusters. *IEEE Transactions on Computing*, 20:68–86, 1971.
- [310] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.
- [311] D.S. Zhang and G. Lu. A comparative study of fourier descriptors for shape representation and retrieval. *Proceedings of the Fifth Asian Conference on Computer Vision (ACCV02)*, pages 646–651, 2002.
- [312] J. Zhang, X. Zhang, H. Krim, and G.G. Walter. Object representation and recognition in shape spaces. *Patt. Recog.*, 36:1143–1154, 2003.
- [313] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing and bayes/mdl for multi-band image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.