



**Universitat  
Autònoma  
de Barcelona**

# **Contributions to the Content-Based Image Retrieval Using Pictorial Queries**

A dissertation submitted by **Agnés Borràs Angosto** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctora en Informàtica**.

Bellaterra, September 2009

Advisor: **Dr. Josep Lladós Canet**  
Computer Vision Center, Universitat Autònoma de Barcelona.



---

This document was typeset by the author using L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

Copyright © 2009 by Agnès Borràs Angosto. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author.

ISBN 978-84-937261-0-2

Printed by Ediciones Gráficas Rey, S.L.

Als meus avis



# Acknowledgment

Aquest treball no hauria estat possible sense la confiança dipositada pel meu director de tesi, Josep Lladós. A ell li vull donar les gràcies pel temps que m'ha dedicat dins d'una agenda, com a mínim, maratoniana :). També voldria agrair al Juanjo Villanueva les tasques en la gestió de les beques i totes les facilitats que m'ha posat en l'estada en aquest centre.

Quan algú em pregunta què m'ha semblat aquesta etapa, el primer que contesto és que m'ho he passat molt bé. I si ha estat així és per tota la gent que he anat coneixent al llarg d'aquests anys.

Qui primer vaig trobar treballant en el Fisonomies va ser el Francesc :D. Allò va ser l'inici d'una època que no es pot resumir en una sola frase ;). Només puc dir que, encara que ja fa temps que no ronda pel CVC, tinc la sort de saber que sigui on sigui sempre està a prop.

Aleertaaaaa!! sense l'Enric tampoc hauria estat el mateix, ni de lluny! Gràcies a les seves funcions de "botones", al seu contracte miraculós i a la seva paciència infinita, és sense dubte el millor suport que algú pot trobar.

També he tingut el luxe de compartir la tesi amb l'Àgata, una admiradora del color taronja i, sobretot, una gran confessora que sempre està a l'alçada.

A en Jaume li dec els grans moments de l'hora de dinar: lliçons sobre el petit guany de la borsa, múltiples performances i la demostració diària de que és possible menjar sense respirar :).

Si en Jaume va apadrinar el Projecte Tivalta, la Débora es va fer càrrec del Proyecto Sonrisas. Sense els seus comentaris els dinars haurien perdut grans temes de conversa i, per un moment, també em podria haver pensat que el meu dinar feia bona pinta.

Dins d'aquest "grup de gent honrada" també ha estat important comptar amb la presència de l'Aura i en Ricard (que, per cert, hauran decidir qui dels dos ens acull a casa seva per veure l'últim capítol de Lost :)).

Hi ha una llista ben llarga de gent del amb qui he pogut compartir molt bons moments tant dins com fora del CVC: l'Anton (sé que els comentaris que li agradaria llegir no es poden posar en un document públic ;)), l'Adrià (aquell petit ésser diabòlic que sempre em lia per anar a qualsevol lloc), el Carles (co-inventor del badmigpong), el Xevi (l'amo d'un canvi de marxas com cal), el Sergio (pez-pescao), l'Alicia (la reina

de les estadístiques), en Batlle (qui millor interpreta la previsió meteorològica), la Carme (que ens va fer descobrir la discomòbil de Batea), l'Anna (sempre a punt) i en David (que tot sovint guanyava les partides de Revolt).

Tampoc voldria oblidar-me de tots aquells qui m'han ajudat en algun moment o altre, ja sigui deixant-me un tros de codi o comentant alguna idea: l'Oriol, el Robert, el Marçal, l'Eduard, el Dimos i sobretot en Joost. Finalment, dins del cercle CVC, també voldria mencionar el Felipe, de qui he rebut un suport especial de manera espontània.

Fora del CVC també he pogut passar moltes estones "ponent l'ou" amb la Sònia (fent el que metafòricament es diu "un cafè") i sobretot amb la Georgina (de qui estic orgullós de dir que conec des de que feiem punxó).

Haver arribat fins aquí no hauria estat possible sense l'ajuda de la meva família, especialment dels meus pares. Des de sempre han invertit gran part del seu temps en portar-me amunt i avall per donar-me el màxim que han pogut. Primer van ser viatges cap a l'escola, després cap a la universitat i finalment cap al Centre de Visió. A ma germana li dono les gràcies per haver obert camí i entendre "de què va tot això". Als meus avis els vull agrair l'alegria amb què em reben a l'hora del Pasapalabra. I finalment, al Cristóbal li vull donar les gràcies per haver estat sempre al meu costat i recordar-me quines són les coses realment importants... (aprofitant l'ocasió, també li demano perdó per haver convertit el menjador en despatx i trigar tant en treure l'ordinador de sobre la taula quan és l'hora de sopar, jeje ;))

Gràcies a tots !!!



# Abstract

The broad access to digital cameras, personal computers and Internet, has lead to the generation of large volumes of data in digital form. If we want an effective usage of this huge amount of data, we need automatic tools to allow the retrieval of relevant information. Image data is a particular type of information that requires specific techniques of description and indexing. The computer vision field that studies these kind of techniques is called *Content-Based Image Retrieval* (CBIR). Instead of using text-based descriptions, a system of CBIR deals on properties that are inherent in the images themselves. Hence, the feature-based description provides a universal via of image expression in contrast with the more than 6000 languages spoken in the world.

Nowadays, the CBIR is a dynamic focus of research that has derived in important applications for many professional groups. The potential fields of application can be such diverse as: the medical domain, the crime prevention, the protection of the intellectual property, the journalism, the graphic design, the web search, the preservation of cultural heritage, etc.

The definition on the role of the user is a key point in the development of a CBIR application. The user is in charge to formulate the queries from which the images are retrieved. We have centered our attention on the image retrieval techniques that use queries based on pictorial information. We have identified a taxonomy composed by four main query paradigms: query-by-selection, query-by-iconic-composition, query-by-sketch and query-by-paint. Each one of these paradigms allows a different degree of user expressivity. From a simple image selection, to a complete painting of the query, the user takes control of the input in the CBIR system.

Along the chapters of this thesis we have analyzed the influence that each query paradigm imposes in the internal operations of a CBIR system. Moreover, we have proposed a set of contributions that we have exemplified in the context of a final application.





# Resum

L'accés massiu a les càmeres digitals, els ordinadors personals i a Internet, ha propiciat la creació de grans volums de dades en format digital. En aquest context, cada vegada adquireixen major rellevància totes aquelles eines dissenyades per organitzar la informació i facilitar la seva cerca. Les imatges són un cas particular de dades que requereixen tècniques específiques de descripció i indexació. L'àrea de la visió per computador encarregada de l'estudi d'aquestes tècniques rep el nom de *Recuperació d'Imatges per Contingut*, en anglès *Content-Based Image Retrieval* (CBIR). Els sistemes de CBIR no utilitzen descripcions basades en text sinó que es basen en característiques extrems de les pròpies imatges. En contrast a les més de 6000 llengües parlades en el món, les descripcions basades en característiques visuals representen una via d'expressió universal.

La intensa recerca en el camp dels sistemes de CBIR s'ha aplicat en àrees de coneixement molt diverses. Així doncs s'han desenvolupat aplicacions de CBIR relacionades amb la medicina, la protecció de la propietat intel·lectual, el periodisme, el disseny gràfic, la cerca d'informació en Internet, la preservació dels patrimoni cultural, etc.

Un dels punts importants d'una aplicació de CBIR resideix en el disseny de les funcions de l'usuari. L'usuari és l'encarregat de formular les consultes a partir de les quals es fa la cerca de les imatges. Nosaltres hem centrat l'atenció en aquests sistemes en què la consulta es formula a partir d'una representació pictòrica. Hem plantejat una taxonomia dels sistemes de consulta en composada per quatre paradigmes diferents: Consulta-segons-Selecció, Consulta-segons-Composició-Icònica, Consulta-segons-Esboç i Consulta-segons-Il·lustració. Cada paradigma incorpora un nivell diferent en el potencial expressiu de l'usuari. Des de la simple selecció d'una imatge, fins a la creació d'una il·lustració en color, l'usuari és qui pren el control de les dades d'entrada del sistema.

Al llarg dels capítols d'aquesta tesi hem analitzat la influència que cada paradigma de consulta exerceix en els processos interns d'un sistema de CBIR. D'aquesta manera també hem proposat un conjunt de contribucions que hem exemplificat des d'un punt de vista pràctic mitjançant una aplicació final.



# Contents

<b>Acknowledgment</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Resum</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Content-Based Image Retrieval . . . . .	1
1.2 Motivation . . . . .	3
1.3 Objectives of the thesis . . . . .	5
1.4 Thesis Outline . . . . .	5
<b>2 Content-Based Image Retrieval</b>	<b>7</b>
2.1 Fundamentals of image retrieval . . . . .	7
2.1.1 Image content modelling . . . . .	7
2.1.2 CBIR Architecture . . . . .	8
2.2 Part Extraction . . . . .	10
2.2.1 Whole Image . . . . .	10
2.2.2 Template Subdivisions . . . . .	10
2.2.3 Segmentation . . . . .	10
2.2.4 Contours . . . . .	13
2.2.5 Parts from local structures . . . . .	13
2.3 Feature Description . . . . .	17
2.3.1 Color . . . . .	17
2.3.2 Texture . . . . .	20
2.3.3 Shape . . . . .	25
2.3.4 Spatial layout . . . . .	37
2.4 Feature Indexation . . . . .	40
2.4.1 Space-ordering methods . . . . .	40
2.4.2 Hash-based methods . . . . .	41
2.4.3 Tree-based methods . . . . .	42
2.5 Part Matching . . . . .	45
2.5.1 Alignment techniques . . . . .	45
2.5.2 Generalized Hough Transform . . . . .	45
2.5.3 Spring-like Connected Configuration . . . . .	47

2.5.4	Bag of words . . . . .	49
2.6	Similarity measures . . . . .	50
2.6.1	Minkowski-Form distance . . . . .	50
2.6.2	Weighted Euclidean distance . . . . .	51
2.6.3	The histogram intersection . . . . .	51
2.6.4	Quadratic Form (QF) Distance . . . . .	52
2.6.5	Mahalanobis Distance . . . . .	52
2.6.6	Hausdorff distance . . . . .	52
2.6.7	Chamfer distance . . . . .	53
2.6.8	Earth Movers distance . . . . .	53
2.7	Metrics for Performance Evaluation . . . . .	54
<b>3</b>	<b>Query-by-Image Selection</b>	<b>57</b>
3.1	Introduction . . . . .	57
3.2	A CBIR system based in a multi-scale triangulation of the image content	61
3.3	Description . . . . .	61
3.3.1	Part detection . . . . .	61
3.3.2	Feature extraction . . . . .	63
3.4	Matching . . . . .	66
3.4.1	Feature indexing . . . . .	66
3.4.2	Evaluation . . . . .	66
3.5	Experiments and Results . . . . .	67
3.5.1	Example in the retrieval of programs and commercials . . . . .	67
3.5.2	Evaluation of the system performance . . . . .	67
3.6	Discussion . . . . .	72
3.6.1	Conclusions of our approach . . . . .	73
3.6.2	Conclusions of the query-by-selection paradigm . . . . .	73
<b>4</b>	<b>Query-by-Iconic Composition</b>	<b>75</b>
4.1	Introduction . . . . .	75
4.2	A CBIR system based in region graph models . . . . .	78
4.3	Description . . . . .	79
4.3.1	Part extraction . . . . .	81
4.3.2	Feature extraction . . . . .	90
4.4	Matching . . . . .	97
4.4.1	Feature indexing . . . . .	98
4.4.2	Evaluation . . . . .	98
4.5	Experiments and Results . . . . .	98
4.5.1	Evaluation of the part extraction . . . . .	100
4.5.2	Evaluation of the feature description . . . . .	104
4.6	Discussion . . . . .	105
4.6.1	Conclusions of our approach . . . . .	105
4.6.2	Conclusions of the query-by-iconic composition paradigm . . . . .	106
<b>5</b>	<b>Query-by-Sketch</b>	<b>109</b>
5.1	Introduction . . . . .	109

5.2	A CBIR system based in geometrical constraints of a vectorial representation . . . . .	113
5.3	Description . . . . .	114
5.3.1	Part extraction . . . . .	114
5.3.2	Feature description . . . . .	115
5.4	Matching . . . . .	119
5.4.1	Feature indexing . . . . .	119
5.4.2	Evaluation . . . . .	121
5.5	Experiments and Results . . . . .	125
5.5.1	Application on architectural plans . . . . .	125
5.5.2	Examples in general contexts . . . . .	129
5.6	Discussion and Conclusions . . . . .	135
5.6.1	Conclusions of our approach . . . . .	135
5.6.2	Conclusions of the query-by-sketch paradigm . . . . .	136
<b>6</b>	<b>Query-by-Paint</b>	<b>139</b>
6.1	Introduction . . . . .	139
6.2	A CBIR system based in the principles of the human perception . . . . .	141
6.3	Description . . . . .	142
6.3.1	Part extraction . . . . .	142
6.3.2	Feature extraction . . . . .	149
6.4	Matching . . . . .	150
6.4.1	Feature indexing . . . . .	150
6.4.2	Evaluation . . . . .	151
6.5	Experiments and Results . . . . .	157
6.5.1	Validation of the part extraction . . . . .	157
6.5.2	Evaluation of the description features . . . . .	159
6.5.3	Application examples . . . . .	168
6.6	Discussion . . . . .	181
6.6.1	Conclusions of our approach . . . . .	181
6.6.2	Conclusions of the query-by-paint paradigm . . . . .	182
<b>7</b>	<b>Conclusions</b>	<b>185</b>
7.1	Influence of the query paradigm in the design and the performance of a CBIR system . . . . .	185
7.2	Usage of pictorial queries in CBIR systems of general purpose . . . . .	187
7.3	Contributions . . . . .	188
7.4	Future lines of research . . . . .	190
	<b>Bibliography</b>	<b>193</b>



# List of Tables

2.1	Part extraction overview . . . . .	16
2.2	Color description overview . . . . .	20
2.3	Texture description overview . . . . .	24
2.4	Contour based shape description . . . . .	31
2.5	Region based shape description . . . . .	35
2.6	Appearance based shape description . . . . .	37
2.7	Spatial layout overview . . . . .	39
2.8	Indexation overview . . . . .	44
2.9	Arrangement validation overview . . . . .	50
4.1	Normalized areas of the garment regions . . . . .	91
4.2	Number of groundtruth images for each model . . . . .	98
4.3	Precision, Recall and Fallout results for each model. . . . .	98
4.4	Percentage of correct classification between models. . . . .	100
6.1	Collection of paints of the test. . . . .	161
6.2	Mean AUC values using the color features . . . . .	163
6.3	Mean AUC values using the shape features . . . . .	163
6.4	Mean AUC values using the color and shape features . . . . .	164
6.5	AUC values for every query using: MeanLuv $\epsilon_c=0.6$ and DCT $\epsilon_s=0.3$ . . . . .	165
6.6	Users ordered by their mean AUC values. . . . .	166
6.7	Objects ordered by their mean AUC values. . . . .	166
7.1	A higher activity in the query formulation implies a higher degree of personalization. . . . .	186
7.2	The human representation incorporates a deformation of the query that can decrease the precision of the retrieval results. . . . .	186
7.3	Strategies of the proposed CBIR systems. The main contributions are highlighted in bold type. . . . .	188





# List of Figures

2.1	Levels of image content decomposition. . . . .	8
2.2	Modules of a CBIR system . . . . .	9
2.3	Automatic scale selection related to the maximum response of LoG operator. The third image shows the optimal scale of the patch. . . . .	14
2.4	a) The EBR detector starts from a corner point p and exploits nearby edge information. b) The IBR detector select intensity extrema and considers intensity profile along rays (reprinted from [TG04] [TM07]). . . . .	15
2.5	MSERs are connected components that remains along the sequence. . . . .	15
2.6	a) Complex regions, such as the eye, exhibit unpredictable local intensity hence high entropy. b) Salient regions detected. c) Graphic of the saliency behavior on the eye point across scales (reprinted from [KB01]). . . . .	16
2.7	a) Original image on the top left and their channels in the YCbCr color space. b) Two-dimensional DCT frequencies. c) Zigzag order of the DCT coefficients. . . . .	19
2.8	Example of construction of a co-ocurrence matrix . . . . .	21
2.9	Examples of regularity. From left to right: highly regular, regular, slightly regular and irregular. . . . .	22
2.10	Example of the wavelet transform. Large coefficients are visualized in black or white and are located, at each scale, along edges of the image. Otherwise, regular areas, for example on the shoulder, present small coefficients that are visualized in gray. Fine scale are in the bottom right. There is three kinds of coefficients per scale: horizontal, vertical and diagonal. . . . .	22
2.11	Representation of the bank of 30 Gabor filters in polar coordinate system used for the Homogeneous Texture descriptor of the MPEG7 standard. . . . .	23
2.12	EHD descriptor: Image subdivision and edge types. . . . .	24
2.13	a) Turning function (reprinted from [VH99]). b) Centroid Distance function (reprinted from [ZL04]). . . . .	26
2.14	a) Construction of the CSS descriptor. b) The rotation transform causes a shift of the signature. Noise causes the apparition of small peaks. . . . .	27

2.15	a) Horse shape is partitioned in correspondence with minima of the curvature function. b) The tokens are arranged in the feature space defined by the curvature and the orientation. c) Angle $\theta$ describes the orientation. . . . .	28
2.16	a) Different polygonal approximations. b) Supersegment features (reprinted from [SM92]). . . . .	28
2.17	Construction of the Shape Context descriptor . . . . .	29
2.18	The shape descriptor is defined by the histogram of angles extracted from the Delaunay triangulation of the contour points (reprinted from [ZL04]). . . . .	30
2.19	Visualization of the groupings. a) Longer linear line. b) Co-terminations. c) L junctions. d) U junction. e) U junction. f) Parallel groups. g) Polygons (reprinted from [IA02]). . . . .	30
2.20	Simple shape global features . . . . .	32
2.21	Example of the reconstruction of the left image with the Zernike moments of order 5, 10, 15 and 20. . . . .	33
2.22	a) and b) Real parts and imaginary parts of the ART basis functions. The imaginary parts have similar shape to the corresponding real parts but with different phases. Note that the brighter the region, the higher the value. . . . .	33
2.23	Example of the grid based method descriptor (reprinted from [SSS00]).	34
2.24	Haar-like features (reprinted from [LM02]). . . . .	36
2.25	Computation of the SIFT descriptor. The gradient magnitude and orientation are computed at each image point. They are weighted by a Gaussian window and then accumulated into orientation histograms. SIFT descriptor is a vector of 128 values. . . . .	36
2.26	Jungert's spatial operators . . . . .	38
2.27	2D C-string example . . . . .	38
2.28	a) The cutting and the corresponding RS-string with A as the rotation center object. b) The initial position of the rotating half line. The dotted lines show the begin and the end bounds in the sector-direction. c) Object A is cut into A' and A'' by the rotating half line at the position with sector coordinate = 0. . . . .	39
2.29	Taxonomy of the feature indexing methods and their reference year. .	40
2.30	Hilbert curve in 2 and 3 dimensions . . . . .	41
2.31	a) Hash table indexation according to the module function. b) VA File Structure. . . . .	42
2.32	Process of model indexing with a geometric hashing technique. . . . .	42
2.33	Examples of tree structures a) k-d-tree b) R-tree c) SS-tree . . . . .	44
2.34	Generalized Hough Transform for curve matching. An R-table stores the geometric features of the contour points respect to a reference point $(x_c, y_c)$ . . . . .	46
2.35	Matching of a car with a Generalized Hough Transform strategy. The look-up table stores the parts that define the car and their position according to a reference point. A high density of votes denote the presence of the car in a scene. . . . .	46

2.36	The image of a horse is modelled by the average position of the subparts according to different points of view. A generalized Hough voting strategy accumulates the evidence of the object detection in a hierarchical structure of three levels (reprinted from [BT05]). . . . .	47
2.37	Figure from the work of Fischler and Elschlager [FE73] . . . . .	47
2.38	a) Graphical geometric models. b) Examples of objects modelled for the above type of structures (reprinted from [FH05] [FPZ03] [FPZ05] [CL06]). . . . .	49
2.39	Bag of words strategy defines a dictionary of image parts and represent an object as a collection of them. (reprinted from [Woj09]) . . . . .	49
2.40	Example on the Minkowski-Form Distances . . . . .	51
2.41	Chamfer distance for shape matching . . . . .	53
2.42	a) An example of a transportation problem with three suppliers and two consumers (reprinted from [RTG00]) b) EMD between the pattern color signature and the signatures for the various portions of the image (reprinted from [CG99]) . . . . .	54
2.43	Illustration of the precision, recall and fallout concepts . . . . .	55
2.44	Comparison of curves of the Precision vs Recall and ROC graphs . . . . .	56
3.1	Example of query-by-internal selection (CIRES system 3.1). a) The database images are categorized by semantic concepts. The user can browse this hierarchy and select an image. b) Otherwise, the query can be selected from random set of samples. c) Once the query is selected the user can adjust the weights related to the Color, Texture or Shape. . . . .	59
3.2	Modalities of part selection of the query image a) Whole Image b) Segmented region c) User's cropped selection . . . . .	60
3.3	Scale-space image stack . . . . .	63
3.4	Image part identification . . . . .	63
3.5	Layout encoding of a resolution level . . . . .	64
3.6	Example of the descriptor construction . . . . .	65
3.7	Computation steps of the image feature $F$ from the histograms of every resolution level $h(T;t)$ . . . . .	66
3.8	Retrieval examples of advertisements (a) and program logos (b). The X axis of the graphs represent the chronological order of the video frames. The Y axis contain the probability of the query detection according to each image of the sequence. . . . .	68
3.9	a) Samples of the queries for each ground truth sequence. b) Precision-recall graphs and ROC curves for our approach. c) For the corner based approach [TG99] d) For our approach with one level of resolution. . . . .	70
3.10	Examples of the main problems of our CBIR system. a) Confusion in images with similar structure b) Changes in the point of view c) Occlusions d) Changes in the illumination. . . . .	71
3.11	Examples of image variations that the system do overcome. . . . .	72

3.12	Visual resume of the modules of our approach. Image extraction: peaks of the distance map obtained from the contours at several resolution levels. Features: combination of the angular histogram of the triangulation at every resolution level. Feature Indexation: list of descriptors.	72
4.1	Media Streams iconic interface for video annotation. Icon path with the meaning: 'On the top of a street in Texas' (repinted from [Dav93])	76
4.2	Examples of icon-based interfaces in retrieval systems of narrow and broad domains a) Like.com b) ImageSearch (reprinted from [Lew00])	77
4.3	a) Icons of the system related to the human appearance. The first row are the icons related to the garments. b) Scene captured at the entrance desk.	78
4.4	a) Icons of the garments. b) Models we can construct combining them: Model 1 is given by the icon 3; Model 2 is the combination of the icons 2 and 3; Model 3 results from the icons 2, 3, 4; Model 4 from 1 and 3; Model 5 is formed by 1, 3, and 4.	79
4.5	Process of the image description according to the models.	80
4.6	Example of the features of the node n.	83
4.7	Example regions $n_1, n_2, n_3, n_4, n_5$ . a) Boundary Distance: $BD(n_2, n_2)=1$ , $BD(n_4, n_5)=0.9$ , $BD(n_2, n_5)=0$ . b) Average Color Distance: $ACD(n_2, n_4)=0.2$ , $ACD(n_1, n_2)=0.68$ . c) Color Histogram Distance: $CHD(n_2, n_4)=0.12$ , $CHD(n_1, n_2)=1$ .	83
4.8	a) Graph at the step $t$ b) Application of the division operator over $\gamma_D : G^t$ as $\gamma_D(n_2)$ into three nodes $n'_3, n'_2, n'_7$ . The result $G^{t+1}$ implies the application of $L_N$ over $n'_3, n'_2, n'_7$ and $L_E$ over the new edges $e_{2,3'}$ , $e_{2,7'}$ and $e_{6,7'}$ . c) Application of the fusion operator over $\gamma_F : G^{t+1}$ as $\gamma_F(n_1, n_2)$ . The result $G^{t+2}$ implies the application of $L_N$ over $n_2'$ and $L_E$ over $e_{2,3'}$ , $e_{2,4'}$ , $e_{2,5'}$ , $e_{2,6'}$ and $e_{2,7'}$ . We also observe the removing of two edges: $e_{1,2}$ , that would connect the same nodes as $e_{2,5'}$ , and $e_{1,5}$ that would form a loop in the node $n_2'$ .	84
4.9	Karu's texture segmentation a) c) Pattern at different scales. b) d) The interest points are the local extrema of (a) and (c). f) Textured regions of (e) with average density of extrema between 0.04 and 0.16. (Reprinted from [KJB96]).	86
4.10	The five steps of the texture discrimination process	86
4.11	Examples of the texture discrimination process	86
4.12	Determining a child node from color bits. Division of the RGB color cube into eight sub-cubes.	87
4.13	Plain split examples: original image, quantized image and palette, resulting regions	88
4.14	Modelling of the five possible clothing compositions	90
4.15	AP: Spatial relations due to the angular relation	91
4.16	LP: Spatial relations due to the limits of the bounding boxes	92
4.17	Cloth classes, number of regions and spatial relations	92

4.18	Interpretation over the segmented results on the upper-left image. We show three possible results belonging to models: 1,2 or 4. The image is classified as Model2 due to the lower value of FCostC. . . . .	96
4.19	Interpretation over the segmented results on the upper-left image. We show three possible results belonging to models: 1,4 or 5. The image is classified as Model5 due to the lower value of FCostC. . . . .	96
4.20	Model Classification. Initialization. Step1: Chooses the best combination for each model. Step2: Chooses the final combination that represents the image. . . . .	97
4.21	Examples of the retrieved images for the five query models. . . . .	99
4.22	Statistic of the error on the part extraction a) Respect to the number of images b) Respect to the success on the model identification . . . .	101
4.23	Segmentation examples . . . . .	102
4.24	Segmentation examples with high error . . . . .	103
4.25	Examples of ideal segmentation. Causes of a mistaken classification: 1) Occlusions 2) Addition of external objects 3) Deformation of the ideal composition . . . . .	104
4.26	Visual resume of the modules of our approach. Part extraction of the database: regions using split-and-merge segmentation. Features: model identification. Feature Indexation: tables images according to their region garments. . . . .	105
5.1	Database image types a) Sketch b) ClipArt c) Isolated Object d) Complete Scene e) Object Inside a Cluttered Scene. (Images reprinted from [Leu03] [FBRJ04] [LC02a] [CNM05]) and [FTG06]) . . . . .	110
5.2	The information of the query and the scene is approximated by vectors	114
5.3	Global features of an image vector $v_i$ . . . . .	115
5.4	Computation of the pairwise features of two vectors $PF(v_{ij})$ . . . . .	116
5.5	a) Primitive types composed by its vector $w_a$ and the reference one $w_r$ (the black vector) b) The whole set of primitives describe the relationships of perpendicularity, parallelism and co linearity according to a reference vector. . . . .	116
5.6	The local description of a vector $v$ is given by the minimum deformation that have to perform the primitive structures to fit the surrounding vectors . . . . .	117
5.7	Indexing tables. a) Index of the vectors identifier according to their local features. b) Index of the global features according to the vector identifiers. . . . .	120
5.8	Fuzzy similarity value assigned to the vectors of the scene according to the local features of the query. It can be represented as a ramp-like function. a) Example of the indexing in the table related to $LF^{\alpha,z}$ b) Example on the table of the $LF^{d,z}$ values. . . . .	121
5.9	Example of the location of the reference point for the votes $h_{18,140,1}$ and $h_{18,140,0}$ . . . . .	122
5.10	The map H contains the votes and their accumulated weights. . . . .	122

5.11	a) Representation of the alignment evaluation that combines the distance and the angular values. b) c) Alignment on related to the votes belonging to the peaks of H. The detection probability is 0.84 and 0.75 respectively. . . . .	125
5.12	a-f) Original symbols and one example of sketch instance. g) Example of 3 database images containing the symbol f). . . . .	126
5.13	Examples of the sketch detection. a-f) Matching probabilities $T$ : 0.75, 0.74, 0.78, 0.73, 0.72, 0.65. . . . .	127
5.14	Precision-recall graph and ROC curves for each class of sketch. On the bottom, mean of the precision and recall values for the sketches of each symbol retrieving the first 40 images (the number of ground truth images). . . . .	128
5.15	Examples of problems in the sketch detection caused by: a) A partial noisy approximation that provokes some parts of the symbol to be lost. b) A global deformation of the symbol. c) Intestable vectorization of the curves. d) Subpart detection of a symbol. . . . .	129
5.16	Example of sketch detection: star. From top to down, the $T$ values of the matching are: 0.63, 0.70, 0.64, 0.69 . . . . .	130
5.17	Example of sketch detection: car logo. From top to down, the $T$ values of the matching are: 0.77, 0.75, 0.75, 0.68. . . . .	131
5.18	Example of sketch detection: alert sign. From top to down, the $T$ values of the matching are: 0.69, 0.70, 0.76, 0.72 . . . . .	132
5.19	Example of sketch detection: catalan donkey. From top to down, the $T$ values of the matching are: 0.61, 0.63, 0.64, 0.54. . . . .	133
5.20	Example of sketch detection: glasses. From top to down, the $T$ values of the matching are: 0.70, 0.66, 0.63, 0.68. . . . .	134
5.21	Visual resume of the modules of our approach. Part extraction: contour vectors. Features: geometric relations between the vectors according to the primitives. Feature Indexation: tables of local and global features. Part Matching: Hough-like voting and alignment by oriented Chamfer distance. . . . .	135
6.1	Example query of QBIC system on the Hermitage Museum Database	140
6.2	Interfaces of the systems a) DrawSearch b) Visualseek c) Picasso . . .	141
6.3	Gestalt laws of Organization. We focus on the laws of proximity and color similarity. . . . .	143
6.4	Illustration of the mean shift iterations in a 2D data. The objective is to find the densest region. . . . .	145
6.5	a) Grid of segmented images using the $MSS$ according to the parameters of color $HC$ and space $HS$ . We show an example of the region $r_3^{(6,2)}$ and its analogous ones. Observe how it grows trough the color, merging with similar pixels, and how it grows trough the space merging with similar regions. b) Some selected regions and their stability value $SCS$ . c) Original image. d) Stability value $S$ of $r_3^{(6,2)}$ according to the analogous region on the cell (6,6). . . . .	147

6.6	Regions extracted by CoReSt algorithm. Notice the grouping of several elements such as the natural textures of the trees or the sail with its shadow. Details like the little reddish tree are also preserved. . . . .	148
6.7	Examples of region detection. We can group the parts of the bomb drawing or the textures of the background posters. . . . .	148
6.8	Example of the regions detected in the scene. Observe that the letters of the SouthPark banner are detected individually but are also understood as a group. . . . .	149
6.9	Graphic of the function $D_F$ with $\alpha = 0.25$ . X axis represent the Euclidean distance between two features normalized by the maximum tolerance $\epsilon_k$ . Y axis shows the function response according the values of the X axis. . . . .	152
6.10	Example of matching between a query region and a scene. The generated vote is showed with a frontal and a perspective view in the voting space. The first and second dimensions of the space are related to the coordinate localization of the reference point, the third one specifies the scale transformation and, a fourth one that is not represented, identifies the image scene. . . . .	153
6.11	a) Scene b) Query c) d) e)Query regions. The reference point is marked with a symbol 'o'. We show the line joining the center of the region with the reference point of the query. . . . .	154
6.12	Voting space: a) 3D view of the accumulated result in the predefined points $ap$ . The size and the color of the spheres represent the goodness of the matching (red and big:good, blue and little:bad). b) Projection of the scale transformation plane. c) Projection of the 2D coordinates plane. d) Analogous representation of c) over the image scene. . . . .	154
6.13	The steps of the matching process of the three regions of the query. a) c) e) Show the scene regions that are similar to the query ones that we show in the previous Figure 6.11. b) d) f) Show the similarity values in the accumulation points of the voting space. The values of the query steps are combined together to form the final result of the figure 6.12. . . . .	155
6.14	a) Query b) Scene c) d) 2D and 3D views of the voting space. . . . .	156
6.15	a) Query b) Scene c) d) 2D and 3D views of the voting space. . . . .	156
6.16	Generation of the segmentation using the CoReSt regions ranked by the stability. The first row shows the progress of the boundaries and the second shows the progressive incorporation of the regions. Those regions that are occluded by other regions that are more stable are not included in the segmentation result. . . . .	158
6.17	Examples of the CoReSt segmentations . . . . .	158
6.18	a) Original Images. b) and c) present the grouping properties. b) Regions that form a textured area. c) Regions that present occlusions. d) Regions detected as outstanding for having contrasted color and being isolated . . . . .	159
6.19	Comparison of the CoReSt strategy against other state-of-the-art methods. GCE results for the 200 Berkeley images (values taken from [VvdWB08]). . . . .	159

6.20	20 objects of the ALOI database that we have used in the experiment of the features evaluation. . . . .	160
6.21	Construction of the 1800 images of the synthetic database combining 10 cluttered backgrounds and 20 objects with 9 variations. . . . .	160
6.22	Examples of the queries. a) Original model. b) and c) paints from users 9 and 11, the one with best performance and the one with the worst. Observe the differences in the color and shape of the paints. . .	166
6.23	Examples of the problems in the query-by-paint retrieval. a) Large color variations. b) Large shape variations. c) Objects represented by few regions and also large color and shape variations. d) Objects that can be confused with the background. . . . .	167
6.24	Retrieval example from the Google images of the query: "rubber duck".	170
6.25	Retrieval example from the Google images of the query: "smurfin". . .	171
6.26	Retrieval example from the Google images of the query: "red clock". . .	172
6.27	Retrieval example of the Time Covers Archive. . . . .	173
6.28	Retrieval example of the Time Covers Archive. . . . .	174
6.29	Retrieval example of the Time Covers Archive. . . . .	175
6.30	Retrieval example of the LaVanguardia logo. . . . .	176
6.31	Retrieval example of an advertisement. . . . .	177
6.32	Retrieval example of a new that contains a starry background. . . . .	178
6.33	Retrieval example of a new that contains a London bus. . . . .	179
6.34	Retrieval example of news that contain a football player dressed in blue.	180
6.35	Visual resume of the modules of our approach. Part extraction of the database: regions using the CoReSt algorithm. Features: color, shape, size, position. Feature Indexation: k-d-trees of color and shape; table of size, position and identifiers. Matching evaluation: Hough-like voting in 4D. . . . .	181



# Chapter 1

## Introduction

### 1.1 Content-Based Image Retrieval

Today we live in what is commonly called the Information Age. The broad access to computers and Internet makes people have instant access to knowledge all over the world. Since year 2000 until 2008 the percentage of population using internet has grown an 342.2% and, as of March 2009, it represents approximately 1.596.270.108 of people. Nowadays, digital cameras and personal computers have converted every individual in a potential producer of information. This way, the digital support has become a common format of data storage. Its reduced physical volume and their potential capabilities of transmission, access and management, makes it ideal to represent large volumes of data.

Digital data that comes from several sources (text, image, video or audio) is called *multimedia*. Everyday we can interact with large amounts of multimedia data including photographs, music, videos, drawings, charts, animations, etc. If we want a widespread usage of this great volume of information, we need for effective tools to manipulate it. In this context, an important issue is the design of automatic systems that facilitate the retrieval of relevant information.

*Information Retrieval* is the field of knowledge that deals with the representation, storage, and access to information items. More specifically, when the retrieved information is a collection of images, this field of knowledge is called *Image Retrieval*.

The origins on Image Retrieval can be traced back to 1979 when a specific conference on Database Techniques for Pictorial Applications was held in Florence [Bla80]. Until that moment, the techniques of image retrieval were based on the textual annotation. Images were first manually annotated with text and then were searched by the traditional text-based management systems.

However, this purely text-based approach posed two significant limitations in the retrieval of images. The first limitation was related to the volume of the database. Manual annotation was such an slow and an expensive process that could not be

applied to large image databases [Yin03]. Moreover, the second limitation affected the performance of the system. The description of the images was found to be a highly subjective task that could generate different text labels to the same image. The works of Markey [Mar88] and Enser [EM92] concluded that there were wide disparities in the keywords that different individuals assigned to the same picture. One of the causes of this phenomena was derived from the know-how of the users in a concrete discipline related to the images. If users have a different degree of expertise, it is highly probable that they provide different textual descriptions to the same image. Moreover, a description could also vary according to the time it was made because if a discipline evolves along the time, its vocabulary also changes and evolves. Keister [Kei94] summarized the ambiguity of the text-based image indexing with the following sentence:

” . . . it is not so much that a picture is worth a thousand words, for many fewer words can describe a still picture for most retrieval purposes, the issue has more to do with the fact that those words vary from one person to another. . . ”

With the beginning of the Information Era, the efficient management of the rapidly expanding visual information became an urgent problem. In 1992, the National Science Foundation of the United States organized a workshop on visual information management systems to identify new directions in the management of the image databases [Jai92]. Difficulties faced by text-based image retrieval brought the researchers to develop new solutions to represent and index visual information. This new trend of image retrieval was based on properties that are inherent in the images themselves and was called *Content-Based Image Retrieval*.

*Content Based Image Retrieval* (CBIR) is the process of retrieving images from a collection based on automatically extracted features.

The feature-based description provides a universal via of image expression in contrast with the more than 6000 languages spoken on the world. The content-based image description can be applied to any image applying the same computational process. This kind of description specially benefits those images that present items that are difficult to be described by text (e.g. irregular shapes or textures). Researchers from the communities of computer vision, database management and human-computer interaction were quickly attracted to the CBIR field. As a consequence, a fast evolution of the CBIR run in parallel with the impressive growth of digital information. The CBIR field acquired such relevance that in October 1996 the *Moving Picture Experts Group* (MPEG) recognized the need to organize and identify the content of the multimedia data. They started a work called *Multimedia Content Description Interface* that is better known as MPEG-7 ISO/IEC standard. The new member of the MPEG family provided a guide to organize the audiovisual content and allow searching for material that is of interest to the user. Thus, from the end of the 20th century until the actual days, the evolution of the CBIR has conquered both academic and commercial fields.

Nowadays, the search for effective and efficient techniques of CBIR is still a dynamic focus of research. The comprehensive works of Rui [RH99], Eakins [EG00] and Smeulders [SMW<sup>+</sup>00] provide some of the most influential surveys on the CBIR until year 2000. Moreover, the extensive work of Veltkamp [VT00] also outstands to describe the functionality of more than 50 CBIR systems. Finally, the recent study of Datta [DJLW08] (2008) poses an actual overview of the fundamental of CBIR and discusses its major future challenges.

The intensive research on CBIR techniques has derived in important applications for many professional groups [EG99]: In the medical domain, the retrieval systems help to assist the medical staff in the management of the images obtained from visualization methods such as X-rays or MRI. Beside the medical diagnosis, other powerful visualization methods have been developed for a wide range of purposes. Satellites that screen our planet send us hundreds of images everyday. These images can be used for the militaries (to identify targets), for the geologists (to study the terrain) or for the farmers (to analyze the health of the harvests). CBIR application can also be found in a very different field such as the preservation of the cultural heritage. Art galleries and museums store their collections digitally and make them public available. Then, CBIR systems are tools that historians can use to trace the artistic influences by identifying objects that share certain visual characteristics. In a similarly way, journalists, publicists and designers can profit of the retrieval on historic archives to illustrate their current works. Moreover, from the reverse point of view, the CBIR systems are also applied in the preservation of the intellectual property to detect fakes and to register trademarks. Finally, in crime prevention area we can find the fingerprint recognition as one of the most specific and well known applications in the CBIR field.

Despite CBIR can be involved in a large amount of areas, the long list of applications is still far to be completed. New technologies will open the market to other retrieval functionalities such as the well-known expected interactive television. This application will become soon a reality that will need services to allow the users to search and download from distant sources all types of television shows and visual material.

## 1.2 Motivation

The research in CBIR field is motivated by the large amount of potential applications that the new technologies offer. The access to massive volumes of image data has become quotidian reality. Hence, users demand for effective tools to manage this large volumes of images according to their needs.

The definition on the role of the user is a key point in the system development of a CBIR application. The user is in charge to formulate the queries from which the images are retrieved. Then, one of the main questions related to the role of the user

is the definition of the media that is used to formulate those queries. The research community devoted to the human-computer interaction have developed a wide variety of inputs including gestures, eye movement, speech, text, sound, pictures, etc.

In this thesis we center our attention on the image retrieval techniques that use queries based on pictorial information. We have used this kind of query modality because it provides a high degree of flexibility: the pictorial information is independent of the user language, can be created in an easy way, it is broadly available in the WWW resources and it does not require high cost electronic devices to be produced (e.g. an eye tracker).

Once the query media is defined, we can identify the solutions have been conceived to query by pictorial elements. In the literature [Deb03], we can find several formats of pictorial-query specifications: by image selection, by iconic composition, by sketch and by paint. We can briefly describe them as follows:

**Query-by-Image-Selection** The user provides a sample image to represent the prototype of what he is looking for. Its visual content is considered to reproduce similar visual features as the target images. In this query paradigm, the user has the possibility to indicate which specific features wants the CBIR system to consider.

**Query-by-Iconic-Composition** Query using icons allow the user to select an icon that represents a high level concept of a category idea. Icons are commonly found in a graphical user interface and are a graphic representation of an item or an entity. Icons are seen as the basic elements that the user can combine to create its own query.

**Query-by-Sketch** User-sketched outlines represent the boundaries of the objects present in a scene. The sketches can be drawn from scratch with total freedom of the user form a design interface of the system.

**Query-by-Paint** In the query-by-paint the user defines the input query as a composition of color regions. Regions are drawn on a board to reproduce salient color patches of the parts of the objects. The region composition is meant to fit the visually salient image regions with their visual attributes.

These four query paradigms present a different degree of the involvement of the user in the query specification. From a simple image selection, to a complete painting of the query, the user takes control of the input in the CBIR system. In a world with a constant increment of available images it seems relevant to analyze "how far can we go" in the development of *general purpose retrieval techniques*. With *general purpose* we mean that retrieval techniques have to be as independent as possible of the final application. This way, the retrieval system cannot use intensive learning processes on a set of concrete queries that are known in advance. Otherwise, the retrieval process has to based in the visual content of each individual query. Then, even thought the content of the database changes, the same retrieval strategy can be applied.

### 1.3 Objectives of the thesis

The objectives of this thesis can be divided in two main blocks. In one hand we want to study the CBIR field from a scientific point of view and, on the other hand, we want to provide a set of contributions in the application domain. We have proposed this dual set of objectives according to a common point of view: the influence of each type of pictorial query in the development of a CBIR system.

The scientific objectives of this thesis comprise the following points:

- Identify the basic processes that a CBIR system comprise.
- Survey the state of the art of each one of these processes.
- Analyze the benefits and limitations of each type of query according to the user needs.
- Analyze the influence that each query paradigm imposes in the internal operations of a CBIR system.
- Study the most suitable options to construct a retrieval system according to the type of query that is used.

Moreover, the application objectives can be summarized as:

- Propose an improvement in any of the processes of a CBIR according to the characteristics of the query.
- Apply each contribution to a practical application.
- Analyze the performance of our contributions and identify their pros and cons.

Along the chapters of this thesis we present both the scientific objectives and the application objectives related to each one of the pictorial query paradigms. Next we present the concrete outline of this work.

### 1.4 Thesis Outline

The outline of this thesis comprises the analysis of the state of the art of the CBIR, the analysis of each query paradigm, and the practical contributions. The following chapters of this thesis contain the following contents:

- Chapter 2: First of all we present the general framework of the CBIR techniques. We explain the structure of the content of an image and we relate it with the general the architecture of a CBIR system. The state of the art is also reviewed according to the modules that compose a retrieval system. Thus, we survey the processes of region extraction, the feature descriptors, the indexing schemes and the matching strategies.

- Chapter 3: This chapter presents the CBIR systems that are based in the query paradigm of image selection. We review the characteristics of the retrieval systems that use this kind of queries. We propose to encode the appearance of an image using a single descriptor based in the shape information. We analyze its performance in a set of images extracted from a video sequence.
- Chapter 4: The iconic paradigm is reviewed in the fourth chapter. The content of this chapter follows the same structure as the previous. First we introduce the state of the art and then we propose a contribution in a real application. In this case, we have worked in an surveillance application that applies an automatic description of the appearance of a person according the clothing. Our proposal can be applied in a general framework to identify the content of an image with a generic model of an object.
- Chapter 5: The paradigm of query-by-paint is also analyzed. We review the processes used to match a query sketch against the database of images. We pay attention to the diversity of approaches that can be applied to describe this kind of query images. We also expose the main difficulties that the matching process comprises. Next, we present an image descriptor to allow the retrieval of a of those database scenes that contain an instance of the query sketch. We have applied the system to a set of architectural maps and we have exposed some other examples in more general contexts.
- Chapter 6: Finally, the CBIR is analyzed from the point of view of the query-by-paint paradigm. In this chapter we include our main contribution in the process of image description. We present an algorithm to extract of the zones of interest of an image according to a perceptual analysis. Moreover, we use a set of painting queries to select the most suitable features to characterize a query-by-paint. We apply the retrieval system to several collections of images extracted from different sources: web images, magazines and newspapers.
- Chapter 7: Finally, in the last chapter we expose the conclusions and the future lines of our work.

# Chapter 2

## Content-Based Image Retrieval

In this chapter we first introduce the fundamental elements that define a CBIR system from the visual interpretation of an image. Then we proceed to detail the modules that form the architecture of the system. We present the most common techniques to describe the content of an image and to match it against a database collection. Content description comprises information extraction and feature characterization. Moreover, matching processes involve image indexing, similarity computation and global arrangement. Finally we expose the evaluation metrics of the performance of a CBIR system.

### 2.1 Fundamentals of image retrieval

In this section we model the content of an image according to the retrieval requirements. The image can be decomposed in a hierarchical structure that is processed by the content-based retrieval system. We also present its general architecture according to the modules that are involved in the retrieval process.

#### 2.1.1 Image content modelling

In this section we briefly define the modelling of the elements that compose an image. Given an image containing a scene, we can decompose its content in a top-down way as an hierarchical structure. This decomposition defines an hierarchical modelling that is graphically shown with an example in the Figure 2.1.

- Scene Level: In a highest level, a scene can be viewed as unique element containing the whole information of an image.
- Object Level: In a second level, a set of objects present in the scene can be distinguished in the image.
- Part Level: In the lowest level we can define a part of an object as an instance of a collection of features.

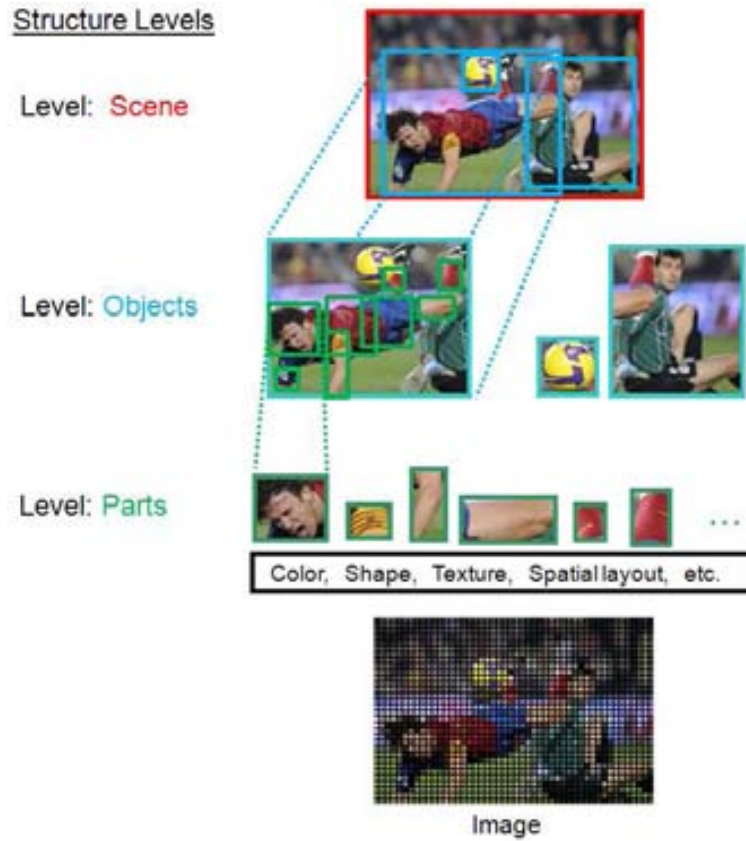


Figure 2.1: Levels of image content decomposition.

According to the hierarchical modelling of an scene we can identify a set of CBIR modules that are in charge to obtain the description of an image. In the next section 2.1.2 we detail the architecture of a CBIR system.

### 2.1.2 CBIR Architecture

The architecture of a CBIR system can be understood as a basic set of modules that interact within each other to retrieve the database images according to a given query. The Figure 2.2 shows them graphically.

From a general point of view we can distinguish between two kind of modules: those related to the image description and those involved in the matching procedures. Concretely, the description stage consists in a first process where the parts of the image are detected and a second phase where these parts are characterized according to certain features. In a CBIR system, the database images are preprocessed and are



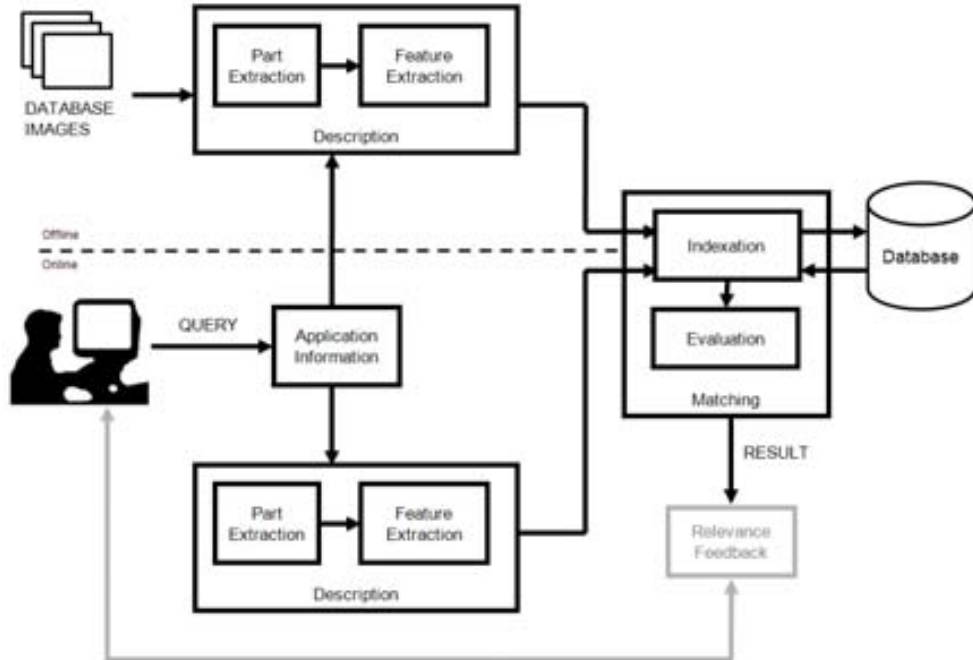


Figure 2.2: Modules of a CBIR system

stored in the database. To perform an efficient retrieval they can be indexed according to the features that define them. Then the matching process is in charge to put in correspondence the description of a query with the information of the database. The indexation structures allow to retrieve those images that contain instances of the features that define the query. Finally, when the query image consists in more than one part or one object, most of the systems need from a process where a global evaluation of the matching is done for each retrieved image. This global evaluation consists mainly in combining the collection of feature matching and checking the spatial arrangement of the objects or parts matched with the query.

Some systems make use of an optional module related to the relevance feedback. Relevance feedback provides the user tools to modify a query and capture precise needs through iterative feedback and query refinement. Thus, from the query results, the user can evaluate which images are relevant and the system can reuse their information in order improve the results. The relevance feedback techniques are out of the scope of this thesis, but some surveys can be found at [Har92] [CFB04] [ZH03].

In the following subsections we are going to detail the possible strategies for each of the submodules.

## 2.2 Part Extraction

Several strategies can be applied in order to extract the parts of an image. They can be grouped in several categories according to the procedures they use.

### 2.2.1 Whole Image

The simplest description of an image is the one that does not distinguish any parts in its content. The whole image is treated as a unity and the system uses a compact description that allows an efficient indexing. Nevertheless, this strategy is not able to distinguish the information that belongs to an object of interest from the information that belongs to the rest of the scene. Then, the whole image description is limited to those retrieval systems that work in a scene concept level (e.g. search for similar landscapes) or the systems that search for similar images containing a single object on a plain background.

### 2.2.2 Template Subdivisions

A predefined and static subdivision is a fast strategy to obtain a blind subdivision of the content of a scene. Some retrieval methods use fixed templates of regions that can be combined in several ways.

One example is the work of Stricker and Dimai [SD97] in which an image is divided in five regions of interest: an oval central region, and four corners. The system assumes that the objects of interest in a scene tend to be placed in the center of the image. Thus, the features concerning to the central region are given higher weight than the ones that lie in the rest of the scene.

Some partition-based approaches adopt a hierarchical representation of the spatial decomposition. The strategy consists in superimposing a fixed grid of rectangular cells over the images [LN98] [SLH99] [ZRZ02]. The cells at distinct hierarchical levels have various sizes and can overlap. For instance, the work of Malki suggests a multi-resolution quadtree approach to describe a scene [MBNW99].

### 2.2.3 Segmentation

Image segmentation is the partition of an image into a set of non-overlapped regions whose union covers the entire image. Some reviews on image segmentation are the survey works of Skarbek [WA94], Haralick and Shapiro [HS85] and Pal and Pal [PP93]. The techniques of image segmentation can be roughly classified into four groups according to the definition of region image: Feature-based, Region-based, Boundary-based and Hybrid.

#### Feature-based techniques

Feature-based techniques segment the image into regions according to the clusters of the features of the image values.

Typical feature based techniques are those called histogram thresholding techniques. In these techniques, a histogram is computed from all of the pixels in the image, and the peaks and valleys in the histogram are used to locate the clusters in the image. A refinement of this technique is to recursively apply the histogram-seeking method to clusters in the image in order to divide them into smaller clusters. This strategy can be applied to one dimensional data (grey scale images) such as the approach of Ohlander [OPR78]. When applied to color spaces, histogram thresholding is usually performed individually to each color component or combinations of them. The work of Lucchese [LM99a] applies the thresholding to each dimension of the image represented in the CIELUV color space. Otherwise, the approach of Otha, Kanade and Sakai [OKS80], suggests combining the RGB channels in in  $(R+G+B)/3$ ,  $(R-B)/2$  and  $(2G-R-B)/4$ .

Although histogram thresholding methods constitute the classical approach for feature based segmentation, there exists an extensive group of measurement space clustering techniques. In that group, the k-means algorithm [LM99b] [LM01] stands out to be one of the most extended technique for color image segmentation. Moreover, strategies such as the Mean Shift allow classify the elements of a N-dimensional space in an unsupervised number of clusters according to predefined measure of color similarity.

Other kind of clustering is the one related to the natural language. Typically, a learning procedure based on the human perception is the basis to obtain a labelling in natural language of the color pixels of an image (black, blue, pink, etc.). Some examples are the works of [BVB05], [vdWSV07] and [Moj02].

The accuracy of the feature-based techniques depends directly on how well the objects of interest on the image separate into the distinct clusters. The main drawback of the histogram-based techniques is that, since the clustering is done in the measurement space, there is no requirement for good spatial continuation and the resulting region boundaries can be noisy.

### Region-based techniques

In the Region-based approaches, a region is defined as a maximal connected set of pixels for which uniformity condition is satisfied. This uniformity is searched in the spatial domain of the image instead of the color space domain. Classical region based techniques comprise the region-growing techniques and split and merge techniques.

Region growing is one of the most simple and popular algorithms in image segmentation. Individual pixels (sometimes called seeds) are merged if their attributes (grey level, color or texture etc.) are similar enough. Color similarity can be established by computing the value of a homogeneity criterion. Each tested pixel is compared to its immediate neighboring regions. If a homogeneity criterion is fulfilled then the tested pixel belongs to region and all attributes of region are updated. Otherwise,

the tested pixel with a new label starts as a new region. The process of growing is continued until all pixels in image merge in regions as homogeneous as possible. Finding seeds starts typical region growing procedure. This procedure is named Seeded Region Growing (SRG). The region grows by adding neighboring pixels that are similar according to certain homogeneity criterion, increasing step by step the size of the region [AB94]. To overcome the initialization dependence of SRG, another procedure, which does not use pixel seeds, is proposed. This procedure employs a simple raster scan of the color pixels from left to right and from top to bottom. At the beginning of the algorithm each pixel has its own label (one-pixel regions) and the whole image is viewed as a set of four-connected regions. There exist different strategies of pixel linkage. Two main strategies are: single linkage and centered linkage. The single linkage strategy includes a pixel in the region if it is four-connected to this region and has color value in the specified range from neighboring pixels (one or more) already included in region. The centered linkage strategy includes a pixel in the region if it is four-connected to this region and has color value in the specified range from the mean color of an already constructed region.

Typically split and merge techniques [HP77] consist of two stages. First, the whole image is considered as one region. If the region does not comply with the homogeneity criterion, the region is split into four quadrants. Each quadrant is tested in the same way until every square region created contains homogeneous pixels. In the second stage all adjacent regions with similar attributes may be merged following other criteria.

Besides these two classical strategies, we could include in the region-based group another relevant segmentation that is based in the Markov Random Field (MRF). In MRF approach the segmentation problem is viewed as a statistical estimation problem where each pixel is statistically dependent in its neighbors. The segmentation is obtained finding the maximum a posteriori solution given the observed data [BS94].

### **Boundary-based techniques**

Instead of concerning in the homogeneity, in the boundary-based techniques, a region is viewed as a connected set of pixels bounded by abrupt changes in their limit pixels. Early approaches have used edge detection operators such as Sobel, Prewitt, Roberts or Canny [Can86] to extract image regions [PB99]. These approaches are simple but have difficulty in establishing the connectivity of the edge segments. To try to solve this problem another techniques emerged such as edge flow strategies [MM97] and adaptive models like snakes [SL01] or geodesic active contour models [PD99].

### **Hybrid-based techniques**

Hybrid techniques combine boundary and region criteria, to profit from both qualities and avoid their drawbacks. Boundary information can be used as a guidance of seed placement to perform a region growing. For instance, Cufi [CM00] makes use of the detected edges to place the seeds on their both sides and Sinclair [Sin99] derive the seed points for region growing as the peaks in the associated Voronoi image to

the color edge map. Region and boundary criteria can cooperate in the decision of merging two regions. Pavlidis [PL90] uses the information of contrast and boundary smoothness to resolve the merging stage on a over segmented result of a split-and-merge-procedure.

Finally, hybrid techniques include morphological watershed segmentation [dSB99]. The watershed method is generally applied to the gradient of the image. This gradient image can be viewed as topography with boundaries between regions as ridges. Given a drowning threshold, segmented regions, as lakes, appears in the image. After the watershed segmentation an optional merge of the neighboring regions can be applied.

Unlike the boundary based methods above, this technique is guaranteed to produce closed boundaries but encounters difficulties in images with regions are noisy and have blurred or indistinct boundaries.

#### 2.2.4 Contours

The contour part detection identifies as the parts of the scene the edges of an image. Its purpose is to capture important events and changes in properties of the world. The edges are likely to correspond to discontinuities in depth or surface orientation, variations in material properties or changes in the scene illumination. The contour detection strategies are the same as those used in the boundary based segmentation techniques. An image part is often identified by a group of connected pixel or a combination of them according to their geometric properties.

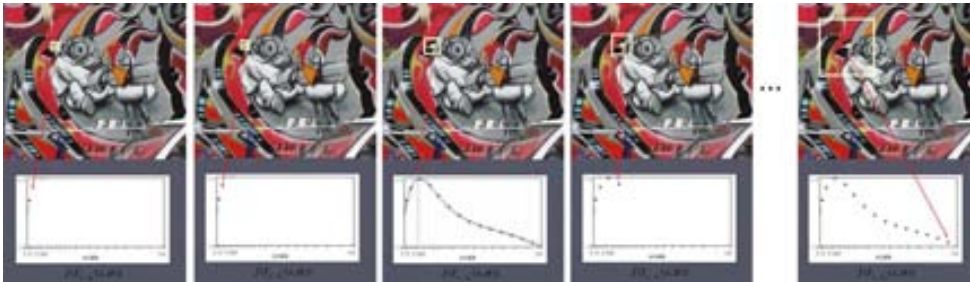
#### 2.2.5 Parts from local structures

Some methods provide a sparse representation of a scene depending on the detection of certain local structures such as blobs, edges, corners, etc. The local structures are obtained by a multi-scale analysis of the image and its detection is invariant to scale transformations. The image descriptions based on local structures rely strongly on the image intensities and have obtained excellent results in matching applications that deal with several instances of the same real object. A very complete survey of local structure detectors is the work of Tuytelaars [TM07]. Moreover, the paper of Mikolajczyk [MTS<sup>+</sup>05] provides a comparative study of these techniques. Next we overview the different techniques according to the structures of interest they focus on.

##### **Blob-based**

Lindeberg introduced the concept of scale-space to model the evidence that the detection of the world structures are dependent on the scale of observation. This phenomena is illustrated with the image of a tree: while the leaves can only be observed at close scales, the concept of forest only appears form a distant point of view. To describe the content of an image, he proposed a detector of blob-like regions based in the Laplacian-of-Gaussian operator (LoG). The detector was applied to the image at several scales and the local structures were selected at the scale that maximizes its

response [Lin93] [Lin98]. The figure 2.3 exemplifies the process of scale selection for a point of a scene.



**Figure 2.3:** Automatic scale selection related to the maximum response of LoG operator. The third image shows the optimal scale of the patch.

Variations on the original idea of Lindeberg have originated other Blob-based detectors. This way, Lowe improved the computational cost of the LoG operator approximating it by a difference of Gaussian filters (DoG) [Low04]. Nevertheless, both DoG and the LoG representation have the common drawback that local maxima can also be detected in the neighborhood of contours or straight edges. To solve this problem and penalize very long structures, the approach of Mikolajczyk used as a detection operator the determinant of Hessian matrix (DoH). Then the system selects the scale for which the trace and the determinant of the Hessian matrix simultaneously assume a local extremum [Mik02].

Moreover, the determinant of the Hessian was also proposed as a blob detector as a hybrid operator that combines it with the Laplacian. The Hessian-Laplacian strategy combines the spatial selection done by the DoH operator and a posterior scale selection performed with the scale-normalized Laplacian [MS04].

Notice that all these blob-based detectors do not show invariance to perspective transformations. This property is very interesting for those applications that intend to match objects despite the viewpoint angle. Thus, a final evolution of the Hessian-Laplacian is the so called Hessian-Affine that adapts the detected blobs to affine regions. The normalization process includes an iterative computation based second moment matrix of the central point of the blob [LG94] [MS04].

### Corner-based

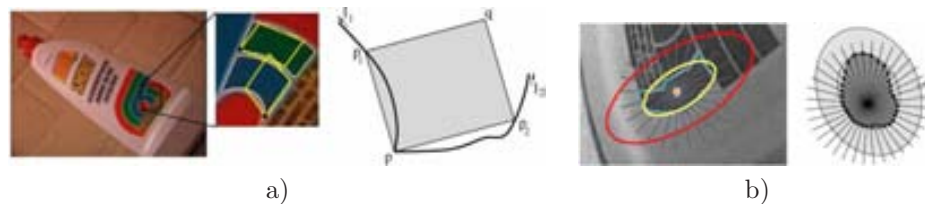
Moreover, the corners are another useful visual cue to describe the content of an image. This way, Mikolajczyk and Schmid proposed the Harris-Laplace detector that merges Harris corner detection with Laplacian based scale selection. The Harris-Laplace detector has also its affine version applying the same normalization process as the Hessian-Affine strategy [LG94] [MS04].

### Edge-based

The edge-based region (EBR) detector starts from a Harris corner point and a nearby edge extracted with the Canny edge detector. To increase the robustness to scale changes, these basic features are extracted at multiple scales. Two points move away from the corner in both directions along the edge. They define a parallelogram region based on local extrema of invariant function [TG04]. The Figure 2.4 shows an example of the intuitive idea of the process.

### Intensity-based

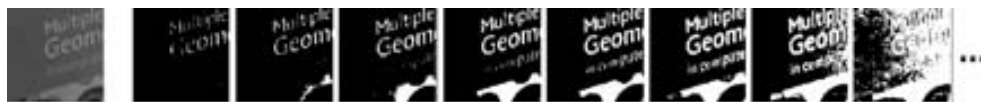
The intensity extrema-based region (IBR) detector starts from intensity extrema (detected at multiple scales) and studies the intensity pattern along rays emanating from this point [TG04]. These rays delineate regions of arbitrary shape, which are then replaced by ellipses. A graphical example is shown in the Figure 2.4.



**Figure 2.4:** a) The EBR detector starts from a corner point  $p$  and exploits nearby edge information. b) The IBR detector select intensity extrema and considers intensity profile along rays (reprinted from [TG04] [TM07]).

### Region-based

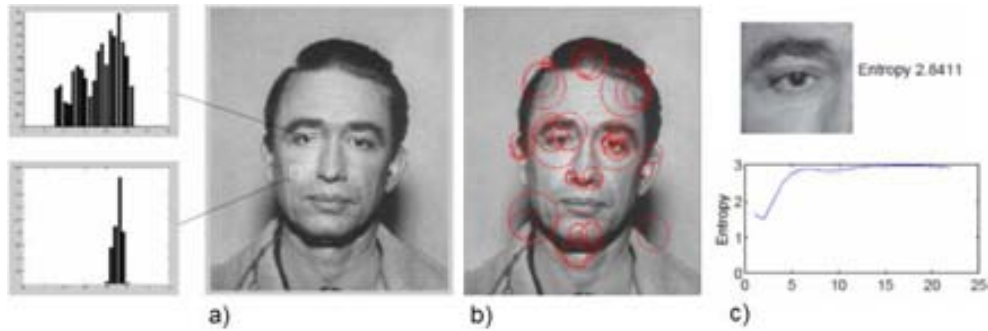
Another region detector based on the intensity of the pixels is the Maximal Stable Extremal Regions [MCMP02]. A Maximally Stable Extremal Region (MSER) is a connected component of an appropriately thresholded image. The word extremal refers to the property that all pixels inside the MSER have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all the pixels on its outer boundary. All extremal regions are found by thresholding the image with all possible thresholds [0-255] for normal gray scale images, and finding then all connected areas. MSER regions are those connected components which area do not change, or change as little as possible, when the threshold is changed (see Figure 2.5).



**Figure 2.5:** MSERs are connected components that remains along the sequence.

### Saliency-based

The case of the region detector proposed by Kadir is related to the entropy function [KB01]. The entropy is understood as a measure of information and is computed from the histogram of an image patch. Hence the algorithm searches through the scales those regions that maximize the entropy measure. The Figure 2.6 exemplifies the process.



**Figure 2.6:** a) Complex regions, such as the eye, exhibit unpredictable local intensity hence high entropy. b) Salient regions detected. c) Graphic of the saliency behavior on the eye point across scales (reprinted from [KB01]).

Strategy	Pros	Cons
Whole Image	Simple	Cannot distinguish objects in cluttered scenes
Template	Simple; Allow weighting of zones	Blind partition; Not adapted to object content
Segmentation	Adapt to object content of the scene	Can be slow; Threshold dependent; Disjoint part representation
Contours	Simple to compute; Rely on object boundaries	Sensitive to noise and changes of illumination
Local Structures	Accurate; Scale invariant; Allow overlapping of parts	Rely on intensity; Partial detection depending on the structure (corner, blob, etc)

**Table 2.1**  
PART EXTRACTION OVERVIEW



## 2.3 Feature Description

The selection of the features to represent an image is one of the keys of a CBIR system. In specific contexts and applications the process of feature extraction can be adapted to detect specialized attributes such as human faces. Nevertheless, general purpose image retrieval deals with generic features such as color, shape, texture, and spatial layout.

### 2.3.1 Color

Each pixel of the image can be represented as a point in a 3D color space. There exist different space models such as RGB, HSV, CIELuv, CIELab, YCbCr, HMM, Munsell or opponent color. If we want to describe an image by its color features, we have to first determine the color space to use. However, there is no agreement on which is the best representation, so the election depends on the special needs of the application. For image retrieval purposes, one of the desirable characteristics of an appropriate color space is its uniformity. Uniformity means that two color pairs that are equal in similarity distance in a color space are perceived as equal by viewers. In other words, the measured proximity among the colors must be directly related to the psychological similarity among them [MC98]. CIELuv and CIELab spaces are suitable models for image retrieval since they accomplish the requirement of spatial uniformity. From a different perceptual criteria, the HSV space has also some advantages since it represents an intuitive way of describing color. The three color components are hue, saturation (lightness) and value (brightness). HSV space is widely used in computer graphics, specially in the interfaces of the applications where the user has to browse the color space to select an instance of a color.

The color descriptors are related to mathematical operations of the pixel values represented in a certain color space. Some of the most popular descriptors are the color histograms [SB91], color moments [MM95], color sets [SfC95] color coherence vectors [PZ96], or color correlograms [HKM<sup>+</sup>97].

#### Color histograms

The color histogram is a commonly used image descriptor because it is easy to compute, robust, and fairly effective [SB91]. A histogram is the distribution of the number of pixels of the image according to the three components in a certain color space. The histogram can be defined for each component and can be quantized in a different amount of bins according to the description requirements. The bin quantization can be fixed at certain intervals or can be optimized using clustering methods to determine the  $K$  best colors for a set of database images. Each of these best colors will be taken as a histogram bin.

Selecting an optimal number of bins for a color histogram is a tradeoff between the discrimination power and the computational efficiency. The more bins a color histogram contains, the more discrimination power it has. However, a very fine bin quantization does not necessarily improve the retrieval performance in many applica-

tions where a certain degree of similarity is required. Furthermore, a histogram with a large number of bins is inappropriate for building efficient indexes for image databases.

One of the main drawbacks of the color histogram is that it does not take into consideration the spatial information of pixels. Thus very different images can be considered similar because they have similar color distributions.

### Color sets

To facilitate fast search over large-scale image collections, Smith and Chang proposed color sets as an approximation to the color histogram [SfC95]. They first transformed the image from its original color space into a perceptually uniform space, and then quantized the transformed color space into  $M$  bins. A color set is defined as a selection of colors from the quantized color space. Because color set feature vectors were binary, a binary search tree was constructed to allow a fast search.

### Color coherence vectors

Pass introduced [PZ96] de color coherence vectors (CCV) to include spatial information to the color histogram. CCV define as coherent those pixels of the image that belong to a connected component up to a certain size. Each histogram bin is partitioned into two types: coherent, if it belongs to a large uniformly-colored region, or incoherent, if it does not. Let  $\alpha_i$  denote the number of coherent pixels in the  $i$ th color bin and  $\beta_i$  denote the number of incoherent pixels in an image. This way, the CCV of the image is defined as the vector  $\langle (\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_N, \beta_n) \rangle$ . Due to its additional spatial information, it has been shown that CCV provides better retrieval results than the color histogram, especially for those images which have either mostly uniform color or mostly texture regions.

### Color correlograms

The color correlogram [HKM<sup>+</sup>97] was proposed to characterize not only the color distributions of pixels, but also the spatial correlation of pairs of colors. The first and the second dimension of the three-dimensional histogram are the colors of any pixel pair and the third dimension is their spatial distance. A color correlogram is a table indexed by color pairs, where the  $k$ -th entry for  $(i, j)$  specifies the probability of finding a pixel of color  $j$  at a distance  $k$  from a pixel of color  $i$  in the image. If we consider all the possible combinations of color pairs the size of the color correlogram will be very large, therefore a simplified version of the feature called the color autocorrelogram is often used instead. The color autocorrelogram only captures the spatial correlation between identical colors. Compared to the color histogram and CCV, the color autocorrelogram provides the best retrieval results, but is also the most computational expensive due to its high dimensionality.

### Color structure descriptor

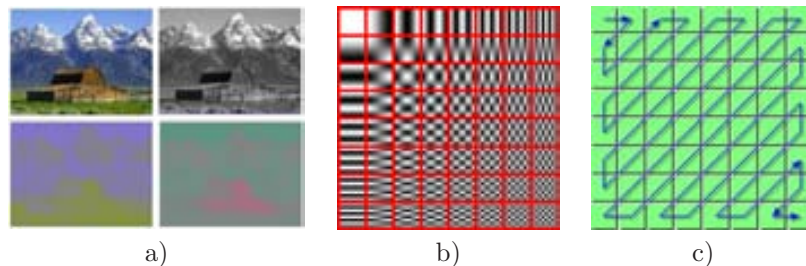
The Color structure is a descriptor included in the MPEG-7 standard [MSS02] that captures both color content and its spatial distribution. The extraction method takes into account all colors in a structuring element of 8x8 pixels that slides over the image. This descriptor characterizes the relative frequency of structuring elements that contain an image sample with a particular color. Color values are represented in the HMMD color space, which is quantized non-uniformly into 32, 64, 128 or 256 bins. The Color Structure descriptor provides additional functionality and improved similarity-based image retrieval performance for natural images compared to the ordinary color histogram.

### Color moments

All the above mentioned color descriptors make use of a quantization process to deal with a set of classes in the color space. Nevertheless, quantization strategies can introduce undesirable effects in those pixel values that lie in the border of two bins. To avoid the quantization drawbacks, Stricker and Orengo proposed using the color moments approach [MM95]. The mathematical foundation of this approach is that any color distribution can be characterized by its moments. Furthermore, since most of the information is concentrated on the low-order moments, only the first moment (mean), and the second and third central moments (variance and skewness) were extracted as the color feature representation.

### Color layout

As the color structure descriptor, the color layout descriptor (CLD) also represents the spatial distribution of colors in an image is a descriptor included in the MPEG-7 standard [MSS02] [KY01]. The CLD descriptor is a very compact representation of the color that allow very fast searches in databases and suits sketch based representations. In the computation process, the image is resized to a square of 8x8 pixels represented in the YCbCr color space. Then, each of the three components (Y, Cb and Cr) is transformed by 8x8 Discrete Cosine Transform (DCT). The descriptor contains 12 DCT coefficients in the zigzag-order, six for the luminance channel (Y) and 3 for each of the chromatic channels (Cb and Cr). (see Figure 2.7).



**Figure 2.7:** a) Original image on the top left and their channels in the YCbCr color space. b) Two-dimensional DCT frequencies. c) Zigzag order of the DCT coefficients.

COLOR DESCRIPTION OVERVIEW		
Strategy	Pros	Cons
Color histograms	Easy to compute; Robust	Quantization effect in bins; Contain no spatial information
Color sets	Easy to compute; Easy search on large collections	Quantization effect in color selection; Contain no spatial information
CCV	Include spatial information; Better performance than histogram	Quantization effect
Correlograms	Include spatial information; Better performance than histogram and CCV	Quantization effect; Computational expensive
Color Structure	Include spatial information; Better performance than histogram; MPEG7 standard	Quantization effect; Dependence on the size of the structuring element
Moments	Easy to compute; No quantization; Compact descriptor	Coarse description
Color Layout	No quantization; Compact descriptor; Suitable for sketch representation; MPEG7 standard	Coarse description

**Table 2.2**  
COLOR DESCRIPTION OVERVIEW

### 2.3.2 Texture

Texture is a very general notion that can be attributed to almost everything in nature. For a human, the texture relates mostly to a specific, spatially repetitive structure formed by repeating a particular element or several elements in different relative spatial positions. Generally, the repetition involves local variations of scale, orientation, or other geometric and optical features of the elements.

Texture is one of the most complex visual cues that uses to be present in most of the images whether containing natural scenes (e.g. clouds, trees) or manmade objects (e.g. bricks, textiles). The interest in characterizing texture patterns have raised to the development of a wide variety of descriptors. Nevertheless, experimental works show that, despite more than 30 years in research on texture descriptors still none texture feature can convey a complete description of the texture properties of an image. Therefore a combination of different texture features will usually lead to best

results [DKN08].

Next we overview some of the most outstanding state-of-the-art descriptors such as the co-occurrence matrix, Tamura and Wold features, simultaneous auto-regressive models, fractals, wavelets or Gabor filters.

### Co-occurrence matrices

Many statistical texture features are based on co-occurrence matrices representing second-order statistics of grey levels in pairs of pixels in an image. A co-occurrence matrix shows how frequent is every particular pair of grey levels in the pixel pairs, separated by a certain distance  $d$  along a certain direction  $\alpha$  (see Figure ). In the early 1970s, Haralick et al. proposed the co-occurrence matrix representation of texture features [HSD73]. Many other researchers followed the same line and further proposed enhanced versions. For example, Gotlieb and Kreyszig studied the statistics originally proposed in [HSD73] and experimentally found out that contrast, inverse deference moment, and entropy had the biggest discriminatory power [GK90].

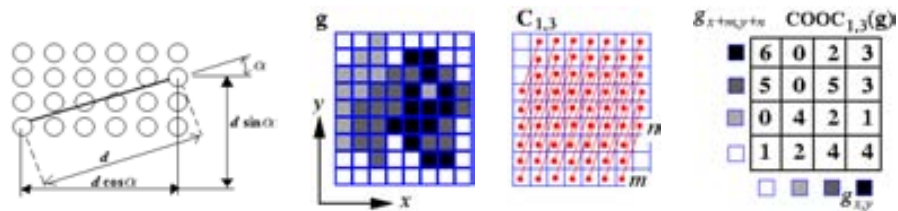


Figure 2.8: Example of construction of a co-occurrence matrix

### Fractal models

Fractals are a set of self-similar functions in the so-called fractal dimension. The fractals models achieve perceptual correlation with the roughness of image textures. Fractal texture occur commonly in natural images [Pen84] and have been used in several works of image retrieval [KMN98] [BTTJ05].

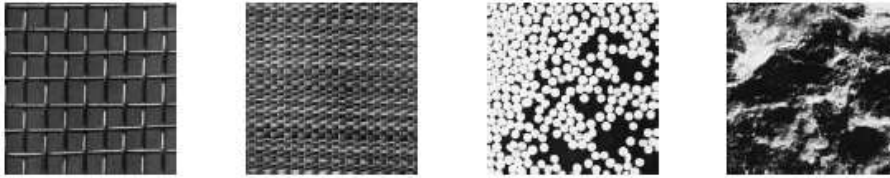
### Tamura and Wold Features

Tamura and Wold features were both designed in accordance with psychological studies on the human perception of texture. Since the texture properties are visually meaningful, these features are very attractive in image retrieval systems. This way, the the user can use a verbal description of the image end texture properties can be represented in a user-friendly interface.

The Tamura features include single-valued measures related to the properties of coarseness, contrast, directionality, linelikeness, regularity and roughness [TMY78]. Directionality, contrast and coarseness were used in some early retrieval systems like

QBIC and Photobook. Moreover, the features related to regularity, coarseness and directionality were accepted as the texture browsing descriptor included in the MPEG-7 standard [MSS02]. Figure 2.9 shows an example of the regularity feature.

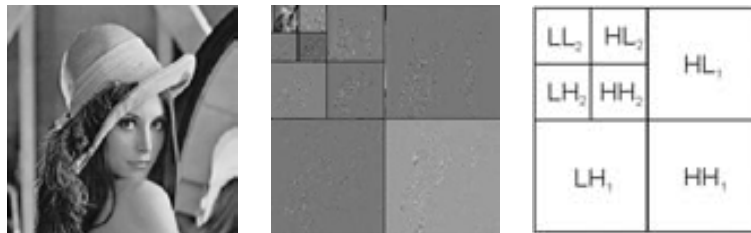
The Wold features model a texture as a random field and are computed from its decomposition into three mutually orthogonal components [LP96]. The three Wold components, harmonic, evanescent, and indeterministic, correspond to periodicity, directionality, and randomness respectively. Periodic textures have a strong harmonic component, highly directional textures have a strong evanescent component, and less structured textures tend to have a stronger indeterministic component.



**Figure 2.9:** Examples of regularity. From left to right: highly regular, regular, slightly regular and irregular.

### Wavelet Transformation

The wavelet transformation provides a multi-resolution approach to texture analysis and representation [SLW<sup>+</sup>97]. Wavelet transforms decompose a signal with a family of basis functions obtained through translation and dilation of a mother wavelet. The computation of Wavelet transforms has its origins in approximation theory and signal processing. Applied to an image, a 2D signal framework, it involves recursive filtering and sub-sampling. At each level, the signal is decomposed into four frequency sub-bands, LL, LH, HL, and HH, where L denotes low frequency and H denotes high frequency (see Figure 2.15).



**Figure 2.10:** Example of the wavelet transform. Large coefficients are visualized in black or white and are located, at each scale, along edges of the image. Otherwise, regular areas, for example on the shoulder, present small coefficients that are visualized in gray. Fine scale are in the bottom right. There is three kinds of coefficients per scale: horizontal, vertical and diagonal.

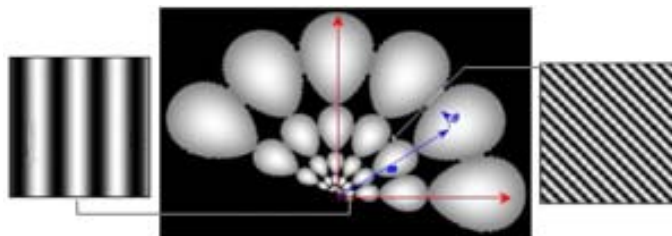
Two major types of wavelet transforms used for texture analysis are the pyramid-structured wavelet transform (PWT) and the tree-structured wavelet transform (TWT). The PWT recursively decomposes the LL band. However, for some textures the most important information often appears in the middle frequency channels. To overcome this drawback, the TWT decomposes other bands such as LH, HL or HH when needed. After the decomposition, feature vectors can be constructed using statistical measures such as the mean and standard deviation of the energy distribution of each sub-band at each level [LF93].

### Simultaneous Auto-Regressive models

The Simultaneous Auto-Regressive model (SAR) is an instance of Markov random field (MRF) models, which have been very successful in texture modelling in the past decades [LWY95]. Compared with other MRF models, SAR uses fewer parameters. In the SAR model, pixel intensities are taken as random variables. The intensity at each pixel can be estimated as a linear combination of the neighboring pixel values and an additive noise term. Because the SAR model itself is very vulnerable to rotation, improvements of the SAR include rotation invariance (RISAR) . Moreover, other models (MRSAR) account for a multi-resolution to describe textures of different granularity and allow multi-scale texture analysis [MJ92].

### Gabor Filter Transformation

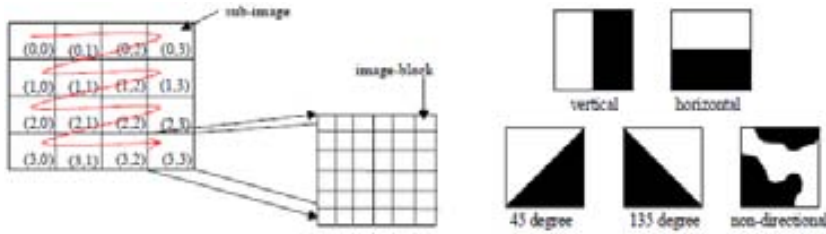
The Gabor filter transformation is regularly used in image retrieval to describe the global structure or texture of images. The features based on the Gabor filters are understood as description of the orientation and frequency of the patterns in that emulates the processes of the primary cortex of the human visual system. Texture descriptors based on Gabor filters were also included in the MPEG-7 standard as a descriptor called Homogeneous Texture [MSS02]. The image is filtered with a bank of orientation and scale sensitive Gabor filters shown in the Figure 2.11. The frequency space is partitioned into 30 channels with 6 equal angular divisions at  $30^\circ$  intervals and 5 division in the radial direction. Then, the means and the standard deviations of the filtered outputs are used as the descriptor components.



**Figure 2.11:** Representation of the bank of 30 Gabor filters in polar coordinate system used for the Homogeneous Texture descriptor of the MPEG7 standard.

### Edge Histogram descriptor

The Edge Histogram Descriptor (EHD) is included in the texture features of the MPEG-7 standard [MSS02]. EDH represents the spatial distribution of five types of edges: four directional edges and one non-directional edge. In the computation process, the content of the image is divided in a two step process (see Figure 2.12). First is decomposed into 4x4 subimages and further each subimage is partitioned into non-overlapping square imageblocks. The size of the blocks depends on the resolution of the input image. The edges in each image-block are categorized into six types: vertical, horizontal, 45° diagonal, 135° diagonal, nondirectional edge and no-edge (See Figure 2.12). Thus, a 5-bin edge histogram of each subimage can be obtained. Finally, each bin value is normalized by the total number of image-blocks in the subimage and the normalized bin values are nonlinearly quantized.



**Figure 2.12:** EHD descriptor: Image subdivision and edge types.

Strategy	Pros	Cons
Cocurrence matrices	Intuitive	Long description
Tamura and Wold features	Perceptual meaning	Not very effective for finer texture discrimination
SAR models	MRSAR has better performance than Wavelets and Wold	No perceptual meaning; Computationally expensive
Fractals	Suitable for natural textures	No perceptual meaning
Wavelets	Multi-resolution analysis	Sensitive to noise
Gabor filters	Emulate human visual perception; MPEG7 standard	Results in over-complete representation of image; Computationally expensive
EHD	Easy to compute; MPEG7 standard	Dependence on the contour extraction; Angular quantization

**Table 2.3**

TEXTURE DESCRIPTION OVERVIEW



### 2.3.3 Shape

Shape is an important visual feature and it is one of the primitive features for image content description. However, shape content description is a difficult task because it is difficult to define perceptual shape features and measure the similarity between shapes. The shape features are meant to characterize an object according to the different types of source information they use. We distinguish between three types of shape descriptors: boundary-based, region-based and aspect-based.

Boundary-based methods, are based on the outer contour of the shape, in other words, the physical limits of the object. Region-based methods, in contrast, use the entire shape region as the mask of the object for the calculation of shape descriptors. Finally, appearance based methods go beyond the region information and aim to codify the values of the inner pixel of an image region.

The information types can be understood as an hierarchy where the information used in each category is included in the parent one. Then, from the original information used by the aspect-based descriptors we can extract the region information as a mask of the object, and from the mask is straightforward to obtain the contour. Traditionally, shape representations were divided into the two first categories, boundary and region. Nevertheless, appearance based methods have been gaining relevance in the works of the last years. These descriptors combined with the decomposition of the image parts according to points of interest are a powerful combination to perform object matching and retrieval.

The literature on shape descriptors is large, a deeper explanation on the computation processes and classification schemes can be found in the surveys of Zhang [ZL04], Safar [SSS00] and Veltkamp [VH99].

#### Contour based shape descriptors

Contour based descriptors focus on the boundary information of an image. We have distinguished four types of contour based descriptors according to the constraints of the information they use. Next we describe the methods that encode a shape as a close curve, the ones that subdivide information, the methods that focus on a subset of points and, finally, the systems that rely on perceptual meaningful structures.

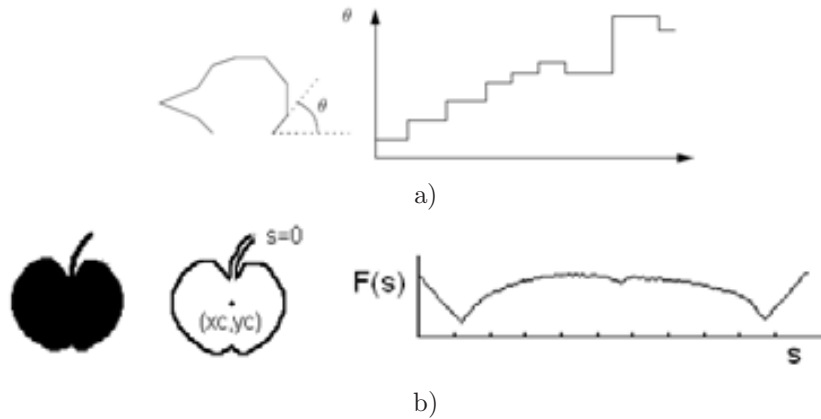
**Closed curves** Some of the contour based descriptors are constrained to the shapes made of a single closed curve. Some strategies model the curve as the response of a certain function and model it with the Fourier transformation. Other methods Curvature Scale-Space describes curves according to the response of an iterative Gaussian smoothing.

- **Fourier Descriptors:** They describe the shape of an object with the Fourier transform of its contour. The contour is represented as closed sequence of successive boundary pixels that can be represented with tree strategies: curvature

features, centroid distance, and complex coordinate function.

The curvature at a point along the contour is defined as the rate of change in tangent direction of the contour. The curvature function is an evolution of the turning function, which measures directly the angle of tangents along the contour (see 2.13 a)).

Being  $(x_c, y_c)$  the centroid of the object, the centroid distance is defined as the distance function between boundary pixels and the centroid (see 2.13 b)). Finally, the complex coordinate function is obtained by representing the coordinates of the boundary pixels as complex numbers. Denoting the contour pixels  $(x_s, y_s)$ , where  $0 \leq s \leq N - 1$  and  $N$  is the total number of pixels on the boundary, the complex coordinate function is expressed  $F(s) = (x_s - x_c) + j(y_s - y_c)$ .

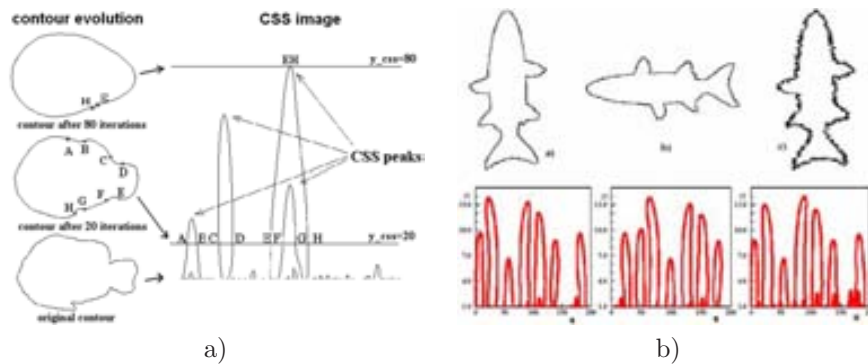


**Figure 2.13:** a) Turning function (reprinted from [VH99]). b) Centroid Distance function (reprinted from [ZL04]).

The Fourier transforms of these three types of contour representations generate three sets of complex coefficients, representing the shape of an object in the frequency domain. Lower frequency coefficients describe the general shape property, while higher frequency coefficients reflect shape details. To achieve rotation invariance (independence of the choice of the initial reference point), only the amplitudes of the complex coefficients are used and the phase components are discarded. To achieve scale invariance, the amplitudes of the coefficients are divided by the amplitude of DC component or the first non-zero coefficient. The translation invariance is obtained directly from the contour representation. To ensure the resulting shape features of all objects in a database have the same length, the boundary of each object is re-sampled to  $M$  samples before performing the Fourier transform.

- **Curvature Scale-Space:** In order to create a CSS description of a contour shape, a set of equidistant points are selected on the contour. The point selection process starts from an arbitrary point on the contour and follows the contour

clockwise. The process consists in finding curvature zero crossing points of the shape contour, named key points, and encode the evolution of these points by applying an iterative Gaussian smoothing. The position of key points are expressed relative to the length of the contour curve and disappear as the Gaussian kernel grows. The curve smoothes gradually it is converted into a convex curve that do not contain any zero crossing. The descriptor signature encodes the moment (the size of the Gaussian kernel) and the position on the curve where each key point disappears. Figure 2.14 shows graphically the construction of the CSS descriptor.

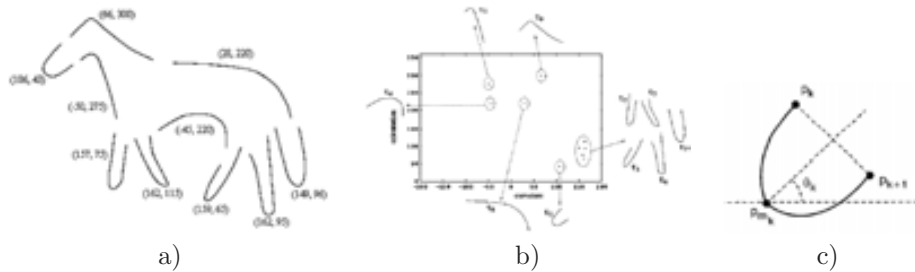


**Figure 2.14:** a) Construction of the CSS descriptor. b) The rotation transform causes a shift of the signature. Noise causes the apparition of small peaks.

Notice that the CSS descriptor cannot encode unconnected components, so it is designed to characterize a single closed contour curve. This descriptor proposed by Mokhtarian [MM92] has been included in the MPEG-7 standard [MSS02]. The shape description consists in the signature peaks characterized by the value pairs of smoothness degree and position on the curve  $(\sigma, u)$ . The peaks are ordered by decreasing values of  $\sigma$  and the number of peaks plus the eccentricity and circularity are added to the descriptor.

**Boundary Decomposition** To archive tolerance to partial occlusions the shape can be modelled as a composition of parts. The parts can be understood as the protrusions of the curve or can be approximated by vectors.

- **Local Protrusions:** Berretti [BDBP00] used the curvature zero-crossing points from a Gaussian smoothed boundary are used to decompose the shape into subparts. These subparts called tokens are characterized by their maximum curvature and their orientation. The similarity between two tokens is measured by the weighted Euclidean distance.



**Figure 2.15:** a) Horse shape is partitioned in correspondence with minima of the curvature function. b) The tokens are arranged in the feature space defined by the curvature and the orientation. c) Angle  $\theta$  describes the orientation.

- **Vectorial Approximations:** A common approach to represent the shape boundary involves the use of consecutive segments obtained from a polygonal approximation of the contour [MG95]. A set of vectors describing a part of the contour are characterized by their geometrical features. As a graphical example the Figure 2.16 shows the representation proposed by Stein [SM92]. This strategy uses vector chains obtained by multiple levels of polygonalization. Groups of consecutive segments, called supersegments, are encoded by the number of segments that contain, angular differences between segments, the location of the midpoint of the middle segment and the eccentricity (see Figure 2.16).



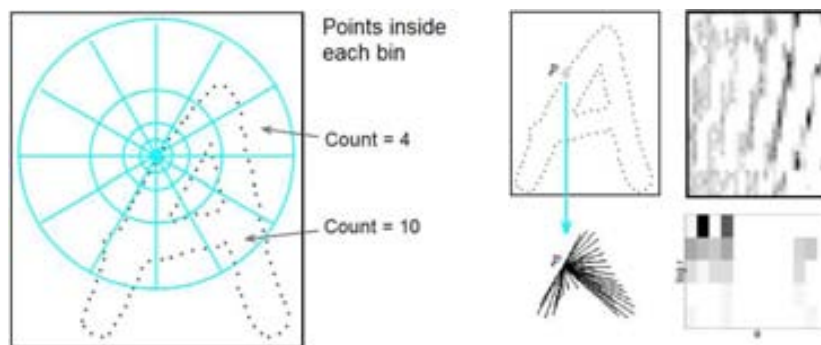
**Figure 2.16:** a) Different polygonal approximations. b) Supersegment features (reprinted from [SM92]).

Vector chain approximations benefit from the decomposition of the boundary in subparts to allow partial matchings. Nevertheless they are restricted to encode groups of consecutive segments. Other vectorial strategies deal with different grouping strategies that do not depend on the vector connectivity. This is the case of the work of Huet [HH98] that describe a shape according to the geometrical features of each segment in relation to the  $N$  closest ones.

Since the polygon approximation can be very sensible to the image noise, this methods are expected to work well for manmade objects, but are of difficult application for natural scenes [ZL04].

**Contour Points** Other methods do not attempt to encode continuous contour information but rely on the spatial relation of subset of points. This is the case of the shape context descriptor or the corner triangulation.

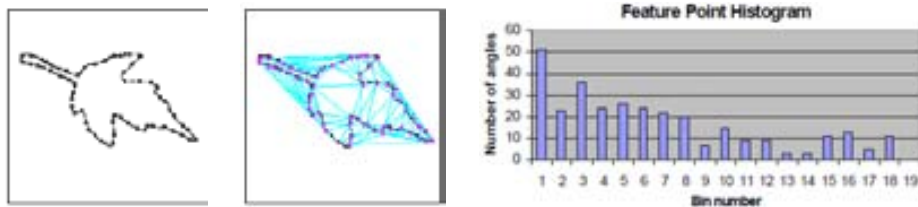
- **Shape context:** The shape context descriptor is computed by the statistical features obtained in the points of the contour of an object. For each contour point an statistic of the position of the other points in the shape is build. Then, by centering a circular grid of angular and radial cells, the histogram of points that lie in each cell is computed. The description on each point is a two-dimensional histogram corresponding to the log polar coordinates of the grid (see Figure 2.17). To obtain the descriptor of the whole shape, each 2D histogram is flattened into a vector. Moreover, the shape descriptor is also constructed as 2D histogram, where the rows correspond to the partial descriptor of each contour point. In order to compare two shapes, a matching process, such as the Hungarian method, is applied to put in correspondence the descriptions of the contour points of both elements [BM00].



**Figure 2.17:** Construction of the Shape Context descriptor

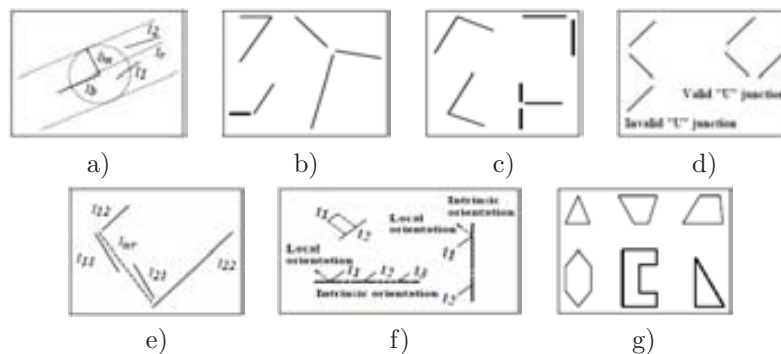
Shape context has been proved to be a powerful descriptor to match shapes in presence of several distortions. However, its main drawback is its dependence on a matching algorithm to compare the shapes. Therefore, this fact makes the descriptor not suitable for retrieval applications that need a straightforward shape indexing.

- **Contour Point Triangulation:** Some works encode the boundary information using specific points such as the corner ones. By instance, the work of Tao [TG99] use a Delaunay triangulation to encode the spatial arrangement of these feature points. The process consists in discretizing the angles produced by this triangulation and arrange them into a histogram. The Figure 2.18 contains an example of the shape description of an object.



**Figure 2.18:** The shape descriptor is defined by the histogram of angles extracted from the Delaunay triangulation of the contour points (reprinted from [ZL04]).

**Perceptual cues** Finally, the contour information can be described with a set of local structures that are grouped according to perceptual criteria. These structures can provide a semantical interpretation to the encoded shape. Several works use perceptual grouping laws to construct higher meaningful structures from lower-level features. Features can be extracted hierarchically to form structures like line segments, longer linear lines, retained lines, co-terminations, L junctions, U junctions, parallel lines, parallel groups, significant parallel groups and polygons (see Figure 2.19).



**Figure 2.19:** Visualization of the groupings. a) Longer linear line. b) Co-terminations. c) L junction. d) U junction. e) U junction. f) Parallel groups. g) Polygons (reprinted from [IA02]).

The work of Hu [HP96] uses curve features and junctions to describe road maps. Moreover he focuses in special strokes such as loops and dots to describe cursive handwritten words. In the work of Kliot [KR98] the number of geometric entities present in an image, such as circles and ellipses, are used to provide a filter in the retrieval process of object images. The CIRES search engine use this perceptual cues to classify the images into semantical classes such as manmade objects and landscapes [IA02]. A more specific application called ARTISAN was proposed by Eakins [ESB96] to deal with the retrieval of trademark images.

Primitive	Strategy	Pros	Cons
Closed curve	Fourier	Cyclic descriptor; Compact	Only for closed curves
Closed curve	CSS	MPEG7 standard; better than Fourier	Only for closed curves; Not rotation invariant
Boundary parts	Curve decomposition	Allow partial occlusions; Matching with Euclidean distance	The descriptor of sub-parts have low discriminance
Boundary parts	Vector decomposition	Allow partial occlusions; Suitable for manmade objects	Difficult to approximate natural structures
Boundary points	Shape Context	Allow partial descriptions; Based on shape points	Matching computation
Boundary points	Delaunay triangulation	Allow partial descriptions; Based on shape points	Quantization of angles
Geometric compositions	Perceptual cues	Association to High-level meaning	Based of predefined models; Related to specific context

Table 2.4

CONTOUR BASED SHAPE DESCRIPTION

### Region based shape descriptors

In addition to the contour information the region based descriptors use information of inner pixels of the shape understood as a binary mask. The most outstanding descriptors describe the shape independent global measures, with moment measures, of with transformation coefficients. Other methods extract the skeleton of the shape and describe the shape according to its features. Finally we can identify a group of methods that uses grid based description of a shape.

**Collection of single valued global features** Global contour shape representation techniques usually compute a multi-dimensional numeric feature vector from single valued measures characterizing the entire object. Common simple global descriptors are area, circularity, eccentricity, compactness, major axis orientation, Euler number, etc. These simple global descriptors usually can only discriminate shapes with large differences, therefore, they are usually used as filters to eliminate false hits or combined with other shape descriptors to discriminate shapes [ZL04]. The first three of these, for instance, were used in QBIC [FSN<sup>+</sup>95]. Definitions for most of these measures can be found in [EBS96] and [PI97]. Figure 2.20 contain some examples of this general features.

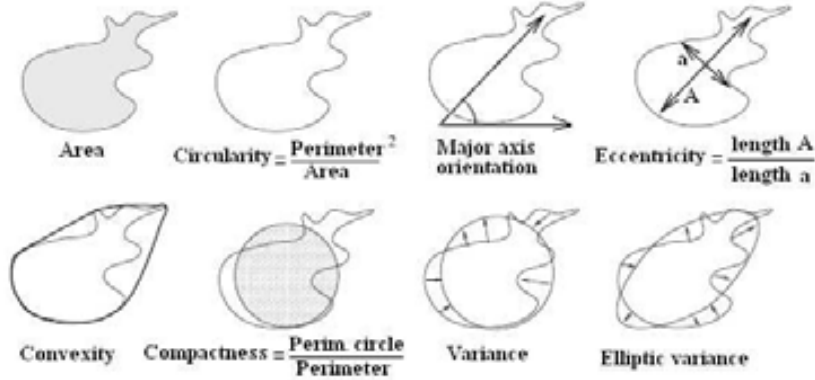


Figure 2.20: Simple shape global features

**Region moment** The moment based approaches can be defined as projections of the function describing the object onto a set of characteristic moment functions. The raw image moments  $M_{ij}$  of an image  $I$  are calculated by:

$$M_{p,q} = \sum \sum x^p y^q I(x,y) \quad p, q = 0, 1, 2, \dots$$

Moments  $M_{p,q}$  are uniquely determined by the function of the image function  $I(x,y)$  and viceversa, the image can be reconstructed by the moments. Moment-based descriptors express pixel distribution within a region allowing to describe complex shapes made of disconnected regions. They are usually concise, robust, easy to compute and match. The disadvantage of moment methods is that it is difficult to correlate high order moments with the shape physical features. Among the large literature on moment computation, the geometrical moments of Hu and the Zernike-based ones, outstand to be the most efficient options for shape description.

- Geometrical moments: Hu proposed the first work using the moments for shape description [Hu62]. He proved that moment-based shape description is information preserving. When a shape is represented as a mask, the function  $I(x,y)$  takes the binary values of its region image. Then, low order moments can be related to perceptually feasible characteristics :  $M_{0,0}$  correspond to the area of the shape,  $\frac{M_{1,0}}{M_{0,0}}$  and  $\frac{M_{0,1}}{M_{0,0}}$  gives the  $x$  and  $y$  coordinates of the centroid for the region  $M_{2,0}$ ,  $M_{1,1}$ ,  $M_{0,2}$  are related to the elongation of the region and orientation of its major axis. Even though the moments describe in a unique way an image, not all the moment values are suitable for shape definition. Hu proposed a set of nonlinear combinations of the lower order moment to obtain seven values invariant under translation, scaling and rotation. Hu moment, also called geometric moments, were applied in works for character recognition and as trademark retrieval [JV98]. Even though geometric moments are computationally simple, the main problem with geometric moments is that only a few invariants derived from lower order moments. They are not sufficient to accurately describe shape



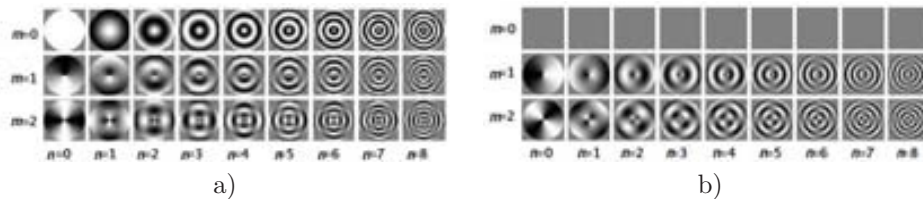
and higher order invariants are difficult to derive [ZL04].

- **Orthogonal moments:** The formulation of the algebraic moment can be extended by replacing the original kernel  $x^p y^q$  with a more general one. Thus, the algebraic invariant descriptor can be improved by defining orthogonal moment invariants. The orthogonal moments have the property that allow images to be recovered from them (see the example of Figure 2.21). Teague [Tea80] introduced the orthogonal moment of Zernike by replacing the original kernel with its polynomial formulation. Several experiments compared the performance of Zernike moment and its variant, the pseudo-Zernike moment, against other strategies: Legendre moments, geometric moments, complex moments and rotation moments. The results conclude that Zernike moment are the most desirable for shape description because they outperform other methods regarding to noise tolerance and reconstruction power [TC88].



**Figure 2.21:** Example of the reconstruction of the left image with the Zernike moments of order 5, 10, 15 and 20.

Moreover, MPEG-7 standards adopted a region shape descriptor based on Zernike moment [MSS02]. This descriptor, named Angular Radial Transform (ART), achieves rotation and scale invariance and uses Zernike moment for fast computing [VR08]. The ART is a 2D complex transform defined on a unit disk in polar coordinates. As can be observed in the Figure 2.22, it encodes the distribution of the pixels in according to the angular and radial subdivision.



**Figure 2.22:** a) and b) Real parts and imaginary parts of the ART basis functions. The imaginary parts have similar shape to the corresponding real parts but with different phases. Note that the brighter the region, the higher the value.

ART descriptor consists in 35 normalized and quantized magnitudes of the coefficients according to twelve angular sectors ( $n$ ) and three angular subdivisions ( $m$ ). It is suitable to describe binary images but it is not robust to perspective deformations. Some extensions have been proposed to apply ART to color images and adapt it to all possible rotations and to perspective deformations [RCB04].

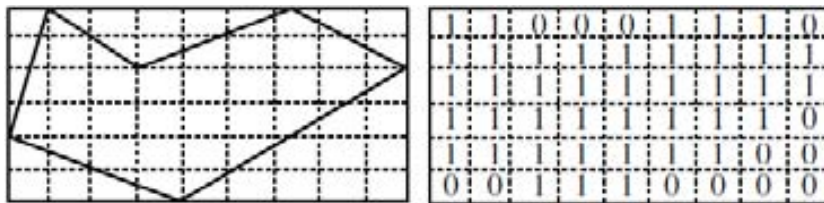
**Medial axis descriptors** The skeleton of a region can be also employed for shape description. A skeleton may be defined as a connected set of medial lines along the limbs of a region. The medial axis representation captures the regional interaction of the boundaries. Then, in the skeleton approach the similarity of structure is more apparent than in boundary representation [SK01].

The skeleton is formed by a set of segments that can be represented according to certain criteria [RdSTdFC03]. Some approaches models the skeleton as a set of parts that can be approximated by a certain geometric primitives [ZY95]. Other approaches like the shock graphs represent them in an hierarchical structure.

The skeleton use to be restricted to describe shapes made of a single connected component. They are usually represented by a graph and the matching between two shapes is done with the use of an edit function [PSZ99] [SKK04].

This shape representation is suitable to describe flexible shapes, understood as shapes that include articulation of parts. Nevertheless the process of skeleton extraction is sensible to image noise and it is of difficult application in regions obtained from a coarse image segmentation.

**Grid based methods** The grids shape descriptor is proposed by Lu and Sajjanhar [LS99] and has been used in the MARS system [COBPM00]. Basically, a grid of cells is overlaid on a shape, the grid is then scanned from left to right and top to bottom. The resulting descriptor of the grid based method is a bitmap (see Figure 2.23). The binary Hamming distance is used to measure the similarity between two shapes.



**Figure 2.23:** Example of the grid based method descriptor (reprinted from [SSS00]).

Improved versions of the Grid based methods include resolution adaptability [COBP<sup>+</sup>99] or circular grid [Gos85]. The advantages of the grid descriptor are its simplicity in representation, conformance to intuition. The main problem with this method is that the major-axis based rotation normalization can be sensitive to noise and occlusions.

Primitive	Strategy	Pros	Cons
2D Points	Global features	Easy to compute; Suitable for fast filtering	Low discriminatively power
2D Points	Hu moments	Fast computation; Invariant to translation, scale and rotation; Can describe unconnected components	Low discriminatively power
2D Points	Zernike moments	Allow shape reconstruction; Robust to noise; Can describe unconnected components	No perceptual interpretation
2D Points	ART	Can describe unconnected components; Compact; MPEG7 standard;	No perceptual interpretation
2D Points	Grid	Easy to compute; Intuitive description	Sensible to rotation normalization, occlusions and noise
Medial Axis	Skeleton	Represent internal structure; Suitable to model flexible objects	Skeleton computing is sensible to noise

**Table 2.5**

REGION BASED SHAPE DESCRIPTION

### Appearance based shape descriptors

This kind of low level features plays a crucial role in the object detection. This descriptors can be mainly extracted from the intensity values or from the gradient information.

**Intensity based descriptors** The intensity based descriptors are computed directly from the image pixel values. The intensity values of an image patch can be used as feature vectors, but in many cases require a good photometric normalization and are suitable for small windows. As an example, the work of Rowley [RBK96] trained a neural network for human face detection using the image intensities in 20x20 sub-window.

A key work on the face detection area was the one of Viola and Jones [VJ04]. They introduce the Haar-like features which have become an outstanding framework in the context of object detection [MPP01]. The initial set of features was further expanded by Lienhart an efficient set of 45° rotated features [LM02]. Haar features consist in a set of patterns that can be designed to detect certain structures such as edges, lines or center surround features, etc. (see Figure 2.24). Haar-like features are

related to the Haar wavelets. They can be understood as spatial filters that contain two kind of regions and compute the difference between the sum of image pixels under both types of regions. Haar-like features are very attractive for real-time applications because they can be efficiently computed with the integral image.

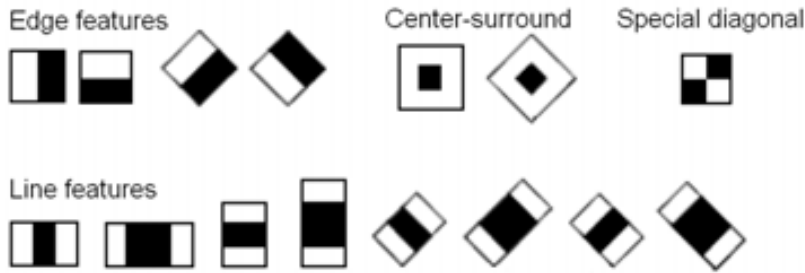


Figure 2.24: Haar-like features (reprinted from [LM02]).

**Gradient based descriptors** The gradient features are characterized to robust to global intensity changes. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of the intensity gradients.

One most successful gradient descriptors is the Scale-invariant Feature Transform (SIFT) proposed by Lowe [Low99]. The descriptor is a vector of 128 values that describes the histograms of the orientations in a grid of 4x4 sub-windows (see Figure 2.25). As a first step, is a computation of the dominant gradient orientation performed in order to obtain rotation invariance.

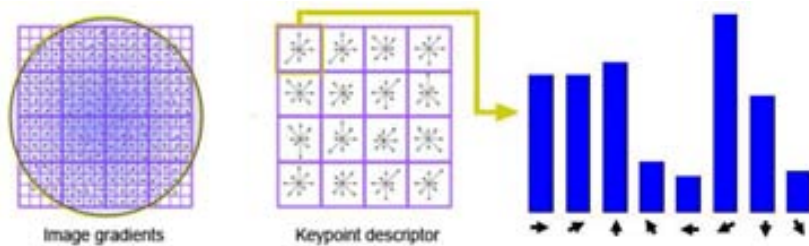


Figure 2.25: Computation of the SIFT descriptor. The gradient magnitude and orientation are computed at each image point. They are weighted by a Gaussian window and then accumulated into orientation histograms. SIFT descriptor is a vector of 128 values.

Variants of the SIFT descriptor include the SURF descriptor introduced by Bay et al. [BTG06]. It follows similar principles of Lowe's SIFT descriptor but achieves higher computation speed making an efficient use of integral images. Moreover, the

Gradient Location and Orientation Histogram (GLOH) is another SIFT-like descriptor that considers more spatial regions for the histograms. The higher dimensionality of the descriptor is reduced to 64 through principal components analysis.

The descriptor called Histogram Oriented Gradient (HOG) counts occurrences of gradient orientation in localized portions of an image [DT05]. This method is similar to SIFT descriptor but differs in that it computes on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved performance. HOG descriptor has been successfully applied in the detection of humans in complex scenes.

Other features also based on gradients and applied to pedestrian detection are the shapelets and the edgelets [SM07] [WN07].

Primitive	Strategy	Pros	Cons
Image Values	Harr-like	Efficient computing; Designed to capture certain features (edges, lines, etc)	Window-shape based
Gradient	SIFT	High discriminatively power; Invariant to affine transformations and global intensity changes	Sensible to local illumination changes
Gradient	HOG	Simple to compute	Not affine invariance

**Table 2.6**

APPEARANCE BASED SHAPE DESCRIPTION

### 2.3.4 Spatial layout

Techniques to access data by spatial locations have their origins on the geographical information systems [CCK84] [RFS88]. Further, similar techniques have been adopted by retrieval in general image collections. Spatial information can be effective in combination with other features to distinguish between object composed by parts with similar properties. For example, regions of the sea and the blue sky might exhibit very similar color histograms, but their spatial locations in an image are different. The spatial features include spatial position and spatial relations between object parts.

Spatial position of the elements of the image use to be addressed by a vector of the coordinates positions (usually extracted by the centroid) or by the description of the spatial boundary of the element (usually approximated by the bounding box). Spatial position is referred to a global coordinate system defined by the image. Otherwise, the spatial relations are defined due to a local coordinate system that describes the relative the position of an element respect to another one. 2D string and its variants are the most widely used representation of spatial relationships.

The 2D string descriptor was initially proposed by Chang et al. [CSY87] with a reduced set of relationships. Further, Jungert [E.88] extended the spatial operators to those shown in the Figure 2.26. This descriptor encodes the relations between the objects enclosed by their minimum bounding boxes. This proposal has problems in encoding objects with overlapped bounding boxes.

$A < B$		$A   B$		$A \setminus B$		$A \cap B$	
$A = B$		$A \% B$		$A \setminus B$		$A \setminus B$	

Figure 2.26: Jungert's spatial operators

To overcome this problem Jungert and Chang [CSY88] extended the idea of the 2D strings to form 2D G-strings introducing several spatial operators and a cutting mechanism to segment the objects. Following the same concept Lee and Hsu [SF] proposed the 2D C-string that is accompanied with a variation on their cutting mechanism that reduces the complexity of the 2D G-strings. The 2D C-string offers an efficient representation of the region images and its codification is often used in image retrieval on databases. Figure 2.27 shows an example of image defined with this technique.

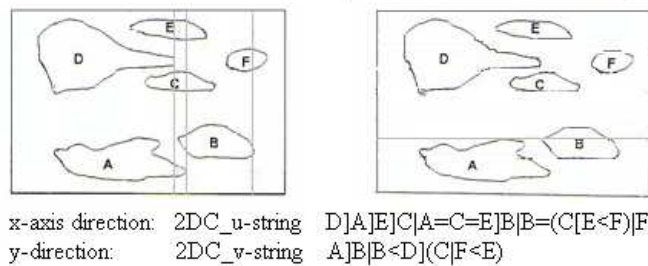
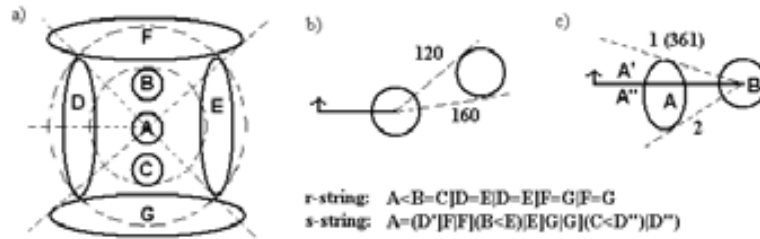


Figure 2.27: 2D C-string example

However, the content of all these string representations depend on the orientation of the picture elements as well as the cutting method. To overcome this problem, Petraglia et al. [PGT93] proposed the concept of 2D R-strings with a cutting mechanism similar to the one proposed by Lee and Hsu, such the cutting lines along the x- and y- direction for generating 2D C-strings are replaced by concentric circles in the ring direction and segments in the sector-direction, respectively, against some rotation center object. Still 2D R-strings have one disadvantage: the rotation center object is

missing from the string representation and then two similar pictures can be misjudged as dissimilar and vice versa. Thus, Huang and Jean [HJ96] creates another type of string, the 2D RS-strings, in which the center rotation object is always considered being codified as the first element in the S-string representation. Figure 2.28 shows an example of the RS-strings image description.



**Figure 2.28:** a) The cutting and the corresponding RS-string with A as the rotation center object. b) The initial position of the rotating half line. The dotted lines show the begin and the end bounds in the sector-direction. c) Object A is cut into A' and A'' by the rotating half line at the position with sector coordinate = 0.

The main problem of the spatial descriptors is that they need a reliable segmentation of the objects that describe. Depending on the complexity of the image content, this fact is still a challenging problem in computer vision.

Strategy	Pros	Cons
Centroid	Reference of object location	Relative to a global coordinate system
Bonding box	Reference of object location and spatial extension	Relative to a global coordinate system; Coarse and not affine invariant
2DString	Encode relative relationships	Not allow encode overlapping bounding boxes; Not affine invariant
2D C-String	Encode relative relationships; Cutting mechanism to encode overlapping bounding boxes	Not affine invariant
2D RS-String	Encode relative relationships; Rotation invariant	Dependence on a the reference element to encode rotation

**Table 2.7**  
SPATIAL LAYOUT OVERVIEW

## 2.4 Feature Indexation

Dealing with large databases of images imply managing large amounts of descriptors. These descriptors can be understood as feature vectors that can take values in high dimensional spaces. Many research efforts have been invested in the development of suitable indexing mechanisms. In this section we provide a coarse overview of the feature indexing structures. A full deep study of indexation methods can be found in the surveys of Gaede [GG98], Jesse [Yin03] and Kurniawati [KJS97]. We have summarized the taxonomy of the indexing methods in the Figure 2.29.

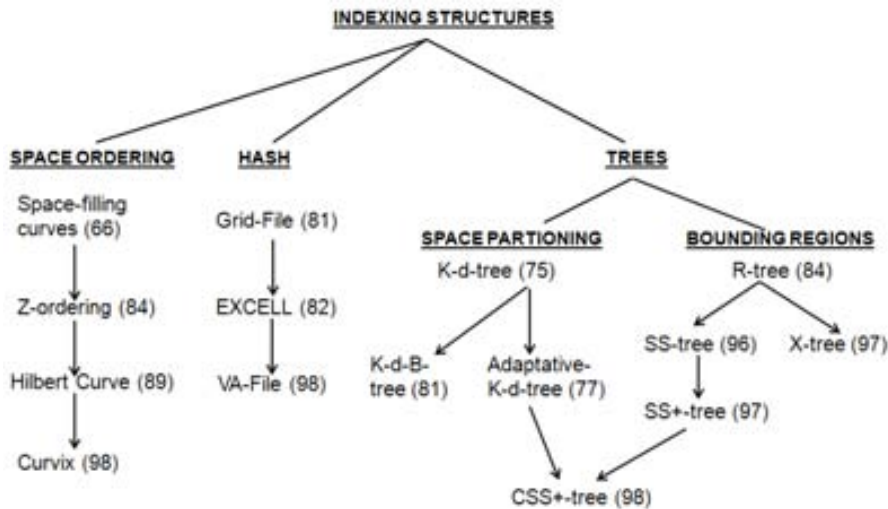


Figure 2.29: Taxonomy of the feature indexing methods and their reference year.

Based on how the indexing methods cope with multidimensional spaces, the access methods can be classified into three general categories: space-ordering, hash-based and tree-based [Yin03].

### 2.4.1 Space-ordering methods

These methods attempt to impose an ordering strategy to a multidimensional space in order to map it into one dimension. Thus, the vectors are projected into points that can be easily organized by one dimensional access methods such as the B-tree.

Space ordering methods include strategies such as the space-filling curves. Informally speaking, a space-filling curve is a continuous path which visits every point in a  $k$ -dimensional grid exactly once and never crosses itself. The space-filling curves provide a way to order linearly the points of a grid. The goal is to preserve the distance, that is, points which are close in space and represent similar data should be



stored close together in the linear order. The space-filling curves are a special case of fractals [FR89]. Widely known examples are z-ordering, the Peano, the Gray and the Hilbert curve. As an illustrative example, the Figure 2.30 shows a Hilbert curve in two and three dimensions.



**Figure 2.30:** Hilbert curve in 2 and 3 dimensions

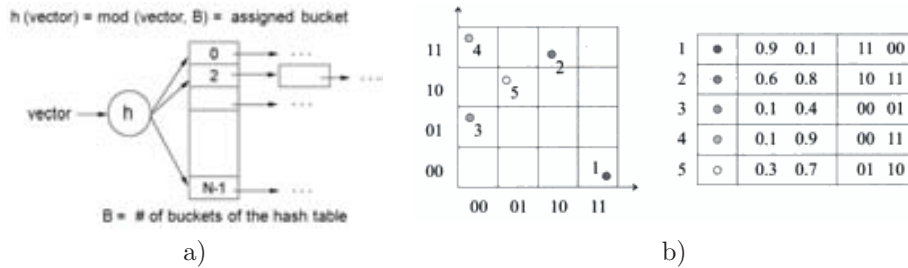
A property of any space filling curve is that neighboring points along the curve are also close in the original space but not viceversa. The result is that certain neighboring pairs in the original space are mapped far apart along the curve. To try to solve this drawback, some systems as Curvix [SZM99] use more than one curve to index the data.

### 2.4.2 Hash-based methods

The indexing techniques based in hash methods divide the space in buckets where the vectors are embedded. The result of the hash function is called signature and is used to map de data in its corresponding buckets in a hashing table. A very simple example of hash function could be the modulo operation applied to a vector regarding the dimension of the hash table (see Figure2.31). Ideally, hashing spreads the database uniformly across the range of the low-dimensional space, so that the metric properties of the hashed space differ significantly from those of the original feature space. The situation in which different input vectors hash in the same bucket of the indexing structure, is called collision and the colliding vectors are called synonyms.

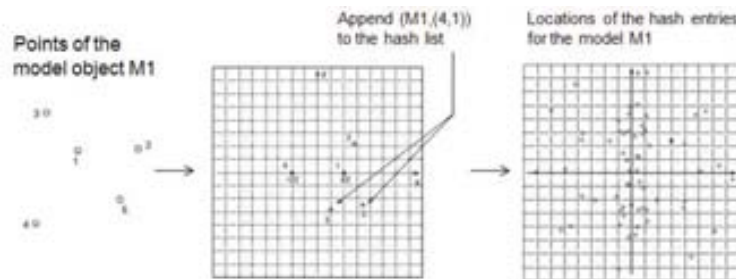
There are two types of hashing: the static hashing, where the input vectors is fixed and given in advance, and the dynamic hashing, where the set of vectors can change dynamically. The static hashing can benefit of previous knowledge of the data to avoid collisions. Nevertheless hash methods suffer from difficulties in designing a good hashing function as the dimensionality of the original space grows [Ull72], [Knu97], [Kno75].

Hash structures can have more than one dimension. By example, the vector approximation file (VA-File) divides the data space into  $2^b$  rectangular cells where  $b$  denotes a user specified number of bits (e.g. some number of bits per dimension). Instead of hierarchically organizing like the tree structures, the VA-File allocates a unique bit-string of length  $b$  for each cell and approximates data points that fall into a cell by that bit-string [WSB98]. Other multidimensional hash structures are the Grid-File [NHS81] or the EXCELL method [Tam82].



**Figure 2.31:** a) Hash table indexation according to the module function. b) VA File Structure.

A special case of hash indexing is the so called geometric hashing [RW97]. This strategy do not index vectors of features but index the structures of the objects that are present in a scene. The geometric hashing assumes that an object is a rigid structure that can be represented as a discrete set of points. As we can see in the Figure 2.32, the object points are mapped to a two dimensional hash table. The process is done with all the possible pairs of the model. At each step two selected points that act as a reference to obtain an affine transformation of the model. The geometric hashing involves in a single step the indexation of the parts of an object and the checking of their spatial arrangement.



**Figure 2.32:** Process of model indexing with a geometric hashing technique.

### 2.4.3 Tree-based methods

Tree-based indexing structures can be classified into Space Partitioning (SP) or Bounding Region (BR) strategies [CM99a]. A SP-based index structure consists of space recursively partitioned into mutually disjoint subspaces. The hierarchy of partitions forms the tree structure. On the other hand, a BR-based index structure consists of bounding regions (BRs) arranged in a (spatial) containment hierarchy. At the data level, the nearby data items are clustered within BRs. At the higher levels, nearby BRs are recursively clustered within bigger BRs, thus forming a hierarchical directory structure. For this reason, we also refer to them as object clustering data structures.

The clusters may overlap with each other.

The k-d-tree and the R-tree are two of the most successful tree-structures belonging respectively to each of the two groups. They have originated an extent family of variants that we overview in this section.

### Space Partitioning Trees

The k-d-tree is a binary search tree that represents a recursive subdivision of the space into subspaces by means of  $(d-1)$  dimensional hyperplanes. The hyperplanes are iso-oriented and their direction alternates between the  $d$  possibilities. Each splitting hyperplane has to contain at least one data point, which is used for its representation in the tree. Interior nodes have one or two descendants each and function as a discriminator to guide the search.

The main disadvantage of the k-d-tree is that its construction is sensible to the ordering of the data and that the points are spread all over the tree. The adaptive k-d-tree mitigates these problems by choosing a split such that one finds about the same number of elements on both sides [Ben75]. The hyperplane subdivision proposed by the k-d-tree could produce elongated spaces. Thus, another variant, the k-d-B-tree solves the problem by allowing the hyperplane in any direction [Rob81]. Other approaches as the VAMSplit-k-d-tree [WJ96a] focuses on the building process of the structure to improve the efficiency of the memory usage.

### Bounding Region Trees

Another main variant of the tree indexing structures are the R-trees [Gut84]. The R-tree appears as an evolution of the one-dimensional B-trees to deal with indexing purposes in a multidimensional space [BM72]. The structure of a R-tree is characterized by defining a maximum and minimum number of entries in a node. Then, every node contains between this maximum and minimum amount of index entries unless it is the root. Moreover, all the leaves of the tree appear in the same level and the root node has at least two children unless it is a leaf. An important property of the R-tree is that each node is bounded by a spatially minimum bounding rectangle that contains all the objects (feature vectors or subnodes) within the node. This fact allows the structure not to deal only with points like in the k-d-tree but deal with spatial objects like lines, polygons or higher dimensional polyhedra.

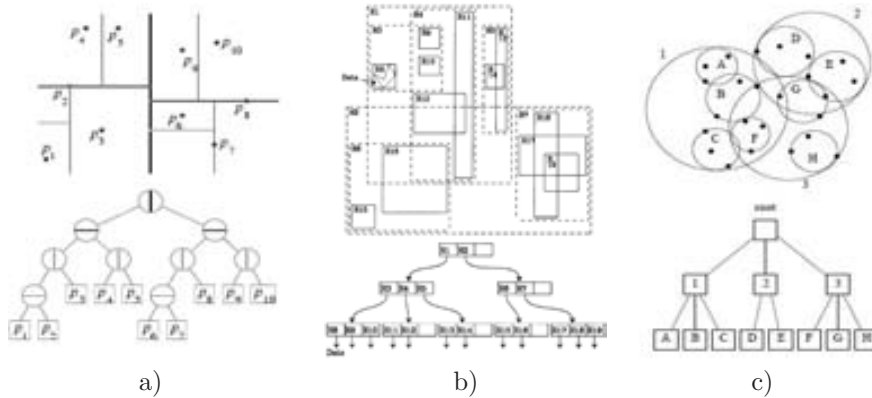
The main difference between the R-tree and its variants are in the parameters used in the tree building algorithm and the shape of the bounding envelopes of its nodes. The SS-tree and the X-tree use bounding spheres instead of bounding rectangles to make the distance query consistent with the bounding envelope. Nevertheless, the spatial overlapping of the nodes could be bigger than the previous approaches specially at top levels [WJ96b][BKK96].

Another approach is the M-tree structure that, instead of fixing the bounding regions to be boxes or spheres, use a distance function that can be provided according

to the user needs [CPZ97]. M-tree attempts to overcome the limitation of using a  $L_p$  metric, such as the Euclidean distance, to compute similarity between the multidimensional indexed vectors.

Finally, the latest evolution of the R-tree, the CSS+-tree, is a tree-based structure which uses a collection of unbalanced trees produced by a multi-resolution adaptive k-means clustering. This structure combines bounding envelopes of rectangles and spheres. Furthermore it share some characteristics with the space partitioning methods so it divides the space by Voronoi regions.

Tree-based indexing structures have been used in several commercial systems: e.g. the QBIC system [FSN<sup>+</sup>95] uses a R\*-tree indexation, the NETRA system [MM99] applies a SS-tree, the PicToSeek engine [GS99] indexes by a SR-trees or the ImageRover [STL97] and ImageScape [Lew00] use a k-d-tree.



**Figure 2.33:** Examples of tree structures a) k-d-tree b) R-tree c) SS-tree

Structure	Pros	Cons
Space Curves	One dimension mapping	Similarity related to curve arrangement
Hash	Fast; Similarity customizable from Hash function	Collisions depending on a good hash function
Trees	Arrange ND-features and spatial elements	Cost maintenance on addition and deletion of elements; Search through structure; Most assume Euclidean distance organization

**Table 2.8**  
INDEXATION OVERVIEW

## 2.5 Part Matching

If we want to retrieve images that contain a certain object, we have to describe them as a collection of parts 2.2. These parts are usually described by a set of features that take values in high dimensional spaces. This way, a retrieval system needs some indexing structures to organize the image parts according to its color features, texture, shape, etc.

A query object is also modelled as a collection of parts encoded by their visual properties. Then, the retrieval system looks for those images that share similar parts. These images are potential candidates to contain the query object. Nevertheless, a further step can be applied if we want to check the global arrangement of the parts and verify the presence of the object. Next we resume several strategies to perform this process.

### 2.5.1 Alignment techniques

Some of the earliest approaches on checking the part arrangement of an object are inspired in the alignment techniques. The alignment strategies aim to find a particular type of transformation to map a query object against the database information [ZF03] [Ols92]. An object is seen as a rigid structure that has to be put in correspondence with the image content. Some alignment approaches use pairwise correspondences to compute the transformation and then evaluate the goodness of the matching between the object and the image. This matching use to be computed from the shape information using a given kind of distance (e.g. Hausdorff, Chamfer, Cross Corretation)[HU90].

One of the main drawbacks of the alignment techniques is their computational dependence on the complexity of the scene. This way, approaches like the RANSAC algorithm are characterized for the ability of dealing with large amount of data and the presence of many outliers [FB80]. The RANSAC algorithm consists of an iterative procedure to estimate the parameters of a transformation using a subset of data items. This procedure has been successfully applied in specific problems like stereo matching, building panoramas, or robot navigation.

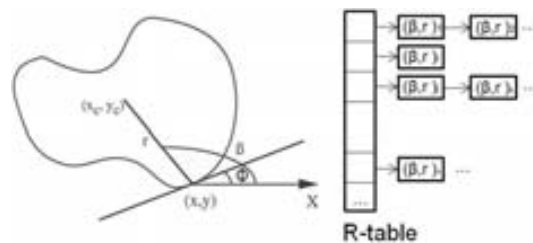
### 2.5.2 Generalized Hough Transform

One of the most known systems to check the spatial correspondence of the object parts is the Generalized Hough Transform (GHT). This system defines the geometry of an object in relation to an external reference point [Bal81]. This technique was originally developed to find arbitrary curves in a given image according to the features of the contour pixels. Later, the same idea was applied in a general framework of object detection replacing the features of the curves by the features of the object parts.

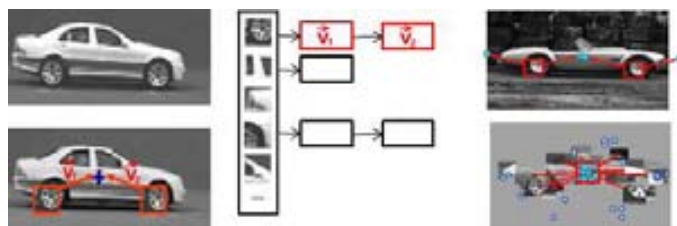
The GHT consists in two main phases. In a first step the object is modelled by a look-up table and then, in a second step, the process of localization is applied. A general idea of the computational process can be graphically exemplified with the

images of the Figures 2.34 2.35.

In the problem of localization of curves, the GHT builds a look-up table known as R-table that contains a set of the features regarding to the pixels of the shape. To model the shape a reference point such as the gravity center is chosen. Then, the position of the reference point is specified respect from each each point on the contour. This way, the used features are the orientation,  $\Phi$ , of the tangential line at the point and the length,  $r$ , and the orientation,  $\beta$ , of the radial vector joining the reference point and the feature point. Using  $\Phi$  as an index to the R-table the tuple storing  $r$  and  $\beta$  is placed at the indexed position. Having done this for each feature point, the R-table will fully represent the template object. Also, since the generation phase is invertible we may use it to localize occurrences of the object elsewhere. In the second phase, we locate the modelled shape inside an unknown image. Each edge point the target image is segmented and its orientation  $\Phi$  is calculated. Again, using  $\omega$  as an index to the R-table, we use the stored  $(r, \beta)$  pairs to vote at this new location. The  $(r, \beta)$  values allow to reconstruct the reference point of the shape inside this new image. Although some false reference points may be calculated, given that the object exists in the image, a maximum will occur at the correct reference location.



**Figure 2.34:** Generalized Hough Transform for curve matching. An R-table stores the geometric features of the contour points respect to a reference point  $(x_c, y_c)$ .



**Figure 2.35:** Matching of a car with a Generalized Hough Transform strategy. The look-up table stores the parts that define the car and their position according to a reference point. A high density of votes denote the presence of the car in a scene.

Some approaches benefits from previous knowledge of the object structure to adapt the Hough voting strategy to a specific configuration of parts. This is the case of the

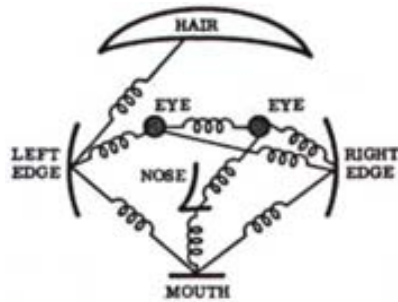
work of Bouchard [BT05] where an object is represented as an hierarchical structure. The method uses a GHT strategy to detect the object in a bottom-up manner. It constructs a voting pyramid according to possible locations and scales of the object parts. For every layer in the hierarchical structure, it combines the votes for the parts locations to their parent's locations. Maxima in the voting pyramid for the top-level part give potential object placements. In the Figure 2.36 we show a three-layer structure representing the image of a horse.



**Figure 2.36:** The image of a horse is modelled by the average position of the subparts according to different points of view. A generalized Hough voting strategy accumulates the evidence of the object detection in a hierarchical structure of three levels (reprinted from [BT05]).

### 2.5.3 Spring-like Connected Configuration

Sparse flexible matching approaches take their origins in the pioneer works of Fischler and Elschlager [FE73]. It considers the problem of matching an object to an image by modelling it as a collection of parts arranged in a deformable configuration. The deformable configuration is represented by spring-like connections between the parts (see Figure 2.37).



**Figure 2.37:** Figure from the work of Fischler and Elschlager [FE73]

The matching problem is that of finding the best placement of the parts in an image, where the quality of a placement depends both on how well each part matches the image and on how well the placements agree with the deformable configuration.

Sparse flexible approaches are used to be applied in the modelling of an object category. The appearance of the object parts and their locations are used to be learned from examples. A training process is used to define the flexibility in the position of the object parts and the constraints of their appearance.

The solution to the matching involves minimizing a certain energy function that takes into account the variability allowed in the configuration. Since the minimization problem can provide a high computational cost, there is a tradeoff between the complexity of the matching process and the richness of an object representation. Thus, an object can be modelled by different structures according to the number of parts in which is represented and the constraints between their connections.

Some works use tree structures to define the relations between the parts. This is the case of the work of Felzenszwalb that modelled body postures and human faces [FH05]. For faces, the parts are features such as the eyes, nose and mouth, and the spring-like connections allow for variation in the relative locations of these features. For people, the parts are the limbs, torso and head, and the spring-like connections allow for articulation at the joints. The matching process is done by a tree search strategy that explores the matching solutions between the object and the scene.

Tree structures are restricted to specific object geometries since do not allow the representation of cyclic connections between parts. This way, other approaches use graph structures to represent more general geometries.

In the constellation model, the parts of an object are fully connected, so every part achieves a spatial relation with all the other parts of the object [FFFP03][FPZ03]. The deformation of the object is modelled by a Gaussian distribution. Its main challenge is that the number of parameters grows exponentially with the number of parts. The recognition process is done by an exhaustive search and it can only suit representations made of a small number of parts (e.g., less than 10 parts).

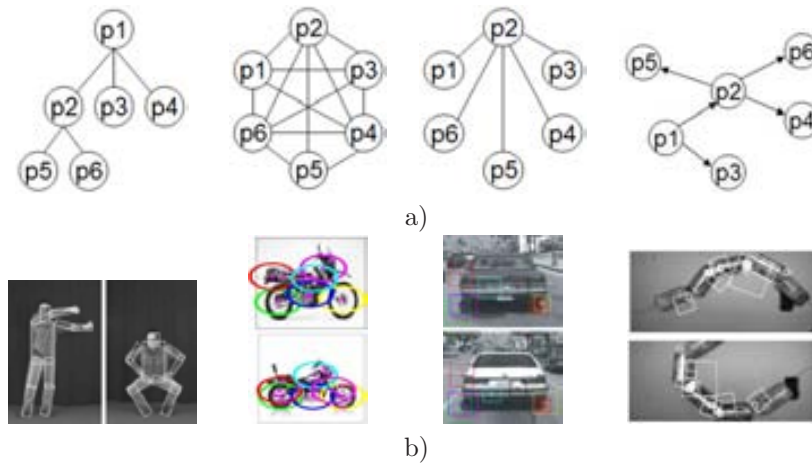
In the opposite direction, the star-shaped configuration defines a unique node to be connected with all the other nodes of the structure. This geometry defines an special node which serves as a starting point for the formation of hypotheses on the geometry of the other parts. In contrast to the constellation approach, the star-shaped model scarifies the number or relation representations and it uses to deal up to 20 or 30 parts [FPZ05][SI07].

Both configurations, star-shaped and constellation, can be understood as particular cases of the  $k$ -fan graph [CFH05].  $K$  is the number of "special" nodes in the graph that are connected to all the rest of the nodes in the structure. When  $k = 1$  the structure is that of a star graph and when  $k = n - 1$  (where  $n$  is the number of parts in the model) the structure is a constellation one.

The most flexible model is the graph structure proposed by Carnerio [CL06]. In this model the geometry of each node is influenced by its  $k$  closest neighbors and mod-



els may contain hundreds of features. The basic matching process consists of finding an initial correspondence set, and iteratively searching for additional correspondences assuming that the previous matches are correct.



**Figure 2.38:** a) Graphical geometric models. b) Examples of objects modelled for the above type of structures (reprinted from [FH05] [FPZ03] [FPZ05] [CL06]).

### 2.5.4 Bag of words

The bag of words treats an object as a collection of regions describing only their appearance and ignoring their spatial structure. The system identifies a set of basic regions that define the content of an image. These regions are the so-called words and the complete set is called dictionary. Figure 2.39 shows this concept graphically. Hence, the presence of an object in the scene is detected by the presence of the words that define the object. An object is represented by an histogram of words and two objects are considered similar if they have close histogram descriptions.



**Figure 2.39:** Bag of words strategy defines a dictionary of image parts and represents an object as a collection of them. (reprinted from [Woj09])

The bag of features is a simple strategy that can obtain very good results in the presence of textured objects that contain a huge number of parts. It has been

successfully applied in some works of object categorization that identify the words according to a set of predefined classes [CDF<sup>+</sup>04][Ff05].

Technique	Pros	Cons
Alignment	Intuitive; Evaluation of the overall object on the scene; Mainly applied to shape matching	Dependence on the image complexity; Only rigid structures
Hough Voting	Global evaluation of the object; Voting scheme adaptable to any feature type	Finding clusters of votes; Only rigid structures
Spring-like Connected Configuration	Adaptable to object part configuration	Time consuming
Bag of Words	Fast matching; Suitable for model categories; Allow flexible structures	No object localization; Rely on statistics; Need high amount of object parts

**Table 2.9**

ARRANGEMENT VALIDATION OVERVIEW

## 2.6 Similarity measures

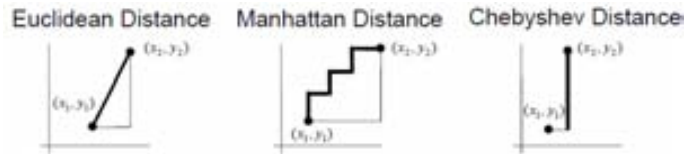
We have seen that an object part can be described with different types of features according to its characteristics. This features regarding to color, shape, etc. can be represented as a point in a N-Dimensional Space. Given a query, the results of the content-based-retrieval system have to be ranked according to a measure of similarity. Many distance measures can be applied to evaluate the similarity of two images according to their features. The choice for a particular measure can affect significantly the retrieval performance depending on their characteristics and the particular needs of the retrieval application. Next we expose some of the measures most commonly used in CBIR. A more extended overview of the similarity distances can be found at [DJLW08].

### 2.6.1 Minkowski-Form distance

The Minkowski-Form distance (MFD) is the most widely used metric for image retrieval . Given two feature vectors  $f_1$  and  $f_2$  of N bins, this measure is defined as follows:

$$D(f_1, f_2) = \left( \sum_1^N |f_1(i) - f_2(i)|^p \right)^{1/p}$$

In this measure each dimension of image feature vector is independent of each other and is of equal importance. Depending on the value of the parameter we talk about three types of distances (see Figure 2.40). When  $p = 1$ , the Minkowski-Form corresponds to the Manhattan Distance (or city-block) ( $L_1$ ), when  $p = 2$  we talk about the Euclidean Distance ( $L_2$ ), and when  $p = \infty$  it is called is Chebyshev Distance ( $L_\infty$ ).



**Figure 2.40:** Example on the Minkowski-Form Distances

Several commercial systems of CBIR make use of the different types of the Minkowski Distance. For instance, MARS system [RHM97] used Euclidean distance to compute the similarity between texture features; Netra [MM99] used Euclidean distance for color and shape feature, and  $L_1$  distance for texture feature; Blobworld [CTB<sup>+</sup>99] used Euclidean distance for texture and shape feature. In addition, Voorhees and Poggio [VP88] used  $L_\infty$  distance to compute the similarity between texture images.

We have to notice that the Euclidean distance takes special relevance in the retrieval application since most of the feature indexing structures assume that the Euclidean distance is the base for feature comparison (see Section 2.4).

### 2.6.2 Weighted Euclidean distance

When different image properties are indexed separately, similarity or matching scores may be obtained for each property. Then the overall similarity criterion may be obtained by linearly combining individual scores. The weights for this linear combination may be user-specified. Thus the user can put more emphasis on one particular visual property

### 2.6.3 The histogram intersection

The histogram intersection is a measure that quantifies the difference between histograms. It was first described by Swain and Ballard to index images by color [SB91]. Histogram intersection has been used in many early CBIR, including the retrieval system proposed by Jain and Vailaya [JV96] or the MARS engine [RHM97]. Its formulation is defined as a special case of the  $L_1$  distance:

$$D(f_1, f_2) = \frac{\sum_1^N \min(f_1(i), f_2(i))}{\sum_1^N f_1(i)}$$

### 2.6.4 Quadratic Form (QF) Distance

The Minkowski distance treats all bins of the feature histogram entirely independently and does not account for the fact that certain pairs of bins correspond to features which are perceptually more similar than other pairs. To solve this problem, quadratic form distance is introduced:

$$D(f_1, f_2) = \sqrt{(f_1 - f_2)^T A (f_1 - f_2)}$$

where  $A = [a(i, j)]$  is a similarity matrix, and  $a(i, j)$  denotes the similarity between bin  $i$  and  $j$ . Quadratic form distance has been used in many retrieval systems such as QBIC [FSN<sup>+</sup>95] for color histogram-based image retrieval. The similarity matrix allows to adapt the feature distances and penalize some mismatches more than the others (by example in the case of the color histogram can achieve more distance between the red bin and the blue one than from the red bin the orange one).

It has been shown that quadratic form distance can lead to perceptually more desirable results than Euclidean distance and histogram intersection method as it considers the cross similarity between colors.

### 2.6.5 Mahalanobis Distance

The Mahalanobis distance metric is appropriate when each dimension of image feature vector is dependent of each other and is of different importance. It takes the sample distribution into account when calculating the distance of a sample from the mean point. It weights the distance based on the variability in the direction of the sample point. Mahalanobis distance is a distance measure based on correlations between the variables, and is a measure of similarity in a multidimensional feature space between a group/cluster of samples and centroid of a cluster. It is a method of measuring how similar sets of values are (e.g. image features in CBIR). Based on a set of known training samples the unknown samples can be classified as members of the class or not. Being  $C$  the  $N \times N$  covariance matrix for the data set, Mahalanobis distance can be formulated as:

$$D(f_1, f_2) = |det C|^{1/N} (f_1 - f_2)^T C^{-1} (f_1 - f_2)$$

### 2.6.6 Hausdorff distance

Hausdorff distance is used to evaluate the distribution between two sets of feature vectors. This is a classical correspondence-based shape matching method that has often been used to locate objects in an image regarding their spatial features. Given two sets of features  $F_1 = f_1^1 \dots f_1^p$  and  $F_2 = f_2^1 \dots f_2^p$  the Hausdorff distance is defined as:

$$D(F_1, F_2) = \max(d(F_1, F_2), d(F_2, F_1))$$

where

$$d(F_1, F_2) = \max_{f_1 \in F_1} (\min_{f_2 \in F_2} \|f_1 - f_2\|)$$

and  $\|\cdot\|$  is the underlying norm on the features of  $F_1$  and  $F_2$ , usually Euclidean distance. The main drawback of the Hausdorff distance is that this measure is too sensitive to noise or outlier.

### 2.6.7 Chamfer distance

Chamfer distance computes the best fit of a set of features from two different images by minimizing a generalized distance between them [GIK03] [BW77]. Chamfer distance have been mainly used for shape matching dealing with contour points as image features. The query image and the scene image are represented by the edge information. From the scene, the distance image is computed. Distance image gives the distance to the nearest edge at every pixel in the image. The Chamfer measure is obtained by translating the query edge model over the distance image scene. The measure minimizes the average of the distance values that lie under the data pixels of the superimposed query (see Figure 2.41). The main advantages of the Chamfer distance are that it is robust to noise and it is computationally cheap.

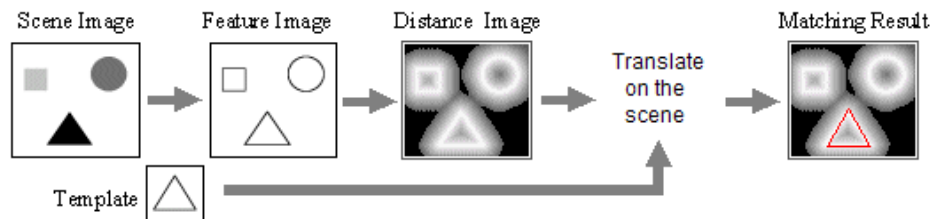


Figure 2.41: Chamfer distance for shape matching

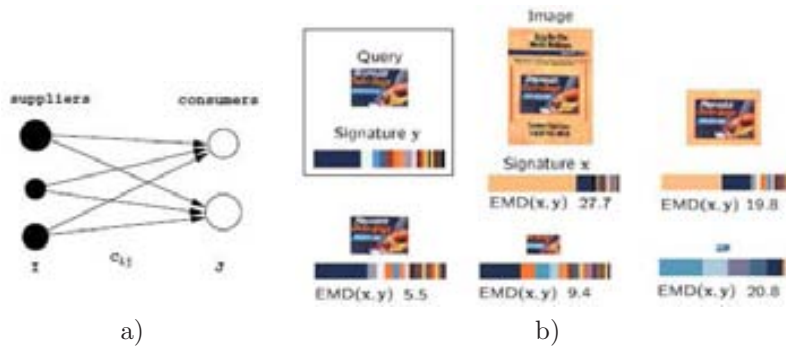
### 2.6.8 Earth Movers distance

The Earth Mover's Distance (EMD) is a method for calculating similarity between multidimensional distributions in a feature space. EMD is informally explained as the minimal amount of work that must be performed to transform one distribution into the other by moving 'distribution mass' around [RTG00]. This case of the transportation problem is a particular instance of the Mallows distance and can be solved by linear optimization algorithms.

In the context of image retrieval, EMD is a flexible method for calculating similarity between two image signatures. An image signature is defined as a multidimensional distribution of their features with an associated weight. Given two signatures that characterize two images, Earth Movers Distance (EMD) is a similarity measure that can be understood as the amount of 'work' needed to transform one signature into another.

One of the main advantages of the EMD is that the compared signatures can have different lengths, by example, two histograms can be compared having different

number of bins. EMD have been applied wide variety of applications such as works of logo retrieval from the color information [CG99] or works of object matching from the contour information [GD04].



**Figure 2.42:** a) An example of a transportation problem with three suppliers and two consumers (reprinted from [RTG00]) b) EMD between the pattern color signature and the signatures for the various portions of the image (reprinted from [CG99])

## 2.7 Metrics for Performance Evaluation

The performance of the retrieval systems are mainly evaluated with two measurements: precision and recall. These two concepts can be defined as follow:

$$\text{precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of retrieved items}}$$

$$\text{recall} = \frac{\text{number of relevant items retrieved}}{\text{total number of relevant items available}}$$

A perfect Precision score of 1.0 means that every result retrieved by a search was relevant but does not say nothing about whether all relevant images were retrieved. Moreover a perfect Recall score of 1.0 means that all relevant images were retrieved by the search but says nothing about how many irrelevant images were also retrieved.

Notice that none of these two values give information about the total amount of items stored in the database. Then, a third measure named fallout can be defined as:

$$\text{fallout} = \frac{\text{number of non relevant items retrieved}}{\text{total number of non relevant items available}}$$

The recall and fallout values can be also named True Positive Rate (TPR) and False Positive Rate (FPR). A retrieval system has to make a tradeoff between precision and recall to decide which is the amount of relevant images presented to the

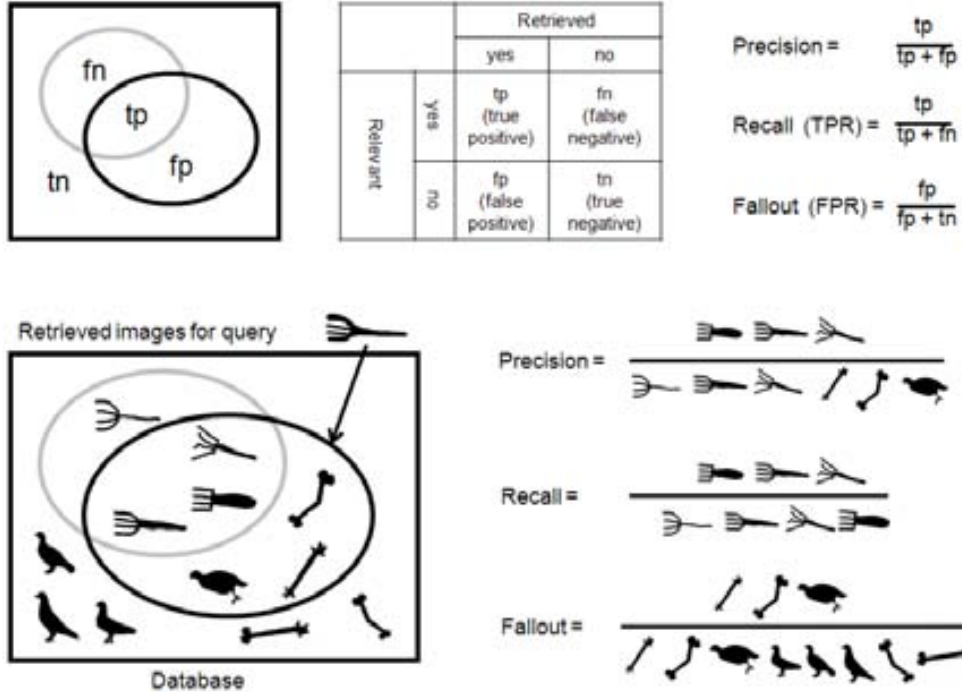


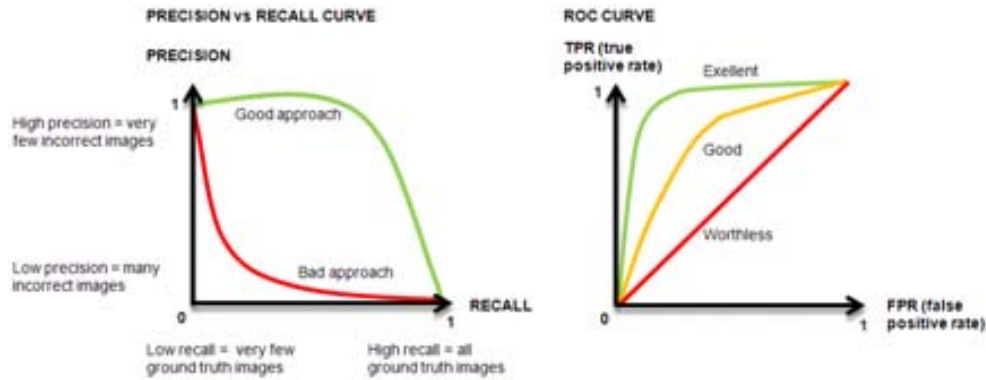
Figure 2.43: Illustration of the precision, recall and fallout concepts

user. This decision is often dependent on the field of knowledge in which the retrieval system is involved. When the omission of relevant results can be critical, by example, in an image medical application, a high recall is preferred than a high precision. Figure 2.43 shows in a very simple and graphical way these mathematic measures.

The most common graphics used to illustrate the performance of a retrieval system are the Precision-Recall Graph (PRG) and the Relative Operating Characteristic Curve (ROC) [DG06][Faw06]. Given a query to the system, the images of the database are given a probability value of being retrieved. Exploring several thresholds on this probability we can construct both graphical curves by plotting the pairs of precision-recall values and the pairs of true and false positive rates values. In a precision-recall graphic, the higher the curve, the better the retrieval performance since for the same recall value, a higher curve signifies a higher precision value.

ROC curves also provide a very intuitive metric of the system evaluation: the area under curve (AUC). An AUC of 1 represents a perfect performance and an AUC value of 0.5 represents a worthless performance. A rough guide for the evaluation is the traditional academic point system: [0.90,1]=excellent (A), [0.8,0.9)=good (B), [0.7,0.8)=fair (C), [0.6,0.7)=poor (D) and [0.5,0.6)=fail (F).

The general performance of a system can be represented by averaging the resulting graphs for multiple queries. Figure 2.44 resumes the properties of PRG graph and the ROC curve.



**Figure 2.44:** Comparison of curves of the Precision vs Recall and ROC graphs

In this thesis we use the PRG graphs and the ROC curves since they are the most common performance evaluation measures in the literature. Nevertheless, there exist other criteria such as the Bull's Eye Performance (BEP) or the Average Normalized Modified Retrieval Rank (ANMRR). BEP measure computes the percentage correct retrieved images, being  $N$  the number of relevant items for a query and  $2N$  the number of retrieved images [ZL02]. Finally, ANMRR combines the number of ground truth images for a given query and the rank position of the retrieved results [Gro99]. The interesting point of the ANMRR measure is that it returns a single value in the range  $[0, 1]$  that allows a straightforward comparison between system performances.

In this chapter we have seen an overview of the general architecture of a CBIR system and we have developed the state of the art related to every module. We have exposed the pros and cons of the methods of part extraction, the feature descriptors, the indexing strategies and the matching techniques. Finally, we have also shown some metrics related to the internal processes of the retrieval system and some metrics to evaluate its performance. In the following chapters we present a set of retrieval systems that make use of some of the methods that we have reviewed. We will see how the query paradigms of every retrieval system constrain the specific processes of its modules.



# Chapter 3

## Query-by-Image Selection

### 3.1 Introduction

The formulation of the query has a direct influence in the final performance of a CBIR engine. Many retrieval systems allow the user to select an image as a representation of the query. Then, this image is used as a reference to retrieve the most similar images from the database. In the literature, the problem of selecting an initial query is referred with the name of *page zero problem* [ZB06]. We can find several options to perform the query-by-selection process. Depending on the source from which the user selects the input image, we can distinguish between two query modalities: query-by-external selection or query-by-internal selection.

Query-by-external selection [GS99] [SI07] is generally perceived as the simplest approach to query formulation. In one hand, it lets the user submit his own query but, in the other hand, it assumes that the user has a representative image of what he is looking for.

Otherwise, query-by-internal selection offers the option to browse the system database and select one of the stored images. There are three main methods of browsing images according to their visual content: unstructured, semi-structured and structured. In the unstructured browsing, the user can scroll through a complete view of all the database images. Generally, they are shown as thumbnails and they are ordered by an associated label such as the filename or a textual description tag. Nevertheless, when the volume of the database is too large, the user cannot search for a suitable query according to the visual appearance. Then, semi-structured browsing presents the user a random subset of the database content. Once an example image is selected, the system discards the less similar images and allows looping until the user is satisfied [STL97]. Finally, in the structured browsing, the user can move through a predefined hierarchy of image clusters that are denoted by semantic labels. This kind of database organization was commonly used in the first retrieval prototypes that were related to research purposes [VT00]. To illustrate the query-by-internal

selection, the Figure 3.1 presents some snapshots of the CIRES <sup>1</sup> system [IA02]. In this system the database images are classified according to a predefined hierarchy of semantic concepts. In the first level of classification it distinguishes among manmade images, natural images and textures. Then, these categories are further subdivided into other clusters. In order to formulate the query, the user can select an image by browsing through the categories of the database (see Figure 3.1 a)) or he can also choose an image from a set of random samples (see Figure 3.1 b)).

In addition to the source of the query, we can distinguish more options in the selection process. These options are related to the part of the query image that is used in the retrieval process. This way, given a selected image, the user have three options: deal with whole image scene, use a predefined segmentation or crop a subpart of the image. These strategies, ordered from less to more freedom in the user interaction, are illustrated in the Figure 3.2.

This way, most of the systems consider images as indivisible entities and do not allow specifying any spatial selection of the query. This is the simplest scenario of image retrieval since there is no need to distinguish between a target object and the background information. Then, the whole image content is used to compute the similarity against the database images. Two examples of these systems are the web engines MUFIN <sup>2</sup> (Multi-feature Indexing Network) and TinEye <sup>3</sup>. MUFIN is used to search for similar images from the web source [www.Flickr.com](http://www.Flickr.com). Moreover TinEye is a commercial product designed to find which are the internet pages that contain an instance of the query image.

Other CBIR systems perform an automatic part detection and let the user select which of the regions wants to be used as a query. This query modality is not very common but is used in some retrieval systems such as NETRA [MM99] or Blobworld [CTB<sup>+</sup>99].

Finally, the most flexible option consists in cropping a region of interest from the query image. Early approaches use to pre-compute the description of fixed subdivisions of database scenes [LN04], [DRD97], [MR97]. Recent works have obtained outstanding results combining the part extraction from local structures and the region description from shape appearance features. As an example, the proposal of Sivic called Video Google [SZ03] uses Lowe's SIFT descriptor to match objects in video images <sup>4</sup>.

Many retrieval applications can benefit from any of these options to select the content of the query image. Retrieval systems using the image selection paradigm are powerful tools to protect the intellectual property and find illegal copies of the query picture. These kind of systems are interesting to control the impressive growth

---

<sup>1</sup><http://amazon.ece.utexas.edu/~qasim/research.htm>

<sup>2</sup><http://mufin.fi.muni.cz/imgsearch/>

<sup>3</sup><http://tineye.com/>

<sup>4</sup><http://www.robots.ox.ac.uk/~vgg/research/vgoogle/>



**Figure 3.1:** Example of query-by-internal selection (CIRES system 3.1). a) The database images are categorized by semantic concepts. The user can browse this hierarchy and select an image. b) Otherwise, the query can be selected from random set of samples. c) Once the query is selected the user can adjust the weights related to the Color, Texture or Shape.



**Figure 3.2:** Modalities of part selection of the query image a) Whole Image b) Segmented region c) User's cropped selection

of images and videos available in web portals such as Google or YouTube. Nonetheless, there are cases in which the user is not interested in retrieve the exact image of the query. By instance, graphical designer could be interested in looking for images according to a specific characteristic. Then, some systems give an additional degree of freedom by complementing the input query with some options regarding to the features. The interface of the system allows the user to select which are the low level features (color, shape, texture, etc.) that he wants to prioritize in the retrieval process [STL97] [CTB<sup>+</sup>99] [IA02]. Then, given the query image, the user can adapt the retrieval options increasing o dismissing a weight related to each feature. By instance, if he is interested in retrieving images with similar texture despite of the color, he can tune the parameter related to the texture to its maximum weight and low the parameters refereing to other characteristics. To illustrate these options, Figure 3.1 c) shows a system interface where the user can adjust the weights related to the color features, the presence of texture and the shape.

In this chapter we present a CBIR system that uses the query-by-selection paradigm. In the next section we make a brief overview of the system and then we detail its concrete implementation. The whole explanation follows the general structure of a CBIR

system (see 2.1.2). This way, we explain the internal processes of each one of the modules of the system and, finally, we expose the experiments and the conclusions of our approach.

## 3.2 A CBIR system based in a multi-scale triangulation of the image content

We have developed a CBIR system where the query formulation is done selecting a sample image. The user is allowed select the query from both internal or external sources. In this system, the whole content of the query image is used to match the database scenes. Thus, we propose a descriptor to encode the appearance of a scene according to the layout information. We illustrate our proposal applying it to a collection of images extracted from the frames of a video. Then, we can retrieve specific frame shots without manually browsing the entire recorded material. In the following sections we describe the modules of the CBIR system that we have developed. A final resume of the whole process is graphically shown in the Figure 3.6. Next we proceed to detail the implementation of the description and the matching modules.

### 3.3 Description

In the literature we can find plenty of image descriptors to characterize the content of a scene. Gagaudakis [GR03] made a set of experiments to measure the performance of the image retrieval adding shape measures to the classical color histogram descriptors. He considered fourteen shape methods and test all their possible combinations, giving a total of over 16000 tests. The experiments identified the potential of measuring indirect shape using the Delaunay triangulation. Tests concluded that the methods using the triangulation were involved in the most successful combinations of image feature descriptors. Inspired in the results of Gagaudakis [GR03] we propose a CBIR system that uses the Delaunay triangulation to describe the content of the images. Next we describe the process of part detection and feature extraction. The grid of images of the Figure 3.6 illustrates the whole process of description.

#### 3.3.1 Part detection

We have based our image description in the layout of the parts that compose a scene. Our proposal is an evolution of the pioneer work of Tao [TG99] which codified the spatial arrangement of the image elements using a Delaunay triangulation. His work identified the corner points of an image as the basic parts of the scene. Then, he applied the triangulation on this set of points and analyzed the angular properties of the resulting mesh (see Figure 2.18). Tao's approach introduced a novel descriptor that was suitable for retrieval proposes since it was compact and indexable. Nevertheless, its dependence on the corners of the contours failed to be very sensitive on the noise and the image variations.

In order to obtain a major robustness of the image description, we have not used the corners as scene parts. Instead, we identify as the image parts a set of zones extracted from the boundary information. Moreover, we have applied a multi-scale analysis to avoid the noise sensibility and capture the relevance of the detected parts. The finality of this analysis is giving more weight to the main elements of the image than the details.

The process of part extraction is applied to every layer of a multi-scale representation of the scene. Thus, given an image, we first construct its multi-scale representation. Then, for every scale, we identify the image parts from the boundary information. Next, we detail how these two phases are computed. The first four columns of the Figure 3.6 exemplify the following process:

- Multi-scale representation:

The motivation to apply a multi-scale representation of the image comes from the space-scale work of Linderberg [Lin96]. Linderberg observed that objects in the world appear in different ways depending on the scale of observation. He exemplified the idea with the image of a branch of a tree. He states that if we observe a tree from a distance of few centimeters, we can identify the concept of the "leaves". Nevertheless, if we observe it at a larger distance, this concept has no sense but other concepts, such as the "tree" or the "forest", appear.

Besides this multi-scale properties of real-world objects, image retrieval systems need to cope with the complexity of unknown scenes and noise. This brings us to the conclusion that for a deep understanding of the image structure, multi-resolution image representation is necessary. Witkin [Wit87] and Koenderink [Koe84] introduced the idea of generating coarser resolution images by convolving the original image with the Gaussian kernel. Thus, the resulting structure is known as linear, or Gaussian scale-space. For a given image  $f(x, y)$ , its linear (Gaussian) scale-space representation is a family of derived signals  $L(x, y; t)$  defined by convolution of  $f(x, y)$  with the Gaussian kernel.

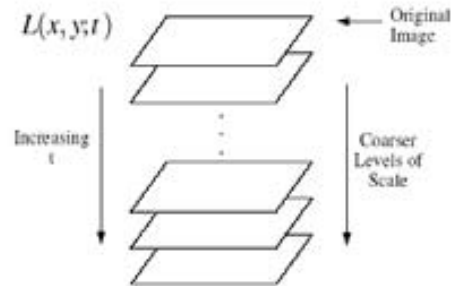
$$g(x, y; t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/2t} \quad (3.1)$$

such that

$$L(x, y; t) = g(x, y; t) * f(x, y) \quad (3.2)$$

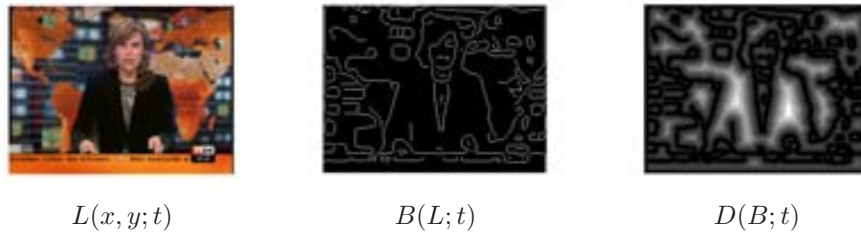
where  $t = \sigma^2$  is the variance of the Gaussian.

Such representation is composed by a stack of successive versions of the original data set at coarser scales. It is assumed that, the bigger the scale, the less information referred to local characteristics of the input data will appear. We use this representation to analyze the image structures from low resolution to general information.



**Figure 3.3:** Scale-space image stack

- Extraction of the image parts from the boundaries:  
 In every resolution level we identify the parts that compose the image content according to the contour information. Given an image  $L(x, y; t)$ , we apply the Canny operator to extract the boundary information  $B(L; t)$ . We use the edges of the image as a binary structure from which we apply a distance transform function. The result is a map  $D(B; t)$  that supplies each pixel of the image with the distance to the nearest edge pixel [RP68]. We understand the distance map as a topological surface where the valleys denote the limits of the image zones. The Figure 3.4 shows an example of this process.



**Figure 3.4:** Image part identification

We have seen that the part extraction process benefits from a procedure that does not require any intensive image segmentation. The image information can be easily extracted from the edges and it is not necessary that they form closed regions. Next, we explain which features are used to characterize the image according to the set of detected parts.

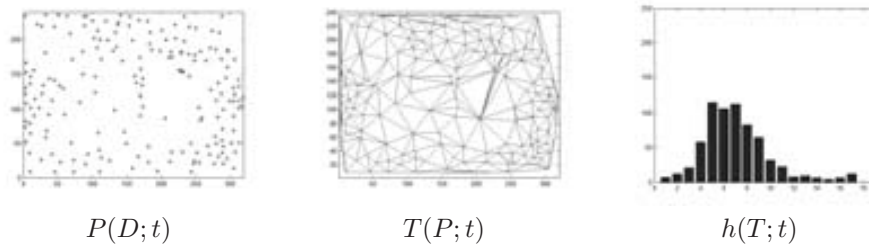
### 3.3.2 Feature extraction

Once we have identified the image parts in every resolution layer, we proceed to encode their spatial arrangement. The feature extraction is composed by three phases that are applied to each resolution layer. The process is illustrated in the last three columns of the Figure 3.6 and the Figure 3.7. Thus, in a first phase we characterize the image

parts as a collection of 2D points. Then, we obtain the triangulation information and, finally, we construct a histogram-based descriptor. We detail the implementation of these steps as follows:

- **Image part characterization:**  
We characterize every image part with 2D spatial coordinate. This coordinate is obtained as the maximum (the peak) of the ridges of the topological map  $P(D; t)$ .
- **Description of the layout of parts:**  
For every image layer we construct a Delaunay triangulation  $T(P; t)$  of the coordinate set  $P(D; t)$ . The Delaunay triangulation of a point set is a collection of edges satisfying an "empty circle" property: for each edge we can find a circle containing the edge's endpoints but not containing any other points. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation [Del34]. These diagrams and their dual (Voronoi diagrams and medial axes) have been deeply studied and have been used in many common methods for function interpolation and mesh generation.

Then, a histogram is obtained by the discrete representation of the angles produced by this triangulation. Hence, following a strategy similar to Tao's work [TG99], we construct the feature description by counting the number of times each discrete angle occurs in the image. Given the property that the three angles of a triangle sum 180 degrees, the histogram is built by counting the two largest angles of each individual triangle. Figure 3.5 shows an example of the histogram construction  $h(T; t)$ .



**Figure 3.5:** Layout encoding of a resolution level

- **Combination of the multilayer description:**  
At this point, the layout information of an image is formed by the set of histograms  $h(T; t)$  computed at each resolution level. Then, we combine all this information to construct the feature  $F$  that describes the image. With the combination of histograms we want to reach two main objectives: obtain a compact feature and accentuate the multi-scale representation of the image zones.

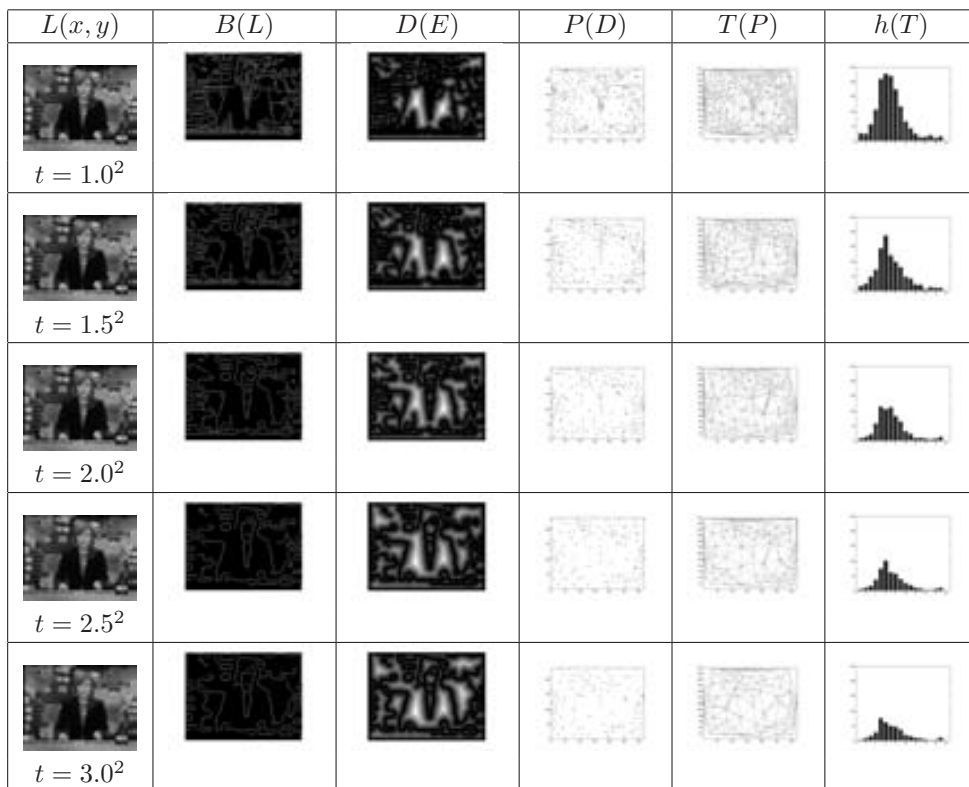
The steps we follow are the next: First we assemble the set of histograms  $h(T; t)$  as the rows of a matrix. Then we compute the vertical and horizontal projec-



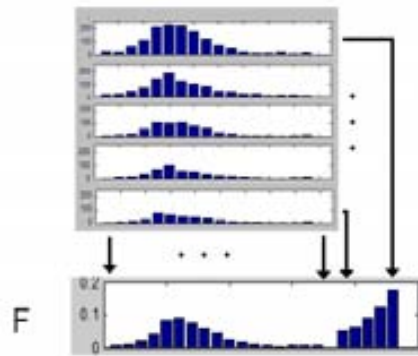
tions of this matrix and concatenate both projections in a single histogram. This single histogram can be understood as a  $n$ -dimensional vector that we finally normalize to have one unit length. This normalization is computed by dividing the value of each bin by the total norm of the vector.

An example of this process is shown in the Figure 3.7. Thus, the vertical projection enforces the layout of the dominant regions by adding repetitiveness of their spatial characteristics. Then the horizontal projection measures the amount of regions present in each resolution levels. This combination provides a considerably reduction of the information dimensions.

From the whole process of image description we characterize the content of an image with a single and normalized vector. Obtaining a compact descriptor is interesting for indexing applications and storage restrictions. Moreover, the normalization property allows to define a closed range of dissimilarity measures. This fact is useful to study the similarity measure of the images in retrieval applications.



**Figure 3.6:** Example of the descriptor construction



**Figure 3.7:** Computation steps of the image feature  $F$  from the histograms of every resolution level  $h(T; t)$ .

Once we have seen the process of description we proceed explaining the matching phase.

## 3.4 Matching

We use the same description process to characterize the database images as well as the query ones. Then, using the features  $F$  we can perform the matching process.

### 3.4.1 Feature indexing

We have not developed any specific indexing strategy in the design of our CBIR system. Since we only have one descriptor that defines the content of the images, we are in the simplest scenario for comparing the similarity of the query against the database scenes. Thus, any of the indexing strategies summarized in the section 2.4 could fit the requirements of the retrieval application. Instead, we have stored the description of the database images in a list and we have evaluated the matching of the query in a sequential order.

### 3.4.2 Evaluation

We have used the Euclidean distance as a measure to compute the similarity among two feature vectors. The distance is in the range of  $[0, 1]$  because we deal with normalized descriptors which norm is the unit. Then, the retrieval probability of a scene can be straightforwardly computed as one minus the Euclidean distance between the descriptors of this image and the query. In the next section 3.5 we present some experimentation on the proposed retrieval system.

Until this point we have seen the internal processes of the retrieval system. In the next section we exemplify them from a practical point of view.

## 3.5 Experiments and Results

We could apply our proposal to any retrieval system that deals with the whole content of an image. Next we show an example of its usage in the detection of certain images of interest along a video sequence.

### 3.5.1 Example in the retrieval of programs and commercials

A potential application of a CBIR by image selection is the detection of specific frames along a video. Some images of interest are those related to the program logos and the advertisements.

By instance, we could think in an application that helps the user in locating the information related to a certain TV program. A user could have a database of videos that he has recorded from a TV broadcast. Then, a CBIR application could show the logos of the programs and, by selecting one of these images, it could retrieve the starting point of the program where the query image appears. This way, the user can visualize the program without manually browsing the entire material of the database.

Moreover, the images of commercials can be useful for those companies that study the TV audiences. Hence, by selecting the image of a certain spot, the retrieval system could locate it along the sequence of images. The information about the moment of its broadcast could be related with other information of interest such as the audience rate data.

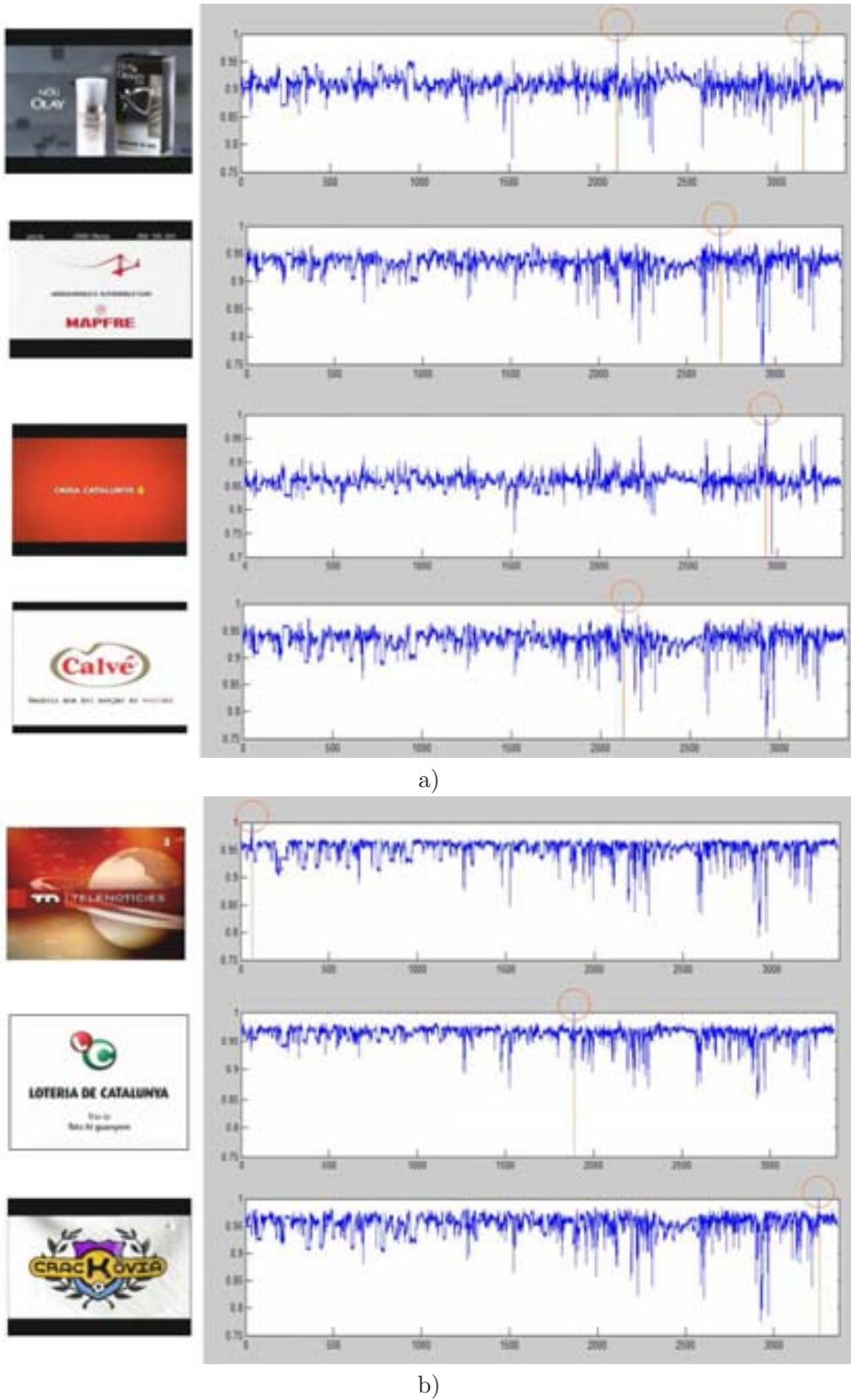
To illustrate this application we have captured a TV signal during one hour of broadcast. We have extracted the frames of the sequence with a frequency of one frame-per-second and we obtained a test set of 3600 images. Then in the Figure 3.8 we show some examples where we locate certain frames of programs and spots. The graphics show the probability of matching each selected image along the sequence of frames in chronological order.

Besides showing a concrete application for the retrieval system, we wanted to evaluate its performance.

### 3.5.2 Evaluation of the system performance

In order to analyze the performance of the system, we have made an experiment with the images of the video that we used in the example of the previous section. We wanted to study several items:

First of all, we wanted to analyze if our image representation supposes a reliable evolution according to the initial approach based on the corners [TG99]. Moreover, we wanted to evaluate the influence of the multi-resolution representation of the image content. This way, we have done a set of tests where we have compared our representation using multiple resolutions against the same proposal using a single resolution



**Figure 3.8:** Retrieval examples of advertisements (a) and program logos (b). The X axis of the graphs represent the chronological order of the video frames. The Y axis contain the probability of the query detection according to each image of the sequence.

level. Moreover we have also tested the Tao's approach based on the corners.

We have used several subsets of image sequences to form the ground truth of the experiment. Concretely we have used some images belonging to the broadcast of a news program. We have selected this kind of test images because the transitions between the news allow us to clearly identify groups of database images to be used as the test set. Moreover, these images contain different variations of their content which allow us to observe the robustness of the system. Next we will see which these variations are and how they affect to the retrieval performance.

The database contains 3600 images (the frames of the whole video) and we have used 260 of these images as the ground truth. The test set is distributed in 6 groups containing 123, 28, 10, 30, 58 and 11 images respectively. The Figure 3.9 a) shows an instance of each group. For each ground truth image we have applied the retrieval process against the whole database. We have represented the performance of the 6 groups of queries with the mean of the precision-recall graphs and the ROC curves. In the Figure 3.9 we can observe the results of our approach, against the results of the corner-based one and the results of our image description using a single resolution level.

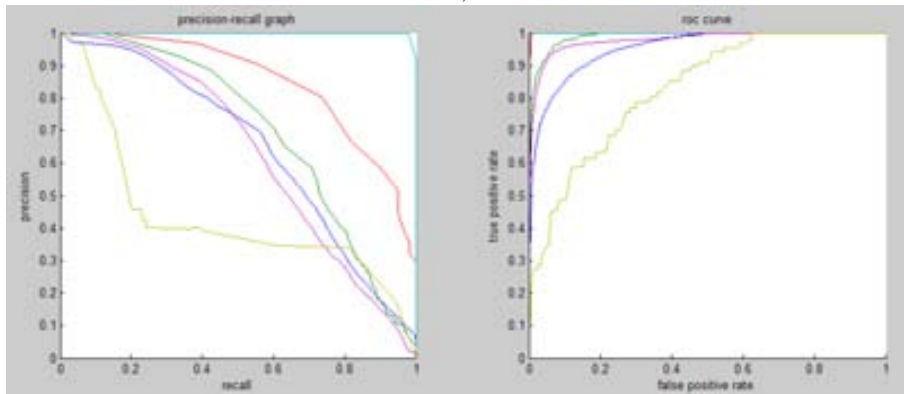
We have experimentally set to 5 the number of scale-space layers codified by our representation approach. Specifically, we have used a  $\sigma$  parameter varying from 1.0 to 3.0. In relation to the sampling of the angular values, we have used the same strategy as [TG99]. We have discretized the whole range of 180 degrees in buckets of 10 degree. Then, we have obtained a descriptor of 23 bins (18 bins for the vertical projection of the histograms, plus 5 bins for the horizontal projection related to every resolution layer). The boundary extraction for the results regarding to the corner approach and the single layer one use the  $\sigma$  parameter of 2.0. We have used the same value in order to allow their comparison.

According to the results we can observe that the multi-scale approach shows a better performance than the corner-based. The precision-recall graphs and the ROC curves present a higher response in all the queries. This way, we can see that a description based in the zones of the image at multiple resolutions (graphic b)) is more stable than a description that focus in the details of the boundaries (graphic c)). It is also interesting to compare the corner approach against our process using one level of resolution. Both experiments obtain similar results, being some of our zone approximation (graphic d)) slightly better than some based in the corners (graphic c)). If we compare the results of the graphics b) and d) we can observe the influence and the benefit of the multiple resolution analysis.

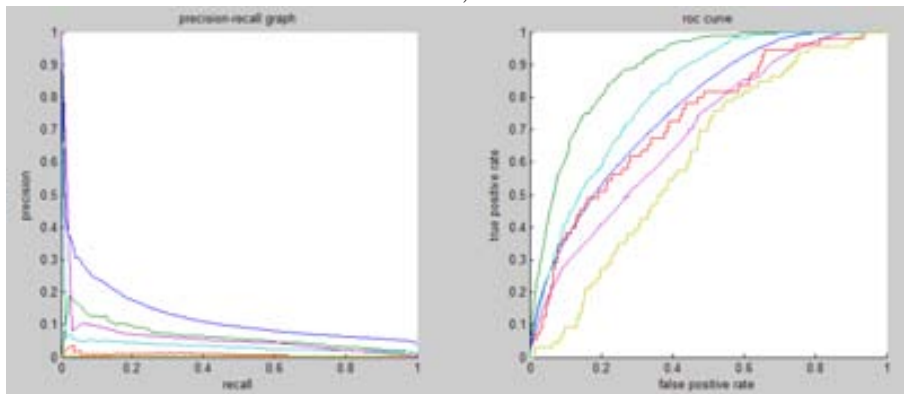
Even though our approach shows a better performance, it does not overcome completely the scene variation effects. Thus, we have analyzed more deeply the results in order to identify the main causes of failure. We illustrate these problems in the Figure 3.10 and we resume them as follows:



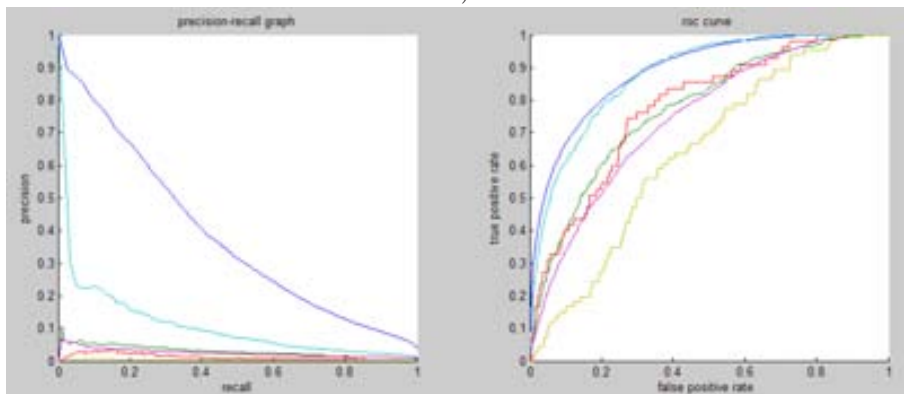
a)



b)



c)



d)

**Figure 3.9:** a) Samples of the queries for each ground truth sequence. b) Precision-recall graphs and ROC curves for our approach. c) For the corner based approach [TG99] d) For our approach with one level of resolution.

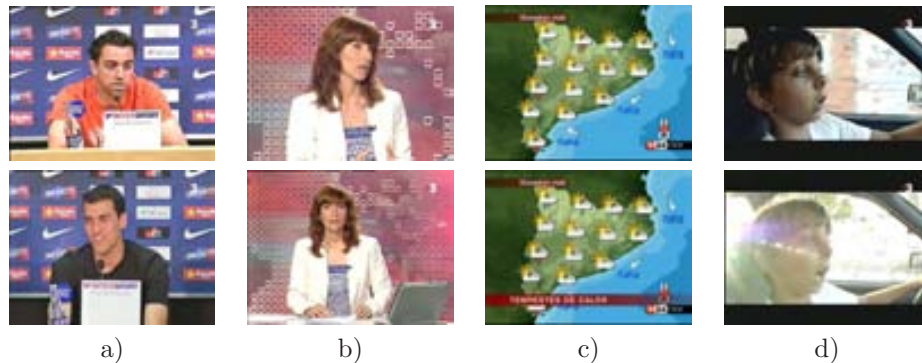
Confusion in images with similar structure We observe that a query can retrieve images that do not belong to the correct ground truth but that have similar structural properties. This is a natural consequence of the design of the system since we wanted to capture the overall similarity of the image and give less relevance to the details.

Changes in the point of view When the point of view of an image is changed, its contents change in the limits of the scene. The number of parts can change (growing when we zoom out and dismissing when we zoom in) and it affects to the final descriptor.

Occlusions We observe that the occlusions can break the regions of the image generating a higher amount of triangulation information. This effect can detriment the response of the system specially when the occlusion is produced in a meaningful part of the image, in a big zone. In this case, the occlusion is contemplated in all the scales of the multi-scale representation and the final description also captures it.

Changes in the illumination Finally, the illumination changes have direct influence in the construction of the descriptor. Thus, they constrain the edge detection and, consequently, the part extraction and its layout configuration.

Some of the test sequences present more than one of these problems. According to the results we observe that the changes of the illumination is the artifact that affects most to the performance of the system. Then, the occlusions and the viewpoint changes are two other meaningful causes that detriment the results.



**Figure 3.10:** Examples of the main problems of our CBIR system. a) Confusion in images with similar structure b) Changes in the point of view c) Occlusions d) Changes in the illumination.

Nevertheless, we see that our approach tolerates some slight image variations than the other approaches penalize. Figure 3.11 shows some examples of these kind of variations.



Figure 3.11: Examples of image variations that the system do overcome.

Once we have seen the experimental part of our proposal we follow with the discussion and the conclusions of the work.

### 3.6 Discussion

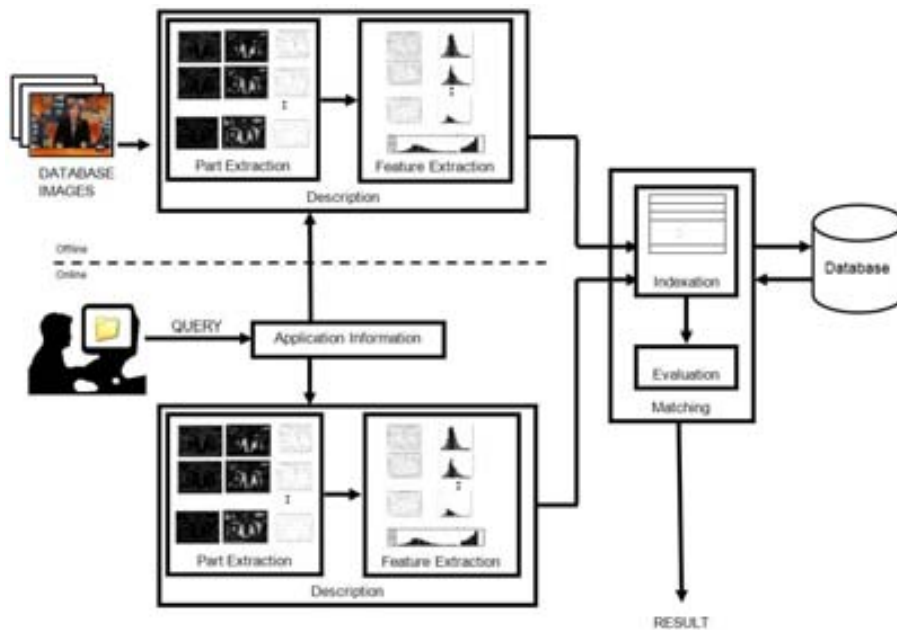


Figure 3.12: Visual resume of the modules of our approach. Image extraction: peaks of the distance map obtained from the contours at several resolution levels. Features: combination of the angular histogram of the triangulation at every resolution level. Feature Indexation: list of descriptors.



### 3.6.1 Conclusions of our approach

We have developed a descriptor that encodes the layout of an image using a histogram-based representation. The descriptor analyzes the image parts from a bottom-up direction using a multi-layer representation. We extract the image parts as the regions defined by boundaries of the image. Then we encode their relative positions using the properties of a Delaunay triangulation.

The descriptor is easy to compute and can be extracted for general purpose in any image. It is compact, it consists in a vector of 23 bins, and its content is normalized. These two properties make it suitable for image retrieval and indexing applications.

We have applied the descriptor in a video browsing application. Analyzing the similarity values of a given image we are able to detect the scenes along the clip that contain this kind of image.

The image description is sensible to changes in the illumination, relevant occlusions and changes in the point of view. Nevertheless it has prove a major robustness than the description based in the corners.

Beyond the comparison of our proposal with the Tao's algorithm, our main conclusion is related to the multi-resolution analysis. We have observed that the incorporation of the multi-resolution analysis plays an important role in the process of image description. If we compare the performance of the system using one or more resolution levels, we can see that the scale-space description helps in a significative manner to improve the results.

We have presented a quite reduced experimentation of our proposal that could be expanded in several ways. Notice that the feature extraction is only defined by the Multi-scale triangulation descriptor. Hence, we could combine it more descriptors related to other visual features such as the color. Moreover, a wider evaluation should be made in order to compare the proposal with other state of the art methods.

### 3.6.2 Conclusions of the query-by-selection paradigm

The query-by-selection paradigm can be understood as the formulation strategy that requires the minimum intervention of the user. He selects an input image either from an external or internal source and submits it to the retrieval system. Then, the engine reports all those database images that are similar to the query ordered from more to less similarity degree.

The main drawback of this query paradigm is the so called page zero problem. Thus, the user needs to provide an image similar enough to the set he expects to retrieve. Notice that many times the similarity is evaluated from the low level image features. Then, it can be difficult to the user to provide a sample image that shares the desired properties that are visual-based, not semantic-based.

As an alternative to the query-by-selection, some systems allow the user to deal with a set of predefined images. These predefined images that act as templates and have an associated semantic concept. Hence, they are represented as icons in the interface of the retrieval system and the user can select them to create his query. In the next chapter we introduce this evolution of the query-by-selection that is named *Query-by-Iconic Composition*.

# Chapter 4

## Query-by-Iconic Composition

### 4.1 Introduction

In the query-by-iconic composition, the user creates the input image using a set of the predefined templates available in the system interface. These templates are represented by icons and use to be related to a semantical meaning. In computer science, an icon can be defined as:

An icon is a visually segmented object which tells the viewer about an inside message or information (concept, function, state, mode, etc.) assigned by the designer [FK91].

The icon-based systems represent an improvement in the friendliness of the human-computer interaction respect from the textual-based interfaces. Nevertheless, special attention has to be given in the design of an iconic application. Thus, a good iconic interface has to minimize the subjectivity of the symbol interpretation and avoid that different people could convey to different meanings. Even though several authors have studied this problem [FK91] [KIK91], no standard approach has been yet imposed. Another keypoint of the interface is the management of a large number of functionalities. Thus, as more functionalities are allowed, a larger amount of icons are needed and the power discrimination of their interpretation decreases. Overcrowding of icons can be controlled by composition mechanisms defined by syntax rules. As an example, the Media Streams [Dav93] is a video editor that allows users to create multi-layered iconic annotation of video content. The organization of the icons in categories allow users to browse and compound over 2500 iconic primitives by means of a hierarchical structure (see Figure 4.2).

In the field of CBIR, iconic queries are related to the concepts contained in the database scenes. Visual-concept detection implies the modelling of the elements related to the icons. For example, to find an image with a beach under a blue sky, most systems require the user to translate the concept of the picture to a model with a particular color and texture.



Finally, we can illustrate two applications for online shopping: Like.com<sup>1</sup> and Pixta<sup>2</sup>. These applications work on visual similarity between images of products such as like shoes, jewelry, bags, etc. They allow composing the desired object according to diverse shape templates and colors patterns. Then they retrieve the images of the products, their price and the website where they can be bought. A screenshot of the Like.com is shown in the Figure 4.2.

Otherwise, retrieval systems of broad domain deal with images of high variability containing real scenes of different thematic. Then, the more variability of the database content, the more difficult is the automatic object labelling. Modelling real world objects from raw image features is a challenging task. Nowadays, the use of learning strategies such as AdaBoost or SVM, is an intensive focus of research in order to improve the detection task. Modelling faces is one of the challenges that have attracted most the attention of the scientific community. Face detection is a very useful tool in image retrieval since it allows classifying which images contain instances of people. In addition to the low level features (color, texture, etc.), several search engines such as MIR [SZR00], Photobook [PPS96] or WebSeer [FSA96], take into account the face presence as an additional feature to describe a scene. Moreover, icons with texture meanings such as 'sky', 'grass', 'sea' or 'buidings' are also common concepts for query composition that allow to describe landscapes and city scenes [ATY<sup>+</sup>95] [Lew00]. Figure 4.2 shows an example of the ImageScape engine [Lew00] where a query is created to retrieve images of water below a person.



**Figure 4.2:** Examples of icon-based interfaces in retrieval systems of narrow and broad domains a) Like.com b) ImageSearch (reprinted from [Lew00])

In this chapter we present a CBIR approach that uses an iconic-based formulation. The system follows a classification strategy and preprocesses the database images

<sup>1</sup><http://www.like.co.uk/>

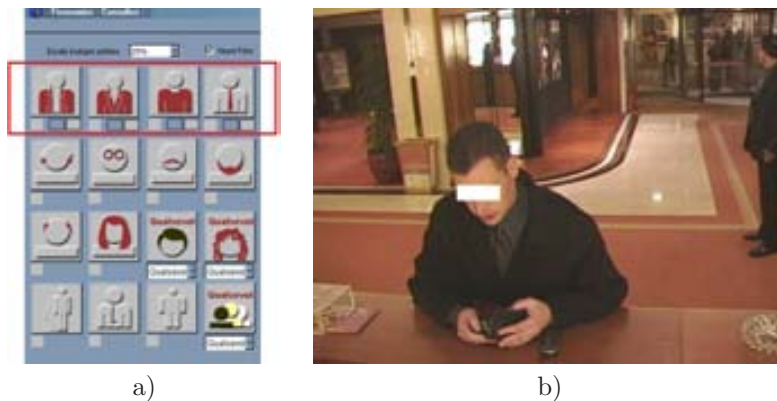
<sup>2</sup><http://www.pixsta.com>

according to the iconic concepts. We propose to model these iconic concepts with a graph of regions. We illustrate this idea with a surveillance application related to the visual appearance of the people.

## 4.2 A CBIR system based in region graph models

We present a CBIR system that uses a set of icons to represent the query objects. These objects are modelled by an attributed graph that contains the features of the parts and their structural relations. Given a scene, the system identifies which object appears in the image. Then, it uses label a description and stores the scene in the database. This way, when a query is submitted, the engine translates the iconic formulation to the label description. Hence, the scenes containing the same description as the query can be retrieved from the database.

To illustrate the retrieval strategy we have used an application related to the surveillance area. The application is sustained by a database of images built at the entrance of a building. When people check in the building, they provide their personal data (name, age, etc.) in the desk of the hall. At this moment the system captures their image and constructs an automatic description of their appearance. The personal data and the visual description are stored in the database of the system. Thus, if the security staff needs to identify somebody, they can query the system using the appearance description of the person. Queries are formulated through an iconic interface that allows creating the symbolic image of the people's physiognomy. We have integrated our retrieval proposal in the specific description of the clothing appearance. Figure 4.3 shows a snapshot of the icons of the system and a sample image of a scene.



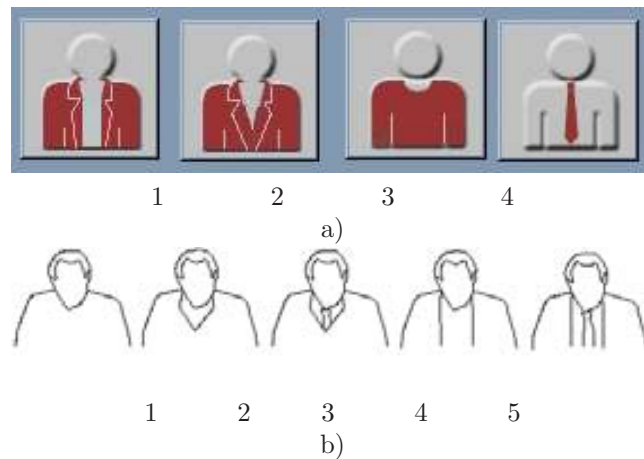
**Figure 4.3:** a) Icons of the system related to the human appearance. The first row are the icons related to the garments. b) Scene captured at the entrance desk.

Next we detail the internal processes of the modules that form the CBIR system. Even though our proposal can be generalized to any application, we exemplify the explanation with the iconic description of the clothing.

### 4.3 Description

The icons of the system allow the user to compose the appearance of query he wants to perform. According to their possible combinations, the system identifies a set of templates that we call models.

In the case of the cloth description there are four icons related to the garments. As we can observe in the Figure 4.4, the first two icons refers to the most external garment and indicates if the person wear it buttoned or unbuttoned. The third icon refers to the most internal garments and the fourth one indicates if the person wears a tie. The selection of these icons can lead to represent five types of model compositions.



**Figure 4.4:** a) Icons of the garments. b) Models we can construct combining them: Model 1 is given by the icon 3; Model 2 is the combination of the icons 2 and 3; Model 3 results from the icons 2, 3, 4; Model 4 from 1 and 3; Model 5 is formed by 1, 3, and 4.

As we have introduced in the previous section, there exists two ways of describing an image according to the iconic options: the *classification* approach of the *abstraction* approach. We have decided to apply the classification option and describe the database images with one of the possible models. This is done before the images are introduced in the database. Since classification can be done off-line, it is interesting in those cases where the image description needs a certain degree of computational effort. Then, the query process can benefit from this previous analysis to achieve maximum speed.

To obtain the description of a scene according to one of the above models (Figure 4.4) we use an attributed graph to represent both the models and the scene. The graph of the database image is submitted to a set of operations until finding the most suitable configuration to assimilate it with one of the cloth models. The Figure 4.5 illustrates the whole process that we describe next.

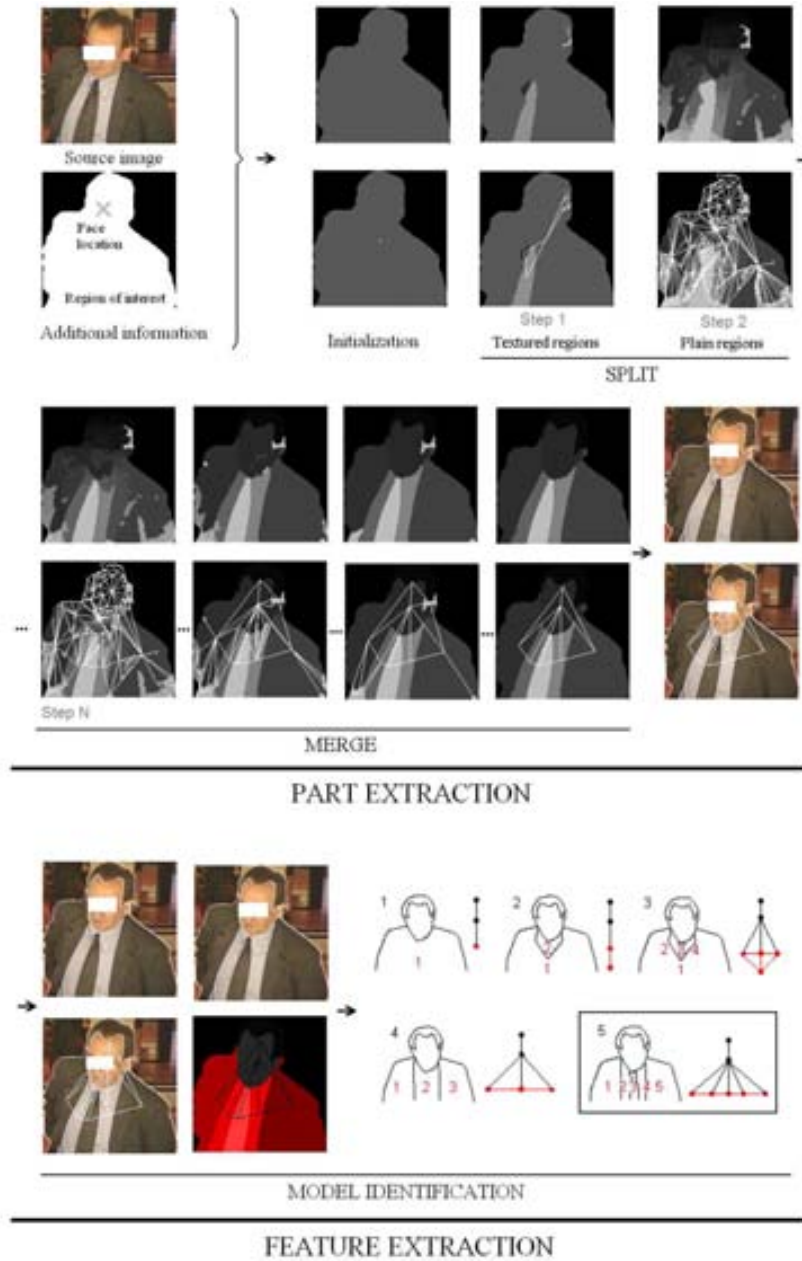


Figure 4.5: Process of the image description according to the models.



### 4.3.1 Part extraction

In the CBIR system we identify as object parts the regions of an image. We understand as a region, a set of connected pixels that result from a segmentation process. The image regions are modelled by an attributed graph that encode their relations and their features. Hence, a segmentation procedure allow the graph structure evolve until the final part extraction is obtained.

Next we detail the exact information of the graph and then we expose its use in the part extraction operations.

#### Structure of the scene graph

We model an image as a set of non-overlapping regions structured by an attributed graph. The graph  $G$  is formed by a set of nodes  $N$  and a set of edges  $E$ . While each node identify a region, each edge represents a relation between two of them. We also provide two labelling functions,  $L_N$  and  $L_E$ , that are in charge to compute the features of the nodes  $F_N$ , and the edges  $F_E$ .

$$G = (N = \{n\}, E = \{e\}, L_N : N \rightarrow F_N, L_E : E \rightarrow F_E)$$

*Node Features*  $F_N = \{P(n), BB(n), A(n), CH(n), AC(n), AI(n), B(n), T(n)\}$ : The features  $F_N$  describe a region and contain information related with its spatial position in the image ( $P(n)$  and  $BB(n)$ ), the area ( $A(n)$ ), the color ( $CH(n)$ ,  $AC(n)$  and  $AI(n)$ ), the presence of boundaries ( $B(n)$ ), and the presence of texture ( $T(n)$ ).

*Edge Features*  $F_E = \{BD(e_{ij}), ACD(e_{ij}), AID(e_{ij}), CHD(e_{ij})\}$ : These features define several metrics of similarity between the regions that compose an image. They concern about the presence of intensity changes between two regions ( $BD(e_{ij})$ ) and their color distances ( $ACD(e_{ij})$ ,  $AID(e_{ij})$ , and  $CHD(e_{ij})$ ).

Next we expose what exactly these features mean and how we compute them. We start defining the node features and we illustrate them in the Figure 4.6.

- **P(n): Position** Coordinates of the region pixels.
- **BB(n): Bounding Box** Coordinates of the minimum bounding box that encloses a region.
- **A(n): Area** Number of region pixels.
- **CH(n): Color Histogram** The feature CH contains the histogram information of the most representative colors of a region. We have set them experimentally to a number  $nc$  of 25. We have obtained them quantizing the image by the octree algorithm of Gervautz and Purgathofer [GP90]. We name  $Q(n)$  the resulting quantized region,  $c$  a representative color,  $Pal$  the complete palette, and  $\vec{H}$  to its histogram.

$$CH(n) = ( \overrightarrow{Pal} = [c_1..c_{nc}], \overrightarrow{H}, Q(n) )$$

Computation: The octree algorithm of Gervautz and Purgathofer [GP90] will be detailed in the next section where we expose the split strategy of the segmentation process.

- **AC(n): Average Chromaticity AC** have the chromatic information of the average color of a region. Thus, we translate the average RGB color of a region to the plane  $r + g + b = 1$ .

Computation: Being  $av$  the average color of the region  $av = \frac{\sum_{i=0}^{A(n)} p_{xy}^i}{A(n)}$

$$AC(n) = \left( \frac{R(av)}{R(av)+G(av)+B(av)}, \frac{G(av)}{R(av)+G(av)+B(av)}, \frac{B(av)}{R(av)+G(av)+B(av)} \right)$$

- **AI(n): Average Intensity AI** contains the mean intensity of the region.
- **B(n): Boundary Information** This feature is related to the detection of the boundary information.

$$B(n) = \{B_h, B_l\}$$

Computation: B contains the unified response of the Canny edge detector on every RGB channel. It captures the edge presence between two color regions even though they have the same intensity. We use two kind of B information according to the degree of Gaussian smoothing applied when obtaining the contours: high ( $B_h$ ) and low ( $B_l$ ).

- **T(n): Texture** It is a boolean value referred to the texture presence. We will detail its computation in the next section where we expose of the split process.

Next we follow with the explanation of the edge features. All their values are in the range  $[0, 1]$ . They are exemplified in the Figure 4.7.

- **BD( $n_i, n_j$ ): Boundary Distance** This function defines a metric about the presence of edges in the commune boundary between two regions  $n_i, n_j$ .

Computation: This metric consists of computing the amount of boundary pixels  $B(n)$  belonging to the commune boundary  $CP$  of  $n_i$  and  $n_j$ . The result is normalized due to the complete length of this boundary.

$$BD(n_i, n_j) = \frac{\#(B) \in CP(n_i, n_j)}{\#CP(n_i, n_j)}$$

- **ACD( $n_i, n_j$ ): Average Chromatic Distance ACD** is a metric that evaluates the Euclidean distance of the chromatic values of two regions.

Computation: This distance  $ACD$  is computed as the Euclidean distance between the average chromatic information  $AC$  of the regions  $n_i$  and  $n_j$ . To obtain a metric in  $[0, 1]$  we normalize it by a value  $D_{max}$ . This value is the maximum distance value that we can calculate between two colors in the chromatic plane ( $r + g + b = 1$ ) where the average chromatic information  $AC$  is computed.

$$ACD(n_i, n_j) = \frac{|AC(n_i) - AC(n_j)|}{D_{max}}$$

- **AID( $n_i, n_j$ ): Average Intensity Distance AID** measures the Euclidean distance of the mean intensity values  $AI$ .
- **CHD( $n_i, n_j$ ): Color Histogram Distance** This is a distance metric over a pair of color histograms.  $CHD$  value corresponds to the quadratic distance of the node features  $CH$ . We detail its concrete computation in the next section where we expose the merging criteria of the image regions.

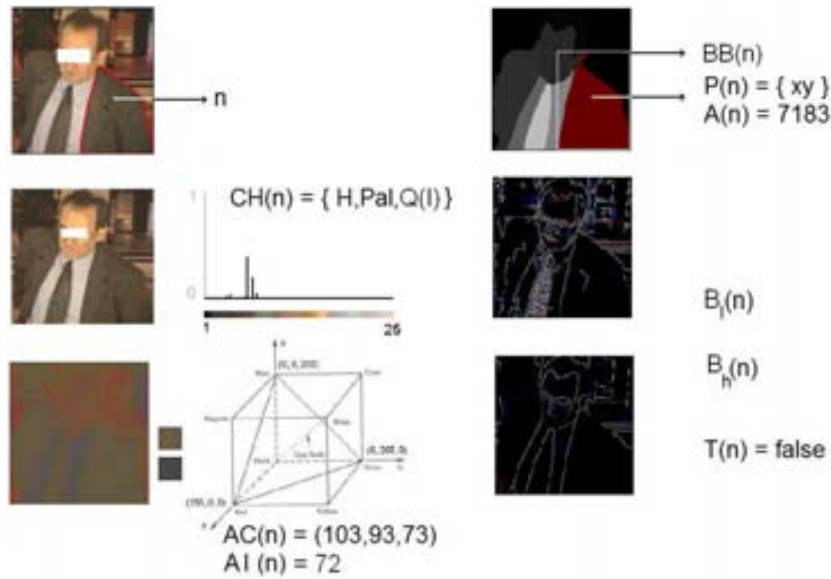


Figure 4.6: Example of the features of the node n.



Figure 4.7: Example regions  $n_1, n_2, n_3, n_4, n_5$ . a) Boundary Distance:  $BD(n_2, n_2)=1$ ,  $BD(n_4, n_5)=0.9$ ,  $BD(n_2, n_5)=0$ . b) Average Color Distance:  $ACD(n_2, n_4)=0.2$ ,  $ACD(n_1, n_2)=0.68$ . c) Color Histogram Distance:  $CHD(n_2, n_4)=0.12$ ,  $CHD(n_1, n_2)=1$ .

Until now, we have exposed the information contained in both nodes and edges of the graph. Next we see the operations that we can apply on this structure.

### Part extraction operations

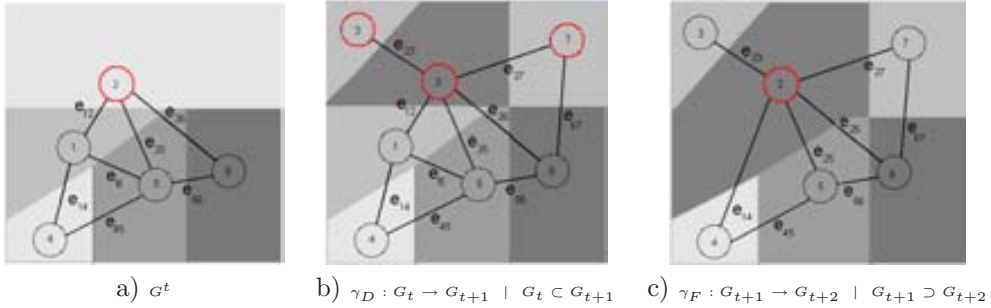
In order to extract the parts of the scene, we use a color segmentation based in a split and merge strategy. The proposed method belongs to the category of hybrid segmentation methods because it combines characteristics of region based methods and boundary based methods. It uses some sort of uniformity conditions to define the regions and some sort of non-uniformity conditions to segment them. Whereas the uniformity conditions guarantee the homogeneity inside the region, the non-uniformity conditions distinguishes abrupt changes in the image that make the regions emerge. Specifically, our method combines criteria of color homogeneity according to a process of clustering in the color space and criteria related with edge information.

The split and merge process is guided by two graph operators that allows the structure to grow and to diminish. These operators are the fusion operator  $\gamma_F$  and the division operator  $\gamma_D$ .

In one hand, applying the divisor operator to the graph  $G$  in the moment  $t$  ( $\gamma_D : G_t$ ) generates a new graph  $G_{t+1}$  that can be considered as the result in the moment  $t + 1$  of an expansion of  $G_t$ . In the other hand, the fusion operator  $\gamma_F : G_t$  obtain a new graph  $G_{t+1}$  but, in this case, the result is a reduction of  $G_t$ .

$$\begin{aligned} \gamma_D : G_t &\rightarrow G_{t+1} & | & G_t \subset G_{t+1} \\ \gamma_F : G_t &\rightarrow G_{t+1} & | & G_t \supset G_{t+1} \end{aligned}$$

As it is illustrated in the Figure 4.8, after a step of graph expansion or contraction, the operators are in charge to recalculate the features of the nodes and the edges and restructure the graph (remove obsolete edges, etc).



**Figure 4.8:** a) Graph at the step  $t$  b) Application of the division operator over  $\gamma_D : G^t$  as  $\gamma_D(n_2)$  into three nodes  $n'_3, n'_2, n'_7$ . The result  $G^{t+1}$  implies the application of  $L_N$  over  $n'_3, n'_2, n'_7$  and  $L_E$  over the new edges  $e_{2,3'}, e_{2,7'}$  and  $e_{6,7'}$ . c) Application of the fusion operator over  $\gamma_F : G^{t+1}$  as  $\gamma_F(n_1, n_2)$ . The result  $G^{t+2}$  implies the application of  $L_N$  over  $n_2'$  and  $L_E$  over  $e_{2,3'}, e_{2,4'}, e_{2,5'}, e_{2,6'}$  and  $e_{2,7'}$ . We also observe the removing of two edges:  $e_{1,2}$ , that would connect the same nodes as  $e_{2,5'}$ , and  $e_{1,5}$  that would form a loop in the node  $n_2'$

The part extraction process of our retrieval system consists in three steps: initialization, split and merge. The split phase is made of two stages that makes the graph grow according the features of the nodes. Otherwise, the merge phase is an iterative process that uses the features of the edges to contract the graph.

**Initialization :**  $G^1$

**STEP 1 :** Split:  $\gamma_D : G_1 \rightarrow G_3$

**STEP 1.1 :** Texture split:  $\gamma_D : G_1 \rightarrow G_2$

**STEP 1.2 :** Plain color split:  $\gamma_D : G_2 \rightarrow G_3$

**STEP 2 :** Merge:  $\gamma_F : G_3 \rightarrow G_N$

Next we detail which procedures and features are involved in the above steps.

**Initialization** The retrieval system has an independent module that is in charge to extract the region of interest where the person is located. It provides a mask and a point of reference where the face is located. This point of reference will be used in the next phase of the description to identify the model. Thus, from the image mask we initialize the graph  $G$  as a unique node  $n$ . In this node we compute the features  $F(N)$  that will be used in the split phase.

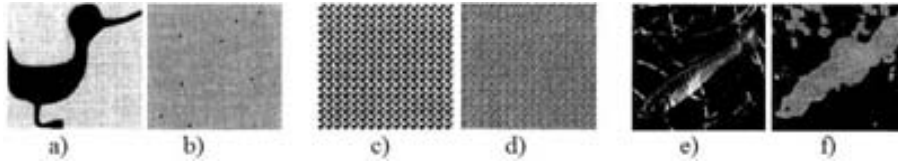
**STEP 1: Split** To split the regions of the image we deal with homogeneity measures on the features related to the nodes of the graph.

**STEP 1.1: Texture split.** Since contours are commonly used to describe textures, we use them to identify the textured regions of the image. The general idea of our process is to consider as textured regions those image zones with a minimal area that present a high density of contours checked at certain frequencies.

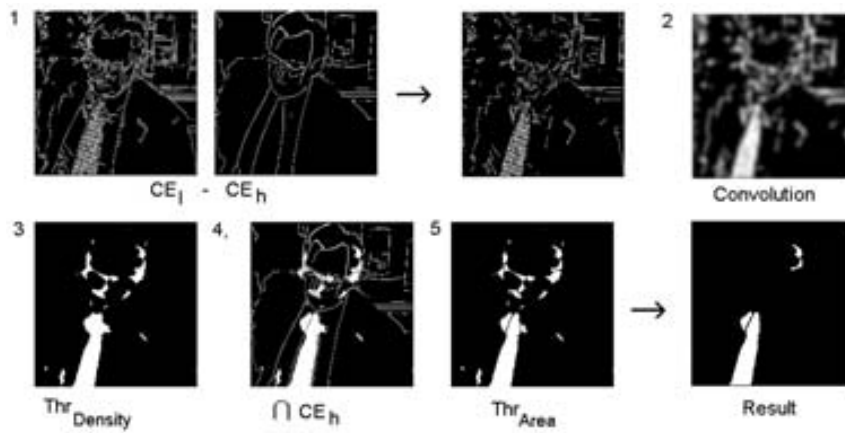
Our strategy is inspired in the Edge Histogram Descriptor (EHD) of the MPEG-7 standard [YCYB02] and in the texture segmentation proposed by Karu [KJB96]. Like the *EHD* descriptor, we use the spatial distribution of edges to describe a texture (see section 2.3.2). Moreover, like the Karu's [KJB96] strategy, we describe the content of the structures of interest at several scales. Then, we apply a density function in a small  $s \times s$  window around each pixel and, finally, a threshold value is set for the density result (see Figure 4.9).

The exact detection steps are graphically showed in the Figure 4.10. The node feature  $B$  contains two Canny edge maps,  $B_l$  and  $B_h$ , obtained after a low image smoothing and a higher one. Thus, we consider as boundary information related to the texture, the result of subtracting of  $B_h$  from  $B_l$ . Some examples of texture splitting are shown in the Figure 4.11.

After the texture-based splitting, the information related to the properties of the nodes are updated and the information of the texture presence  $T$  can be instanced.



**Figure 4.9:** Karu's texture segmentation a) c) Pattern at different scales. b) d) The interest points are the local extrema of (a) and (c). f) Textured regions of (e) with average density of extrema between 0.04 and 0.16. (Reprinted from [KJB96]).



**Figure 4.10:** The five steps of the texture discrimination process



**Figure 4.11:** Examples of the texture discrimination process

**STEP 1.2: Plain color split** We want the regions of our objects to be of homogeneous color. Thus, we analyze the dominant colors of the regions and we split them according to this property.

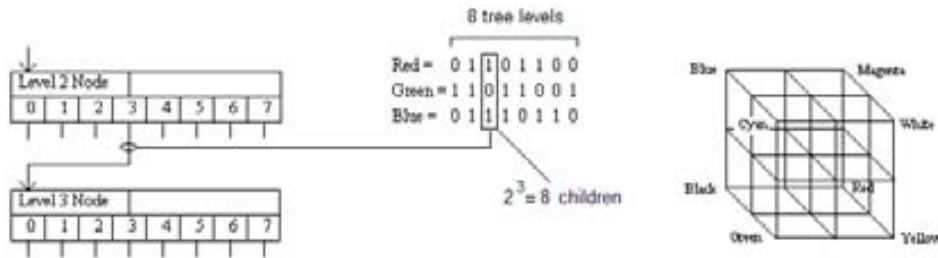
In the previous definition of the node features, we denoted  $CH(n)$  the information of the color quantization of a region. Concretely, the feature  $CH$  contains the quantized region  $Q(n)$ , the array of representative colors  $Pal$  and its histogram  $\vec{H}$ .

$$CH(n) = ( \vec{Pal} = [c_1..c_{nc}], \vec{H}, Q(n) )$$

A plain region is split according those sets of connected pixels that share a commune representative color  $c_i$  of the palette  $Pal$ . In other words, a plain region is formed by all the connected pixels that have the same color value in the quantized region.

To obtain the quantization we apply a pixel-based technique that consists in a clustering of the color space. The clustering is done according to a fixed number of colors  $nc$  that is known in advance. We have concretely used the octree quantization algorithm of Gervautz and Purgathofer [GP90].

The octree quantization is based in an hierarchical representation of the RGB space. It defines a tree structure where each node has eight children that represent the subdivision of the space into eight octants. The tree itself has nine levels, corresponding to the root node plus eight levels referred to the bits representing each primary color. Every color pixel is analyzed and assigned into a bucket of the tree. The tree path is defined by the binary representation of the color, from the most significant bits to the less significant bits (see Figure 4.12).



**Figure 4.12:** Determining a child node from color bits. Division of the RGB color cube into eight sub-cubes.

The octree is expanded at the same time that the region pixels are analyzed. Meanwhile, the size of the octree is reduced whenever the number of inserted colors (that is, the current number of leaf nodes) exceeds our predetermined limit  $nc$ . When the process is complete, the average colors of the leaves are used to initialize the palette  $Pal$ . The quantized region  $Q(n)$  is obtained by exploring the routes of the octree according to the original color pixels of the region.

We have observed that the search of dominant colors adapts the splitting process to the content of the region even though it contains areas of very similar colors. Some examples of its results can be observed in the Figure 4.13.



**Figure 4.13:** Plain split examples: original image, quantized image and palette, resulting regions

At the end of the split step we obtain an oversegmented representation of the regions of the scene. Then, we apply a merge step that provides the final result of the part extraction.

**STEP 2: Merge** We evaluate the similarity of the pairs of adjacent regions in order to decide if they can be merged together. This similarity is computed by a function  $D$  that uses the features stored in the edges. It uses a different formulation depending on if the pair of nodes are textured or plain. The result is a measure between 0 and 1 that can be computed as follows:

$$D = \begin{cases} \alpha_{ACD}ACD(n_i, n_j) + \alpha_{BD}BD(n_i, n_j) & \text{if } T(n_i) \text{ and } T(n_j) == \text{false} \\ CHD(n_i, n_j) & \text{Otherwise} \end{cases}$$



The fusion operator is applied iteratively to the pair of adjacent regions with minimum distance. The merge process stops when all the distance values between the adjacent nodes are higher than the minimum value  $Thr_d$ . Next we detail the benefits of choosing these concrete feature combinations.

**Plain regions merge.** When we deal with scenes obtained in an uncontrolled environment, we can find many artifacts that difficult the segmentation of the objects. One of the most common is the effect of the illumination, which provides intensity changes on the surface of the objects. To try to avoid it, we have analyzed the similarity of two colors treating in a separate way the chromatic information from the intensity one.

Thus, we have combined the average chromaticity of two regions  $ACD(n_i, n_j)$  with the presence of contours in their commune boundaries  $BD(n_i, n_j)$ . The parameters  $\alpha_{ACD}$  and  $\alpha_{BD}$  of the function  $D$  allow to weight the two type of information. We consider that two regions do not belong to the same object, if they have a different chromatic value or if there exists an abrupt intensity change between them. This change is represented by the detected boundaries in their commune region limits. Notice that our distance do supports the merging of regions that present a progressive intensity change. Then, we can merge those regions that have similar chromatic values and do not have any edge between them.

**Textured regions merge** The scenes of the retrieval system can present a wide variety of patterns. Since they can be even unstructured, we only rely on the properties of their color histograms.

We consider that two regions of the graph can be merged if its commune edge presents a low value of the feature  $CHD$ . This feature contains the distance computed from the color histograms  $CH$  of the two nodes. We have used the quadratic form distance because it considers the cross similarity between colors. Then, if two textured regions present a set of colors that are slightly different, this metric can take it into account (see 2.6). The quadratic form is defined as a similarity color descriptor in the MPEG-7 encoding [YCYB02] and is used in some works [DMK<sup>+</sup>01] for region based image retrieval. Its formulation is as follows:

$$CHD(CH_i, CH_j) = \sqrt{((H_i - H_j)^T A (H_i - H_j))}$$

where  $A = [a(k, l)]$  is a similarity matrix, and  $a(k, l)$  denotes the similarity between the colors of the bin  $k$  and  $l$ .

$$a_{k,l} = 1 - \frac{\min(\|c_k - c_l\|, d_{max})}{d_{max}}$$

We use the Euclidean metric to compute the color distance. Moreover, we set  $d_{max}$  as the maximum feasible difference and normalize the result according to it.

Until this point, we have seen the first phase of the image description process. We have presented the steps to extract the relevant information from the image using a graph structure. Next, we expose the second part of the description that is centered in the model clothing identification.

### 4.3.2 Feature extraction

The feature extraction module is in charge to identify the graph of parts of the scene with one of the models of the application. In order to do that, the models are also represented as a graph of attributes. Then, a process is in charge to evaluate the best assimilation between the graph of the scene and the graphs of the models. Next we explicit which features are used in the graph of the models.

#### Structure of the model graphs

We describe a model  $M$  as an attributed graph  $G_M$  that contains the ideal properties of an object. The nodes  $N_M$  represent the object parts and the edges  $E_M$  represent their relationships (see Figure 4.14). Like the representation of the scene, the parts of the model objects are also identified with regions.



Figure 4.14: Modelling of the five possible clothing compositions

$$G_M = (N_M, E_M, L_{N_M} : N_M \rightarrow F_{N_M}, L_{E_M} : E_M \rightarrow F_{E_M})$$

*Model Node Features*  $F_{N_M} = \{A(n_m^i), BB(n_m^i)\}$  : The node features of a class  $c_i$  are defined by its ideal area ( $A$ ) and its bounding box ( $BB$ ).

*Model Edge Features*  $F_{E_M} = \{S(n_m^i, n_m^j), SP(n_m^i, n_m^j)\}$  : The edge features store the similarity restrictions ( $SI$ ) between the nodes and their relative spatial positions ( $SP$ )

Next we define deeply the meaning of the node features of the models:

- $A(n_m^i)$ : **Area**

Each model part has an ideal area that is represented as a normalized value between 0 and 1. This value refers to the percentage of area of the part regarding to the total area of the model. In the table 4.1 we show the values we have used in the models of the clothing application.

Class	Region 1	Region 2	Region 3	Region 4	Region 5
1	1.0	-	-	-	-
2	0.85	0.15	-	-	-
3	0.85	0.05	0.05	0.05	-
4	0.35	0.30	0.35	-	-
5	0.35	0.1	0.1	0.1	0.35

Table 4.1

NORMALIZED AREAS OF THE GARMENT REGIONS

- $BB(n_m^i)$ : **Bounding Box** Besides the area value, we extract information of the bounding box of the regions. This data is used to define the information of the edges related to the spatial positions of between the nodes.

The edge features of the models are defined as follows:

- $SP(n_m^i, n_m^j)$ : **Spatial Position** To define the relative spatial position between two image parts, we use their bounding box information. Inspired in the 2D String features, we define two kind of labels: AP, referred to the angular position, and LP referred to the topological relation. Figures 4.15 and 4.16 show them graphically. Thus, the spatial positions are denoted:

$$SP(n_m^i, n_m^j) = [AP, LP]$$

where  $AP = \{N, NW, W, SW, S, SE, E, NE, C\}$  and  $LP = \{I, A, O\}$

- AP is computed from the center coordinates of the bounding boxes. It takes eight possible values according to the discretion of the angular differences. We use bins of  $45^\circ$  starting from the angle  $22.5^\circ$  in the opposite clockwise direction. For an easier understanding, we name the labels with a cardinal point nomenclature (East, North-East, North, etc.).

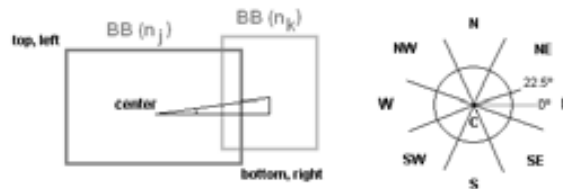
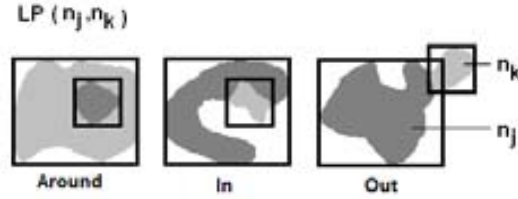


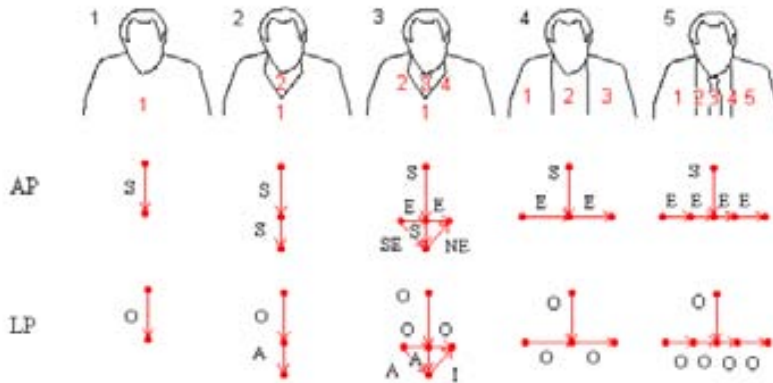
Figure 4.15: AP: Spatial relations due to the angular relation

- $LP$  is related with the limit of the bounding boxes. It defines three possible labels according to the relation of the part  $n_m^i$  respect from the part  $n_m^j$ . These relations are: Around, In and Out.



**Figure 4.16:** LP: Spatial relations due to the limits of the bounding boxes

Given the two sets of labels,  $AP$  and  $LP$ , we can compute a total of twenty-six region relations. In the figure 4.17 we can observe the relative positions between the model parts.



**Figure 4.17:** Cloth classes, number of regions and spatial relations

- $S(n_m^i, n_m^j)$ : **Similarity Restrictions** Depending on the object that a model represents, we can be interested in defining some restrictions among the similarity of two parts. The similarity features takes a boolean value that indicates the similarity requirement between two parts.

$$S(n_m^i, n_m^j) = true|false$$

In the case of the clothing models, we observe that the regions that belong to the same garment have the restriction to be similar. This include regions that

even though they are not adjacent, they belong to the same piece of cloth. The classes 3, 4 and 5 present this particularity because they contain an external unbuttoned garment of an internal garment divided by the tie.

Once we have presented the ideal properties of the graphs of the models we expose how we identify the graph of the scene with one of them.

### Feature extraction operations

In our CBIR system, the features that describe the scenes are the labels regarding the model objects that are present. To detect the presence of the model objects we apply some operations that compare the graph of a scene with the graphs of the models. These operations comprise three phases:

**Initialization:** Set reference node of  $G$  according to every model  $G_M$ .

**STEP 1: For every model** Evaluate the mapping of  $G$  to  $G_M$ :

**STEP 1.1:** Generate mapping combinations.

Identify the scene parts with the model parts (n to 1).

**STEP 1.2:** Evaluate the combinations cost.

**STEP 2:** Classification of  $G$  with a model  $M$

Next we expose step by step how we solve the model classification of the scene graph. The figure 4.20 illustrates the process.

**Initialization** The external information of the CBIR system can provide additional data that participates in the feature extraction. In the case of the clothing application we deal with a reference point regarding to the location of the people face. Then we can set the initial mapping between the scene region that contains this reference with the corresponding regions of the models.

**STEP 1** We want to evaluate which is the best mapping between the parts of the scene  $G$  and the parts of every model  $G_M$ . To explain the process, we focus in a single model  $M$ . Then, our first objective is to generate all the segmented region combinations and then evaluate their proprieties. According to this evaluation, we will choose the solution that fits the model best.

**STEP 1.1** For a certain model  $M$  we generate all the possible region combinations to map the graph of the scene  $G$  with the graph of the model  $G_M$ . This mapping allows a corresponding of n-to-1. That means that a set of scene regions can be grouped to be identified with a single region of a model. Before detailing the algorithm, let us introduce a set of formalized concepts:

$N_M$ :	number of regions for the model $G_M$
$N_S$ :	number of segmented regions belonging to the scene graph $G$
$sn_M^i$ :	a set of scene regions assigned to represent the region $i$ of the model $M$
$csn_M$ :	a combination of scene regions to map the model $G_M$ . It consists in an array of $N_M + 1$ sets of regions $sn$ . Each position represents region $i$ of the model $M$ , $csn_M[i] = sn_M^i$ . The position $N_M + 1$ represent the nodes that do not take part in the interpretation combination.

Being  $N_M$  the number of regions of the model  $G_M$ , we generate the set of all possible combinations  $csn_M$  that match the regions of the scene with the regions of the model. Each possible combination can be understood as a subgraph of  $G$ . We restrict the assignment of a set of segmented regions to a region garment to those groups that form a single connected component.

The process is done by expanding a tree of possibilities. The tree has  $N_S$  levels and each node has  $N_M + 1$  sons. The root of the tree contains the combination related to an empty set of nodes. Then, we apply an iterative process of  $N_S$  steps, one step for each scene region. This process consists in expanding with a new level the tree of possible garment combinations. Then, for each leaf we generate  $N_M + 1$  sons. Each son represents the assignment of a scene region to one of the model regions. Notice that each model is formed by  $N_M$  regions and that we are expanding each branch with  $N_M + 1$  sons. Thus, the group  $N_M + 1$ th represents all the regions that do not take part of the interpretation solution. Once the tree is expanded, every leaf contains one of the possible combinations  $csn_M$ .

**STEP 1.2** At the end of the previous step we had a set of the candidate combinations for a certain model:  $\{csn_M\}$ . Then, the aim of this step is to choose the region combination that fits the model best. We choose the candidate combination that minimizes the cost function  $FCost$ . This function evaluates the matching of scene parts with the ideal features of the models.

The features regarding to the nodes are evaluated by the functions  $FCostA$  and  $FCostI$ . Moreover, the features related to the edges are evaluated by  $FCostSP$  and  $FCostS$ .

- **FCostA (Area):** It evaluates the rates of the ideal areas with the areas of each garment region and the overall amount of areas.
- **FCostI (Internal cohesion):** It computes the internal cohesion of the regions that are assigned to the same model part.
- **FCostSP (Spatial Positions):** The evaluation of the spatial positions of the region garments is a hard condition in order to identify a model. Thus, **FCostSP**

takes cost 0 if the position labels of the garment regions agree with the model. Otherwise it takes maximum cost 1.

- **FCostS (Similarity)**: It checks the similarity restrictions of the disjoint parts of the model that require it (they have a true value). The similarity is measured regarding a maximum value  $Thr_S$ . This cost function provides cost 0 if the similarity is not accomplished, otherwise, it takes value 1.

$$FCost(G_M, G, csn) = \max \left( (\alpha_{AC} FCostA(G_M, csn) + \alpha_{SC} FCostI(G_M, csn)), FCostSP(G_M, csn), \alpha_{SC} FCostS(G_M, csn) \right)$$

The  $\alpha$  parameters weight the individual cost functions for the node feature so that  $\alpha_{AC} + \alpha_{SC} = 1$ . Next we expose the exact computation of the individual cost functions. We have also defined an additional function  $DM$  that computes the similarity between the scene regions. This function is an expansion of the distance  $D$  that we have used in the merging process (4.3.1). We also distinguish between the similarity of textured and plain regions. Thus, the similarity of two textured regions is computed from the color histogram. Otherwise, the similarity of the plain regions is computed from the average chromaticity and the bounding edges (in the case of adjacent regions) or the average intensity (in the case unconnected ones).

$$FCostA(G_M, G, csn) = \frac{1}{N_M} * \sum_{j=1}^{N_M} \frac{\min(A(n_m^i), A(csn[j]))}{\max(A(n_m^i), A(csn[j]))}$$

$$FCostI(G_M, G, csn) = \frac{1}{N_M} * \sum_{k=1}^{N_M} \overline{DM}(n_i, n_j) \\ \forall n_i, n_j \in csn[k] \mid n_i, n_j \text{ are adjacent}$$

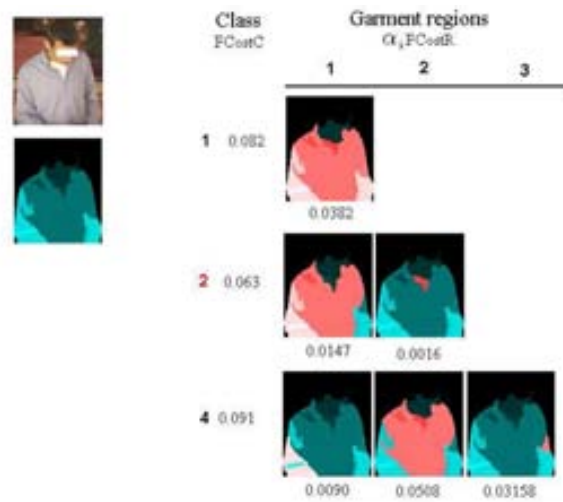
$$FCostS(G_M, G, csn) = DM(csn[k], csn[l]) < Thr_S \quad \forall k, l \mid S(n_m^k, n_m^l) == \text{true}$$

$$FCostSP(G_M, G, csn) = 1 - \min\{SP(n_j^i) == SP(csn[j])\} \quad \forall j = 1..N_M$$

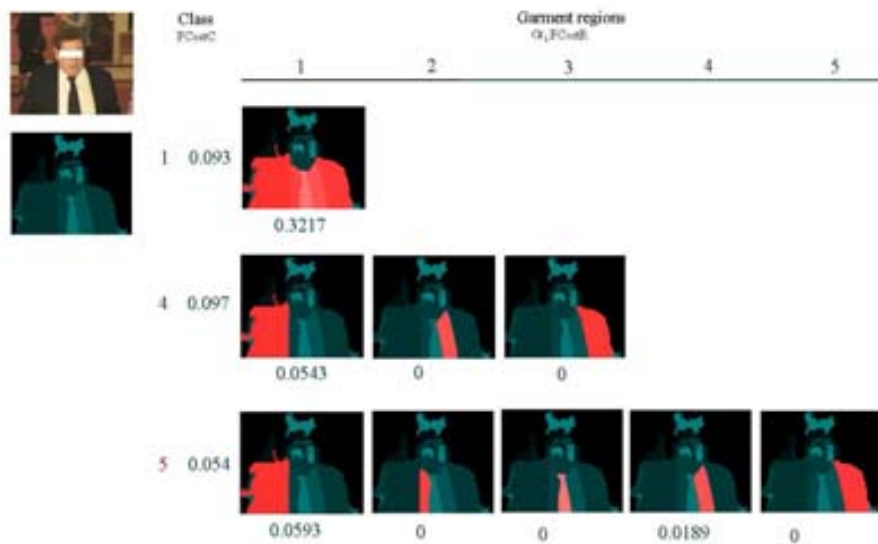
$$DM(n_i, n_j) = \begin{cases} \alpha_{ACD} ACD(n_i, n_j) + \alpha_{AID} AID(n_i, n_j) & \text{if } T(n_i) \text{ and } T(n_j) == \text{false} \\ & \text{and not adjacent} \\ \alpha_{ACD} ACD(n_i, n_j) + \alpha_{BD} BD(n_i, n_j) & \text{if } T(n_i) \text{ and } T(n_j) == \text{false} \\ & \text{and adjacent} \\ CHD(n_i, n_j) & \text{Otherwise} \end{cases}$$

**STEP 2** Once we have computed the best region combinations  $csn_i^l$  due to each clothing class  $c_i$ , we have to compare them in order to decide which one fits our segmented image. At this point, we choose the solution that has obtained a minimum matching cost  $FCostC$ .

Figures 4.18 and 4.19 show the examples of the result on the interpretation over two images belonging to the Model2 and Model5 respectively.

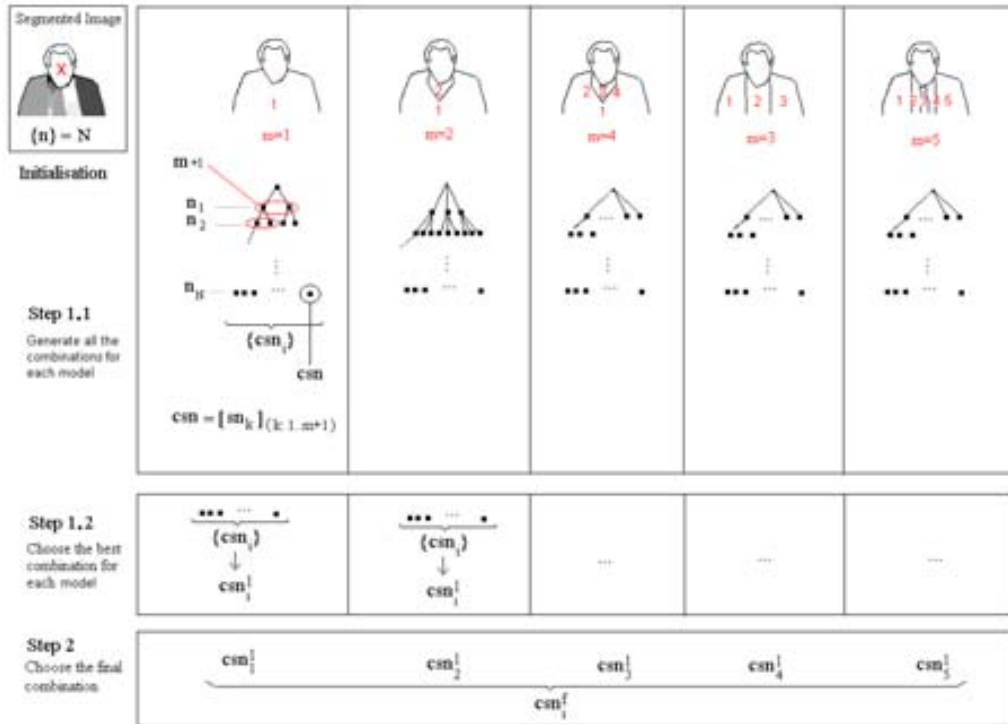


**Figure 4.18:** Interpretation over the segmented results on the upper-left image. We show three possible results belonging to models: 1,2 or 4. The image is classified as Model2 due to the lower value of FCostC.



**Figure 4.19:** Interpretation over the segmented results on the upper-left image. We show three possible results belonging to models: 1,4 or 5. The image is classified as Model5 due to the lower value of FCostC.





**Figure 4.20:** Model Classification. Initialization. Step1: Chooses the best combination for each model. Step2: Chooses the final combination that represents the image.

Once the scene parts are identified with a model, we can describe the image with a label regarding to the possible iconic compositions.

In the case of the clothing application the image features are simply represented as a numerical code (from 1 to 5).

## 4.4 Matching

The matching process allows to retrieve the indexed images according to their features. In the case of the CBIR by-iconic-composition, the features are given by the set of possible models.

#### 4.4.1 Feature indexing

The indexing of the database images can be easily done with a table where the key index is the code of the model composition. This way, we do not deal with any similarity function to retrieve them. Instead, when we make a query, we are just interested in obtaining the database images that match the specified model.

#### 4.4.2 Evaluation

In the CBIR we present the evaluation phase can be omitted. This fact is given because the system characterizes the whole content of a scene with a single descriptor. Hence, there is no need to evaluate any combination of features or check the global coherency of the subparts of the scene. Thus, we can present to the user the set of images that are retrieved from the indexation tables.

### 4.5 Experiments and Results

To evaluate the performance of the system we have used a set of 861 images taken during a day from the real environment of the entrance desk of a building. The test images are distributed into the possible five models as is shown in the Table 4.2:

Database Test Models					
	M1	M2	M3	M4	M5
# images	222	95	14	423	117

**Table 4.2**  
NUMBER OF GROUNDTRUTH IMAGES FOR EACH MODEL

We have applied five queries to the collected data, each one with the iconic combination that defines a model clothing. Then, we have computed the success of every query model using the classical measures of retrieval evaluation (see Table 4.3).

	M1	M2	M3	M4	M5
Precision	0.5276	0.7917	0.7143	0.8960	0.5897
Recall	0.689	0.7055	0.6423	0.7607	0.578
Fallout	0.2111	0.0683	0.0012	0.1339	0.0013

**Table 4.3**  
PRECISION, RECALL AND FALLOUT RESULTS FOR EACH MODEL.

Most of the test images belong to the fourth model (two garments, the external one unbuttoned). This model is the one that provides a better precision performance

followed by the models 2, 1, 5 and 3. The recall also follows a similar tendency, but the fallout obtains their best results in the models 3 and 5 which are less favored by precision/recall measures. Figure 4.21 show some examples of the retrieved images for each of the query models.



**Figure 4.21:** Examples of the retrieved images for the five query models.

Beside the query results we have analyzed the internal processes of the CBIR. In the Table 4.4 we present the classification percentages where we can observe the confusion between model types. The results provide an overall evaluation of the 71.89% of success in relation to the classification of all the images.

The performance of the retrieval system is determined by the *classification* strategy of the description module. Thus, we want to analyze more deeply the two phases that compose it: the part extraction and the feature computation. To study the effects of both submodules we have used a subset of 100 sample images that we have manually segment according to their cloth models.

Correct Classification %					
	M1	M2	M3	M4	M5
M1	68.9189	18.9189	0	12.1622	0
M2	26.3158	70.5263	0	3.1579	0
M3	7.1429	21.4286	64.2857	0	7.1429
M4	22.0402	1.8913	0	76.0686	0
M5	15.3846	0	0.8547	25.9145	57.8462
Overall Success % :				<b>71.89</b>	

**Table 4.4**

PERCENTAGE OF CORRECT CLASSIFICATION BETWEEN MODELS.

### 4.5.1 Evaluation of the part extraction

The set of 100 manual segmented images represents the ground truth of the ideal part extraction. Thus, we want to evaluate the goodness of our automatic approach against the ground truth set.

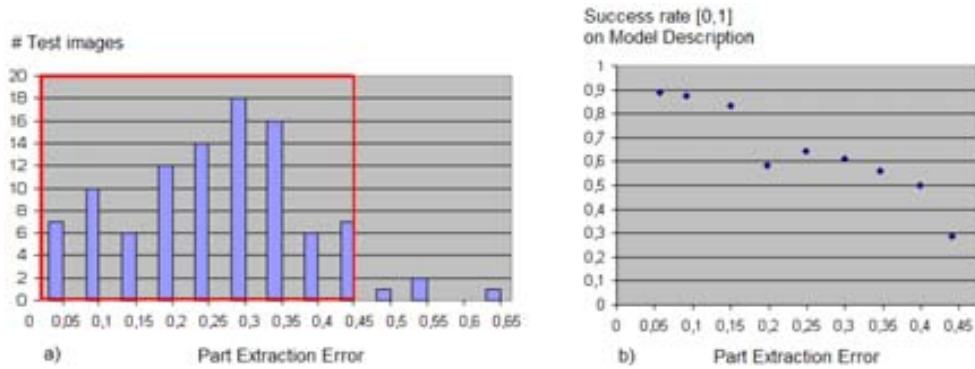
There exists a wide variety of strategies to compare two segmented images [UPH07] [Mar02] [Zha96]. Some of them use internal coherency measure, the mutual information, boundary elements, region differencing, etc. Since our case of analysis is hardly related to fixed combination of regions we have chosen a measure related to the region differencing evaluation. We propose a variation of the known measure called Bidirectional Consistency Error (BCE). BCE does not allow the refinement between two segmentations and measures at pixel level the intersection of the regions. In our case we are interested to make the segmentation comparison not at pixel level but at region level. Notice that two models may differ on the presence of certain garment of very different areas, by instance, the small region of the inner garment that presents model 2 differentiates it from the model 1. Then, our process is applies the same strategy but balances the weight of every garment region despite of its area. Being  $IS_1$  the segmentation of the ground truth and  $IS_2$  the segmentation we evaluate, let define the evaluation function measure Bidirectional Consistency Region Error (BCRE):

$$BCRE(IS_1, IS_2, c_i) = \frac{1}{N_{cl}^1} \sum_{i=1}^{N_{cl}^1} \frac{A(P(n_i^1) \cap P(n_j^2))}{\max(A(n_i^1), A(n_j^2))} \quad \text{where } A(P(n_i^1) \cap P(n_j^2))$$

is maximum  $\forall j = 1..N_{cl}^2$

In other words, BCRE evaluates the average of overlapping between the ground truth image and our segmentation result. Its range is in  $[0,1]$  meaning that the lower the error value, the better the region extraction.

Applying this function to the 100 pairs of test images we have obtained the statistics of the Figure 4.22.



**Figure 4.22:** Statistic of the error on the part extraction a) Respect to the number of images b) Respect to the success on the model identification

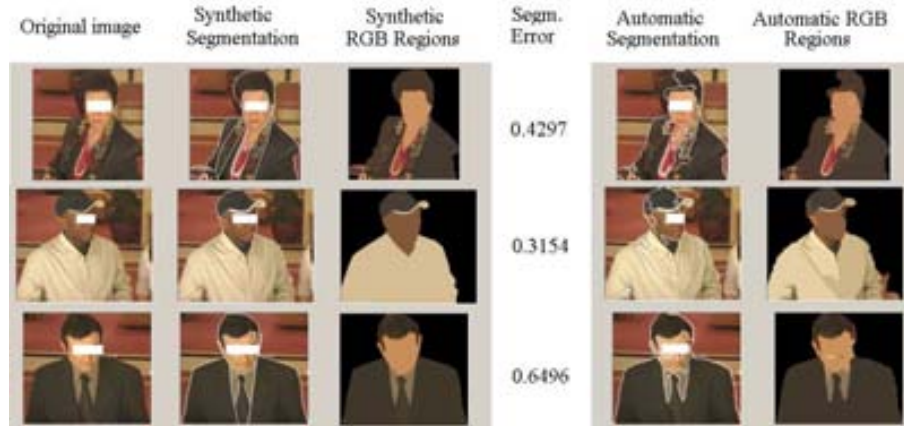
The distribution of the segmentation error according to the amount of test images is shown in the graphic a). According to this statistic we want to analyze which are the requisites of the part extraction accuracy in relation the overall performance of the system. We consider that an image is successfully retrieved if their automatic labelling coincide with the manual one. Given the average error segmentation values we compute the success percentage on the set of test images comprised between each interval of 0.05 segmentation error. We have taken the first nine groups of test images where its cardinality is significant enough to perform the sadistic. The graphic b) shows the result of its process. Its interesting to observe the relation between the metric of segmentation error in relation to the retrieval success. Thus, as the segmentation error grows, the model classification success decreases. Hence, we can see that the segmentation has a direct influence on the final result. Images with extraction error under 0.15 obtain a retrieval success of more than the 80% but when this error exceeds the 0.4 the performance falls into the 50%. For all the image set we have obtained an average segmentation error of 0.247 over 1, so we point to an statistical success between the 60% and 70%.

Figure 4.23 shows some examples of the automatic part extraction and its segmentation error.

Notice that our segmentation strategy is of general purpose and does not use additional knowledge the problem to solve. Then, when we evaluate it according to a specific application we find some discrepancies between the results and the goal. This discrepancies can be coarsely classified in two groups according to an image over-segmentation or an image under-segmentation.



Figure 4.23: Segmentation examples



**Figure 4.24:** Segmentation examples with high error

Over-segmentation Image over-segmentation is usually given for two factors: the presence of complex objects in the image or the presence of objects with a very irregular surface. In the first case, since the synthetic images are constructed by a human, this construction is done according to its knowledge about the aim of the segmentation. In the distinction of our image regions, the human operator deal with additional information of the diversity of the garments that a person can wear. Thus, even though a garment could be viewed as a complex set of regions, the knowledge about the object makes all the regions be identified as a single object. When the automatic segmentation does not group these regions, an image over-segmentation is produced. One example of the over-segmentation can be seen the first image of the figure 4.24. There, we can see a woman wearing a scarf that we expect to be identified as a single object. On the contrary, the automatic part extraction divides it into several zones due to its big and irregular texture. Furthermore, the similarity conditions in the merge process can be too restrictive in those objects that have an irregular surface. In our application it most irregularities are done to the clothes folds in light plain garments. The second image of the figure 4.24 shows, in a white garment, an example of this case of over-segmentation caused by the shadows of its folds.

Under-segmentation Color similarity, in addition to the lack of sharp boundaries in plain regions, are the factors that bring about image under-segmentation. Since the merge step deals with similarity values on the adjacent regions, is difficult to set an agreement to decide, in all cases, when two colors are similar enough to be joined. This effect is exemplified in the third image of the figure 4.24 where we can see a man wearing a black jacket and a black tie. Their color similarity and their nearly imperceptible boundaries causes the application to considers them a single garment.

Notice that the interpretation process is completely dependent on the results of the segmentation step. Thus, if the segmentation process is not good enough, there is

no way to provide the right interpretation solution. A wrong segmentation is specially critical in the case of under-segmentation model classification has not been designed to overcome it.

### 4.5.2 Evaluation of the feature description

We would like to evaluate the performance of the feature description with independence of the previous part extraction. Then, we have ran our method form the 100 synthetic images we have obtained an overall success of 76% on the clothing classification. Even though the input data does not contain any segmentation error the success rate does not reach the maximum. We have observed that there are several causes that explains this phenomena, which we have exemplified in the Figure 4.25.

Occlusions by external objects Sometimes people carry some complements such as handbags or wallets that are classified in the cloth group. When these objects causes important occlusions in the real cloth regions the interpretation process cannot reach the correct solution.

Addition of external objects The complements of the clothing such as scarves, collars, etc. can be understood as other garments that take part in the clothing composition. This can cause the classification of a model 1 into the category 2 when a neck complement is understood as the inner garment.

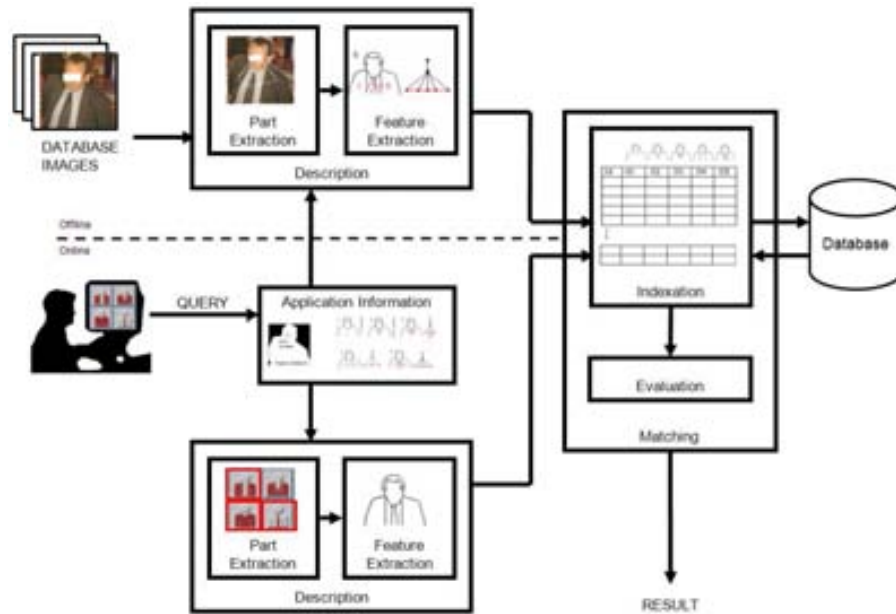
Deformation of the ideal composition The clothing elements are flexible objects that can present a notable variability regarding to the visible areas. This variability can be caused by an altered position of the person in front of the camera. In some cases the person is not situated strictly facing the camera. Then, the spatial positions and the areas of the clothing vary, producing a wrong interpretation. Moreover, we observe that the system deals with a drastic approximation of several types of garments (a pullover, a gabardine, a raincoat, a jacket...) to the concept of a layer structure. When the pieces of cloth do not fit the ideal properties and they are of similar color or texture, they can be joined in order to adapt to another model. The compositions 2 and 3 are the most sensible to this effect because contain little regions that can suffer considerable variations. By instance, notice that the area of a shirt that juts out from a closed pullover is not the same than the area of the same shirt that is visible wearing a buttoned jacket.



**Figure 4.25:** Examples of ideal segmentation. Causes of a mistaken classification: 1) Occlusions 2) Addition of external objects 3) Deformation of the ideal composition



## 4.6 Discussion



**Figure 4.26:** Visual resume of the modules of our approach. Part extraction of the database: regions using split-and-merge segmentation. Features: model identification. Feature Indexation: tables images according to their region garments.

### 4.6.1 Conclusions of our approach

We have developed a CBIR strategy where the queries are done according to an iconic composition. Each possible iconic composition defines a model object.

We have proposed a strategy to recognize and describe a model object in terms of a graph of regions. The graph contains the ideal visual properties according to the color, texture and structural features.

We have observed that there are two main situations that difficulties the task of modelling an iconic object. The first problem is given when the application deals with an uncontrolled scene environment. The second one is given by the need of modelling an object that tolerates a certain degree of flexibility. To try to overcome this two main difficulties we have applied certain strategies in the image description module.

The process of image description involves relevant difficulties when it has to deal with images of general purpose. The presence of shadows, the irregular surfaces and the variability of textures makes the part detection a hard task. We have applied a segmentation process that identifies the image regions with the object parts. The

method follows a top-down approach that, in a first phase splits the image, and then it follows a bottom up-analysis that merges the content.

From the regions obtained in the part extraction process we attempt to identify a models object. Given the difficulties of the part extraction, we do not want the model identification to rely on a single and fixed segmentation. Thus, we have applied an strategy that uses several cost functions to evaluate the best matching between the regions of a scene and the ideal regions of the models. The process attempts to reduce the segmentation problems by allowing an n-to-one region mapping. Thus, we expand a tree of of candidate solutions and we select the one that provides minimal cost. Despite the mapping process presents an exponential complexity, it can be reduced using purge strategies. The hard restrictions related to the position of the region garments can be used to stop the expansion of the tree. Other strategies as a more simple greedy approach could be also applied. Nevertheless, since the feature extraction is done off-line, the proposed method can profit from a more exhaustive evaluation of possible solutions.

We have applied our proposal to a retrieval application where the icons are related to clothing compositions. The results of the experiments show that the main problems are related to the occlusion effects and the variability of the instances of the models. Scenes containing altered positions of the clothing or containing additional regions can provide undesirable effects in the cost functions.

The obtained results are not satisfactory enough for a surveillance application. In this specific case the use of learning techniques could improve the results of the clothing classification. Nevertheless, our objective was to propose a general framework to model the iconic queries.

We have used a certain set of features, but the same strategy could be adapted using other descriptors. In the case of the clothing compositions we have applied a coarse description of the shape and the texture. Thus, these descriptors could be substituted if a model object needs for a more accurate characterization of certain features.

#### **4.6.2 Conclusions of the query-by-iconic composition paradigm**

In the CBIR systems that use an iconic interface the user can formulate the query in semantical way. The iconic templates are related to specific concepts and the query composition does not need to be analyzed from the low level features. The content of the database images can also be pre-computed to be characterized according the iconic concepts. This previous analysis benefits the retrieval process making it faster than the approaches that search for a combination of low level features. Nevertheless if we want to expand the set of icons of the application we should recompute the information of the database.

The intervention of the user in the querying process is more active than in the selection paradigm but it is restricted to the iconic items that the application offers.

When the user needs more freedom to express their queries, some systems allow to create them from the scratch. They provide a drawing interface from which the user can sketch the idea of what he is looking for. This kind of query paradigm is the so called *Query-by-Sketch*. In the next chapter we analyze with are their drawbacks and their benefits in content-based image retrieval.



# Chapter 5

## Query-by-Sketch

### 5.1 Introduction

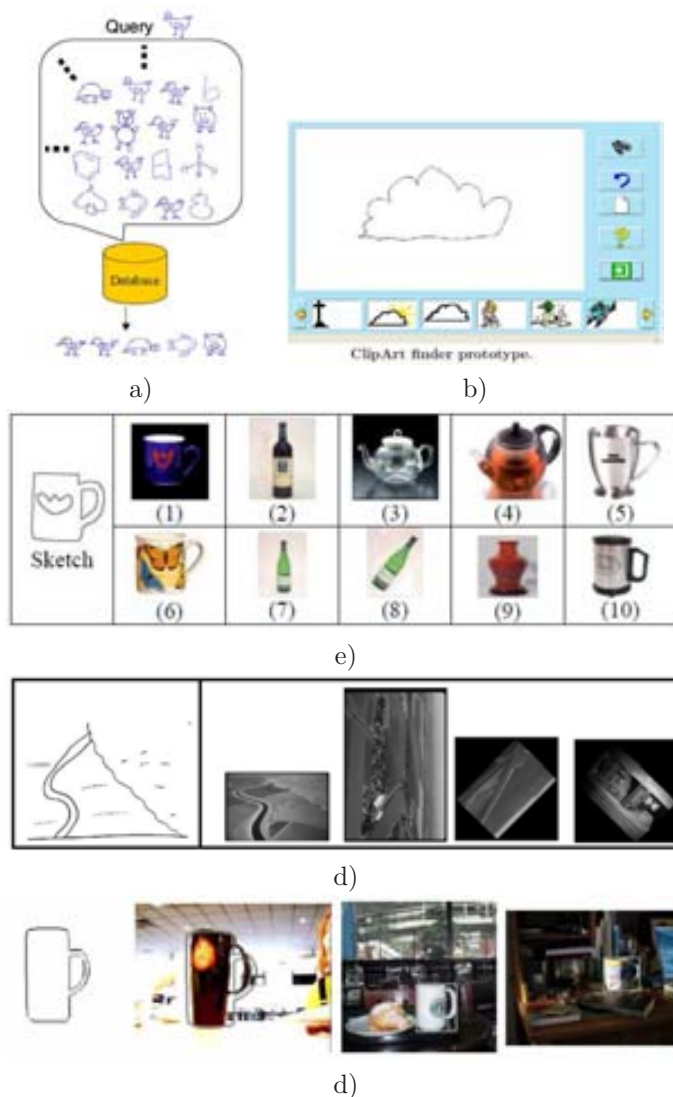
Sketch-based engines simulate the pen and paper tools to provide a natural way to express the information in a graphical way. Nowadays, portable devices that accept handwritten data are increasingly being used for information entry and data manipulation. Since engines such as PDAs, PocketPCs or TabletPCs are available at an affordable cost, there is a growing interest to develop applications that are able to deal with hand-drawn sketches. Sketch interfaces that deal with graphical elements suit a wide variety of applications. Banks and supermarkets have included sketch interfaces to capture and store the signature of the users that make a money transaction. In design applications, some professionals have incorporated the drawing sketches as tools for creating in an interactive way technical charts (electrical or architectural) or even musical scores [MA07] [SM05].

Motivated by this technological trend, CBIR have also been focused in the study of incorporating sketch based queries in the image retrieval systems. This way, the user creates his own query by drawing a rough sketch of the goal image he searches for. This kind of query assumes that the sketch corresponds to the boundaries of the objects present in the scene. Unlike the query-by-selection, that can deal with a wide variety of features (color, texture, etc.), the information provided by a sketch are mainly related to the shape description. Shape is one of the most important perceptual features, and successful shape-based techniques would significantly improve the spreading of general-purpose image retrieval systems.

Among the existing CBIR techniques for retrieving an image, those based in sketch representations are particularly challenging. A sketch representation is characterized by a severe simplification of the image content and by the inherent shape deformation that the user introduces according his own drawing style. These two characteristics involve important difficulties in the retrieval modules related to the part extraction process and the evaluation step. Even though the literature related to sketch matching is very large, we can overview it according these two main points: the nature of

the database images and the matching capabilities of the system.

The query-by-sketch engines draw on different solutions depending on the nature of the database images: sketches, clip arts, or real word scenes. This way, we can find different retrieval approaches to illustrate the benefits and drawbacks of dealing with different types of image collections (see Figure 5.1).



**Figure 5.1:** Database image types a) Sketch b) ClipArt c) Isolated Object d) Complete Scene e) Object Inside a Cluttered Scene. (Images reprinted from [Leu03] [FBRJ04] [LC02a] [CNM05]) and [FTG06])

Some works deal with queries based in sketches to also retrieve sketch images from a database. The database images use to correspond to sketches extracted from handwritten documents corresponding to concept illustrations, flow charts, graphs, drawings of objects, etc. Hence, a retrieval engine that can deal with sketch queries can considerably improve the ability to retrieve associated documentation to this kind of images. In the scenario of sketch-to-sketch comparison the part extraction process can treat both query and database images using the same strategy.

Namboodiri and Jain [NJ04] developed an approach for matching hand-drawn sketches approximating them as a set of lines. Each segment is represented by the position, direction and length. The comparison of two sketches involves computing the weighted Euclidean distance between the features of the two sets of lines. Even though this representation can be generalized to model any shape, is very dependent on the drawing style and is particularly instable to approximate curved shapes.

Leung and Chen [LC02b] used shape information from each stroke to identify it as a line, circle or polygon. They also extract geometric features to define them such as the convex hull, area, etc. The matching between two sketches is done by a combination of their feature similarity and the distance of their centroids. Later on, authors improved their system by considering spatial relationships between strokes.

Liang et al. [LSLF05] performed stroke segmentation based on both user's pen speed and curvature to cut the input strokes into primitive shapes as lines, arcs and ellipses. Then, they represented a sketch as a topological graph that encodes both the primitive type and their relationships (adjacency, parallelism, cross, etc). Instead of computing the isomorphism between topology graphs, they used the set eigenvalues calculated from the adjacency matrix. This kind of signature is known with the name of graph spectrum.

The graph spectrum was also explored by Fonseca in several works related to the matching in clip art collection [FBRJ04]. Clip art use to contain a drawing of an object on a plain background. Then, an hierarchy of parts of this object can be extracted to compute the topology graph. Nevertheless, the automatic application of this strategy can only suit simple objects and, depending on the object, it is not always clear which is the most suitable part hierarchy to define them.

Other works address the problem of retrieving images containing not clip art but photographs or real object on a plain background. Leung et al. [LC02a] partitioned the shape boundary into segments, called tokens, at the points where the turn angle is locally minimum or at the change of sign. Maximum turn angle and orientation are taken as token features. Then, the token set of each shape is stored in the database using M-tree indexing. They introduced an hierarchical analysis to allow the possibility of describing an object according to some inner shapes in addition to the boundary curve.

In addition to the type of database images, we observe that the design of the retrieval system is also related to their matching capabilities. This means that the techniques used would vary if the system allows locating an object in a scene or is limited to a whole scene match.

Some systems avoid the phase of part extraction process and encode the whole information of an image in a single feature vector. These approaches attempt to select the shape information from a scene applying a first phase of contour detection. Then the shape is encoded as a 2D signal using some kind of transformation. In [CNM05] the angular-spatial distribution of the edges is employed for feature extraction using the Fourier transform. Another example is the work of Dong [LK01] where the shape information is encoded using a wavelet transform. Even though these approaches are attractive because of their simplicity and their compact representation, they can provide unexpected results when the system deals with cluttered backgrounds.

In all the above presented approaches, the user is constrained to retrieve images of a very specific type: drawings, clipart, isolated objects or simple scenes. Nevertheless in many kind of applications the users can be interested in retrieving images of objects that are present in cluttered scenes. To be able to retrieve objects inside complex images, the system needs to decompose the image information to analyze which parts belong to the object of interest.

Extract the shape information from a real scene is still one of the unsolved challenges in computer vision. Early proposals in sketch retrieval used manual segmentations for the database information. This is the case of the pioneer system QBIC the database images are represented by a reduced binary map of edge points. The edge map, extracted by the Canny operator, is partitioned into blocks for a further comparison. Then, in the presence of a sketch query, the matching is computed combining the correlation response between the blocks of the database image and the blocks of the query [FSN<sup>+</sup>95]. Another different strategy is the one presented by Berretti that, instead of subdividing the edge map, adapted the part extraction to the shape. Thus, he encoded a shape by the composition of convex and concave parts characterized by their maximum curvature and their orientation [BDBP00] (see 2.3.3).

Automatize the manual segmentation of the database images is an extremely difficult task. Illumination effects, surface irregularities, the presence of textures, etc. make automatic segmentation an open field of research when dealing with general purpose images. Most of the systems that preprocess the database images in an unsupervised way use the shape features extracted from the boundary information. This is the case of one of the first approaches proposed by Del Bimbo [BP97] that used an elastic matching to perform the similarity between the sketch and the edges of a scene. The measure of the elastic deformation of the query sketch was also used by other CBIR systems such as ImageScape [Lew00]. The interesting point of this strategy is that the deformation of the input shape, can be corrected and adapted to the content of a scene. Nevertheless, the elastic deformation implies a slow procedure that cannot be performed image by image in a large collection of scenes.



This dependence on the individual matching of every database image is a common characteristic of some of the most successful and adaptable processes for sketch detection. As an example, the work of Ferrari et al. [FTG06] starts from the edge map and preprocess the contours by linking them at their discontinuities and partitioning them into roughly straight segments. The segments are connected according their sequential arrangement to several other ones and form a global complex branching structure. This structure, called network of segments, comprises all possible contour paths. The content of the query sketch is also segmented defining the contour chains that outline the query object. The detection problem is formulated as finding paths through the network which resemble the query chains. Even though this approach is able to detect sketches in high cluttered scenes, the matching step has to be done individually for each image and cannot be integrated in the indexation retrieval module. Nevertheless this kind of approach could be applied to evaluate the final matching of a subset of candidate images retrieved by a first filtering process according to other strategies. Generalized Hough Transform has been used in several approaches as a suitable strategy to perform a first spotting process to locate the query sketch inside a scene. Then, to assure a reliable location of the query, an alignment evaluation the both shapes is performed [ACS07] [GLMZ03].

In this chapter we present a CBIR system that uses a query-by-sketch paradigm. The aim of our proposal is to allow the retrieval of those cluttered scenes that contain an instance of the query sketch. Next, we detail the concrete implementation of the system.

## 5.2 A CBIR system based in geometrical constraints of a vectorial representation

We present a system that, given an sketch of an object, it is capable to detect it inside the set of database images. The system does not use any previous learning process and it is invariant to transformations of translation, rotation and scale. Moreover, it has both inexact matching and localization capabilities for a sketch-based retrieval problem.

We combine a descriptor based in the line approximation of the contour with a Hough like voting approach. To illustrate the performance, we have used a set of sketches consisting in architectural symbols to retrieve images from a collection of floor plans. Moreover we also present other examples that deal with real world scenes.

The retrieval process is applied in a sequential way to the images of the database providing a numerical evaluation of the matching of the query sketch inside a scene. From this numerical evaluation of all the database images, the system presents to the user the retrieved the scenes ordered from more to less reliability.

### 5.3 Description

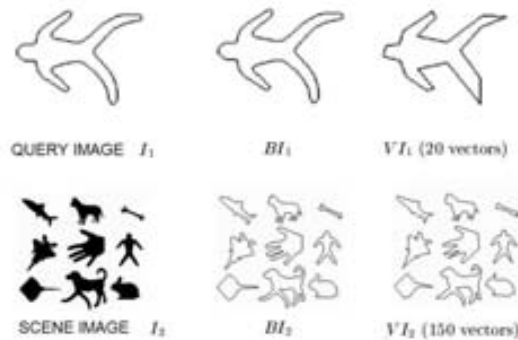
We have chosen a representation of the database images based on the boundary information. This representation allows us to match the inner information of a scene with an sketch than can be made of several strokes and open curves. Both the query sketch and the boundaries of the scene are represented by a polygonal approximation. Afterwards, they are encoded using geometric features in terms of predefined structures.

#### 5.3.1 Part extraction

The nature of the sketch information constraints the process of part extraction of the scene. Then, the process is also centered in obtaining a representation made of curves in order to compare them. When the database images are yet sketches, only a thinning preprocessing is applied to reduce the width of the strokes. Otherwise, when the scene is a clip art image or is an instance of the real world, we apply the Canny edge detector to obtain the edge information.

The curve information of the query and the scene are used to obtain the feature description. For an easier comparison of their content we transform them to have the same size. Given the information  $BI$  of an image (either a query sketch or a scene), we characterize it using a polygonal approximation in terms of segments. Then, we assign each one a reference orientation, thus we refer them as vectors instead of segments. We denote  $VI$  the collection of  $N$  vectors  $v$  that form the line approximation of the data  $BI$ .

$$VI = \{v_i\}, \quad i = 1..N$$



**Figure 5.2:** The information of the query and the scene is approximated by vectors

Since points belonging to a same segment likely belong to the same object [EZ96], we consider the vector entity as the basic part of an image. Then, we relate to every vector the features that encode the content of the image.

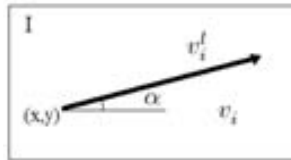
### 5.3.2 Feature description

The feature description of the query and the scene are obtained from the characteristics of their parts. We distinguish between two types of features: the *global features* and the *local features*. In one hand, the global features identify every vector of the image in a unique way according to the whole information and, in the other hand, the local features describe the image parts using every vector as a center of reference. Next we define them more deeply.

#### Global features

The global features  $GF$  of a vector  $v$  are composed by its length, angle and spatial coordinates (see Figure 5.3).

$$GF(v) = [v^l, v^\alpha, v^{(x,y)}]$$



**Figure 5.3:** Global features of an image vector  $v_i$

Notice that the  $GF$  contain the data that describes the scale, rotation and translation of a vector in an image. On the contrary, we want our system to be invariant to these three affine transformations. Thus, we focus on each individual vector to obtain the features that define each image part. We call these features: *local features*.

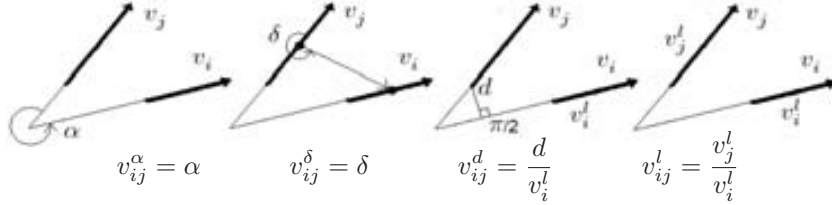
#### Local features

The local features are obtained for each image vector as the result of comparing the geometric properties of the surrounding vectors according to a set of predefined structures. Let us first introduce which are these properties that we call *pairwise features* and then detail the definition of the model structures that we call *primitives*.

- Pairwise features

We can characterize a vector  $v_j$  regarding to another vector  $v_i$  using the global features of both vectors. From them we can compute set of geometric properties such as the relative distance, the relative angle, the relative size, and the medium relative angle. For the sake of simplicity, we will denote  $v_{ij}$  the vector pair  $(v_i, v_j)$  and we define  $PF(v_{ij})$  the set of pairwise features for the vector  $v_j$  respect from the vector  $v_i$ .

$$PF(v_{ij}) = \{v_{ij}^d, v_{ij}^\alpha, v_{ij}^l, v_{ij}^\delta\}$$



**Figure 5.4:** Computation of the pairwise features of two vectors  $PF(v_{ij})$

The Figure 5.4 shows graphically the computation of the pairwise values.

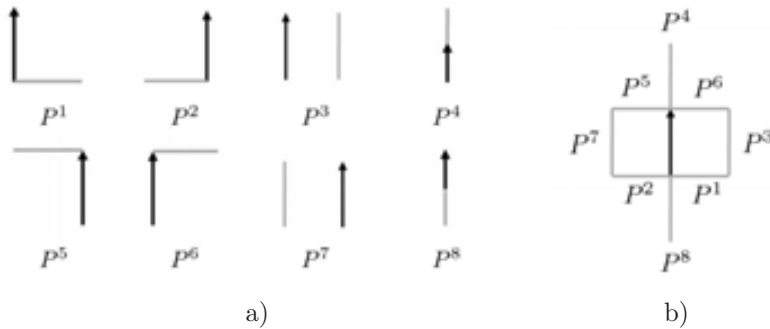
To obtain the local features, we combine these pairwise geometric properties in a higher abstraction level. In this process we use a set of predefined vector structures called primitives.

- Primitive structures

Many shape recognition strategies search for particular line arrangements according to perceptual grouping of salient features [ESB96] [SM92] [ESM<sup>+</sup>91]. In our case, we describe the relationships of perpendicularity, parallelism and co linearity due to several predefined structures that we call primitives.

We can define a primitive  $P$  as a particular arrangement of two vectors. We denote  $w$  instead of  $v$  the special vectors that form a primitive  $(w_{az}, w_r)$ , and we refer as  $\mathcal{P}$  a collection of primitives. In the Figure 5.5 we can observe the eight types that we consider.

$$P^z = (w_r, w_{az}) \quad \mathcal{P} = \{P^z\} \quad z = 1..N^{\mathcal{P}}$$

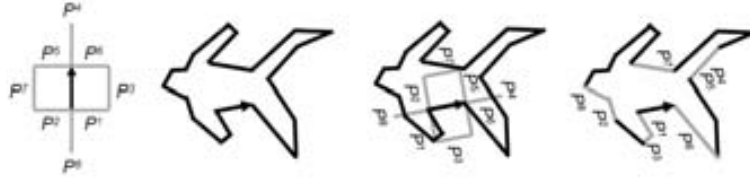


**Figure 5.5:** a) Primitive types composed by its vector  $w_a$  and the reference one  $w_r$  (the black vector) b) The whole set of primitives describe the relationships of perpendicularity, parallelism and co linearity according to a reference vector.

The computation of the local features comprises a process where we analyze the similarity between the pairwise features of the image vectors with the pairwise features

of the primitive vectors. The idea is easily shown in the Figure 5.6: to obtain the description of a certain image vector, we identify it with the reference vector of every primitive and we find the most similar vector of the image for every vector of the primitive structure. Informally speaking, the description of an image vector  $v_i$  due to a primitive  $P^z$  consists in answering the question:

'How good fits a primitive type  $P^z$  in the shape if we identify the primitive reference vector  $w_r$  with the image vector  $v_i$ ? If we identify  $v_i$  and  $w_r$ , which is the vector of the image that perform the best matching with the other primitive vector  $w_{az}$ ?'



**Figure 5.6:** The local description of a vector  $v$  is given by the minimum deformation that have to perform the primitive structures to fit the surrounding vectors

The local features of a vector  $v_i$  are defined by each primitive  $P^z$  and each pairwise feature and are denoted  $LF$ :

$$LF_i^z = [LF_i^{\alpha,z}, LF_i^{\delta,z}, LF_i^{d,z}, LF_i^{l,z}]$$

Then, a local description of the image around  $v_i$  is defined by 32 values provided by the 8 primitives and the 4 pairwise features types (relative angle, relative medium angle, relative distance and relative length)

Given a primitive  $P^z$ , the local features of a vector  $v_i$  are computed from the pairwise features of the primitive  $PF(w_{ra^z})$  and the pairwise features of the vector  $v_i$  regarding another image vector  $v_j$ . This other vector  $v_j$  is chosen as the one that minimizes a certain cost function  $D$ .

Next we will see how the local features are computed and then we will expose which is the formulation of the cost function  $D$ . Thus, we have developed a set of distance function  $d$  that evaluate the similarity between a pairwise features  $PF(v_{ij})$   $PF(w_{ra})$ . Next we denote its formulation and then we detail how they are computed.

$$LF_i^{\alpha,z} = d^\alpha(v_{ij}^\alpha, w_{ra^z}^\alpha)$$

$$LF_i^{\delta,z} = d^\delta(v_{ij}^\delta, w_{ra^z}^\delta)$$

$$LF_i^{l,z} = d^l(v_{ij}^l, w_{ra^z}^l)$$

$$LF_i^{d,z} = d^d(v_{ij}^d, w_{ra^z}^d)$$

where  $v_j$  minimizes the function  $D \forall v_j \in VI$ :

$$D(v_i, VI, P^z) = \max(|d^\alpha(v_{ij}^\alpha, w_{ra^z}^\alpha)|, |d^\delta(v_{ij}^\delta, w_{ra^z}^\delta)|, |d^l(v_{ij}^l, w_{ra^z}^l)|, |d^d(v_{ij}^d, w_{ra^z}^d)|)$$

The information represented by the local features can be understood as the distortion that the primitives experiment when they fit into each image zone. The election of the  $v_j$  that minimizes the function  $D$  searches for the minimum distortion according to the geometric properties of the primitives  $PF(w_{ra})$ .

Next we detail the computation of every single function  $d$  involved in the calculus of  $D$ . For every feature we have developed a distance measure that comprises the range  $[-1, 1]$ . This indicates how different is a geometric property of a vector pair  $v_{ij}$  respect the one of the primitive pair  $w_{ra}$ . When the distance is 0 it indicates that are identical, when it is close to -1 it indicates that the value of  $v_{ij}$  is lower than the value of  $w_{ra}$ , and viceversa, when it is close to 1 indicates that is higher.

The general idea of the computation consists in finding the difference of both features and normalize it by the value of their maximum feasible variation. The functions follow two kind of strategies according if the computation is related to angular values ( $d^\alpha$  and  $d^\delta$ ) or distances ( $d^d$  and  $d^l$ ).

- $d^\alpha$  : Distance of the pairwise angle

$$\Delta^\alpha(v_{ij}^\alpha, w_{ra^z}^\alpha) = \frac{\min(2\pi - |w_{ra^z}^\alpha - v_{ij}^\alpha|, |w_{ra^z}^\alpha - v_{ij}^\alpha|)}{\pi}$$

$$d^\alpha(v_{ij}^\alpha, w_{ra^z}^\alpha) = \begin{cases} -\Delta^\alpha(v_{ij}^\alpha, w_{ra^z}^\alpha) & \text{if } w_{ra^z}^\alpha < v_{ij}^\alpha \\ \Delta^\alpha(v_{ij}^\alpha, w_{ra^z}^\alpha) & \text{if } w_{ra^z}^\alpha \geq v_{ij}^\alpha \end{cases}$$

- $d^\delta$ : Distance of the pairwise medium angle

This function is computed the same way as  $d^\alpha$  but replacing the values of  $v_{ij}^\alpha$  for  $v_{ij}^\delta$  and  $w_{ra^z}^\alpha$  for  $w_{ra^z}^\delta$ .

- $d^l$ : Distance of relative length

Unlike the angles, the relative values of length have no limited ranges. For this reason we have to establish an upper and lower threshold in order to obtain a comparative value. These limits are obtained due to a tolerance coefficient  $\lambda_l$ .

$$thr_u^l = v_{ij}^l * \lambda_l \quad thr_l^l = \frac{v_{ij}^l}{\lambda_l} \quad \Delta^l(v_{ij}, v_{bm}) = \frac{|v_{ij}^l - w_{ra^z}^l|}{\lambda_l}$$

$$d^l(v_{ij}, v_{bm}) = \begin{cases} -1 & \text{if } w_{ra^z}^l < thr_l^l \\ -\Delta^l(v_{ij}, v_{bm}) & \text{if } w_{ra^z}^l < v_{ij}^l \\ \Delta^l(v_{ij}, v_{bm}) & \text{if } w_{ra^z}^l \geq Rv_{ij}^l \\ 1 & \text{if } w_{ra^z}^l > thr_u^l \end{cases}$$

- $d^d$ : Distance of relative distance

This function is analogous to  $d^l$  replacing the values of  $v_{ij}^l$  for  $v_{ij}^d$  and  $w_{raz}^l$  for  $w_{raz}^d$ . It also uses a bounding parameter  $\lambda_d$ .

Until this point we have presented the description of every image vector using two kind of features, the global features  $GF$  and the local ones  $LF$ . Next we expose how these values are organized in the indexing structures and how the matching of the query sketch and the scene is performed.

## 5.4 Matching

The content of the image can be described as a collection of parts that are characterized with a vector of global features and a vector of local features. The indexation of the information of the database images is done by the values of these vectors. Then, given a query sketch, it is located in a scene by a voting procedure on the spatial domain. Finally, an alignment evaluation provides a probability value to rank de database image in the retrieval result.

### 5.4.1 Feature indexing

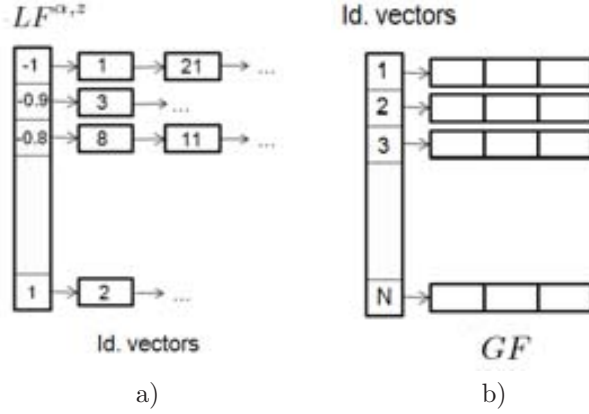
The system has to store the information related to the vectors of the scene images in order to allow the matching with the query ones. Two kind of tables are used to store the necessary information:

#### Indexing tables

The first type of table indexes the image parts from their local features  $LF$ . The local features are a descriptor of 32 values (4 relative distances x 8 primitives) that are in the range  $[-1,1]$ . Since the values are bounded and the number of features is not too high, we can construct 32 of tables to index the image vectors. We have discretized the range in 20 buckets that go from -1 and until 1 with a steps of 0.1. These value are used primary keys to retrieve the vectors of the scene that have similar local descriptions as the ones of the query sketch.

Moreover we are also interested in keeping the information related the global features. Then, we can maintain a simple table where every row refers to an image part and that stores these values.

Given the description of the vector of the query, we can access to the vectors that have similar local descriptors. Moreover we can also recover the global information represented by the absolute features. The figure 5.7 we illustrate the idea of the indexation tables.



**Figure 5.7:** Indexing tables. a) Index of the vectors identifier according to their local features. b) Index of the global features according to the vector identifiers.

### Descriptors similarity

To evaluate the similarity of two features, we treat in an independent way the values related to every type of primitive. These partial similarities are computed using the Chebyshev distance and are normalized in the range  $[0,1]$ . Finally all the values are averaged to obtain the final result.

$$D(LF_i(I_1), LF_j(I_2)) = \frac{\sum_{z=1}^{N^P} \max(|LF_i^z(I_1) - LF_j^z(I_2)|)}{2 * N^P}$$

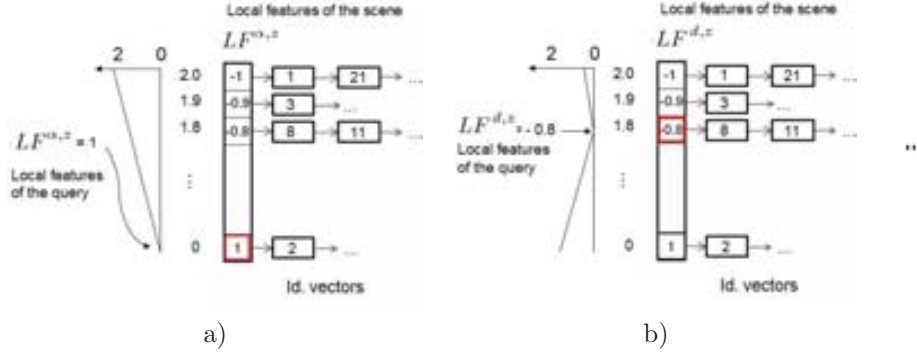
Notice that the Chebyshev distance is equivalent to access the tables of every local feature and assign a ramp-like-value to the indexed vectors. Then, from the values assigned to every vector we take the maximum. The Figure 5.8 illustrates this idea.

Function  $D$  compares the similarity of two descriptors that are associated to a vector of the query and a vector of the scene. Nevertheless, in the description phase we have added a random orientation to the lines of the image in order to treat them as vectors. Then, we can be interested in comparing the similarity of two vectors to evaluate if they match in the opposite orientation. Notice that the primitives 5 to 8 can be seen as a variation of the primitives 1 to 4 where the relative vector  $w_r$  has opposite orientation (see Figure 5.5). Then the similarity computation of two descriptors can be easily be compared despite of the orientation if we check both matchings by swapping the primitives.

We can reformulate the function  $D$  adding a parameter  $O$  that refers to the matching orientation (where  $O=1$  means equal orientation and  $O=0$  means opposite orientation).

$$D(LF_i(I_1), LF_j(I_2), O) = \frac{\sum_{z=1}^{N^P} \max(|LF_i^{z^O}(I_1) - LF_j^z(I_2)|)}{2 * N^P}$$





**Figure 5.8:** Fuzzy similarity value assigned to the vectors of the scene according to the local features of the query. It can be represented as a ramp-like function. a) Example of the indexing in the table related to  $LF^{\alpha,z}$  b) Example on the table of the  $LF^{d,z}$  values.

$$z_O = \begin{cases} z & \text{if } O == 1 \\ \text{mod}(z + \frac{N^P}{2} - 1, N^P) + 1 & \text{otherwise} \end{cases}$$

For every vector of the query we can find the most similar vectors of the scene. Then we apply an evaluation phase to check if their spatial arrangement is also consistent.

### 5.4.2 Evaluation

Given the features of the query and the scene, we start a process of finding the location where one matches into the other. We propose to use a modified version of the Generalized Hough Transform (GHT) (see section 2.5.2) to deal with vectors instead of points. Moreover we introduce an additional step that evaluates the matching with an alignment process based on the contours.

#### The voting process

The sketch matching involves a voting procedure in the spatial domain inspired in the generalized Hough transform [LKW][GLMZ03] [ACS07]. The process uses a reference point  $rp$  in the query image that is used to locate the query sketch in the scene.

For every vector of the query we search in the indexing structures for the similar vector of the scene using the two orientations. We define a vote  $h_{ijO}$  as the hypothesis of the local matching of the vector  $v_i$  belonging to  $I_1$  with the vector  $v_j$  belonging to  $I_2$  in the orientation  $O$ . The process generates  $N_1 \times N_2 \times 2$  votes that are accumulated in a voting space  $H$ .  $H$  has the same dimensions as the scene and accumulates the evidences of the location of the query sketch inside the scene.

Every vote has a specific weight  $W$  that is proportional to the similarity of the local features. Then, the distance  $D(LF_i(I_1), LF_j(I_2), O)$  is used as the weight value

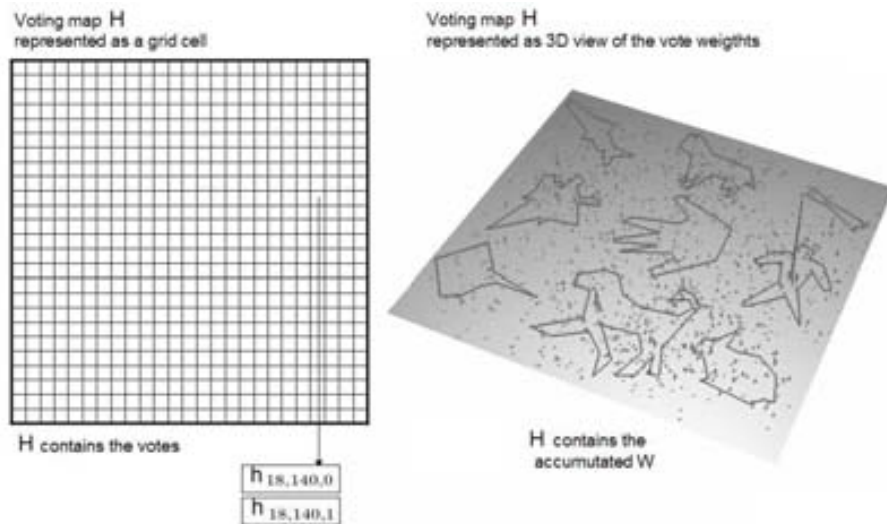
of a vote  $h_{ijO}$

The location of the vote inside  $H$  is found by the transformation of the reference point  $rp$  when we match  $v_i$  and  $v_j$ . This transformation is computed from the information of the position, scale and orientation of both vectors. This information correspond to the global features  $L(GF_i(I_1), GF_j(I_2), O)$ .



**Figure 5.9:** Example of the location of the reference point for the votes  $h_{18,140,1}$  and  $h_{18,140,0}$

The map  $H$ , viewed in a 3D representation, shows as peaks the locations where the query shape is more probably located (see the example Figure 5.10). Because the query sketch can present some distortion, the votes can be spread in the voting map  $H$ . Thus, we apply a Gaussian filter in order to enforce the weight of those votes that have also other neighboring votes with a high weight.



**Figure 5.10:** The map  $H$  contains the votes and their accumulated weights.

Then we proceed to validate the shape detection on those positions such  $H(x, y)$  exceeds a certain confidence value  $Thr_H$ .

### The alignment process

The accumulation of votes in the map  $H$  generate the evidence of finding the query sketch inside the scene. Then, we validate its coherence using an alignment process on the original boundary information  $BI_1$  and  $BI_2$ .

The shape alignment is a very intuitive strategy of pattern matching. The process only needs two pairs of points: one pair belonging to the query image and the other belonging to the scene. In our case, given an hypothesis  $h_{ijO}$ , the alignment points are defined by the initial and final coordinates of  $v_i$  and  $v_j$ . Then, the information of the query suffers a set of transformations of translation, rotation and scale according to the pair of vectors of the vote  $h_{ijO}$ . The information of transformed sketch named  $BI'_1$  is compared against the information of the scene  $BI_2$ .

To evaluate the goodness of the sketch matching we have adapted the computation of the Chamfer distance to the requirements of our system. We have chosen the Chamfer distance since it is efficient to compute and, unlike the Hausdorff distance, it tolerates a considerable degree of pattern misalignment [FH05][SST06] [HKR93]. The original Chamfer distance is computed as the mean of the nearest distance values from the points of the sketch template to the points to a contour scene. To make the computation, a distance map of the contours of the scene  $BI_2$  can be generated in linear time [Gav98]. Then, the values overlapped by the pixels of the query  $BI'_1$  can be averaged.

Even though the Chamfer computation is fast and efficient, it suffers of two main weaknesses. The first one is that the Chamfer computation is not scale invariant. Then, the matching is not independent on the size in which the query is present in the scene. Moreover, the second need of the Chamfer distance is that it can lead to undesired results when it deals with high cluttered images. Being a scene plenty of edges, every pixel of the query sketch will have a high probability of being close to a boundary element of the scene. Then, the Chamfer distance would provide a high score and would lead us to detect false positives of the query template.

In order to solve these drawbacks, we have applied some modifications to the original Chamfer distance. Then, to provide scale invariance we have used the information of the vectors that provide the alignment. As we have introduced before, the alignment of the sketch is done according the identification of two vectors,  $v_i$  and  $v_j$ , one belonging to the query and the other to the scene. Because we know which is the scale transformation that  $v_i$  suffers to be aligned with  $v_j$ , we can weigh the results of Chamfer distance in consequence.

To deal with the presence of clutter, some works use the edge orientation information. The proposal of Gavrilu [Gav98] applies this idea by dividing both boundary images,  $BI_1$  and  $BI_2$  into discrete orientation channels and summing the individual Chamfer scores. However, this option is sensible to the quantization of the angular information. Then, to avoid the discretization effects, Shotton [SBC05] proposes to

incorporate an explicit cost for orientation mismatch given by the mean difference in orientation between the two edge patterns. Then, we have applied a similar strategy that uses the angular matching from the normal vector of the contour information. Thus, we gain robustness in the alignment evaluation because in addition to the proximity requirement we analyze the orientation coherency.

The main idea of the our alignment process can be resumed as follows: we are interested in transforming the boundary information of the query to align it with the boundary information of the scene and obtain the maximum number of query pixels close to the scene pixels that have similar angular information. Then, we have formulated a computation using two kind of costs: one for the distance match and the other for the orientation match.

Let us name  $T$  the function that assigns the probability  $[0,1]$  of matching of the query into the scene according to a the transformation of a vote  $h_{ijO}$ . Its computation, illustrated in the figure 5.11, uses a function  $t^D$  that measures the distance information and a function  $t^A$  that evaluates the angle of the normal vector of the contour curve.

$t^D$  benefits the pairs of points that are closer than a distance  $Thr_d$ . The threshold  $Thr_d$  can be tuned according to the degree of accuracy that we want to archive. In a similar way,  $t^A$  benefits those points with an angular difference lower than a value  $Thr_a$  but also penalizes them if the difference is higher. Since the maximal angular difference is  $\pi/2$ , we have fixed the threshold  $Thr_a$  to  $\pi/4$ . The parameter that weights the scale transformation of the matching is called  $\alpha_s$ . It is computed using the ratio of the vector lengths.

$$T(m_{ijO}, BI'_1, BI_2) = 0.5 + \frac{\sum t^A(p_1, p_2) * t^D(p_1, p_2)}{2 * (\#BI'_1)} \quad \forall (x, y) \mid p_1(x, y) \in BI'_1$$

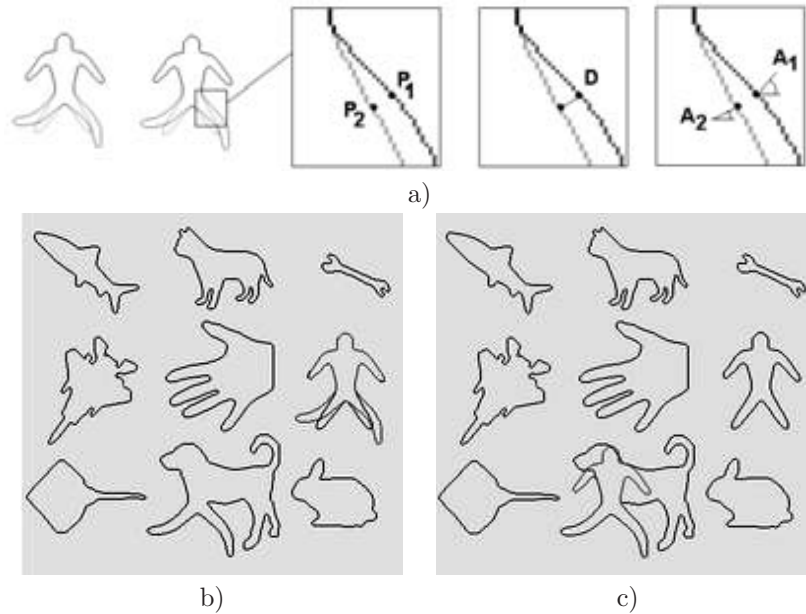
being  $p_2$  the point of  $\in BI_2$  that minimizes the Euclidean distance  $|p_1 - p_2|$

$$t^A(x, y) = 1 - \frac{|A(p_1) - A(p_2)|}{\frac{\pi}{4}}$$

$$t^D(x, y) = \frac{Thr_d - \min(D(p_1, p_2) * \alpha_s, Thr_d)}{Thr_d} \quad \text{where} \quad \alpha_s = \min(1, \frac{v_i^l}{v_j^l})$$

Most of the computation related to the database images can be done off-line. Then, for every scene we can pre-compute the distance map and a map containing the angular information of the closest contour point. Then, in the online matching we only need to compute the information of the query contour.

We perform the computation of the matching function  $T$  on the set of the best votes. We use the maximum alignment result of all the hypothesis as a measure to rank the database images in the retrieval process.



**Figure 5.11:** a) Representation of the alignment evaluation that combines the distance and the angular values. b) c) Alignment on related to the votes belonging to the peaks of  $H$ . The detection probability is 0.84 and 0.75 respectively.

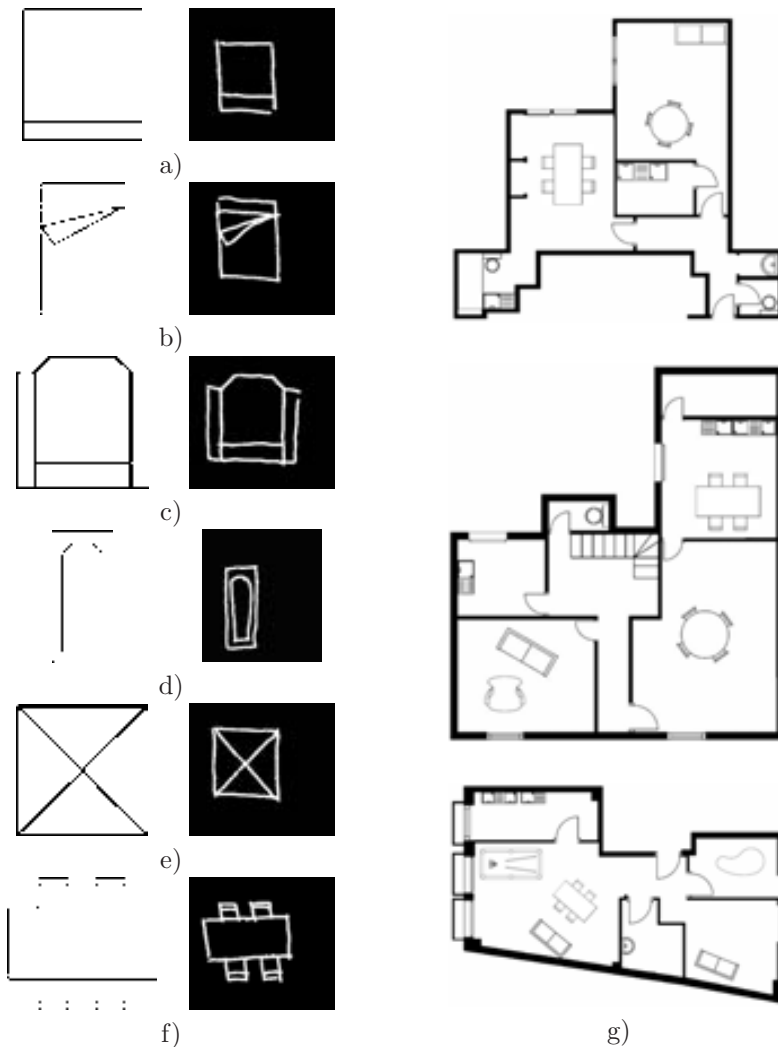
## 5.5 Experiments and Results

We have applied our proposal to a database of images composed by a collection of architectural plans. Next we present some results on this application and we also show other examples using images of other contexts.

### 5.5.1 Application on architectural plans

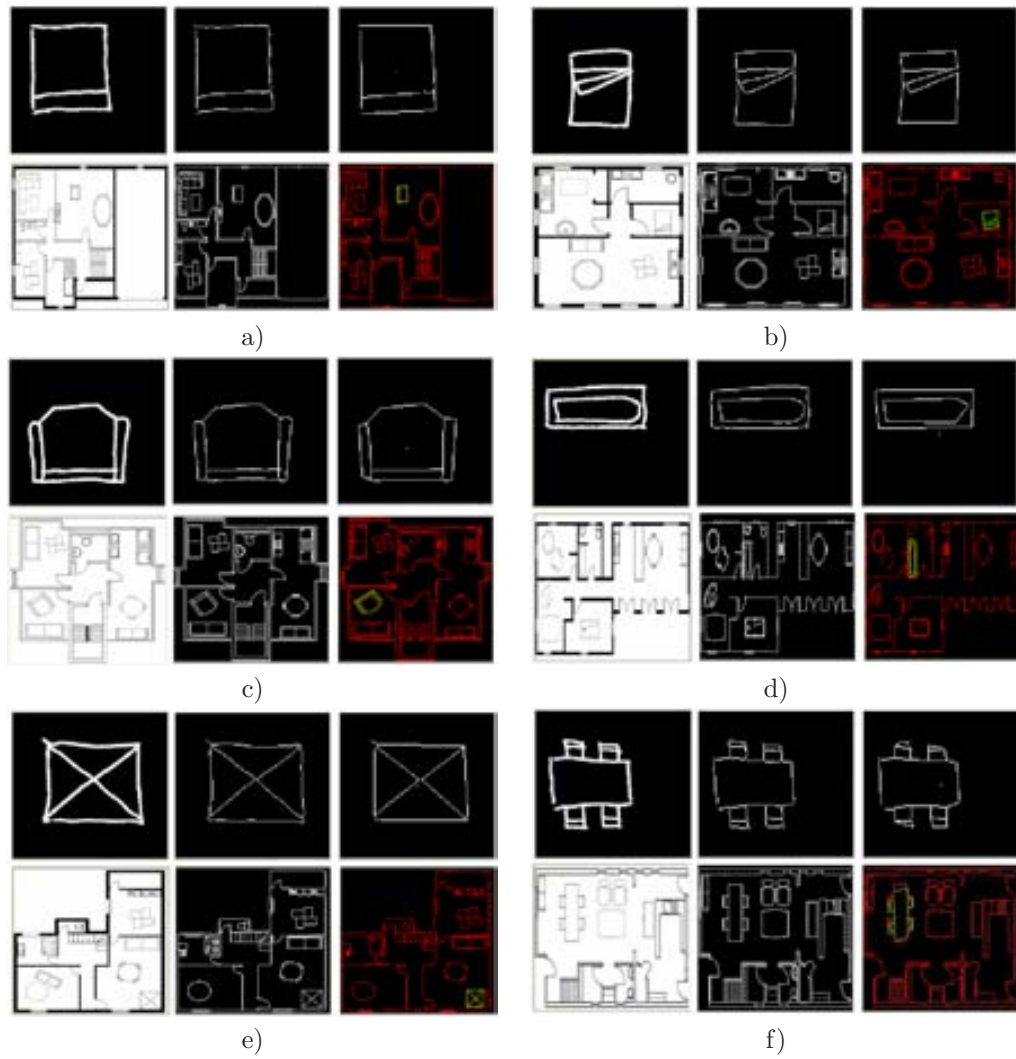
Using a database of architectural plans, the user can provide a query that consists in the sketch of a symbol: a table, an armchair, etc. Then, the system performs a matching of the query in the architectural plans and can retrieve those that contain an instance of the searched symbol.

We have evaluated the performance with a set of 240 database images and 6 symbols of interest. Then, 8 users have provided 11 sketches for every symbol generating a test set of 88 queries per symbol and a total amount of 704 queries. The ground truth of the experiment is formed by 6 classes of architectural plans, with 40 images having an instance of one of the symbols. The Figure 5.12 shows some examples of the database images, the original symbols and the user sketches. Moreover, the Figure 5.13 shows some examples of the detection of the sketched symbols in the plans.

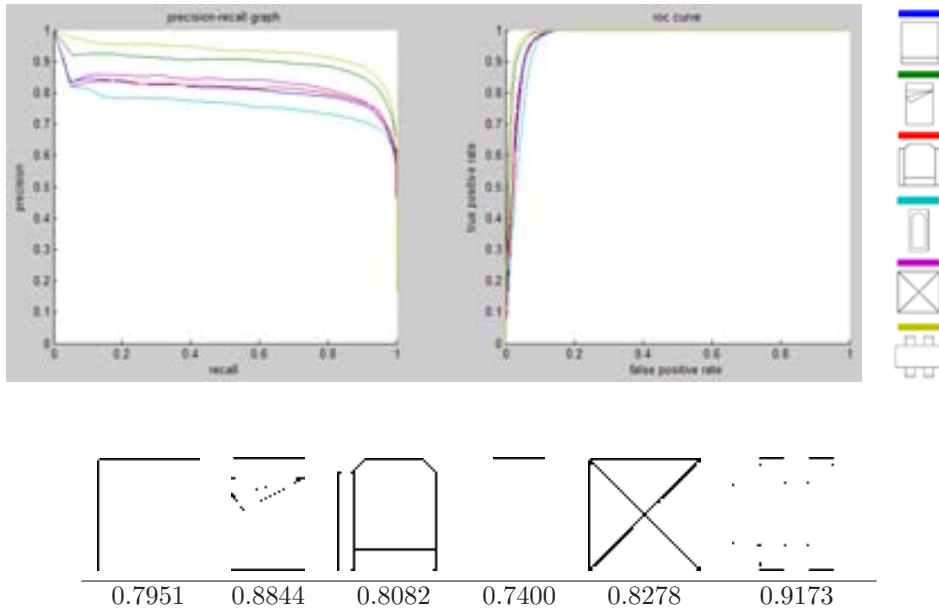


**Figure 5.12:** a-f) Original symbols and one example of sketch instance. g) Example of 3 database images containing the symbol f).

From the retrieval results of each sketch class we have computed the graphs of Precision-Recall that are illustrated in the Figure 5.14. Analyzing the results, we have observed that almost all types of symbols show a good performance. The mean of the precision and recall values for each sketch class on the first 40 images (the number of ground truth images) oscillate between 0.74 and 0.91. This means that in the worst case, when we retrieve the same amount of images as the ground truth, we obtain a 74% of correct results.



**Figure 5.13:** Examples of the sketch detection. a-f) Matching probabilities  $T$ : 0.75, 0.74, 0.78, 0.73, 0.72, 0.65.



**Figure 5.14:** Precision-recall graph and ROC curves for each class of sketch. On the bottom, mean of the precision and recall values for the sketches of each symbol retrieving the first 40 images (the number of ground truth images).

We have also analyzed the problems of the retrieval process and we have found out the main following causes (see Figure 5.15):

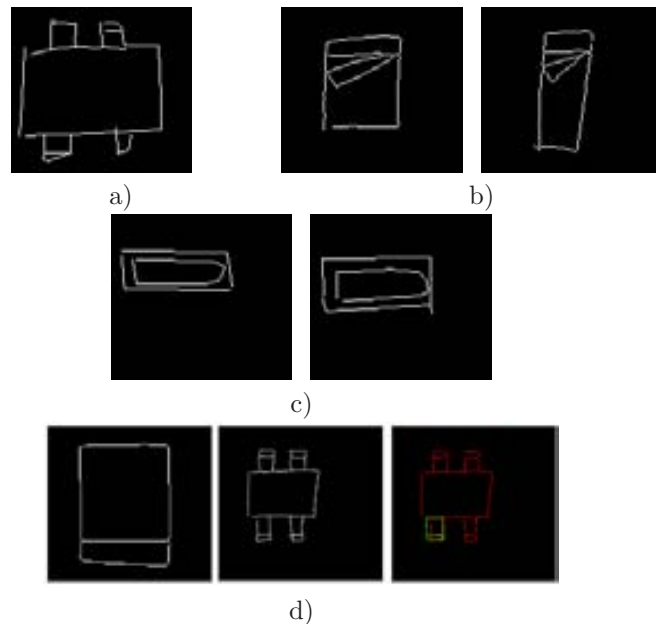
Line approximation Each line has associated a local descriptor that is used to index the symbol. Thus, the line approximation is an important processing step that directly affects to the description phase. We observe that the problems caused by a defective line approximation become more severe as fewer parts form the query sketch.

- Partial noisy approximation Because the object parts vote in an independent in the matching process, a defective line approximation in a part of the symbol can also be solved by the correct detection of the other parts. Then it is natural that the symbols that are more complex and are composed of more lines, such as the symbol f) have better performance than the simpler ones.
- Global deformation An altered representation of a sketch can affect the retrieval result. Nevertheless the deformation can be outbalanced by increasing the tolerances of the steps regarding to the matching module of the CBIR. In the first phase, relaxing the retrieval of the indexed descriptors we can retrieve sketch parts that have suffered deformations. Moreover, increasing the ratio of vote-clustering can help in detecting the presence of the sketch parts.
- Polygonal approximation of curves Also the curved components, such as the part of the symbol d), can be unstable to the polygonal approximation. This



can be caused by the inherent deformation in the drawing style and can be aggravated by changes in the scale representation. Curves with different sizes can be represented by a different amount of vectors and its descriptors can be different.

Sub-part detection Another different focus of problems can be given by the sub partial matching capabilities according to the symbols of interest. This is the case of the symbol of the class a) that can be easily detected as a part of the symbol f) and even b) or and c). This effect is a logical consequence of allowing a partial matching of the scene elements.



**Figure 5.15:** Examples of problems in the sketch detection caused by: a) A partial noisy approximation that provokes some parts of the symbol to be lost. b) A global deformation of the symbol. c) Intestable vectorization of the curves. d) Subpart detection of a symbol.

### 5.5.2 Examples in general contexts

Next we present some other examples of the sketch matching in images of general purpose that are not related to any particular application. In the images we see the original information of the query and the scene, the boundary information and the vectorial approximation with the matching location (see Figures 5.16, 5.17, 5.18, 5.19, 5.20).



**Figure 5.16:** Example of sketch detection: star. From top to down, the  $T$  values of the matching are: 0.63, 0.70, 0.64, 0.69



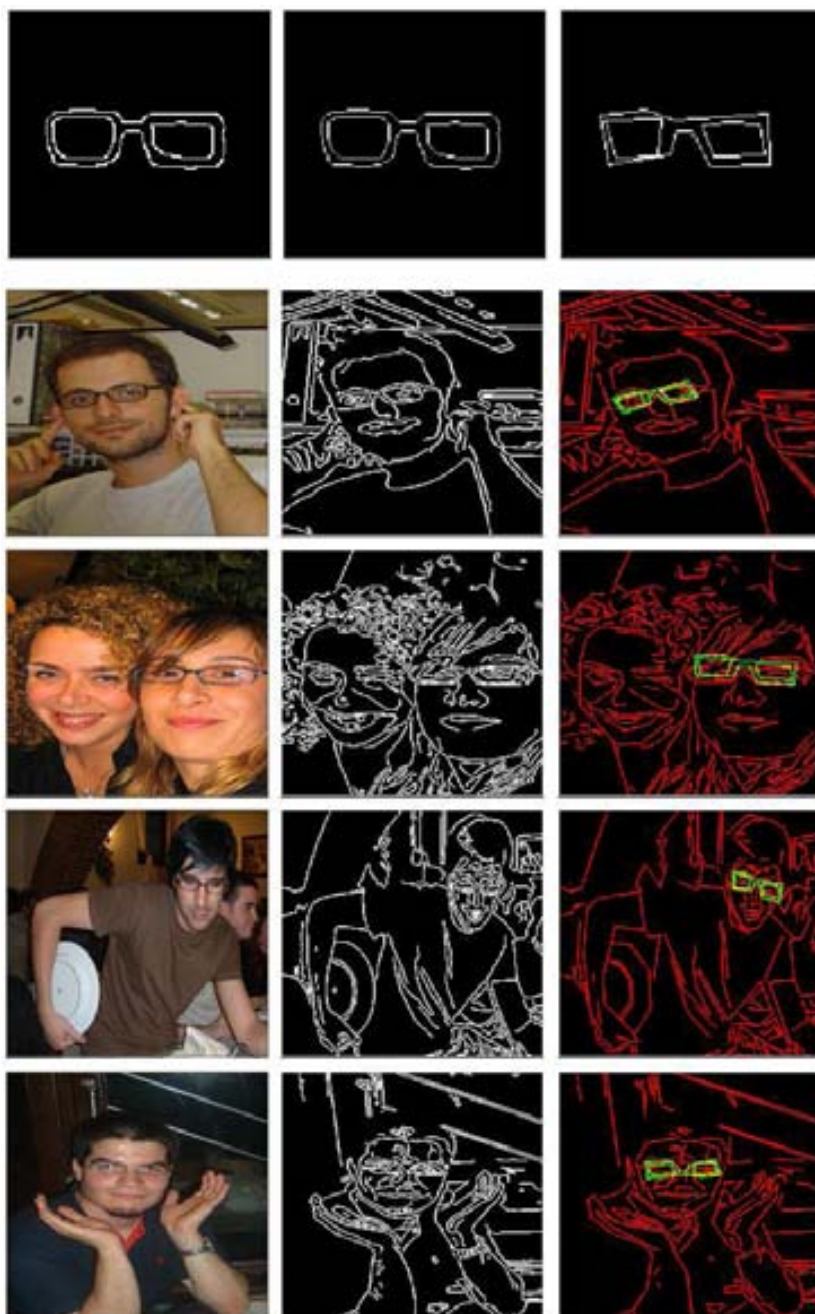
**Figure 5.17:** Example of sketch detection: car logo. From top to down, the  $T$  values of the matching are: 0.77, 0.75, 0.75, 0.68.



Figure 5.18: Example of sketch detection: alert sign. From top to down, the  $T$  values of the matching are: 0.69, 0.70, 0.76, 0.72

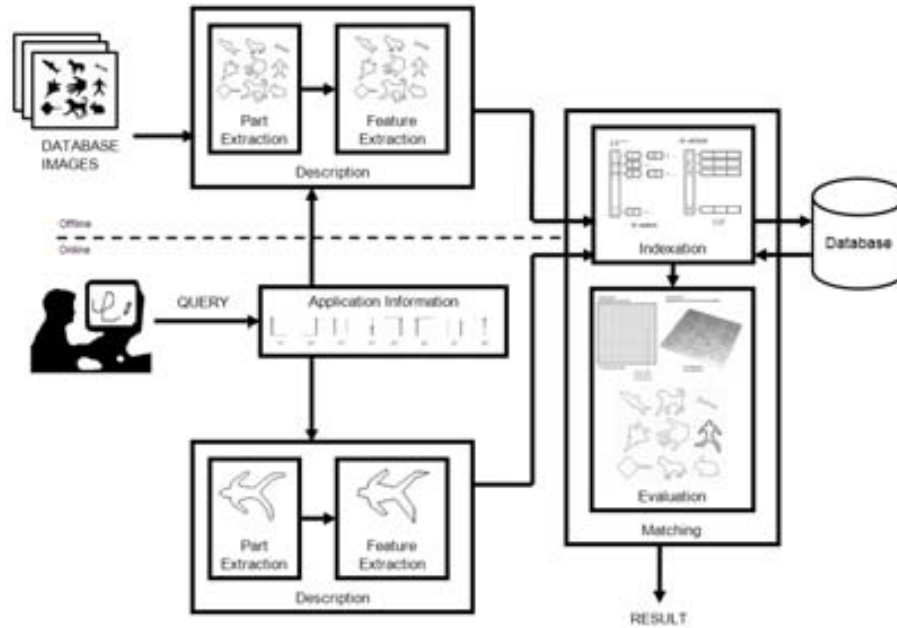


**Figure 5.19:** Example of sketch detection: catalan donkey. From top to down, the  $T$  values of the matching are: 0.61, 0.63, 0.64, 0.54.



**Figure 5.20:** Example of sketch detection: glasses. From top to down, the  $T$  values of the matching are: 0.70, 0.66, 0.63, 0.68.

## 5.6 Discussion and Conclusions



**Figure 5.21:** Visual resume of the modules of our approach. Part extraction: contour vectors. Features: geometric relations between the vectors according to the primitives. Feature Indexation: tables of local and global features. Part Matching: Hough-like voting and alignment by oriented Chamfer distance.

### 5.6.1 Conclusions of our approach

We have developed a retrieval system which aim is to detect a query sketch as a part of a scene. The strategy is invariant to affine model transformations such as scaling, rotation and translation. It also tolerates slightly deformations and partial occlusion. The process deals with the boundary information of the scenes and it is able to manage open curves and detect wiry objects.

We have developed a descriptor based in the geometric features of the polygonal approximation of the edges. The description of the database scenes can be done off-line because it is independent of the query and can be applied to general purpose images.

As we have seen in the examples, the system can be used in a wide variety of collections. We have performed a test using a database of architectural plans where the images can be retrieved according the sketch of the symbols that they contain.

Nevertheless, some limitations are still found in the proposed approach. The main one is the use of vectorial information that, even though it suits most of the man-made objects, do not deal properly with deformable elements composed of curved parts. Nevertheless, our approach is still more flexible than other shape descriptors that also use vectorial information. Some approaches that encode the relation of consecutive segments can be very sensitive to breaking of the boundary information [MG95] [SM92]. On the contrary, we deal with independent vectors that adapt to a set of primitive structures. This is also an advantage in comparison to other sketch descriptors that use a relation of neighborhood based in the proximity of the segments [HH98]. This relation of proximity leads to develop a local description that includes little segments that, even though they can be close are the results of a noisy polygonalization of the image.

Moreover some improvements could be done in the phases of indexation and evaluation. We have implemented our proposal using an image per image matching but this could be automatized to work with all the images of the database at the same time. The improvement implies to join the descriptors of all the individual hash tables of the image scenes. Then, given the features of a part of a query sketch, the system could access to all the scenes with a similar local description. Using this strategy, the voting map would also use an auxiliary dimension related to the database image identification.

A further study could evaluate the benefits of treating in a separate manner the transformations of scale and rotation. This could be done by adding two additional dimensions to the voting map related to them. In one hand, this could reduce the false positives in the accumulator space but, in the other hand, it would imply a more complex process to detect the candidates of the final evaluation.

### 5.6.2 Conclusions of the query-by-sketch paradigm

The two special properties of a sketch are the simplification of the image content and the intrinsic deformation provided by the drawing process. These characteristics are fairly what makes the sketch-based-retrieval a challenging task.

A query-by-sketch suits the applications where the content of the images are also restricted to sketches, or technical drawings or even to images containing instances of wiry objects (glasses, bikes, etc.). Nevertheless, as the information contained in the database images becomes more complex, the matching of the sketch presents important difficulties. The presence of a cluttered background implies the difficulty of distinguish what is the part of the object we search for and what belongs to the rest of the scene. Moreover the presence of natural textures adds a high degree of noise in the phase of extracting the relevant information of a scene.

Most of the systems opt by working with edge detection processed to extract the image parts. Otherwise, processes based on region segmentation processes use to be



avoided when dealing with a generic image collections. When the system does not have any external knowledge of the possible queries, it is a hard task to obtain a reliable segmentation of the shape of the scene objects.

The matching phase involves difficulties due to the deformation of the sketch representation. Inexact matching strategies have to be applied that in most of the cases need to be evaluate the sketch matching for the database scenes in a sequential way.

The query formulation of this paradigm is more elaborated than the query formulation of the query-by-selection and query-by-iconic composition. Hence, the user has a higher degree of freedom in order to create the query according its needs. Nevertheless, the sketch query is mainly restricted to the shape information and cannot incorporate other perceptual cues such as the color. Color that can be a powerful descriptor to retrieve pictures of certain types of objects, by instance, the flag of a county, a concrete art painting, the picture of certain animal (e,g, a bee), etc. Thus, some systems add painting tools for to provide queries that involve color information as well as shape information. In the next chapter we focus our attention in this kind of query paradigm that we call *Query-by-Paint*.



# Chapter 6

## Query-by-Paint

### 6.1 Introduction

Color features are present in most of the visual data taken from the real world. Nowadays, all the conventional capture devices deal with color images and most of the images available in internet are also in color. Studies of human perception state the overall color of the images is an important factor in judging similarity [RFS<sup>+</sup>98]. Consequently, CBIR have also paid an special attention to this visual feature. Some systems include a set of tools to develop a query formulation called query-by-paint. Thus, the system interfaces present an empty canvas where the user can paint his own composition of color regions. Then this image is further used to match the most similar scenes in the database.

One of the most critical points of the query-by-paint retrieval is the part image decomposition of the database scenes. Hence, QBIC [FSN<sup>+</sup>95], which is probably the best-known system using this kind of queries, do not support fully automatic part extraction. When the objects of interest do not lay on a plain background, a manual segmentation is performed with the help of snakes and flood-fill operations. QBIC was the first commercial CBIR system and it is actually installed in the website of the Hermitage Museum <sup>1</sup>. The user can search for an art work of the painting collection by creating a color region composition. The query is painted on a virtual canvas using a color palette and geometrical templates of ellipses and rectangles. Figure 6.1 illustrates a query on this online system.

Nevertheless, we can find other engines using query-by-paint that do have set on the problem of automatic part extraction. From the simplest to the most complex, the approaches to solve this problem include the indivisible image encoding, the image quantization, or a multi-layered segmentation.

DrawSearch [SSM99] is an example of a CBIR that characterizes the color distri-

---

<sup>1</sup><http://hermitagemuseum.org>



**Figure 6.1:** Example query of QBIC system on the Hermitage Museum Database

bution of the whole scene. An image is codified as an histogram of the region areas obtained from coarse uniform quantization of the RGB color space. Moreover, the WBIIS system [WWFW97] also uses a compact representation but avoids the color quantization effect using a descriptor based on the coefficients of a wavelet transform. WBIIS system only allows partial matching by recomputing the image descriptors of the database. The painting interface interprets that the user do not cares about the image content of the canvas zones where he has not drawn any color region. For example, if a user wants to find all images with a racing car of any color in the center of an image, the user may simply form a query with a central empty area. Then, these empty zones are taken as masks that have to be applied to the database scenes before recomputing their description.

In order to improve the partial query performance, some strategies divide the image content into regions. Visualeek [SfC96] defines a set of 166 colors in the HSV space and extract the regions using a back projection strategy. For every region the shape is described by the area, the centroid and the bounding box. The matching between the query regions and the database ones is done by minimizing the distances of the color and shape features.

Finally, the Picasso system [BMPT98] supports queries based on shape, color regions and their spatial relationships. The system exploits a pyramidal color segmentation where each level of the pyramid corresponds to a resolution level of the segmentation. The image is represented as a multi-layered graph with relations of the adjacent regions in the same level and between the hierarchy of levels. The description of a color region include: a binary 187-dimensional color vector, the position of the region centroid, the region area and the region shape approximated by the axis of its best fit ellipse. Moreover, spatial relationships between regions are represented using 2D strings. In the querying process, the pyramidal structure of each candidate image is analyzed from top to bottom to find the best matching region for each query

region. The matching score between a query region and a candidate image region is given by a weighted sum of distances between the computed region attributes (color, region centroid's position, area and shape). The similarity score between the query image and a candidate image is obtained by summing all the scores of the matched query regions. In a similar way to QBIC, Picasso engine is used by Alinari Archives in Florence for the electronic cataloging of paintings and sculptures of Italian museums and galleries.



**Figure 6.2:** Interfaces of the systems a) DrawSearch b) Visualseek c) Picasso

The popularity of the retrieval systems using query-by-paint reached its maximum popularity at the beginning of the last decade. Nevertheless due to its inherent complexity they have been devoted to research prototypes and, exceptionally, to retrieval images of art collections. Nevertheless, a more modern application has been recently proposed by Watai et al. [WYA07]. They propose a system to retrieve web pages according to their visual appearance. Using color signature features and Earth-Mover's Distance, the system compares the query composition with the designs of the pages in the user's local web browsing history.

In this chapter we have studied the problem of retrieving images using a query-by-paint approach. Inspired in the human perception, we have studied the most suitable approach to obtain the relevant information of the images and represent it.

## 6.2 A CBIR system based in the principles of the human perception

The goal of our work is to develop a retrieval system that is able to retrieve complex scenes containing an instance of a query object. This object is represented by the user as a composition of color regions.

Our proposal represents the parts of the objects as image regions. We have developed a region detection algorithm inspired in the human perception. Moreover, we have tested several descriptors in order to find the most suitable ones to match the

information of a real scene with the information of a painted object.

The image parts are described with features regarding to the color and the shape. Thus, the database of the system has to store large amounts of multidimensional vectors. To archive enough efficiency, we have applied a k-d-tree indexing structure to organize and access the data. Given a query, the system looks for similar images in the database searching for similar features that define the query regions. Finally, the systems applies a Hough-like voting strategy to locate the query object in the database scenes.

Finally we present several examples of the potential applications of our proposal.

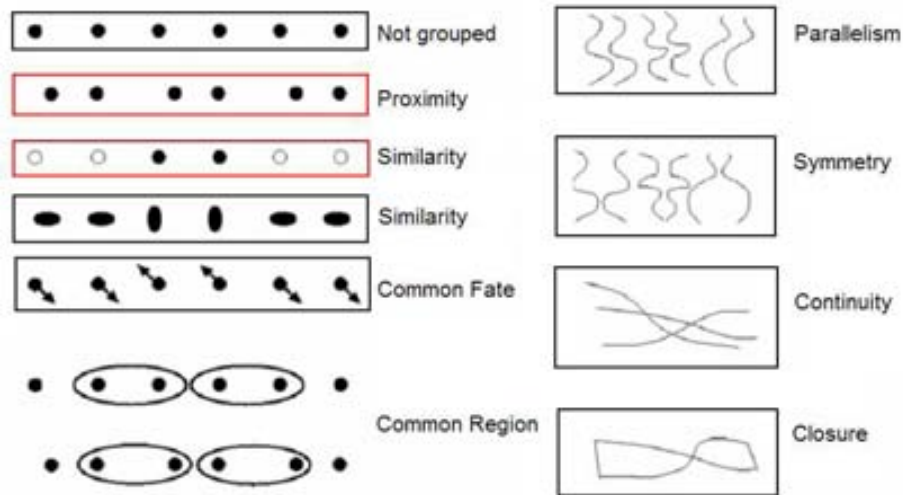
## 6.3 Description

In a similar scenario as the query-by-sketch, the query-by-paint presents particular characteristics that have to be attached from the description phase. Notice that the paint-based query has been created according to a human-based representation. Thus, in order to match it against the database we have to extract the information of the scenes according to a perceptual analysis.

Moreover, the human-based paintings present some important distortions of the image features in terms of color and shape. Facing with this kind of images provoke that not all the image descriptors could be applied to perform the matching of the content of a sketch paint against the content of a real image. We have selected the descriptors according to an experiment that we expose in the section 6.5.

### 6.3.1 Part extraction

Image segmentation is one of the most common strategies to extract relevant information from images. Nevertheless, unsupervised segmentation turns into a very severe problem when it has to emulate human-based criteria. Processes involved in the human perception have been studied since decades from a psychological point of view. This way, the computer science community has attempted to translate the psychological framework into effective algorithms of image segmentation. The Gestalt school is a well-known organization of psychologists who modelled the perceptual process according to a set of rules called Laws of Organization. These rules explain the grouping processes of the image components into higher level patterns [Wer55]. We have developed a segmentation strategy inspired in two of the Gestalt laws: Similarity and Proximity. The Law of Similarity declares that the mind groups elements that share similar features. Moreover, the Law of Proximity asserts that humans perceive close elements as a collective (see Figure 6.3). Our part extraction method translates the Similarity Law into the analysis of the color features and also translates the Proximity Law into the analysis of their spatial distribution. Next we expose more deeply the basis of our segmentation proposal and then we detail its implementation.



**Figure 6.3:** Gestalt laws of Organization. We focus on the laws of proximity and color similarity.

The development is inspired in the work of Matas that defines the concept of maximally stable extremal regions (MSER)[MCMP02]. Matas informally explained this concept as follows. Imagine all possible thresholds of a gray-level image. Then, we can see the thresholded images as a movie where we start with the lowest threshold and we gradually increase it. Thus, the first image is white and subsequently black spots begin to appear and grow. These spots correspond to local intensity minima, so they continue merging until reaching a whole black image in the last movie frame. To evaluate this evolution, a stability function between regions of consecutive frames is defined. This function consists in the rate of change of the region areas. This way, threshold levels that are local minima of this rate change are selected to produce the maximally stable extremal regions. The MSER has been exported to color images by Forssén [For07] applying the same idea of stability by looking at successive time-steps of an agglomerative clustering of image pixels. Following the movie example, imagine that we have a color image instead of a grayscale one. Then, we have a threshold related to the color distance of two pixels. We begin with an image where every pixel is an independent region. As we increase the threshold the connected pixels which distance is lower than this value begin to fuse. Finally, all the pixels form a single region in the last frame. This way, the more distant the color of a region is respect from the color of the surrounding pixels, the more the stable the region is.

Notice that the color analysis takes into account the structures formed by connected pixels but does not analyze the emerging structures according to the scale of observation. Going back to the human perception, it seems a natural process to group similar color regions if they are close enough. The space scale analysis is the tool that

allows performing this kind of region association. The process takes an important role to identify objects in real images when they suffer from partial occlusions or they present textured patterns. This way, we propose to export the stability measure to the spatial arrangement of the pixels. To illustrate the spatial stability, imagine that we have an image with a set of segmented regions. Then we also have a threshold related to the spatial distance of the region centroids. Our movie starts with this set of regions being independent and, as we increase the threshold, the regions are progressively joined. In the last frame they are all joined in a single one. Then, the more isolated a region is, the more stable the region is.

Finally, we quantify the saliency of a region mixing the stability measures of color and space. We have named CoReSt (Color Region Stability) our region extraction algorithm. Next we present its concrete formulation.

### The CoReSt implementation

To extract the image parts we have used the mean shift clustering algorithm proposed by Comaniciu [CM99b]. It enables clustering of a set of data-points into several different clusters, without prior knowledge of the number of clusters. The mean shift is based on the non parametric density estimator at the  $d$ -dimensional feature vector  $\vec{x}$  in the feature space, which can be obtained with a kernel  $K(\vec{x})$  and a  $d$ -dimensional hypersphere with radius  $h$ :

$$f(\vec{x}) = \frac{1}{nh^d} \sum_{i=1}^n \left( \left\| \frac{\vec{x} - \vec{x}_i}{h} \right\| \right)$$

where  $\vec{x}_i$  are the  $n$  feature vectors of the data set. Calculating the gradient of the density estimator shows that the Mean Shift defined by  $m(\vec{x})$  points toward the direction of the maximum increase in the density.

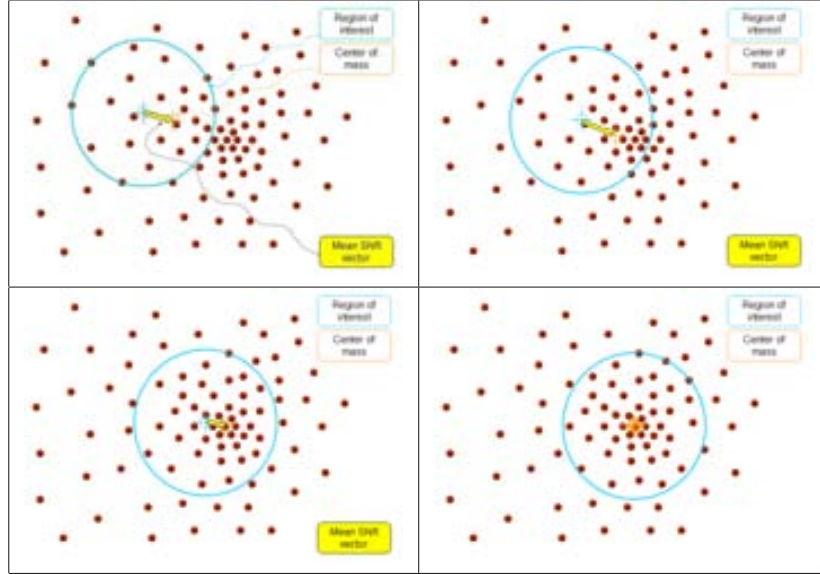
$$m(\vec{x}) = \frac{\sum_{i=1}^n \vec{x}_i K(\vec{x} - \vec{x}_i)}{\sum_{i=1}^n K(\vec{x} - \vec{x}_i)}$$

Then, the body of the algorithm is to move the hypersphere centered on each point iteratively by  $\vec{x}^{t+1}$ .

$$\vec{x}^{t+1} = \vec{x}^t + m(\vec{x}^t)$$

It is guaranteed that the shift converges to a point where the gradient of the underlying density function is zero. The Figure 6.4 shows an intuitive illustration of the mean shift computation. When the mean shift procedure is applied to every point in the feature space, the points of convergence aggregate in groups which can be merged. These are the detected *modes*, and the associated data points define their *basin of attraction*. The clusters are delineated by the boundaries of the basins, and thus can have arbitrary shapes. The number of significant clusters present in the feature space is automatically determined by the number of significant modes detected.





**Figure 6.4:** Illustration of the mean shift iterations in a 2D data. The objective is to find the densest region.

In our case, a pixel is understood like a point in a 5D space where its first three dimensions are related to the color values in the Luv space the other two represent the  $(x,y)$  coordinates in the image. We define a part of the image as the set of pixels belonging to the same *basin of attraction*, in other words, the pixels that have lead to the same local density maximum in the 5D space. Notice that the pixels of an image part belong to the same *mode* but they do not need to form a a single connected component in the image. We use an implementation of the mean shift clustering that depends on two thresholds,  $hc$  and  $hs$ , that control respectively the similarity constraints on the color and the space [CGM02]. The distribution of the points along the five dimensions (Luvxy) are previously scaled according to  $hc$  and  $hs$  in order to use a unique density radius  $h = \max(hc, hs)$ . After applying the mean shift process we obtain a segmentation of the input image.

Then, as we show in Figure 6.5, we construct a bidimensional grid  $G$  filled up with the segmented images using different values for the parameters. Let us denote  $MSS$  the mean shift function and  $HC$  and  $HS$  the two sets of thresholds,  $HC = \{hc_1, \dots, hc_{NC}\}$   $HS = \{hs_1, \dots, hs_{NS}\}$ . Given a region  $r$  of a cell in the grid, we define as *analogous regions* the regions of the other grid cells that maximize the overlapping area with  $r$ . Analogous regions are therefore found along the color or space dimensions, varying the corresponding thresholds in HC and HS respectively. The intuitive idea of an analogous region of a region  $r$  is that it is the evolution, i.e. the closest region, to  $r$  in another segmentation scale. We denote  $r_i^{(x,y)}$  the region  $i$  of the grid cell  $(x,y)$  and  $ar_{(x,y)}^{(x,y)'}_i$  its analogous region of another cell  $(x,y)'$ .

Once we have the segmentation evolution along the color and the space dimensions, we need a function to evaluate the stability of the regions. The stability function models the shape variation of a region along the two dimensions of the grid. The features used in the computation are the first and the second central moments. These features visually correspond to the area of a region and the axis lengths of the minimum enclosing ellipse of the region. Then, given a region  $r_i^{(x,y)}$  and another analogous one  $ar_{(x,y)'}^{(x,y)_i}$  we calculate the stability  $S$  as a combination of the variation of the area rate and the axis length. Let us denote with  $A$  the function that computes the area of a region and with  $L$  and  $l$  the functions that compute the maximum and minimum lengths of the axis. For the sake of readability, we simplify the notation of  $r_i^{(x,y)}$  to  $r_1$  and  $ar_{(x,y)'}^{(x,y)_i}$  to  $r_2$ . Thus, the stability measure between two regions  $r_1$  and  $r_2$  is defined as:

$$S(r_1, r_2) = S_{area}(r_1, r_2) * 0.5 + S_{axis}(r_1, r_2) * 0.5$$

$$S_{area}(r_1, r_2) = \frac{\min(A(r_1), A(r_2))}{\max(A(r_1), A(r_2))}$$

$$S_{axis}(r_1, r_2) = \min\left(\frac{\min(L(r_1), L(r_2))}{\max(L(r_1), L(r_2))}, \frac{\min(l(r_1), l(r_2))}{\max(l(r_1), l(r_2))}\right)$$

For each region of each cell we compute its stability along the two dimensions of the grid. The computation is done by the mean of the  $S$  values regarding the analogous regions. We name  $SC$  to the function that measures the stability along the color, and  $SS$  its equivalent in the space.

$$SC(r_i^{(x,y)}) = \sum_{X=1}^{\#HC} \frac{S(r_i^{(x,y)}, at_{(X,y)}^{(x,y)_i})}{\#HC} \quad SS(r_i^{(x,y)}) = \sum_{Y=1}^{\#HS} \frac{S(r_i^{(x,y)}, ar_{(x,Y)}^{(x,y)_i})}{\#HS}$$

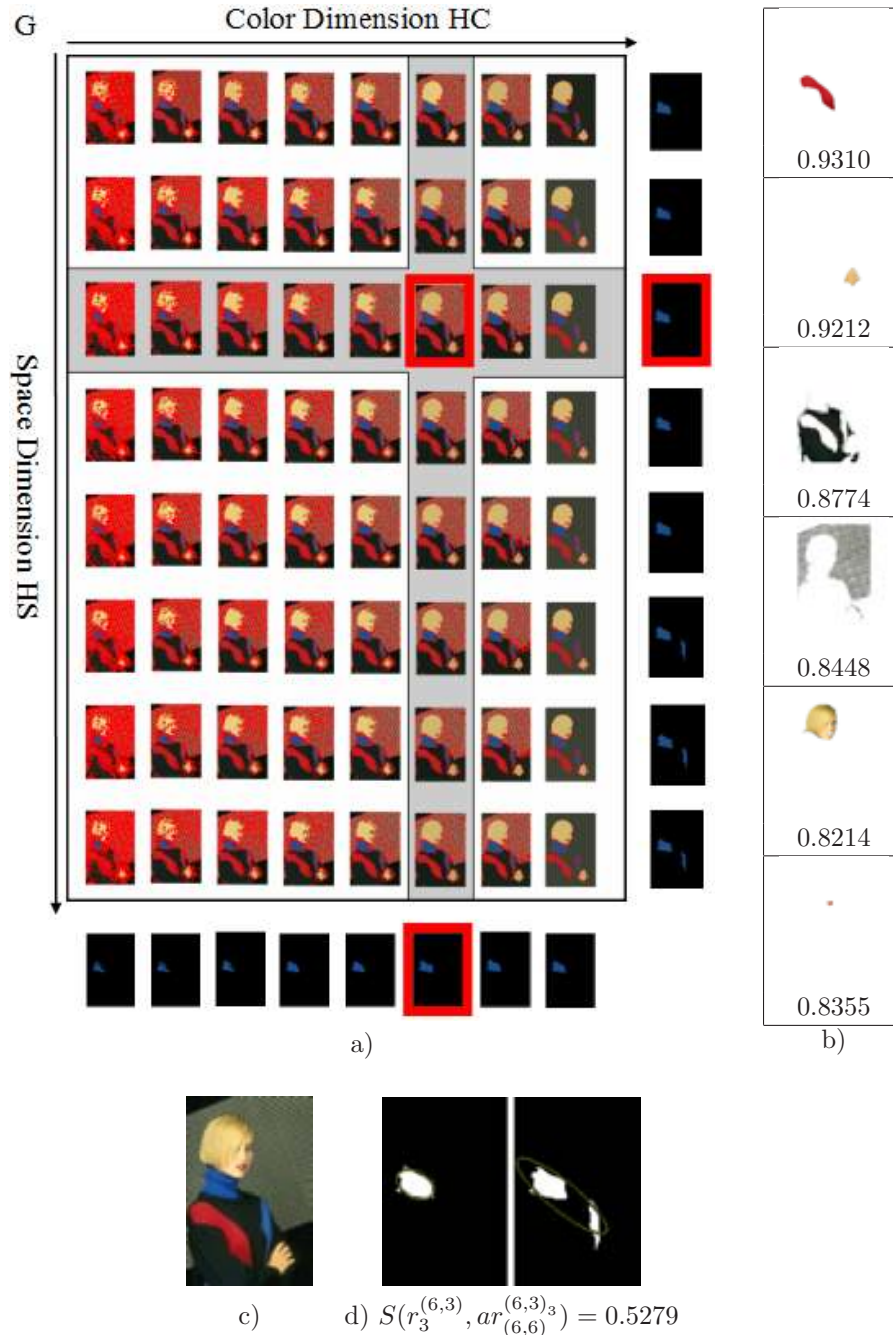
At this point we have two measures of stability for every region of the segmented images of the grid. Nevertheless, we search for a representative subset of regions that fits the human perception. Following the idea of the laws of similarity and proximity we propose to select the subset of regions  $R$  that maximize the stability functions along the color and space dimensions. This set of regions are those that form the output response of the CoReSt method.

$$R = \{r_i^{(x,y)}\} \text{ where } r_i^{(x,y)} \mid PC(r_i^{(x,y)}) \text{ or } PS(r_i^{(x,y)})$$

$$PC(r_i^{(x,y)}) = SC(ar_{(x-1,y)}^{(x,y)_i}) \leq SC(r_i^{(x,y)}) > SC(ar_{(x+1,y)}^{(x,y)_i})$$

$$PS(r_i^{(x,y)}) = SS(ar_{(x,y-1)}^{(x,y)_i}) \leq SS(r_i^{(x,y)}) > SS(ar_{(x,y+1)}^{(x,y)_i})$$

Among the final output regions a global measure of relevance is also computed. This value combines the stability of color and space using the function  $SCS$ . Notice



**Figure 6.5:** a) Grid of segmented images using the *MSS* according to the parameters of color *HC* and space *HS*. We show an example of the region  $r_3^{(6,2)}$  and its analogous ones. Observe how it grows trough the color, merging with similar pixels, and how it grows trough the space merging with similar regions. b) Some selected regions and their stability value *SCS*. c) Original image. d) Stability value  $S$  of  $r_3^{(6,2)}$  according to the analogous region on the cell (6,6).

that all the computations we have presented work in the range  $[0, 1]$ , then the global stability is also in this range.

$$SCS(r) = \frac{SC(r) + SS(r)}{2}$$

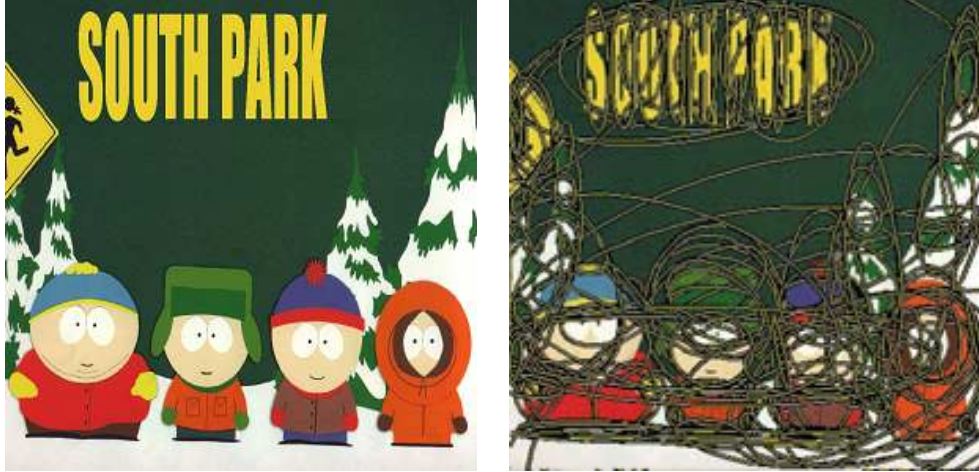
Next we show some examples of the detected regions in scenes of different nature: Figure 6.6 (a real outdoor scene), Figure 6.7 (a real indoor scene) and Figure 6.8 (a paint). The regions of these examples are highlighted by the ellipses that approximate them.



**Figure 6.6:** Regions extracted by CoReSt algorithm. Notice the grouping of several elements such as the natural textures of the trees or the sail with its shadow. Details like the little reddish tree are also preserved.



**Figure 6.7:** Examples of region detection. We can group the parts of the bomb drawing or the textures of the background posters.



**Figure 6.8:** Example of the regions detected in the scene. Observe that the letters of the SouthPark banner are detected individually but are also understood as a group.

The images of the database of our retrieval system are preprocessed and the first  $N$  regions, ordered by decreasing stability, are taken to describe the content. In the case of the painted-query the CoReSt algorithm is also applied. Both sets of regions, the ones of the query and the ones of the database, have to be described with a set of features in order to allow the further matching.

### 6.3.2 Feature extraction

An image, either the query or a scene, is represented by a set of  $N$  regions. In the case of the query image we can benefit of the part extraction process in order to characterize its content. Notice that the user paints the desired object on an empty canvas that provides no information and that constraint the size of the object. Then, from the parts that we have obtained we can remove those belonging to the background and encode the ones that do belong to the paint.

Then, every region  $r$  of the image is described with a set of features  $F$ :

$F_r = \{P(r), A(r), Sz(r), C(r), S(r)\}$ : The features describe with information related to the spatial position ( $P(r)$ ), the area ( $A(r)$ ), the color ( $C(r)$ ) the shape ( $S(r)$ ) and the size ( $Sz(r)$ ).

- **P(r): Position** This feature  $P$  includes the coordinates  $(x, y)$  of the centroid of the region.

Computation: the coordinates of the position take their origin in relation to the bounding box  $BB$  of the image information.

$P(r_i) = (x, y) \mid (0, 0)$  is the upper left corner of  $BB(\cup_{j=1}^N r_j)$

- **A(r): Area** A region is defined by its area in relation to the total area of the object.

Computation: It is computed as the rate:

$$A(r_i) = \frac{A(r_i)}{A(\cup_{j=1}^N r_j)}$$

- **Sz(r): Size** To define the size of an image we use the second central moments of the region. Then Sz takes the value of the length of major axis of the ellipse that approximates the region.

- **C(r): Color** To describe the color information we have used the Luv value of the mean shift mode that defines the region.

$C(r) = [L, u, v] \mid [L, u, v]$  are the first 3 dimensions of its correspondent mean shift mode.

- **S(r): Shape** The shape information is encoded using the binary mask of the region.

Computation: To extract the shape feature we crop the mask of the region with a square window that is centered in the centroid. The size of the side of the square is equal to the size feature Sz(r). Then, we use the first six coefficients of the Discrete Cosinus Transform (DCT) of the mask representation. The number of coefficients is the same as the amount of data used to encode the intensity channel in the color layout descriptor of the MPEG-7 standard [MSS02].

## 6.4 Matching

In the matching process we access to the database information using several indexing structures and then we perform a Hough-like voting process to evaluate the presence of the query in the scenes.

### 6.4.1 Feature indexing

As we want to perform a query detection as a part of the scene we have to store two kind of low level information of the scene parts: the features that allow the similarity matching and the features that allow the localization of the regions in a scene.

The features used to measure the region affinity are those that describe the color  $C(r)$  and the shape  $S(r)$ . We have used a k-d-tree to index the features that describe the regions. The k-d-tree is a binary search tree that represents a recursive subdivision of the space into subspaces by means of (d-1) dimensional hyperplanes [GG98]. k-d-tree structures are commonly used in content-based image retrieval field to index the features [STL97] [Lew00]. For specific dimension and under certain assumptions about the underlying data, the k-d tree requires  $O(n \log n)$  operations to construct and

$O(\log n)$  operations to search. The k-d tree is very efficient to employ, in vectors of low dimensionality, that do not exceed 15 dimensions [KC04]. In our case we find this indexing method suitable because the color features have only 3 dimensions and the shape ones have 6 dimensions.

We have constructed two trees, one for each feature. We can access them independently according to the values of a query region and then, in further step, we combine the results. For an easier indexation of the feature vectors, we have scaled them to narrow their Euclidean norm between [0:1]. The transformation is done according to the maximum and minimum range values that each feature type can take in their corresponding d-dimensional space. We retrieve those database regions which features are within a fixed distance range of  $T_k$  from the features of the query. This distance is quantified using a D-dimensional Euclidean (2-norm). Thus, given a query image, we can search across the indexed values and retrieve which database regions have similar features within a certain distance range.

Moreover, the localization features are related to the size  $Sz(r)$  and the spatial position  $P(r)$  of the regions. Because we want to detect the presence of the query as a subpart of a scene, the system maintains a table with these features. Notice that the retrieval is done at the same time for all the regions of all the database images. This way, the indexing structures need to address the identifier of the region and also the identifier of the image where it belongs. Using as the primary key these both identifiers, we can access to the table information that allows the localization of the regions.

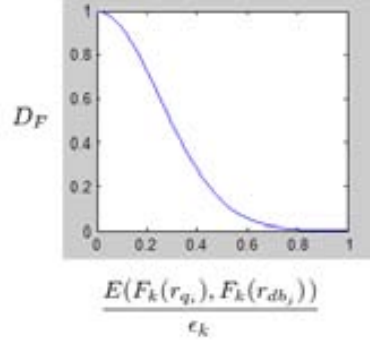
These two kind of structures (k-d-trees and indexing tables) work together in the evaluation step of the matching process.

### 6.4.2 Evaluation

Given two feature values  $F_k(r_{q_i})$  and  $F_k(r_{db_j})$ , one from the query and the other from a scene, we use a function  $D$  to evaluate the similarity distance. Instead of the ramp function that the application of the Euclidean distance would provide, we use an exponential modulation to obtain a faster decrement of the similarity according to a range  $\epsilon_k$ . Its formulation is like follows:

$$D_F(F_k(r_{q_i}), F_k(r_{db_j}), \epsilon_k, \alpha) = \begin{cases} 0 & \text{if } E(F_k(r_{q_i}), F_k(r_{db_j})) > \epsilon_k \\ e^{-\frac{(E(F_k(r_{q_i}), F_k(r_{db_j}))/\epsilon_k)^2}{2\alpha^2}} & \text{Otherwise} \end{cases}$$

where  $\alpha = 0.25$ ,  $E$  is the Euclidean distance and  $\epsilon_k$  is the retrieval tolerance range. The function  $D_F$  is graphically shown in the Figure 6.9.



**Figure 6.9:** Graphic of the function  $D_F$  with  $\alpha = 0.25$ . X axis represent the Euclidean distance between two features normalized by the maximum tolerance  $\epsilon_k$ . Y axis shows the function response according the values of the X axis.

As we have introduced before, the similarity of the regions is evaluated from the color and the shape. Given a query image, each region  $r_{q_i}$  is treated individually using their description features,  $C(r_{q_i})$  and  $S(r_{q_i})$ . Then we find those database regions  $\{r_{db_j}\}$  that are visually similar. We have chosen to use of an *and* combination to assure that the features of database regions accomplish both similarity restrictions. Then, we can formulate  $D_R$  as the function that measures the similarity of two regions:

$$D_R(F(r_{q_i}), F(r_{db_j}), \epsilon, \{\alpha\}) = D_F(C(r_{q_i}), C(r_{db_j}), \epsilon_c, \alpha) * D_F(S(r_{q_i}), S(r_{db_j}), \epsilon_s, \alpha)$$

The regions of the query can be matched individually with the regions of the scene according to its appearance. Nevertheless, this independent part matching needs a further evaluation in order to assure that the spatial arrangement of the query parts is coherent in the scene images. Since the information of the k-d-trees points the identifiers of the scene regions, we proceed to check their location.

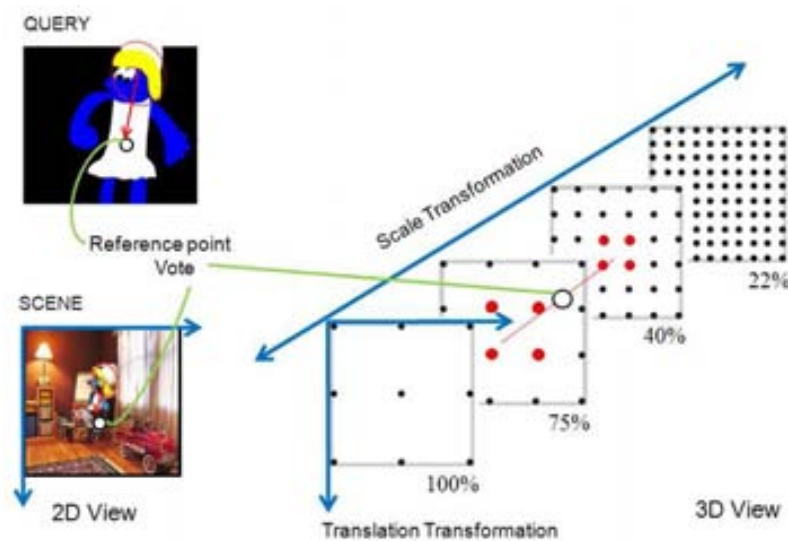
We use a Hough-like voting strategy to evaluate the matching and locate the target query in the scenes. The weight of the vote is computed as the similarity distance  $D_R$  and is weighted by the area of the query region  $r_{q_i}$ .

$$W(F(r_{q_i}), F(r_{db_j}), \epsilon, \alpha) = \frac{A(r_{q_i})}{\sum_i^N A(r_{q_i})} * D_R(F(r_{q_i}), F(r_{db_j}), \epsilon, \alpha)$$

We want the system to be flexible enough to search for objects that have suffered scale transformations as long as translation ones. We take as a reference point the centroid of the query object. Then we use the indexing tables of the retrieved region scenes  $\{r_{db_j}\}$  to recover the location features  $Sz(r)$  and  $P(r)$ .



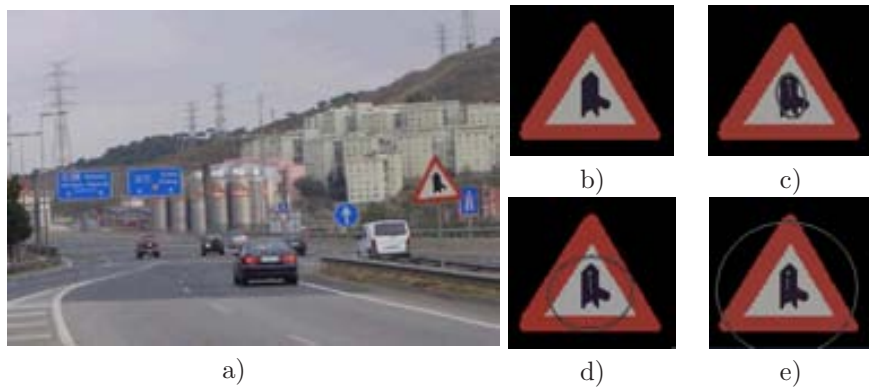
One of the critical points of the Hough voting strategies is the search of the maximum density of votes to prove the evidence of the object detection. To avoid this drawback, we have designed a voting system that deals with a predefined set of accumulator points that we name  $ap$ . Thus, we distribute them with certain accuracy along the spatial coordinates and along the scale variations that we allow. In the Figure 6.10 we show an example of the distribution of the accumulator points using four scales and a accuracy of the 50% in its coordinate distribution. We have explained that, for every query region  $r_q$  we obtain a set of database regions  $\{r_{db}\}$  that vote on the accumulator map. Every vote is accumulated in eight  $ap$  points, four for each of the two closest scales.



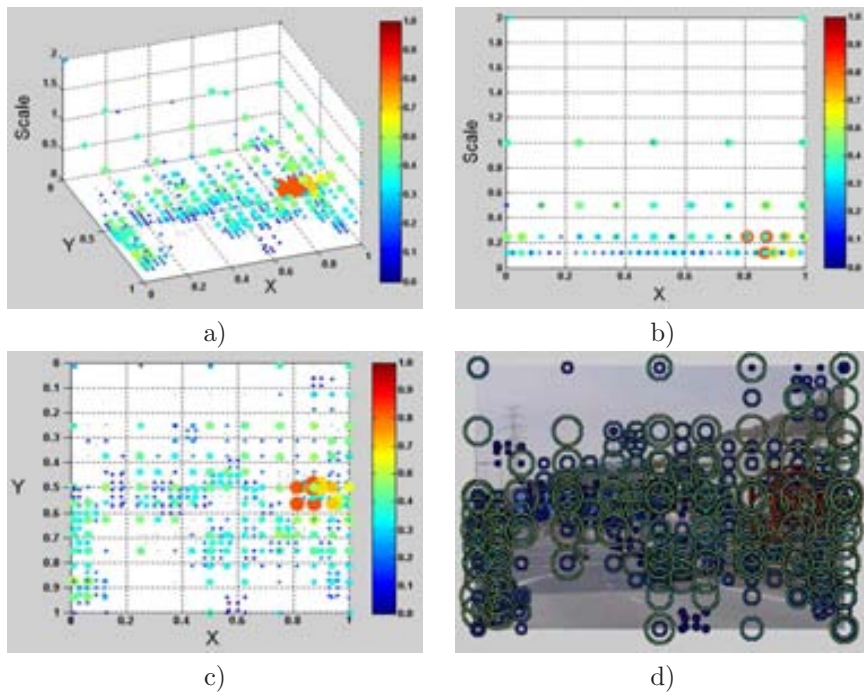
**Figure 6.10:** Example of matching between a query region and a scene. The generated vote is showed with a frontal and a perspective view in the voting space. The first and second dimensions of the space are related to the coordinate localization of the reference point, the third one specifies the scale transformation and, a fourth one that is not represented, identifies the image scene.

Another critical point of the Hough-like voting is the accumulation in one cluster of multiple instances of the same element. To solve this effect the matching values of the accumulator points increase with the maximum instance of their corresponding votes. If the query has  $N$  regions, each accumulator point sums a maximum of  $N$  votes. The goodness of the query matching in every scene is computed as the maximum accumulation of votes for all the  $ap$  points that belong each scene.

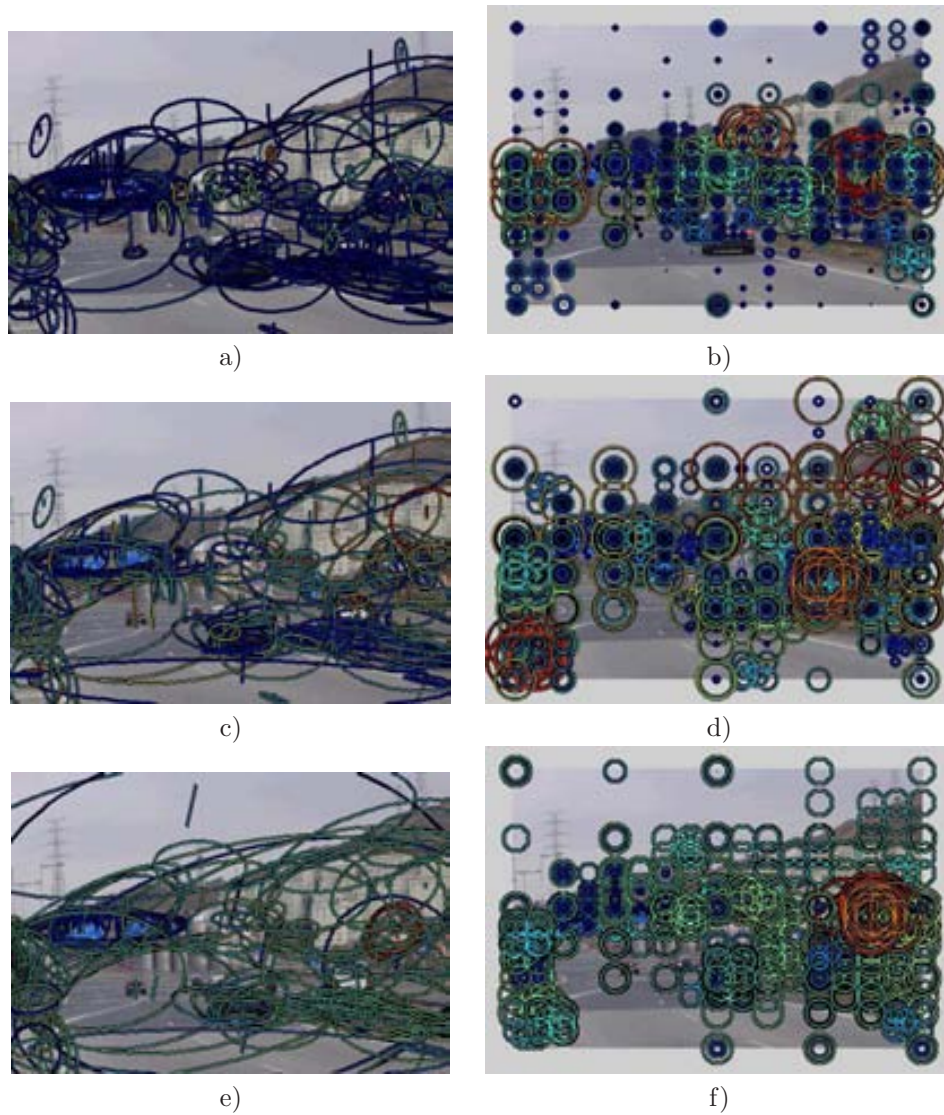
Figures 6.11 6.12 and 6.13 show an example of the matching process. The Figure 6.11 presents the scene image, the query point and the three regions that compose it. Figure 6.11 illustrates the values of the accumulation points in the voting space after the query matching. Finally, the Figure 6.13 details the voting process of the three steps (each one of a query region).



**Figure 6.11:** a) Scene b) Query c) d) e) Query regions. The reference point is marked with a symbol 'o'. We show the line joining the center of the region with the reference point of the query.



**Figure 6.12:** Voting space: a) 3D view of the accumulated result in the predefined points  $ap$ . The size and the color of the spheres represent the goodness of the matching (red and big:good, blue and little:bad). b) Projection of the scale transformation plane. c) Projection of the 2D coordinates plane. d) Analogous representation of c) over the image scene.



**Figure 6.13:** The steps of the matching process of the three regions of the query. a) c) e) Show the scene regions that are similar to the query ones that we show in the previous Figure 6.11. b) d) f) Show the similarity values in the accumulation points of the voting space. The values of the query steps are combined together to form the final result of the figure 6.12.

Other examples are shown in the Figures 6.14 and 6.15.

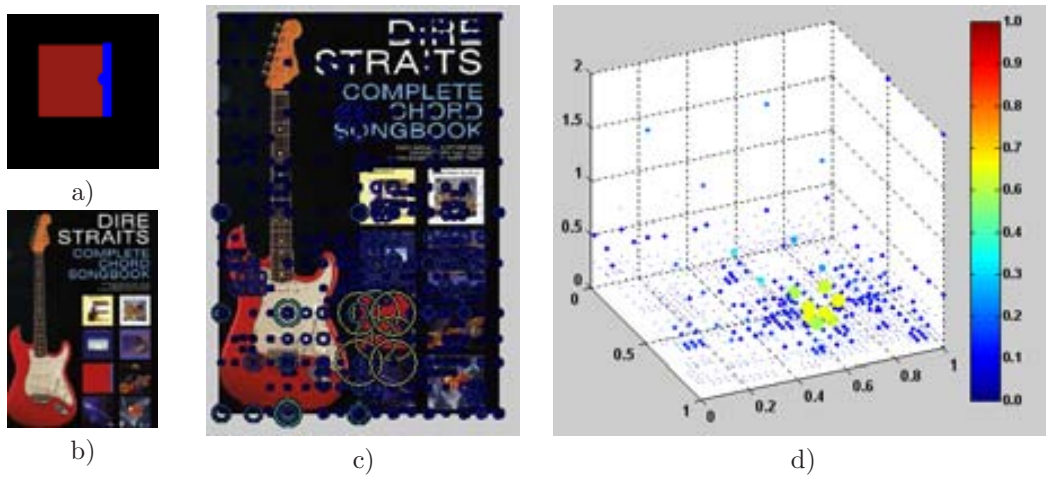


Figure 6.14: a) Query b) Scene c) d) 2D and 3D views of the voting space.

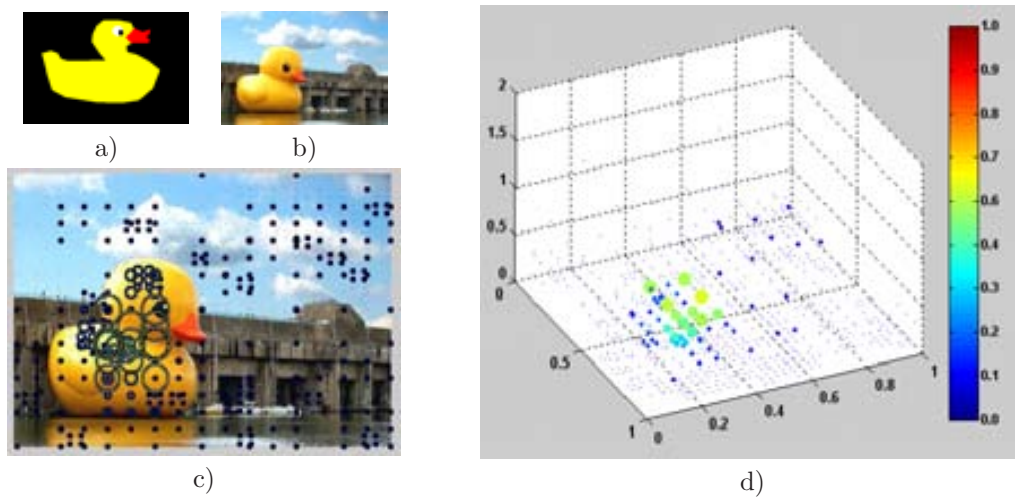


Figure 6.15: a) Query b) Scene c) d) 2D and 3D views of the voting space.

## 6.5 Experiments and Results

We have made two kind of experiments in order to validate the performance of the image description process. The first experiment validates that the process of region extraction approximates the human criteria. The second experiment is used to obtain the most suitable descriptors of color and shape according to a query paint.

### 6.5.1 Validation of the part extraction

We have evaluated the performance of the proposed method with the public segmentation dataset of Berkeley [MFTM01]. The test set comprises 200 color photographs which have been manually segmented. For each of the images, at least 5 segmentations produced by different people are available. We have generated a segmentation result on every original image of the dataset. The experimentation provides a numerical evaluation among the CoReSt solution and the manual benchmark.

In the literature we can find several measures to evaluate the segmentation results [UPH07]. We have chosen to use the Global Consistency Error (GCE) since it is a standard framework in a number of a state-of-the-art methods. The GCE measure takes care of the refinement between two segmentations: being  $IS_1$  the segmentation of the benchmark and  $IS_2$  the segmentation we evaluate, it produces an error measure in the range  $[0, 1]$  (the lower, the better). For each pixel  $p_i$  the GCE evaluates the difference between the regions of both segmentations  $r \in (IS_1, p_i)$  and  $r \in (IS_2, p_i)$  that contain this pixel. Let us denote  $n_{pix}$  the number of pixels in a image,  $\setminus$  difference operator, and  $|\cdot|$  the cardinality one.

$$GCE = \frac{1}{n_{pix}} \min \left( \sum_i \frac{|r \in (IS_1, p_i) \setminus r \in (IS_2, p_i)|}{|r \in (IS_1)|}, \sum_i \frac{|r \in (IS_2, p_i) \setminus r \in (IS_1, p_i)|}{|r \in (IS_2, p_i)|} \right)$$

To apply this kind of evaluation we need to resume the extracted regions in a unique segmented image. The result of *SCS* allow to rank the regions by its meaningfulness: the greater, the more meaningful. Taking into account this ranking we generate an image segmentation that combines the regions.

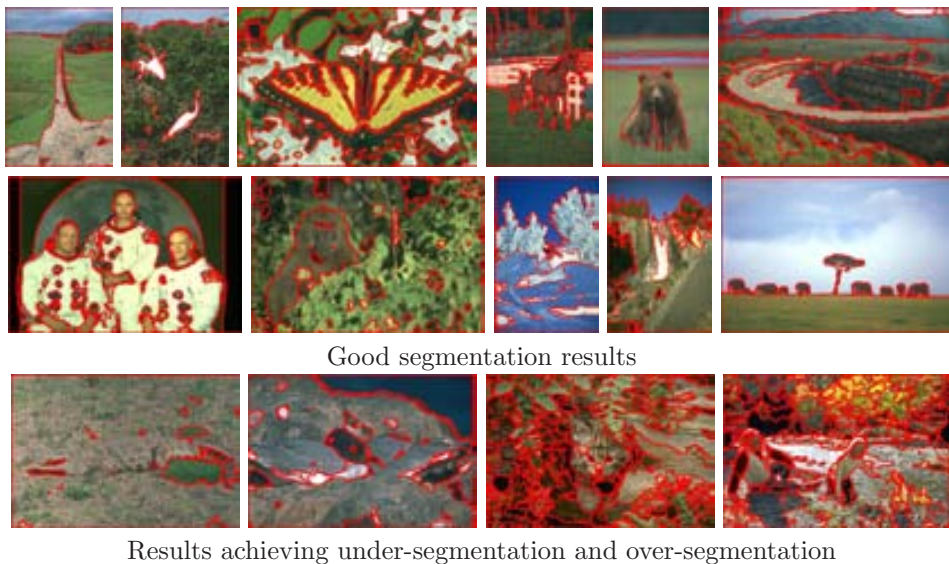
The process consists in constructing a pile of regions ordered by stability. Then, in the deepest positions we find the less stable regions and in the most superficial positions we have the most stable ones. The segmentation generation can be understood as a z-buffering analysis of this pile of regions. The segmentation incorporates the boundaries of the regions following the priority order defined by the stability measure. Notice that a region  $r_i$  will not be included in the final segmentation if its pixels are overlapped by another region  $r_j$  that is more stable. The figure 6.16 provides an example of this segmentation construction.



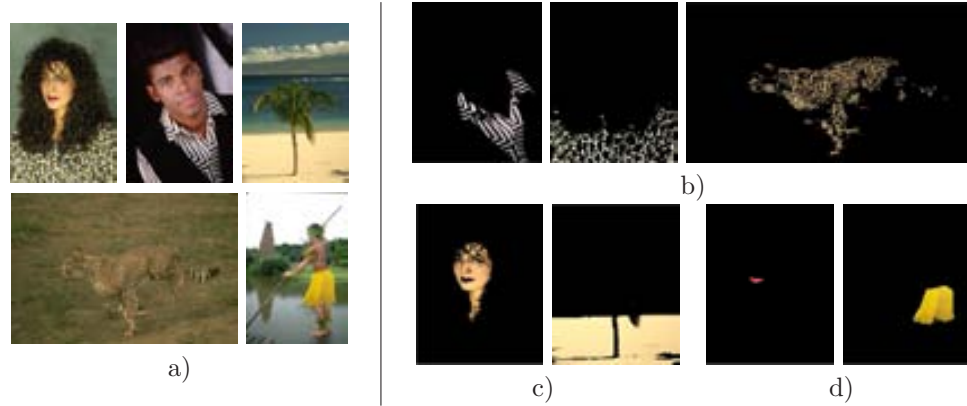
**Figure 6.16:** Generation of the segmentation using the CoReSt regions ranked by the stability. The first row shows the progress of the boundaries and the second shows the progressive incorporation of the regions. Those regions that are occluded by other regions that are more stable are not included in the segmentation result.

We have fixed the method parameters for all the test set and we have computed the GCE values for the test set of 200 images. We have used a grid of  $[8 \times 8]$  cells with a set of space thresholds that explore the whole image  $HS = \{1 \dots \max(I_{height}, I_{width})\}$  and a set of color thresholds in the range 3 to 32  $HC = \{3 \dots 32\}$ .

Some qualitative examples of the CoReSt segmentation are shown in the Figure 6.17. The images of the Figure 6.18 present some regions that illustrate the properties of the CoReSt method.



**Figure 6.17:** Examples of the CoReSt segmentations



**Figure 6.18:** a) Original Images. b) and c) present the grouping properties. b) Regions that form a textured area. c) Regions that present occlusions. d) Regions detected as outstanding for having contrasted color and being isolated

The inter-variability among the human segmentations obtains a mean GCE value of 0.08 and our method obtains a score of 0.1946. The table 6.19 summarizes the GCE values obtained by other segmentation methods: the Ridge based Distribution Analysis (RAD)[VvdWB08], the Multiple Seed Segmentation (Seed)[MH06], the pairwise pixel affinity algorithm proposed by Fowlkes (Fow)[FMM03] and the Normalized Cuts (nCuts)[SM00].

	Human	CoReSt	RAD	Seed	Fow	MS	nCuts
GCE	0.0800	0.1946	0.2048	0.209	0.214	0.2598	0.336

**Figure 6.19:** Comparison of the CoReSt strategy against other state-of-the-art methods. GCE results for the 200 Berkeley images (values taken from [VvdWB08]).

An interesting point in the comparative table is to see that the proposed method outperforms the results obtained by original Mean Shift algorithm (MS). This way, we can see that the stability measure helps on building a segmentation result that adapts better to the human based criteria.

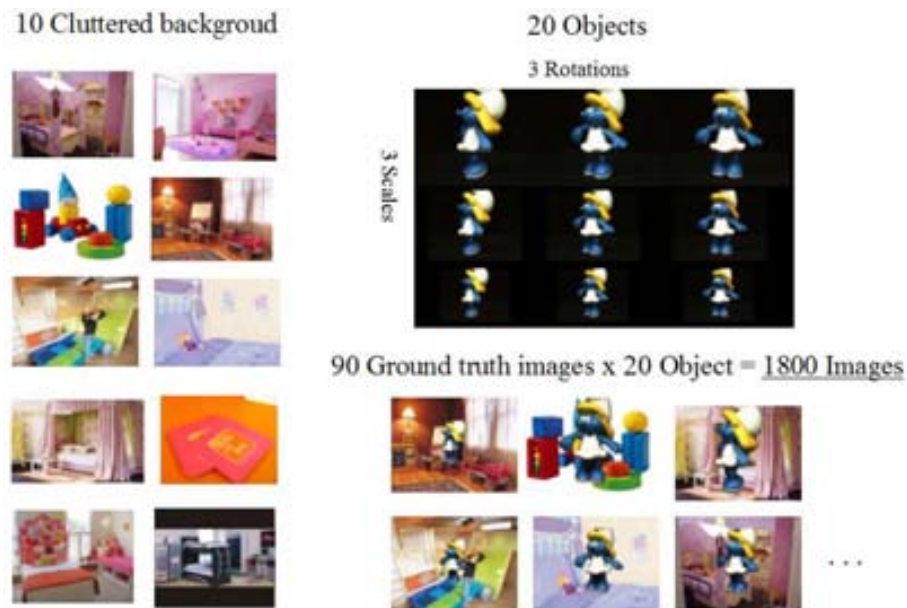
## 6.5.2 Evaluation of the description features

To analyze the CBIR process and the paint features we have constructed a database of images and we have collected a set of user made queries. We have used as models some images of the public database of the Amsterdam University, the ALOI database. This database contains instances of several objects with variations of viewpoint and lightning conditions [GBS05]. We have selected 20 objects with various colors, different textures, and with different degree of complexity regarding the number of parts and the details they have.

We have constructed a database of scenes from 10 cluttered backgrounds and the original real images of the test ALOI objects. We have composed a total of 1800 scene images using 3 views and 3 scales of each object.



**Figure 6.20:** 20 objects of the ALOI database that we have used in the experiment of the features evaluation.



**Figure 6.21:** Construction of the 1800 images of the synthetic database combining 10 cluttered backgrounds and 20 objects with 9 variations.

To test the paint features we have collected the samples of 11 users. They have painted a sample image for each of the 20 objects using a simple painting software such as the Microsoft Paint tool. The users have drawn the queries with total freedom on the size and color of the objects. The Table 6.1 shows the query paints.



	user1	user2	user3	user4	user5	user6	user7	user8	user9	user10	user11
											
											
											
											
											
											
											
											
											
											
											
											
											
											
											
											
											
											
											
											

Table 6.1  
COLLECTION OF PAINTS OF THE TEST.

We have performed a set of experiments combining several features. Then we have evaluated the results with the Area Under Curve (AUC) measure of the ROC plot for the results of paint query.

First of all, let us focus on the set of color and shape features we have used. In the color description we have used the following features:

**MeanLuv** The MeanLuv descriptor contains the mean values for every channel of the image region in the Luv space. We have translated the RGB values to this color space because, unlike the RGB, the distances in the Luv space are perceptually consistent.

**HistColorNames** An histogram of the colors of the pixels present the region. This approach is very related to the human perception of the image content since the colors values are obtained by a color naming process. The process of color consists in assign to every image pixel the probability to be labelled with a set of basic colors (red, blue, green, brown, etc.). We have used an approach of 11 colors so, after the color naming process, every pixel is assigned a probability vector of 11 positions. To obtain a compact representation of the whole region we have taken the mean values related to each label [vdWSV07].

**ColorLayoutMpeg7** We have chosen a the color layout descriptor included in the Mpeg7 standard. This descriptor is obtained from the coefficients of the distance cosine transform applied on the channels of the image in the YCbCr space [MSS02].

We have chosen these color features because they have specific characteristics:

The MeanLuv provides a very compact descriptor with that represents the dominant color of the region. Otherwise, we have chosen HistColorNames because histogram-based descriptors have proven to be powerful descriptors in the retrieval field [SB91]. Because we are dealing with man-made paints that involve human perception, we have used a color naming technique instead of an image quantization. Histogram based approaches suffer from lack of spatial information of the color distribution. Due to this requirement we have tested the ColorLayoutMpeg7. This descriptor belongs to the MPEG7 standard and it is the unique color descriptor of the MPEG7 referred as suitable for color paint proposes [MSS02].

In relation to the shape measures we have used this set of descriptors.

**AspectRatio** The AspectRatio is a simple measure extracted from the second central moments of a region. From a graphical point of view is understood as the rate between the maximal and minimal values of the axis of the ellipse that approximates a region.

**DCT** The Distance Cosine Transform (DCT) is applied to the mask of the region and the 6 first coefficients are taken as the shape descriptor.

**ART** The Angular Radial Transform (ART) is one of the shape descriptors selected by the MPEG7 standard. The ART is a descriptor based in the region information (not the contour) and allows the description of disconnected components.

One more, the selection of the space features has been done according their characteristics: The AspectRatio is numerical descriptor that can useful in cases where the shape of the regions suffer form high variations, so that only a very coarse description is kept. Moreover we have applied the Distance Cosine Transform (DCT) to obtain a compact representation of the shape according to its binary mask. Finally, the ART gives a richer description regarding to the spatial distribution of the region pixels.

Even thought in the literature we can find a wide variety of color and shape descriptors we have chosen this reduced set taking into account their indexing capabilities. The selected features have rather low dimensional representation and their similarity can be measured using the Euclidean distance.

Then, using these nine descriptors we have performed an experiment using the collection of paintings and the database we have created. The experiment aims to evaluate which descriptors are more suitable to match a human-based paint and how this representation affects to the tolerance of the feature similarity. In the following tables (6.2, 6.3 and 6.4) we show the combination of features we have made and the threshold tolerances of color and space,  $\epsilon_c$  and  $\epsilon_s$ , we have test. The values shown in the tables correspond to the mean AUC values of all the 220 queries (20 objects x 11 users).

First we have performed the experiment using the features individually and taking into account three degrees of feature similarity. The feature similarity is represented with a parameter  $\epsilon$  and indicates witch is the tolerance range we use to access the information of the k-d-trees. Table 6.2 presents the results using the color features and Table 6.3 present the results using the shape ones.

Color Descriptor	Similarity tolerance $\epsilon_c$		
	0.6	0.3	0.15
MeanLuv	<b>0.7504</b>	0.7444	0.7331
HistColorNames	<b>0.7444</b>	0.7334	0.7230
ColorLayoutMpeg7	0.7477	0.7482	<b>0.7491</b>

**Table 6.2**

MEAN AUC VALUES USING THE COLOR FEATURES

Shape Descriptor	Similarity tolerance $\epsilon_s$		
	0.6	0.3	0.15
AspectRatio	0.6624	<b>0.6631</b>	0.6630
DCT	0.7060	<b>0.7088</b>	0.7067
ART	<b>0.6653</b>	0.6652	0.6646

**Table 6.3**

MEAN AUC VALUES USING THE SHAPE FEATURES

Using only one feature we observe that the color performs better than the shape. Within color features we do not find any outstanding differences and, for a slight advantage, we can see that the best description is provided by the MeanLuv. In relation with the shape, we can see that the best performance is provided by DCT descriptor. DCT is in the half way of the simplest AspectRatio and the accurate ART.

According to the best results of both experiments (tables 6.2 and 6.3) we have combined the parameters and we have made again the retrieval test. The results are shown in the table 6.4).

Color Descriptor		Shape Descriptor		
		AspectRatio $\epsilon_s=0.3$	DCT $\epsilon_s=0.3$	ART $\epsilon_s=0.6$
MeanLuv	$\epsilon_c=0.6$	0.6750	<b>0.7718</b>	0.7444
HistColorNames	$\epsilon_c=0.6$	0.7075	0.7466	0.7445
ColorLayoutMpeg7	$\epsilon_c=0.15$	0.6759	0.7561	0.7478

**Table 6.4**  
MEAN AUC VALUES USING THE COLOR AND SHAPE FEATURES

If we combine the color and shape descriptors we observe that the MeanLuv is benefits from the information of the DCT and it increases two tenth its result (see Table 6.4). The addition of the ART features do not implies a significant improvement and, in the contrary, the inclusion of the AspectRatio decrements the results.

We can analyze more deeply the individual results of the queries for the best combination (MeanLuv with  $\epsilon_c = 0.6$  and DCT with  $\epsilon_s = 0.3$ ). Then, in the Table 6.5 we present the 220 AUC values according to the model object and the user that has performed the query paint.

We have computed the mean AUC value for each user and the mean AUC vale of each object. The results are shown in the Tables 6.6 and 6.7. Looking at the results according to the user (Table 6.6) we find that there are notable differences between the one that their queries perform best (0.88 AUC mean) and the one that performs worst (0.64 AUC mean). If we do a visual inspection of the paints we realize that effectively they are quite in different styles and that the paints of the user11 are more close to the original object. In the Figure 6.5.2 we show some examples of the query instances of both users. We can observe the differences in the color selection as well as the shape.





















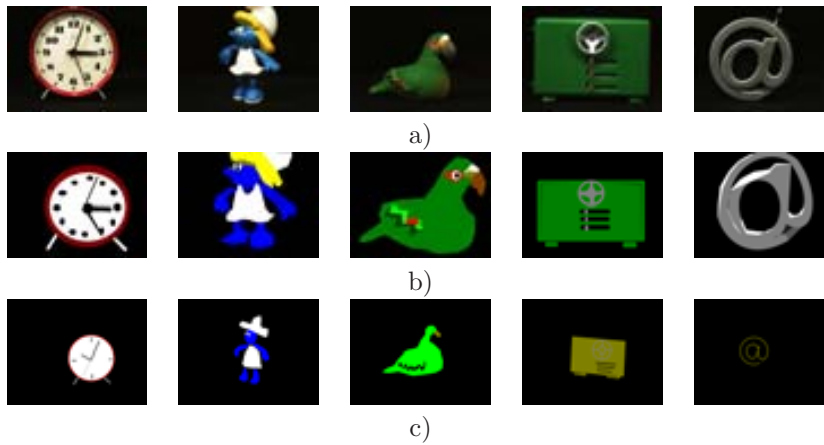
	user1	user2	user3	user4	user5	user6	user7	user8	user9	user10	user11
	0.64	0.87	0.61	0.60	0.60	0.48	0.67	<b>0.90</b>	0.34	0.69	0.85
	0.59	0.64	<b>0.73</b>	0.70	0.62	0.53	0.60	0.47	0.72	0.64	0.63
	0.76	0.86	0.99	0.91	0.92	0.98	0.98	0.85	0.45	0.94	<b>0.99</b>
	0.67	0.71	0.90	0.84	0.47	0.79	<b>0.98</b>	0.95	0.71	0.94	0.88
	0.47	0.63	0.74	0.37	0.64	0.54	<b>0.84</b>	0.74	0.55	0.58	0.80
	0.93	0.84	0.93	0.83	0.67	0.95	0.84	0.65	0.87	<b>0.97</b>	0.95
	0.75	0.60	<b>0.86</b>	0.43	0.37	0.40	0.62	0.52	0.69	0.71	0.78
	0.51	0.78	0.82	0.54	0.55	0.43	0.45	0.36	0.36	0.48	<b>0.89</b>
	0.57	0.94	0.94	0.95	0.75	0.98	<b>1.0</b>	0.71	0.58	0.84	0.89
	0.55	0.67	0.66	0.76	0.68	0.57	0.53	0.55	0.66	0.65	<b>0.77</b>
	0.62	0.85	<b>0.96</b>	0.51	0.87	0.63	0.60	0.59	0.82	0.94	0.92
	0.90	0.90	0.96	0.81	0.97	0.95	0.96	0.99	0.65	0.99	<b>1.0</b>
	0.95	0.88	0.95	0.87	0.92	0.94	0.91	0.74	0.83	0.91	<b>0.97</b>
	0.99	0.94	0.94	0.98	<b>0.99</b>	0.93	0.96	0.96	0.88	0.97	0.91
	0.87	0.77	0.83	0.83	0.88	0.72	0.79	0.89	0.54	0.70	<b>0.92</b>
	0.88	<b>0.99</b>	0.90	0.93	0.43	0.97	0.60	0.99	0.87	0.99	0.87
	<b>1.0</b>	0.99	0.98	0.99	1.0	0.95	0.99	0.35	0.51	0.91	0.99
	0.95	0.84	0.86	0.94	<b>0.97</b>	0.78	0.82	0.87	0.40	0.89	0.87
	0.71	0.83	0.86	0.77	<b>0.87</b>	0.95	0.97	0.84	0.75	0.64	0.75
	0.48	0.55	0.93	0.50	0.45	0.79	0.50	0.57	0.61	0.74	<b>0.98</b>

Table 6.5

AUC VALUES FOR EVERY QUERY USING: MEANLUV  $\epsilon_c=0.6$  AND DCT  $\epsilon_s=0.3$

user11	user3	user10	user2	user7	user6
0.8803	0.8675	0.8063	0.8045	0.7795	0.7624
user4	user1	user5	user8	user9	
0.7538	0.7404	0.7317	0.7239	0.6395	

**Table 6.6**  
USERS ORDERED BY THEIR MEAN AUC VALUES.

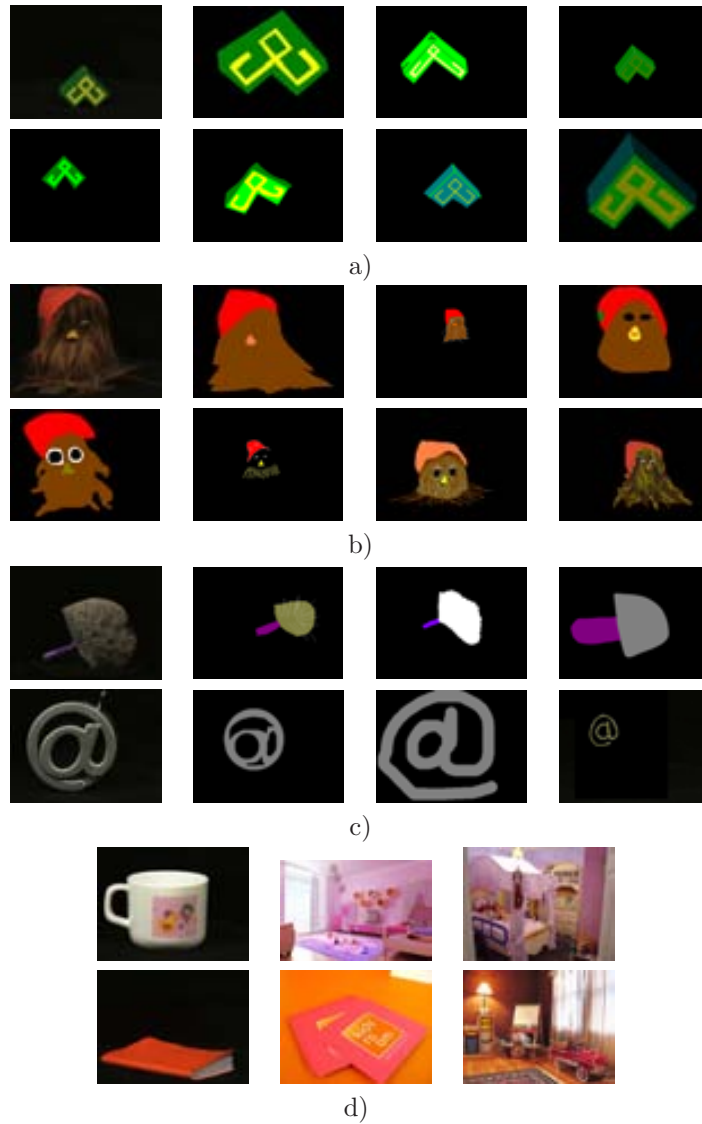


**Figure 6.22:** Examples of the queries. a) Original model. b) and c) paints from users 9 and 11, the one with best performance and the one with the worst. Observe the differences in the color and shape of the paints.

0.9517	0.9171	0.8977	0.8777	0.8763	0.8562	0.8556
0.8361	0.8332	0.8133	0.8044	0.7945	0.7550	0.6586
0.6450	0.6416	0.6267	0.6258	0.6101	0.5593	

**Table 6.7**  
OBJECTS ORDERED BY THEIR MEAN AUC VALUES.

Moreover, we can also find large differences according to the model objects. The main causes of a defective retrieval in the last objects of the Table 6.7 are illustrated in the Figure 6.5.2. They can be summarized in three main categories: large variations of color and shape, few regions and similar background.



**Figure 6.23:** Examples of the problems in the query-by-paint retrieval. a) Large color variations. b) Large shape variations. c) Objects represented by few regions and also large color and shape variations. d) Objects that can be confused with the background.

Large color and shape variations Some query paints representing the same object present a notable degree of color and shape variations. When we relax the tolerance in the color similarity the system can include false positives in the result.

Few regions In a similar way as the sketch-based query, the amount of object parts plays an important role in the retrieval process. Because the votes of the image parts are accumulated in the voting space, the more parts we have, the more robust is the evidence of the object localization. Then, the object with few parts tend to perform worse than the ones that are more complex.

Similar background In the database images that we have generated for the experiment, we have used some backgrounds that can be confused with some of the models of the test. Since we allow the matching with transformations of scale and translation, the similar items belonging to the background can have more probabilities to be detected as objects of interest. This fact, combined to the color and shape variations that we have to allow, and can cause the system to retrieve undesired images.

### 6.5.3 Application examples

The developed retrieval system is a tool that can be used in archive search applications. Next we show some qualitative examples in three different sets of images: The covers of the Time magazine, the archives of the newspaper La Vanguardia and the text-based retrieved images from the search engine Google. The image results are showed in a decreasing order according to the query similarity. We have highlighted the relevant results with a green frame so we can have a visual evaluation of the query performance.

#### Google images: text-based plus query-paint

The query-by-paint could be attached as a support tool when we want to search for images in CBIR engines that use text queries. As an example, the web-based searcher Google allows the user to search by text and retrieve images that have been labelled according the query. Pictorial queries cannot always be specified by text and, sometimes, its description is too complex to be matched as a text label. In this case, the user has to provide a more generic query and make the tedious task of browsing through the thumbnail image pages. A query-by-paint could help in reordering automatically the results according to the pictorial representation that the user provides. We have illustrated this approach using three paint examples of the previous experiment: the duck, the smurfin and the clock. We have collected 100 images provided by Google according to the queries: "rubber duck", "smurfin" and "red clock". The examples correspond to the Figures 6.24, 6.25 and 6.26.



**Time Covers collection**

We have used the set of covers of the Time magazine to illustrate the query-by-paint retrieval system. The user may be interested in access to a certain publication from which he does not remember the date. Then, by painting a coarse composition of the elements that compose the cover can retrieve the original image and access to the information. We have used a set of 737 cover images corresponding to 14 years of publications of the Time magazine (from January 1995 to December 2008). This database is public available at the Time web archive <http://www.time.com/time/coversearch>. Next we show some examples of the first 25 images retrieved according to a paint query. The examples correspond to the Figures 6.27, 6.28 and 6.29

**La Vanguardia archives**

Nowadays, more newspapers have their electronic version that can be accessed via Internet. They also have an archive of previous editions that can be browsed by date and even thought by text-based queries. Nevertheless, an additional source of information could be related to the visual appearance of the images. The user could access to a certain article that he doesn't know to which edition belongs but he remembers that contains a certain photograph. Then, the user can retrieve the information using a query paint and does not need to browse the whole content of several newspapers. In the example we present, we have used the set of newspapers that correspond to a whole week of publications of LaVanguardia (<http://www.lavanguardia.es>). Each page of a newspaper is represented by an image, and every newspaper contains an average of 70 pages. The database we have used contains 7 newspapers and total amount of 492 images (pages). The examples correspond to the Figures 6.30, 6.31, 6.32, 6.33 and 6.34.

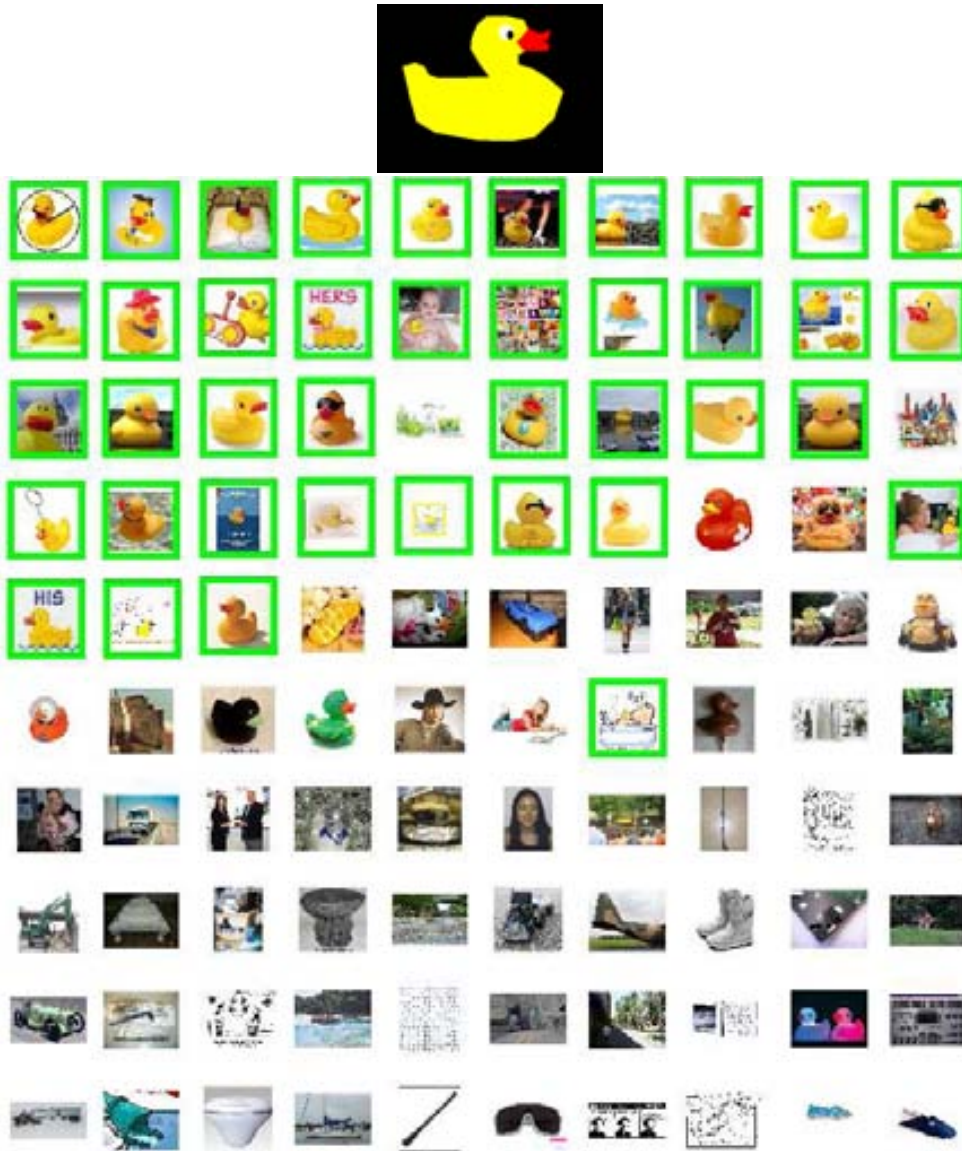


Figure 6.24: Retrieval example from the Google images of the query: "rubber duck".



Figure 6.25: Retrieval example from the Google images of the query: "smurfin".

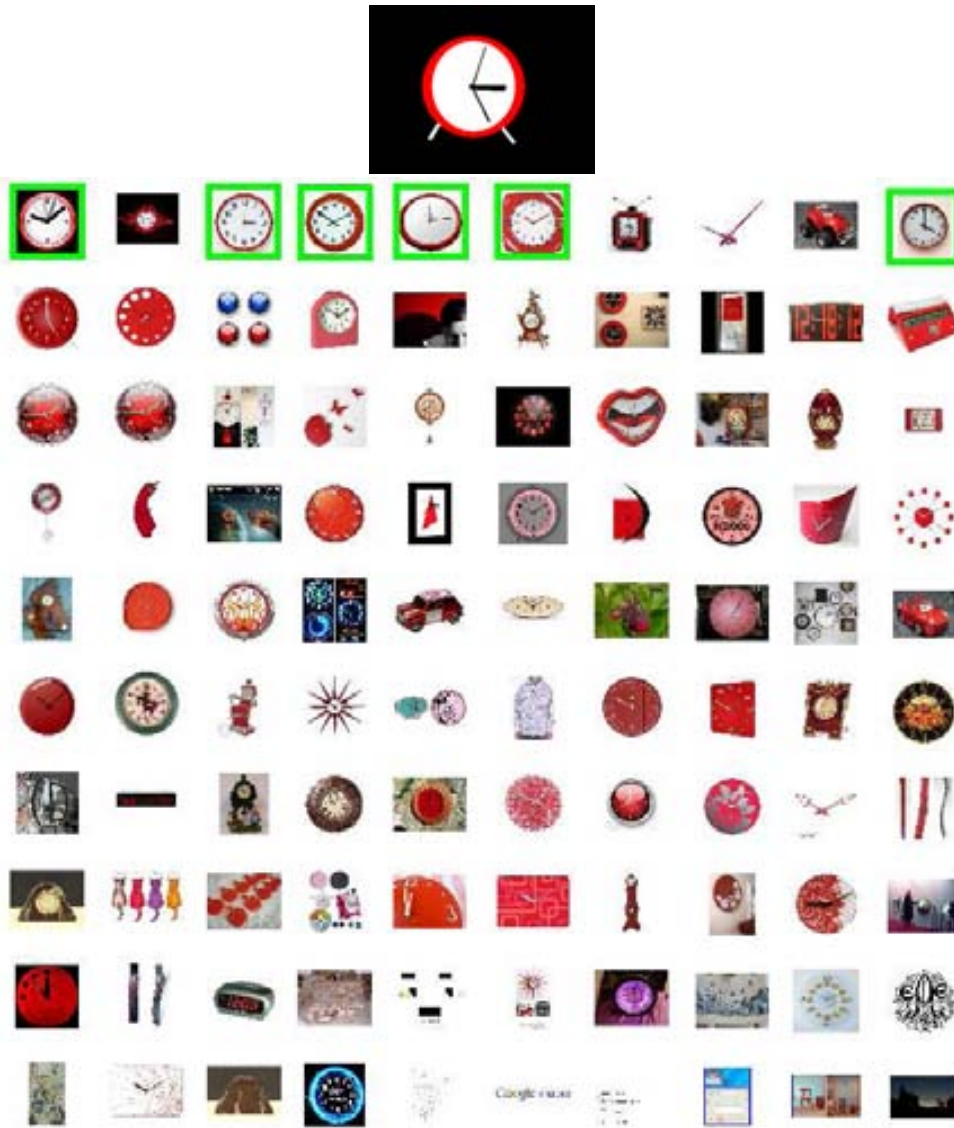


Figure 6.26: Retrieval example from the Google images of the query: "red clock".



Figure 6.27: Retrieval example of the Time Covers Archive.



Figure 6.28: Retrieval example of the Time Covers Archive.



Figure 6.29: Retrieval example of the Time Covers Archive.



Figure 6.30: Retrieval example of the LaVanguardia logo.





Figure 6.31: Retrieval example of and advertisement.

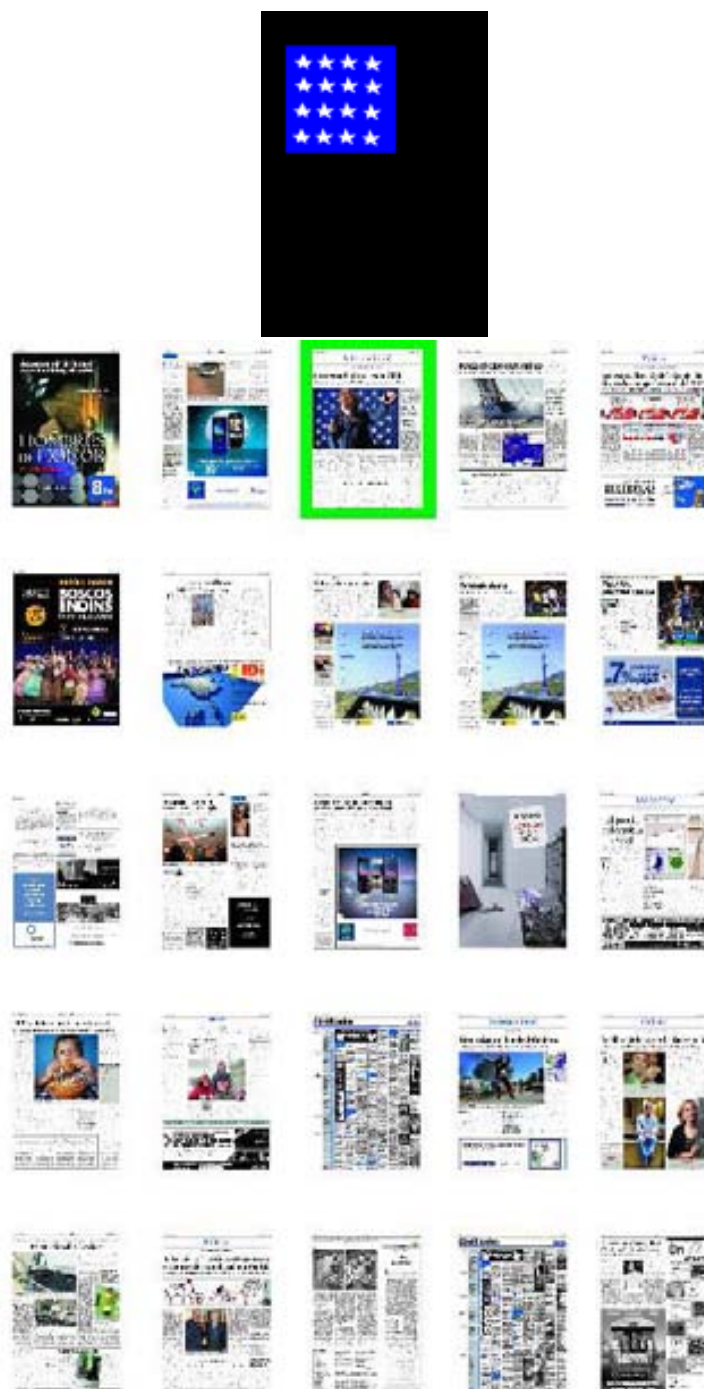


Figure 6.32: Retrieval example of a new that contains a starry background.



Figure 6.33: Retrieval example of a new that contains a London bus.

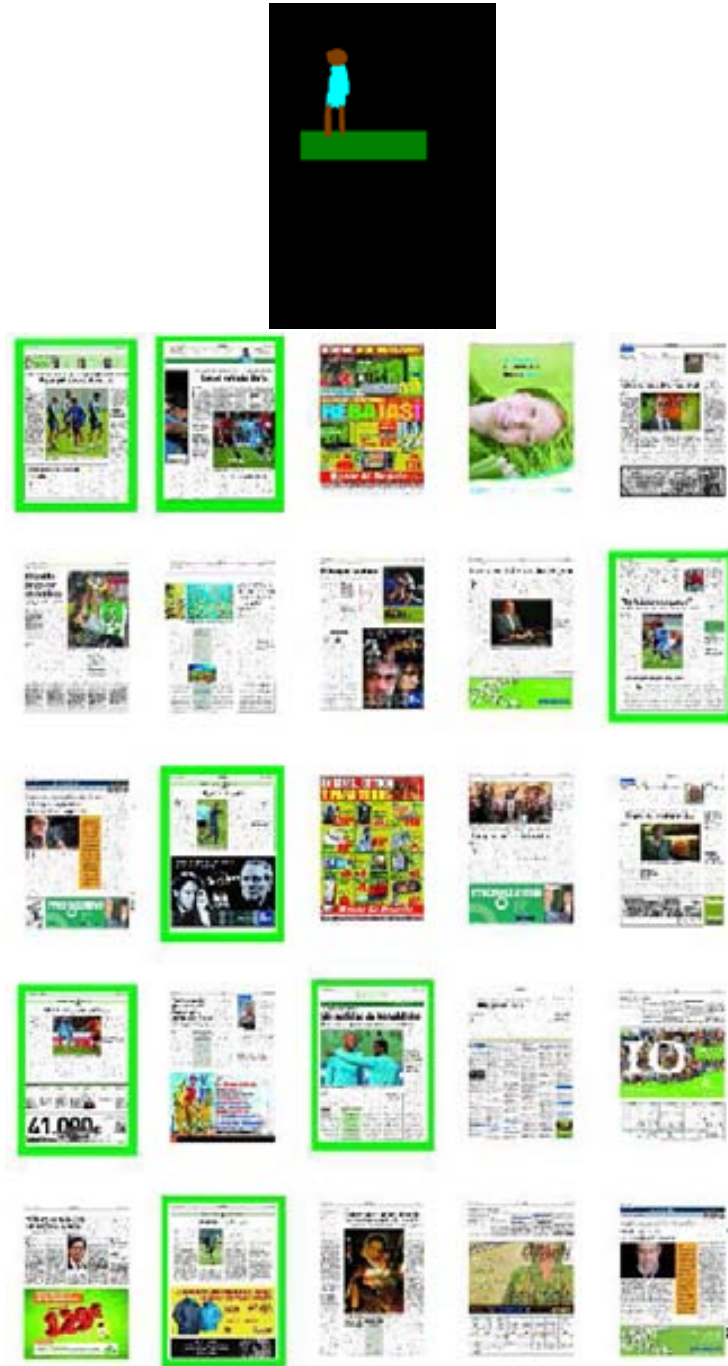
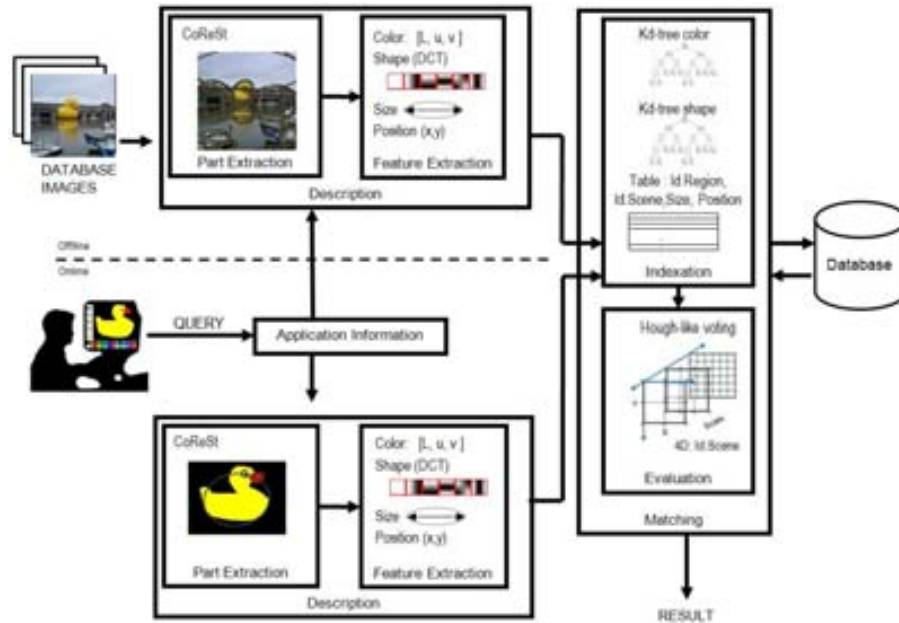


Figure 6.34: Retrieval example of news that contain a football player dressed in blue.

## 6.6 Discussion



**Figure 6.35:** Visual resume of the modules of our approach. Part extraction of the database: regions using the CoReSt algorithm. Features: color, shape, size, position. Feature Indexation: k-d-trees of color and shape; table of size, position and identifiers. Matching evaluation: Hough-like voting in 4D.

### 6.6.1 Conclusions of our approach

We have presented a CBIR where the queries can be composed by the user as a set of color regions. The information of an image, either the query or a scene, is processed to extract those meaningful parts according to the human perception. We have developed an algorithm to extract regions of interest that is inspired in the Gestalt laws of Similarity and Proximity. The algorithm, called CoReSt, explores the color features of an image according to its spatial distribution. This twofold analysis allows the perceptual grouping of elements in a image and gives the segmentation process interesting properties. This way, we can detect entities that belong to be a textured element (a repetitive color pattern) or a plain surface that has been "broken" due to a partial occlusion. The method stands out big and homogeneous regions as well as isolated and contrasted ones. Due to this behavior, we can obtain a final segmentation that comprises a coarse description of the content but that also preserves the meaningful details. To prove the consistency of the perceptual segmentation we have test the method on the Berkeley benchmark of human-based segmentations. The proposal gives a very high degree of freedom on the shape of the output regions and can be

applied in images of general purpose.

Another important point of the CBIR is the election of the features that adapts most to the human-based representation of the query. From the experiment that we have done, we can see that the color is the most descriptive feature according to the characteristics of the queries. Even though this fact could be presupposed from the nature of the images, we observe that in some cases humans can paint the same object with a notable variation in the color representation. The shape can help in the description but taking into account that the shapes of the queries are quite deformed respect to the original ones. In one hand, a very coarse shape descriptor can introduce a high degree of false positives, but in the other hand, a very accurate description can cause the opposite effect and provoke a large amount of false negatives.

The deformation of the query has also consequences in the design of the matching modules of the system. To index the features we have used a k-d-tree that allows the search of regions within a certain degree of tolerance. Then, according to the matching of the regions of the query with the regions of the database, we have proposed a Hough-like matching. To speed up the process we have used a set of predefined accumulation points. These points are distributed in the space to capture the transformations of translation. Moreover, we also have used representative accumulation points to deal with the scale changes.

The system we propose can be applied to a wide variety of contexts. By instance we have shown some examples where a concrete query-by-paint can help in searching for specific objects. Then, we have illustrated it with examples using images extracted from internet, a collection of magazine covers and the digital edition of a newspaper.

### 6.6.2 Conclusions of the query-by-paint paradigm

Query-by-Paint is the query modality that requires major interaction of the user. In this case he has total freedom to create the image of what he is looking for. Despite of the complexity in the process of query creation, the query can have a high expressivity power. The use of color eases the retrieval of certain objects and narrows the amount of matching options between the regions of the query and the scenes. The Query-by-Paint can outperform the text based systems in those cases where a textual label is difficult to assign to an image. By instance, it could help in those cases where the concept we search is difficult to express in words (e.g. an abstract picture). Moreover it could also help in retrieving images of concrete objects which textual description should be too detailed (e.g. a user can remember a concrete image of the database and he wants to retrieve it).

From an implementation point of view, we observe that the CBIR systems that use Query-by-Paint have to deal with severe distortions of the query representation. The first observation is that the human representation can vary considerably according between the users. Both the color and the shape can suffer from high variations according the human representation. We have made an experiment to model which

are the most suitable features to describe a human paint. We have found that the best combination for the feature representation is done by the mean Luv value of the regions and their shape encoding using the DCT. Nevertheless, more experiments could be done using a higher amount of test images and analyzing other important factors such as the illumination of the scenes.

Finally we observe that, in a similar way as the Query-by-Sketch, the Query-by-Paint also needs for tools to provide a flexible matching that copes with the severe deformations of the human based representation. Finding the evidence of a query in a scene becomes a hard problem as the number of parts of the object decreases. In the case of the paintings the number of parts in the query image uses to be low. Then, it is important to obtain an accurate description of the parts to make them as much discriminant as possible. In the system we propose we have avoided to make a final evaluation of the matching and we have relied on the accumulation of the Hough voting. Nevertheless, depending on the constraints of the time response of the application, a more accurate matching could be done in order to improve the results.





# Chapter 7

## Conclusions

Nowadays, the massive use of the new technologies generates large volumes of images. The need of managing all this information motivates the creation of CBIR systems of general purpose. This way, we have oriented this thesis in the analysis the CBIR systems that use pictorial queries. We have chosen to focus in the pictorial queries because they represent a step further from the classical textual annotation. Thus, pictorial queries express the information of an image in a universal way according to visual features.

We have defined a taxonomy of the retrieval systems according to de degree personalization that the user can provide when defining the query. Then, four types of pictorial queries are identified: query-by-image-selection, query-by-iconic-composition, query-by-sketch or query-by-paint.

Next we expose the conclusions that we have extracted from the analysis of the pros and cons of every query paradigm. First, we present the observations related to the influence that each query paradigm exerts on the performance of the system. Next we analyze the widespread usage of the CBIR systems that use pictorial queries. Finally, we summarize the contributions of this thesis and we present the future work.

### **7.1 Influence of the query paradigm in the design and the performance of a CBIR system**

The query formulation is the first phase of the retrieval process. Then, we observe that all the following phases, the internal process and the final results, depend on the query paradigm that is used in the first step.

From the point of view of the query formulation, we realize that each paradigm allows a certain degree of freedom in order to personalize 'what the user is looking for'. A certain level of personalization requires an equivalent degree of activity in the process of formulation. The activity can vary from a simple image selection to the

complete creation of the input image. Hence, the role of the user in the query step forces the system to deal with a human-based representation of the image content. We have summarized the concrete relations between the user activity and the query personalization in the Table 7.1.

Query paradigm	User activity	Personalization and expressivity
Q-Selection	Browse and select	No human influence
Q-Iconic	Click on iconic representation	Composition of elements with a semantic charge
Q-Sketch	Draw	Characterization of an object by the shape
Q-Paint	Draw and paint	Characterization by the shape and the color

**Table 7.1**

A HIGHER ACTIVITY IN THE QUERY FORMULATION IMPLIES A HIGHER DEGREE OF PERSONALIZATION.

The influence of the humans in the query formulation implies variations of the visual features that define the query. Then, the matching process has to cope with these deformations in order to make the comparison with the database information. Hence, similarity distances, indexing structures and spatial evaluation systems have to deal with high tolerance ranges to perform the matching. The relaxation of the similarity requirement helps to provide a good performance according to the recall. Nevertheless, the precision can decrease because the comparison tolerance allows including false positives in the result. The relations between the human-based representation and the degree of retrieval precision are shown in the Table 7.2.

Query paradigm	Description precision	Matching precision
Q-Selection	The same nature of images allow the same process of feature extraction	Depending on the application requirements
Q-Iconic	Depends of a generic representation according to a model	Depends on a goodness of the model description
Q-Sketch	Has to be flexible enough to tolerate a certain degree of the shape deformation	Needs to evaluate the global shape deformations
Q-Paint	Has to cope with shape and color variations	Needs to evaluate the global deformations and combine the shape and color features

**Table 7.2**

THE HUMAN REPRESENTATION INCORPORATES A DEFORMATION OF THE QUERY THAT CAN DECREASE THE PRECISION OF THE RETRIEVAL RESULTS.

## 7.2 Usage of pictorial queries in CBIR systems of general purpose

If we collect the sequence of observations of the previous section, we realize that the retrieval process has to face severe difficulties when the user is involved in the creation of the pictorial query. Then, the usage of the human based queries is compromise between the need of expressivity power and the ambit of the application.

The query-by-selection is the best option when the retrieval system is related to a specific application that deals with a finite set of queries. In this paradigm, the selection of a suitable image means a key step that, when it is not satisfied, can lead to unexpected results. In the chapter 3 we have exemplified it with an application that deals with a set of logos and predefined images.

In the case where the system deals with object categories, the most suitable option is to represent them using an iconic query. This kind of paradigm is the least suitable to be applied in systems of general purpose. The construction of a generic model of an object category is hard to obtain since it has to cope with all the possible variations of its instances.

Finally, the query-by-sketch and the query-by-paint are suitable to describe objects that can be represented with a simple illustration according their shape and their color. As we have seen in the experiments of the chapters 5 and 6, this kind of queries comprise severe deformations. Thus, the description of two instances of the same object can be defined by different representations of the visual features. This way, the sketch-based and the paint-based paradigms are those that imply most difficulties in the retrieval process.

According to the above mentioned 'compromise between the expressivity and the application area', we pose that a suitable way to obtain this balance could comprise the mix of several media in the query process. This way, the commonly used textual queries could be combined with the pictorial ones to obtain a more powerful CBIR system. In one hand, textual annotation can help in narrowing the domain area of the database images and, in the other hand visual features can help in the search of concrete images which description is too complex to be defined as a label. The usage of pictorial queries could also help in improving the precision of those textual queries hat can have different meanings (the polysemy phenomena). This collaboration between different media of queries seems to be a future trend in the CBIR research. By instance, Google image search has incorporated a very simple tool to select the color of the images. This visual feature acts like a filter of the text-based retrieved images and helps the user in the search process. This point of view could be exploited by using more complex descriptions that were focused not in the whole image but in an object inside a scene. Following this trend, in the chapter 6 we have posed several examples that use a collection of images obtained from the web according to a textual query. Then, pictorial queries use concrete visual-based features to refine the query result.

In this thesis we have presented several proposals to go on with the research related to the internal processes of the CBIR systems that use pictorial queries. Next we present the contributions according to each of the query paradigms that we have analyzed.

### 7.3 Contributions

We have proposed four retrieval systems, each one related to a query paradigm. An important key of the CBIR we propose is that they are in the context of a retrieval system of general purpose. This fact provides two important restrictions: The first one is that the descriptors should not be based in any intensive learning process of a specific object. The second constraint poses that the features that describe the scenes have to be indexable.

Our contributions are mainly focused in the image description process even though we have also proposed some improvements in the matching step. The Table 7.3 summarizes the processes of the proposed retrieval systems.

Query paradigm	Description		Matching	
	Parts	Features	Indexing	Evaluation
Q-Selection	Contour based segmentation	<b>Multi-scale triangulation</b>	Sequential structure	No evaluation
Q-Iconic	<b>Graph based region modelling</b>	Label annotation	A field in a database table	No evaluation
Q-Sketch	Contour line approximation	<b>Geometric constraints according to Primitive structures</b>	Hash table	<b>Generalized Hough-like voting combined with an oriented Chamfer distance</b>
Q-Paint	<b>Perceptual segmentation from color spatial properties</b>	<b>Color and shape (Mean Luv and DCT)</b>	k-d-tree structures for the color and the shape	<b>Generalized Hough-like voting with predefined accumulators</b>

**Table 7.3**

STRATEGIES OF THE PROPOSED CBIR SYSTEMS. THE MAIN CONTRIBUTIONS ARE HIGHLIGHTED IN BOLD TYPE.

Let us overview the contributions related to the description process:

- Scene description according to an selected query

We have proposed a descriptor to encode the whole scene information in a compact manner. We have used a Delaunay triangulation to obtain the spatial arrangement of the zones of a scene. Then, we also have taken into account a space-scale analysis to provide more meaningfulness to the large zones of the image than the details. We have observed that the multi-scale description improves the discriminatively power of the descriptor.

- Scene description according to an iconic composition

Our contribution consists in modelling the iconic objects according to an attributed graph of regions. We have proposed a matching method that deals with a set of cost functions to compare the ideal graph of the models with the content of a scene. The method applies an over-segmentation of the image and then uses these cost functions to identify the features of the scene with the features of the iconic models.

- Scene description according to a sketch representation

To describe the content of an image we use a polygonal approximation of the boundary and we treat every vector as a potential part of interest. We have proposed a descriptor based in the geometric similarity between the local arrangement of vectors and a set of primitive structures. This way, our image descriptor do not attempt to extract the whole boundary of the object. Instead it uses a part-based strategy to cope with the clutter of the scene and the partial deformations.

- Scene description according to a paint representation

A reliable process of extract information from a scene should imitate the human based representation. Thus we have developed an automatic system to extract the image information that means to reproduce the human perception. In order to do that, we have to take into consideration that the human based representation of an image has not a single solution. Then, different persons can describe the same object with a different set of regions. Nevertheless, from a psychological point of view, there exists many studies that attempt to model the rules of the human perception. The Gestalt School has provided one of the most relevant works in this field.

Thus, we have proposed an automatic system that is inspired in the Gestalt laws of proximity and similarity. The process analyzes the color of the pixels and their spatial positions to obtain a set of regions that are detected to be perceptually meaningful. This set of regions does not provide a unique solution of the image segmentation. Otherwise, in order to emulate the human based

description, the resulting regions that can be even overlapped and can also be made of several unconnected components.

Once the image parts are identified, we have to extract the features that describe them. We have made an experiment testing the combination of six descriptors: three to encode the color and three to encode the shape. We have found out that the best results of the experiment are provided by the combination of the mean Luv values of their color and the DCT of their shapes.

In addition to the description contributions we have made some approaches to improve the matching step. The most relevant are summarized as follows:

- Matching of a query sketch with a scene

The query sketch contains shape deformations that difficulties the matching phase. For these reason we have performed a fuzzy evaluation of the similarity of the descriptors. The similarity values are accumulated in a voting space by the use of a general-Hough strategy. To cope with the sketch deformations, we have adapted the Chamfer distance as an evaluation strategy. This kind of evaluation uses the whole boundary information and helps to overcome a defective line approximation in the description phase.

- Matching of a query point with a scene

The matching process has to deal with severe variations according to the painting styles of the users. Then, given a feature of the query, we have to provide enough tolerance to retrieve the similar features of the scenes. To index the features of the images we have used two k-d-trees. Then, in a similar way as the query-by-sketch, we have proposed to use a matching inspired in the generalized-Hough-transform. Nevertheless, in order to avoid an intensive search of the voting space, we have used a set of predefined voting locations at several scales.

Future lines of research suggest some improvements to expand the actual research.

## 7.4 Future lines of research

During this thesis we have focused in the procedures of a retrieval system according the paradigm of query creation. Nevertheless we have not analyzed the optional module related to the relevance feedback. We propose to follow the research with simple strategies that make use of the output images.

The first option related to the system feedback could be the possibility of applying an incremental query. The user, would not need to create the whole query before submitting it to the system. Otherwise, he could paint a first region, submit it to the system, and check the resulting images. If he is not satisfied with the result, he could add another region and keep on with the loop.

Moreover, it could be interesting to mix several paradigms of the pictorial queries. The combination of query types would offer more possibilities to refine retrieval results. By instance we could start a query by using a paint-based-query. Then, if we detect a meaningful image in the result we could reuse the output image in the retrieval process. Obviously, the usage of several types of queries would imply an additional load to the system because it should maintain a larger volume of descriptors.

Another improvement could be related with the query-by-selection. Notice that the CBIR that we have presented describes the whole content of a scene. Then, we could also expand this kind of paradigm by adding the modality of partial selection. This facility could give a notable freedom to the user in order to formulate a query. Then, he could crop a part of an image where an object of interest appears. The internal process of the CBIR that use sketch-based queries and the paint-based queries, could be applied to construct a CBIR using partial-selection. Both strategies allow a partial matching. Then, we could use one or another according if we are interested in the shapes or the color of the cropped object of interest of the query.

Moreover, the research could be expanded with additional experiments according to the human perception. An interesting point could be centered in the study of the image representation according to the visual memory of the users. We could perform an experiment to compare variations between the representation of the same object. The user would be asked to create a query image while he sees he model and also would be asked to repeat the process based on his remains. Finally, we could construct a larger database of paints in order to make a deeper analysis of the human representation. More descriptors related to the color and the shape could also be tested.





# Bibliography

- [AB94] R. Adams and L. Bischof. Seeded region growing. *PAMI*, 16(6):641–647, June 1994.
- [ACS07] M. Anelli, L. Cinque, and Enver Sangineto. Deformation tolerant generalized hough transform for sketch-based image retrieval in complex scenes. *Image Vision Comput.*, 25(11):1802–1813, 2007.
- [ATY<sup>+</sup>95] Y. Alp Aslandogan, Chuck Thier, Clement T. Yu, Chengwen Liu, and Krishnakumar R. Nair. Design, implementation and evaluation of score (a system for content based retrieval of pictures). In *ICDE '95: Proceedings of the Eleventh International Conference on Data Engineering*, pages 280–287, Washington, DC, USA, 1995. IEEE Computer Society.
- [Bal81] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [BDBP00] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia (TMM)*, 2(4):225–239, December 2000.
- [Ben75] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communication on the ACM*, 18(9):509–517, 1975.
- [BKK96] Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. The x-tree : An index structure for high-dimensional data. In *VLDB*, pages 28–39, 1996.
- [Bla80] Albrecht Blaser, editor. *Data Base Techniques for Pictorial Applications, Florence, Italy, June 20-22, 1979, Proceedings*, volume 81 of *Lecture Notes in Computer Science*. Springer, 1980.
- [BM72] Rudolf Bayer and Edward M. McCreight. Organization and maintenance of large ordered indices. *Acta Inf.*, 1:173–189, 1972.
- [BM00] S. Belongie and J. Malik. Matching with shape context, 2000.

- [BMPT98] Alberto Del Bimbo, Mauro Mugnaini, Pietro Pala, and F. Turco. Visual querying by color perceptive regions. *Pattern Recognition*, 31(9):1241–1253, 1998.
- [BP97] Alberto Del Bimbo and Pietro Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):121–132, 1997.
- [BS94] C. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IP*, 3(2):162–177, March 1994.
- [BT05] Guillaume Bouchard and Bill Triggs. Hierarchical part-based visual object categorization. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:710–715, 2005.
- [BTG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.
- [BTTJ05] Andre G. R. Balan, Agha J. M. Traina, and Caetano Traina Jr. Fractal analysis of image textures for indexing and retrieval by content. In *CBMS '05: Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*, pages 581–586, Washington, DC, USA, 2005. IEEE Computer Society.
- [BVB05] Robert Benavente, Maria Vanrell, and Ramon Baldrich. A data set for fuzzy colour naming. *Color Research & Application*, 31(1):48–56, December 2005.
- [BW77] Tenenbaum J. M. Bolles R. C. Barrow, H. G. and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical Report 153, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, 1977.
- [Can86] J. Canny. A computational approach to edge detection. *PAMI*, 8(6):679–698, November 1986.
- [CCK84] Margaret Chock, Alfonso F. Cardenas, and Allen Klinger. Database structure and manipulation capabilities of a picture database management system (picdms). *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(84):484–492, 1984.
- [CDF<sup>+</sup>04] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.
- [CFB04] Michel Crucianu, Marin Ferecatu, and Nozha Boujemaa. Relevance feedback for image retrieval: a short survey. In *In State of the Art in Audiovisual Content-Based Retrieval, Information Universal Access and Interaction including Datamodels and Languages (DELOS2 Report*, 2004.

- [CFH05] David Crandall, Pedro Felzenszwalb, and Daniel Huttenlocher. Spatial priors for part-based recognition using statistical models. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:10–17, 2005.
- [CG99] Scott Cohen and Leonidas Guibas. The earth mover’s distance under transformation sets. In *In Proceedings, 7th International Conference on Computer Vision*, pages 1076–1083, 1999.
- [CGM02] C.M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *ICPR*, pages IV: 150–155, 2002.
- [CL06] Gustavo Carneiro and David Lowe. Sparse flexible models of local features. In *ECCV (3)*, pages 29–43, 2006.
- [CM99a] Kaushik Chakrabarti and Sharad Mehrotra. High dimensional feature indexing using hybrid trees. In *Proc. of the 15th IEEE International Conference on Data Engineering (ICDE, 1999)*.
- [CM99b] D. Comaniciu and P. Meer. Mean Shift Analysis and Applications. In *Proceedings of the IEEE ICCV*, pages 1197–1203, Kerkyra, Greece, 1999.
- [CM00] Munoz X. Freixenet J. Cufi, X. and J. Marti. A concurrent region growing algorithm guided by circumscribed contours. In *ICPR00*, pages Vol I: 432–435, 2000.
- [CNM05] A. Chalechale, G. Naghdy, and A. Mertins. Sketch-based image matching using angular partitioning. *IEEE Transactions on Systems, Man, Cybernetics - Part A: Systems and Humans*, 35(1):28–41, January 2005.
- [COBP+99] Kaushik Chakrabarti, Michael Ortega-Binderberger, Kriengkrai Porkaew, Peng Zuo, and Sharad Mehrotra. Similar shape retrieval in mars. Technical report, IEEE Int. Conf. On Multimedia and Expo, 1999.
- [COBPM00] K. Chakrabarti, M. Ortega-Binderberger, K. Porkaew, and S. Mehrotra. Similar shape retrieval in mars. volume 2, pages 709–712, 2000.
- [CPZ97] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB’97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, pages 426–435. Morgan Kaufmann, 1997.
- [CSY87] S.K. Chang, Q.Y. Shi, and C.W. Yan. Iconic indexing by 2-d strings. *PAMI*, 9(3):413–428, May 1987.

- [CSY88] Jungert E. Chang S.K. and Li Y. Representation and retrieval of symbolic pictures using generalized 2d strings. Technical report, Unicersity of Pittsburg, 1988.
- [CTB<sup>+</sup>99] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *In Third International Conference on Visual Information Systems*, pages 509–516. Springer, 1999.
- [Dav93] M. Davis. Media streams: an iconic visual language for video annotation. pages 196–202, August 1993.
- [Deb03] Sagarmay Deb. *Multimedia Systems and Content-Based Image Retrieval*. Information Resources Press, Arlington, VA, USA, 2003.
- [Del34] B. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, (7):793–800, 1934.
- [DG06] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.
- [DJLW08] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, April 2008.
- [DKN08] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: an experimental comparison. *Inf. Retr.*, 11(2):77–107, 2008.
- [DMK<sup>+</sup>01] Y. Deng, B.S. Manjunath, C. Kenney, M.S. Moore, and H. Shin. An efficient color representation for image retrieval. *IP*, 10(1):140–147, January 2001.
- [DRD97] Madirakshi Das, Edward M. Riseman, and Bruce A. Draper. Focus: Searching for multi-colored objects in a diverse image database. In *IEEE Conf. on Comp. Vis. and Pattern Recognition*, pages 756–761, 1997.
- [dSB99] Pires Rui Luís V. P. M. de Vleeschauwer D. de Smet, P. and Ignace Bruyland. Activity driven nonlinear diffusion for color image watershed segmentation. *Journal of Electronic Imaging*, (8):270–278, July 1999.
- [DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.
- [E.88] Jungert E. Extended symbolic projection used in a knowledge structure for spatial reasoning. In *4th BPRA Conf. on Pattern Recognition*, March 1988. , Springer, Cambridge.

- [EBS96] J P Eakins, J M Boardman, and K Shields. Retrieval of trademark images by shape feature - the artisan project. In *In IEE Colloquium on Intelligent Image Databases*, pages 101–109, 1996.
- [EG99] John P. Eakins and Margaret E. Graham. Content-based image retrieval: A report to the jisc technology applications programme. Technical report, Institute for Image Data Research, University of Northumbria at Newcastle, 1999.
- [EG00] John P. Eakins and Margaret E. Graham. Content based image retrieval. technical report. Technical Report JTAP-039, JISC Technology Application Program, Newcastle upon Tyne, 2000.
- [EM92] P.G.B. Enser and C.G. McGregor. Analysis of visual information retrieval queries. *Personal Communication*, August 1992.
- [ESB96] John P. Eakins, Kevin Shields, and Jago Boardman. ARTISAN – a shape retrieval system based on boundary family indexing. In *Storage and Retrieval for Still Image and Video Databases IV. Proceedings SPIE 2670*, pages 17–28, 1996.
- [ESM<sup>+</sup>91] A. Etemadi, J.P. Schmidt, G. Matas, J. Illingworth, and J.V. Kittler. Low-level grouping of straight line segments. pages 119–126, 1991.
- [EZ96] James H. Elder and Steven W. Zucker. Computing contour closure. In *In Proc. 4th European Conference on Computer Vision*, pages 399–412, 1996.
- [Faw06] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.
- [FB80] M. A. Fishler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Tech report 213, AI Center, SRI International*, 1980.
- [FBRJ04] Manuel Fonseca, Bruno Barroso, Pedro Ribeiro, and Joaquim Jorge. Sketch-based retrieval of clipart drawings. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 429–432, New York, NY, USA, 2004. ACM.
- [FE73] M.A. Fischler and R.A. Elschlager. The representation and matching of pictorial structures. *Computer*, C-22:46–53, January 1973.
- [Ff05] Li Fei-fei. A bayesian hierarchical model for learning natural scene categories. In *In CVPR*, pages 524–531, 2005.
- [FFFFP03] Li Fei-Fei, Rob Fergus, and Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2003. IEEE Computer Society.

- [FH05] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61:2005, 2005.
- [FK91] H. Fujii and R. R. Korfhage. Features and a model for icon morphological transformation. In *Proc. of the 1991 IEEE Workshop on Visual Languages*, pages 240–245, Kobe, Japan, 1991.
- [FMM03] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:54, 2003.
- [For07] Per-Erik Forssén. Maximally stable colour regions for recognition and matching. In *IEEE Conference on CVPR*, Minneapolis, USA, June 2007.
- [FPZ03] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *In CVPR*, pages 264–271, 2003.
- [FPZ05] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *2005 Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 380–387, June 2005.
- [FR89] AC. Faloutsos and S. Roseman. Fractals for secondary key retrieval. In *Proc. 8th Symposium on Principles of Database Systems*, pages 247–252, 1989.
- [FSA96] Charles Frankel, Michael J Swain, and Vassilis Athitsos. Webseer: An image search engine for the world wide web. Technical report, Chicago, IL, USA, 1996.
- [FSN<sup>+</sup>95] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, September 1995.
- [FTG06] Vittorio Ferrari, Tinne Tuytelaars, and Luc J. Van Gool. Object detection by contour segment networks. In *ECCV (3)*, pages 14–28, 2006.
- [Gav98] D M Gavrilu. Multi-feature hierarchical template matching using distance transforms. *Pattern Recognition, International Conference on*, 1:439, 1998.
- [GBS05] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam library of object images. *Int. J. Comput. Vis.*, 61(1):103–112, 2005.

- [GD04] Kristen Grauman and Trevor Darrell. Fast contour matching using approximate earth mover's distance. In *CVPR (1)*, pages 220–227, 2004.
- [GG98] Volker Gaede and Oliver Gunther. Multidimensional access methods. *ACM Computing Surveys*, 30:170–231, 1998.
- [GIK03] Abdul Ghafoor, Rao Naveed Iqbal, and Shoab Ahmed Khan. Image matching using distance transform. In Josef Bigün and Tomas Gustavsson, editors, *SCIA*, volume 2749 of *Lecture Notes in Computer Science*, pages 654–660. Springer, 2003.
- [GK90] Calvin C. Gotlieb and Herbert E. Kreyszig. Texture descriptors based on co-occurrence matrices. *Comput. Vision Graph. Image Process.*, 51(1):70–86, 1990.
- [GLMZ03] J. M. González-Linares, Nicolás Guil Mata, and Emilio L. Zapata. An efficient 2d deformable objects detection and location algorithm. *Pattern Recognition*, 36(11):2543–2556, 2003.
- [Gos85] A. Goshtasby. Description and discrimination of planar shapes using shape matrices. *T-PAMI*, 7:738–743, 1985.
- [GP90] M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization. *Graphics Gems I*, pages 287–293, 1990.
- [GR03] G. Gagaudakis and P. L. Rosin. Shape measures for image retrieval. *Pattern Recogn. Letters*, 24(15):2711–2721, 2003.
- [Gro99] MPEG Video Group. *Description of Core Experiments for MPEG-7 Color/Texture Descriptors, ISO/MPEGJCT1/SC29/WG11 MPEG98/M2819*. July 1999.
- [GS99] Theo Gevers and Arnold W. M. Smeulders. The pictoseek www image search system. In *In Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 264–269. IEEE, 1999.
- [Gut84] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In Beatrice Yormark, editor, *SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984*, pages 47–57. ACM Press, 1984.
- [Har92] Donna Harman. Relevance feedback and other query modification techniques. pages 241–263, 1992.
- [HH98] Benoit Huet and Edwin R. Hancock. Object recognition from large structural libraries. In *SSPR/SPR*, pages 190–199, 1998.
- [HJ96] P.W. Huang and Y.R. Jean. Spatial reasoning and similarity retrieval for image database systems based on rs-strings. *PR*, 29:2103–2114, 1996.

- [HKM<sup>+</sup>97] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. *In Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec.*, pages 762–768, 1997.
- [HKR93] D.P. Huttenlocher, G.A. Klanderman, and W.A. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
- [HP77] S.L. Horowitz and T. Pavlidis. Picture segmentation by a directed split and merge procedure. *In CMetImAly77*, pages 101–111, 1977.
- [HP96] Jianying Hu and Theo Pavlidis. A hierarchical approach to efficient curvilinear object searching. *Comput. Vis. Image Underst.*, 63(2):208–220, 1996.
- [HS85] R.M. Haralick and L.G. Shapiro. Image segmentation techniques. *CVGIP*, 29(1):100–132, January 1985.
- [HSD73] R. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), 1973.
- [Hu62] M.K. Hu. Visual pattern recognition by moment invariants. IT-8:179–187, 1962.
- [HU90] Daniel P. Huttenlocher and Shimon Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.
- [IA02] Qasim Iqbal and J. K. Aggarwal. Cires: A system for content-based retrieval in digital image libraries. *In in Invited Session on Content-based Image Retrieval: Techniques and Applications, 7th International Conference on Control Automation, Robotics and Vision (ICARCV*, pages 205–210, 2002.
- [Jai92] R.C. Jain. NSF workshop on visual information management systems: workshop report. In W. Niblack, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 1908 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 198–218, April 1992.
- [JV96] Anil K. Jain and Aditya Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29:1233–1244, 1996.
- [JV98] Anil Jain and Aditya Vailaya. Shape-based retrieval: A case study with trademark image databases. *Pattern Recognition*, 31:1369–1390, 1998.
- [KB01] Timor Kadir and Michael Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.



- [KC04] Chia-Chen Kuo and Ming-Syan Chen. Dds: an efficient dynamic dimension selection algorithm for nearest neighbor search in high dimensions. In *ICME*, pages 999–1002, 2004.
- [Kei94] L Keister. User types and queries: impact on image access systems. *Proceedings of the ASIS 57th Annual Meeting, Alexandria, VA*, 31:7–22, August 1994.
- [KIK91] S. Kaneko, H. Ikemoto, and Y. Kusui. Approach to designing easy-to-understand icons. In *Proc. of the 1991 IEEE Workshop on Visual Languages*, pages 246–253, Kobe, Japan, 1991.
- [KJB96] K. Karu, A.K. Jain, and R.M. Bolle. Is there any texture in the image. In *ICPR96*, page B94.3, 1996.
- [KJS97] R. Kurniawati, J. S. Jin, and J. A. Shepherd. Techniques for supporting efficient content-based retrieval in multimedia databases. *The Australian Computer J*, 29:122130, 1997.
- [KMN98] Lance M. Kaplan, Romain Murenzi, and Kameswara Rao Namuduri. Fast texture database retrieval using extended fractal features. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 162–175, 1998.
- [Kno75] Gary D. Knott. Hashing functions. *Comput. J.*, 18(3):265–278, 1975.
- [Knu97] Donald Ervin Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison Wesley, Reading, Massachusetts, 2nd edition, April 1997.
- [Koe84] J.J. Koenderink. The structure of images. In *Biological Cybernetics*, pages 363–370, 1984.
- [KR98] Michael Kliot and Ehud Rivlin. Invariant-based shape retrieval in pictorial databases. In *ECCV (1)*, pages 491–507, 1998.
- [KY01] E. Kasutani and A. Yamada. The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 674–677 vol.1, 2001.
- [LC02a] Man-Wai Leung and Kwok-Leung Chan. Object-based image retrieval using hierarchical shape descriptor. In *CIVR: Proceedings of the International Conference on Image and Video Retrieval*, pages 165–174, London, UK, 2002. Springer-Verlag.
- [LC02b] Wing Ho Leung and Tsuhan Chen. User-independent retrieval of free-form hand-drawn sketches. In *In Proc. of the IEEE ICASSP*, pages 2029–2032. IEEE Press, 2002.

- [Leu03] Howard Wing Ho Leung. *Representations, feature extraction, matching and relevance feedback for sketch retrieval*. PhD thesis, Pittsburgh, PA, USA, 2003. Adviser-Chen, Tsuhan.
- [Lew00] Michael S. Lew. Next-generation web searches for visual content. *Computer*, 33(11):46–53, 2000.
- [LF93] A. Laine and J. Fan. Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1186–1191, 1993.
- [LG94] Tony Lindeberg and Jonas Gårding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. In *ECCV '94: Proceedings of the third European conference on Computer vision (vol. 1)*, pages 389–400, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
- [Lin93] Tony Lindeberg. *Scale-Space Theory in Computer Vision (The International Series in Engineering and Computer Science)*. Springer, December 1993.
- [Lin96] T. Lindeberg. Scale-space: A framework for handling image structures at multiple scales. In *Proceedings of CERN School of Computing*, pages 8–21, Egmond aan Zee, The Netherlands, September 1996.
- [Lin98] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, November 1998.
- [LK01] Dong-Ho Lee and Hyoung-Joo Kim. A fast content-based indexing and retrieval technique by the shape information in large image database. *Journal of Systems and Software*, 56(2):165–182, 2001.
- [LKW] H.M. Lee, J. Kittler, and K.C. Wong. Generalised hough transform in object recognition. *ICPR*, C:285–2–89.
- [LM99a] L. Lucchese and S.K. Mitra. Unsupervised low-frequency driven segmentation of color images. In *ICIP99*, page 27AP1, 1999.
- [LM99b] L. Lucchese and S.K. Mitra. Unsupervised segmentation of color images based on k-means clustering in the chromaticity plane. In *CBAIVL99*, 1999.
- [LM01] L. Lucchese and S.K. Mitra. Colour segmentation based on separate anisotropic diffusion of chromatic and achromatic channels. *VISP*, 148(3):141–150, June 2001.
- [LM02] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP 2002*, pages 900–903, 2002.

- [LN98] Carson Kai-Sang Leung and Raymond T. Ng. Multiresolution subimage similarity matching for large image databases. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 259–270, 1998.
- [LN04] Jie Luo and Mario A. Nascimento. Content-based sub-image retrieval using relevance feedback. In *MMDB '04: Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 2–9, New York, NY, USA, 2004. ACM.
- [Low99] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.
- [Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [LP96] Fang Liu and Rosalind W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(7):722–733, 1996.
- [LS99] Guojun Lu and Atul Sajjanhar. Region-based shape representation and similarity measure suitable for content-based image retrieval. *Multimedia Syst.*, 7(2):165–174, 1999.
- [LSLF05] Shuang Liang, Zheng-Xing Sun, Bin Li, and Gui-Huan Feng. Effective sketch retrieval based on its contents. volume 9, pages 5266–5272, August 2005.
- [LWY95] Hsin-Chih Lin, Ling-Ling Wang, and Shi-Nine Yang. Color image retrieval based on hidden markov models. *Image Processing, International Conference on*, 1:342, 1995.
- [MA07] Sébastien Macé and Éric Anquetil. Design of a pen-based electric diagram editor based on context-driven constraint multiset grammars. In *HCI (2)*, pages 418–428, 2007.
- [Mar88] K. Markey. Access to iconographical research collections. *Library Trends*, 37(2):154–174, 1988.
- [Mar02] David Royal Martin. *An empirical approach to grouping and segmentation*. PhD thesis, 2002. Chair-Malik, Jitendra and Chair-Patterson, David.
- [MBNW99] Jamal Malki, Nozha Boujemaa, Chahab Nastar, and Re Winter. Region queries without segmentation for image retrieval by content. In *In Third International Conference on Visual Information Systems (Visual'99)*, pages 115–122, 1999.
- [MC98] E. Mathias and A. Conci. Comparing the influence of color spaces and metrics in content-based image retrieval. *Computer Graphics and Image Processing, Brazilian Symposium on*, 0:371, 1998.

- [MCMP02] J Matas, O Chum, U Martin, and T Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the BMVC*, volume 1, pages 384–393, London, 2002.
- [MFTM01] D.R. 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. Technical report, EECS Department, University of California, Berkeley, 2001.
- [MG95] Rajiv Mehrotra and James E. Gary. Similar-shape retrieval in shape data management. *Computer*, 28(9):57–62, 1995.
- [MH06] Branislav Micusík and Allan Hanbury. Automatic image segmentation by positioning a seed. In *ECCV (2)*, pages 468–480, 2006.
- [Mik02] K. Mikolajczyk. *Interest point detection invariant to affine transformations*. PhD thesis, Institut National Polytechnique de Grenoble, 2002.
- [MJ92] Jianchang Mao and Anil K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recogn.*, 25(2):173–188, 1992.
- [MM92] Farzin Mokhtarian and Alan K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(8):789–805, 1992.
- [MM95] Stricker M. and Orengo M. Similarity of color images. In *Proc. SPIE Storage and Retrieval for Image and Video Databases*, volume 1908, 1995.
- [MM97] W.Y. Ma and B.S. Manjunath. Edge flow: A framework of boundary detection and image segmentation. In *CVPR97*, pages 744–749, 1997.
- [MM99] Wei Y. Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.
- [Moj02] Aleksandra Mojsilovic. A method for color naming and description of color composition in images. In *in Proc. IEEE Int. Conf. Image Processing*, pages 789–792, 2002.
- [MPP01] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:349–361, 2001.
- [MR97] R. Manmatha and S. Ravela. A syntactic characterization of appearance and its application to image retrieval. In *Proc. of the SPIE conf. on Human Vision and Electronic Imaging II*, February 1997.

- [MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.
- [MSS02] B. S. Manjunath, Philippe Salembier, and Thomas Sikora, editors. *Introduction to MPEG-7: Multimedia Content Description Language*. Wiley, April 2002.
- [MTS<sup>+</sup>05] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.
- [NHS81] Jürg Nievergelt, Hans Hinterberger, and Kenneth C. Sevcik. The grid file: An adaptable, symmetric multi-key file structure. In *ECI*, pages 236–251, 1981.
- [NJ04] Anoop M. Namboodiri and Anil K. Jain. Retrieval of on-line hand-drawn sketches. *Pattern Recognition, International Conference on*, 2:642–645, 2004.
- [OKS80] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *CGIP*, 13(3):222–241, July 1980.
- [Ols92] Clark F. Olson. Fast alignment by eliminating unlikely matches. Technical report, EECS Department, University of California, Berkeley, October 1992.
- [OPR78] R. Ohlander, K.E. Price, and R. Reddy. Picture segmentation by a recursive region splitting method. *CGIP*, 8:313–333, 1978.
- [PB99] Maria Petrou and Panagiota Bosdogianni. *Image Processing: The Fundamentals*. John Wiley & Sons, Inc., New York, NY, USA, 1999.
- [PD99] N. Paragios and R. Deriche. Coupled geodesic active regions for image segmentation. *INRIA*, (3), October 1999.
- [Pen84] Alex P. Pentland. Fractal-based description of natural scenes. Technical Report 280, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Feb 1984.
- [PGT93] Tucci M. Petraglia G., Sebillio M. and TortoraG. Towards normalized iconic indexing. In *In Proc. IEEE symp. on visual languages*, pages 392–394, 1993. , Bergen.
- [PI97] M. Peura and J. Iivarinen. Efficiency of simple shape descriptors. In *In Aspects of Visual Form*, pages 443–451. World Scientific, 1997.
- [PL90] T. Pavlidis and Y.T. Liow. Integrating region growing and edge detection. *PAMI*, 12(3):225–233, March 1990.

- [PP93] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *PR*, 26(9):1277–1294, September 1993.
- [PPS96] Alex Pentland, Rosalind W. Picard, and Stan Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.
- [PSZ99] M. Pelillo, K. Siddiqi, and S. Zucker. Continuous-based heuristics for graph and tree isomorphisms, 1999.
- [PZ96] G. Pass and R. Zabih. Histogram refinement for content based image retrieval. *IEEE Workshop on Applications of Computer Vision*, pages 96–102, 1996.
- [RBK96] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:203, 1996.
- [RCB04] Julien Ricard, David Coeurjolly, and Atilla Baskurt. Generalizations of angular radial transform for 2d and 3d shape retrieval. Technical report, LIRIS UMR 5205 CNRS/INSA de Lyon, June 2004.
- [RdSTdFC03] A. X. Falcão R. da S. Torres and L. da F. Costa. A graph-based approach for multiscale shape analysis. Technical Report IC-0303, Institute of Computing, University of Campinas, January 2003.
- [RFS88] N. Roussopoulos, C. Faloutsos, and T. Sellis. An efficient pictorial database system for psql. *IEEE Trans. Softw. Eng.*, 14(5):639–650, 1988.
- [RFS<sup>+</sup>98] Bernice E. Rogowitz, Thomas Frese, Johnr. Smith, Charles A. Bouman, and Edward Kalin. Perceptual image similarity experiments. In *In SPIE Conference on Human Vision and Electronic Imaging*, pages 576–590, 1998.
- [RH99] Yong Rui and Thomas S. Huang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62, 1999.
- [RHM97] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Content-based image retrieval with relevance feedback in mars. In *In Proc. IEEE Int. Conf. on Image Proc.*, pages 815–818, 1997.
- [Rob81] John T. Robinson. The k-d-b-tree: a search structure for large multidimensional dynamic indexes. In *SIGMOD '81: Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, pages 10–18, New York, NY, USA, 1981. ACM.
- [RP68] A. Rosenfeld and J. Pfaltz. Distance functions in digital pictures. *Pattern Recognition*, 1:33–61, 1968.

- [RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, November 2000.
- [RW97] Isidore Rigoutsos and Haim Wolfson. Geometric hashing. *IEEE Computational Science Engineering*, 4:1070–9924, 1997.
- [SB91] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
- [SBC05] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Contour-based learning for object detection. *Computer Vision, IEEE International Conference on*, 1:503–510, 2005.
- [SD97] Markus Stricker and Er Dimai. Spectral covariance and fuzzy regions for image indexing. *Machine Vision and Applications*, 10:66–73, 1997.
- [SF] Lee S.Y. and Hsu F.J. 2d c-string: a new spatial knowledge representation for image database systems. In *Pattern Recognition*, volume 23.
- [SfC95] John R. Smith and Shih fu Chang. Single color extraction and image query. In *In Proc. IEEE Int. Conf. on Image Proc*, pages 528–531, 1995.
- [SfC96] John R. Smith and Shih fu Chang. Querying by color regions using the visualseek content-based visual query system. In *Intelligent Multimedia Information Retrieval*, pages 23–41. AAAI Press, 1996.
- [SI07] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)*, June 2007.
- [Sin99] D. Sinclair. Voronoi seeded colour image segmentation. Technical Report 3, AT-T Laboratories Cambridge, 1999.
- [SK01] Thomas B. Sebastian and Benjamin B. Kimia. Curves vs skeletons in object recognition. In *In IEEE International Conference of Image Processing*, pages 247–263, 2001.
- [SKK04] 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.
- [SL01] S. Sclaroff and L. Liu. Deformable shape detection and description via model-based region grouping. *PAMI*, 23(5):475–489, May 2001.
- [SLH99] Nicu Sebe, Michael S. Lew, and D. P. Huijsmans. Multi-scale sub-image search. In *In Proceedings of ACM International Conference on Multimedia*, pages 79–82, 1999.

- [SLW<sup>+</sup>97] P. Scheunders, S. Livens, G. Van De Wouwer, P. Vautrot, and D. Van Dyck. Wavelet-based texture analysis. *Int. Journal of Computer Science and Information Management, Special issue on Image Processing (IJCSIM)*, 1, 1997.
- [SM92] F. Stein and G. Medioni. Structural indexing: Efficient 2d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1198–1204, 1992.
- [SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [SM05] Elodie Garrivier Bruno Bossis Sébastien Macé, Éric Anquetil. A pen-based musical score editor. In *In proceedings ICMC, Barcelona, Spain*, pages 415–418, 2005.
- [SM07] Payam Sabzmezdani and Greg Mori. Detecting pedestrians by learning shapelet features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007.
- [SMW<sup>+</sup>00] Arnold W. M. Smeulders, Senior Member, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1349–1380, 2000.
- [SNSI98] Kayo Suzuki, Mitsuru Nagao, Yoshifumi Shimodaira, and Hiroaki Ikeda. Retrieval of butterfly from its sketched image on internet. *IPSI SIG Notes. MBL*, 98(83):31–37, 1998.
- [SS96] Hanan Samet and Aya Soffer. Marco: Map retrieval by content. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):783–798, 1996.
- [SS98] Aya Soffer and Hanan Samet. Integrating symbolic images into a multimedia database system using classification and abstraction approaches. *The VLDB Journal*, 7, 1998.
- [SSM99] Eugenio Di Sciascio, Eugenio Di Sciascio, and Marina Mongiello. Query by sketch and relevance feedback for content-based image retrieval over the web. *Journal of Visual Languages and Computing*, 10:565–584, 1999.
- [SSS00] Maytham Safar, Cyrus Shahabi, and Xiaoming Sun. Image retrieval by shape: A comparative study. In *IEEE International Conference on Multimedia and Expo (I)*, pages 141–144, 2000.
- [SST06] Muhammad Saleem, Adil Masood Siddiqui, and Imran Touqir. Efficient feature correspondence for image registration. In *SSIP'06: Proceedings of the 6th WSEAS International Conference on Signal*,



- Speech and Image Processing*, pages 101–104, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).
- [STL97] Stan Sclaroff, Leonid Taycher, and Marco LaCascia. Imagerover: A content-based image browser for the world wide web. Technical report, Boston, MA, USA, 1997.
- [SZ03] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. *Computer Vision, IEEE International Conference on*, 2:1470, 2003.
- [SZM99] John A. Shepherd, Xiaoming Zhu, and Nimrod Megiddo. A fast indexing method for multidimensional nearest neighbor search. In *SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, volume 3656, San Jose, California, USA, 1999.
- [SZR00] Rohini K. Srihari, Zhongfei Zhang, and Aibing Rao. Intelligent indexing and semantic retrieval of multimodal documents. *Inf. Retr.*, 2(2-3):245–275, 2000.
- [Tam82] Markku Tamminen. The extendible cell method for closest point problems. *BIT*, 22(1):27–41, 1982.
- [TC88] C.H. Teh and R.T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.
- [Tea80] M. R. Teague. Shape-based retrieval: A case study with trademark image databases. *Journal of the Optical Society of America*, 70:920–930, August 1980.
- [TG99] Y. Tao and W. Grosky. Delaunay triangulation for image object indexing: A novel method for shape representation. In *IST SPIE's Symposium on Storage and Retrieval for Image and Video Databases VII*, San Jose, California, January 1999.
- [TG04] Tinne Tuytelaars and Luc J. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.
- [TM07] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.
- [TMY78] Hideyuki Tamura, Shunji Mori, , and Takashi Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–473, 1978.
- [Ull72] J. D. Ullman. A note on the efficiency of hashing functions. *J. ACM*, 19(3):569–575, 1972.

- [UPH07] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. 29(6):929–944, June 2007.
- [vdWSV07] Joost van de Weijer, Cordelia Schmid, and Jakob Verbeek. Learning color names from real-world images. In *IEEE Conference on Computer Vision & Pattern Recognition*, jun 2007.
- [VH99] Remco C. Veltkamp and Michiel Hagedoorn. State-of-the-art in shape matching. Technical report, Principles of Visual Information Retrieval, 1999.
- [VJ04] Paul Viola and Michael Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.
- [VP88] H. Voorhees and T. Poggio. Computing texture boundaries from images. *Nature*, (333):364–367, 1988.
- [VR08] Vibha S. Vyas and Priti P. Rege. Geometric transform invariant texture analysis based on modified zernike moments. *Fundam. Inform.*, 88(1-2):177–192, 2008.
- [VT00] R.C. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical report, Department of Computer Science, Utrecht University, October 2000.
- [VvdWB08] Eduard Vazquez, Joost van de Weijer, and Ramon Baldrich. Image segmentation in the presence of shadows and highlights. In *ECCV (4)*, Lecture Notes in Computer Science, pages 1–14. Springer, 2008.
- [WA94] Skarbek W. and Koschan A. Colour image segmentation - a survey -. Technical report, Institute for Technical Informatics, Technical University of Berlin, October 1994.
- [Wer55] M. Wertheimer. *Laws of Organization in Perceptual Forms*, pages 71–88. Routledge Kegan Paul Ltd., 1955.
- [Wit87] A. P. Witkin. Scale-space filtering. In M. A. Fischler and O. Firschein, editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Kaufmann, 1987.
- [WJ96a] David A. White and Ramesh Jain. Algorithms and strategies for similarity retrieval. Technical Report VCL-96-101, Visual Computing Laboratory, University of California, San Diego, 9500 Gilman Drive, Mail Code 0407, La Jolla, CA 92093-0407, July 1996.
- [WJ96b] David A. White and Ramesh Jain. Similarity indexing with the ss-tree. In Stanley Y. W. Su, editor, *Proceedings of the Twelfth International Conference on Data Engineering, February 26 - March 1, 1996, New Orleans, Louisiana*, pages 516–523. IEEE Computer Society, 1996.

- [WN07] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *Int. J. Comput. Vision*, 75(2):247–266, 2007.
- [Woj09] Wojciech Wojcikiewicz. *Optimizing Visual Vocabularies with Dimension Reduction and Feature Selection Methods*. PhD thesis, Humboldt University of Berlin, 2009.
- [WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, pages 194–205, 1998.
- [WWFW97] J. Wang, G. Wiederhold, O. Firschein, and S. Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. In *International Conference on the Advances in Digital Libraries*, 1997.
- [WYA07] Yasuyuki Watai, Toshihiko Yamasaki, and Kiyoharu Aizawa. View-based web page retrieval using interactive sketch query. In *ICIP (6)*, pages 357–360. IEEE, 2007.
- [YCYB02] Choi Y., Won C.S., Ro Y.M, and Manjunath B.S. *Texture Descriptors*. 2002.
- [Yin03] Jesse S Yin. *Indexing and Retrieving High Dimensional Visual Features*. Springer Verlag, 2003.
- [ZB06] Djemel Ziou and Sabri Boutemedjet. An information filtering approach for the page zero problem. In *MRCIS*, pages 619–626, 2006.
- [ZF03] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.
- [ZH03] Xiang Sean Zhou and Thomas S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Syst.*, 8(6):536–544, 2003.
- [Zha96] Y. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, August 1996.
- [ZL02] D. Zhang and G. Lu. Generic fourier descriptor for shape-based image retrieval. *Signal Processing: Image Communication*, 17:825–848, 2002.
- [ZL04] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.
- [ZRZ02] Lei Zhu, Aibing Rao, and Aidong Zhang. Theory of keyblock-based image retrieval. *ACM Trans. Inf. Syst.*, 20:224–257, 2002.
- [ZY95] Song Chun Zhu and A. L. Yuille. Forms: A flexible object recognition and modeling system. *International Journal of Computer Vision*, 20:187–212, 1995.