

SEMANTICS-SENSITIVE INTEGRATED MATCHING  
FOR PICTURE LIBRARIES  
AND BIOMEDICAL IMAGE DATABASES

A DISSERTATION  
SUBMITTED TO THE DEPARTMENT OF BIOMEDICAL INFORMATICS  
AND THE COMMITTEE ON GRADUATE STUDIES  
OF STANFORD UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

By  
James Ze Wang  
August 2000

© Copyright 2000 by James Ze Wang  
All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

---

Professor Gio Wiederhold  
Computer Science and Biomedical Informatics  
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

---

Professor Hector Garcia-Molina  
Department of Computer Science

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

---

Professor Stephen T. C. Wong  
University of California

Approved for the University Committee on Graduate Studies:

---

*To my mother Baowen Guo, my father Yuan Wang,  
and my wife Jia Li*

# Abstract

The need for efficient content-based image retrieval has increased tremendously in many application areas such as biomedicine, military, commerce, education, and Web image classification and searching. In the biomedical domain, content-based image retrieval can be used in patient digital libraries, clinical diagnosis, searching of 2-D electrophoresis gels, and pathology slides. In this thesis, we present a wavelet-based approach for feature extraction, combined with integrated region matching. An image in the database, or a portion of an image, is represented by a set of regions, roughly corresponding to objects, which are characterized by color, texture, shape, and location. A measure for the overall similarity between images is developed as a region-matching scheme that integrates properties of all the regions in the images. The advantage of using such a “soft matching” is that it makes the metric robust to poor segmentation, an important property that previous work has not solved. An experimental image retrieval system, SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries), has been built to validate these methods on various image databases, including a database of about 200,000 general-purpose images and a database of more than 70,000 pathology image fragments. We have shown that our methods perform much better and much faster than existing methods. The system is exceptionally robust to image alterations such as intensity variation, sharpness variation, intentional distortions, cropping, shifting, and rotation. These features are important to biomedical image databases because visual features in the query image are not exactly the same as the visual features in the images in the database. The work has also been applied to the classification of on-line images and web sites.

# Acknowledgments

I would never have made it to Stanford in the first place if I hadn't met my undergraduate thesis advisor, Dennis A. Hejhal, who is not only a talented mathematician, but also a brilliant mentor. I would like to thank him for introducing me to the excitement of conducting scientific research using high-performance computers, and for being everlastingly supportive during the past nine years.

This work would not have been possible without the guidance and advice of Professor Gio Wiederhold. When I started as a Ph.D. student in Stanford Biomedical Informatics almost three years ago, I had no idea what I was getting into. I feel fortunate to have chosen Gio as my advisor. Gio and his family have always treated me with great warmth. He has led me to new areas of research, pointed me to interesting research problems, and offered me substantial encouragement. Gio has cultivated a creative atmosphere and provided me with unconditional support. He has taught me what an advisor should be.

I would like to thank Professors Russ B. Altman, Hector Garcia-Molina, Mu-Tao Wang, and Stephen T.C. Wong, who have served on my defense committee, or both my defense and reading committees, for spending much time on this dissertation and providing numerous constructive comments.

I would like to thank Martin A. Fischler and Oscar Firschein for inspiring me with the fascinating field of image understanding, and encouraging me throughout the three years in the Ph.D. program. Marty's rigorous attitude to research will influence my entire academic career. I am indebted to Oscar, who carefully went through many of my papers and this dissertation, and greatly improved the quality of the explanations throughout.

My dissertation is on applications of the wavelet theory in image databases. I would like to acknowledge Professor Ingrid Daubechies at the Mathematics Department at Princeton University for her fundamental research work in the field of time-frequency analysis.

My most rewarding experience at Stanford has been my interaction with people, both on-campus and in local industry, who have a wide variety of backgrounds and interests. I had the privilege of working with, and learning from, many talented individuals. Among these are Yuval Shahar, Edward H. Shortliffe, Russ B. Altman, Mark Musen, Parvati Dev, and Larry Fagan, who have been great teachers and have always guided me to the right direction in the Ph.D. program. Dragutin Petkovic, Carlo Tomasi, Wayne Niblack, and Sha Xin Wei provided inspiration in the field of image retrieval. Quang-Tuan Luong introduced me to the excitement of professional photography, as well as image understanding. Discussions with Scott Atlas, Michel Bilello, Desmond Chan, Edward Chang, Junghoo Cho, Terry Desser, Eldar Giladi, Robert M. Gray, Yoshi Hara, Maj Hedehus, Kyoji Hirata, Xiaoming Huo, Yvan Leclerc, Chen Li, Yanyuan Ma, John Nguyen, Richard Olshen, Donald Regula, Daniel Rubin, Smadar Shiffman, Maria Tovar, Michael Walker, and Tong Zhang have been very helpful in different stages of my research.

I would also like to thank my friends in the Database Group, the Biomedical Informatics Group, the Mathematics Department, the Perception Research Group at SRI International, and the QBIC Group at the IBM Almaden Research Center for their generous help. Especially, Andy Kacsmar, Marianne Siroker, Ricky Connell, Darlene Vian, and Barbara Morgan have offered me much support.

My wife Jia Li is the most essential contributor to my success and my well-being. I would like to thank her for the never-ending love and incredible support she has given me. Even though she herself was conducting her Ph.D. research at Stanford at the same time, she has always devoted a lot of time to the family. She was always with me when I stayed up at midnight to finish my work. Her talents and professional expertise in statistics, information theory, and image processing have enlightened me numerous times throughout my research. We have coauthored several publications and experimental software systems.

And last, I would like to thank my father Yuan Wang and my mother Baowen Guo who taught me to be kind to others, taught me to be honest, taught me to be ambitious, and taught me to be diligent and persistent. This thesis is dedicated to them.

My graduate school experience was funded primarily by a research grant from the National Science Foundation's Digital Libraries initiative and a research fund from the Stanford University Libraries. I have also received support from IBM Almaden Research Center, NEC Research Lab, SRI International, Stanford Computer Science Department, Stanford Mathematics Department, and Stanford Medical Informatics. I am truly grateful for the support. After graduation, my research work will be continued at the new School of Information Sciences and Technology at Penn State University.



# Contents

<b>Abstract</b>	<b>v</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Text-based image retrieval . . . . .	2
1.2 Content-based image retrieval . . . . .	3
1.3 Applications of CBIR . . . . .	3
1.3.1 Biomedical applications . . . . .	4
1.3.2 Web-related applications . . . . .	6
1.3.3 Other applications . . . . .	7
1.4 Contributions . . . . .	8
1.4.1 Semantics-sensitive image retrieval . . . . .	8
1.4.2 Image classification . . . . .	10
1.4.3 Integrated Region Matching (IRM) similarity measure . . . . .	11
1.4.4 Applications of the methods . . . . .	13
1.5 Structure of dissertation . . . . .	13
1.6 Summary . . . . .	16
<b>2 Related Work</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Content-based image retrieval . . . . .	18
2.2.1 Major challenges . . . . .	18
2.2.2 Previous work . . . . .	24

2.2.3	CBIR for biomedical image databases . . . . .	34
2.3	Image semantic classification . . . . .	36
2.3.1	Semantic classification for photographs . . . . .	36
2.3.2	Medical image classification . . . . .	38
2.4	Summary . . . . .	39
<b>3</b>	<b>Wavelets</b>	<b>40</b>
3.1	Introduction . . . . .	40
3.2	Fourier transform . . . . .	41
3.3	Wavelet transform . . . . .	42
3.3.1	Haar wavelet transform . . . . .	43
3.3.2	Daubechies' wavelet transform . . . . .	44
3.4	Applications of wavelets . . . . .	48
3.5	Summary . . . . .	50
<b>4</b>	<b>Statistical Clustering and Classification</b>	<b>51</b>
4.1	Introduction . . . . .	51
4.2	Artificial intelligence and machine learning . . . . .	52
4.3	Statistical clustering . . . . .	53
4.3.1	The k-means algorithm . . . . .	54
4.3.2	The TSVQ algorithm . . . . .	56
4.4	Statistical classification . . . . .	58
4.4.1	The CART algorithm . . . . .	59
4.5	Summary . . . . .	65
<b>5</b>	<b>Wavelet-Based Image Indexing and Searching</b>	<b>66</b>
5.1	Introduction . . . . .	66
5.2	Preprocessing . . . . .	67
5.2.1	Scale normalization . . . . .	67
5.2.2	Color space normalization . . . . .	68
5.3	Multiresolution indexing . . . . .	69
5.3.1	Color layout . . . . .	69

5.3.2	Indexing with the Haar wavelet . . . . .	70
5.3.3	Overview of WBIIS . . . . .	71
5.4	The indexing algorithm . . . . .	72
5.5	The matching algorithm . . . . .	75
5.5.1	Fully-specified query matching . . . . .	75
5.5.2	Partial query . . . . .	79
5.6	Performance . . . . .	81
5.7	Limitations . . . . .	89
5.8	Summary . . . . .	90
<b>6</b>	<b>Semantics-sensitive Integrated Matching</b>	<b>91</b>
6.1	Introduction . . . . .	91
6.2	Overview . . . . .	92
6.3	Image segmentation . . . . .	95
6.4	Image classification . . . . .	99
6.4.1	Textured vs. non-textured images . . . . .	99
6.4.2	Graph vs. photograph images . . . . .	100
6.5	The similarity metric . . . . .	101
6.5.1	Integrated region matching . . . . .	101
6.5.2	Distance between regions . . . . .	107
6.6	System for biomedical image databases . . . . .	110
6.6.1	Feature extraction . . . . .	111
6.6.2	Wavelet-based progressive transmission . . . . .	112
6.7	Clustering for large databases . . . . .	112
6.8	Summary . . . . .	114
<b>7</b>	<b>Evaluation</b>	<b>115</b>
7.1	Introduction . . . . .	115
7.2	Overview . . . . .	115
7.3	Data sets . . . . .	116
7.3.1	The COREL data set . . . . .	116
7.3.2	Pathology data set . . . . .	118

7.4	Query interfaces . . . . .	119
7.4.1	Web access interface . . . . .	119
7.4.2	JAVA drawing interface . . . . .	120
7.4.3	External query interface . . . . .	121
7.4.4	Progressive browsing . . . . .	122
7.5	Characteristics of IRM . . . . .	122
7.6	Accuracy . . . . .	124
7.6.1	Picture libraries . . . . .	124
7.6.2	Systematic evaluation . . . . .	133
7.6.3	Biomedical image databases . . . . .	139
7.7	Robustness . . . . .	141
7.7.1	Intensity variation . . . . .	147
7.7.2	Sharpness variation . . . . .	148
7.7.3	Color distortions . . . . .	148
7.7.4	Other intentional distortions . . . . .	152
7.7.5	Cropping and scaling . . . . .	152
7.7.6	Shifting . . . . .	152
7.7.7	Rotation . . . . .	155
7.8	Speed . . . . .	155
7.9	Summary . . . . .	156
<b>8</b>	<b>Conclusions and Future Work</b>	<b>157</b>
8.1	Summary . . . . .	157
8.2	Limitations . . . . .	158
8.3	Areas of future work . . . . .	159
<b>A</b>	<b>Image Classification By Image Matching</b>	<b>162</b>
A.1	Introduction . . . . .	162
A.2	Industrial solutions . . . . .	163
A.3	Related work in academia . . . . .	164
A.4	System for screening objectionable images . . . . .	165
A.4.1	Moments . . . . .	166

A.4.2	The algorithm . . . . .	167
A.4.3	Evaluation . . . . .	171
A.5	Classifying objectionable websites . . . . .	172
A.5.1	The algorithm . . . . .	173
A.5.2	Statistical classification process for websites . . . . .	175
A.5.3	Limitations . . . . .	181
A.5.4	Evaluation . . . . .	182
A.6	Summary . . . . .	182
<b>Bibliography</b>		<b>183</b>
<b>Index</b>		<b>198</b>

# List of Tables

6.1	Performance of the system for an image database of $N$ images. $C$ is the maximum number of images in each leaf node. . . . .	113
7.1	The contents of the first 10,000 images in the COREL image database according to the titles of the CDs. . . . .	117
7.2	COREL categories of images tested for comparing with WBIIS. . . . .	134
7.3	COREL categories of images tested for comparing with color histogram. . . . .	136
7.4	The performance of SIMPLicity (with an average of 4.3 regions per image) on categorizing picture libraries. The average performance for each image category evaluated by precision $p$ , the mean rank of matched images $r$ , and the standard deviation of the ranks of matched images $\sigma$ . . . . .	136
7.5	The performance of the EMD-based color histogram approach (with an average of 42.6 filled color bins) on categorizing picture libraries. The average performance for each image category evaluated by precision $p$ , the mean rank of matched images $r$ , and the standard deviation of the ranks of matched images $\sigma$ . . . . .	137
7.6	The performance of the EMD-based color histogram approach (with an average of 13.1 filled color bins) on categorizing picture libraries. The average performance for each image category evaluated by precision $p$ , the mean rank of matched images $r$ , and the standard deviation of the ranks of matched images $\sigma$ . . . . .	139

# List of Figures

1.1	Future integrated medical image/record retrieval system. . . . .	4
1.2	Multiple sclerosis plaques under different MR imaging methods. Two axial slices are shown. The images were enhanced using non-linear histogram mapping functions. . . . .	5
1.3	Sample textured images. (a) surface texture (b) fabric texture (c) artificial texture (d) pattern of similarly-shaped objects . . . . .	10
1.4	Region-to-region matching results are incorporated in the Integrated Region Matching (IRM) metric. A 3-D feature space is shown to illustrate the concept. . . . .	12
2.1	Queries to be handled by CBIR systems using primitive feature indexing. (a) histogram query (b) layout query (c) shape query (d) hand-drawn sketch query (e) query by example . . . . .	20
2.2	Query interface of the Blobworld system developed at the University of California, Berkeley. . . . .	21
2.3	It is difficult to define relevance for image semantics. Left: images labeled as “dog” by photographers. Right: images labeled as “Kyoto, Japan” by photographers. . . . .	23
2.4	The architecture of a typical CBIR system. . . . .	25
2.5	The indexing process of a typical CBIR system. . . . .	25
2.6	The signature of an image is a set of features. . . . .	26
2.7	The retrieval process of a typical CBIR system. . . . .	27

2.8	Two sample color histogram query results, one good, one poor. The image in the upper-left corner of each block is the query image. DB size: 10,000 images. . . . .	31
3.1	The Fourier transforms create visible boundary artifacts. . . . .	41
3.2	Plots of some analyzing wavelets. First row: father wavelets, $\phi(x)$ . Second row: mother wavelets, $\psi(x)$ . . . . .	45
3.3	Comparison of Haar's wavelet and Daubechies wavelets on a 1-D signal. (a) original signal ( $x\epsilon^{-x^2}$ ) of length 1024 (b) coefficients in high-pass bands after a 4-layer Haar transform (c) coefficients in high-pass bands after a 4-layer Daubechies-3 transform (d) coefficients in high-pass bands after a 4-layer Daubechies-8 transform . . . . .	46
3.4	Comparison of Haar's wavelet and Daubechies-8 wavelet. . . . .	47
3.5	A 3-level wavelet transform of an MRI image slice using Daubechies' wavelet. . . . .	48
3.6	Multi-scale structure in the wavelet transform of an image. Dots indicate non-zero wavelet coefficients after thresholding. Daubechies-8 wavelet is used for this transform. . . . .	49
4.1	The k-means algorithm partitions the feature space using hyper-planes.	55
4.2	An example of tree structured partition of feature space. 'x' indicates an individual feature. '.' indicates the centroid of a cluster cell. . . .	57
4.3	Generating a classification tree using the CART algorithm. . . . .	59
4.4	CART partitions the 2-D feature space into cells using straight lines parallel to the coordinate axes. . . . .	60
4.5	Semantic analysis of outdoor scenes using the classification and regression trees (CART) algorithm. No post-processing is performed. Color scheme: Deep blue (darkest) for sky, yellow (very light gray) for stone, light blue (lightest gray) for river/lake, light green (light gray) for grass, deep green (dark gray) for tree/forest. (Wang and Fischler [128]) . . .	62



4.6	Semantic analysis of outdoor scenes using the classification and regression trees (CART) algorithm. No post-processing is performed. Color scheme: Deep blue (darkest) for sky, light blue for river/lake, light green (light gray) for grass, deep green (dark gray) for tree/forest, white for non-classified regions. (Wang and Fischler [128]) . . . . .	63
4.7	Classification of MRI images using CART. No pre- or post-processing.	64
5.1	Two images with the upper-left corner submatrices of their fast wavelet transforms in $(C_1, C_2, C_3)$ color space. We use the standard deviations of wavelet coefficients to distinguish images with very different object composition. The standard deviations we stored for the first image are $\sigma_{C_1} = 215.93$ , $\sigma_{C_2} = 25.44$ , and $\sigma_{C_3} = 6.65$ while means of the coefficients in the lowest frequency band are $\mu_{C_1} = 1520.74$ , $\mu_{C_2} = 2124.79$ , and $\mu_{C_3} = 2136.93$ . The standard deviations we stored for the second image are $\sigma_{C_1} = 16.18$ , $\sigma_{C_2} = 10.97$ , and $\sigma_{C_3} = 3.28$ while means of the coefficients in the lowest frequency band are $\mu_{C_1} = 1723.99$ , $\mu_{C_2} = 2301.24$ and $\mu_{C_3} = 2104.33$ . . . . .	74
5.2	Histogram of the standard deviations of the wavelet coefficients in the lowest frequency band. Results were obtained from a database of more than 10,000 general-purpose images. . . . .	76
5.3	Types of partial sketch queries our WBIIS system handles. Black areas in a query image represent non-specified areas. . . . .	79
5.4	Comparisons with a commercial algorithm (IBM QBIC) on a galaxy-type image. Note that 12 out of 15 images retrieved by the commercial algorithm are unrelated to the galaxy query image. WBIIS retrieved only 6 unrelated images. The upper-left corner image in each block of images is the query. The image to the right of that image is the best matching image found. Matches decrease in measured closeness from left to right and from top to bottom. Results were obtained from a database of approximately 10,000 images. . . . .	82

5.5	A query example. 9 images unrelated to a water scene were retrieved by the University of Washington algorithm. WBIIS retrieved only one unrelated image. The upper-left corner image in each block of images is the query. Results were obtained from a database of approximately 10,000 images. . . . .	83
5.6	Another query example. . . . .	84
5.7	Comparison on a texture image. . . . .	85
5.8	Partial sketch queries in different resolutions. The upper-left corner image in each block of images is the query. Black areas in a query image represent non-specified areas. Database size: 10,000 images. . .	86
5.9	Query results on a hand-drawn query image (with blue, black, yellow, and green blocks). Black areas in a query image represent non-specified areas. Equivalent query were used. Database size: 10,000 images. . .	87
5.10	Two other query examples using WBIIS. The upper-left corner image in each block of images is the query. . . . .	88
6.1	The architecture of feature indexing process. The heavy lines show a sample indexing path of an image. . . . .	93
6.2	The architecture of query processing process. The heavy lines show a sample querying path of an image. . . . .	94
6.3	Decomposition of images into frequency bands by wavelet transforms.	97
6.4	Segmentation results by the k-means clustering algorithm: (a) Original texture images, (b) Regions of the texture images, (c) Original non-textured images, (d) Regions of the non-textured images. . . . .	98
6.5	The histograms of average $\chi^2$ 's over 100 textured images and 100 non-textured images. . . . .	100
6.6	Integrated Region Matching (IRM) is robust to poor image segmentation.	102
6.7	Integrated region matching (IRM). . . . .	104
6.8	Feature extraction in the SIMPLIcity system. (* The computation of shape features is omitted for textured images.) . . . . .	109

6.9	Automatic object segmentation of pathology images is an extremely difficult task. When different thresholds are provided, an edge detector either gives too many edges or too few edges for the object grouping. We avoid the precise object segmentation process by using our IRM “soft matching” metric. . . . .	111
7.1	A random set of 24 image fragments from the pathology image database.	118
7.2	The Web access interface. A different random set of 18 images from the database is shown initially. . . . .	119
7.3	The JAVA drawing query interface allows users to draw sketch queries.	120
7.4	The external query interface. The best 17 matches are presented for a query image selected by the user from the Stanford top-level Web page. The user enters the URL of the query image (shown in the upper-left corner, <a href="http://www.stanford.edu/home/pics/h-quad.jpg">http://www.stanford.edu/home/pics/h-quad.jpg</a> ) to form a query. . . . .	121
7.5	Multiresolution progressive browsing of pathology slides of extremely high resolution. HTML-based interface shown. The magnification of the pathology images are shown on the query interface. . . . .	123
7.6	The empirical PDF and CDF of the IRM distance. . . . .	124
7.7	Comparison of SIMPLIcity and WBIIS. The query image is a landscape image on the upper-left corner of each block of images. SIMPLIcity retrieved 8 related images within the best 11 matches. WBIIS retrieved 7 related images. . . . .	125
7.8	Comparison of SIMPLIcity and WBIIS. The query image is a photo of food. SIMPLIcity retrieved 10 related images within the best 11 matches. WBIIS did not retrieve any related images. . . . .	126
7.9	Comparison of SIMPLIcity and WBIIS. The query image is a portrait image that probably depicts life in Africa. SIMPLIcity retrieved 10 related images within the best 11 matches. WBIIS did not retrieve any related images. . . . .	127

7.10	Comparison of SIMPLIcity and WBIIS. The query image is a portrait of a model. SIMPLIcity retrieved 7 related images within the best 11 matches. WBIIS retrieved only one related image. . . . .	128
7.11	Comparison of SIMPLIcity and WBIIS. The query image is a photo of flowers. SIMPLIcity retrieved 10 related images within the best 11 matches. WBIIS retrieved 4 related images. . . . .	129
7.12	SIMPLIcity gives better results than the same system without the classification component. The query image is a textured image. . . . .	131
7.13	SIMPLIcity does not mix clip art pictures with photographs. A graph-photograph classification method using image segmentation and statistical hypothesis testing is used. The query image is a clip art picture.	132
7.14	Comparison of SIMPLIcity and WBIIS: average precision and weighted precision of 9 image categories. . . . .	135
7.15	Comparing SIMPLIcity with color histogram methods on average precision $p$ , average rank of matched images $r$ , and the standard deviation of the ranks of matched images $\sigma$ . <i>The lower numbers indicate better results for the last two plots (i.e., the <math>r</math> plot and the <math>\sigma</math> plot).</i> Color Histogram 1 gives an average of 13.1 filled color bins per image, while Color Histogram 2 gives an average of 42.6 filled color bins per image. SIMPLIcity partitions an image into an average of only 4.3 regions. . .	138
7.16	Segmentation results obtained using an algorithm based on k-means. A region is defined as a collection of pixels. . . . .	140
7.17	A sample query result. The first image is the query. . . . .	141
7.18	The results of hand-drawn sketch queries. . . . .	142
7.19	The robustness of the system to intensity alterations. 6 images were randomly selected from the database. Each curve represents the robustness on one of the 6 images. . . . .	143
7.20	The robustness of the system to sharpness alterations. 6 images were randomly selected from the database. Each curve represents the robustness on one of the 6 images. . . . .	144

7.21	The robustness of the system to color alterations. 6 images were randomly selected from the database. Each curve represents the robustness on one of the 6 images. . . . .	145
7.22	The robustness of the system to other image alterations. 6 images were randomly selected from the database. Each curve represents the robustness on one of the 6 images. . . . .	146
7.23	The robustness of the system to image intensity changes. . . . .	147
7.24	The robustness of the system to sharpness variations. . . . .	149
7.25	The robustness of the system to intentional color distortions. . . . .	150
7.26	The robustness of the system to two other intentional distortions. . .	151
7.27	The robustness of the system to image cropping and scaling. . . . .	153
7.28	The robustness of the system to image shifting. . . . .	154
7.29	The robustness of the system to image rotation. . . . .	155
A.1	Basic structure of the algorithm in WIPE. . . . .	167
A.2	Typical benign images being marked mistakenly as objectionable images by WIPE. (a) areas with similar features (b) fine-art image (c) animals (without clothes) (d) partially undressed human (e) partially obscured human. . . . .	172
A.3	Basic structure of the algorithm in IBCOW. . . . .	174
A.4	Distributions assumed for the percentage ( $p$ ) of objectionable images on objectionable websites. . . . .	175
A.5	Dependence of correct classification rates on sensitivity and specificity of WIPE (for the Gaussian-like distribution of $p$ ). Left: $q_2 = 91\%$ , $q_1$ varies between 80% to 100%. Right: $q_1 = 96\%$ , $q_2$ varies between 80% to 100%. Solid line: correct classification rate for objectionable websites. Dash dot line: correct classification rate for benign websites. . . . .	180

# Chapter 1

## Introduction

*Make everything as simple as possible, but not simpler.*

— Albert Einstein (1879-1955)

The need for efficient content-based image retrieval has increased tremendously in many application areas such as biomedicine, crime prevention, military, commerce, culture, education, and entertainment. Content-based image retrieval is also crucial to Web image classification and searching.

With the steady growth of computer power, rapidly declining cost of storage, and ever-increasing access to the Internet, digital acquisition of information has become increasingly popular in recent years. Digital information is preferable to analog formats because of convenient sharing and distribution properties. This trend has motivated research in image databases, which were nearly ignored by traditional computer systems because of the large amount of data required to represent images and the difficulty of automatically analyzing images. Currently, storage is less of an issue since huge storage capacities are available at low cost. However, effective indexing<sup>1</sup> and searching of large-scale image databases remain as challenges for computer systems.

---

<sup>1</sup>Here, the term *indexing* means the combination of both feature extraction and feature space indexing.

The automatic derivation of semantically-meaningful information from the content of an image is the focus of interest for most research on image databases. The image “semantics”, i.e., the meanings of an image, has several levels. From the lowest to the highest, these levels can be roughly categorized as follows:

1. Semantic types (e.g., MRI, X-ray, landscape photograph, clip art)
2. Object composition (e.g., a lesion in the left brain, a bike and a car parked on a beach, a sunset scene)
3. Abstract semantics (e.g., people fighting, happy person, objectionable photograph)
4. Detailed semantics (e.g., a detailed description of a given picture)

*Image retrieval* is defined as the retrieval of semantically-relevant images from a database of images. In the following sections (Section 1.1 and Section 1.2), we discuss text-based image retrieval and content-based image retrieval.

## 1.1 Text-based image retrieval

In current commercial image databases, the prevalent retrieval techniques involve human-supplied text annotations to describe image semantics. These text annotations are then used as the basis for searching, using mature text search algorithms developed in the database [37] community. It is often easier to design and implement an image search engine based on keywords (e.g., classification codes) or full-text descriptions (e.g., surrounding text) than on the image content. The query processing of such search engines is typically very fast due to the available efficient database management technology. The text-based image retrieval approach is accepted for high-value pictures such as museum pictures.

Recently, researchers have proposed community-wide *social* entry of descriptive text to facilitate subsequent retrieval. This approach is feasible with the widely-available Internet. However, it is limited to image sets that are of wide interest and stable.

There are many problems in using text-based approach alone. For example, different people may supply different textual annotations for the same image. This makes it extremely difficult to answer user queries reliably. Furthermore, entering textual annotations manually is excessively expensive for large-scale image databases (e.g., space-based observations).

## 1.2 Content-based image retrieval

*Content-based image retrieval (CBIR)* is the set of techniques for retrieving relevant images from an image database on the basis of automatically-derived image features.

CBIR functions differently from text-based image retrieval. Features describing image content, such as color histogram, color layout, texture, shape and object composition, are computed for both images in the database and query images. These features are then used to select the images that are most similar to the query. High-level semantic features, such as the types of objects in the images and the purpose of the images are extremely difficult to extract. Deviation of semantically-meaningful features remains a great challenge.

CBIR is also important for video indexing and retrieval. In a typical video retrieval system, long video sequences are broken up into separate clips and key frames are extracted to represent the content of the clips. Searching of relevant clips is done by combining CBIR, speech recognition, and searching for specific movements of the objects in the shots [106, 15]. In this dissertation, we focus on content-based *image* retrieval.

## 1.3 Applications of CBIR

Content-based image retrieval (CBIR) has applications in various domains in many areas of our society.



### 1.3.1 Biomedical applications

CBIR is critical in developing patient care digital libraries. McKeown, Chang, Cimino, and Hripcsak [45] of Columbia University plan<sup>2</sup> to develop a personalized search and summarization system over multimedia information within a healthcare setting. Both patients and healthcare providers are targeted users of the system. Efficient CBIR is the most important core technology within such systems. With the help of such a mediator [132, 133], healthcare consumers and providers can quickly access a wide range of online resources: patients and their families can find information about their personal situation, and clinicians can find clinically relevant information for individual patients. A similar research effort is the Stanford SHINE project [53].

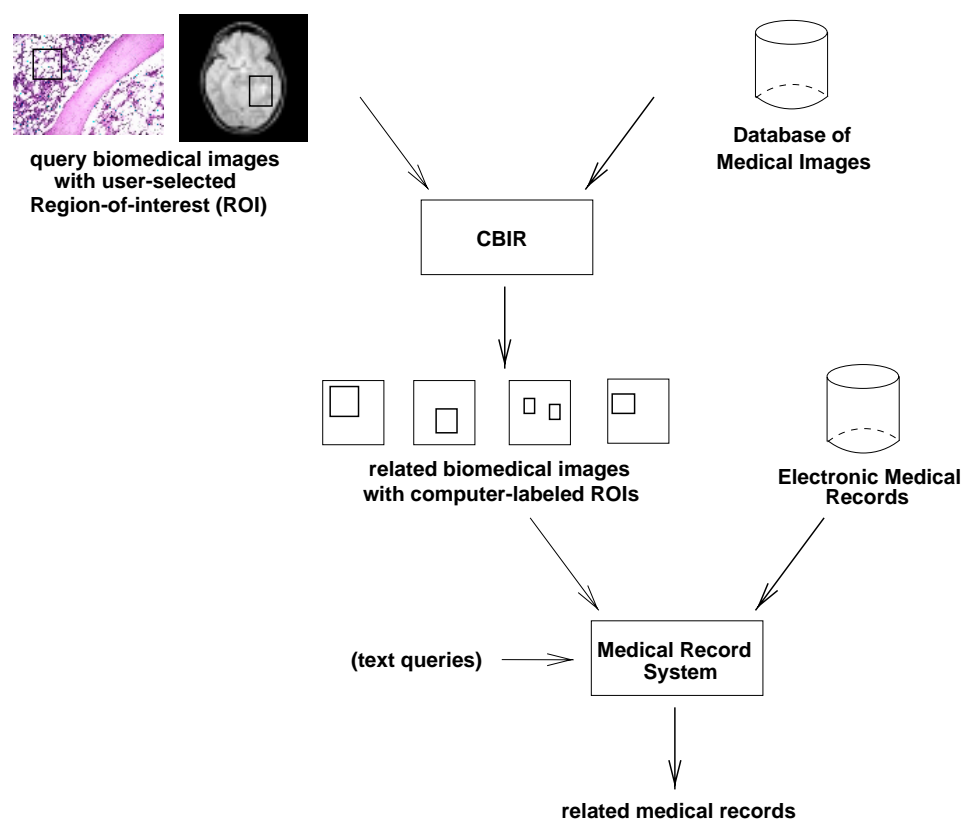


Figure 1.1: Future integrated medical image/record retrieval system.

<sup>2</sup>Recently funded by a joint National Science Foundation and National Institute of Health grant.

CBIR can be applied to clinical diagnosis and decision making. Currently, more and more hospitals and radiology departments are equipped with Picture Archive and Communications Systems (PACS) [134]. Besides the traditional X-ray and mammography, newer image modalities such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) can produce up to 512 slices per patient scan. Each year, a typical hospital can produce several terabytes of digital and digitized medical images. Efficient content-based image indexing and retrieval will allow physicians to identify similar past cases. By studying the diagnoses and treatments of past cases, physicians may be able to better understand new cases and make better treatment decisions. Furthermore, learning-based computer classification programs may be developed, based on the features of past cases. Figure 1.1 shows the architecture of a future integrated medical image/record retrieval system.

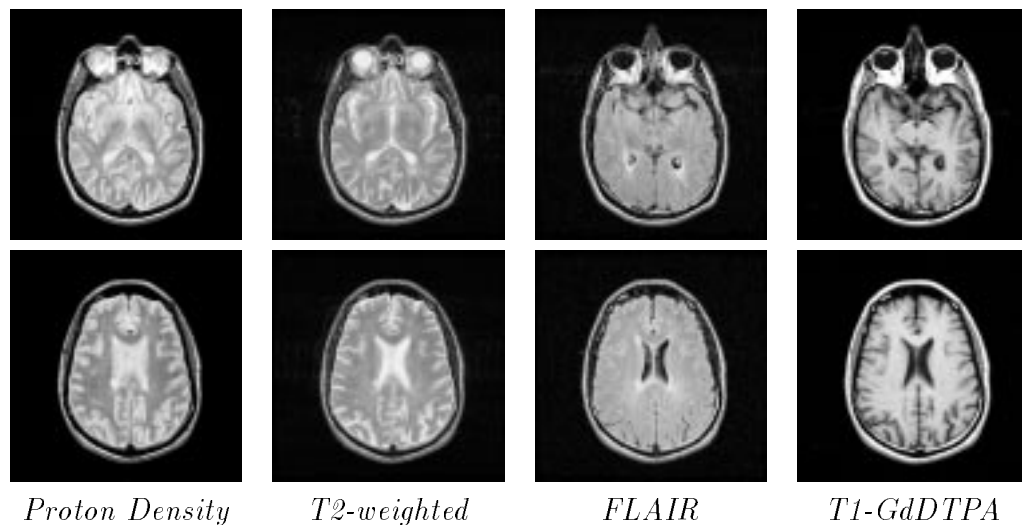


Figure 1.2: Multiple sclerosis plaques under different MR imaging methods. Two axial slices are shown. The images were enhanced using non-linear histogram mapping functions.

CBIR can also be applied to large-scale clinical trials. As brain MRI image databases used in clinical research and clinical trials (e.g., for tackling multiple sclerosis, Figure 1.2) become larger and larger in size, it is increasingly important to design an efficient and consistent algorithms that are able to detect and track the growth of lesions automatically.

CBIR can be utilized to classify and screen objectionable multimedia contents. Such systems are needed in hospitals, schools, libraries and businesses. As more and more hospitals are equipped with Internet-capable terminals, there is a possible concern that health care aids, technicians, and hospitalized children patients gain access to objectionable contents on the World-Wide-Web (Web) when they are not supposed to. For the healthy environment and the productivity of the hospitals, an efficient filtering system is desired.

Within bioinformatics [2], CBIR can be used for managing large-scale protein image databases obtained from 2-D electrophoresis gels. Presently, the screening process on very large gel databases is done manually or semi-automatically at pharmaceutical companies. With CBIR techniques, users may be able to find similar protein signatures based on an automatically-generated index of the visual characteristics.

CBIR is important in medical image database security. Before digital medical images in computer-based patient record systems can be distributed online, it is necessary for confidentiality reasons to eliminate patient identification information that appears in the images [124, 127]. The security managers of X-ray databases are interested in queries such as “find images with imprinted patient identification information” and “find images with the faces of the patients”. Proper content-based indexing is necessary to process such queries.

And lastly, we can exploit CBIR in biomedical education. Medical education is an area that has been revolutionized by the emergence of the Web and its related technologies. The Web is a rich medium that can increase access to educational materials and can allow new modes of interaction with these materials. We may utilize CBIR to organize images of slides prepared for use in medical school pathology courses [129].

### **1.3.2 Web-related applications**

The World-Wide Web (WWW), established in the early 1990s, allows people all over the world to access multimedia data from any place on earth. It has served as a catalyst to the massive creation of on-line images.

According to a report published by Inktomi Corporation and NEC Research in January 2000 [55], there are about 5 million unique Web sites ( $\pm 3\%$ ) on the Internet. Over one billion web pages ( $\pm 35\%$ ) can be downloaded from these Web sites. Approximately one billion images can be found on-line. Searching for information on the Web is a serious problem [65, 66]. Moreover, the current growth rate of the Web is exponential, at an amazing 50% annual rate.

Surrounding text has been shown to be useful in indexing images in the Web [106]. However, surrounding text alone is often not sufficient because text that appears in the same page may not describe the images in that page.

There are several general-purpose image search engines. In the commercial domain, IBM QBIC [30, 32, 88] is one of the earliest developed systems. Recently, additional systems have been developed at IBM T.J. Watson [107], VIRAGE [46] (Alta Vista), NEC AMORA [85], Bell Laboratory [86], Interpix (Yahoo), Excalibur, and Scour.net. In the academic domain, MIT Photobook [89, 91] is one of the earliest. Berkeley Blobworld [11], Columbia VisualSEEK and WebSEEK [106], CMU Informedia [110], UIUC MARS [83], UCSB NeTra [77], UCSD [59], Stanford (EMD [96], WBIIS [118]) are some of the recent systems. None of them has the capability to handle the vast amount of image data on the Web and allow users to search for semantically-related images.

### 1.3.3 Other applications

Besides its potential exciting biomedical and Web-related applications, CBIR is critical in many other areas. The following is only a partial list.

- Crime prevention (fingerprint, face recognition)
- Military (radar, aerial, satellite target recognition)
- Space observation (satellite observations of agriculture, deforestation, traffic, etc)
- Intellectual property (trademark, image copy detection [13, 14, 126])

- Architectural and engineering design (CAD database)
- Commercial (fashion catalogue, journalism)
- Cultural (museums, art galleries)
- Educational and training (lecture slides, graphs)
- Entertainment (photo, video, movie)
- Image classification (filtering of *adult-only* objectionable images and Web sites)
- Image security filtering (locate images with certain critical patterns [121])

## 1.4 Contributions

CBIR is a complex and challenging problem spanning diverse disciplines, including computer vision, color perception, image processing, image classification, statistical clustering, psychology, human-computer interaction (HCI), and specific application domain dependent criteria as found in radiology, pathology and biochemistry. Details of the major challenges and related work are given in Chapter 2. While we are not claiming to be able to solve all the problems related to CBIR, we have made some advances towards the final goal, close to human-level automatic image understanding and retrieval performance.

In this dissertation, we discuss issues related to the design and implementation of a semantics-sensitive content-based image retrieval system for picture libraries and biomedical image databases. An experimental system has been built to validate the methods. We summarize the main contributions as follows:

### 1.4.1 Semantics-sensitive image retrieval

The capability of existing CBIR systems is essentially limited by the way they function, i.e., they rely on only primitive features of the image. None of them can search for, say, a photo of a car near a tree, though some attempts have been made to specify

semantic queries as a combination of primitive queries. A zebra picture, for example, can be described as having areas of green (as grass) as the background of the image, and some zebra-like texture in the center. Specifying complex queries like this can be very time-consuming (Figure 2.2). Experiments with the Blobworld system have shown that it does not provide significantly better searching results. Moreover, the same low-level image features and image similarity measures are typically used for images of all semantic types. However, different image features are sensitive to different semantic types. For example, a color layout indexing method may be good for outdoor pictures while a region-based indexing approach is much better for indoor pictures. Similarly, global texture matching is suitable only for textured pictures.

We propose a *semantics-sensitive* approach to the problem of searching general-purpose image databases. Semantic classification methods are used to categorize images so that semantically-adaptive searching methods applicable to each category can be applied. At the same time, the system can narrow down the searching range to a subset of the original database to facilitate fast retrieval. For example, automatic classification methods can be used to categorize a general-purpose picture library into semantic classes including “graph”, “photograph”, “textured”, “non-textured”, “benign”, “objectionable”, “indoor”, “outdoor”, “city”, “landscape”, “with people”, and “without people”. A biomedical image database may be categorized into “X-ray”, “MRI”, “pathology”, “graphs”, “micro-arrays”, etc. We then apply a suitable feature extraction method and a corresponding matching metric to each of the semantic classes.

As part of the dissertation, we built an experimental SIMPLIcity (Semantics-sensitive Integrated Matching for Picture Libraries) system, targeted for applications such as Web and biomedical image retrieval. We compare the SIMPLIcity system to a system without high-level semantic classification, two color histogram systems, and the WBIIS system (Wavelet Based Image Indexing and Searching) [122, 118].

### 1.4.2 Image classification

For the purpose of searching picture libraries such as those on the Web or in a patient digital library, we are initially focusing on techniques to classify images into the classes “textured” vs. “non-textured”, “graph” vs. “photograph”, and “objectionable” vs. “benign”. Several other classification methods have been previously developed elsewhere, including “city” vs. “landscape” [115], and “with people” vs. “without people” [16, 9]. As a part of this dissertation, we report on several classification methods we have developed and their performance.

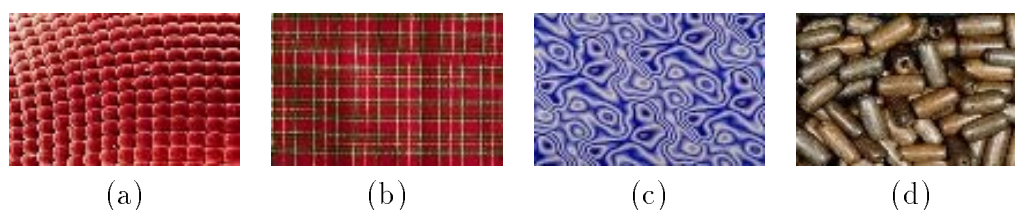


Figure 1.3: Sample textured images. (a) surface texture (b) fabric texture (c) artificial texture (d) pattern of similarly-shaped objects

A *textured* image is defined as an image of a surface, a pattern of similarly-shaped objects, or an essential element of an object. For example, the structure formed by the threads of a fabric is a textured image. Figure 1.3 shows some sample textured images. As textured images do not contain isolated objects or object clusters, the perception of such images focuses on color and texture, but not shape, which is critical for understanding non-textured images. Thus an efficient retrieval system should use different features to depict these two types of images. To our knowledge, the problem of distinguishing textured images and non-textured images has not been explored in the image retrieval literature.

An image is a *photograph* if it is a continuous-tone image. A *graph image* is an image containing mainly text, graph and overlays. We have developed a graph-photograph classification method. This method is important for retrieving general-purpose picture libraries. For example, we may apply Optical Character Recognition (OCR) techniques on graph images on the Web because they often contain textual information. Image features are suitable to photographs.

Details of our methods for *objectionable-image* classification are given as an appendix in this dissertation. Different people may have different definitions of objectionable image. We call a photograph of nude people an objectionable image. This automatic image classifier is not only part of our CBIR architecture for general-purpose images, but also a direct application of CBIR techniques we have developed.

The concepts and methods of image classification we have developed can be extended by comparison to class databases to obtain additional class dichotomies. It is possible to develop methods to categorize images into classes that are difficult to describe formally. For example, it is difficult to describe the class of images containing a dog, even though we can recognize such images base on our experiences. Similarly, radiologists often find it difficult to formally define the set of images with certain lesions, even though they are trained to recognize such images.

### 1.4.3 Integrated Region Matching (IRM) similarity measure

Besides using semantics classification, another strategy of SIMPLIcity to better capture the image semantics is to define a robust region-based similarity measure, the Integrated Region Matching (IRM) metric. That is, IRM is a similarity measure between images based on region representations. It incorporates the properties of all the segmented regions so that information about an image can be fully used. Region-based matching is a difficult problem because of the problems of inaccurate segmentation. Semantically-precise image segmentation is an extremely difficult process [69, 102, 78, 136] and is still an open problem in computer vision. For example, an image segmentation algorithm may segment an image of a dog into two regions: the dog and the background. The same algorithm may segment another image of a dog into six regions: the body of the dog, the front leg(s) of the dog, the rear leg(s) of the dog, the eye(s), the background grass, and the sky.

Traditionally, subsequent region-based matching is performed on individual regions [11, 77]. The IRM metric we have developed has the following major advantages:



1. Compared with retrieval based on individual regions, the overall “soft similarity” approach in IRM reduces the influence of inaccurate segmentation, an important property that previous work has not solved.
2. In many cases, knowing that one object usually appears with another object helps to clarify the semantics of a particular region. For example, flowers typically appear with green leaves, and boats usually appear with water.
3. By defining an overall image-to-image similarity measure, the SIMPLIcity system provides users with a *simple* querying interface. To complete a query, a user only needs to specify the query image. The image is then divided into segments (or regions). The process is called segmentation. These regions and information about the regions are used in the overall matching process. If desired, the system can also be adjusted to allow users to query based on a specific region or a few regions.

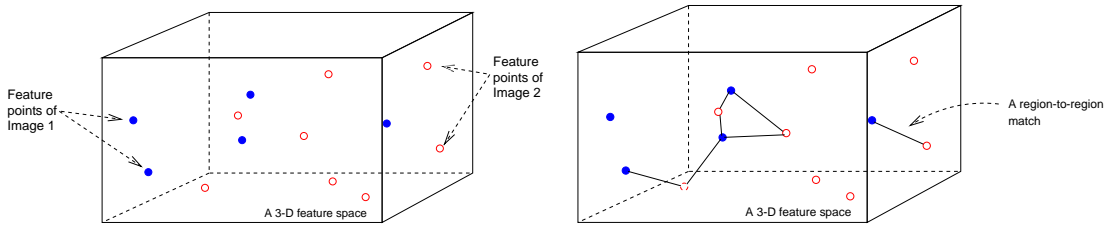


Figure 1.4: Region-to-region matching results are incorporated in the Integrated Region Matching (IRM) metric. A 3-D feature space is shown to illustrate the concept.

Mathematically, defining a similarity measure is equivalent to defining a distance between sets of points in a high-dimensional space, i.e., the feature space. Every point in the space corresponds to the feature vector, or the descriptor, of a region. Although distance between two points in feature space can be easily defined by various measures such as the Euclidean distance, it is not obvious how to define a distance between two sets of feature points. The distance must correspond to a person’s concept of semantic “closeness” of two images.

We argue that a similarity measure based on region segmentation of images can be tolerant of inaccurate image segmentation if it takes all the regions in an image into

consideration. To define the similarity measure, we first attempt to match regions in two images. Being aware that segmentation process cannot be perfect, we “soften” the matching by allowing one region of an image to be matched to several regions of another image. Here, a region-to-region *match* is obtained when the regions are significantly similar to each other in terms of the features extracted.

The principle of matching is that the most similar region pair is matched first. We call this matching scheme *integrated region matching* (IRM) to stress the incorporation of regions in the retrieval process. After regions are matched, the similarity measure is computed as a weighted sum of the similarity between region pairs, with weights determined by the matching scheme. Figure 1.4 illustrates the concept of IRM in a 3-D feature space. The features we extract on the segmented regions are of high dimensions. The problem is more complex in a high-dimensional feature space.

#### 1.4.4 Applications of the methods

We implemented the SIMPLIcity experimental system and applied it to several domains including the screening of objectionable images and Web sites [119, 125], picture libraries [130, 71, 72, 73], and biomedical image databases [131, 129]. Promising results have been reported in our papers and will be summarized in this dissertation.

### 1.5 Structure of dissertation

The remainder of the dissertation is organized as follows:

- **Chapter 2. Related work**

Both content-based image retrieval (CBIR) and semantic classification of images are active and interrelated research fields within computer vision. CBIR is a technique for retrieving relevant images from an image database on the basis of automatically-derived image features. Semantic classification of images is a technique for classifying images based on their semantics. We review the related work in content-based image retrieval and semantic classification of images. Examples of their biomedical applications are also given in this chapter.

- **Chapter 3. Wavelets**

The accurate representation of a local region, i.e., the localization property in the context of signal processing, is a prime consideration when processing signals and images (a signal of two dimensions). Many mathematical transforms using *basis functions* provide ways to gain insight of the fundamental properties of the given signals or images. For example, the Fourier transform, based on the sine functions and the cosine functions, is used to analyze signals in different frequency bands.

For the purpose of image retrieval, we seek a basis function that can effectively represent the color variations in each local spatial region of the image. We examine the various mathematical transforms and their properties to select a transform (the wavelet transform) that has attractive properties for the image retrieval problem.

- **Chapter 4. Statistical clustering and classification**

Statistical clustering and classification methods are tools that search for and generalize concepts based on a large amount of high-dimensional numerical data. In modern CBIR systems, statistically clustering and classification methods are often used to extract visual features, index the feature space, and classify images into semantic categories. In our work, we apply statistical clustering to the block-wise feature space to extract region features. For very large databases, we use statistical clustering methods to index the high-dimensional feature space. The semantic classification process is a statistical classification process.

We briefly review the statistical clustering methods we have used, the k-means algorithm and the Tree-Structured Vector Quantization (TSVQ) algorithm. We also review the statistical classification method in our semantic classification process, the Classification and Regression Trees (CART) algorithm.

- **Chapter 5. Wavelet-based image indexing and searching**

In this chapter, we describe WBIIS (Wavelet-Based Image Indexing and Searching), an image indexing and retrieval algorithm we developed in 1996 as a first

step in image retrieval using wavelets. The indexing algorithm uses wavelet coefficients and their statistical properties to represent the content of an image. Daubechies' advanced wavelet transforms are utilized. To speed up retrieval, a two-step procedure is used that first does a crude selection based on the statistics of the coefficients, and then refines the search by performing a feature vector match between the selected images and the query. For better accuracy in searching, two-level multiresolution matching is used. Like traditional color layout indexing, based only on color and texture, WBIIS has limitations and weaknesses.

- **Chapter 6. Semantics-sensitive integrated matching**

In this chapter, we present the main ideas and algorithms of our recently developed SIMPLIcity (Semantics-sensitive Integrated Matching for Picture Libraries), an image database retrieval system which uses high-level semantic classification and integrated region matching (IRM) based upon image segmentation. The SIMPLIcity system represents an image by a set of regions, roughly corresponding to objects, which are characterized by color, texture, shape, and location. Based on segmented regions, the system classifies images into semantically meaningful categories (e.g., graph, photograph, textured, non-textured, objectionable, benign, indoor, outdoor). These high-level categories enhance retrieval by narrowing down the searching range in a database and permitting semantically-adaptive searching methods to be used. Details of the classification methods and the IRM metric are provided.

We then describe the Pathfinder, a system we developed specially for retrieving biomedical images of extremely high resolution, based on wavelet feature extraction, progressive wavelet image indexing, and the IRM matching metric.

In **Appendix A**, we discuss image classification by image database matching. The idea of high-level image semantic classification in our SIMPLIcity CBIR system came from the Web image classification project. We also developed the idea of categorizing an image by comparing it to an exemplar category database from the objectionable-image classification application. The classifier

is useful not only for the problem of CBIR, but also in preventing children from accessing objectionable documents. As hospitals are equipped with Internet-capable terminals, proper screening of objectionable media on the Web becomes necessary.

- **Chapter 7. Evaluation**

We introduce the experimental system we have developed, the SIMPLIcity system, using the concepts we proposed in previous chapters. Due to the difficulty of collecting a large number of radiology or pathology images, we provided comprehensive evaluation on general-purpose images as well as evaluation on relatively small medical image databases. We present the data sets we used for the experiments, the functions of the query interfaces, the accuracy comparisons, the robustness to image alterations or feature variations, and the speed of indexing and searching.

- **Chapter 8. Conclusions and future work**

We conclude the dissertation in this chapter. We summarize the main themes of our research on semantics-sensitive integrated matching for picture libraries and biomedical image databases. Then we examine limitations of our solution, and discuss how our results might be affected when our work is applied to a real-world image database. Areas of future work are indicated.

## 1.6 Summary

In summary, we discuss the aspects of designing a content-based image retrieval (CBIR) system in this dissertation. Our contributions include the development of a novel architecture, several specific image classification methods, and an integrated matching measure for region-based feature indexing. The methods are implemented in an experimental system, the SIMPLIcity system, and has been applied to both large-scale picture libraries and biomedical image databases.

# Chapter 2

## Related Work

*It's kind of fun to do the impossible.*

— Walter E. Disney (1901-1966)

### 2.1 Introduction

Content-based image retrieval (CBIR) and image semantic classification are both active research fields. They are interrelated topics within computer vision. CBIR is a technique for retrieving relevant images from an image database on the basis of automatically-derived image features. Image semantic classification is a technique for classifying images based on their semantics. Semantically-adaptive searching methods applicable to each category can then be applied. In this chapter, we review the related work in content-based image retrieval and image semantic classification, and also provide examples of their biomedical applications.

We review related work in content-based image database retrieval in Section 2.2, and related work in image semantic classification in Section 2.3. In each of the sections, we give examples of the biomedical applications.

## 2.2 Content-based image retrieval

CBIR for general-purpose image databases is a highly challenging problem because of the large size of the database, the difficulty of understanding images, both by people and computers, the difficulty of formulating a query, and the problem of evaluating the results. These challenges and related previous work are discussed below.

### 2.2.1 Major challenges

#### Data size

To build an image search engine for domains such as biomedical imaging and the World-Wide Web, it is necessary to store and process millions of images efficiently. Without compression, a database of one million normal resolution images takes about 1000 gigabytes (GB) of space. Even with efficient, lossy compression, 30 GB of space is required. It is clearly not feasible to process all the image data for each query on-line. Off-line feature-based indexing is necessary to reduce the computation for the actual query processing.

Biomedical images are typically of high resolution. Pathology slides are scanned at approximately  $3600 \times 2400$  pixels. Each slide consists of 26 million bytes (MB) of information. Some radiology images are even larger. A gray-scale 4-spectrum 100-slice MRI scan consists of roughly 200 MB of data. Furthermore, a typical hospital creates tens of thousands of scans per year.

#### Understandability and computer vision

It is said that *a picture is worth a thousand words*. Indeed, it often can require a thousand words to describe the content of a picture. Moreover, the “meaning” of a picture depends on the *point-of-view* and experience of the viewer. That is, the description provided by one viewer may be very different from the description provided by another viewer of the same image. For example, a builder and an architect may have a different point-of-view on the same blueprint. In medicine, inter- and intra-observer variabilities for identifying multiple sclerosis lesion regions in MRI are as

high as 17-30% and 6%, respectively [56]. Computer vision remain as one of the most challenging unsolved problems in computer science. It is extremely difficult to recognize objects in pictures and translate image content and structure into linguistic terms.

### Query formulation

A successful computerized information retrieval system must take the user into consideration. After all, most information retrieval systems are developed to enhance the ability of *human* in obtaining information. The interface between the human and information has always been the focus of library science. The interface between the human and technology is considered to be in the realm of psychology. Computer scientists and informaticians work primarily in the interface between information and technology or within the technology itself. We argue that a joint effort among library science, psychology, and computer and information sciences is essential in building practical information retrieval systems such as CBIR systems.

The indexing algorithm of a successful CBIR system must be tailored to real-world applications. The requirements of image database users vary significantly, based on the domains of the users. For instance, a clinical trial database user is interested in tracking the progression of lesions over time. A user of a satellite image database is interested in images with regions having certain patterns. A visitor to a museum Web site may be interested in painting collections with styles similar to a given painting. Consequently, it is necessary to understand the ways in which users of the system search for images before attempting to design the indexing strategies and to build a system to meet their needs.

At the current stage of development, computer-based image technology is not mature enough to handle the disparate needs of people in a variety of fields. Our work focuses on improving computer-based CBIR technology with emphases on general-purpose picture libraries and biomedical image databases.

Most existing systems index images by their primitive features, such as the color histogram, and target on *similarity* search. That is, the system is designed to answer queries (Figure 2.1) such as:



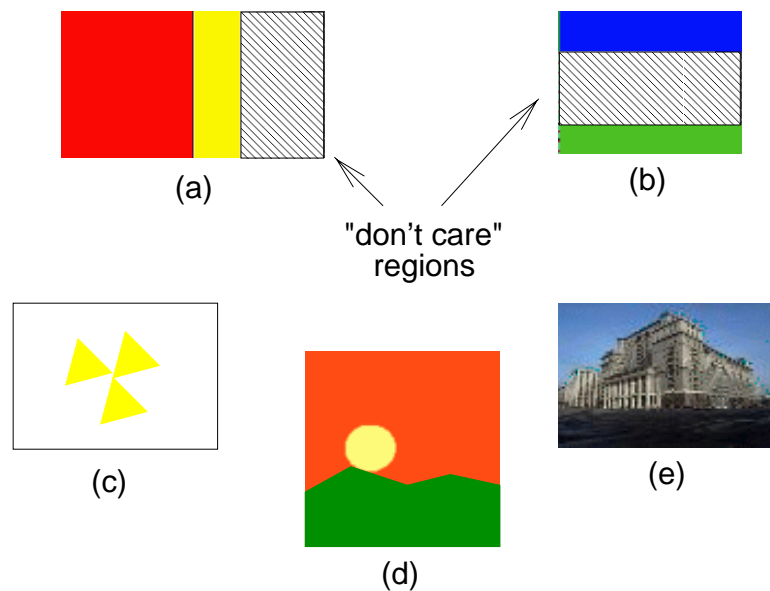


Figure 2.1: Queries to be handled by CBIR systems using primitive feature indexing. (a) histogram query (b) layout query (c) shape query (d) hand-drawn sketch query (e) query by example



- Histogram query: find pictures with 50% red and 20% yellow (Figure 2.1(a))
- Layout query: find pictures with a blue object on the top part and a green object on the bottom part (Figure 2.1(b))
- Shape query: find pictures with three yellow triangular stars arranged in a ring (Figure 2.1(c))
- hand-drawn sketch query: find pictures that look like a given drawing (Figure 2.1(d))
- query by example: find pictures that look like a given picture (Figure 2.1(e))

However, most CBIR users are interested in queries involving high-level semantics. Examples include:

- *Object*: contains a lesion
- *Object relationship*: contains a lesion near the cerebro-spinal fluid (CSF)

- *Mood*: a happy picture
- *Time/Place*: Yosemite evening

1. Select up to two regions

2. Fill out this form for each region

	Not	Somewhat	Very
How important is the selected region?	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
.....			
How important are the features of this region?			
Color	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Texture	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Location	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Shape/Size	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
		<b>Not</b>	<b>Somewhat</b>
How important is the background (everything outside the region)?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Figure 2.2: Query interface of the Blobworld system developed at the University of California, Berkeley.

Users are typically reluctant to use a computerized system unless the interaction is simple and intuitive. In a recent study [60], Jansen et al. report that, over a sample of 51,473 text queries submitted to a major search service (Excite.com), the average length of query was less than three words and that more than 30% of the queries were single-term queries. Clearly, complicated queries such as Boolean queries are not often used in retrieval situations, even though provision for such queries is available to users at major search sites and single terms in English are often overloaded.

A simple user interface is also critical for image database retrieval systems. A study conducted by the IBM QBIC group has shown that users are likely to use the simplest searching interface (i.e., click on an image to find similar images), in preference to a set of much more sophisticated user interfaces, such as drawing tools. Figure 2.2 shows the user interface for the Blobworld system [11] developed by Carson et al. of the University of California at Berkeley. The NeTra region-based image retrieval system developed by Ma et al. of the University of California at Santa Barbara uses an even more complicated user interface<sup>1</sup>. Despite the fact that it takes minutes for a typical user to specify a query, the query results are often not better than those with much simpler query interfaces.

Recently, several CBIR systems have been developed with relevance feedback and query modification techniques [49]. Such systems provide trade-offs between the complexity and the functionality of the user interface. However, it requires that the systems be at the same high-level semantics to utilize the user feedback.

It is necessary to design an interface that is easy to use for the specific application area. For example, in general, Web users are expected to be interested in searching based on the overall structures or object semantics of the images, while users of biomedical image databases are often interested in fine details within the images. Consequently, biomedical image database users will not find it useful to find images based on a given color histogram or a given color layout sketch.

## Evaluation

There are deterministic ways to evaluate a CBIR system in specific domains such as image copyright violation and identifying online objectionable images. However, it is difficult to evaluate a general-purpose CBIR system due to the complexity of image semantics and the lack of a “gold standard”.

In the field of information retrieval, *precision* and *recall* are frequently used to evaluate text-based retrieval systems.

---

<sup>1</sup>URL: <http://maya.ece.ucsb.edu>



Figure 2.3: It is difficult to define relevance for image semantics. Left: images labeled as “dog” by photographers. Right: images labeled as “Kyoto, Japan” by photographers.

$$Precision = \frac{Retrieved\ Relevant\ Items}{Retrieved\ Items} \quad (2.1)$$

$$Recall = \frac{Retrieved\ Relevant\ Items}{Relevant\ Items} \quad (2.2)$$

However, the “relevance” in the above definitions depends on the readers’ *point-of-view*. For example, Figure 2.3 shows a group of three images labeled as “dog” and a group of three images labeled as “Kyoto, Japan”, by photographers. If we use the descriptions such as “dog” and “Kyoto, Japan” as definitions of relevance to evaluate CBIR systems, it is unlikely that we can obtain a consistent performance evaluation. A system may perform very well on one query (such as the dog query), but very poorly on another (such as the Kyoto query).

Currently, most developers evaluate the retrieval effectiveness of their systems through the following methods:

- Provide a few examples of retrieval results and compare with the results of previously developed systems or methods

- Systematic evaluation using a small database with only a few distinct categories (e.g., sunset, oranges, tigers)
- Systematic evaluation over several added distinct categories within a larger database

These evaluation methods provide meaningful information as to whether one system is better than another system. However, they are not ideal in measuring exactly how good a system is in real world. A few examples are not sufficient in evaluating complex systems. Systematic evaluation over a small categorized database is not sufficient either, because distinct categories of images may easily form distinct clusters in the feature space. It is much more difficult to retrieve a certain number of relevant images within a semantic category when the database contains a large number of images because the discriminative function will have to be more precise. In real-world applications, images in the same semantic category are often distributed near uniformly and hence rarely well-clustered in known feature spaces.

### 2.2.2 Previous work

Many CBIR systems have been developed, such as the IBM QBIC System [30, 32] developed at the IBM Almaden Research Center, the VIRAGE System [46] developed by the Virage Incorporation, the Photobook System developed by the MIT Media Lab [91, 89], the WBIIS System [118] developed at Stanford University, and the Blobworld System [11] developed at U.C. Berkeley.

The common ground for CBIR systems is to extract a signature for every image based on its pixel values, and to define a rule for comparing images. Figure 2.4 shows the architecture of a typical CBIR system. The components of a CBIR system function as follows:

- The *image manager module* manages image file access, image format conversions, and the retrieval of any textual information from the image database.
- The *feature indexing module* is an off-line process which assigns signatures to images in the database. Figure 2.5 shows the process of indexing an image using

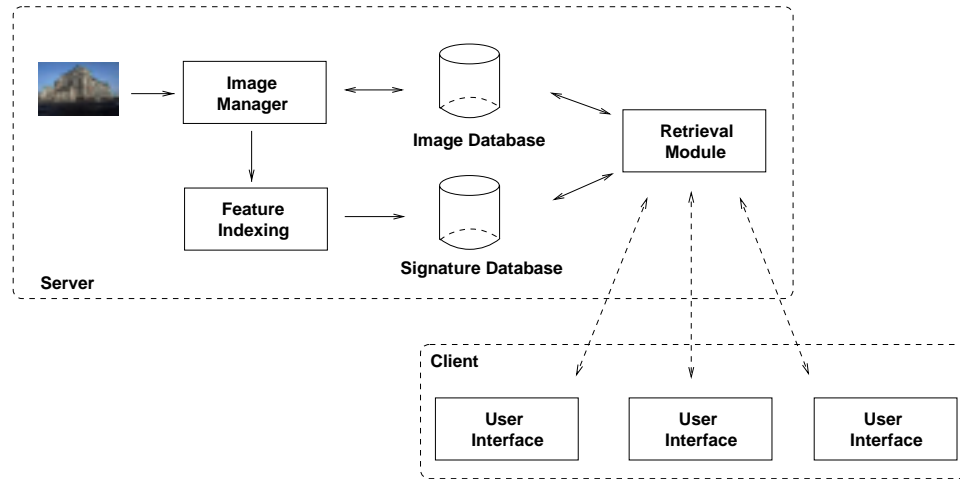


Figure 2.4: The architecture of a typical CBIR system.

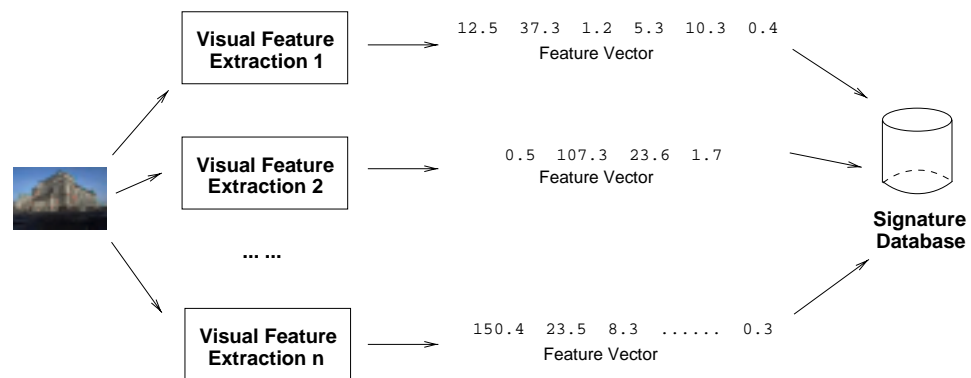


Figure 2.5: The indexing process of a typical CBIR system.

a set of feature extraction processes. The features can be further indexed in the feature space before they are stored in a signature database.

- The *retrieval module* is an on-line server handling user queries based on image content. As shown in Figure 2.7, the main functions of this module are:
  1. Accept queries from different user interfaces (interface manager)
  2. Process the query (feature extraction module)
  3. Perform image comparisons using features stored in the signature database (feature comparison module)
  4. Sort the query results (sorting module)
  5. Retrieve the relevant images from the image database (image manager)
  6. Present the querying results to the users (interface manager)
- The *user interface modules* are client programs for users to formulate content-based image queries and to visualize the query results.

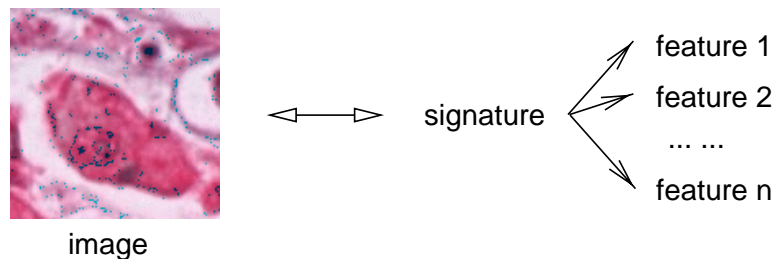


Figure 2.6: The signature of an image is a set of features.

The *signature* serves as an image representation in the ‘view’ of a CBIR system. The components of the signature are called *features*. Figure 2.6 shows the relationship among an image, its signature, and features.

One advantage of using a signature instead of the original pixel values is the significant compression of image representation. However, a more important reason for using the signature is the improved correlation with image semantics. Actually, the main task of designing a signature is to bridge the gap between image semantics and

the pixel representation, that is, to create a better correlation with image semantics. The human vision system (HVS), which may be regarded as a perfect image analyzer, after looking at an image may describe the image as “some brown horses on grass.” With the same image, a simple example signature stored in a database might be “90% of the image is green and 10% is brown.” Similarly, the HVS may describe a radiology image as “a round-shaped lesion near the cerebro-spinal fluid (CSF) region”, while the image may have a signature “a cluster of bright pixels located near the center of the image.” The CSF is the center part of the brain containing the cerebro-spinal fluid.

Existing general-purpose CBIR systems roughly fall into three categories depending on the signature extraction approach used: histogram, color layout, and region-based search. We will briefly review these methods later in this section. There are also systems that combine retrieval results from individual algorithms by a weighted sum matching metric [46, 32], or other merging schemes [101].

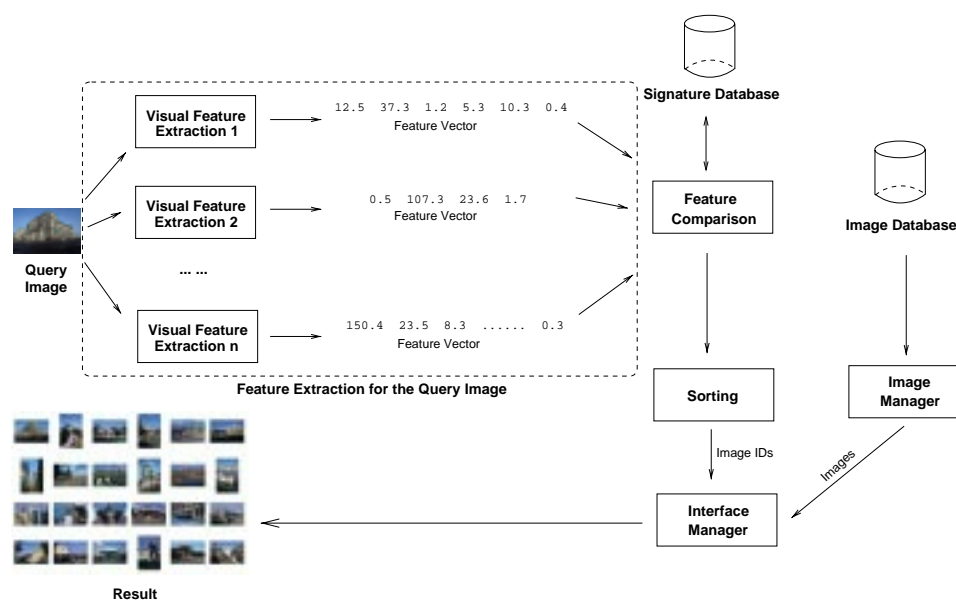


Figure 2.7: The retrieval process of a typical CBIR system.

After extracting signatures, the next step is to determine a comparison rule, including a querying scheme and the definition of a similarity measure between images. Most image retrieval systems perform a query by having the user supply an image



to be matched; the system then searches for images similar to the specified one. We refer to this as global search, since similarity is based on the overall properties of images. In contrast to global search, there are also “partial search” querying systems that retrieve based on a particular region in an image, such as the NeTra system [77] and the Blobworld system [11].

## Color spaces

Before an image can be indexed in a CBIR system, a proper transformation to a suitable color space is typically required. A *color space* is defined as a model for representing color in terms of intensity values. Typically, a color space defines a one- to four-dimensional space. A color component, or a color channel, is one of the dimensions. A one-dimensional space (i.e., one dimension per pixel) represents the gray-scale space.

Color spaces are related to each other by very simple mathematical formulas. The following is a list of commonly used color spaces in image processing and image indexing [135]:

### 1. Gray spaces

Gray spaces typically have one single component, ranging from black to white. Gray spaces are the most common color space in biomedical imaging, as most medical scanners produce 2-D or 3-D (spatially) gray-scale images and 2-D electrophoresis gels are typically of gray-scale.

### 2. RGB-based spaces

The RGB space is a three-dimensional color space with components representing the red, green, and blue intensities that make up a given color. The RGB-based spaces are commonly used for devices such as color scanners and color monitors. They are also the primary color spaces in computer graphics due to the hardware support. The family of the RGB-based spaces include the RGB space, the HSV (hue, saturation, value) space, and the HLS (hue, lightness, saturation) space.

Any color expressed in the RGB space is a mixture of three primary colors: red, green, and blue. For example, the color cyan can be viewed as the combination of the blue color and the green color.

The HSV space and the HLS space are transformations of RGB space that can describe colors in terms more natural to a person. The HSV and the HLS spaces are slightly different in their mathematical definitions.

### 3. CMYK-based spaces

CMYK stands for Cyan, Magenta, Yellow, and black. CMYK-based color spaces model the way dyes or inks are applied to paper in the printing or drawing process. Ideally, the relation between RGB values and CMYK values is as simple as:

$$\left\{ \begin{array}{l} C = max - R \\ M = max - G \\ Y = max - B \\ K = 1 \text{ when } R = G = B = 0 \\ K = 0 \text{ when } R \neq 0, G \neq 0, \text{ or } B \neq 0 \end{array} \right. \quad (2.3)$$

Here  $max$  is the maximum possible value for each color component in the RGB color space. For a standard 24-bit color image,  $max = 255$ .

### 4. CIE-based spaces

The RGB color spaces and the CMYK color spaces are all *device-dependent* because they were developed mainly to facilitate computer devices including monitors and printers. They are not very well correlated to the human perception. There are classes of color spaces that can express color in a device-independent way. They are based on the research work done in 1931 by the Commission Internationale d'Eclairage (CIE). They are also called interchange color spaces because they are used to convert color information from the native color space

of one device to the native color space of another device. XYZ, CIE LUV, CIE Lab are examples of the CIE-based color spaces [135].

The CIE-based color spaces simulates human color perception. Research in human vision has revealed that three *sensations* are generated after the sensory membrane in the eye (or the retina) receives three color stimuli (red, green and blue). The three sensations are a red-green sensation, a yellow-blue sensation, and a brightness sensation. The CIE-based color spaces are considered a global color reference systems because of its perception correlation properties.

### Histogram search

Histogram search [88, 32, 96] characterizes an image by its color distribution, or histogram. For example, to index a color image of  $1024 \times 1024 = 1M$  pixels in the standard 3-D RGB color space, we may generate  $8 \times 8 \times 8 (= 512)$  number of counting bins, each representing a range of 32 values in each of the three color components. By counting the pixels falling into these bins, a color histogram feature of 512 dimensions is created to represent the color image. The components of this 512-dimensional feature vector represent a distribution of colors in an image, without considering the location of the colors.

Many distances have been used to define the similarity of two color histogram representations. Euclidean distance and its variations are the most commonly used [47, 81]. A spatial-augmented histogram matching has recently been proposed [17]. Rubner et al. of Stanford University proposed an earth mover's distance (EMD) [96, 97] using linear programming [51] for matching histograms.

The drawback of a global histogram representation is that information about object location, shape, and texture is discarded. Figure 2.8 shows two sample query results. The global color histogram indexing method correlates to the image semantics well in the first example. However, images retrieved in the second example are not all semantically related, even though they share similar color distribution. Color histogram search is sensitive to intensity variation, color distortions, and cropping.



(a) good result



(b) poor result

Figure 2.8: Two sample color histogram query results, one good, one poor. The image in the upper-left corner of each block is the query image. DB size: 10,000 images.

### Color layout search

The “color layout” approach attempts to mitigate the problems with histogram search. For traditional color layout indexing [88], images are partitioned into blocks and the average color of each block is stored. Thus, the color layout is essentially a low resolution representation of the original image. A later system, WBIIS [118] (see Chapter 5 for details), uses significant Daubechies’ wavelet<sup>2</sup> coefficients instead of averaging. By adjusting block sizes or the levels of wavelet transforms, the coarseness of a color layout representation can be tuned. The finest color layout using a single pixel block is the original pixel representation. We can hence view a color layout representation as an opposite extreme of a histogram. At proper resolutions, the color layout representation naturally retains shape, location, and texture information. However, as with pixel representation, although information such as shape is preserved in the color layout representation, the retrieval system cannot “see” it explicitly. Color layout search is sensitive to shifting, cropping, scaling, and rotation because images are characterized by a set of local properties [118].

The approach taken by the recent WALRUS system [86] to reduce the shifting and scaling sensitivity for color layout search is to exhaustively reproduce many subimages based on an original image. The subimages are formed by sliding windows of various sizes and a color layout signature is computed for every subimage. The similarity between images is then determined by comparing the signatures of subimages. An obvious drawback of the system is the sharply increased computational complexity and increase of size of the search space due to exhaustive generation of subimages. Furthermore, texture and shape information is discarded in the signatures because every subimage is partitioned into four blocks and only average colors of the blocks are used as features. This system is also limited to intensity-level image representations.

### Region-based search

Region-based retrieval systems attempt to overcome the deficiencies of color layout search by representing images at the object-level. A region-based retrieval system

---

<sup>2</sup>Details about wavelets are given in Chapter 3

applies image segmentation [69, 70, 120, 102, 78, 136] to decompose an image into regions, which correspond to objects if the decomposition is ideal. The object-level representation is intended to be close to the perception of the human visual system (HVS). However, semantically-precise image segmentation is nearly as difficult as image understanding because the images are 2-D projections of 3-D objects and computers are not trained in the 3-D world the way human beings are.

Since the retrieval system has identified what objects are in the image, it is easier for the system to recognize similar objects at different locations and with different orientations and sizes. Region-based retrieval systems include the NeTra system [77], the Blobworld system [11], and the query system with color region templates [107].

The NeTra and the Blobworld systems compare images based on individual regions. Although querying based on a limited number of regions is allowed, the query is performed by merging single-region query results. The motivation is to shift part of the comparison task to the users. To query an image, a user is provided with the segmented regions of the image, and is required to select the regions to be matched and also attributes, e.g., color and texture, of the regions to be used for evaluating similarity. Such querying systems provide more control for the users. However, the key pitfall is that the user's semantic understanding of an image is at a higher level than the region representation. When a user submits a query image of a horse on grass, the intent is most likely to retrieve images with horses. But since the concept of horses is not explicitly given in region representations, the user has to convert the concept into shape, color, texture, location, or combinations of them. For objects without distinctive attributes, such as special texture, it is not obvious for the user how to select a query from the large variety of choices. Thus, such a querying scheme may add burdens on users without any corresponding reward. On the other hand, because of the great difficulty of achieving accurate segmentation, systems in [77, 11] tend to partition one object into several regions with none of them being representative for the object, especially for images without distinctive objects and scenes. Queries based on such regions often yield images that are indirectly related to the query image.

Not much attention has been paid to developing similarity measures that combine

information from all of the regions. One effort in this direction is the querying system developed by Smith and Li [107]. Their system decomposes an image into regions with characterizations pre-defined in a finite pattern library. With every pattern labeled by a symbol, images are then represented by region strings. Region strings are converted to composite region template (CRT) descriptor matrices that provide the relative ordering of symbols. Similarity between images is measured by the closeness between the CRT descriptor matrices. This measure is sensitive to object shifting since a CRT matrix is determined solely by the ordering of symbols. Robustness to scaling and rotation is also not considered by the measure. Because the definition of the CRT descriptor matrix relies on the pattern library, the system performance depends critically on the library. The performance degrades if region types in an image are not represented by patterns in the library. The system in [107] uses a CRT library with patterns described only by color. In particular, the patterns are obtained by quantizing color space. If texture and shape features are used to distinguish patterns, the number of patterns in the library will increase dramatically, roughly exponentially in the number of features if patterns are obtained by uniformly quantizing features.

### 2.2.3 CBIR for biomedical image databases

CBIR is more challenging in biomedical domain than in many general-purpose image domains. The main reason is that important features in biomedical images are often local features rather than global features. This makes feature extraction much more demanding. Features extracted for biomedical images must be able to both describe fine details of the images and allow quick retrieval of relevant images.

Shyu et al. [104] of Purdue University have developed a semi-automatic hierarchical human-in-the-loop (or physician-in-the-loop) system. The human delineates the pathology-bearing regions (PBR) and a set of anatomical features of the image at the time the image is entered into the database. From these marked regions, the system applies low-level computer vision and image processing algorithms to extract features related to the variations of gray scale, texture, shape, etc. To form an image-based query the physician first marks some interesting regions. The system then extracts

the relevant image features, computes the distance of the query image to all image indices in the database. Promising results have been reported for some small medical image databases. However, the approach is not suitable to very large image databases due to the inefficiency of the manual labeling process.

Retrieval systems have been developed in specific fields such as brain MRI. Liu et al. [75] of the Robotics Institute of Carnegie Mellon University have developed a system for 3-D neuroradiology image databases. The system is designed for a 3-D MRI of the human brain, and requires registration of the 3-D MRI images in the database. A combination of both visual and collateral information is used for indexing and retrieval. The system is specialized to the neuroradiology application.

More biomedical CBIR efforts are summarized in a recent book edited by S. Wong [134]. They are:

- The interface and query formulation work at the University of California, Los Angeles
- The tumor shape matching work at the University of Maryland
- Using artificial neural network for event detection at the University of Pittsburgh
- Medical image filtering at Stanford University
- Remote sensing for public health at the IBM T.J. Watson Research Center
- Integration of CBIR in decision support at the Tokyo Medical and Dental University

Most biomedical systems are domain-specific. In this dissertation, we address the general problems associated with content-based image retrieval. Techniques and methods we have developed are applicable to different types of images, including picture libraries and biomedical images.



## 2.3 Image semantic classification

The purpose of CBIR is to retrieve relevant images from an image database on the basis of automatically-derived image features. The underlying assumption is that semantically-relevant images have similar visual characteristics, or features. Consequently, a CBIR system is not necessarily capable of understanding image semantics.

Image semantic classification, on the other hand, is a technique for classifying images based on their semantics. While image semantics classification is a limited form of image understanding, the goal of image classification is not to understand images the way human beings do, but merely to assign the image to a semantic class. Image class membership is then used in the retrieval process. In this section, we discuss the related work in classification of images.

Despite the fact that it is currently impossible to reliably recognize objects in general-purpose images, there are methods to distinguish certain semantic types of images. Any information about semantic types is helpful since a system can constrict the search to images with a particular semantic type. The semantic classification schemes can also improve retrieval by using various matching schemes tuned to the semantic class of the query image. Most of these classification schemes use statistical classification methods based on training data.

### 2.3.1 Semantic classification for photographs

Although region-based systems attempt to decompose images into constituent objects, a representation composed of pictorial properties of regions is indirectly related to its semantics. There is no clear mapping from a set of pictorial properties to semantics. An approximately round brown region might be a flower, an apple, a face, or part of a sunset sky. Moreover, pictorial properties such as color, shape, and texture of an object vary dramatically in different images. If a system understood the semantics of images and could determine which features of an object are significant, it would be capable of fast and accurate search. However, due to the great difficulty of recognizing and classifying images, not much success has been achieved in identifying high-level semantics for the purpose of image retrieval. Therefore, most systems are confined to

matching images with low-level pictorial properties.

One example of semantic classification is the identification of natural photographs versus artificial graphs generated by computer tools [68, 119]. The classifier breaks an image into blocks and segments every block into either of the two classes. If the percentage of blocks classified as photograph-type is higher than a threshold, the image is marked as photograph; otherwise, it is marked as text.

Other examples include the ‘WIPE’ system to detect objectionable images developed by Wang et al. [119] and an earlier system by Fleck et al. [31, 34] of University of California at Berkeley. Details of the WIPE system will be given in Appendix A of this dissertation. WIPE uses training images and CBIR to determine if a given image is closer to the set of objectionable training images or the set of benign training images. The system developed by Fleck et al., however, is more deterministic and involves a skin filter and a human figure grouper.

Szummer and Picard [112] has developed a system to classify indoor and outdoor scenes. Classification over low-level image features such as color histogram and DCT coefficients is performed. A 90% accuracy has been reported over a database of 1300 images from Kodak.

Wang and Fischler [128] have shown that rough but accurate semantic understanding can be very helpful in computer vision tasks such as image stereo matching.

Vailaya et al. of Michigan State University has developed a classification method for city vs. landscape images [115]. Low-level features such as the color histogram, the color coherence vector DCT coefficient, and the edge direction histogram are used in the classification process. Higher than 90% accuracy has been reported on an image database of 2,716 images.

Face detection [16, 9] from color images is an active research area. In 1995, Chen et al. [16] developed a real-time face detection program using parallel computers. The face candidates are detected by finding “face-like” regions in the input image using the fuzzy pattern matching method. A perceptually uniform color space is used to obtain reliable results. Wang et al. of Stanford University proposed the use of reliable face detection in a medical image security application to blur patient faces appearing in pathology images before distributing the image over the Internet [121].

### 2.3.2 Medical image classification

In the medical domain, image classification can be categorized into global classification and local classification, based on the targeted tasks.

- Global classification

To identify images with lesions (abnormal cases) and images without lesions (normal cases).

Researchers at the University of Chicago Medical Center [12] have developed and deployed the first clinical computer-assisted system to read mammograms. The system assists radiologists specializing in mammography in determining whether a mammogram is normal. Computer-aided diagnosis is expected to greatly reduce the number of breast cancers missed by radiologists.

A recent review article by Duncan and Ayache [28] provides more examples of medical image classification research efforts over the past two decades.

- Local classification or image segmentation

To identify the regions, or set of pixels, representing certain objects (e.g., the lesions, the blood vessels).

Among others [62, 41, 7], Udupa and Grossman's research group at the University of Pennsylvania [113] has designed and implemented an algorithm to segment and estimate the volume of MS lesions in MRI. Their algorithms involve a fuzzy-connectedness principle. A human operator indicates a few points in the images by pointing to the white matter, the gray matter, and the cerebrospinal fluid (CSF). Each of these objects is then completed as a fuzzy connected set using region growing. The remaining areas are potential lesion sites which are utilized to detect each potential lesion as a three-dimensional (3-D) fuzzy connected object. These objects are then presented to the operator who indicates acceptance/rejection. As indicated in [113], a coefficient of variation<sup>3</sup> (due

---

<sup>3</sup>Defined as the ratio of standard deviation to mean.

to subjective operator actions) of 0.9% (based on 20 patient studies, three operators, and two trials) for volume and a mean false-negative volume fraction of 1.3%, with a 95% confidence interval of 0%-2.8% (based on ten patient studies) has been obtained. However, a consistent algorithm may not produce correct classification consistently.

## 2.4 Summary

CBIR is a technique for retrieving relevant images from an image database on the basis of automatically-derived image features. CBIR systems can be categorized roughly into histogram, layout and region-based systems. Image semantic classification is a technique for classifying images based on their semantics. By using image semantic classification as an initial step in CBIR, we permit semantically-adaptive searching methods and reduce the searching range in a database. We discussed related work in the field of CBIR and image classification. Examples of their biomedical applications were provided.

# Chapter 3

## Wavelets

*Mathematics is the art of giving the same name to different things.*

— Jules Henri Poincare (1854-1912)

### 3.1 Introduction

When constructing basis functions of a transform, the prime consideration is the localization, i.e., the characterization of local properties, of the basis functions in time and frequency. The signals we are concerned with are 2-D color or gray-scale images, for which the time domain is the spatial location of a pixel, and the frequency domain is the color variation around a pixel. We thus seek a transform that can effectively represent color variations in any local spatial region of the image so that selected coefficients of this transform can be used in the image feature vector. In this chapter, we compare various transforms and their properties to select a transform suitable for CBIR.

We briefly review the Fourier transform in Section 3.2. In Section 3.3 we discuss wavelet transforms, including the Haar transform and the Daubechies' wavelet transforms. Several related applications of wavelets are introduced in Section 3.4.

## 3.2 Fourier transform

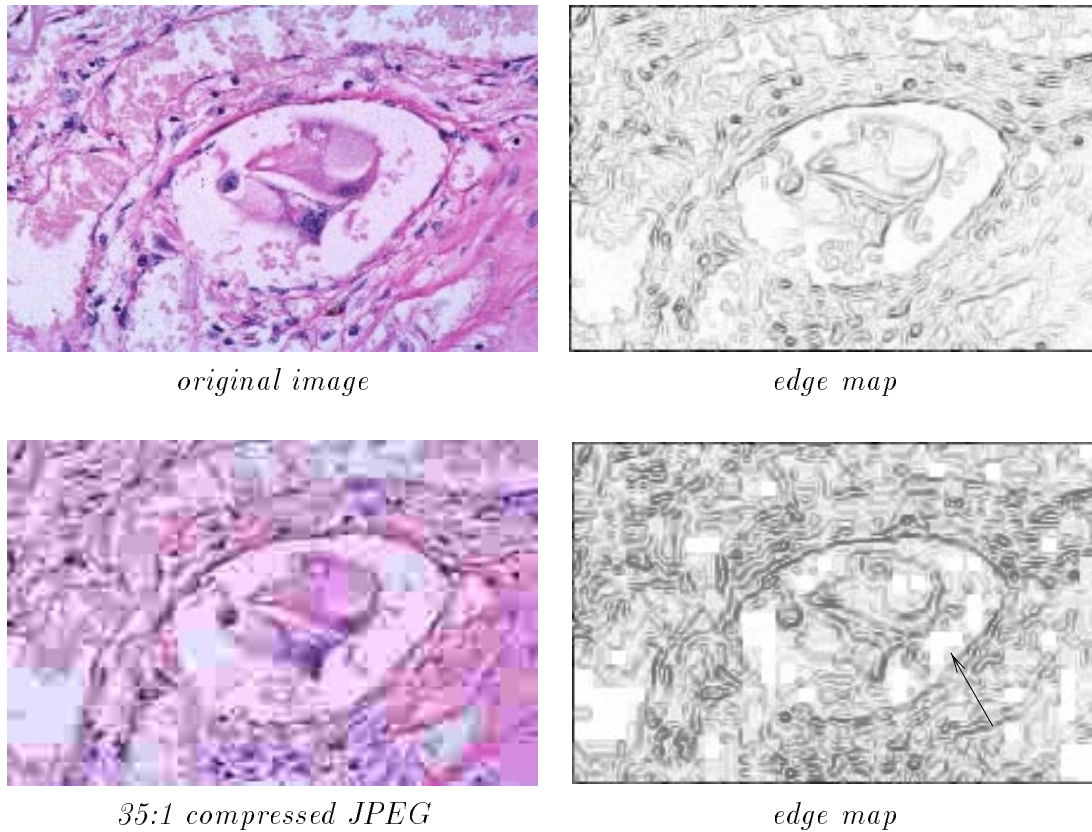


Figure 3.1: The Fourier transforms create visible boundary artifacts.

Fourier-based transforms [21, 33], such as the Discrete Cosine Transform (DCT), aim to capture the frequency content of a signal. The Discrete Fourier Transform and its inverse are defined as

$$F[k] = \sum_{n=0}^{N-1} f[n] e^{-j2\pi nk/N} \quad (3.1)$$

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} F[k] e^{j2\pi nk/N}. \quad (3.2)$$

The Discrete Fourier Transform is widely used in signal and image processing, and has proven effective for numerous problems because of its frequency domain

localization capability. It is ideal for analyzing periodic signals because the Fourier expansions are periodic. However, it lacks spatial localization due to the infinitely extending basis functions. Spline-based methods are efficient for analyzing the spatial localization of signals containing only low frequencies.

Image compression and CBIR share an important goal, i.e., to reduce the number of bits needed to represent the content of the original image. We now study the advantages and disadvantages of using the DCT transform for image compression.

The DCT is used in the JPEG (Joint Photographic Experts Group) compression coding and decoding (codec) approach and standard. In order to improve spatial localization, JPEG divides images into  $8 \times 8$  non-overlapping pixel blocks and applies the DCT to each block. The system quantizes each of the 64 frequency components uniformly using a quantization step specified by a table based on the human visual system (HVS)'s sensitivity to different frequencies. A smaller quantization step is assigned to a frequency component to which the HVS is more sensitive so that this component is encoded with higher accuracy. Entropy encoding is applied using the Huffman code to further reduce the bits needed for the representation.

Although spatial domain localization is improved by the windowed DCT, processing the  $8 \times 8$  non-overlapping blocks separately results in boundary artifacts at block borders, as shown in Figure 3.1. Wavelet-based compression methods typically generate much less visible artifacts at the same compression ratio.

### 3.3 Wavelet transform

Two important mathematical methods are available for non-periodic signals, the Windowed Fourier Transform (WFT) and the wavelet transform. WFT analyzes a signal in spatial and frequency domains simultaneously by examining the portion of the signal constrained in a moving window with fixed shape. Therefore, a signal is likely under-localized or over-localized in spatial domain since the spatial localization of WFT is restricted by the window. Wavelets are basis functions with similarities to both splines and Fourier series. They are more efficient than WFT for analyzing aperiodic signals, such as images, that contain impulsive sharp changes.

Wavelets, studied in mathematics, quantum physics, statistics, and signal processing, are functions that decompose signals into different frequency components, each with a resolution matching its scale [19]. There is active research [98, 109] in applying wavelets to signal denoising, image compression, image smoothing, fractal analysis, and turbulence characterization [109, 98].

### 3.3.1 Haar wavelet transform

Wavelet analysis can be based on an approach developed by Haar [84]. In 1909, A. Haar described an orthonormal bases (in an appendix to his thesis), defined on  $[0, 1]$ , namely  $h_0(x), h_1(x), \dots, h_n(x), \dots$ , other than the Fourier bases, such that for any continuous function  $f(x)$  on  $[0, 1]$ , the series

$$\sum_{j=1}^{\infty} \langle f, h_j \rangle h_j(x) \quad (3.3)$$

converges to  $f(x)$  uniformly on  $[0, 1]$ . Here,  $\langle u, v \rangle$  denotes the inner product of  $u$  and  $v$ :

$$\langle u, v \rangle = \int_0^1 u(x) \overline{v(x)} dx, \quad (3.4)$$

where  $\bar{v}$  is the complex conjugate of  $v$ , which equals  $v$  if the function is real-valued.

One version of Haar's construction [84, 19, 20] is as follows:

$$h(x) = \begin{cases} 1, & x \in [0, 0.5) \\ -1, & x \in [0.5, 1) \\ 0, & \text{elsewhere} \end{cases} \quad (3.5)$$

$$h_n(x) = 2^{j/2} h(2^j x - k) \quad (3.6)$$

where  $n = 2^j + k$ ,  $k \in [0, 2^j)$ ,  $x \in [k2^{-j}, (k+1)2^{-j})$ .

There are limitations in using Haar's construction. Because Haar's base functions



are discontinuous step functions, they are not suitable for analyzing smooth functions with continuous derivatives. Since images often contain smooth regions, the Haar wavelet transform does not provide satisfactory results in many image applications.

### 3.3.2 Daubechies' wavelet transform

Another type of basis for wavelets is that of Daubechies. For each integer  $r$ , the orthonormal basis [23, 24, 63] for  $L^2(\mathbb{R})$  is defined as

$$\phi_{r,j,k}(x) = 2^{j/2} \phi_r(2^j x - k), \quad j, k \in \mathbb{Z} \quad (3.7)$$

where the function  $\phi_r(x)$  in  $L^2(\mathbb{R})$  has the property that  $\{\phi_r(x - k) | k \in \mathbb{Z}\}$  is an orthonormal sequence in  $L^2(\mathbb{R})$ . Here,  $j$  is the scaling index,  $k$  is the shifting index, and  $r$  is the filter index.

Then the *trend*  $f_j$ , at scale  $2^{-j}$ , of a function  $f \in L^2(\mathbb{R})$  is defined as

$$f_j(x) = \sum_k \langle f, \phi_{r,j,k} \rangle \phi_{r,j,k}(x). \quad (3.8)$$

The *details* or *fluctuations* are defined by

$$d_j(x) = f_{j+1}(x) - f_j(x). \quad (3.9)$$

To analyze these details at a given scale, we define an orthonormal basis  $\psi_r(x)$  with properties similar to those of  $\phi_r(x)$  described above.

$\phi_r(x)$  and  $\psi_r(x)$ , called the *father wavelet* and the *mother wavelet*, respectively, are the wavelet prototype functions required by the wavelet analysis. Figure 3.2 shows several popular mother wavelets. The family of wavelets such as those defined in Eq.( 3.7) are generated from the father or the mother wavelet by changing scale and translation in time (or space in image processing).

Daubechies' orthonormal basis has the following properties:

- $\psi_r$  has the compact support interval  $[0, 2r + 1]$ ;
- $\psi_r$  has about  $r/5$  continuous derivatives;

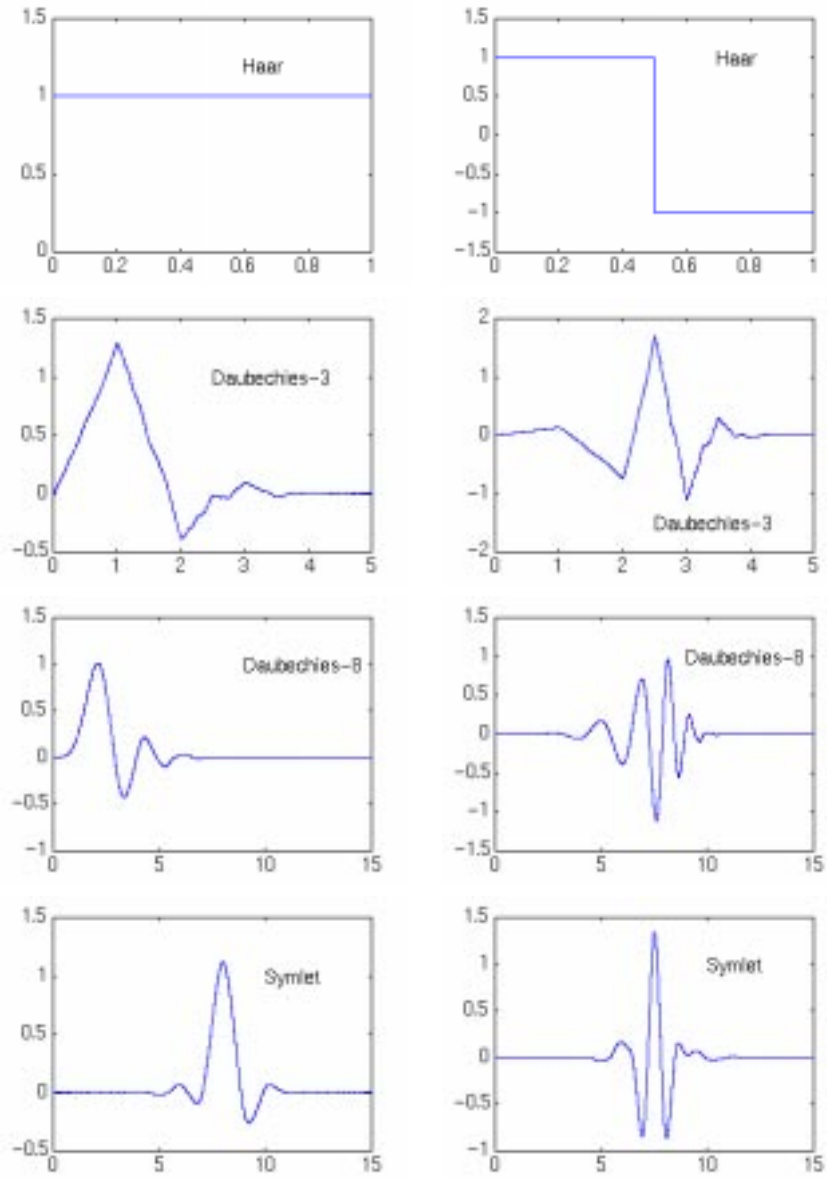


Figure 3.2: Plots of some analyzing wavelets. First row: father wavelets,  $\phi(x)$ . Second row: mother wavelets,  $\psi(x)$

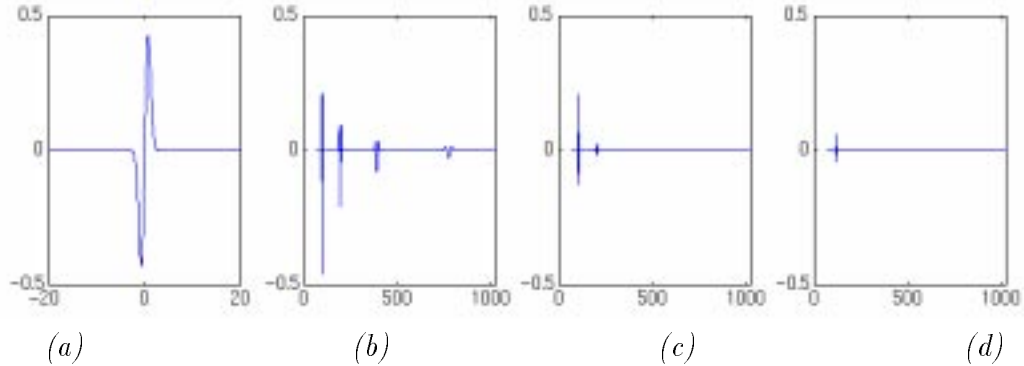
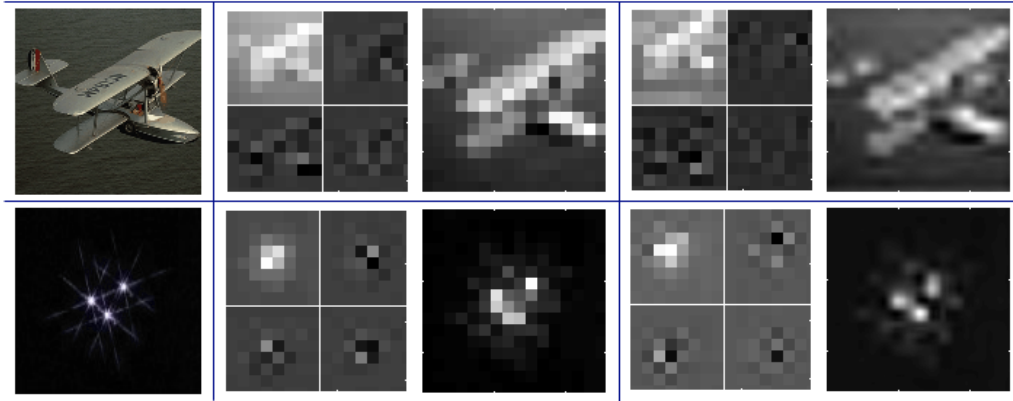


Figure 3.3: Comparison of Haar's wavelet and Daubechies wavelets on a 1-D signal. (a) original signal ( $x e^{-x^2}$ ) of length 1024 (b) coefficients in high-pass bands after a 4-layer Haar transform (c) coefficients in high-pass bands after a 4-layer Daubechies-3 transform (d) coefficients in high-pass bands after a 4-layer Daubechies-8 transform

- $\int_{-\infty}^{\infty} \psi_r(x) dx = \dots = \int_{-\infty}^{\infty} x^r \psi_r(x) dx = 0.$

Daubechies' wavelets provide excellent results in image processing due to the above properties. A wavelet function with compact support can be easily implemented by finite length filters. Moreover, the compact support enables spatial domain localization. Because the wavelet basis functions have continuous derivatives, they decompose a continuous function more efficiently with edge artifacts avoided. Since the mother wavelets are used to characterize details in a signal, they should have a zero integral so that the trend information is stored in the coefficients obtained by the father wavelet. A Daubechies' wavelet representation of a function is a linear combination of the wavelet basis functions.

Daubechies' wavelets are usually implemented numerically by quadratic mirror filters [84, 6, 22]. Multiresolution analysis of the trend and fluctuation of a function is implemented by convolving it with a low-pass filter and a high-pass filter that are versions of the same wavelet. The Haar wavelet transform is a special case of Daubechies' wavelet transform with  $r = 2$ , which is termed as Daubechies 2 wavelet transform. Eq.( 5.3) and Eq.( 5.4) provide the transform of signal  $x(n)$ ,  $n \in \mathbb{Z}$  by the Haar's wavelet. The corresponding low-pass filter is  $\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\}$ ; and the high-pass filter is  $\{\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\}$ . In fact, average color block layout image indexing is equivalent to using transform coefficients obtained by the low-pass filter of the Haar's wavelet.



*original image*    *saved Haar's coeff. (16 × 16)*    *saved Daubechies' coeff. (16 × 16)*  
 $128 \times 128$     *and its image reconstruction*    *and its image reconstruction*

Figure 3.4: Comparison of Haar's wavelet and Daubechies-8 wavelet.

Daubechies' wavelets transforms with  $r > 2$  are more like weighted averaging which better preserves the trend information in signals if we consider only the low-pass filter part. Various experiments and studies [109] have shown that in many cases Daubechies' wavelets with  $r > 2$  result in better performance than the Haar's wavelet.

Figures 3.3 and 3.4 show comparisons of the Haar wavelet, equivalent to average color blocks, and the Daubechies' 8 wavelet. In Figure 3.3, we notice that the signal with a sharp spike is better analyzed by Daubechies' wavelets because much less energy or trend is stored in the high-pass bands. Daubechies' wavelets are better suited for natural signals or images than the Haar wavelet. In layout image indexing, we want to represent as much energy in the image as possible in the low-pass band coefficients, which are used as features. When using the Haar wavelet, we lose more trend information in the discarded high-pass bands. Figure 3.4 shows the reconstruction of two images based only on the feature vectors of traditional layout indexing (same as Haar) and those using Daubechies' wavelets. Later in Chapter 5, we use Daubechies' wavelets in our experimental WBIIS system. Clearly, images reconstructed by low-pass band Daubechies' coefficients are closer to the original images than those by the Haar's coefficients. Here, we use image reconstruction to compare information loss or

encoding efficiency between Haar's and Daubechies' wavelets in the course of truncating discrete wavelet representations. Although these two examples in themselves do not imply for sure that a searching scheme using Daubechies' wavelets is better than one using Haar's wavelet, they may provide insights on how the schemes function.

In general, Daubechies' wavelets with long-length filters gives better energy concentration than those with short-length filters. However, it is not feasible to process discrete images using long-length wavelet filters due to the border problems associated with long-length filters. We use Daubechies-4 and Daubechies-8 as compromises.

### 3.4 Applications of wavelets

Because the original signal can be represented by coefficients in a linear combination of the wavelet basis functions, similar to Fourier analysis, data operations can be performed on the wavelet coefficients. Image data can be sparsely represented if we discard coefficients below a threshold value.

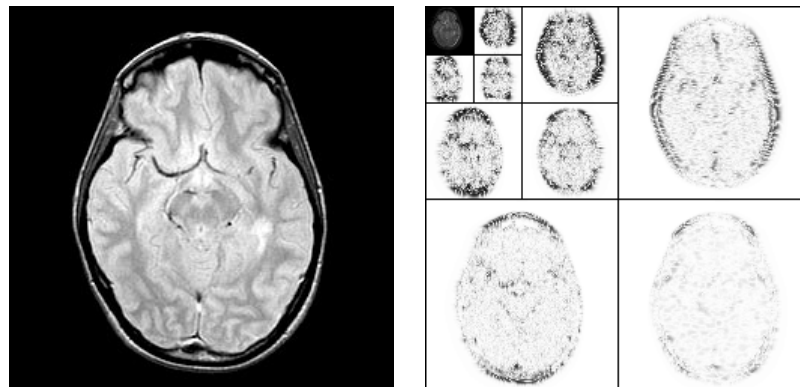


Figure 3.5: A 3-level wavelet transform of an MRI image slice using Daubechies' wavelet.

The wavelet transform offers good time and frequency localization. Information stored in an image is decomposed into averages and differences of nearby pixels. For smooth areas, the difference elements are near zero. The wavelet approach is therefore a powerful tool for data compression, especially for functions with long-range slow variations and short-range sharp variations [116]. The time and frequency localization

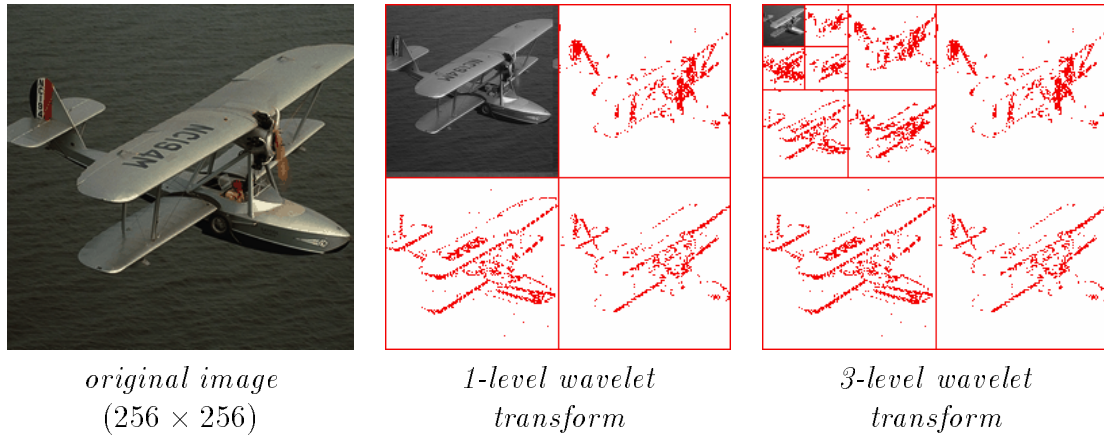


Figure 3.6: Multi-scale structure in the wavelet transform of an image. Dots indicate non-zero wavelet coefficients after thresholding. Daubechies-8 wavelet is used for this transform.

of the basis functions are adjusted by both scale index  $j$  and position index  $k$ . We may decompose the image even further by applying a wavelet transform several times recursively.

Figures 3.5 and 3.6 show the multi-scale structure in the wavelet transform of an image. In Figure 3.6, an original image of  $256 \times 256$  pixels is decomposed into a 1-level wavelet transform of  $256 \times 256$  coefficients. These coefficients are organized into 4 bands, emphasizing low frequency trend information, vertical-directional fluctuations, horizontal-directional fluctuations, and diagonal-directional fluctuations, respectively. The low frequency band of coefficients can be further decomposed to form higher-level wavelet transforms. The figure shows wavelet transforms after thresholding near zero. Most of the high frequency coefficients are of near-zero values. Note that information about the airplane's shape, color, and surface texture are well preserved and organized in different scales for analysis.

Since wavelet transforms decompose images into several resolutions, the coefficients, in their own right, form a successive approximation of the original images. For this reason, wavelet transforms are naturally suited for progressive image compression algorithms. Many current progressive compression algorithms apply quantization on coefficients of wavelet transforms [100, 98], which became more widely used after

Shapiro's [100] invention of the zero-tree structure, a method to group wavelet coefficients across different scales to take advantage of the hidden correlation among coefficients. Much subsequent research has taken place based on the zero-tree idea, including a very significant improvement made by Said and Pearlman [98], referred to as the S & P algorithm. This algorithm was applied to a large database of mammograms [1, 90] and was shown to be highly efficient even by real clinical-quality evaluations. An important advantage of the S & P algorithm and many other progressive compression algorithms is the low encoding and decoding complexity. No training is needed since trivial scalar quantization of the coefficients is applied in the compression process. However, by trading off complexity, the S & P algorithm was improved by tuning the zero-tree structure to specific data [29].

Other than compression, wavelet transforms have been applied to almost all research areas in signal processing [79, 80] and image processing [3, 27]. In speech recognition, fingerprint recognition, and denoising, techniques based on wavelet transforms represent the best available solutions.

In addition, wavelet transforms have been actively used in solving ordinary and partial differential equations, numerical analysis [6], statistics [25, 26], econometrics [92, 93, 61], fractals [36], communication theory [74], computer graphics [99, 43], and physics [35, 64]. A comprehensive list of related articles and books is maintained by MathSoft Inc. and is provided on-line [137].

### 3.5 Summary

In this chapter, we briefly reviewed an important mathematical tool for signal and image processing — the wavelet transform. Compared to other tools such as the Fourier transform, the wavelet transform provides much better spatial domain localization, an important property for signal processing. We compared several transforms and their properties to gain insights into choosing transforms for the image retrieval problem.

# Chapter 4

## Statistical Clustering and Classification

*All knowledge is, in the final analysis, history.  
All sciences are, in the abstract, mathematics.  
All judgements are, in their rationale, statistics.*

— C. Radhakrishna Rao (1920- )

### 4.1 Introduction

In modern CBIR systems, statistical clustering and classification methods are often used to extract visual features, index the feature space, and classify images into semantic categories. In our work, we apply statistical clustering to the block-wise feature space to extract region features. For very large databases, we use statistical clustering methods to index the high-dimensional feature space. Our semantic classification process is a statistical classification process.

Machine learning algorithms search for and generalize concepts within domain-specific search spaces. Clustering and classification are both important machine



learning methods. Clustering is an example of unsupervised learning, while classification is an example of supervised learning. In this chapter, we briefly review a number of commonly used statistical clustering and classification methods.

In Section 4.2, we discuss the history of, and concepts in, artificial intelligence and its subfield of machine learning. We then discuss two statistical clustering methods we used in our work: the k-means algorithm and the TSVQ algorithm (Section 4.3). Lastly, we review a statistical classification method we used in our work, the CART algorithm, in Section 4.4.

## 4.2 Artificial intelligence and machine learning

Herbert Simon [105] defined *learning* as “any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population”. The improvement of expertise is obtained through acquisition of knowledge. Most living creatures must learn in order to survive. The ability to learn should also be a fundamental capability of artificially intelligent systems.

Machine learning algorithms search and generalize concepts within domain-specific search spaces. New domain-specific concepts can be accepted by generalizing the concepts. Depending on whether the class identities of objects used in training are provided, a learning method is categorized as supervised learning, unsupervised learning, or a hybrid learning algorithm.

In 1943, Warren McCulloch and Walter Pitts developed a model of artificial neurons [82], one of the earliest works in machine learning. They demonstrated that any computable function could be computed by some network of connected neurons and argued that suitably structured networks could learn.

In 1950s, John McCarthy wrote a high level programming language called LISP, now the most dominant AI programming language. While LISP is not specifically targeted to learning, it has been used in almost all early learning systems. Allen Newell and Herbert Simon developed “General Problem Solver (GPS)”, which was

the first known program built to imitate human problem-solving protocols [87]. GPS did not have learning capabilities.

Researchers quickly realized that it was necessary to use more domain knowledge for complicated and larger reasoning tasks. The DENDRAL program developed by Bruce Buchanan in 1969 and the MYCIN program [103] developed by Edward H. Shortliffe in mid-1970's were based on knowledge bases. MYCIN is known as the first program that addressed the problem of reasoning with uncertain or incomplete information.

Many of these knowledge-based systems involve manually entered domain knowledges and supervised learning. With the available large amount of data and significantly improved computing power of the modern computers, statistical machine learning has become a popular research trend. We review recent work in statistical clustering and classification in the following sections.

### 4.3 Statistical clustering

A learner is classified as unsupervised if the learning process is given a set of examples that are not labeled as to class. Statistical clustering algorithms are unsupervised learning methods. Other unsupervised learning methods include conceptual clustering, Kohonen net clustering, theory-driven discovery, and data-driven discovery.

In statistical clustering, objects are described by numerical feature vectors. Class identities of the objects used in training are not provided. Depending on how a clustering algorithm adjusts to an added object in training, it can be non-incremental or incremental.

For any statistical clustering algorithm, a similarity or distance metric between feature vectors of objects is needed. Examples include the Euclidean distance and the Manhattan distance:

$$\text{Euclidean distance} = \left( \sum_i (x_i - y_i)^2 \right)^{\frac{1}{2}} \quad (4.1)$$

$$\text{Manhattan distance} = \sum_i |x_i - y_i| \quad (4.2)$$

where  $x_i$  and  $y_i$  are the elements of two feature vectors.

The distance between a feature vector and a cluster may be computed as the distance from the feature vector to the nearest or furthest point within the cluster, or to the centroid of the cluster, the latter being used much more often. The distance between two clusters can be computed likewise.

In this section, we focus on two statistical clustering algorithms, the k-means algorithm and the Tree-Structured Vector Quantization (TSVQ) algorithm. These algorithms were used in the dissertation research described in the following chapters.

### 4.3.1 The k-means algorithm

The  $k$ -means algorithm is a well-known statistical clustering algorithm [50]. To briefly introduce the idea, suppose observations (e.g., feature vectors in CBIR) are  $\{x_i : i = 1, \dots, L\}$ . The goal of the  $k$ -means algorithm is to partition the observations into  $k$  groups with means  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_k$  such that

$$D(k) = \sum_{i=1}^L \min_{1 \leq j \leq k} (x_i - \hat{x}_j)^2 \quad (4.3)$$

is minimized. That is, the average distance between a feature vector and the cluster with the nearest centroid to it is minimized. This “average distance” is also referred to as “the average class variance” since they are equivalent if the Euclidean distance is used. Two necessary conditions for the  $k$  clusters are:

1. Each feature vector is partitioned into the cluster with the nearest centroid to it.

2. The centroid of a cluster is the vector minimizing the average distance from it to any feature vector in the cluster. In the special case of the Euclidean distance, the centroid should be the mean vector of all the feature vectors in the cluster.

A simple k-means algorithm consists of the following steps:

1. Initialization: choose the initial  $k$  cluster centroids.
2. Loop until the stopping criterion is met:
  - (a) For each feature vector in the data set, assign it to a class such that the distance from this feature to the centroid of that cluster is minimized.
  - (b) For each cluster, recalculate its centroid as the mean of all the feature vectors partitioned to it.

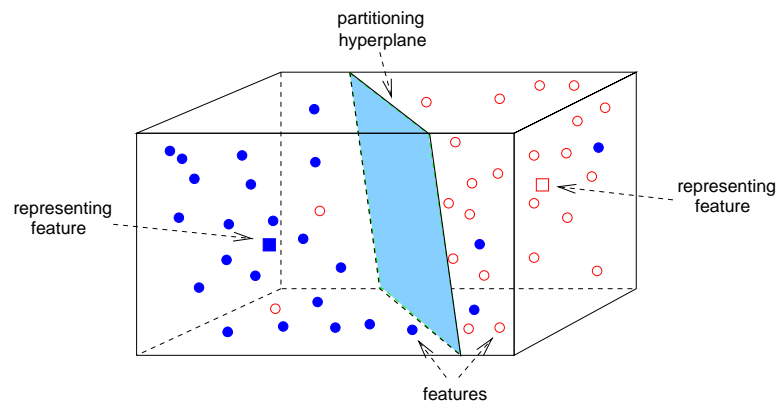


Figure 4.1: The k-means algorithm partitions the feature space using hyper-planes.

If the Euclidean distance is used, the k-means algorithm results in hyper-planes as cluster boundaries. That is, for the feature space  $\mathbb{R}^n$ , the cluster boundaries are hyper-planes in the  $n - 1$  dimensional space  $\mathbb{R}^{n-1}$ . Figure 4.1 shows an example in the three-dimensional Euclidean feature space. Each partition of the space is a two dimensional hyperplane.

The initialization process is critical to the results. There are a number of different ways to initialize the k-means algorithms:

1. Assign classes to features randomly
2. Choose centroids as random samples from an uniform distribution on the feature space.
3. Randomly select feature vectors in the data set as centroids.

The k-means algorithm terminates when no more feature vectors are changing classes. It can be proved that the k-means algorithm is guaranteed to terminate, based on the fact that both steps of k-means (i.e., assigning vectors to nearest centroids and computing cluster centroids) reduce the average class variance. In practice, running to completion may require a large number of iterations. The cost for each iteration is  $O(kn)$ , for the data size  $n$ . Typically stopping criteria are:

1. Stop after a fix number of iterations
2. Stop after the average class variance is smaller than a threshold
3. Stop after the reduction of the class variance is smaller than a threshold

### 4.3.2 The TSVQ algorithm

Vector Quantization (VQ) [39] is the process of mapping vectors (usually continuous, real valued vectors), from a subset  $\mathcal{A}$  of  $n$ -dimensional Euclidean space  $\mathbb{R}^n$  onto a finite set of vectors in  $\mathbb{R}^n$ , denoted by  $\mathcal{C} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N\}$ . Although techniques of VQ are applied primarily to data compression, they are naturally suitable to classification. If for each  $\hat{x}_i$  in  $\mathcal{C}$ , a class  $c_i$  is assigned to it and a vector  $x \in \mathcal{A}$  mapped to  $\hat{x}_i$  is identified as class  $c_i$ , the vector quantizer is also a classifier. The goal of a vector quantizer is to minimize the average distance between a feature vector and a vector in  $\mathcal{C}$  that is closest to this feature vector. It is clear that this goal is the same as that of the k-means algorithm. In fact, the same algorithm as the k-means was developed in the signal processing community and was referred to as the Lloyd vector quantization (LVQ) [76] algorithm.

In image retrieval, similar images are retrieved by finding images with feature vectors close in certain distance to that of the query image. Since a vector quantizer

divides feature vectors into cells so that vectors clustered together are grouped into one cell, it can be used to structure an image database so that similar images are grouped together. Given any query image, we first decide which cell its feature vector belongs to, and then search for similar images within this cell. Compared with algorithms that search the whole database, this restricted searching method reduces complexity significantly. One may point out that by constraining the search to the cell containing the query feature vector, we are not guaranteed to find the images with the smallest distances to the query in the entire database. However, this is not necessarily a disadvantage in view of data clustering. The underlying assumption for retrieval, that images with feature vectors closer to each other are more similar, may not be absolutely correct. If similar images form clusters and the clusters are identified, constraining search within the clusters actually prevents unrelated images from being retrieved. The statistical aspects of data clustering are discussed in [58]. We have applied the TSVQ algorithm not only to large-scale image databases, but also to large-scale DNA sequence databases [40].

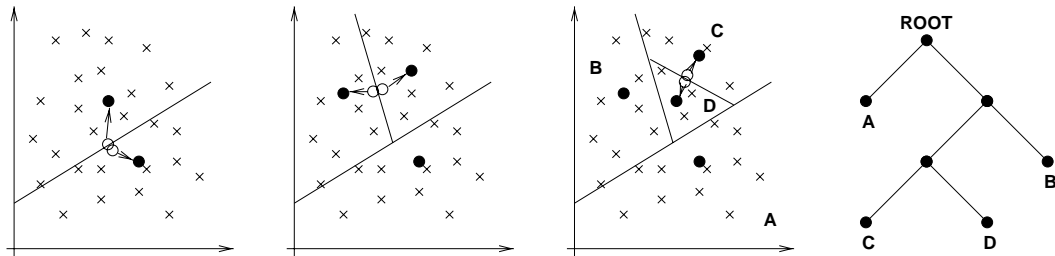


Figure 4.2: An example of tree structured partition of feature space. 'x' indicates an individual feature. '.' indicates the centroid of a cluster cell.

As vector quantization algorithms are used in communication systems, computational complexity is an important issue, especially for real-time systems. In communication systems, a vector quantizer consists of two parts: the *encoder* and the *decoder*. The encoder maps  $x \in \mathcal{A}$  to an index  $i$ , which is transmitted by a finite number of bits to a receiver. The decoder at the receiving end maps the index  $i$  into the representative vector  $\hat{x}_i$ . The LVQ algorithm is highly asymmetric in terms of computation needed by the encoder and the decoder. The decoder is very fast regardless of the vector dimension and the number of representative vectors  $N$  because it can

be implemented by a simple table lookup. The amount of computation in encoding, however, increases linearly with  $N$ . Usually, since the bit rate per dimension  $\frac{\log N}{n}$  is fixed, the complexity of encoding grows exponentially with the dimension  $n$ . Many algorithms have been developed to reduce the encoding complexity [39]. A widely used one is the tree-structured vector quantization (TSVQ) algorithm [94, 95, 39] (Figure 4.2). As the TSVQ algorithm also finds cluster centroids significantly faster than the Lloyd VQ or k-means algorithm, applying TSVQ to image retrieval speeds up searching as well as indexing, which is crucial if real-time indexing of an image database is demanded.

The TSVQ algorithm progressively partitions the feature vector space into cells. As suggested by the name of the algorithm, the partition is characterized by a tree structure. In the case of image retrieval, feature vectors of all the images in a database are used as training data to design the tree. Initially, all the feature vectors belong to one cell, i.e., the entire space, marked as the root node of the tree. The root node is split into two child nodes, each representing one cluster, by the LVQ algorithm. The two child nodes divide the training feature vectors into two subsets. According to LVQ, a feature vector is grouped to the subset whose centroid is closer to the vector. LVQ is then applied to each subset represented by a child node to further divide the feature space. The tree grows by splitting leaf nodes recursively. The algorithm is greedy in that at every step, the node that provides the largest “goodness” if it is divided is chosen to be split [95]. This strategy is optimal at every step, but not globally optimal in general. A node becomes terminal if the number of feature vectors in the node is smaller than a threshold. The procedure of generating a tree in two-dimensional space is shown in Figure 4.2. The complexity for TSVQ is  $O(n \log(k))$ , for data size  $n$  and the number of clusters  $k$ .

## 4.4 Statistical classification

A learner is regarded as supervised if the learning process is given a set of examples, each with the class to be returned for a given input. Classification algorithms are

supervised learning methods. In this section, we introduce the Classification and Regression Trees ( $\text{CART}^R$ ) algorithm. A large variety of classification algorithms have been developed, such as the kernel method, the Gaussian mixture model method, the  $k$ -nearest neighbor ( $k$ -NN), and the learning vector quantization (LVQ) algorithm.

#### 4.4.1 The CART algorithm

CART [8] is a non-parametric statistical classification algorithm. The output can be either categorical (discriminant analysis or classification), or continuous (non-parametric regression). We restrict our interest to classification in the sequel. CART is powerful in that it is capable of dealing with incomplete data, multiple types of input and output features, and the classification trees it produces contain rules easy to interpret.

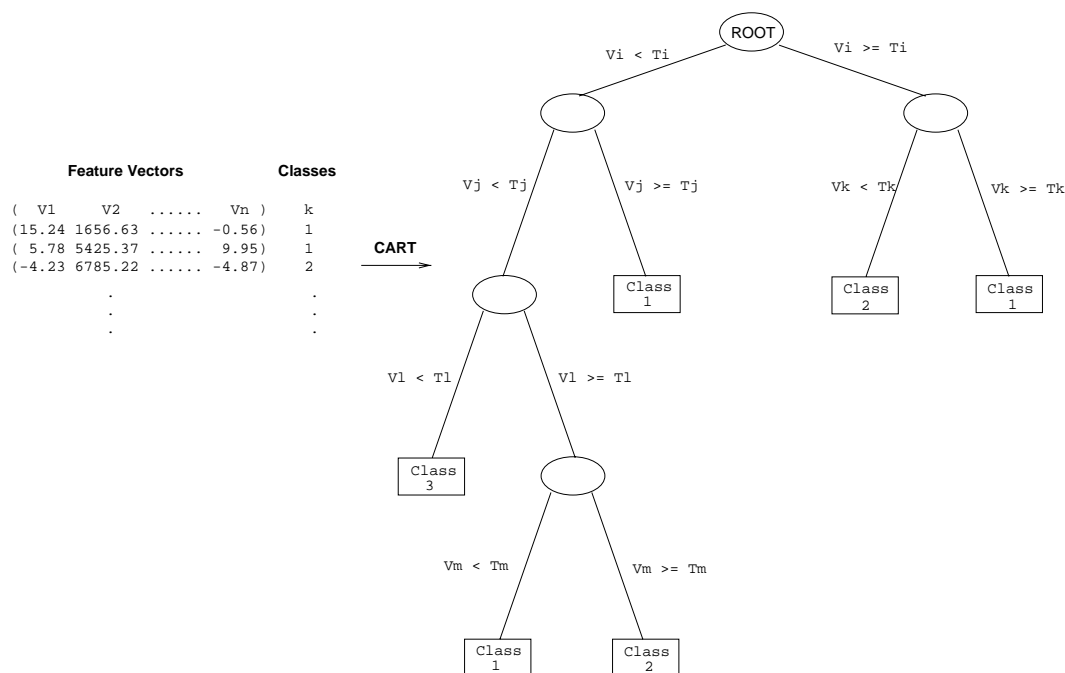


Figure 4.3: Generating a classification tree using the CART algorithm.

Given a set of training data containing both feature vectors and class identities, CART produces a binary decision tree (Figure 4.3) with each decision or split characterized by a condition  $x_i > \alpha?$ , where  $x_i$  is one component of the feature vector and



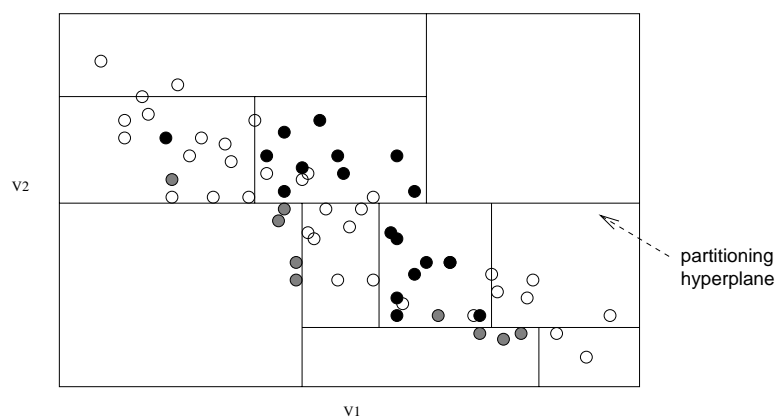


Figure 4.4: CART partitions the 2-D feature space into cells using straight lines parallel to the coordinate axes.

$\alpha$  is a threshold. The binary decision tree ultimately partitions the feature vector space into cells, each corresponding to a leaf on the tree. Any feature vector in a cell is identified as a unique class selected by the majority vote based upon all the feature vectors in the training data set that are partitioned to the cell. A byproduct of the decision tree is the empirical probability mass function of classes in each cell.

The binary decision tree is formed by splitting nodes recursively such that the “impurity” of the tree is minimized. The procedure is recursive in that it starts with splitting the feature space into two cells, or nodes, and then applies the same splitting process to each newly generated node, which represents a subset of the feature space. Definitions of the “impurity” indicators will be discussed shortly. There are a number of stopping criteria for growing the decision tree. One example is a minimum number of feature vectors in a node. CART is a suboptimal greedy algorithm. It is computationally too expensive to search for an optimal set of splitting criteria.

As illustrated in Figure 4.4, the CART algorithm results in hyper-planes parallel to coordinate planes as partition boundaries in the Euclidean space. That is, for the feature space  $\mathbb{R}^n$ , the cluster boundaries are defined by hyper-planes  $x_i = \alpha$  in the  $n - 1$  dimensional space  $\mathbb{R}^{n-1}$ .

There are a few commonly used “impurity” indicators. One is the entropy. The

entropy of a node is

$$\sum_{allclasses} -\hat{p}(c) \log \hat{p}(c), \quad (4.4)$$

where  $\hat{p}(c)$  is the empirical distribution of classes for feature vectors in the node. The “impurity” of a tree is defined as a weighted summation of the entropy values of all the leaves with the weight being the percentage of training vectors that belong to each leaf. Another popular “impurity” indicator, which yields better results in many applications compared with entropy is the Gini index. The Gini index of each node is

$$\sum_{allclasses} -\hat{p}(c)(1 - \hat{p}(c)). \quad (4.5)$$

Other “impurity” indicators include the mean Euclidean distance between all vectors of parameters in the sample set.

The splitting criteria are set for each member of the feature. For example, one criterion may be whether the  $i$ -th member of the feature vector has value greater than  $x$ . Usually the range of each member of the feature is linearly split into predefined mutually exclusive intervals. This is not optimal but offers a reasonable compromise between accuracy and the computational time.

Once the initial full tree is grown, i.e., the stopping criterion is met, CART employs a *backward pruning* process to avoid shortsightedness in splitting and over-fitting to the training data. Cross-validation is used to select an intermediate classification tree.

To penalize over-fitting to the training data caused by a large tree (too many splits), the cost of a tree  $T$  is defined by

$$C_\alpha(T) = \sum D_g(T) + \alpha|T|, \quad (4.6)$$

where the sum is over all the leaves, or terminal nodes of  $T$ ,  $|T|$  is the number of terminal nodes in  $T$ , and  $\alpha$  is a cost-complexity parameter. A subtree of  $T$  is

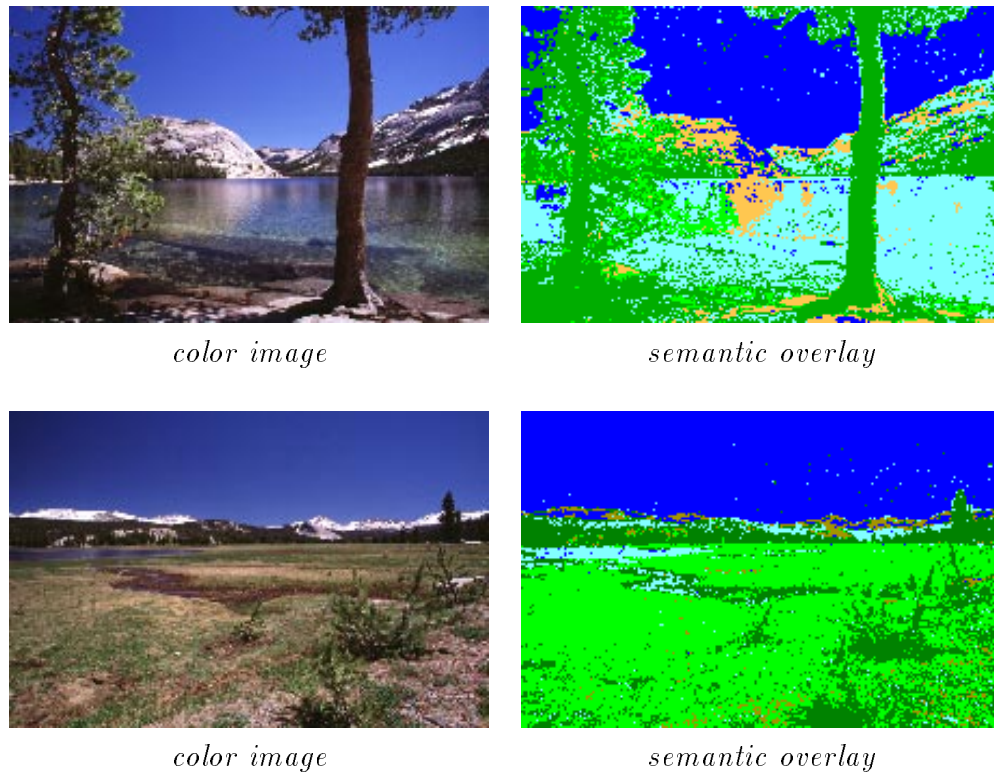


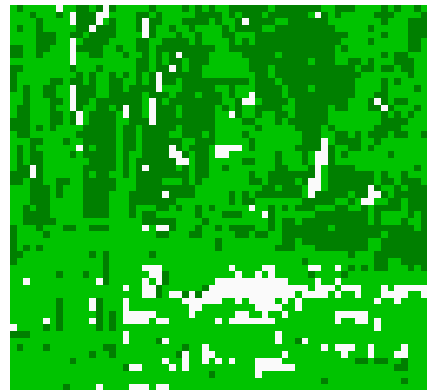
Figure 4.5: Semantic analysis of outdoor scenes using the classification and regression trees (CART) algorithm. No post-processing is performed. Color scheme: Deep blue (darkest) for sky, yellow (very light gray) for stone, light blue (lightest gray) for river/lake, light green (light gray) for grass, deep green (dark gray) for tree/forest. (Wang and Fischler [128])

obtained by pruning off branches in the tree. For each  $\alpha$ , the best subtree  $T_\alpha$  is found via pruning. The optimal  $\alpha$ ,  $\hat{\alpha}$ , is estimated via cross-validation. The corresponding subtree  $T_{\hat{\alpha}}$  is the final result. The complexity of tree generation is  $O(n)$ , for data size  $n$ .

One major advantage of CART is that trees generated are easy to understand and interpret. In addition, the computational complexity of applying decision trees to classification is extremely low. In fact, the complexity depends on the distribution of the training data. For a fixed distribution, the complexity of classification is  $O(1)$ .



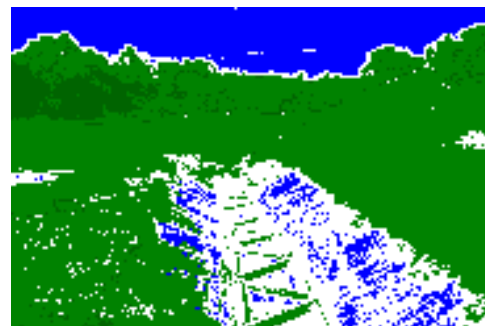
*gray-scale image*



*semantic overlay*



*gray-scale image*



*semantic overlay*

Figure 4.6: Semantic analysis of outdoor scenes using the classification and regression trees (CART) algorithm. No post-processing is performed. Color scheme: Deep blue (darkest) for sky, light blue for river/lake, light green (light gray) for grass, deep green (dark gray) for tree/forest, white for non-classified regions. (Wang and Fischler [128])

A drawback is that the classification boundaries are not smooth. To approximate a circle cluster in a 2-D feature space, CART creates a zig-zag boundary, the closeness

to the circle being determined by the number of training vectors available.

Wang and Fischler used CART in the problem of creating a semantic “overlay” for natural outdoor scenes [128]. A sequence of only seven training images are used to represent *sky*, *stone*, *river/lake*, *grass* and *tree/forest*. The mean colors and variances of  $4 \times 4$  blocks in RGB color space are the components of the training feature vectors. These features are simple but have proven capable of distinguishing the above five classes. For gray-scale images, only the mean intensities and variances of  $4 \times 4$  blocks are used as the components of the training feature vectors.

It takes about one minute on a Pentium III PC to create the classification tree structure. After the classification tree is created, it takes only a few seconds to classify a given image to create the semantic overlay for a color image of  $768 \times 512$  pixels. Figures 4.5 and 4.6 show the classification results on color and gray-scale images. Each of the five different classes is given a unique “pseudo” color in the final result.

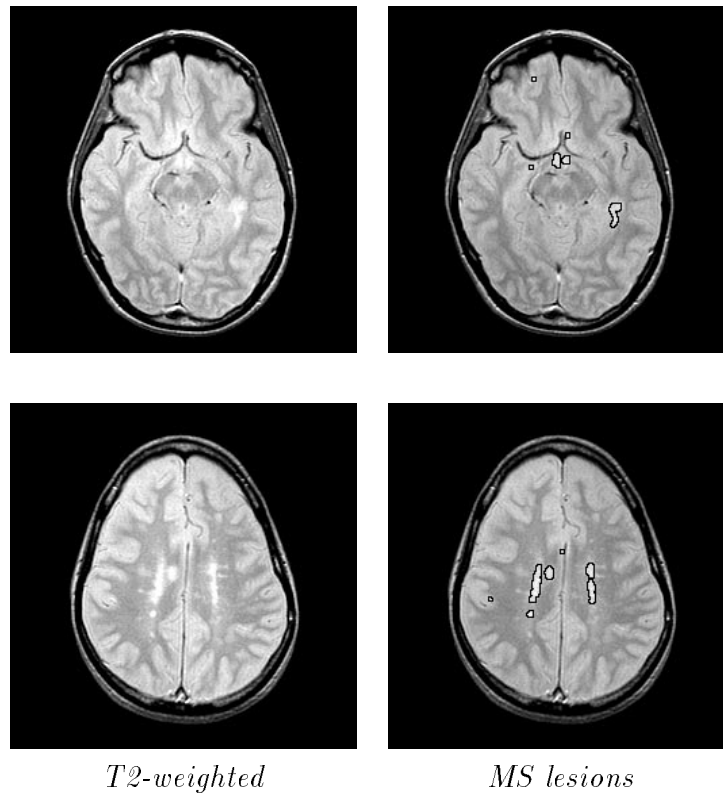


Figure 4.7: Classification of MRI images using CART. No pre- or post-processing.

CART can be used to create a classification “overlay” for biomedical images. We applied a similar algorithm on multi-spectrum MRI images. The pixels in the images are classified fully automatically as regions of white matter, gray matter, CSF, and multiple sclerosis (MS) lesions. As shown in Figure 4.7, pixels of MS lesion regions can be found. The algorithm was developed for indexing clinical trial image databases. CART has been used in many other biomedical applications.

## 4.5 Summary

Statistical clustering and classification are both important machine learning methods that are used in CBIR to extract visual features, index the feature space, and classify images into semantic categories. Clustering is an example of unsupervised learning, while classification is an example of supervised learning. In this chapter, we reviewed three important statistical clustering and classification methods, including the k-means algorithm, the TSVQ algorithm, and the CART algorithm. The cost for each iteration in the k-means algorithm is  $O(kn)$ , for the data size  $n$ . The complexity for the TSVQ algorithm is  $O(n \log(k))$ , for generating  $k$  clusters. In CART, the complexity of tree generation is  $O(n)$ .

# Chapter 5

## Wavelet-Based Image Indexing and Searching

*A journey of a thousand miles must begin with a single step.*

— Lao-Tzu (~ 570-490 B.C.)

### 5.1 Introduction

This chapter describes WBIIS (Wavelet-Based Image Indexing and Searching), an image indexing and retrieval algorithm with partial-sketch image-searching capability for large image databases. We developed WBIIS in 1996 as a first step in image retrieval using wavelets [122]. The SIMPLIcity system (Chapter 6), another wavelet-based image indexing and searching system, has been developed to address the limitations of the WBIIS system.

The WBIIS algorithm characterizes the color variations over the spatial extent of the image in a manner that often provides semantically-meaningful image comparisons. The indexing algorithm applies a Daubechies' wavelet transform for each of the three opponent color components. The wavelet coefficients in the lowest few frequency

bands, and their variances, are stored as feature vectors. To speed up retrieval, a two-step procedure is used that first does a crude selection based on the variances, and then refines the search by performing a feature-vector match between the selected images and the query. For better accuracy in searching, two-level multiresolution matching is used. Masks are used for partial-sketch queries. This technique performs much better in capturing coherence of image, object granularity, local color/texture, and bias avoidance than earlier color layout algorithms [88, 46]. WBIIS is much faster and more accurate than earlier algorithms. When tested on a database of more than 10,000 general-purpose COREL photograph images, the best 100 matches were found in two seconds.

In Section 5.2, we discuss details of the preprocessing step of WBIIS. An overview of multiresolution indexing is given in 5.3. The indexing and retrieval processes are discussed in Section 5.4 and Section 5.5, respectively. Some experimental results are shown in Section 5.6. WBIIS has limitations (Section 5.7) due to its close relationship with color layout indexing. These limitations are addressed in the recently developed SIMPLIcity system, which is described in Chapter 6.

## 5.2 Preprocessing

As discussed in Chapter 2, many color spaces are in use to represent images. As a result, many image formats are currently in use, e.g., GIF, JPEG, PPM and TIFF are the most widely used formats for picture libraries. In the medical domain, DICOM is becoming a dominant image format for radiology images. Because images in an image database can have different formats and different sizes, we must first normalize the data so that comparisons among images is possible.

### 5.2.1 Scale normalization

For the test database of images, a rescaled thumbnail consisting of  $128 \times 128$  pixels in Red-Green-Blue (i.e., RGB) color space is adequate for the purpose of computing the feature vectors. For medical images, we retain the original resolution to keep the



details. Typically, medical images of the same modality share the same resolution. For example, most Computed Tomography (CT) images are of  $512 \times 512$  pixels.

Bilinear interpolation is used for the rescaling process. This method resamples the input image by overlaying on the input image a grid with  $128 \times 128$  points. This gives one grid point for each pixel in the output image. The input image is then sampled at each grid point to determine the pixel colors of the output image. When grid points lie between input pixel centers, the color values of the grid point are determined by linearly interpolating between adjacent pixel colors (both vertically and horizontally).

This rescaling process is more effective than a Haar-like rescaling, i.e., averaging several pixels to obtain a single pixel to decrease image size, and replicating pixels to increase image size, especially when the image to be rescaled has frequent sharp changes such as local texture. It is necessary to point out, however, that the rescaling process is in general not important for the indexing phase when the size of the images in the database is close to the size to be rescaled. The sole purpose for the rescaling is to make it possible to use the wavelet transforms and to normalize the feature vectors. Here, we assume the images in the database to have sizes close to  $128 \times 128$ . In fact, images may be rescaled to any other size as long as each side length is a power of two. Therefore, to obtain a better performance for a database of mostly very large images, we would suggest using a bilinear interpolation to rescale to a large common size, with side lengths being powers of two, and then apply more levels of Daubechies' wavelets in the indexing phase.

### 5.2.2 Color space normalization

Since color distances in RGB color space do not reflect the actual human perceptual color distance, we convert and store the image in a component color space with intensity and perceived contrasts. We define the new values at a color pixel based on

the RGB values of an original pixel as follows:

$$\begin{cases} C_1 = (R + G + B)/3 \\ C_2 = (R + (max - B))/2 \\ C_3 = (R + 2 * (max - G) + B)/4 \end{cases} \quad (5.1)$$

Here *max* is the maximum possible value for each color component in the RGB color space. For a standard 24-bit color image, *max* = 255. Clearly, each color component in the new color space ranges from 0 to 255 as well. This color space is similar to the opponent color axes

$$\begin{cases} RG = R - 2 * G + B \\ BY = -R - G + 2 * B \\ WB = R + G + B \end{cases} \quad (5.2)$$

defined in [5] and [111].

Besides the perception correlation properties [54] of such an opponent color space, one important advantage of this alternative space is that the  $C_1$  axis, or the intensity, can be more coarsely sampled than the other two axes. This reduces the sensitivity of color matching to a difference in the global brightness of the image, and it reduces the number of bins and subsequent storage in the color histogram indexing.

## 5.3 Multiresolution indexing

In this section, we review related work in multiresolution indexing, including earlier color layout indexing algorithms and the work at the University of Washington using the Haar wavelet [57].

### 5.3.1 Color layout

Storing color layout information is another way to describe the contents of the image. It is especially useful when the query is a partial sketch rather than a full image. In simple color layout image indexing, an image is divided into equal-sized blocks.

The average color on the pixels in each block is computed and stored for subsequent image matching using Euclidean metric or variations of the Euclidean metric. It is also possible to form the features based on statistical analysis of the pixels in the block. Both techniques are very similar to image rescaling or subsampling. However, they do not perform well when the image contains high-frequency information such as sharp color changes. For example, if there are pixels of various colors ranging from black to white in one block, an effective result value for this block cannot be predicted using these techniques.

### 5.3.2 Indexing with the Haar wavelet

The system developed at the University of Washington [57] applies the Haar wavelet to multiresolution image querying. Forty to sixty of the largest magnitude coefficients are selected from the  $128^2 = 16,384$  coefficients in each of the three color channels. The coefficients are stored as  $+1$  or  $-1$  along with their locations in the transform matrix. As demonstrated in the cited paper, the algorithm performs much faster than earlier algorithms, with an accuracy comparable to earlier algorithms when the query is a hand sketch or a low-quality image scan.

One drawback of using Haar to decompose images into low frequency and high frequency is that the Haar transform cannot efficiently separate image signals into low-frequency and high-frequency bands. From the signal processing point of view, since the wavelet transform is essentially a convolution operation, performing a wavelet transform on an image is equivalent to passing the image through a low-pass filter and a high-pass filter [44]. The low-pass and high-pass filters corresponding to the Haar transform do not have a sharp transition and fast attenuation property. Thus, the low-pass filter and high-pass filter cannot separate the image into clean distinct low-frequency and high-frequency parts. On the other hand, the Daubechies wavelet transform with longer length filters [23] has better frequency properties. Because in our algorithm we rely on image low-frequency information to do comparison, we applied the Daubechies wavelet transform instead of the Haar transform.

Moreover, due to the normalization of functional space in the wavelet basis design,

the wavelet coefficients in the lower frequency bands, i.e., closer to the upper-left corner in a transform matrix, tend to be more dominant (are of larger magnitude) than those in the higher frequency bands. Coefficients obtained by sorting and truncating will most likely be in the lower frequency bands. For the Haar case,

$$F_0(x(n)) = \frac{1}{\sqrt{2}}(x(n) + x(n + 1)) \quad (5.3)$$

$$F_1(x(n)) = \frac{1}{\sqrt{2}}(x(n) - x(n + 1)) \quad (5.4)$$

coefficients in each band are expected to be  $\frac{2}{\sqrt{2}}$  times larger in magnitude than those in the next higher frequency band, i.e., those in one level previous to the current level. For a  $128 \times 128$  image, we expect the coefficients in the transform to have an added weight varying from 1 to 8 before the truncation process. As indicated in Eq. 5.3, the low-frequency band in a Haar wavelet transform is mathematically equivalent to the averaging color block or image rescaling approach in earlier layout algorithms mentioned above. Thus, the accuracy is not improved when the query image or the images in the database contain high-frequency color variation.

Although the University of Washington approach can achieve a much faster comparison by storing only 40 to 60 coefficients for each color channel as a feature vector, much useful information about the image is discarded. Thus, it is possible for two images having the same feature vector to differ completely in semantic content. In addition, two pictures with similar content but different locations of sharp edges may have feature vectors that are far apart in feature space. This is why the University of Washington algorithm has a sharp decrease in performance when the query image consists of a small translation of the target image.

### 5.3.3 Overview of WBIIS

We have developed a color layout indexing scheme using Daubechies' wavelet transforms that better represents image semantics, namely, object configuration and local color variation, both represented by Daubechies' wavelet coefficients. For large databases, feature vectors obtained from multi-level wavelet transforms are stored to

speed up the search. We apply a fast wavelet transform (FWT) with Daubechies' wavelet to each image in the database, for each of the three color components. Two-level low-frequency coefficients of the wavelet transform, and their standard deviations, are stored as feature vectors.

Given a query image, the search is carried out in two steps. In the first step, a crude selection based on the stored standard deviations is carried out. Images with similar semantics usually have similar standard deviation values. An image with almost the same color, i.e., with low standard deviation values, is unlikely to have the same semantics as an image with very high variation or high standard deviation values.

In the second step, a weighted version of the Euclidean distance between the feature coefficients of an image selected in the first step and those of the querying image is calculated, and the images with the smallest distances are selected and sorted as matching images to the query. We use two levels of wavelet coefficients to process the query in a multiresolution fashion. We will show below that this algorithm can be used to handle partial hand-drawn sketch queries by modifying the computed feature vector.

## 5.4 The indexing algorithm

The discrete wavelet transform (DWT) we described can be directly used in image indexing for color layout type queries. Our indexing algorithm is described below.

For each image to be inserted to the database, obtain  $128 \times 128$  square rescaled matrices in  $(C_1, C_2, C_3)$  components following Eq. 5.1 in Section 5.2. We then compute a 4-layer 2-D fast wavelet transform on each of the three matrices using Daubechies' wavelets. Denote the three matrices obtained from the transforms as  $W_{C_1}(1 : 128, 1 : 128)$ ,  $W_{C_2}(1 : 128, 1 : 128)$  and  $W_{C_3}(1 : 128, 1 : 128)$ <sup>1</sup>. Then the upper-left  $8 \times 8$  corner of each transform matrix,  $W_{C_i}(1 : 8, 1 : 8)$ , represents the lowest frequency band of the 2-D image in a particular color component for the level of wavelet transform we

---

<sup>1</sup>Here we use MATLAB [48] notation. That is,  $A(m_1 : n_1, m_2 : n_2)$  denotes the submatrix with opposite corners  $A(m_1, m_2)$  and  $A(n_1, n_2)$ .

used. The lower frequency bands in the wavelet transform usually represent object configurations in the images and the higher frequency bands represent texture and local color variation. The three  $8 \times 8$  submatrices (namely,  $W_{C_i}(1 : 8, 9 : 16)$ ,  $W_{C_i}(9 : 16, 1 : 8)$  and  $W_{C_i}(9 : 16, 9 : 16)$ ) closest to the  $8 \times 8$  corner submatrix  $W_{C_i}(1 : 8, 1 : 8)$  represent detailed information in the original image to some extent, though most of the fluctuation information is stored in the thrown-away higher frequency band coefficients. Extracting a submatrix  $W_{C_i}(1 : 16, 1 : 16)$  of size  $16 \times 16$  from that corner, we get a semantic-preserving compression of 64:1 over the thumbnail of  $128 \times 128$  pixels and a higher compression over the original image. We store this as part of the feature vector.

Then we compute the standard deviations, denoted as  $\sigma_{c_1}, \sigma_{c_2}, \sigma_{c_3}$ , of the  $8 \times 8$  corner submatrices  $W_{C_i}(1 : 8, 1 : 8)$ . Three such standard deviations are then stored as part of the feature vector as well. Figure 5.1 shows two images with the upper-left corner submatrices of their 2-D fast wavelet transforms in  $(C_1, C_2, C_3)$  color space. Notice that the standard deviation of the coefficients in the lowest frequency band obtained from the first image differs considerably from that obtained from the second image. Since the standard deviations are computed based on the wavelet coefficients in the lowest frequency band, we have eliminated disturbances arising from detailed information in the image.

We also obtain a 5-level 2-D fast wavelet transform using the same bases. We extract and store a submatrix of size  $8 \times 8$  from the upper-left corner. Thus, we have stored a feature index using the multiresolution capability of the wavelet transform.

Because the set of wavelets is an infinity set, different wavelets may give different performance for different types of image. One should take advantage of this characteristic in designing an image retrieval system. To match the characteristics of the signal we are analyzing, we used a Daubechies-8 or Symmlet-8 wavelet for the DWT process. Symmlets were designed by Daubechies [24] to be orthogonal, smooth, nearly symmetric, and non-zero on a relatively short interval (compact support). Wavelet subclasses are distinguished by the number of coefficients and by the level of iteration. Most often they can be classified by the number of vanishing moments. The number of vanishing moments is weakly linked to the number of oscillations of the

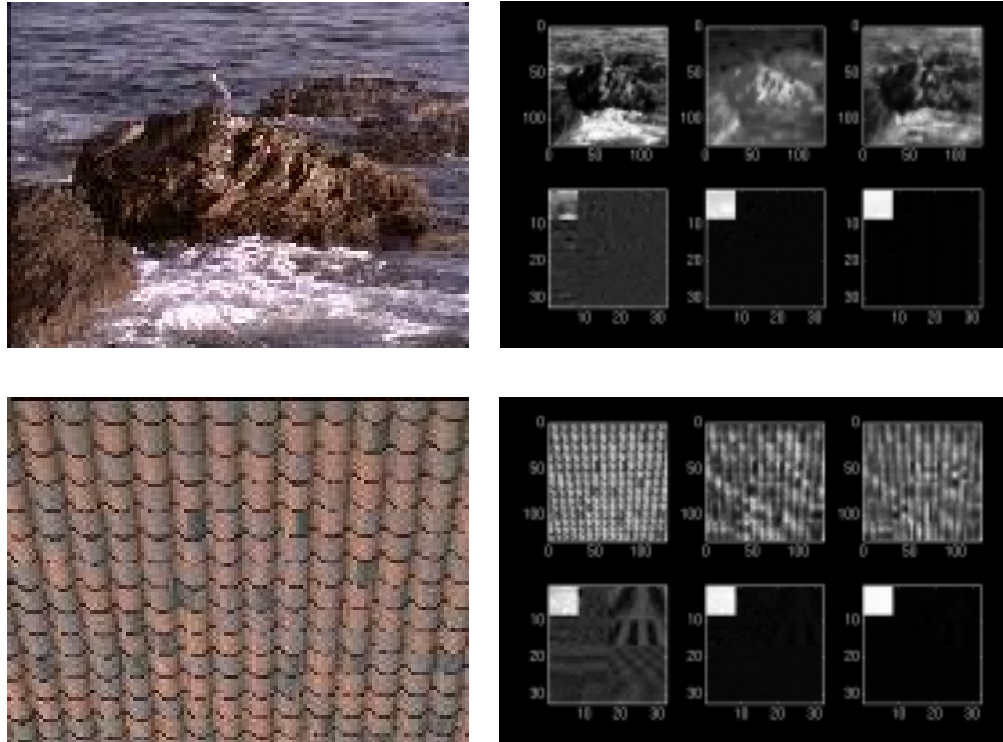


Figure 5.1: Two images with the upper-left corner submatrices of their fast wavelet transforms in  $(C_1, C_2, C_3)$  color space. We use the standard deviations of wavelet coefficients to distinguish images with very different object composition. The standard deviations we stored for the first image are  $\sigma_{C_1} = 215.93$ ,  $\sigma_{C_2} = 25.44$ , and  $\sigma_{C_3} = 6.65$  while means of the coefficients in the lowest frequency band are  $\mu_{C_1} = 1520.74$ ,  $\mu_{C_2} = 2124.79$ , and  $\mu_{C_3} = 2136.93$ . The standard deviations we stored for the second image are  $\sigma_{C_1} = 16.18$ ,  $\sigma_{C_2} = 10.97$ , and  $\sigma_{C_3} = 3.28$  while means of the coefficients in the lowest frequency band are  $\mu_{C_1} = 1723.99$ ,  $\mu_{C_2} = 2301.24$  and  $\mu_{C_3} = 2104.33$ .

wavelet, and determines what the wavelet does or does not represent. The number of vanishing moments for the subclass of our Symmlet wavelet is 8, which means that our wavelet will ignore linear through eighth degree functions.

Daubechies' wavelets perform better than earlier layout coding because the coefficients in wavelet-created compression data actually contain sufficient information to reconstruct the original image at a lower loss rate using an inverse wavelet transform. By using the low-frequency coefficients of Daubechies' wavelet transforms to index images, we retain the most important information in the image, the trend information. However, even with Daubechies' wavelet transforms, we drop a small amount of detailed information by not keeping high-frequency wavelet coefficients. Without region segmentation, it takes too much space to keep all the wavelet coefficients. In Chapter 6, we introduce our recent region-based SIMPLIcity system using both the trend information and the fluctuation information represented in the wavelet coefficients.

## 5.5 The matching algorithm

In WBIIS, the matching process for fully-specified queries is not the same as that for partial sketch queries. We discuss the details of each of the processes below.

### 5.5.1 Fully-specified query matching

When a user submits a query, we must compute the feature vector for the querying image and match it to the pre-computed feature vectors of the images in the database. This is done in two phases.

In the first phase, we compare the standard deviations stored for the querying image with the standard deviations stored for each image in the database.

Figure 5.2 demonstrates the histograms of the standard deviations we computed for general-purpose photograph images. Studying the three histograms, we found that the standard deviations of the intensity component (the  $C_1$  component) are a lot more diverse than those of the other two (the  $C_2$  and  $C_3$  components). In fact, the values



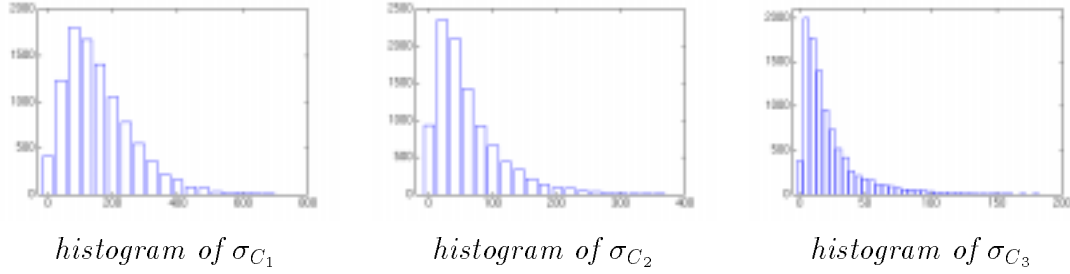


Figure 5.2: Histogram of the standard deviations of the wavelet coefficients in the lowest frequency band. Results were obtained from a database of more than 10,000 general-purpose images.

of  $\sigma_{C_1}$  are much much higher than the values of  $\sigma_{C_2}$  and  $\sigma_{C_3}$ . We consider  $\sigma_{C_1}$  more dominant than  $\sigma_{C_2}$  or  $\sigma_{C_3}$  alone. Moreover, most images in this general-purpose image database have low standard deviations. For any given standard deviation computed for the query, we want to find roughly the same number of images having standard deviations close to those of the query. Based on the trends shown in the histograms, we have developed the following selection criterion for the first step.

Denote the standard deviation information computed for the querying image as  $\sigma_{c_1}$ ,  $\sigma_{c_2}$  and  $\sigma_{c_3}$ . Denote the standard deviation information stored in the database indexing for an image as  $\sigma'_{c_1}$ ,  $\sigma'_{c_2}$  and  $\sigma'_{c_3}$ .

If the acceptance criteria<sup>2</sup>

$$\left( \sigma_{c_1} \beta < \sigma'_{c_1} < \frac{\sigma_{c_1}}{\beta} \right) \mid \mid \left[ \left( \sigma_{c_2} \beta < \sigma'_{c_2} < \frac{\sigma_{c_2}}{\beta} \right) \&\& \left( \sigma_{c_3} \beta < \sigma'_{c_3} < \frac{\sigma_{c_3}}{\beta} \right) \right] \quad (5.5)$$

fails, then we set the distance of the two images to 1, which means that the image will not be further considered in the matching process. Here,  $\beta = 1 - \frac{percent}{100}$  and *percent* is a threshold variable set to control the number of images passing the first matching phase. Usually it is set to around 50. Note that the above acceptance criteria holds if and only if the following expression holds.

$$\left( \sigma'_{c_1} \beta < \sigma_{c_1} < \frac{\sigma'_{c_1}}{\beta} \right) \mid \mid \left[ \left( \sigma'_{c_2} \beta < \sigma_{c_2} < \frac{\sigma'_{c_2}}{\beta} \right) \&\& \left( \sigma'_{c_3} \beta < \sigma_{c_3} < \frac{\sigma'_{c_3}}{\beta} \right) \right] \quad (5.6)$$

<sup>2</sup>Here we use standard C notation. That is,  $\mid \mid$  denotes OR and  $\&\&$  denotes AND.

Having first a fast and rough cut and then a more refined pass maintains the quality of the results while improving the speed of the matching. With  $percent = 50$ , about one-fifth of the images in the entire database passes through the first cut. We obtain a speed-up of about five by doing this step, compared to comparing all images in the database using stored wavelet coefficients. For a database of 10,000 images, about 2000 images will still be listed in the queue for the Euclidean distance comparison. Although it is possible that the first pass may discard some images that should be in the result list, the quality of the query response is slightly improved in more than 70% of the test cases due to this first pass. For example, an image with almost the same color, i.e., with low standard deviation values, is unlikely to have the same semantics as an image with very high color and texture variations, i.e., with high standard deviation values.

A weighted variation of Euclidean distance is used for the second phase comparison. The Euclidean distance compares all components in the feature vector with equal weightings and has a low computational complexity. If an image in the database differs from the querying image too much when we compare the  $8 \times 8 \times 3 = 192$  dimensional feature vector, we discard it. The remaining image vectors are used in the final matching, using the  $16 \times 16 \times 3 = 768$  dimensional feature vector with more detailed information considered. Let  $w_{1,1}$ ,  $w_{1,2}$ ,  $w_{2,1}$ ,  $w_{2,2}$ ,  $w_{c_1}$ ,  $w_{c_2}$  and  $w_{c_3}$  denote the weights. Then our distance function is defined as

$$\begin{aligned}
 & Dist(Image, Image') \\
 &= w_{1,1} \sum_{i=1}^3 (w_{c_i} \| W_{C_{i,1,1}} - W'_{C_{i,1,1}} \|) + w_{1,2} \sum_{i=1}^3 (w_{c_i} \| W_{C_{i,1,2}} - W'_{C_{i,1,2}} \|) \\
 &+ w_{2,1} \sum_{i=1}^3 (w_{c_i} \| W_{C_{i,2,1}} - W'_{C_{i,2,1}} \|) + w_{2,2} \sum_{i=1}^3 (w_{c_i} \| W_{C_{i,2,2}} - W'_{C_{i,2,2}} \|)
 \end{aligned} \tag{5.7}$$

where

$$\begin{aligned} W_{C_i,1,1} &= W_{C_i}(1 : 8, 1 : 8), & W_{C_i,1,2} &= W_{C_i}(1 : 8, 9 : 16), \\ W_{C_i,2,1} &= W_{C_i}(9 : 16, 1 : 8), & W_{C_i,2,2} &= W_{C_i}(9 : 16, 9 : 16) \end{aligned}$$

and  $\| u - v \|$  denotes the Euclidean distance. In practice, we may compute the square of the Euclidean distances instead in order to reduce computation complexity. If we let  $w_{j,k} = 1$ , then the function  $Dist(I_1, I_2)$  is the Euclidean distance between  $I_1$  and  $I_2$ . However, we may raise  $w_{2,1}$ ,  $w_{1,2}$ , or  $w_{2,2}$  if we want to emphasize the vertical, horizontal or diagonal edge details in the image. We may also raise  $w_{c_2}$  or  $w_{c_3}$  to emphasize the color variation more than the intensity variation. These weights can be determined by database developers based on the types of the images stored. For example, we raise  $w_{1,1}$  if the images in the database are all graphs (i.e., images with smooth local texture) because color is more important than local texture in distinguishing images in this case. For a photograph database, we assign the same value for all the weights. For a textured-image database, we emphasize the texture features by raising  $w_{2,1}$ ,  $w_{1,2}$ , and  $w_{2,2}$ .

To further speed up the system, we use a component threshold to reduce the amount of Euclidean distance computation. That is, if the difference at any component within the feature vectors to be compared is higher than a pre-defined threshold, we set the distance of the two images immediately to 1 so that the image will not be further considered in the matching process. For example, by setting this threshold to be the median value of the component distances obtain through random sampling, we obtain a speed-up of 2. The accuracy is degraded when we attempt to obtain a higher speed-up. We can use parallel computation to obtain a high speed-up and low degradation in accuracy.

The angle of any two feature vectors in the n-dimensional feature vector space is an alternative measure to the Euclidean distance we discussed above. The cosine value of the angle can be obtained by computing the vector dot product in a normalized vector space. This alternative measure reduces the sensitivity to color or brightness shift. However, the distance is much slower to compute than the Euclidean distance.

In our WBIIS system, we use the Euclidean distance to reduce the query processing time.

### 5.5.2 Partial query

A partial image query can be based on an image of low resolution, a partial image, a very low resolution block sketch or a hand-drawn sketch. Figure 5.3 shows the different types of partial image queries our system is designed to handle. We assume that the users do not care about the non-specified areas, but are only interested in finding images in the database that best match the specified areas of the query image. This kind of query is very useful in real-world digital libraries. 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 by cutting out the center area of an image with a white car. Figure 5.8 shows a similar example.

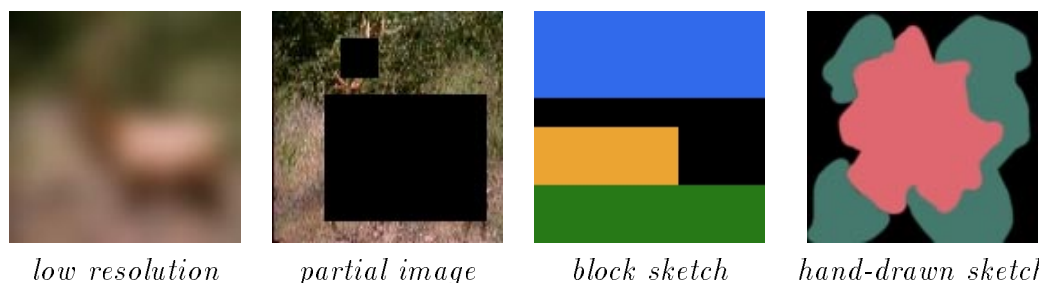


Figure 5.3: Types of partial sketch queries our WBIIS system handles. Black areas in a query image represent non-specified areas.

To handle partial image queries, spatial localization of the feature vector is crucial. For example, if we use some variations of the color moments to represent images, we would not be able to answer partial sketch queries because each element in a feature vector is a function of all pixels in the image. Due to the spatial localization properties of our wavelet-based image indexing, we can implement a retrieval algorithm for partial sketch queries with ease.

When a user submits a partial image query, we first rescale the query image into a  $128 \times 128$  rescaled image. At the same time, the non-specified areas are rescaled to fit in the  $128 \times 128$  rescaled image. A binary mask, denoted initially as

$M_0(1 : 128, 1 : 128)$  is created to represent the specified areas. Then we compute the feature vector of the rescaled query image using the wavelet-based indexing algorithm we discussed above with the non-specified areas being assigned as black. Here, the standard deviations are computed based on the wavelet coefficients within an  $8 \times 8$  mask  $M_4(1 : 8, 1 : 8)$  which is a subsample of  $M_0(1 : 128, 1 : 128)$ .

Comparison of the query feature vector with the stored vectors for the image database is done in two phases.

In the first phase, we compare the standard deviations computed for the querying image with the standard deviations within the mask for the wavelet coefficients stored for each image in the database. That is, we need to first re-compute the standard deviations of the wavelet coefficients in the masked areas for each image in the database. In cases where the users specify a majority of pixels in the query, we may simply use the pre-computed and stored standard deviation information. Then a similar distance measure is used to compare the standard deviation information.

A masked weighted variation of the Euclidean distance is used for the second phase comparison. The distance function is defined as<sup>3</sup>

$$\begin{aligned}
 & \text{Dist}(\text{Image}, \text{Image}') \\
 &= w_{1,1} \sum_{i=1}^3 ( w_{c_i} \| M_4 .* W_{C_{i,1,1}} - M_4 .* W'_{C_{i,1,1}} \| ) \\
 &+ w_{1,2} \sum_{i=1}^3 ( w_{c_i} \| M_4 .* W_{C_{i,1,2}} - M_4 .* W'_{C_{i,1,2}} \| ) \\
 &+ w_{2,1} \sum_{i=1}^3 ( w_{c_i} \| M_4 .* W_{C_{i,2,1}} - M_4 .* W'_{C_{i,2,1}} \| ) \\
 &+ w_{2,2} \sum_{i=1}^3 ( w_{c_i} \| M_4 .* W_{C_{i,2,2}} - M_4 .* W'_{C_{i,2,2}} \| )
 \end{aligned} \tag{5.8}$$

If an image in the database differs from the querying image too much when we

---

<sup>3</sup>Here we use standard MATLAB notation. That is, ‘.\*’ denotes component-wise product.

compare the  $8 \times 8 \times 3 = 192$  dimensional feature vector, we again discard it. The remaining image vectors are used in the final matching, using the  $16 \times 16 \times 3 = 768$  dimensional feature vector. The measure is the same as discussed in the previous subsection except that we assign different weights in the three color components for partial queries with low resolution. In fact, when the resolution in the partial sketch is low, we need to emphasize the color variation rather than the intensity variation. For example, a red block (i.e.,  $R=255, G=0, B=0$ ) shows the same color intensity with a green block (i.e.,  $R=0, G=255, B=0$ ). As a result, we raise  $w_{c_2}$  and  $w_{c_3}$  to about twice the setting for  $w_{c_1}$ .

## 5.6 Performance

The WBIIS algorithm has been implemented by embedding it within the IBM QBIC multimedia database system [88]. The discrete fast wavelet transforms are performed on IBM RS/6000 workstations. To compute the feature vectors for the 10,000 color images in our database requires approximately 2 hours of CPU time.

The matching speed is very fast. Using a SUN Sparc-20 workstation, a fully-specified query takes about 3.3 seconds of response time with 1.8 seconds of CPU time to select the best 100 matching images from the 10,000 image database using our similarity measure. It takes about twice the time to answer a partially specified query. The speed is about twice as fast as the IBM QBIC system and the VIRAGE system. The system developed at the University of Washington is much faster due to their fast binary matching algorithm.

There are ways to further speed up the system for very large image databases. For example, we may pre-sort and store the standard deviation information within the feature vectors of the images in the database because we must compare this information for each query. Also, we may use a better algorithm to find the first  $k$  matching images if  $k$  is smaller than  $\log_2(n)$  if the database contains  $n$  images. In fact, an algorithm of execution time of  $O(kn)$  can be constructed for this task to replace the quick-sort algorithm with run time  $O(n \log(n))$  we are currently using.

Figures 5.4, 5.5, 5.6 and 5.7 show accuracy comparisons of our wavelet algorithm



Figure 5.4: Comparisons with a commercial algorithm (IBM QBIC) on a galaxy-type image. Note that 12 out of 15 images retrieved by the commercial algorithm are unrelated to the galaxy query image. WBIIS retrieved only 6 unrelated images. The upper-left corner image in each block of images is the query. The image to the right of that image is the best matching image found. Matches decrease in measured closeness from left to right and from top to bottom. Results were obtained from a database of approximately 10,000 images.



*algorithm by University of Washington*



*WBIS*

Figure 5.5: A query example. 9 images unrelated to a water scene were retrieved by the University of Washington algorithm. WBIS retrieved only one unrelated image. The upper-left corner image in each block of images is the query. Results were obtained from a database of approximately 10,000 images.





*algorithm by University of Washington*



*WBIS with Haar Wavelet*

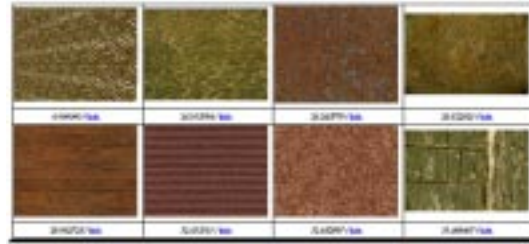


*WBIS with Daubechies' Symmlet-8 Wavelet*

Figure 5.6: Another query example.



*algorithm by University of Washington*



*commercial algorithm by VIRAGE*



*WBIIS with Haar Wavelet*



*WBIIS with Daubechies' Symmlet-8 Wavelet*

Figure 5.7: Comparison on a texture image.



(a) fine block sketch



(b) rough block sketch



(c) image with omitting block(s)

Figure 5.8: Partial sketch queries in different resolutions. The upper-left corner image in each block of images is the query. Black areas in a query image represent non-specified areas. Database size: 10,000 images.



*WBIS*



*algorithm by University of Washington (query image not shown)*

Figure 5.9: Query results on a hand-drawn query image (with blue, black, yellow, and green blocks). Black areas in a query image represent non-specified areas. Equivalent query were used. Database size: 10,000 images.

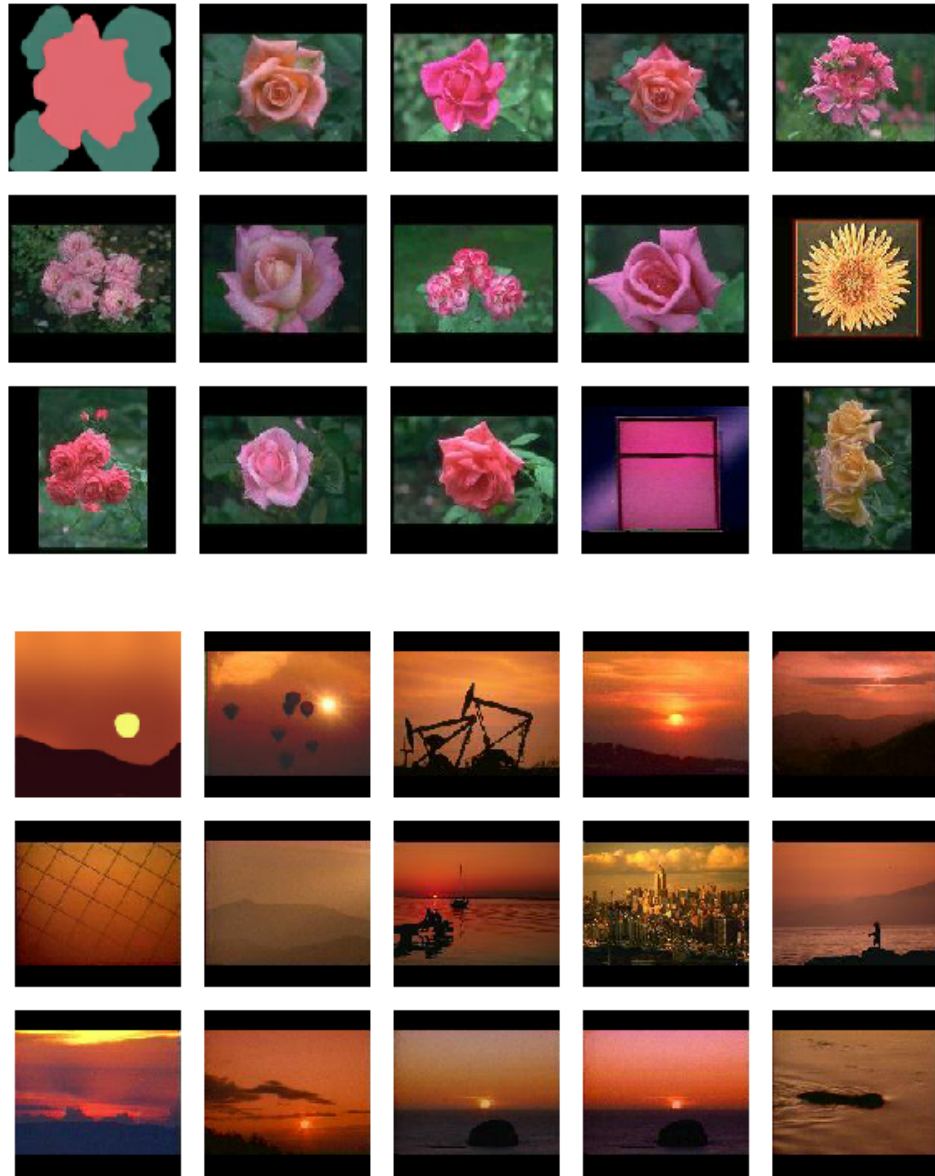


Figure 5.10: Two other query examples using WBIIS. The upper-left corner image in each block of images is the query.

with the color layout algorithms in IBM QBIC and VIRAGE [46], two of the most popular commercial multimedia databases, and the system developed at the University of Washington. Figures 5.8, 5.10 and 5.9 show the query results obtained from partial sketch image queries. Default parameters are used for the University of Washington's algorithm. In all cases, the number of reasonably similar images retrieved by our algorithm within the best matches is higher. In our comparisons of query results, we consider one retrieval algorithm as better than another if the number of similar images among a fixed number of best matching images is higher. We do not attempt to compare two images which are both very similar to a query image because we do not have a quantitative measure of the similarity between two images. When several images are all very close to the query image, it is meaningless to rank their similarities to the query image since subjective opinions often dominate and the distances are too close to make ranking orders simply based on sorting results. For example, in Figure 5.5, the second and third images retrieved by our algorithm are both very close to the query image (the first image). Some reader may favor the second image because the color of the boat is the same as that of the boat in the query image; on the other hand, some may favor the third image since it has a closer composition to the query image.

We compared the WBIIS system with the IBM QBIC, the VIRAGE, and the University of Washington's system using the same COREL database. We manually tested about 100 random queries in the database. In about one half of the cases, the WBIIS system performs equally well or slightly better. When the query image is not smooth (as in the shown query examples), the WBIIS system outperform the other systems significantly. The results are expected because of the energy-concentration properties (shown in Chapter 3) of the Daubechies' wavelet transforms.

## 5.7 Limitations

WBIIS is designed to be invariant to scale and aspect ratio changes since query images and all images in the database are normalized to the same size and aspect ratio before the matching step. Color and intensity shift can be handled by the alternative

measure discussed at the end of Section 3.5. However, like color layout indexing, severe cropping (i.e., query based on subregions) cannot be handled. In Chapter 6, we discuss our region-based approach and the recently developed SIMPLIcity system, an image indexing and retrieval system using wavelet-based features. The integrated region matching (IRM) metric for the SIMPLIcity system provides much better robustness with respect to cropping and scaling changes, as well as other variations.

WBIIS system is designed to handle color layout type queries. Because of the nature of color layout search, WBIIS has limitations in certain types of applications when high degrees of rotation and translation invariance are important. However, WBIIS can handle small amount of rotation and translation changes. In the searching phase, a global measure, i.e., the set of the standard deviations of the saved wavelet coefficients, is utilized to measure the image coherence. Multi-scale indexing scheme is also used to avoid bias.

We have performed robustness tests using several randomly selected image queries, WBIIS with Daubechies' wavelets is capable of handling a maximum rotation of 20 degrees and a maximum translation of around 20% in general. In Figure 5.5, for instance, the system successfully finds images with wind surfers in various parts, many of which differ considerably from that of the query. Similar situations can be found in Figures 5.6, 5.7 and 5.9. The system is sensitive to rotation and translation changes when performing partial sketch search with large non-specified areas.

## 5.8 Summary

In this chapter, we have explored some alternatives for improving both the speed and accuracy of earlier color layout image indexing algorithms used in large multimedia database systems. An efficient wavelet-based multi-scale indexing and matching system using Daubechies' wavelets developed by us has been demonstrated. The system is capable of handling both fully-specified queries and partial sketch queries. Like color layout indexing, it has limitations with respect to cropping, translational and rotational changes. In the next chapter, we address this issue by introducing details of our recently developed SIMPLIcity system.

# Chapter 6

## Semantics-sensitive Integrated Matching

*The important thing is not to stop questioning.*

*Curiosity has its own reason for existing.*

— Albert Einstein (1879-1955)

### 6.1 Introduction

We present here SIMPLIcity (Semantics-sensitive Integrated Matching for Picture Libraries), an image database retrieval system, which uses high-level semantics classification and integrated region matching (IRM) based upon image segmentation. The SIMPLIcity system represents an image by a set of regions, roughly corresponding to objects, which are characterized by color, texture, shape, and location. Based on segmented regions, the system classifies images into semantically meaningful categories. These high-level categories, such as textured-nontextured, indoor-outdoor, objectionable-benign, graph-photograph, enhance retrieval by narrowing down the



searching range in a database and permitting semantically-adaptive searching methods. A measure for the overall similarity between images is defined by a region-matching scheme that integrates properties of all the regions in the images. Armed with this global similarity measure, the system provides users a simple querying interface. The integrated region matching (IRM) similarity measure is insensitive to inaccurate segmentation.

Section 6.2 gives an overview of the system architecture. The image segmentation process is introduced in Section 6.3. In Section 6.4, we give details of the classification process. The similarity metric, i.e., the IRM metric, is defined in Section 6.5. In Section 6.6, we describe the main concepts of a system specially developed for biomedical image databases, based on concepts and methods discussed earlier. For very large image databases, we propose the use of TSVQ to cluster the data. Details of the process are given in Section 6.7.

## 6.2 Overview

The architecture of the SIMPLIcity system is described by Figure 6.1, the indexing process, and Figure 6.2, the querying process. During indexing, the system partitions an image into  $4 \times 4$  pixel blocks and extracts a feature vector for each block. A statistical clustering [50] algorithm is then used to quickly segment the image into regions. The segmentation result is fed into a classifier that decides the semantic type of the image. An image is currently classified as one of the  $n$  manually-defined mutually exclusive and collectively exhaustive semantic classes. The system can be extended to allow an image to be softly classified into multiple classes with probability assignments. Examples of semantic types are indoor-outdoor, objectionable-benign, textured-nontextured, city-landscape, with-without people, and graph-photograph images. Features including color, texture, shape, and location information are then extracted for each region in the image. The features selected depend on the semantic type of the image. The signature of an image is the collection of features for all of its regions. Signatures of images with various semantic types are stored in separate databases.

In the querying process, if the query image is not in the database as indicated by the user interface, it is first passed through the same feature extraction process as was used during indexing. For an image in the database, its semantic type is first checked and then its signature is extracted from the corresponding database. Once the signature of the query image is obtained, similarity scores between the query image and images in the database with the same semantic type are computed and sorted to provide the list of images that appear to have the closest semantics.

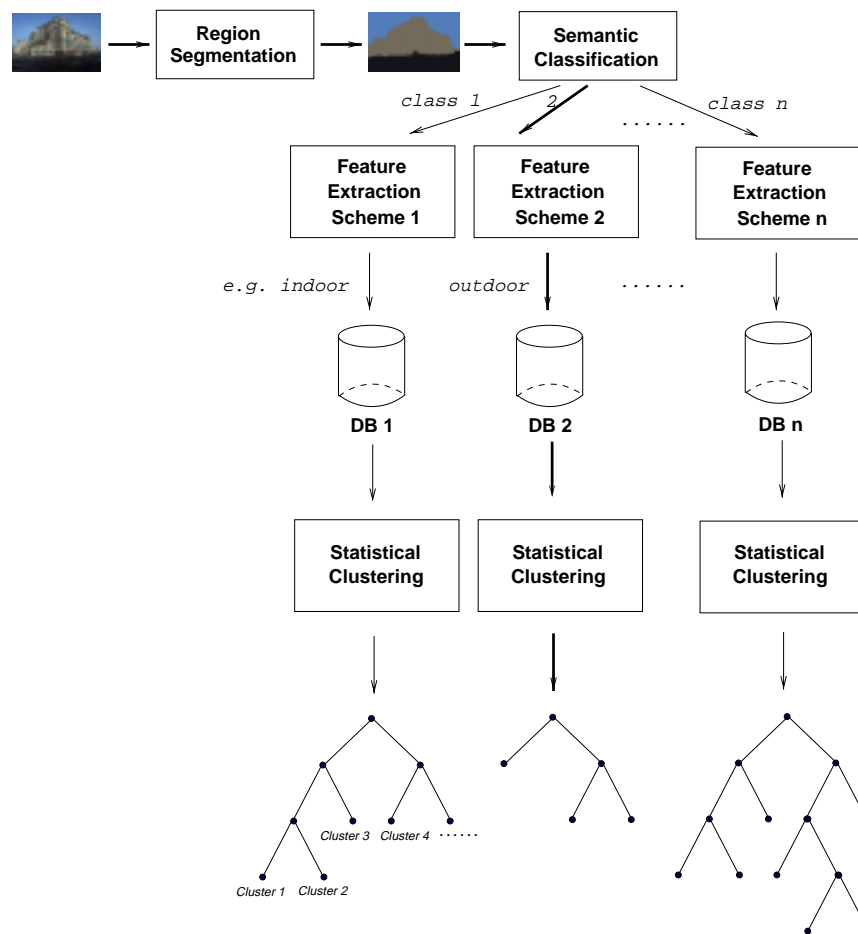


Figure 6.1: The architecture of feature indexing process. The heavy lines show a sample indexing path of an image.

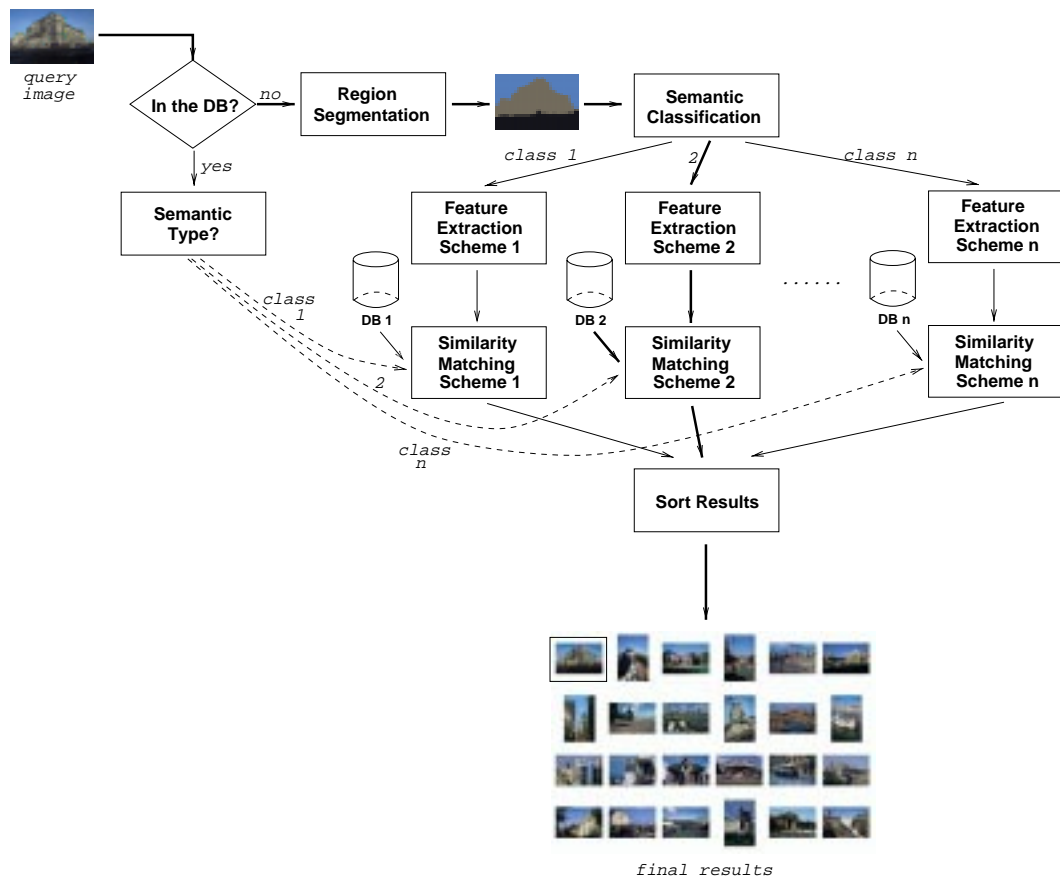


Figure 6.2: The architecture of query processing process. The heavy lines show a sample querying path of an image.

### 6.3 Image segmentation

This section describes our image segmentation procedure based on color and frequency features using the k-means algorithm [50] described in Chapter 4. For general-purpose images such as the images in a photo library or the images on the World-Wide Web (WWW), automatic image segmentation is almost as difficult as automatic image semantic understanding. Currently there is no non-stereo image segmentation algorithm that can perform at the level of the human visual system (HVS). The segmentation accuracy of our system is not crucial because we use a more robust integrated region-matching (IRM) scheme which is insensitive to inaccurate segmentation.

To segment an image, SIMPLIcity partitions the image into blocks with  $4 \times 4$  pixels and extracts a feature vector for each block. The k-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one region in the segmented image. An alternative to the block-wise segmentation is a pixel-wise segmentation by forming a window centered around every pixel. A feature vector for a pixel is then extracted from the windowed block. The advantage of pixel-wise segmentation over block-wise segmentation is the removal of blockiness at boundaries between regions. Since we use rather small block size and boundary blockiness has little effect on retrieval, we choose block-wise segmentation with the benefit of 16 times faster segmentation.

Details of the  $k$ -means algorithm are given in Chapter 4. Suppose observations are  $\{x_i : i = 1, \dots, L\}$ . The goal of the  $k$ -means algorithm is to partition the observations into  $k$  groups with means  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_k$  such that

$$D(k) = \sum_{i=1}^L \min_{1 \leq j \leq k} (x_i - \hat{x}_j)^2 \quad (6.1)$$

is minimized. The  $k$ -means algorithm does not specify how many clusters to choose. We adaptively choose the number of clusters  $k$  by gradually increasing  $k$  and stop when a criterion is met. We start with  $k = 2$  and stop increasing  $k$  if one of the following conditions is satisfied.

1. The distortion  $D(k)$  is below a threshold. A low  $D(k)$  indicates high purity in

the clustering process. The threshold is not critical because the IRM measure is not sensitive to  $k$ .

2. The first derivative of distortion with respect to  $k$ ,  $D(k) - D(k - 1)$ , is below a threshold with comparison to the average derivative at  $k = 2, 3$ . A low  $D(k) - D(k - 1)$  indicates convergence in the clustering process. The threshold determines the overall time to segment images and needs to be set to a near-zero value. It is critical to the speed, but not the quality of the final image segmentation. The threshold can be adjusted according to the experimental run-time.
3. The number  $k$  exceeds an upper bound. We allow an image to be segmented into a maximum of 16 segments. That is, we assume an image has less than 16 distinct types of objects. Usually the segmentation process generates much less number of segments in an image. The threshold is rarely met.

Six features are used for segmentation. Three of them are the average color components in a  $4 \times 4$  block. The other three represent energy in high frequency bands of wavelet transforms [24, 84], that is, the square root of the second order moment of wavelet coefficients in high frequency bands. We use the well-known LUV color space, where L encodes luminance, and U and V encode color information (chrominance). The LUV color space has good perception correlation properties. Details of the color spaces are given in Chapter 2. We chose the block size to be  $4 \times 4$  to compromise between the texture detail and the computation time.

To obtain the other three features, we apply either the Haar wavelet transform or the Daubechies-4 wavelet transform to the L component of the image. We use these two wavelet transforms because they capture more texture information. After a one-level wavelet transform, a  $4 \times 4$  block is decomposed into four frequency bands as shown in Figure 6.3. Each band contains  $2 \times 2$  coefficients. Without loss of generality, suppose the coefficients in the HL band are  $\{c_{k,l}, c_{k,l+1}, c_{k+1,l}, c_{k+1,l+1}\}$ . One feature

is then computed as

$$f = \left( \frac{1}{4} \sum_{i=0}^1 \sum_{j=0}^1 c_{k+i,l+j}^2 \right)^{\frac{1}{2}}. \quad (6.2)$$

The other two features are computed similarly from the LH and HH bands. The motivation for using the features extracted from high frequency bands is that they reflect texture properties. Moments of wavelet coefficients in various frequency bands have been shown to be effective for representing texture [114]. The intuition behind this is that coefficients in different frequency bands show variations in different directions. For example, the HL band shows activities in the horizontal direction. An image with vertical strips thus has high energy in the HL band and low energy in the LH band. This texture feature is a good compromise between computational complexity and effectiveness.

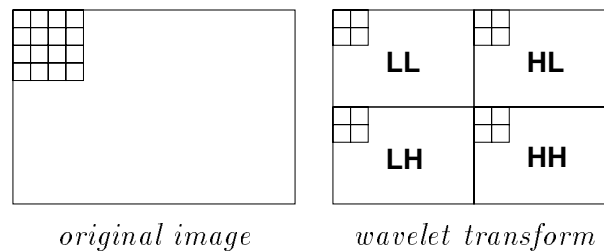
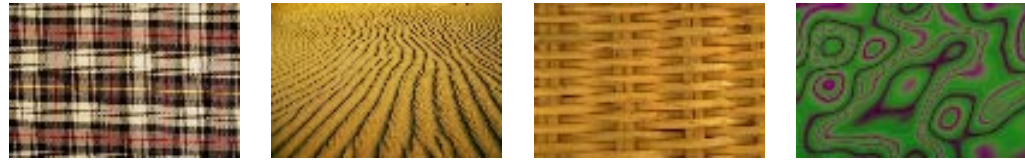


Figure 6.3: Decomposition of images into frequency bands by wavelet transforms.

Examples of segmentation results for both textured and non-textured images are shown in Figure 6.4. Segmented regions are shown in their representative colors. It takes about one second on average to segment a  $384 \times 256$  image on a Pentium Pro 430MHz PC using the Linux operating system. We do not apply post-processing techniques to smooth region boundaries or to delete small isolated regions because these errors are rarely significant. Since our retrieval system is designed to tolerate inaccurate segmentation, refining the segmentation results by post-processing (at the cost of speed) is unnecessary.



(a)



#regions=4  
 $\bar{\chi}^2 = 0.125$

#regions=2  
 $\bar{\chi}^2 = 0.207$

#regions=2  
 $\bar{\chi}^2 = 0.009$

#regions=3  
 $\bar{\chi}^2 = 0.066$

(b)



(c)



#regions=2  
 $\bar{\chi}^2 = 0.694$

#regions=4  
 $\bar{\chi}^2 = 1.613$

#regions=4  
 $\bar{\chi}^2 = 1.447$

#regions=12  
 $\bar{\chi}^2 = 1.249$

(d)

Figure 6.4: Segmentation results by the k-means clustering algorithm: (a) Original texture images, (b) Regions of the texture images, (c) Original non-textured images, (d) Regions of the non-textured images.

## 6.4 Image classification

The image classification methods described in this section have been developed mainly for searching picture libraries such as Web images. We are initially interested in classifying images into the classes textured vs. non-textured, graph vs. photograph, and objectionable vs. benign. Other classification methods such as city vs. landscape [115] and with people vs. without people [16, 9] were developed elsewhere. Details of the objectionable image classification method are given as an appendix (Appendix A).

### 6.4.1 Textured vs. non-textured images

In this section we describe the algorithm to classify images into the semantic classes *textured* or *non-textured*. By textured images, we refer to images that are composed of repeated patterns and appear like a unique texture surface.

For textured images, color and texture are much more important perceptually than shape, since there are no clustered objects. As shown by the segmentation results in Figure 6.4, regions in textured images tend to scatter in the entire image, whereas non-textured images are usually partitioned into clumped regions. A mathematical description of how evenly a region scatters in an image is the goodness of match between the distribution of the region and a uniform distribution. The goodness of fit is measured by the  $\chi^2$  statistics [108].

We partition an image evenly into 16 zones,  $\{Z_1, Z_2, \dots, Z_{16}\}$ . Suppose the image is segmented into regions  $\{r_i : i = 1, \dots, m\}$ . For each region  $r_i$ , its percentage in zone  $Z_j$  is  $p_{i,j}$ ,  $\sum_{j=1}^{16} p_{i,j} = 1$ ,  $i = 1, \dots, m$ . The uniform distribution over the zones should have probability mass function  $q_j = 1/16$ ,  $j = 1, \dots, 16$ . The  $\chi^2$  statistics for region  $i$ ,  $\chi_i^2$ , is computed by

$$\chi_i^2 = \sum_{j=1}^{16} \frac{(p_{i,j} - q_j)^2}{q_j} = \sum_{j=1}^{16} 16(p_{i,j} - \frac{1}{16})^2. \quad (6.3)$$

The classification of textured or non-textured image is performed by thresholding the average  $\chi^2$  for all the regions in the image,  $\bar{\chi}^2 = \frac{1}{m} \sum_{i=1}^m \chi_i^2$ . If  $\bar{\chi}^2 < 0.32$ , the image



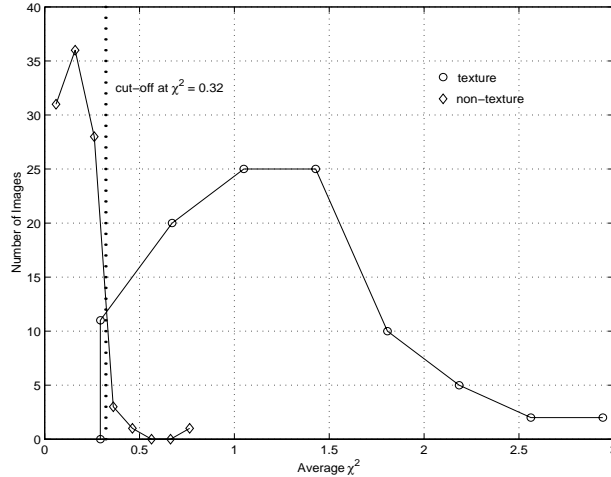


Figure 6.5: The histograms of average  $\chi^2$ 's over 100 textured images and 100 non-textured images.

is labeled as textured; otherwise, non-textured. We randomly chose 100 textured images and 100 non-textured images and computed  $\bar{\chi}^2$  for them. The histograms of  $\bar{\chi}^2$  for the two types of images are shown in Figure 6.5. It can be seen that the two histograms separate significantly around the decision threshold 0.32.

### 6.4.2 Graph vs. photograph images

Now we describe our algorithm to classify images into the semantic classes *graph* or *photograph*. We use this classification method for general-purpose image databases (e.g., WWW images). An image is a photograph if it is a continuous-tone image. A graph image is an image containing mainly text, graph and overlays. We have developed a graph-photograph classification method.

The classifier partitions an image into blocks and classifies every block into either of the two classes. If the percentage of blocks classified as photograph-type is higher than a threshold, the image is marked as photograph; otherwise, it is marked as text. The algorithm we used to segment image blocks is based on a probability density analysis for wavelet coefficients in high frequency bands. For every block, two feature values, which describe the distribution pattern of the wavelet coefficients in high

frequency bands, are evaluated. Then the block is marked as a corresponding class according to the two feature values. We do not count pure black or pure white blocks because index images usually have black or white backgrounds. The algorithm is based on a multiresolution document image segmentation algorithm [69].

We tested the classification method on a database of 12,000 photographic images and a database of 300 randomly downloaded graph-based image maps from the web. We achieved 100% sensitivity for photographic images and higher than 95% specificity.

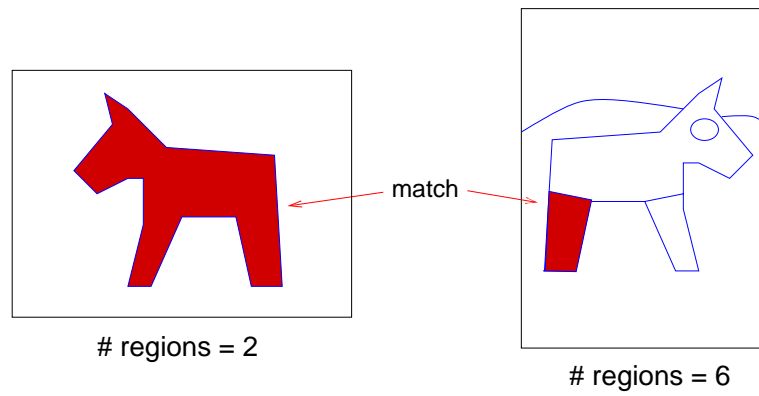
## 6.5 The similarity metric

In this section, we give details of our novel integrated region matching (IRM) algorithm. IRM is a measure for the overall similarity between images that integrates properties of all the regions in the images. The advantage of using such a soft matching is that it makes the metric robust to poor segmentation (Figure 6.6), an important property that previous work [11, 77] has not solved.

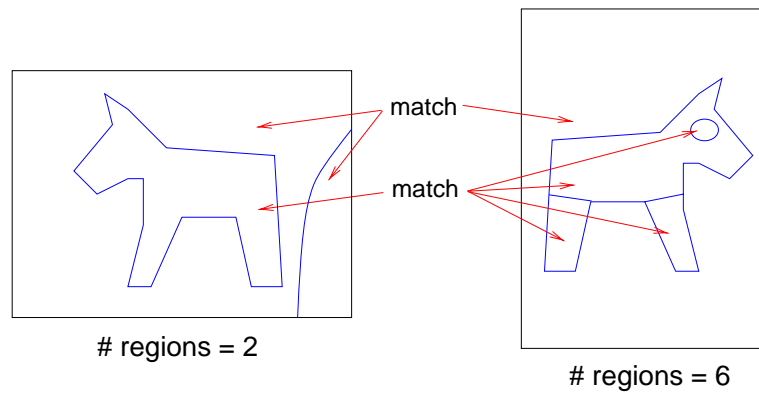
IRM can be used for image classes that are suitable to region-based matching. As shown in Figure 6.2, different feature extraction and matching schemes are used in SIMPLiCity for different semantic classes. Color layout features are suitable for outdoor landscape photographs. Texture features gives better results on textured images. Region-based features are needed for many other semantic classes, such as indoor photos, portraits, and city photos.

### 6.5.1 Integrated region matching

In this section, we define the similarity measure between two sets of regions. Assume that Image 1 and 2 are represented by region sets  $R_1 = \{r_1, r_2, \dots, r_m\}$  and  $R_2 = \{r'_1, r'_2, \dots, r'_n\}$ , where  $r_i$  or  $r'_i$  is the descriptor of region  $i$ . Denote the distance between region  $r_i$  and  $r'_j$  as  $d(r_i, r'_j)$ , which is written as  $d_{i,j}$  in short. Details about features included in  $r_i$  and the definition of  $d(r_i, r'_j)$  will be discussed later. To compute the similarity measure between region sets  $R_1$  and  $R_2$ ,  $d(R_1, R_2)$ , we first match all regions in the two images. When we judge the similarity of two animal photographs,



Traditional region-based matching



Integrated Region Matching (IRM)

Figure 6.6: Integrated Region Matching (IRM) is robust to poor image segmentation.

we usually compare the animals in the images before comparing the background areas in the images. The overall similarity of the two images depends on the closeness in the two aspects. The correspondence between objects in the images is crucial to our judgment of similarity since it would be meaningless to compare the animal in one image with the background in another. Our matching scheme aims at building correspondence between regions that is consistent with human perception. To increase robustness against segmentation errors, we allow a region to be matched to several regions in another image. A matching between  $r_i$  and  $r'_j$  is assigned with a significance credit  $s_{i,j}$ ,  $s_{i,j} \geq 0$ . The significance credit indicates the importance of the matching for determining similarity between images. The matrix

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,n} \end{pmatrix}, \quad (6.4)$$

is referred to as the significance matrix.

A graphical explanation of the integrated matching scheme is provided in Figure 6.7. The figure shows that matching between images can be represented by an edge weighted graph in which every vertex in the graph corresponds to a region. If two vertices are connected, the two regions are matched with a significance credit being the weight on the edge. To distinguish from matching two sets of regions, we refer to the matching of two regions as they are *linked*. The length of an edge can be regarded as the distance between the two regions represented. If two vertices are not connected, the corresponding regions are either from the same image or the significance credit of matching them is zero. Every matching between images is characterized by links between regions and their significance credits. The matching used to compute the distance between two images is referred to as the *admissible matching*. The admissible matching is specified by conditions on the significance matrix. If a graph represents an admissible matching, the distance between the two region sets is

the summation of all the weighted edge lengths, i.e.,

$$d(R_1, R_2) = \sum_{i,j} s_{i,j} d_{i,j} . \quad (6.5)$$

We call this distance the integrated region matching (IRM) distance.

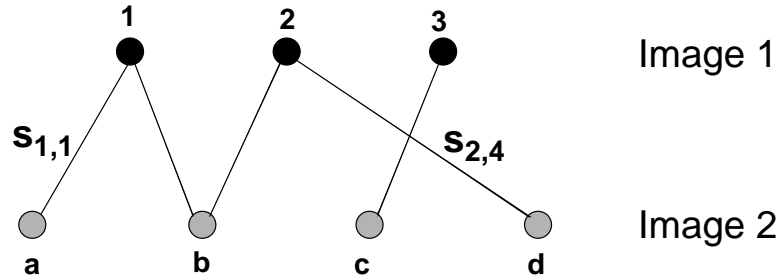


Figure 6.7: Integrated region matching (IRM).

The problem of defining distance between region sets is then converted to choosing the significance matrix  $S$ . A natural issue to raise is what constraints should be put on  $s_{i,j}$  so that the admissible matching yields good similarity measure. In other words, what properties do we expect an admissible matching to possess? The first property we want to enforce is the fulfillment of significance. Assume that the significance of  $r_i$  in Image 1 is  $p_i$ , and  $r'_j$  in Image 2 is  $p'_j$ , we require that

$$\sum_{j=1}^n s_{i,j} = p_i, \quad i = 1, \dots, m \quad (6.6)$$

$$\sum_{i=1}^m s_{i,j} = p'_j, \quad j = 1, \dots, n . \quad (6.7)$$

For normalization, we have  $\sum_{i=1}^m p_i = \sum_{j=1}^n p'_j = 1$ . The fulfillment of significance ensures that all the regions play a role for measuring similarity. We also require an admissible matching to link the most similar regions at the highest priority. For example, if two images are the same, the admissible matching should link a region in Image 1 only to the same region in Image 2. With this matching, the distance between the two images equals zero, which coincides with our intuition. The IRM

algorithm attempts to fulfill the significance credits of regions by assigning as much significance as possible to the region link with minimum distance. We call this the “most similar highest priority (MSHP)” principle. Initially, assume that  $d_{i',j'}$  is the minimum distance, we set  $s_{i',j'} = \min(p_{i'}, p'_{j'})$ . Without loss of generality, assume  $p_{i'} \leq p'_{j'}$ . Then  $s_{i',j} = 0$ , for  $j \neq j'$  since the link between regions  $i'$  and  $j'$  has filled the significance of region  $i'$ . The significance credit left for region  $j'$  is reduced to  $p'_{j'} - p_{i'}$ . The updated matching problem is then solving  $s_{i,j}$ ,  $i \neq i'$ , by the MSHP rule under constraints:

$$\sum_{j=1}^n s_{i,j} = p_i \quad 1 \leq i \leq m, i \neq i' \quad (6.8)$$

$$\sum_{i:1 \leq i \leq m, i \neq i'} s_{i,j} = p'_j \quad 1 \leq j \leq n, j \neq j' \quad (6.9)$$

$$\sum_{i:1 \leq i \leq m, i \neq i'} s_{i,j'} = p'_{j'} - p_{i'} \quad (6.10)$$

$$s_{i,j} \geq 0 \quad 1 \leq i \leq m, i \neq i'; 1 \leq j \leq n. \quad (6.11)$$

We apply the previous procedure to the updated problem. The iteration stops when all the significance credits  $p_i$  and  $p'_j$  have been met. The algorithm is summarized as follows.

1. Set  $\mathcal{L} = \{\}$ , denote  $\mathcal{M} = \{(i, j) : i = 1, \dots, m; j = 1, \dots, n\}$ .
2. Choose the minimum  $d_{i,j}$  for  $(i, j) \in \mathcal{M} - \mathcal{L}$ . Label the corresponding  $(i, j)$  as  $(i', j')$ .
3.  $\min(p_{i'}, p'_{j'}) \rightarrow s_{i',j'}$ .
4. If  $p_{i'} < p'_{j'}$ , set  $s_{i',j} = 0$ ,  $j \neq j'$ ; otherwise, set  $s_{i,j'} = 0$ ,  $i \neq i'$ .
5.  $p_{i'} - \min(p_{i'}, p'_{j'}) \rightarrow p_{i'}$ .
6.  $p'_{j'} - \min(p_{i'}, p'_{j'}) \rightarrow p'_{j'}$ .
7.  $\mathcal{L} + \{(i', j')\} \rightarrow \mathcal{L}$ .

8. If  $\sum_{i=1}^m p_i > 0$  and  $\sum_{j=1}^n p'_j > 0$ , go to Step 2; otherwise, stop.

Consider an example of applying the integrated region matching algorithm. Assume that  $m = 2$  and  $n = 3$ . The values of  $p_i$  and  $p'_j$  are

$$p_1 = 0.4, \quad p_2 = 0.6 \quad (6.12)$$

$$p'_1 = 0.2, \quad p'_2 = 0.3, \quad p'_3 = 0.5. \quad (6.13)$$

The region distance matrix  $\{d_{i,j}\}$ ,  $i = 1, 2$ ,  $j = 1, 2, 3$ , is

$$\begin{pmatrix} 0.5 & 1.2 & 0.1 \\ 1.0 & 1.6 & 2.0 \end{pmatrix}. \quad (6.14)$$

The sorted  $d_{i,j}$  is

$$\begin{array}{l} (i,j) : (1,3) \quad (1,1) \quad (2,1) \quad (1,2) \quad (2,2) \quad (2,3) \\ d_{i,j} : 0.1 \quad 0.5 \quad 1.0 \quad 1.2 \quad 1.6 \quad 2.0. \end{array} \quad (6.15)$$

The first two regions matched are regions 1 and 3. As the significance of region 1,  $p_1$ , is fulfilled by the matching, region 1 in Image 1 is no longer in consideration. The second pair of regions matched is then regions 2 and 1. The region pairs are listed below in the order of being matched:

$$\begin{array}{l} \text{region pairs : } (1,3) \quad (2,1) \quad (2,2) \quad (2,3) \\ \text{significance : } 0.4 \quad 0.2 \quad 0.3 \quad 0.1. \end{array} \quad (6.16)$$

The significance matrix is

$$\begin{pmatrix} 0.0 & 0.0 & 0.4 \\ 0.2 & 0.3 & 0.1 \end{pmatrix}. \quad (6.17)$$

We now come to the issue of choosing  $p_i$ . The value of  $p_i$  is chosen to reflect the significance of region  $i$  in the image. If we assume that every region is equally important, then  $p_i = 1/m$ , where  $m$  is the number of regions. In the case that Image

1 and Image 2 have the same number of regions, a region in Image 1 is matched exclusively to one region in Image 2. Another choice of  $p_i$  is the percentage of the image covered by region  $i$  based on the view that important objects in an image tend to occupy larger areas. We refer to this assignment of  $p_i$  as the *area percentage scheme*. This scheme is less sensitive to inaccurate segmentation than the uniform scheme. If one object is partitioned into several regions, the uniform scheme raises its significance improperly, whereas the area percentage scheme retains its significance. On the other hand, if objects are merged into one region, the area percentage scheme assigns relatively high significance to the region. The SIMPLicity system uses the area percentage scheme.

The scheme of assigning significance credits can also take region location into consideration. For example, higher significance may be assigned to regions in the center of an image than to those around boundaries. Another way to count location in the similarity measure is to generalize the definition of the IRM distance to

$$d(R_1, R_2) = \sum_{i,j} s_{i,j} w_{i,j} d_{i,j} . \quad (6.18)$$

The parameter  $w_{i,j}$  is chosen to adjust the effect of region  $i$  and  $j$  on the similarity measure. In the SIMPLicity system, regions around boundaries are slightly down-weighted by using this generalized IRM distance.

In the future, we will explore different schemes for assigning significance credits. We are especially interested in center-weighted region significance.

### 6.5.2 Distance between regions

We now discuss the definition of distance between a region pair,  $d(r, r')$ . The SIMPLicity system characterizes a region by color, texture, and shape. The feature extraction process is shown in Figure 6.8. We have described in Section 6.3 the features used by the  $k$ -means algorithm for segmentation. The mean values of these features in one cluster are used to represent color and texture in the corresponding region. These features are restated in the following list:



1.  $f_1$  = the average L component of color
2.  $f_2$  = the average U component of color
3.  $f_3$  = the average V component of color
4.  $f_4$  = the square root of the second order moment of wavelet coefficients in the HL band
5.  $f_5$  = the square root of the second order moment of wavelet coefficients in the LH band
6.  $f_6$  = the square root of the second order moment of wavelet coefficients in the HH band

To describe shape, normalized inertia [38] of order 1 to 3 are used. For a region  $H$  in  $k$  dimensional Euclidean space  $\mathfrak{R}^k$ , its normalized inertia of order  $\gamma$  is

$$l(H, \gamma) = \frac{\int_H \|x - \hat{x}\|^\gamma dx}{[V(H)]^{1+\gamma/k}} \quad (6.19)$$

where  $\hat{x}$  is the centroid of  $H$  and  $V(H)$  is the volume of  $H$ . Since an image is specified by pixels on a grid, the discrete form of the normalized inertia is used, that is,

$$l(H, \gamma) = \frac{\sum_{x:x \in H} \|x - \hat{x}\|^\gamma}{[V(H)]^{1+\gamma/k}} \quad (6.20)$$

where  $V(H)$  is the number of pixels in region  $H$ . The normalized inertia is invariant with scaling and rotation. The minimum normalized inertia is achieved by spheres. Denote the  $\gamma$ th order normalized inertia of spheres as  $L_\gamma$ . We define shape features as  $l(H, \gamma)$  normalized by  $L_\gamma$ :

$$f_7 = l(H, 1)/L_1, \quad f_8 = l(H, 2)/L_2, \quad f_9 = l(H, 3)/L_3. \quad (6.21)$$

The computation of shape features is skipped for textured images because in this case region shape is not important perceptually. The region distance  $d(r, r')$  is defined

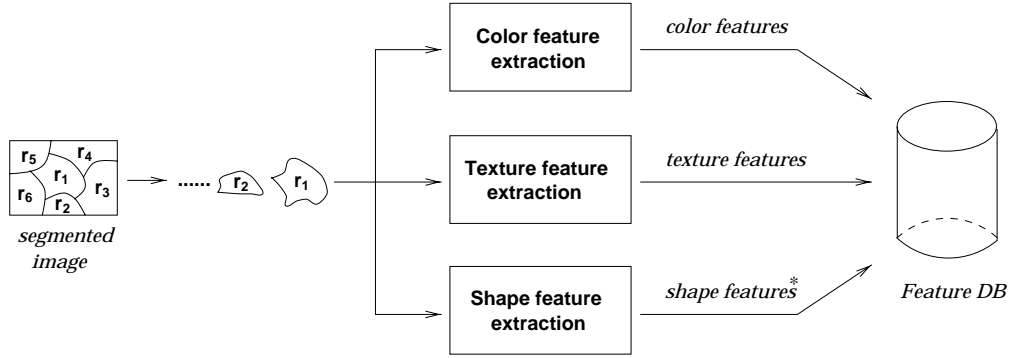


Figure 6.8: Feature extraction in the SIMPLiCity system. (\* The computation of shape features is omitted for textured images.)

as

$$d(r, r') = \sum_{i=1}^6 w_i (f_i - f'_i)^2 . \quad (6.22)$$

For non-textured images,  $d(r, r')$  is defined as

$$d(r, r') = g(d_s(r, r')) \cdot d_t(r, r') , \quad (6.23)$$

where  $d_s(r, r')$  is the shape distance computed by

$$d_s(r, r') = \sum_{i=7}^9 w_i (f_i - f'_i)^2 , \quad (6.24)$$

and  $d_t(r, r')$  is the color and texture distance defined the same as the distance between textured image regions, i.e.,

$$d_t(r, r') = \sum_{i=1}^6 w_i (f_i - f'_i)^2 . \quad (6.25)$$

The function  $g(d_s(r, r'))$  is a converting function to ensure a proper influence of the shape distance on the total distance. In our system, it is defined as

$$g(d) = \begin{cases} 1 & d \geq 0.5 \\ 0.85 & 0.2 < d \leq 0.5 \\ 0.5 & d < 0.2 \end{cases} . \quad (6.26)$$

It is observed that when  $d_s(r, r') \geq 0.5$ , the two regions bear little resemblance. It is then not meaningful to distinguish the extent of similarity by  $d_s(r, r')$  because perceptually the two regions simply appear different. We thus set  $g(d) = 1$  for  $d$  greater than a threshold. When  $d_s(r, r')$  is very small, we intend to keep the influence of color and texture. Therefore  $g(d)$  is bounded away from zero. We set  $g(d)$  to be a piece-wise constant function instead of a smooth function for simplicity. Because rather simple shape features are used in our system, we emphasize color and texture more than shape. As can be seen from the definition of  $d(r, r')$ , the shape distance serves as a “bonus.” If two regions match very well in shape, their color and texture distance is attenuated by a smaller weight to provide the final distance.

## 6.6 System for biomedical image databases

The purpose of content-based indexing and searching for biomedical image databases is very different from that for picture libraries. Users of a general-purpose picture library are typically interested in images with similar object and color configurations at a global scale, while users of a biomedical image database are often interested in images with similar objects at the finest scale.

With this in mind, we have developed the Pathfinder system [131], a CBIR system for retrieving biomedical images with extremely high resolution. The system was initially designed for pathology slides [129]. It can be extended for other biomedical image databases and satellite image databases because of the ability to search for a region with desired objects.

We applied the Pathfinder system to a database of more than 70,000 pathology

image fragments. Promising results have been obtained and summarized in Chapter 7.



Figure 6.9: Automatic object segmentation of pathology images is an extremely difficult task. When different thresholds are provided, an edge detector either gives too many edges or too few edges for the object grouping. We avoid the precise object segmentation process by using our IRM “soft matching” metric.

Like the SIMPLIcity system, we use a region-based approach with the IRM metric. Shape matching is usually impossible without a highly accurate and robust object segmentation algorithm. However, automatic object segmentation of pathology images is an extremely difficult task, if at all possible. As shown in Figure 6.9, when different thresholds are provided, an edge detector [10] either gives too many edges or too few edges for the object grouping. As we have mentioned before, the advantage of using our novel IRM is that the “soft matching” process makes the metric robust to inaccurate segmentation. We avoid the precise object segmentation process by using our IRM “soft matching” metric.

Besides the usage of the IRM, the Pathfinder system is very different from the SIMPLIcity system for picture libraries. Specifically, the system has a different feature extraction component and a progressive image browsing component. We discuss these differences in the following sections.

### 6.6.1 Feature extraction

The feature extraction is performed on multiresolution image blocks (or segments) of the original images, rather than the original images.

In fact, we first partition the images and lower resolution versions of the same image into overlapping blocks. A user may submit an image patch or a sketch of a desired object to form a query. The system attempts to find image segments within the database to match with the object specified by the user.

In the biomedical domain, color information is often less important than other visual features such as texture and object shape. Most radiology images are only gray-scale. In our Pathfinder system, we turn off the color features in the matching phase when the images are gray-scale images.

### 6.6.2 Wavelet-based progressive transmission

Digital pathology and radiology images typically have very high resolution, making it difficult to display them in their entirety on the computer screen and inefficient to transmit over the network for educational purposes. Progressive zooming of pathology images is desirable despite the availability of inexpensive networking bandwidth. We have developed an efficient progressive image resolution refining system for on-line distribution of pathology image using wavelets. Details of the server design can be found in [129].

The system is practical for real-world applications, pre-processing and coding each 24-bit image of size  $2400 \times 3600$  within 40 seconds on a Pentium III PC. The transmission process is in real-time. Besides its exceptional speed, the algorithm has high flexibility. The server encodes the original pathology images without loss. Based on the image request from a client, the server dynamically generates and sends out the part of the image at the requested scale and quality requirement. The algorithm is expandable for medical image databases such as PACS.

## 6.7 Clustering for large databases

For very large image databases, we apply the tree-structured vector quantization (TSVQ) algorithm (Chapter 4) to progressively partition the feature vector space into cells. Every feature vector, which corresponds to an region of an image in the

	Memory	CPU Time	I/O Time	Storage
<b>Indexing</b>	$O(1)$	$O(N)$	$O(N)$	$O(N)$
<b>Clustering</b>	$O(N)$	$O(N \log_2(\frac{N}{C}))$	$O(N)$	$O(N)$
<b>Searching</b>	$O(\frac{N}{C} \log_2 \frac{N}{C} + C)$	$O(\log_2(\frac{N}{C}) + C \log_2 C)$	$O(1)$	–

Table 6.1: Performance of the system for an image database of  $N$  images.  $C$  is the maximum number of images in each leaf node.

database, lands in one of the cells.

In the retrieval process, the feature vector of a region is calculated and the cell it belongs to is decided using the tree-structured partition. Then, the distances between the feature vector and all the feature vectors in the database which are included in the cell are calculated. A certain number of images in the cell with minimum distances is retrieved. The searching process is significantly speeded up because we only need to calculate to which cell of the feature space the query image belongs. Besides, the computation needed for finding the correct cell is negligible compared to the calculation of the distances.

After the tree is designed, to retrieve similar images for a query image, we first locate the cell of the feature space to which the query image belongs. Then we compare the query image with all the training images in the cell and nearby cells to find the most similar ones in terms of small distance between their feature vectors.

To quantitatively compare the amount of computation needed for both algorithms, we assume there are totally  $N$  training images and the splitting of a node is stopped when the number of training images in the node is less than  $C$ . The space is thus divided into roughly  $\frac{N}{C}$  cells, which correspond to the approximately  $\frac{N}{C}$  leaf nodes in the tree. For a binary tree, the depth of the tree is then about  $\log \frac{N}{C}$  with 2 as base.

In order to locate the cell to which the feature vector of a query image belongs, the feature vector must be compared with  $2 \log \frac{N}{C}$  centroids of nodes. That is,  $2 \log \frac{N}{C}$  distances need to be computed. After the cell is located, another  $O(C)$  distances

are calculated to select the most similar images from the  $C$  training images in the cell. To present the final result, a quick sort taking  $O(C \log C)$  time must be performed. Hence, we need a total of  $C + 2 \log \frac{N}{C}$  distance calculations. However, for the traditional algorithms, a total of  $N$  distance calculations are necessary. Clearly,  $C + 2 \log \frac{N}{C}$  is usually much smaller than  $N$ . For instance, if  $\frac{N}{C} = 100$ , for a large  $N$ , the first value,  $\frac{N}{100} + 2 \log 100$ , is approximately one percent of  $N$ . Usually, we choose  $\frac{N}{C}$  larger than 100 for very large image databases. The details on the performance of the algorithms are provided in Table 6.1.

This clustering process is designed to significantly speed up the retrieval process. However, the retrieval accuracy is degraded when the dimensionality of the feature space is high. The process is therefore useful to applications when high precision and high speed are more critical than high recall. We are exploring more suitable alternatives to handle very large image databases.

## 6.8 Summary

In this chapter, we gave an overview of the SIMPLIcity system architecture and details of the concepts and methods used in the system, including the image segmentation process, the classification methods, and the IRM region-based similarity metric. For very large image databases, we use TSVQ to cluster the feature data.

# Chapter 7

## Evaluation

*Few things are harder to put up with than a good example.*

— Mark Twain (1835-1910)

### 7.1 Introduction

In this chapter, we further describe the experimental system we have developed, the SIMPLIcity system, using the concepts we proposed in previous chapters. Specifically, we present the data sets we used for the experiments, functions of the query interfaces, the characteristics of the IRM distance, the accuracy comparisons, the robustness to various changes, and the speed of indexing and searching.

### 7.2 Overview

The SIMPLIcity system has been implemented with a general-purpose image database including about 200,000 pictures, which are stored in JPEG format with size  $384 \times 256$  or  $256 \times 384$ . The Pathfinder system (described in Section 6.6) has been tested on a pathology image database with more than 70,000 image fragments, which are stored in raw format with size  $256 \times 256$ . Neither of two systems uses any textual information



in the matching process because we try to explore the possible advances of CBIR. In a real-world application, however, textual information is often used as a helpful addition to CBIR systems.

For each image, the features, locations, and areas of all its regions are stored. Images of different semantic classes are stored in separate databases. Because the EMD-based color histogram system [97] and the WBIIS system are the only other systems we have access to, we compare the accuracy of the SIMPLIcity system to these systems using the same COREL database. WBIIS had been compared with the IBM QBIC system and found to perform better [118] (also see Chapter 5). WBIIS has been incorporated into the current IBM QBIC system. It is difficult to design a fair comparison with existing region-based searching algorithms such as the Blobworld system which depends on additional information to be provided by the user during the process.

With the Web, on-line demonstration has become a popular direction in letting user evaluate CBIR systems. An on-line demonstration, with WBIIS, SIMPLIcity, and WIPE, is provided at URL: <http://WWW-DB.Stanford.EDU/IMAGE>. Readers are encouraged to compare the performance of SIMPLIcity with other systems. A list of on-line image retrieval demonstration websites can be found on our site.

## 7.3 Data sets

Several data sets have been used to test the accuracy and speed of the system and compare the system with existing systems and traditional indexing methods.

### 7.3.1 The COREL data set

The COREL data set, a collection of photographic stock images and clip art, is the most widely used standard data set for testing CBIR systems. Typically, researchers use between 1,000 and 30,000 images from the COREL data set. To validate the hypothesis of this dissertation, we used all the 200,000 COREL images available to us. For the purpose of simulating feature-based indexing under difficult situations,

Content	#	Content	#	Content	#	Content	#
Africa	100	autumn	200	Bhutan	100	Cal sea	100
Canada sea	100	Canada West	100	China	100	Croatia	100
Death Val	100	dogs	100	England	100	Galapago	100
Grand Canyon	100	Holland	100	ice frost	100	Ireland	100
Kyoto	100	lizard	100	Mexico	100	models	200
Monaco	100	mushroom	200	novascot	100	NY city	100
Ottawa	100	perennial	100	Peru	100	plants	100
Pyramids	100	Rome	100	royal grd	100	France	100
rural UK	100	sail fast	100	subsea	100	texture	500
Thailand	100	Turkey	100	US garden	100	vegetable	100
vineyard	100	wild Alaska	100	wild cats	100	wild cougar	100
wild eagle	100	wild goat	100	wild nests	100	sea animals	200
rare animals	100	young animals	100	Yosemite	100	bird	100
deer	100	lion	100	penguin	100	elephant	100
fish	100	wolf	200	gates	100	water ox	100
tiger	100	glacier	100	orchard	100	market	100
beach	400	church	300	animal	100	flower	500
mountain	100	cloud	200	people	100	India	100
indoor	100	button	100	cave	100	owl	100
Europe	200	boat	100	tourist	100	ski	100

Table 7.1: The contents of the first 10,000 images in the COREL image database according to the titles of the CDs.

we used JPEG-compressed images of relatively low resolution ( $256 \times 256$ ).

The COREL image database contains a wide variety of photographic images and clip art pictures. According to SIMPLicity, there are 3772 texture images in the database, about 6% of the total collection. It is difficult to summarize all the contents of the 200,000 images. Table 7.1 illustrates the contents of the first 10,000 (5%) images in the COREL image database.

### 7.3.2 Pathology data set

As we have mentioned before, the purpose of content-based indexing and searching for biomedical image databases is very different from that for picture libraries. We provide evaluation results on pathology images to illustrate the ideas of using block-based IRM to match regions for high-resolution images.

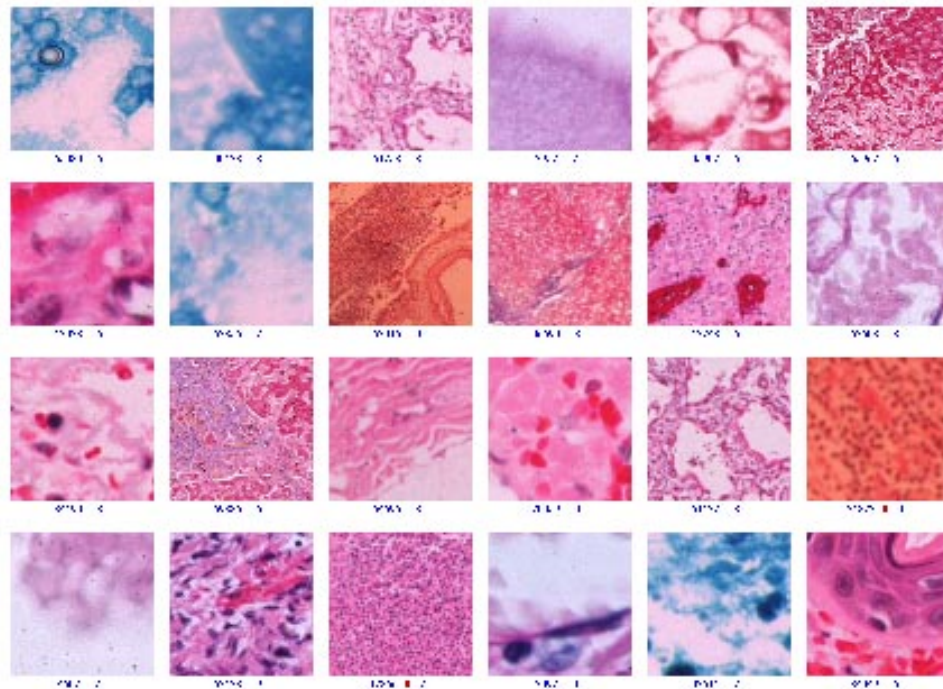


Figure 7.1: A random set of 24 image fragments from the pathology image database.

The pathology image database we are using were provided by Donald Regula, M.D., at the Stanford University Pathology Department. The database contains of more than 70,000 image segments, obtained through multiresolution partitioning of digitized pathology slides with extremely high resolution. Figure 7.1 shows a random set of 24 image fragments from the pathology image database.

## 7.4 Query interfaces

The current implementation of the SIMPLicity system provides several query interfaces: a CGI-based Web access interface, a JAVA-based drawing interface, and a CGI-based Web interface for submitting a query image of any format anywhere on the Internet.

### 7.4.1 Web access interface

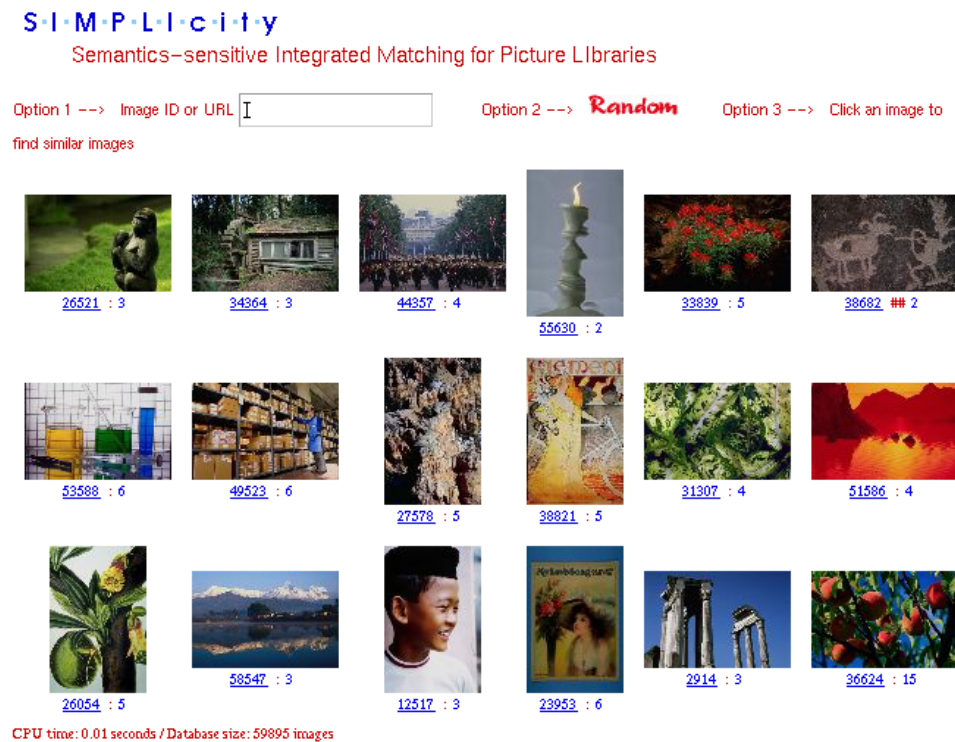


Figure 7.2: The Web access interface. A different random set of 18 images from the database is shown initially.

This interface (Figure 7.2) is written in CGI and is designed for accessing images in the database with a query image from the database. The user may select a random set of images from the database to start with and click on an image in the window to form a query. Or, the user may enter the ID of an image as the query.

If the user places the mouse on top of a thumbnail image shown in the window (without clicking on it), the thumbnail image will be automatically changed to its region segmentation and each region is painted with its representing color. This feature is important for partial region matching. The user may click in one of the regions in the segmentation map to notify the server that the object represented by this region is desired. The image server will then give higher weights to the regions surrounding the region chosen by the user. We will implement this partial-search function in the system in the near future.

### 7.4.2 JAVA drawing interface

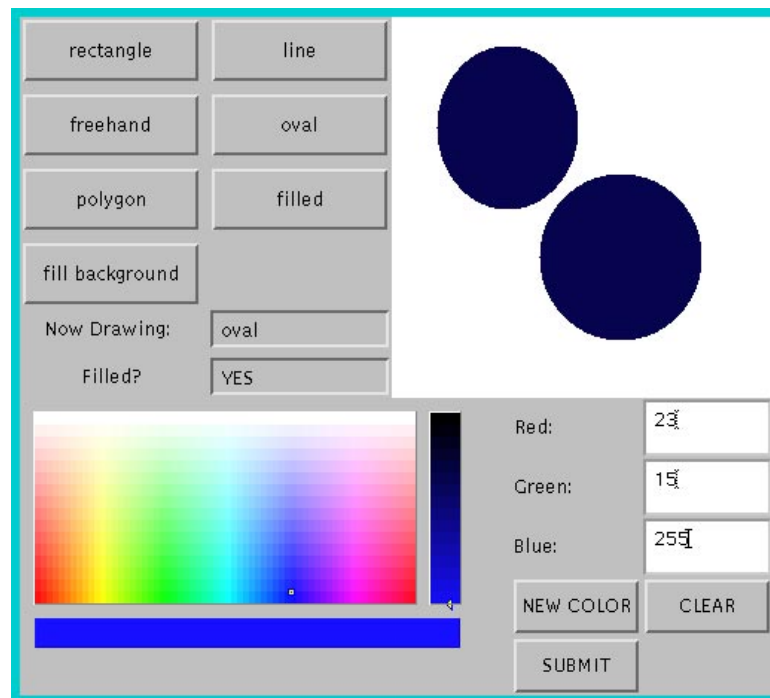


Figure 7.3: The JAVA drawing query interface allows users to draw sketch queries.

We have developed a JAVA-based drawing interface (Figure 7.3) for users to make free hand sketch queries<sup>1</sup>. We allow users to draw sketches, straight lines, polygons,

<sup>1</sup>Developed by Desmond Chan and Xin Wang at Stanford University.

rectangles, and eclipses. A 24-bit color palette is provided on the interface for users to choose a representing color for each region or line drawn. We are exploring ways to specify desired textures.

### 7.4.3 External query interface

**S-I-M-P-L-I-C-I-T-Y**  
Semantics-sensitive Integrated Matching for Picture Libraries

Option 1 --> Image ID or URL  Option 2 --> **Random** Option 3 --> Click an image to find similar images

CPU time: 2.02 seconds / Database size: 59895 images

Figure 7.4: The external query interface. The best 17 matches are presented for a query image selected by the user from the Stanford top-level Web page. The user enters the URL of the query image (shown in the upper-left corner, `http://www.stanford.edu/home/pics/h-quad.jpg`) to form a query.

We allow the user to submit any images on the Internet as a query image to the system by entering the URL of an image (Figure 7.4). Our system is capable of handling any image format from anywhere on the Internet and reachable by our server via the HTTP protocol. The image is downloaded and processed by our system on-the-fly. The high efficiency of our image segmentation and matching algorithms made this

feature possible<sup>2</sup>. To our knowledge, this feature of our system is unique in the sense that no other commercial or academic systems allow such queries.

#### 7.4.4 Progressive browsing

We have developed a Web-based user interface that allows the users to magnify any portion of the pathology images in different level of resolutions. The details of the wavelet-based progressive transmission server are provided in our paper [129]. We use Web interface and JAVA primarily because of the wide acceptance of the Internet and the Web in health-care environments. Figure 7.5 shows sample image query results with the HTML-based client user interface.

### 7.5 Characteristics of IRM

To study the characteristics of the IRM distance, we performed 100 random queries on our COREL photograph data set. We obtained 5.6 million IRM distances. Based on these distances, we estimated the distribution of the IRM distance. The empirical mean of the IRM is 44.30, with a 95% confidence interval of [44.28,44.32]. The standard deviation of the IRM is 21.07. Figure 7.6 shows the empirical probability distribution function and the empirical cumulative distribution function.

Based on this empirical distribution of the IRM, we may give more intuitive similarity distances to the end user than the distances themselves using the *similarity percentile*. As shown in the empirical cumulative distribution function, an IRM distance of 15 represents approximately 1% of the images in the database. We may notify the user that two images are considered to be very close when the IRM distance between the two images is less than 15. Likewise, we may advise the user that two images are likely to be far away in similarity when the IRM distance between the two images is greater than 50.

---

<sup>2</sup>It takes some other region-based CBIR system [11] approximately 8 minutes CPU time to segment an image.



Figure 7.5: Multiresolution progressive browsing of pathology slides of extremely high resolution. HTML-based interface shown. The magnification of the pathology images are shown on the query interface.



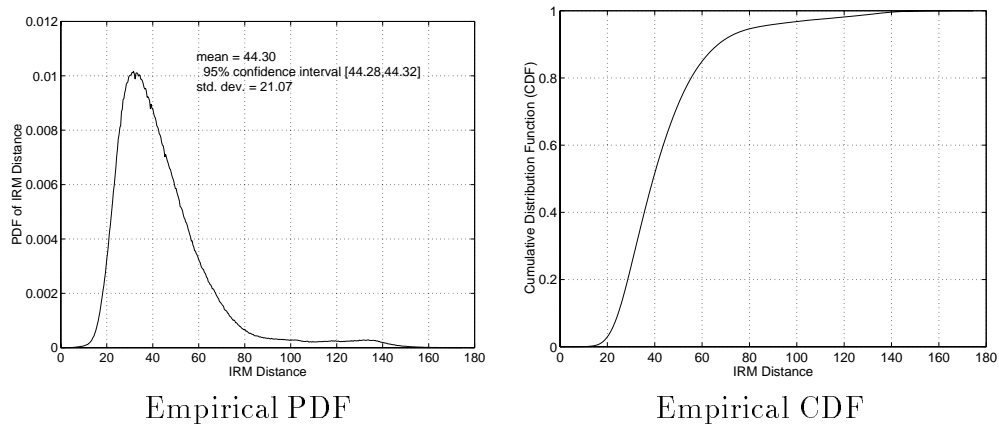


Figure 7.6: The empirical PDF and CDF of the IRM distance.

## 7.6 Accuracy

We evaluated the accuracy of the system in three ways. First, we used a 200,000-image COREL database to compare with existing systems such as EMD-based color histogram and WBIIS. Then, we designed systematic evaluation methods to judge the performance statistically. Finally, we used a database of pathology images to test the applicability of the system to other domains. The image classification performance of the system is evaluated and reported in Appendix A.

In Section 7.6.1, we show the accuracy of SIMPLIcity using the COREL database. We systematically compare SIMPLIcity with EMD-based color histogram and WBIIS in Section 7.6.2. The SIMPLIcity system has demonstrated much improved accuracy over the other systems. Performance on a biomedical image database is reported in Section 7.6.3.

### 7.6.1 Picture libraries

We compare the SIMPLIcity system with the WBIIS (Wavelet-Based Image Indexing and Searching) system [118] with the same image database. In this section, we show the comparison results using query examples. In the next section, we provide numerical evaluation results by systematically comparing several systems.

As WBIIS forms image signatures using wavelet coefficients in the lower frequency



*SIMPLIcity*



*WBIIS*

Figure 7.7: Comparison of SIMPLIcity and WBIIS. The query image is a landscape image on the upper-left corner of each block of images. SIMPLIcity retrieved 8 related images within the best 11 matches. WBIIS retrieved 7 related images.



Figure 7.8: Comparison of SIMPLIcity and WBIIS. The query image is a photo of food. SIMPLIcity retrieved 10 related images within the best 11 matches. WBIIS did not retrieve any related images.

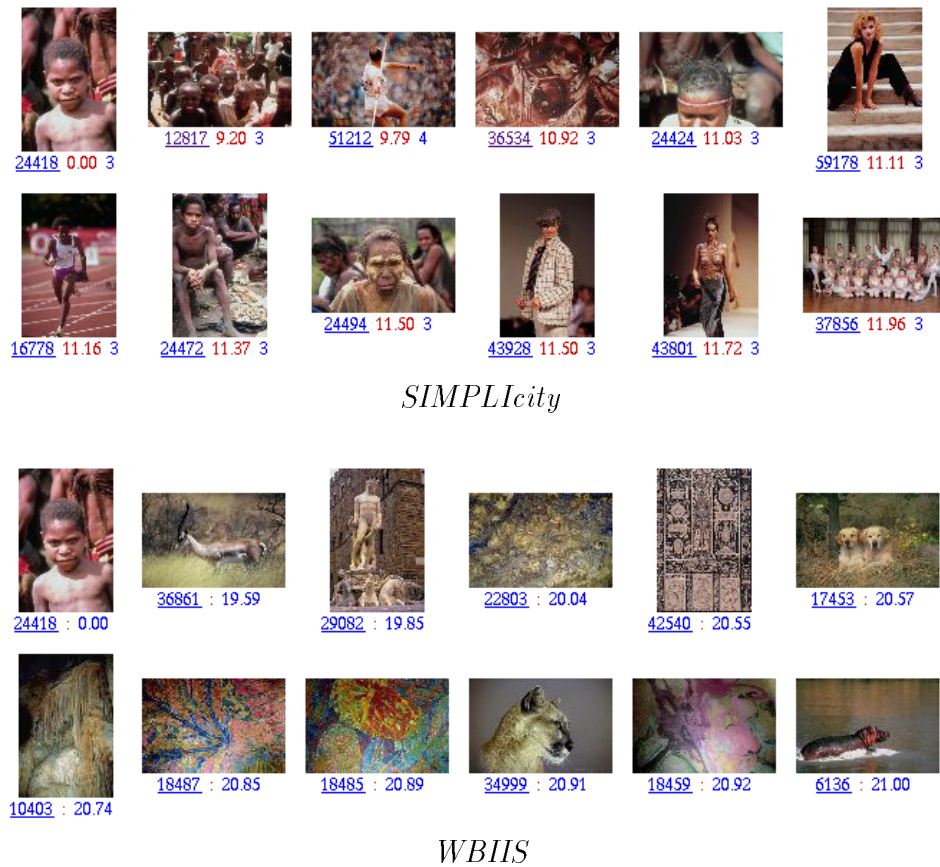
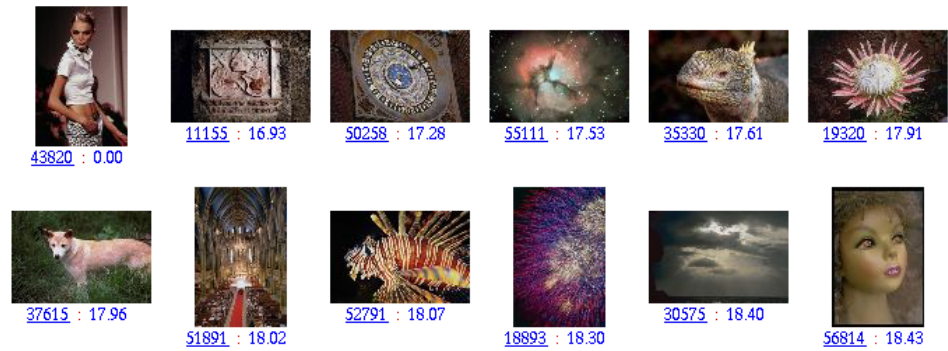


Figure 7.9: Comparison of SIMPLIcity and WBIIS. The query image is a portrait image that probably depicts life in Africa. SIMPLIcity retrieved 10 related images within the best 11 matches. WBIIS did not retrieve any related images.



*SIMPLIcity*



*WBIIS*

Figure 7.10: Comparison of SIMPLIcity and WBIIS. The query image is a portrait of a model. SIMPLIcity retrieved 7 related images within the best 11 matches. WBIIS retrieved only one related image.

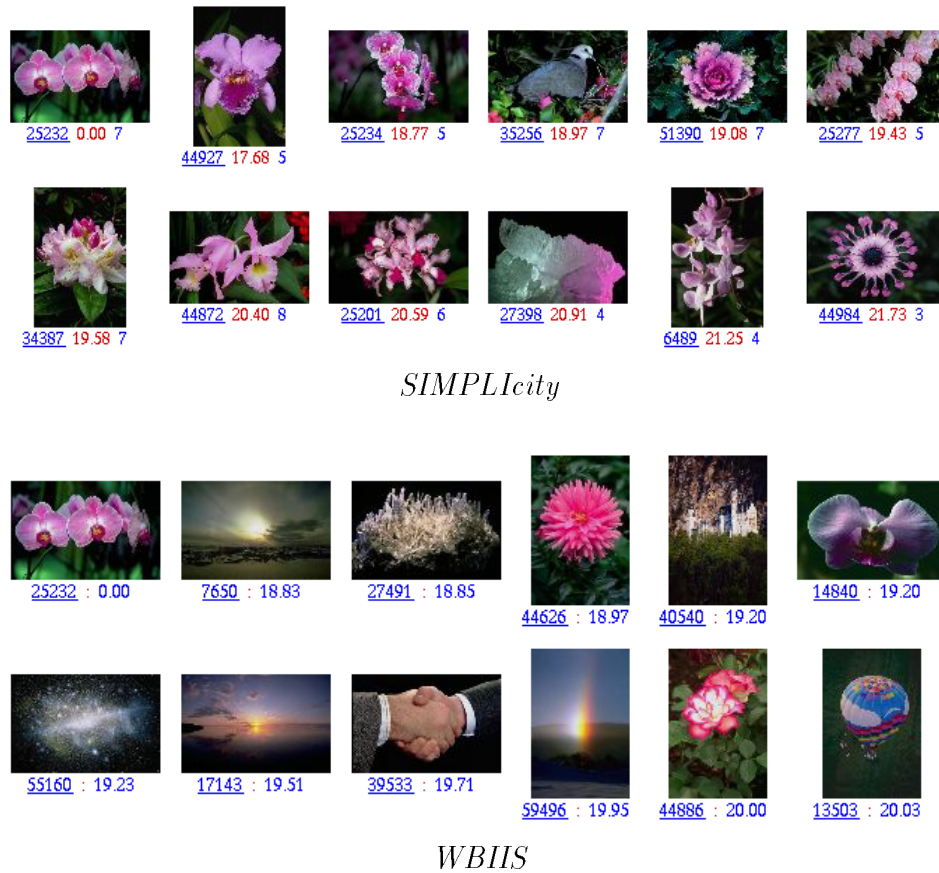


Figure 7.11: Comparison of SIMPLicity and WBIIS. The query image is a photo of flowers. SIMPLicity retrieved 10 related images within the best 11 matches. WBIIS retrieved 4 related images.

bands, it performs well with relatively smooth images, such as most landscape images. For images with details crucial to semantics, such as pictures with people, the performance of WBIIS degrades. In general, SIMPLIcity performs as well as WBIIS for smooth landscape images. One example is shown in Figure 7.7. The query image is the image at the upper-left corner. The underlined numbers below the pictures are the ID numbers of the images in the database. The other two numbers are the value of the similarity measure between the query image and the matched image, and the number of regions in the image. To view the images better or to see more matched images, users can visit the demonstration web site and use the query image ID to repeat the retrieval.

SIMPLIcity also performs well for images composed of fine details. Retrieval results with a photo of a hamburger as the query are shown in Figure 7.8. The SIMPLIcity system retrieves 10 images with food out of the first 11 matched images. The WBIIS system, however, does not retrieve any image with food in the first 11 matches. The top match made by SIMPLIcity is also a photo of hamburger, which is perceptually very close to the query image. WBIIS misses this image because the query image contains important fine details, which are smoothed out by the multi-level wavelet transform in the system. The smoothing also causes a textured image (the third match) to be matched. Such errors of WBIIS are observed with other image queries when fine details are important in distinguishing image semantics. The SIMPLIcity system, however, prevents images of different classes to be matched by classifying them before searching.

Another three query examples are compared in Figure 7.9, 7.10, and 7.11. The query images in Figure 7.9 and 7.10 are difficult to match because objects in the images are not distinctive from the background. Moreover, the color contrast for both images is small. It can be seen that the SIMPLIcity system achieves much better retrieval. For the query in Figure 7.9, only the third matched image is not a picture of a person. A few images, the 1st, 4th, 7th, and 8th matches, depict a similar topic as well, probably about life in Africa. The query in Figure 7.11 also shows the advantages of SIMPLIcity. The system finds photos of similar flowers with different sizes and orientations. Only the 9th match does not have flowers in it.

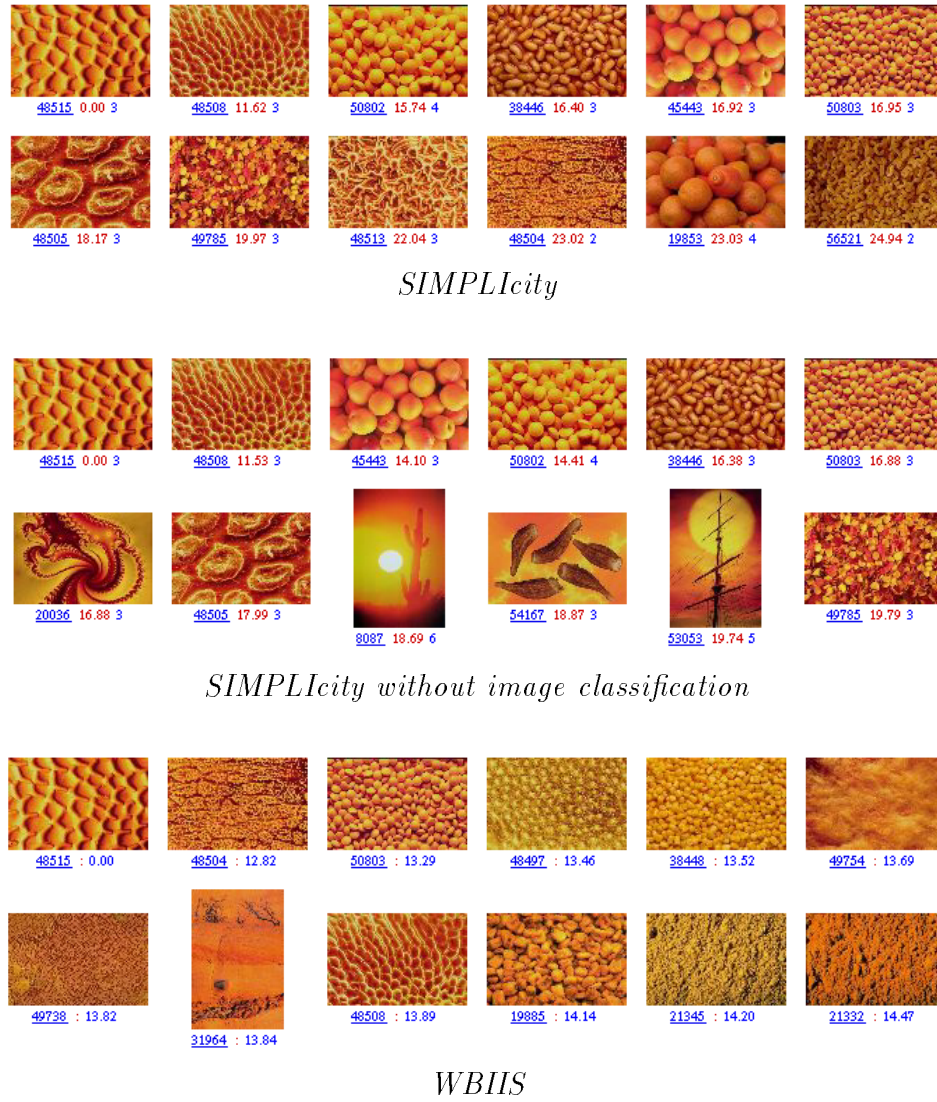


Figure 7.12: SIMPLIcity gives better results than the same system without the classification component. The query image is a textured image.



For textured images, SIMPLIcity and WBIIS often perform equally well. However, SIMPLIcity captures high frequency texture information better. An example of textured image search is shown in Figure 7.12. The granular surface in the query image is matched more accurately by the SIMPLIcity system. We performed another test on this query using SIMPLIcity system without the image classification component. As shown in Figure 7.12, the degraded system found several non-textured pictures (e.g., sunset scenes) for this textured query picture.

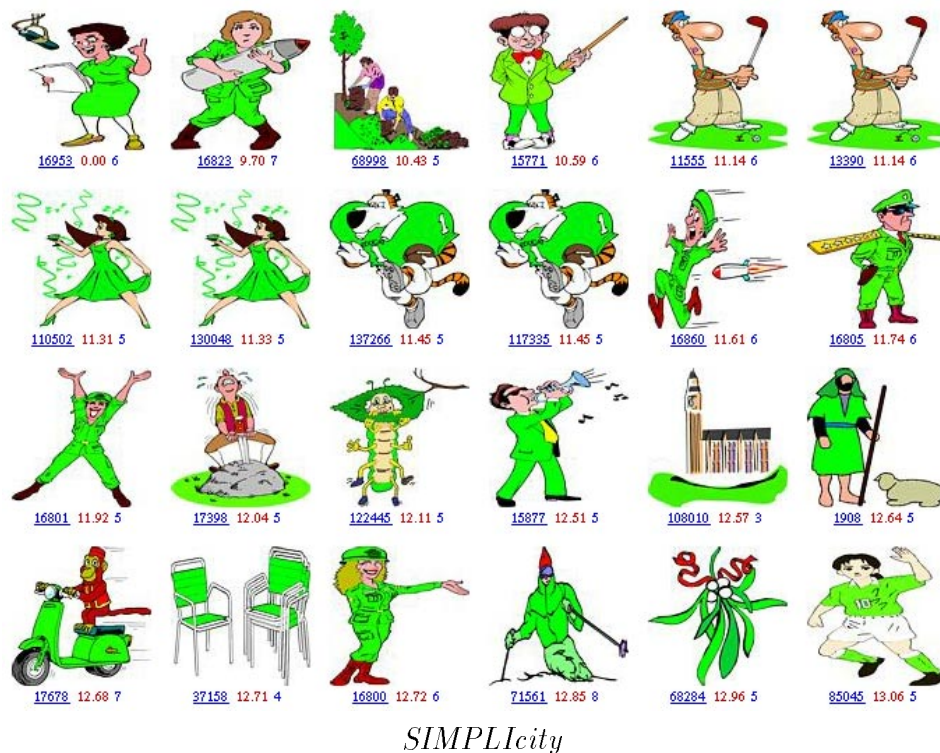


Figure 7.13: SIMPLIcity does not mix clip art pictures with photographs. A graph-photograph classification method using image segmentation and statistical hypothesis testing is used. The query image is a clip art picture.

Existing CBIR systems do not perform well when the image databases contain both photographs and graphs. Graphs, such as clip art pictures and image maps, appear frequently on the Web. The semantics of clip art pictures are typically more abstract and significantly different from photos with similar low-level visual features,

such as the color histogram. For image maps on the Web, an indexing method based on Optical Character Recognition (OCR) may be more efficient than CBIR systems based on visual features. Wang et al. used wavelets and OCR in retrieving medical images with patient identification information [127]. SIMPLIcity classifies picture libraries into graphs and photographs [71] using image segmentation and statistical hypothesis testing before the feature indexing step. Figure 7.13 shows the result of a clip art query. All the best 23 matches of this 200,000-picture database are clip art pictures, many with similar semantics.

### 7.6.2 Systematic evaluation

We performed two sets of systematic evaluation tests to provide objective comparison of SIMPLIcity with various other systems.

#### Performance on image queries

To provide numerical results, we tested 27 sample images chosen randomly from 9 categories, each containing 3 of the images. Image matching is performed on the COREL database of 200,000 images. A retrieved image is considered a match if it belongs to the same category of the query image. The categories of images tested are listed in Table 7.2. Most categories simply include images containing the specified objects. Images in the “sports and public events” class contain people in a game or public event, such as a festival. Portraits are not included in this category. The “landscape with buildings” class refers to outdoor scenes featuring man-made constructions such as buildings and sculptures. The “beach” class refers to scenery at coasts or river banks. For the “portrait” class, an image has to show people as the main feature. A scene with human beings as a minor part is not included.

Precision was computed for both SIMPLIcity and WBIIS. Recall was not calculated because the database is large and it is difficult to estimate the total number of images in one category, even approximately. As we have mentioned in Chapter 2, the “relevance” in the definitions of *recall* depends on the readers’ *point-of-view*. For the database of 60,000 photographs, we have to manually read all the images for each

ID	Category Name
1	Sports and public events
2	Beach
3	Food
4	Landscape with buildings
5	Portrait
6	Horses
7	Tools and toys
8	Flowers
9	Vehicle

Table 7.2: COREL categories of images tested for comparing with WBIIS.

query to determine the number of relevant images to the given query. In the future, we will develop a large-scale sharable test database to evaluate the recall.

To account for the ranks of matched images, the average of the precision values within  $k$  retrieved images,  $k = 1, \dots, 100$ , is computed, that is,

$$\bar{p} = \frac{1}{100} \sum_{k=1}^{100} \frac{n_k}{k},$$

$n_k = \#$  of matches in the first  $k$  retrieved images .

This average precision is called the “weighted precision”<sup>3</sup> because it is equivalent to a weighted percentage of matched images with a larger weight assigned to an image retrieved at a higher rank. For instance, a relevant image appearing earlier in the list of retrieved images would enhance the weighted precision more significantly than if it appears later in the list.

For each of the 9 image categories, the average precision and weighted precision based on the 3 sample images are plotted in Fig. 7.14. The image category identification number is indicated in Table 7.2. Except for the tools and toys category, in which case the two systems perform about equally well, SIMPLicity has achieved better results than WBIIS measured in both ways. For the two categories of landscape with buildings and vehicle, the difference between the two systems is quite significant.

---

<sup>3</sup>The weighted precision has been used in information retrieval.

On average, the precision and the weighted precision of SIMPLIcity are higher than those of WBIIS by 0.227 and 0.273 respectively. In another perspective, SIMPLIcity retrieves on average twice as many relevant images than WBIIS does.

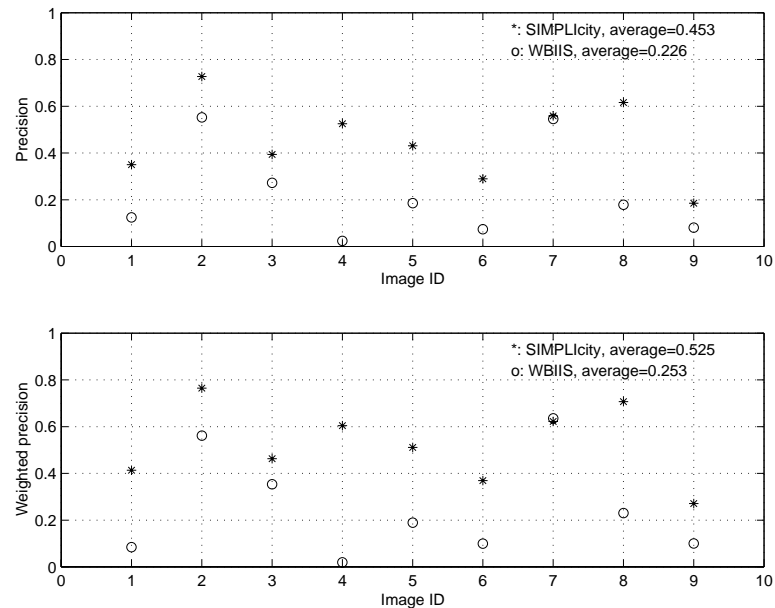


Figure 7.14: Comparison of SIMPLIcity and WBIIS: average precision and weighted precision of 9 image categories.

### Performance on image categorization

The SIMPLIcity system was also evaluated based on a subset of the COREL database, formed by 10 image categories (shown in Table 7.3), each containing 100 pictures. These image categories were randomly selected from about 600 photograph categories of the COREL image database. Within this database, it is known whether any two images are of the same category. In particular, a retrieved image is considered a match if and only if it is in the same category as the query. This assumption is reasonable since the 10 categories were chosen so that each depicts a distinct semantic topic. Every image in the sub-database was tested as a query, and the retrieval ranks of all the rest images were recorded. Three statistics were computed for each query:

1. The precision within the first 100 retrieved images

ID	Category Name
1	Africa people and villages
2	Beach
3	Buildings
4	Buses
5	Dinosaurs
6	Elephants
7	Flowers
8	Horses
9	Mountains and glaciers
10	Food

Table 7.3: COREL categories of images tested for comparing with color histogram.

Category	Average $p$	Average $r$	Average $\sigma$
1	0.475	178.2	171.9
2	0.325	242.1	180.0
3	0.330	261.8	231.4
4	0.363	260.7	223.4
5	0.981	49.7	29.2
6	0.400	197.7	170.7
7	0.402	298.4	254.9
8	0.719	92.5	81.5
9	0.342	230.4	185.8
10	0.340	271.7	205.8

Table 7.4: The performance of SIMPLIcity (with an average of 4.3 regions per image) on categorizing picture libraries. The average performance for each image category evaluated by precision  $p$ , the mean rank of matched images  $r$ , and the standard deviation of the ranks of matched images  $\sigma$ .

Category	Average $p$	Average $r$	Average $\sigma$
1	0.288	312.9	252.9
2	0.286	280.0	225.8
3	0.233	332.4	270.1
4	0.267	283.7	259.1
5	0.914	54.7	225.2
6	0.384	187.7	214.0
7	0.416	235.3	236.0
8	0.386	278.4	266.5
9	0.218	324.6	265.6
10	0.207	427.3	346.1

Table 7.5: The performance of the EMD-based color histogram approach (with an average of 42.6 filled color bins) on categorizing picture libraries. The average performance for each image category evaluated by precision  $p$ , the mean rank of matched images  $r$ , and the standard deviation of the ranks of matched images  $\sigma$ .

2. The mean rank of all the matched images
3. The standard deviation of the ranks of matched images

The recall is identical to the precision in this special case because both the number of retrieved items and the number of relevant items equal to 100. The average performance for each image category in terms of the three statistics is listed in Table 7.4, where  $p$  denotes precision,  $r$  denotes the mean rank of matched images, and  $\sigma$  denotes the standard deviation of the ranks of matched images. For a system that ranks images randomly, the average  $p$  is about 0.1, and the average  $r$  is about 500. An ideal CBIR system should demonstrate an average  $p$  of 1 and an average  $r$  of 50.

Similar evaluation tests were carried out for the state-of-the-art EMD-based color histogram match. We used LUV color space and a matching metric similar to the EMD described in [97] to extract color histogram features and match in the categorized image database. Two different color bin sizes, with an average of 13.1 and 42.6 filled color bins per image, were evaluated. We call the one with less filled color bins the Color Histogram 1 system and the other the Color Histogram 2 system. Tables 7.5 and 7.6 show the performance. Figure 7.15 shows the performance when compared to

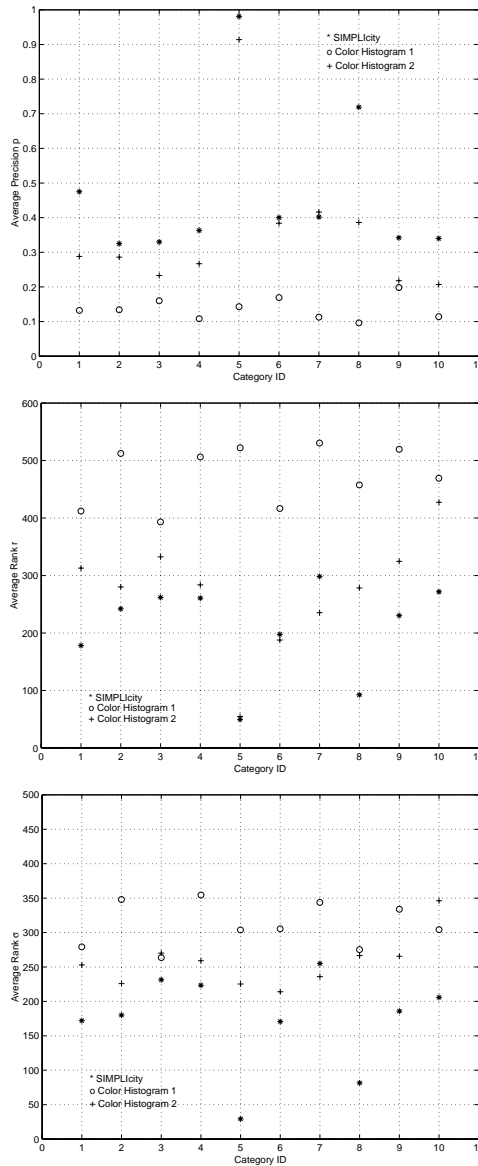


Figure 7.15: Comparing SIMPLicity with color histogram methods on average precision  $p$ , average rank of matched images  $r$ , and the standard deviation of the ranks of matched images  $\sigma$ . The lower numbers indicate better results for the last two plots (i.e., the  $r$  plot and the  $\sigma$  plot). Color Histogram 1 gives an average of 13.1 filled color bins per image, while Color Histogram 2 gives an average of 42.6 filled color bins per image. SIMPLicity partitions an image into an average of only 4.3 regions.

Category	Average $p$	Average $r$	Average $\sigma$
1	0.132	412.3	279.0
2	0.134	512.3	347.9
3	0.160	393.4	263.5
4	0.108	506.1	354.6
5	0.143	522.1	303.8
6	0.169	416.5	305.4
7	0.113	530.4	343.8
8	0.096	457.5	275.3
9	0.198	519.7	333.9
10	0.114	469.2	304.1

Table 7.6: The performance of the EMD-based color histogram approach (with an average of 13.1 filled color bins) on categorizing picture libraries. The average performance for each image category evaluated by precision  $p$ , the mean rank of matched images  $r$ , and the standard deviation of the ranks of matched images  $\sigma$ .

the SIMPLIcity system. Clearly, both of the two color histogram-based matching systems perform much worse than the SIMPLIcity region-based CBIR system in almost all image categories. The performance of the Color Histogram 2 system is better than that of the Color Histogram 1 system due to more detailed color separation obtained with more filled bins. However, the Color Histogram 2 system is so slow that it is impossible to obtain matches on databases with more than 50,000 images. SIMPLIcity runs at about twice the speed of the faster Color Histogram 1 system and still gives much better searching accuracy than the slower Color Histogram 2 system.

### 7.6.3 Biomedical image databases

The evaluation for biomedical images is more difficult than the evaluation for general-purpose images. Ideally, the system must be incorporated into a real-world PACS system and evaluate with pathologists or radiologists in realistic settings. Moreover, we need to measure the user satisfaction in addition to the standard precision and recall. In the medical domain, healthcare practitioners are typically busy during work. They are unlikely to use a CBIR system unless the system interaction is intuitive, stable, and swift. We are currently working closely with the Radiology Department at the



University of California at San Francisco (UCSF) and the SHINE/eSkolar project at Stanford University to integrate our search engine in their PACS image database management systems.

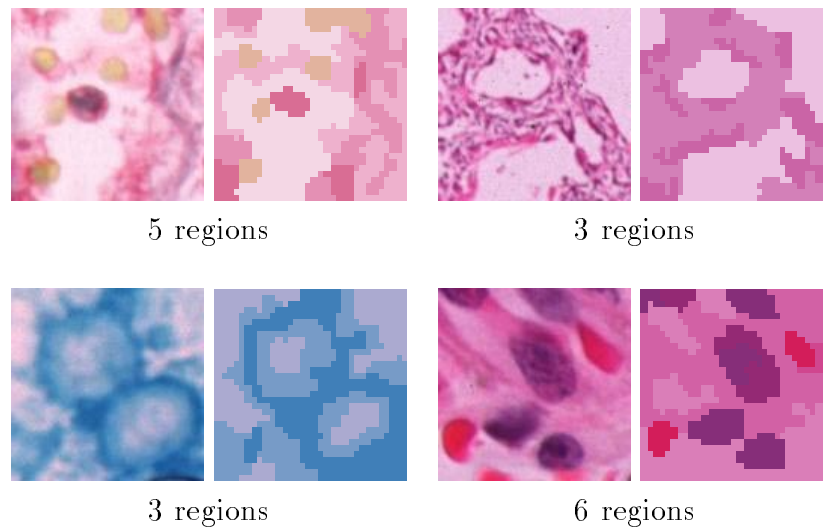


Figure 7.16: Segmentation results obtained using an algorithm based on k-means. A region is defined as a collection of pixels.

The system is tolerant to imprecise image segmentation. Figure 7.16 shows the results obtained using our fast segmentation algorithm based on the k-means algorithm. Based on our approximate image segmentation, the system is able to perform region-based matching incorporating intensity, texture, location, and shape features.

Figure 7.17 shows a sample query result. The user supplied a query with a round-shaped cell. The Pathfinder system successfully found images across different resolution, each with one or more round cells and similar visual characteristics, and ranked them according to their visual similarity to the query image. Figure 7.18 shows the results of two hand-drawn sketch queries. Again, the system found visually similar images.

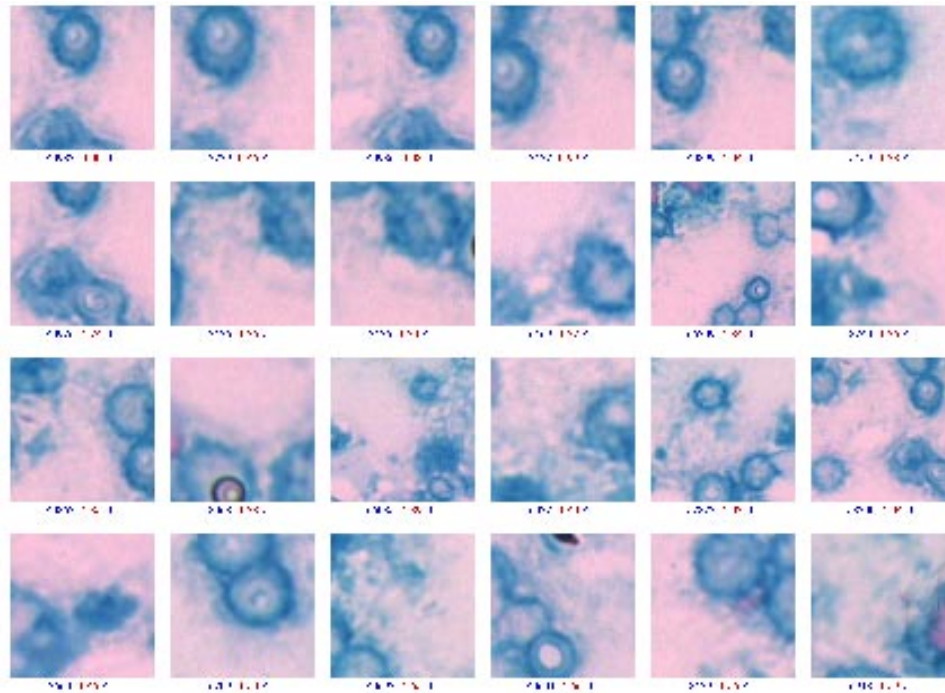


Figure 7.17: A sample query result. The first image is the query.

## 7.7 Robustness

We have performed extensive experiments on the robustness of the system. Figures 7.19, 7.20, 7.21, and 7.22 summarize the results. The graphs in the first row show the changes in ranking of the target image as we increase the significance of image alterations. The graphs in the second row show the changes in IRM distance between the altered image and the target image, as we increase the significance of image alterations.

The system is exceptionally robust to image alterations such as intensity variation, sharpness variation, intentional color distortions, other intentional distortions, cropping, shifting, and rotation. On average, the system is robust to approximately 10% brightening, 8% darkening, blurring with a  $15 \times 15$  Gaussian filter, 70% sharpening, 20% more saturation, 10% less saturation, random spread by 30 pixels, and pixelization by 25 pixels.

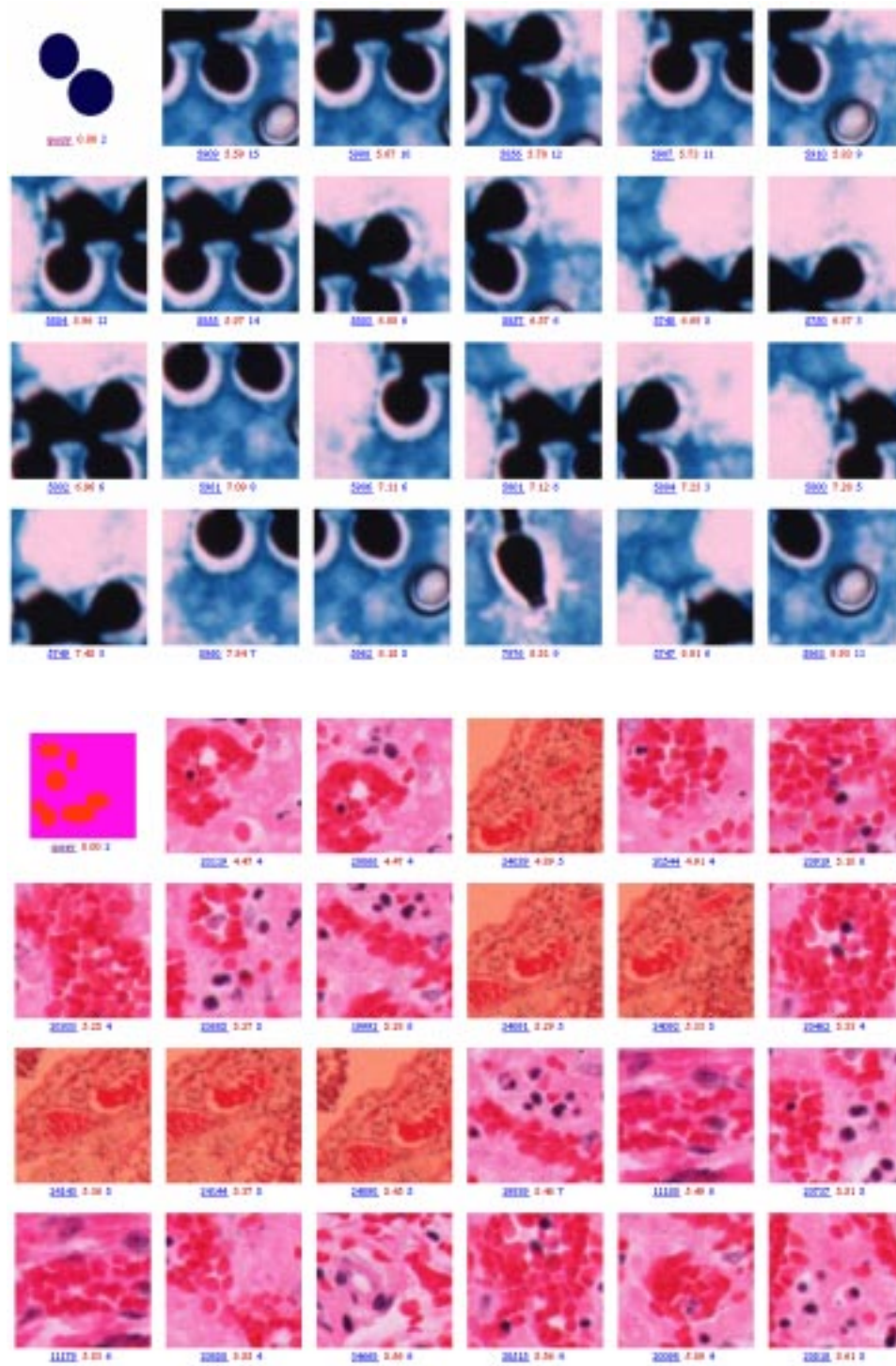


Figure 7.18: The results of hand-drawn sketch queries.

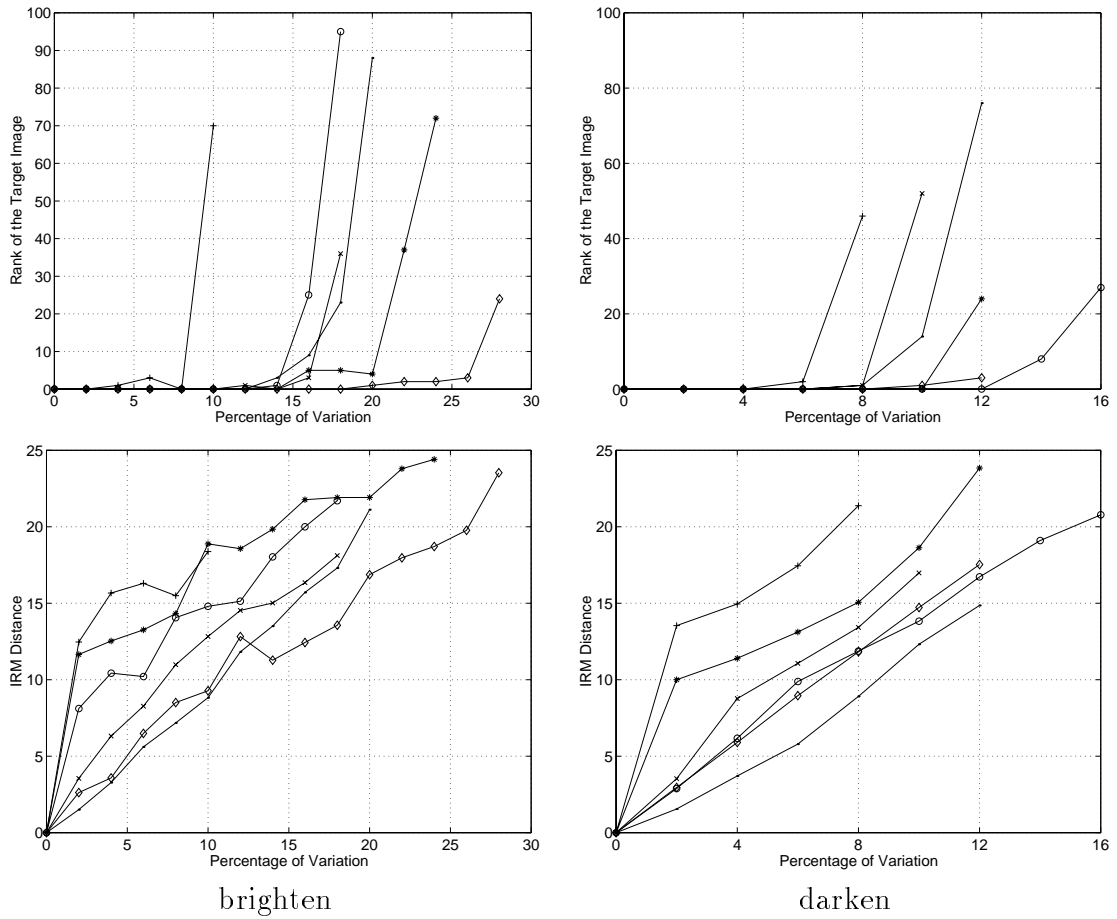


Figure 7.19: The robustness of the system to intensity alterations. 6 images were randomly selected from the database. Each curve represents the robustness on one of the 6 images.

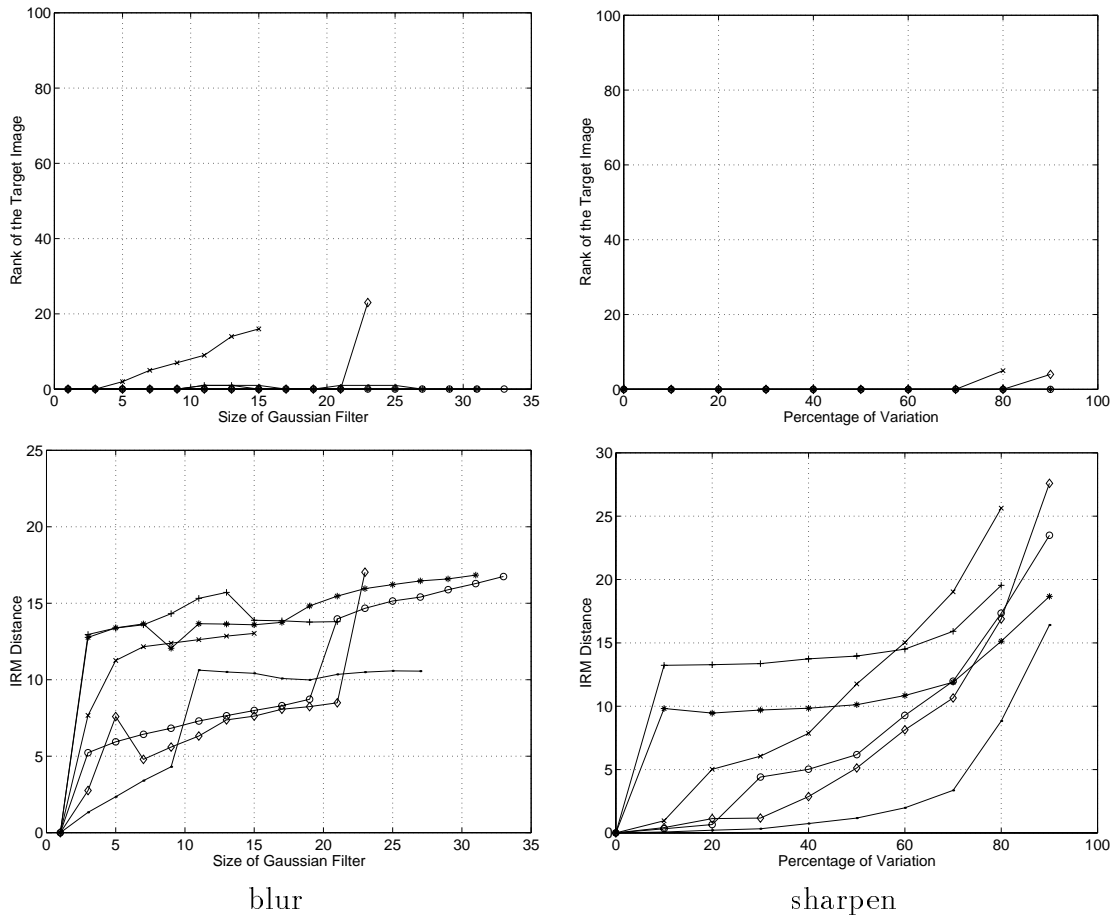


Figure 7.20: The robustness of the system to sharpness alterations. 6 images were randomly selected from the database. Each curve represents the robustness on one of the 6 images.

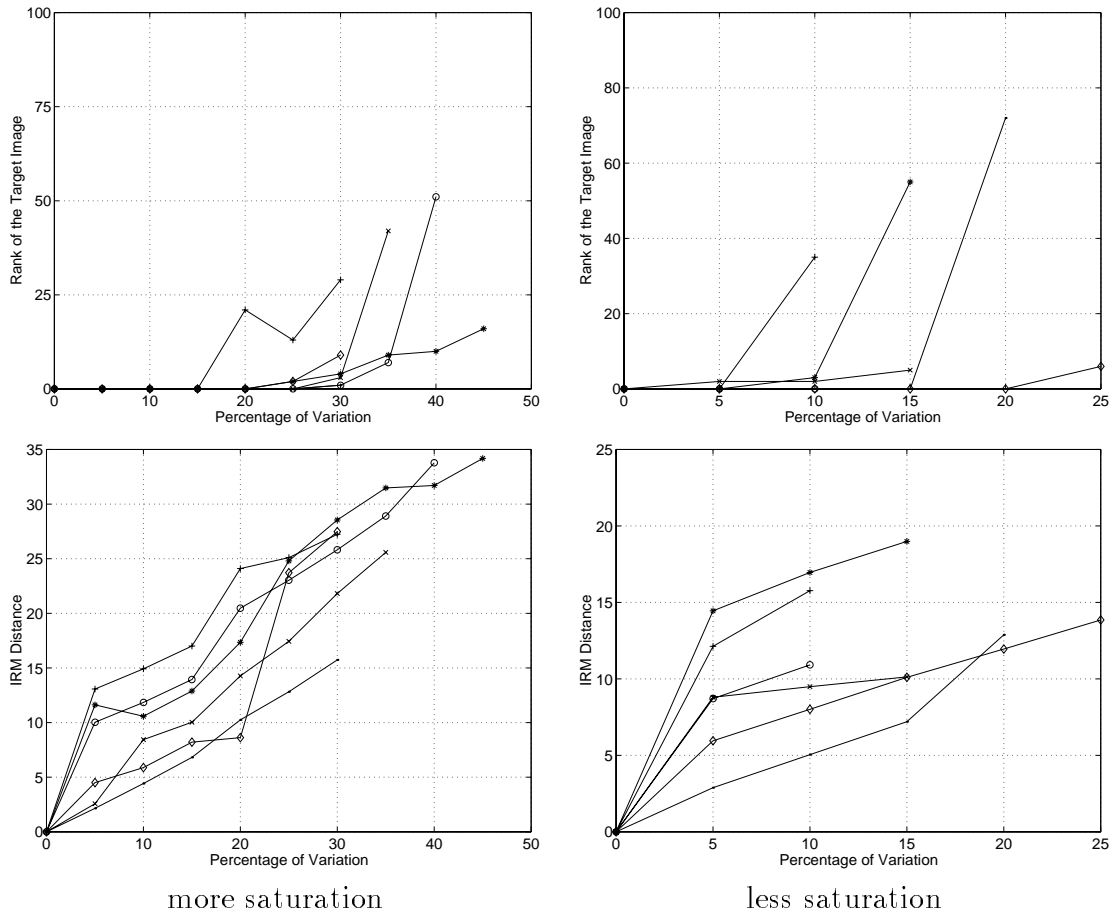


Figure 7.21: The robustness of the system to color alterations. 6 images were randomly selected from the database. Each curve represents the robustness on one of the 6 images.

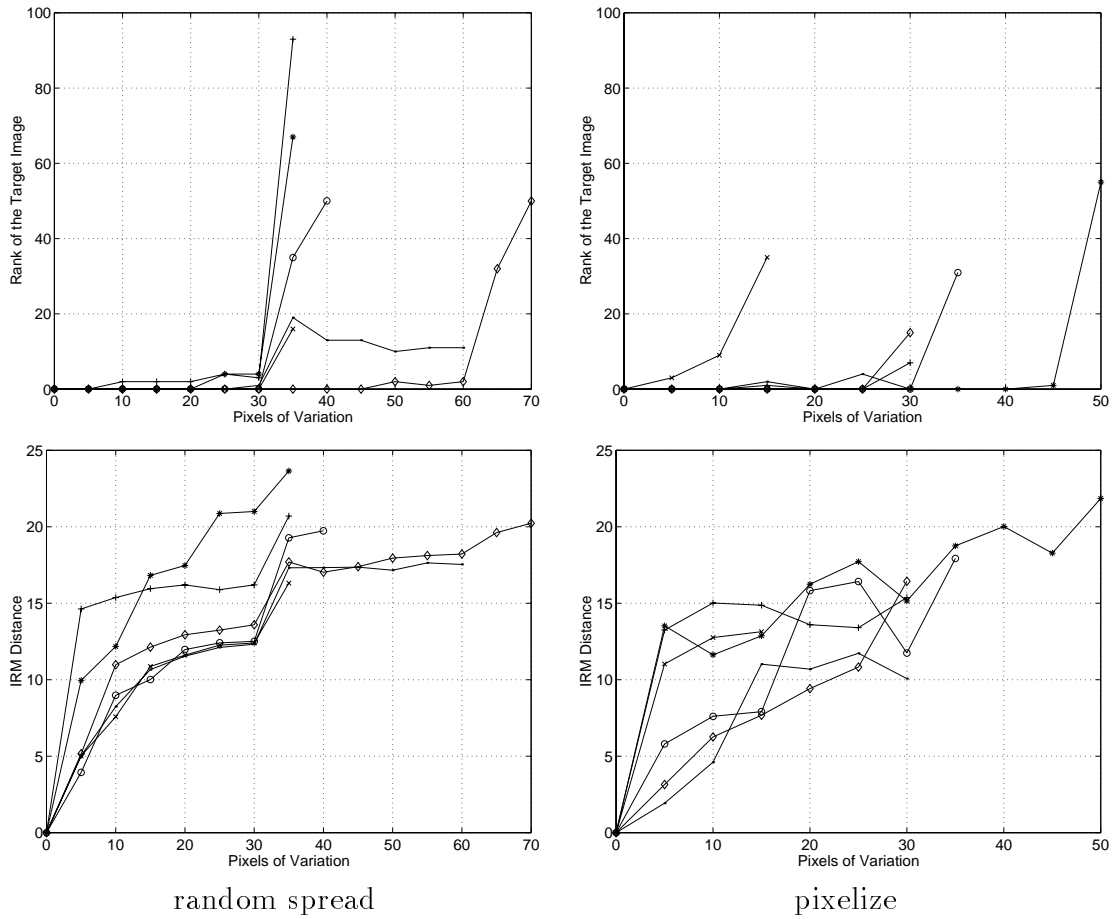


Figure 7.22: The robustness of the system to other image alterations. 6 images were randomly selected from the database. Each curve represents the robustness on one of the 6 images.

These features are important to biomedical image databases because usually visual features of the query image are not identical to the visual features of those semantically-relevant images in the database because of problems such as occlusion, difference in intensity, and difference in focus.

In the future, we will evaluate the robustness of the EMD-based color histogram systems and the color layout systems (e.g., WBIIS). We expect the color histogram systems to be sensitive to intensity variation, color distortions, and cropping. Color layout indexing is not robust to shifting, cropping, scaling, and rotation [118].

### 7.7.1 Intensity variation



Figure 7.23: The robustness of the system to image intensity changes.



The system seems robust to intensity changes. In Figure 7.23, a 10% brightened or a 6% darkened version of a query image can be used to find the original image as the best match.

In medicine, lesions differ by intensity from one person to another. In order to match images from a database of images, a similarity measure must be insensitive to slight intensity variations.

### 7.7.2 Sharpness variation

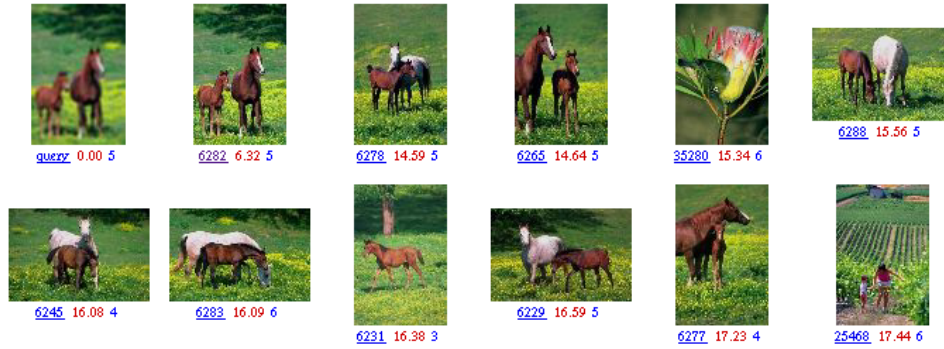
A blurred or a sharpened version of an image can be used to find the original image. Figure 7.24 shows two examples using Gaussian blurring filters and one example using a sharpening filter. This feature is important for handling hand-sketch queries because typical hand-sketch user interfaces provide blurry query images.

This feature is crucial in medicine. Radiology images often appear blurry due to the limitations in medical imaging technology. The sharpness differs from one machine to another. It is necessary for an image matching program to be robust to sharpness variations.

### 7.7.3 Color distortions

The SIMPLIcity system can tolerate many intentional color distortions. Figure 7.25 shows two examples. In the first example, the saturation of the targeted image is increased by 20%. In the second example, the saturation of the targeted image is decreased by 20%. In both cases, the system is able to locate the original image as the best match.

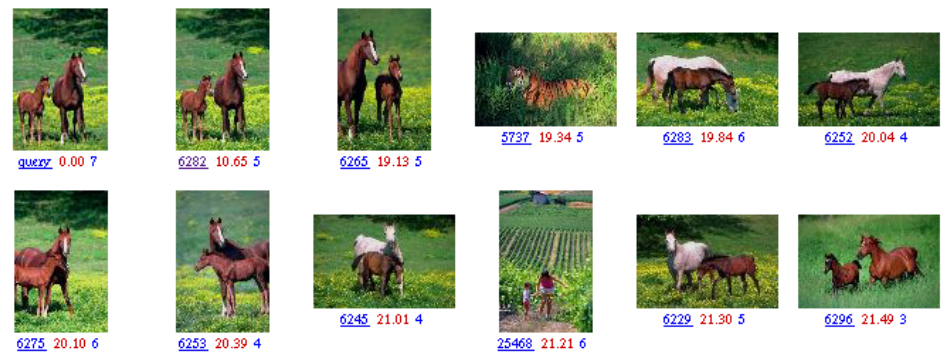
This feature is important in matching color biomedical images. In pathology, images of the same type of specimens often appear in different color saturation. Moreover, different dye preparation results in different color in the staining process.



Blur with a  $11 \times 11$  Gaussian filter



Blur with a  $17 \times 17$  Gaussian filter



Sharpen by 70%

Figure 7.24: The robustness of the system to sharpness variations.

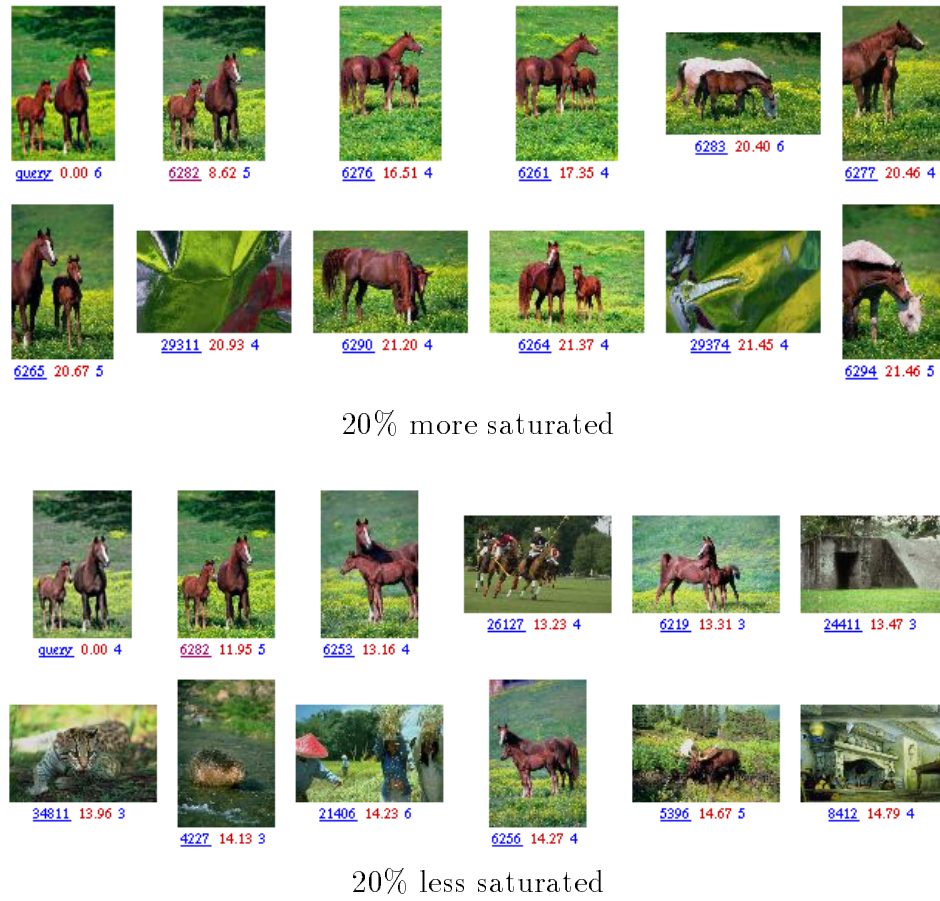


Figure 7.25: The robustness of the system to intentional color distortions.



Random spread 10 pixels



Pixelize at 20 pixels

Figure 7.26: The robustness of the system to two other intentional distortions.

### 7.7.4 Other intentional distortions

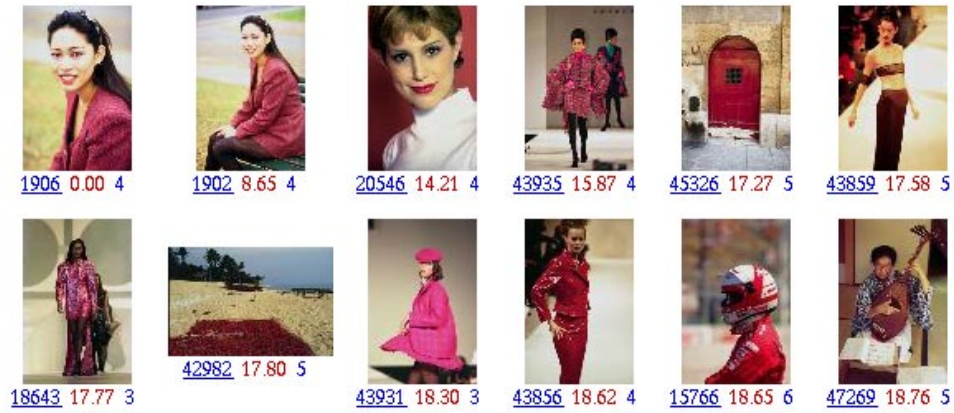
The SIMPLIcity system can tolerate many intentional image shape distortions. Figure 7.26 shows two examples. In the first example, the pixels in the query image are randomly displaced to up to 10 pixels away. The second example is a pixelized (at 20 pixels) query image. The system is able to locate the original image as the best match in both cases.

### 7.7.5 Cropping and scaling

To show the robustness of the SIMPLIcity system to cropping and scaling, querying examples are provided in Figure 7.27. One query image is a cropped and scaled version of the other. Using either of them as query, SIMPLIcity retrieves the other one as the top match. Retrieval results based on both of the queries are good. However, the retrieval performed by WBIIS using one of the images misses the other one. The performance of SIMPLIcity is better because the IRM distance is computed using area percentage as region significance measures. The SIMPLIcity system is robust when the cropped image differs slightly in area percentage of regions. On the other hand, WBIIS is not robust to image cropping because the cropped image differs very much in the layout from the original image.

### 7.7.6 Shifting

To test the robustness to shifting, we shifted two example images and used the shifted images as query images. Results are shown in Figure 7.28. The original images are both retrieved as the top match. In both cases, SIMPLIcity also finds many other semantically related images. This is expected since the shifted images are segmented into regions nearly the same as those of the original images. In general, if shifting does not affect region segmentation significantly, the system will be able to retrieve the original images with a high rank.



*50% cropping with SIMPLIcity*



*Inverse 50% cropping with SIMPLIcity*



*Inverse 50% cropping with WBIIS*

Figure 7.27: The robustness of the system to image cropping and scaling.



Horizontal shifting by 15%



Diagonal shifting by 22%

Figure 7.28: The robustness of the system to image shifting.

### 7.7.7 Rotation

Another example is provided in Figure 7.29 to show the effect of rotation. SIMPLIcity retrieves the original image as the top match. All the other images matched are also horse pictures. For an image without strong orientational texture, such as the query image in Figure 7.29, its rotation will be segmented into regions with similar features. Therefore, SIMPLIcity will be able to match images similar to those retrieved by the original image.

## 7.8 Speed

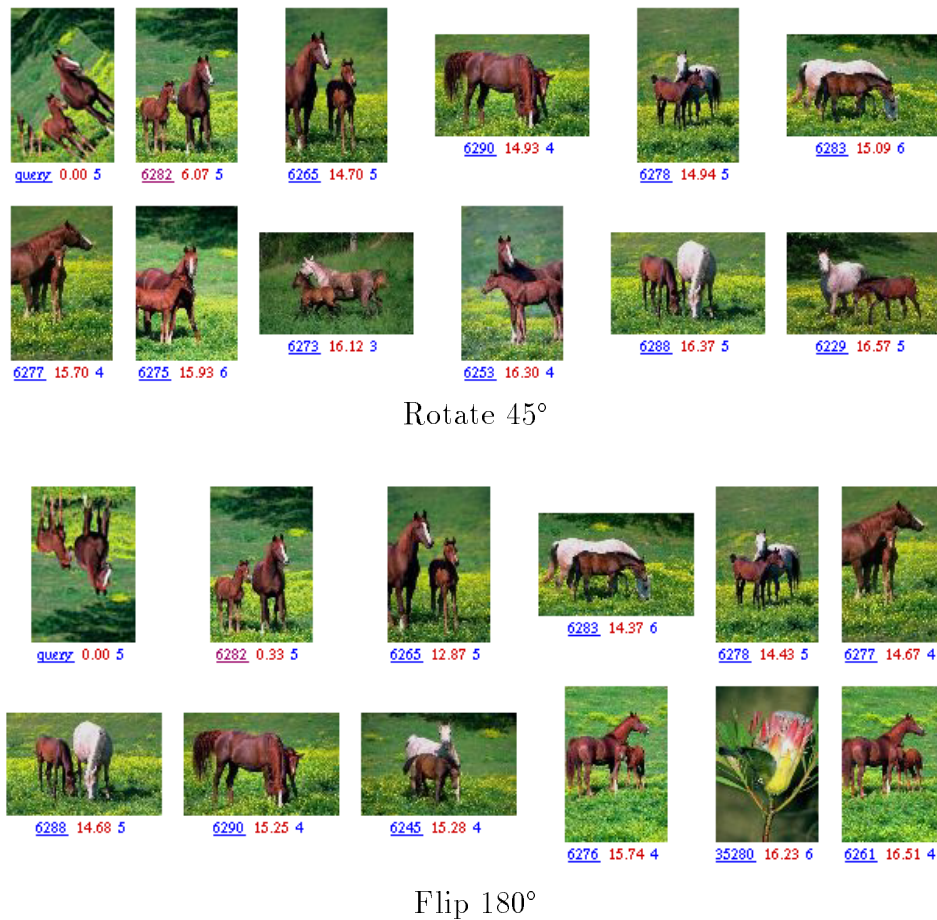


Figure 7.29: The robustness of the system to image rotation.



The algorithm has been implemented on a Pentium III 430MHz PC using the Linux operating system. To compute the feature vectors for the 200,000 color images of size  $384 \times 256$  in our general-purpose image database requires approximately 60 hours. On average, one second is needed to segment an image and to compute the features of all regions. The speed is much faster than other region-based methods. For example, the Blobworld system developed by University of California at Berkeley segments each image in 8 minutes. Fast indexing has provided us with the capability of handling external queries and sketch queries in real-time.

The matching speed is very fast. When the query image is in the database, it takes about 1.5 seconds of CPU time on average to sort all the images in the 200,000-image database using our similarity measure. If the query image is not already in the database, one extra second of CPU time is spent to extract the feature from the query image. Without feature space clustering, the complexity of the query processing time is  $O(N \log(N))$ , for class database size  $N$ . For example, if the size of the “textured” class database is 10,000 images, the time to process a textured image query is  $O(10000 \log(10000))$ . The current image semantic classification process is sequential. With very large image databases, we will use parallel processing to classify the query images into semantic classes.

## 7.9 Summary

In this chapter, we gave some implementing details of the experimental systems we have developed, the SIMPLIcity system and the Pathfinder system. Explanations pertaining to their performance should help in understanding their methods. We introduced the data sets, the query interfaces, the accuracy evaluation, the robustness evaluation, and the speed evaluation. In general, our SIMPLIcity system performs much better and much faster than the systems with which we have compared it. Besides, SIMPLIcity is robust to intensity variations, sharpness variations, color distortions, shape distortions, cropping, scaling, shifting, and rotation. The Pathfinder system, a system based on the SIMPLIcity system, provides visual similarity querying capabilities for high-resolution biomedical image databases.

# Chapter 8

## Conclusions and Future Work

*A scholar who cherishes the love of comfort*

*is not fit to be deemed a scholar.*

— Lao-Tzu (~ 570-490 B.C.)

This chapter concludes the dissertation. In Section 8.1 we summarize the main themes of our research on semantics-sensitive integrated matching for picture libraries and biomedical image databases. In Sections 8.2, we examine limitations of our solution. Suggestions for future work are given in Section 8.3.

### 8.1 Summary

One contribution of this work is the idea that images can be classified into global semantic classes, such as textured or nontextured, indoor or outdoor, objectionable or benign, graph or photograph, radiology and pathology, and that much can be gained if the feature extraction scheme is tailored to best suit each class.

For the purpose of searching general-purpose image databases, we have developed a series of statistical image classification methods, including the graph-photograph, textured-nontextured, objectionable-benign classifiers. Evaluation over real-world images is given for each classification method.

We have explored the application of advanced wavelets in feature extraction and image coding. We have developed an image region segmentation algorithm using wavelet-based feature extraction and the k-means statistical clustering algorithm. The algorithm is much faster than the existing algorithms. We rely on a robust region matching measure than on precise image segmentation.

We have developed a measure for the overall similarity between images, i.e., the Integrated Region Matching (IRM) measure, defined based on a region-matching scheme that integrates properties of all the regions in the images, resulting in a simple querying interface. The advantage of using such a soft matching is the increased robustness against poor segmentation, an important property overlooked in previous work.

Finally, we have implemented these methods in an experimental system SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) with Web-based query interfaces. The application of SIMPLIcity to a database of about 200,000 general-purpose images and a database of pathology images shows more accurate and much faster retrieval compared with the existing algorithms. An important feature of the algorithms implemented in SIMPLIcity is that it is robust to intensity variations, sharpness variations, color distortions, other distortions, cropping, scaling, shifting, and rotation. The system is also easier to use than other region-based retrieval systems.

## 8.2 Limitations

The SIMPLIcity system is not perfect. We itemize the main limitations of the system:

- **Semantic classification:** At the current stage, we use only low-level image semantics or semantic types in distinguishing images in the database. The performance is still far from human performance in understanding image semantics. We discuss possible future work to address these issues in the next section.
- **Region-based features:** Like other low-level features, region-based features do not provide human-level perception of objects and semantics. Region-based

systems are unlikely to be able to handle high-level queries including object queries (e.g., find pictures with a double-decker bus) and image purpose queries (e.g., find people fighting, find happy mood pictures).

- **Integrated Region Matching metric:** The basic assumption of the IRM metric is that images with similar semantics have similar object compositions or similar regions in the feature space. This assumption may not always hold.
- **Classifiers:** The statistical semantic classification methods do not distinguish images in different classes perfectly. Furthermore, an image may fall into several semantic classes simultaneously.
- **Querying interfaces:** The querying interfaces are not powerful enough to allow users to formulate their queries freely. For different user domains, the query interfaces should ideally provide different sets of functions.
- **Evaluation:** The evaluation of CBIR is still a difficult task. We need a valid large-scale evaluation method which gives absolute measurements of how a CBIR system performs.

Still, SIMPLiCity provides dynamically a good selection of relevant images for users so that searching in large-scale image databases becomes feasible.

### 8.3 Areas of future work

Advances in CBIR are possible in the following areas:

- **Architecture:** A statistical soft classification architecture can be developed to allow an image to be classified based on its probability of belonging to a certain semantic class. We are also planning to apply parallel processing for semantic classification. We will allow users to choose between overall matching and center-weighted matching.

- **High-level classifiers:** We have obtained good progress in applying image matching and statistics in image classification. One example is the WIPE system, described in Appendix A, an effective tool for Web image classification.

We need to design more high-level classifiers. We must improve existing classifiers with the help of a large amount of training data and advanced statistical methods.

- **Feature selection:** Region-based features and other low-level features can be used to determine high-level semantic features. For example, a sunset scene may contain a round region of certain bright yellow color over a dark reddish background. Can we develop a statistical learning method to *recognize* certain objects and index images based on their object composition? We are also working on to extract text from icons for image-based Web page identification.

- **Querying interface:** We need to continue our effort in designing simple but capable graphical user interfaces. Domain knowledge will be required for domain-specific applications, such as biomedicine. Collaborations with experts of other fields (e.g., psychology, library science) will be required.

- **Evaluation:** Science requires standard or sharable methods to assess progress. We are planning to build a sharable testbed for statistical evaluation of different CBIR systems.

We will also evaluate the robustness of other CBIR systems. We expect the color histogram systems to be sensitive to intensity variation, color distortions, and cropping. Color layout indexing will be sensitive to shifting, cropping, scaling, and rotation.

- **Accuracy with very large databases:** Currently, most CBIR system developers are working on relatively small-scale image databases. The database we have used is one of the largest. However, it is still too small to experience problems that may arise from very large image databases, such as the set of images available on the Web and being extracted by some research efforts [18]. We

need to create an image database with millions of images and test the retrieval methods on this database.

- **Speed with very large databases:** Statistical feature clustering research focuses mainly on well-defined distances such as the Euclidean distance. In real-world CBIR systems, often special distances are used [97]. We need to explore clustering methods for specialized metrics such as the IRM metric.

Currently, we use the disk storage to store all the feature vectors in the database. On average, 400 bytes are used to store the feature vector of an image. A database of 2,000,000 images takes less than 1.0 GB of space. To further speed up the system, we may store the feature data in the main memory.

- **Special image databases:** We are planning to expend more effort on special image databases, especially for biomedical applications. Biomedical image databases are more challenging due to the size of the image data and the amount of details required in representing images. Currently, we are initiating a joint research effort with the Radiology Department of University of California at San Francisco. We will develop a wavelet-based feature extraction scheme and a region-based matching scheme for three-dimensional medical images.
- **Special applications:** We are interested in more applications of CBIR. For example, we need to explore the application of CBIR in image classification, for both general-purpose pictures and biomedical images.

- **High level indexing:**

Finally, we are exploring the question of the extent to which high-level indexing can be made possible. Can we build indexes so that retrieval based on high-level queries becomes possible?

# Appendix A

## Image Classification By Image Matching

*It pays to have a healthy skepticism about  
the miracles of modern technology.*

— Dennis A. Hejhal (1949- ), commenting on the Internet

### A.1 Introduction

Image classification is usually performed by making measurements on the image itself. For example, texture measurements are made to classify an image as “natural” or “urban.” This appendix describes an experiment that we carried out to determine whether an image could be classified by matching the image against databases of images, where each database belongs to a different class. Because we wanted a classification problem that could not be solved in a reasonable time by making measurements on the image, we selected the classes of interest as “benign images” and “objectionable images.” This classification problem is an extremely difficult one because although people can usually make the classification, it is almost impossible to formally define “benign” and “objectionable.”

Success in this domain would show that the database matching approach to image classification is practical. In addition, the “benign/objectionable” problem is of practical interest since certain users, such as children, should be denied access to objectionable images.

Our work was inspired by the work at the computer vision group at the University of California, Berkeley. Sections A.2 and A.3 review related work in both the software industry and academia. We give details of our objectionable-image screening system in Section A.4. In Section A.5, we describe a related algorithm to classify a website based on image classification.

## A.2 Industrial solutions

There are many attempts to solve the problem of objectionable images in the software industry. Pornography-free websites such as the *Yahoo! Web Guides for Kids* have been set up to protect those children too young to know how to use the web browser to get to other sites. However, it is difficult to control access to other Internet sites.

Software programs such as *NetNanny*, *Cyber Patrol*, or *CyberSitter* are available for parents to prevent their children from accessing objectionable documents. However, the algorithms used in this software do not check the image contents. Some software stores more than 10,000 IP addresses and blocks access to objectionable sites by matching the site addresses, some focus on blocking websites based on text, and some software blocks all unsupervised image access. There are problems with all of these approaches. The Internet is so dynamic that more and more new sites and pages are added to it every day. Manually maintaining lists of sites is not sufficiently responsive. Textual matching has problems as well. Sites that most of us would find benign, such as the sites about breast cancer, are blocked by text-based algorithms, while many objectionable sites with text hidden in elaborate images are not blocked. Eliminating all images is not a solution since the Internet will not be useful to children if we do not allow them to view images.



### A.3 Related work in academia

Academic researchers are actively investigating alternative algorithms to screen and block objectionable media. Many recent developments in shape detection, object representation and recognition, people recognition, face recognition, and content-based image and video database retrieval are being considered by researchers for use in this problem [31, 34].

To make such algorithms practical for our purposes, extremely high sensitivity (or recall of objectionable images) with reasonably high speed and high specificity is necessary. In this application, *sensitivity* is defined as the ratio of the number of objectionable images identified to the total number of objectionable images downloaded; *specificity* is defined as the ratio of the number of benign images passed to the total number of benign images downloaded. A perfect system would identify all objectionable images and not mislabel any benign images, and would therefore have a sensitivity and specificity of 1. The “gold standard” definition of objectionable and benign images is a complicated social problem and there is no objective answer. In our experiments, we use a combination of human judgment and the source of the images to serve as the gold standard.

For real-world application needs, a high sensitivity is desirable, i.e., the correct identification of almost every objectionable image even though this may result in some benign images being mislabeled. Parents might be upset if their children are exposed to even a few objectionable images.

The following properties of objectionable images found on the Internet make the problem extremely difficult:

- mostly contain non-uniform image background;
- foreground may contain textual noise such as phone numbers, URLs, etc;
- content may range from grey-scale to 24-bit color;
- some images may be of very low quality (sharpness);
- views are taken from a variety of camera positions;

- a single image may be an indexing image containing many small icons;
- images may contain more than one person;
- persons in the picture may have different skin colors;
- images may contain both people and animals;
- images may contain only some parts of a person;
- persons in the picture may be partially dressed.

Forsyth's research group [31, 34] has designed and implemented an algorithm to screen images of naked people. Their algorithms involve a skin filter and a human figure grouper. As indicated in [31], 52.2% sensitivity and 96.6% specificity have been obtained for a test set of 138 images with naked people and 1401 assorted benign images. However, it takes about 6 minutes on a workstation for the figure grouper in their algorithm to process a suspect image passed by the skin filter.

## A.4 System for screening objectionable images

WIPE<sub>TM</sub> (Wavelet Image Pornography Elimination) is a system we have developed that is capable of classifying an image as objectionable or benign. The algorithm uses a combination of an icon filter, a graph-photo detector, a color histogram filter, a texture filter, and a wavelet-based shape matching algorithm to provide robust screening of on-line objectionable images. Semantically-meaningful feature vector matching is carried out so that comparisons between a given on-line image and images in a pre-marked training data set can be performed efficiently and effectively.

The system is practical for real-world applications, processing queries at the speed of less than 2 seconds each, including the time to compute the feature vector for the query, on a Pentium Pro PC. Besides its exceptional speed, it has demonstrated 96% sensitivity over a test set of 1,076 digital photographs found on objectionable news groups. It wrongly classified 9% of a set of 10,809 benign photographs obtained from various sources (9% specificity). The specificity in real-world applications is expected

to be much higher because benign on-line graphs can be filtered out with our graph-photo detector with 100% sensitivity and nearly 100% specificity, and surrounding text can be used to assist the classification process.

Our approach is different from previous approaches. Instead of carrying out a detailed analysis of an image, we match it against a small number of feature vectors obtained from a training database of 500 objectionable images obtained from the Web and 8,000 benign images, after passing the images through a series of fast filters. If the image is close in content to a threshold number of pornographic images, e.g., matching two or more of the marked objectionable images in the training database within the closest 15 matches, it is considered objectionable. To accomplish this, we attempt to effectively code images based on image content and match the query with statistical information on the feature indexes of the training database.

#### A.4.1 Moments

Moments are descriptors widely used in shape and region coding [42] because a moment-based measure of shape can be derived that is invariant to translation, rotation, and scale. For a 2-D continuous surface  $f(x, y)$  embedded on the  $xy$ -plane, the *moment of order*  $(p + q)$  is defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (\text{A.1})$$

for  $p, q \in \mathbb{N} \cup \{0\}$ . The theory of moments has shown that the moment sequence  $\{m_{pq}\}$  is uniquely determined by  $f(x, y)$  and vice versa.

The *central moment* is defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(x - \frac{m_{10}}{m_{00}}\right)^p \left(y - \frac{m_{01}}{m_{00}}\right)^q f(x, y) dx dy. \quad (\text{A.2})$$

For discrete cases such as a digitized image, we define the *central moment* as

$$\mu_{pq} = \sum_x \sum_y \left(x - \frac{m_{10}}{m_{00}}\right)^p \left(y - \frac{m_{01}}{m_{00}}\right)^q f(x, y). \quad (\text{A.3})$$

Then the *normalized central moments* are defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad \text{where} \quad \gamma = \frac{p+q+2}{2} \quad (\text{A.4})$$

for  $p+q = 2, 3, 4, \dots$

A set of seven *translation, rotation, and scale invariant moments* can be derived from the 2nd and 3rd moments. A detailed introduction to these moments can be found in [42, 52]. These moments can be used to match two objectionable images containing people having the same posture but taken from different camera angles.

#### A.4.2 The algorithm

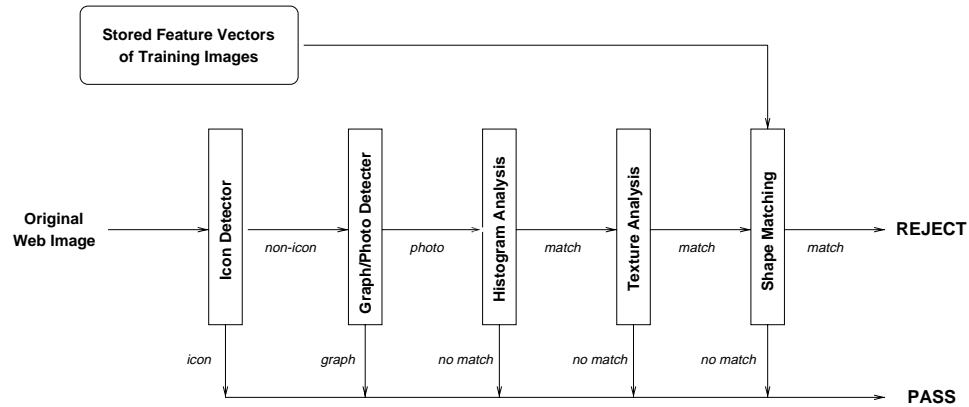


Figure A.1: Basic structure of the algorithm in WIPE.

We have developed a new shape-based indexing scheme using forward and backward Daubechies' wavelet transforms, variant and invariant normalized central moments, and color histogram analysis that is able to capture the object posture. The screening algorithm uses several major steps, as shown in Figure A.1. The layout of these filters is a result of a cost-effectiveness analysis. Faster filters are placed earlier in the pipeline so that benign images can be quickly passed.

Our design has several immediate advantages.

1. It does not rely too much on color when detecting sharp edges. That means that naked people of different races can be detected without bias. It also has

the potential for shape-based matching of benign images. Image background does not affect the querying results unless the background has sharp features. Also, the submitted image can be of different color quality.

2. We used multiresolution wavelet analysis rather than a traditional edge detector to capture the shape information in the images. This reduces the dependence on the quality or the sharpness of the images.
3. We used a combination of variant and invariant normalized central moments to make the querying independent of the camera position.

### **Icon filter and image normalization**

We first apply an icon filter to the image downloaded from the Internet. The current implementation of the icon filter is rather simple. If the length of any side of the image is small, we consider the image an icon image, and hence benign.

Many color image formats are currently in use, e.g., GIF, JPEG, PPM and TIFF are the most widely used formats on the Internet. Because images can have different formats and different sizes, we must first normalize the data for histogram computation. For the wavelet computation and moment analysis parts of our algorithm, any image size is acceptable. To save computation time, we rescale images using bi-linear interpolation so that the length of the longest side is 256 pixels. Red-Green-Blue (i.e., RGB) color space is used for histogram computation.

### **Graph/photo classification**

We apply a classifier to decide whether an image is a photograph, i.e., a continuous-tone image, or, a graph image, i.e., a image containing mainly text, line graphics and overlays. If the image is a graph image, it is very likely that the image is a benign image map commonly used on the web pages. The details of this specific classification method is given in Chapter 6.

Classification as a graph causes the WIPE algorithm to exit. Misclassifying a photographic image as graph image leads to false classification of objectionable images.

However, misclassifying a text image as photograph simply means that the image is sent to the next stage filter in the whole WIPE screening system. Consequently, we sacrifice the accuracy in classifying graph images to obtain very high sensitivity in classifying photographic images. In this step, we achieved 100% sensitivity for photographic images and higher than 95% specificity. This result was obtained on a database of 12,000 photographic images and a database of 300 randomly downloaded graph-based image maps from the web.

### **Color histogram analysis and texture analysis**

Examination of color histograms revealed that objectionable images have a different color distribution than benign images [123]. We use a total of 512 bins to compute the histogram. An efficient histogram analysis was designed and implemented. We manually define a certain color range in the color spectrum as human body colors. Then we define a weight for each of these colors based on the probability of being a human body color. While this assessment was done manually, in the future version of WIPE this will be done statistically. Finally a weighted amount of human body colors that the given image contains can be obtained by summing over the entire histogram. If we set a threshold of, say, 0.15, about one half of the benign images are then classified correctly, while only a small number of skin images are classified incorrectly.

The texture analysis part of the WIPE algorithm is rather simple due to the simple texture shown by the human body in objectionable images. We statistically analyze the histogram of the high frequency bands of the wavelet transform. If the areas of human body colors contain much high frequency variation, we consider the area a non-human body area.

### **Edge and shape detection and matching**

Clearly the color histogram approach alone is not sufficient. Sometimes two images may be considered very close to each other using this measure when in actuality they have completely unrelated semantics.

We apply the wavelet transform to perform multidirectional and multiscale edge detection. Readers are referred to [4] for the theoretical arguments on the effectiveness of a similar algorithm. Our purpose is not to obtain a high quality edge detection algorithm for this application. Rather, since the goal here is to effectively extract the conceptual shape information for objects and textural information for areas from the image, it is not necessary to produce a perceptually pleasant edge image. Consequently, we kept the algorithm simple to achieve a fast computation speed.

We start edge detection by transforming the image using the Daubechies-3 wavelet basis. The image is decomposed into four frequency bands with corresponding names LL, HL, LH and HH. The notation is borrowed from the filtering literature [117]. The letter 'L' stands for low frequency and the letter 'H' stands for high frequency. The left upper band is called 'LL' band because it contains low frequency information in both the row and column directions. An even number of columns and rows in the querying image is required due to the downsampling process of the wavelet transform. However, if the dimensions of the image are odd, we simply delete one column or one row of pixels from the boundaries.

The LH frequency band is sensitive to the horizontal edges, the HL band is sensitive to the vertical edges, and the HH band is sensitive to the diagonal edges [24, 4]. We detect the three types of edges separately and combine them at the end to construct a complete edge image. To detect the horizontal edges, we perform an inverse Daubechies-3 wavelet transform on a matrix containing only the wavelet coefficients in the LH band. Then we apply a zero-crossing detector in vertical direction to find the edges in the horizontal direction. The mechanism for using zero-crossing detector to find the edges can be found in [4]. Similar operations are applied to the HL and HH band, but different zero-crossing detectors are applied. For the HL band, we use a zero-crossing detector in the horizontal direction to find vertical edges and for the HH band, we use zero-crossing detector in the diagonal direction to find diagonal edges.

After we obtain the three edge maps, we combine them to get the final edge

image. To numerically show the combination, let us denote<sup>1</sup> the three edge maps by  $E_1[1 : m, 1 : n]$ ,  $E_2[1 : m, 1 : n]$  and  $E_3[1 : m, 1 : n]$ . The image size is  $m \times n$ . Then the final edge image, denoted by  $E[1 : m, 1 : n]$ , can be obtained from  $E[i, j] = (E_1[i, j]^2 + E_2[i, j]^2 + E_3[i, j]^2)^{\frac{1}{2}}$ .

Once the edge image is computed, we compute the normalized central moments up to order five and the translation, rotation, and scale invariant moments based on the gray scale edge image using the definitions in Section A.4.1. A feature vector containing these  $21 + 7 = 28$  moments is computed and stored for each image in the training database. When a submitted image comes in that has passed the histogram matching step, a moment feature vector is computed and a weighted Euclidean distance is used to measure the distance of the query and an image in the training database. The weights are determined so that matching of the 21 normalized central moments has higher priority than the matching of the 7 invariant moments. In fact, many objectionable images are of similar orientation.

If the query matches with objectionable images in the training database, we classify it as an objectionable image, otherwise we classify it as a benign image. The most recent version of our WIPE system uses a better alternative. We apply the CART (Chapter 4) algorithm on features of the matching results. For instance, we train the CART tree over a list of manually classified images and their query scores and results. Then we use the tree generated by CART to classify new cases. The accuracy is improved over simple nearest neighbor search.

### A.4.3 Evaluation

This algorithm has been implemented on a Pentium Pro 200MHz workstation. We selected about 500 objectionable images from news groups and 8,000 benign images from various sources such as the COREL Photo CD-ROM series for our training database. When we downloaded the objectionable images, we tried to eliminate those from the same source, i.e., those of extremely similar content. To compute the training

---

<sup>1</sup>Here we use MATLAB notation. That is,  $A(m_1 : n_1, m_2 : n_2)$  denotes the submatrix with opposite corners  $A(m_1, m_2)$  and  $A(n_1, n_2)$ .



feature vectors for the 8,000 color images in our database requires approximately one hour of CPU time.

We also selected 1,076 objectionable photographic images and 10,809 benign photographic images as our queries in order to test WIPE. The matching speed is very fast. It takes less than one second to process a submitted image and select the best 100 matching images from the 8,500 image database using our similarity measure. Once the matching is done, it takes almost no extra CPU time to determine the final answer, i.e., if the query is objectionable or benign.

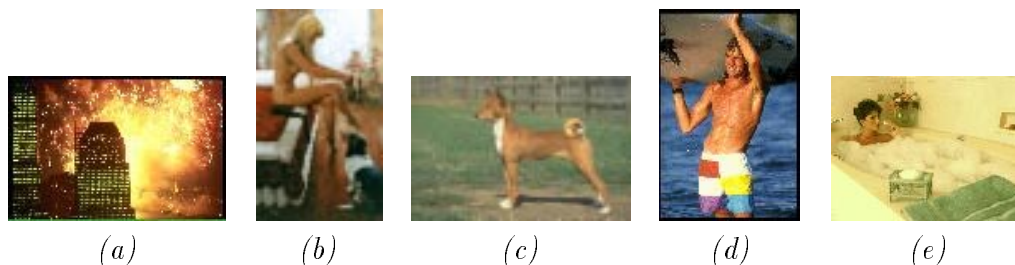


Figure A.2: Typical benign images being marked mistakenly as objectionable images by WIPE. (a) areas with similar features (b) fine-art image (c) animals (without clothes) (d) partially undressed human (e) partially obscured human.

Besides the fast speed, the algorithm has achieved remarkable accuracy. It has demonstrated 96% sensitivity and 91% specificity. Figure A.2 shows typical benign images being mistakenly marked by the WIPE system. We expect the specificity in real-world applications to be much higher than we reported here because there are many graph images in web pages. These images can be classified as benign images without any error. Also, we did not experiment on methods for assisting WIPE by processing surrounding text. We expect the performance to be much improved once image and textual information is combined.

## A.5 Classifying objectionable websites

This section describes IBCOW (Image-based Classification of Objectionable Websites), a system capable of classifying a website as objectionable or benign based on image content. The system uses  $WIPE_{TM}$  (Wavelet Image Pornography Elimination)

and statistics to provide robust classification of on-line objectionable World Wide Web sites. Semantically-meaningful feature vector matching is carried out so that comparisons between a given on-line image and images marked as "objectionable" and "benign" in a training set can be performed efficiently and effectively in the WIPE module. If more than a certain number of images sampled from a site is found to be objectionable, then the site is considered to be objectionable. The statistical analysis for determining the size of the image sample and the threshold number of objectionable images are given.

The system is practical for real-world applications, classifying a Web site at a speed of less than 2 minutes each, including the time to compute the feature vector for the images downloaded from the site, on a Pentium Pro PC. Besides its exceptional speed, it has demonstrated higher than 97% sensitivity and 97% specificity in classifying a Web site based solely on images. Both the sensitivity and the specificity in real-world applications is expected to be higher because our performance evaluation is relatively conservative and surrounding text can be used to assist the classification process.

IBCOW can be incorporated in a World Wide Web client software program so that a website is first screened by the system before the client starts to download the contents of the website. Once the website is screened, it is saved in the local storage so that it is considered safe for some period of time. IBCOW can also be used as a tool for screening software companies to generate lists of potentially objectionable World Wide Web sites.

### **A.5.1 The algorithm**

In this section, we derive the optimal algorithm for classifying a World Wide Web site as objectionable or benign based on an image classification system like the WIPE system developed by us.

Figure A.3 shows the basic structure of the IBCOW system. For a given suspect website, the IBCOW system first downloads as many pages as possible from the website by following the links from the front page. The process is terminated after a pre-set timeout period. Once the pages are downloaded, we use a parser to extract

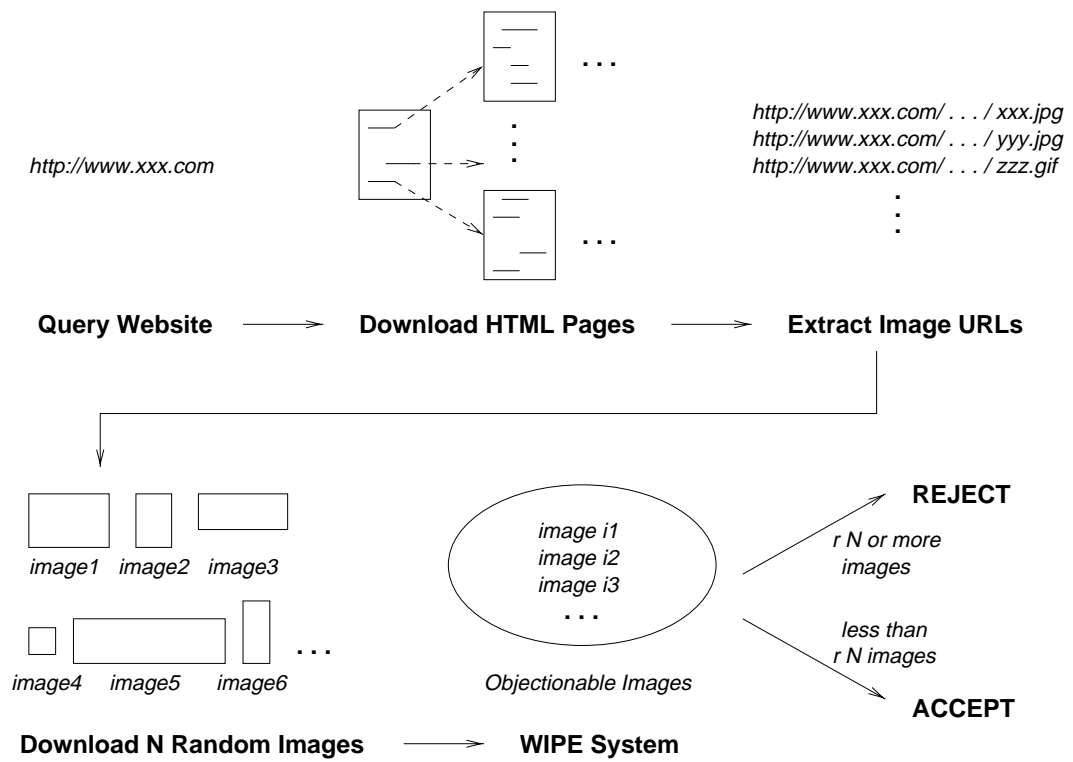


Figure A.3: Basic structure of the algorithm in IBCOW.

image URLs, i.e., URLs with suffixes such as ‘.jpg’ or ‘.gif’.  $N$  randomly selected images from this list are downloaded from this website. Then we apply WIPE to classify the images. If at least a subset, say  $r \times N$  images ( $r < 1$ ), of the  $N$  images are classified as objectionable by the WIPE system, the website is classified as objectionable by the IBCOW system; otherwise, the website is classified as a benign website. The following subsection will address the ideal combination of  $N$  and  $r$  given the performance of WIPE using statistical analysis.

### A.5.2 Statistical classification process for websites

In order to proceed with the statistical analysis, we must make some basic assumptions. A World Wide Web site is considered a benign website if none of the images provided by this website is objectionable; otherwise, it is considered an objectionable website. This definition can be refined if we allow a benign website to have a small amount, say, less than 2%, of objectionable images. In our experience, some websites that most of us would find benign, such as some university websites, may still contain some personal homepages with a small number of partially naked movie stars’ images.

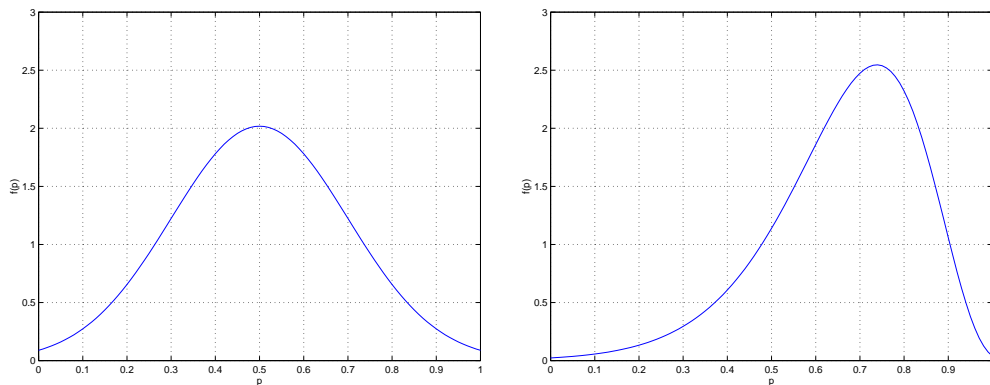


Figure A.4: Distributions assumed for the percentage ( $p$ ) of objectionable images on objectionable websites.

For a given objectionable website, we denote  $p$  as the chance of an image on the website to be an objectionable image. The probability  $p$  varies between 0.02 and 1 for various objectionable websites. Given a website, this probability equals to the

percentage of objectionable images over all images provided by the website. The distribution of  $p$  over all websites in the world physically exists, although we would not be able to know what the distribution is. Therefore, we assume that  $p$  obeys some hypothetical distributions, which are as shown in Figure A.4.

The performance of our WIPE system was evaluated by two parameters: sensitivity, denoted as  $q_1$ , is the accuracy of detecting an objectionable image as objectionable, and specificity, denoted as  $q_2$ , the accuracy of detecting a benign image as benign. The false positive rate, i.e., the failure rate of blocking an objectionable image, is thus  $1 - q_1$ , and the false negative rate, i.e., the false alarm rate for benign images, is  $1 - q_2$ .

For IBCOW, we must find out the minimum number of images, denoted as  $N$ , from a suspect website to be tested by the WIPE module in order to classify a website as objectionable or benign at a confidence level  $\alpha$ , i.e., with a probability  $1 - \alpha$  of being correct. The confidence level requirements on objectionable websites and benign websites may differ. For objectionable websites, we denote the desired confidence level to be  $\alpha_1$ , while for benign websites, we denote the desired confidence level to be  $\alpha_2$ . Furthermore, we must decide the threshold, denoted as  $r$ , for the percentage of detected objectionable images at which the IBCOW system will classify the website as objectionable. Therefore, the system tests  $N$  images from a website and classifies the website as objectionable if more than  $r \times N$  images are detected as objectionable by WIPE. Our objective is that when a website is rated as objectionable with probability higher than  $1 - \alpha_1$ , it will be classified as objectionable, and when a website is rated benign, with probability higher than  $1 - \alpha_2$ , it will be classified as benign.

According to the above assumptions, we can calculate the probabilities of misclassifying objectionable websites and benign websites. We start with the simpler case of benign websites.

$$P\{ \text{classified as benign} \mid \text{a website is benign} \} = P(I_2 \leq rN) \quad , \quad (\text{A.5})$$

where  $I_2$  is the number of images detected as objectionable by WIPE. Since  $I_2$  is a

binomial variable [67] with probability mass function

$$p_i = \binom{n}{i} (1 - q_2)^i q_2^{n-i}, \quad i = 0, 1, \dots, n, \quad (\text{A.6})$$

we have

$$P(I_2 \leq rN) = \sum_{i=1}^{[rN]} \binom{N}{i} (1 - q_2)^i q_2^{N-i}. \quad (\text{A.7})$$

Similarly, for objectionable websites, we get

$$P\{ \textit{classified as objectionable} \mid \textit{a website is objectionable} \} = P(I_1 > rN) \quad (\text{A.8})$$

For an objectionable website, suppose that any image in this website has probability  $p$  of being objectionable and it is independent of the other images, then the probability for this image to be classified as objectionable image is evaluated as follows:

$$\begin{aligned} & P\{ \textit{classified as objectionable} \} \\ &= P(A)P(\textit{classified as objectionable} \mid A) + \\ & \quad P(\tilde{A})P(\textit{classified as objectionable} \mid \tilde{A}) \\ &= pq_1 + (1 - p)(1 - q_2) \end{aligned}$$

where

$$\begin{aligned} A &= \{ \textit{the image is objectionable} \}, \\ \tilde{A} &= \{ \textit{the image is benign} \}. \end{aligned}$$

For simplicity, we denote

$$\lambda(p) = pq_1 + (1 - p)(1 - q_2). \quad (\text{A.9})$$

Similarly,  $I_1$  follows a binomial distribution with a probability mass function

$$p_i = \binom{n}{i} (\lambda(p))^i (1 - \lambda(p))^{n-i}, \quad i = 0, 1, \dots, n \quad . \quad (\text{A.10})$$

For this specific website,

$$P(I_2 > rN) = \sum_{[rN]+1}^N \binom{N}{i} (\lambda(p))^i (1 - \lambda(p))^{n-i} \quad . \quad (\text{A.11})$$

If  $p$  follows a truncated Gaussian distribution, i.e., the first hypothetical distribution, we denote the probability density function of  $p$  as  $f(p)$ . Thus,

$$\begin{aligned} & P\{ \textit{classified as objectionable} \mid \textit{a website is objectionable} \} \\ &= \int_0^1 \left[ \sum_{[rN]+1}^N \binom{N}{i} (\lambda(p))^i (1 - \lambda(p))^{n-i} \right] f(p) dp \quad . \end{aligned} \quad (\text{A.12})$$

As  $N$  is usually large, the binomial distribution can be approximated by a Gaussian distribution [67]. We thus get the following approximations.

$$\begin{aligned} & P\{ \textit{classified as benign} \mid \textit{a website is benign} \} \\ &= \sum_{i=1}^{[rN]} \binom{N}{i} (1 - q_2)^i q_2^{n-i} \approx \Phi \left( \frac{(r - (1 - q_2))\sqrt{N}}{\sqrt{q_2(1 - q_2)}} \right) \quad , \end{aligned} \quad (\text{A.13})$$

where  $\Phi(\cdot)$  is the cumulative distribution function of normal distribution [67]. Supposing  $r > (1 - q_2)$ , the above formula converges to 1 when  $N \rightarrow \infty$ .

$$\begin{aligned}
& P\{ \text{classified as objectionable} \mid \text{a website is objectionable} \} \\
& \approx \int_0^1 \left( 1 - \Phi \left( \frac{(r - \lambda(p))\sqrt{N}}{\sqrt{\lambda(p)(1 - \lambda(p))}} \right) \right) f(p) dp \quad , \quad (\text{A.14})
\end{aligned}$$

where  $\lambda(p) = pq_1 + (1 - p)(1 - q_2)$  as defined before.

When  $r < \lambda(p)$ ,

$$\lim_{N \rightarrow \infty} \Phi \left( \frac{(r - \lambda(p))\sqrt{N}}{\sqrt{\lambda(p)(1 - \lambda(p))}} \right) \rightarrow 0 \quad . \quad (\text{A.15})$$

Obviously, for any reasonable objectionable-image screening system,  $q_1 > 1 - q_2$ , i.e., the truth positive (TP) rate is higher than the false positive (FP) rate. Hence, we can choose  $r$  so that  $r \in (1 - q_2, \epsilon q_1 + (1 - \epsilon)(1 - q_2))$  for  $\epsilon > 0$ . The inequality  $r > 1 - q_2$  will guarantee that the probability of misclassifying benign websites approaches zero when  $N$  becomes large, which we concluded in a previous analysis. On the other hand, the inequality  $r < \epsilon q_1 + (1 - \epsilon)(1 - q_2)$  will enable the probability of misclassifying objectionable websites to become arbitrarily small when  $N$  becomes large.

To simplify notation, we let

$$\Delta_{r,N}(p) = 1 - \Phi \left( \frac{(r - \lambda(p))\sqrt{N}}{\sqrt{\lambda(p)(1 - \lambda(p))}} \right) \quad . \quad (\text{A.16})$$

Note that

$$\int_0^1 \Delta_{r,N}(p) f(p) dp \geq \int_\epsilon^1 \Delta_{r,N}(p) f(p) dp \quad . \quad (\text{A.17})$$

By increasing  $N$ , we can choose arbitrarily small  $\epsilon$  so that  $\Delta_{r,N}(p)$  is as close to 1 as we need, for all  $p > \epsilon$ . Hence,  $\int_\epsilon^1 \Delta_{r,N}(p) f(p) dp$  can be arbitrarily close to  $\int_\epsilon^1 f(p) dp$ . Since we can choose  $\epsilon$  arbitrarily small, this integration approaches to 1. In conclusion, by choosing  $r$  slightly higher than  $1 - q_2$  and  $N$  large, our system can perform near-perfect classification of both objectionable and benign websites.



As we only require a confidence level  $\alpha$ , i.e.,  $1 - \alpha$  correctness, we have much more freedom in choosing  $r$  and  $N$ . Our WIPE system can provide a performance with  $q_1 = 96\%$  and  $q_2 = 91\%$ . The actual  $q_1$  and  $q_2$  in real world can be higher because icons and graphs on the Web can be easily pre-classified with close to 100% sensitivity. When we assume  $f(p)$  being a truncated Gaussian with mean  $\bar{p} = 0.5$  and standard deviation  $\sigma = 0.2$ , which is plotted in Figure A.4, we may test  $N = 25$  images from each website and mark the website as objectionable once 5 or more images are identified as objectionable by the WIPE system. Under this configuration, we can achieve higher than 97% correctness for classifying both objectionable websites and benign websites.

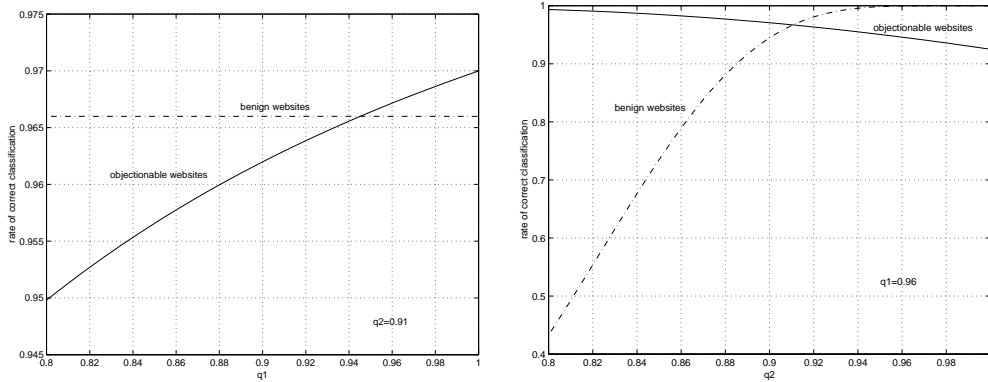


Figure A.5: Dependence of correct classification rates on sensitivity and specificity of WIPE (for the Gaussian-like distribution of  $p$ ). Left:  $q_2 = 91\%$ ,  $q_1$  varies between 80% to 100%. Right:  $q_1 = 96\%$ ,  $q_2$  varies between 80% to 100%. Solid line: correct classification rate for objectionable websites. Dash dot line: correct classification rate for benign websites.

If we fix the decision rule of our system, i.e., test a maximum of 25 images from each website and mark the website as objectionable once 5 or more images are identified as objectionable, the percentages of correctness for classification of both types of websites depend on the sensitivity parameter  $q_1$  and the specificity parameter  $q_2$ . By fixing  $q_2$  to 91% and changing  $q_1$  between 80% to 100%, the percentages of correctness for both types of websites are shown in the left panel of Figure A.5. Similarly, the results are shown in the right panel of Figure A.5 for the case of fixing  $q_1$  to 96%

and changing  $q_2$  between 80% to 100%. As shown in the graph on the left side, when  $q_2$  is fixed, the correct classification rate for benign websites is a constant. On the other hand, the correct classification rate for objectionable websites degrades with the decrease of  $q_1$ . However, the decrease of the correct classification rate is not sharp. Even when  $q_1 = 0.8$ , the correct classification rate is higher than 92%. On the other hand, when  $q_1 = 96\%$ , no matter what  $q_2$  is, the correct classification rate for objectionable websites is always above 90%. The rate of correctness for benign websites monotonically increases with  $q_2$ . Since benign images in an objectionable website are less likely to be classified as objectionable when  $q_2$  increases, the number of objectionable images found in the set of test images is less likely to pass the threshold 5. As a result, the correct classification rate for objectionable websites decreases slightly with the increase of  $q_2$ . However, the correct classification rate will not drop below 90%.

In the above statistical analysis, we assumed that the probability of an image being objectionable in an objectionable website has distribution  $f(p)$  with mean 0.5. In real life, this mean value is usually higher than 0.5. With a less conservative hypothetical distribution of  $p$ , as shown in the right panel of Figure A.4, we can achieve higher than 97% correctness by testing only 12 images from each website and marking the website as objectionable if 3 or more images are identified as objectionable by the WIPE system.

### A.5.3 Limitations

The screening algorithm in our IBCOW system assumes a minimum of  $N$  images downloadable from a given query website. For the current system set up,  $N$  can be as low as 12 for the less conservative assumption. However, it is not always possible to download 12 images from each website. We have noticed that some objectionable websites put only a few images on its front page for non-member netters to view without a password. For these websites, surround text will be more useful than images in the classification process. Also, we are considering to assign probabilities of objectionable to such sites based on accessible images.

In the statistical analysis, we assume each image in a given website is equally likely to be an objectionable image. This assumption may be false for some websites. For example, some objectionable websites put objectionable images in deep links and relatively benign images in front pages. In this case, we have to download more images from the website.

#### **A.5.4 Evaluation**

This algorithm has been implemented on a Pentium Pro 200MHz workstation. We selected 20 objectionable websites and 40 benign websites from various categories. It takes in general less than 2 minutes for the system to process each website. Besides the fast speed, the algorithm has achieved remarkable accuracy. It correctly identified all the 20 objectionable websites and did not mismark any one of the 40 benign websites. We expect the speed to be much faster once image and textual information is combined in the classification process.

### **A.6 Summary**

In this appendix, we described the application of our CBIR system to the problem of image classification. An experimental system, targeted to the classification of Web images, has been developed. Success in this domain showed that the database matching approach to image classification is practical. Classifying biomedical images may be possible with this approach combined with a large amount of training image data. The “benign/objectionable” problem is of practical interest since certain users, such as children, should be denied access to objectionable images.

# Bibliography

- [1] C. N. Adams et al., "Evaluating quality and utility of digital mammograms and lossy compressed digital mammograms," *Proceedings of the Third International Workshop on Digital Mammography*, Elsevier, pp. 169-176, Amsterdam, 1996.
- [2] R. B. Altman, "Bioinformatics in support of molecular medicine," *Proceedings of the 1998 AMIA Annual Symposium*, Orlando, FL, pp. 53-61, 1998.
- [3] A. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, 1992.
- [4] T. Aydin, Y. Yemez, E. Anarim, B. Sankur, "Multidirectional and multiscale edge detection via M-band wavelet transform," *IEEE Trans. on Image Processing*, vol. 5, no. 9, pp. 1370-1377, 1996.
- [5] D. H. Ballard, C. M. Brown, *Computer Vision*, Prentice Hall, New Jersey, 1982.
- [6] G. Beylkin, R. Coifman, V. Rokhlin, "Fast wavelet transforms and numerical algorithms," *Comm. Pure Appl. Math.*, vol. 44, pp. 141-183, 1991.
- [7] P. Blonda, G. Satalino, A. Baraldi, R. De Blasi, "Segmentation of multiple sclerosis lesions in MRI by fuzzy neural networks: FLVQ and FOSART," *1998 Conference of the North American Fuzzy Information Processing Society*, Pensacola Beach, FL, Aug. 1998.
- [8] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, *Classification and Regression Trees*, Wadsworth International Group, Belmont, Calif., 1984.

- [9] M.C. Burl, M. Weber, P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," *Computer Vision - ECCV'98 5th European Conference on Computer Vision*, pp. 628-41, vol. 2, Springer-Verlag, Freiburg, Germany, 2-6 June 1998.
- [10] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-698, 1986.
- [11] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, J. Malik, "Blobworld: a system for region-based image indexing and retrieval," *Third Int. Conf. on Visual Information Systems*, D. P. Huijsmans, A. W.M. Smeulders (eds.), Springer, Amsterdam, The Netherlands, June 2-4, 1999.
- [12] H.P. Chan, K. Doi, C.J. Vyborny, C.E. Metz, H. MacMahon, P.M. Jokich, S. Galhotra, "Digital mammography: development of a computer-aided system for detection of microcalcifications," *Proc. SPIE*, vol. 767, pt. 1, pp. 367-70, Newport Beach, CA, February 1-6, 1987.
- [13] E. Chang, J. Z. Wang, C. Li, G. Wiederhold, "RIME: a replicated image detector for World Wide Web," *Proceedings of SPIE Symposium of Voice, Video, and Data Communications*, vol. 3527, pp. 58-67, Boston, November 1998.
- [14] E. Chang, C. Li, J. Z. Wang, P. Mork, G. Wiederhold, "Searching near-replicas of images via clustering," *Proceedings of SPIE Symposium of Voice, Video, and Data Communications*, Boston, September 1999.
- [15] S.-F. Chang, W. Chen, H. Meng, H. Sundaram, D. Zhong. "VideoQ: an automated content based video search system using visual cues," *Proceedings of ACM Multimedia 1997*, Seattle, November 1997.
- [16] Q. Chen, H. Wu, M. Yachida, "Face detection by fuzzy pattern matching," *Proceedings of IEEE International Conference on Computer Vision*, pp. 591-6, Cambridge, MA, USA 20-23 June 1995.

- [17] Y. Chen, E. K. Wong, "Augmented image histogram for image and video similarity search," *Proceedings of the SPIE, Storage and Retrieval for Image and Video Databases VII*, vol. 3656, pp. 523-32, San Jose, CA, USA 26-29 Jan. 1999.
- [18] J. Cho, H. Garcia-Molina, L. Page, "Efficient crawling through URL ordering," *7th International World Wide Web Conference, Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 161-72, Elsevier, April, 1998.
- [19] C. K. Chui, *An Introduction to Wavelets*, Academic Press, Boston, 1992.
- [20] C. K. Chui, *Wavelets: A Tutorial in Theory and Applications*, Academic Press, Inc., San Diego, 1992.
- [21] R. V. Churchill, *Fourier Series and Boundary Value Problems*, McGraw-Hill, New York, 1987.
- [22] A. Cohen, I. Daubechies, P. Vial, "Wavelets on the interval and fast wavelet transforms," *Appl. Comput. Harm. Anal.*, vol. 1, pp. 54-82, 1993.
- [23] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure and Appl. Math.*, vol. 41, pp. 909-996, 1988.
- [24] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, 1992.
- [25] D. L. Donoho, I.M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425-455, 1994.
- [26] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, D. Picard, "Wavelet shrinkage: asymptopia?" *Journal of the Royal Statistical Society ser. B*, vol. 57, pp. 301-337, 1995.
- [27] D.L. Donoho, I.M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Am. Statist. Association*, vol. 90, pp. 1200-1224, 1995.
- [28] J. S. Duncan, N. Ayache, "Medical image analysis: progress over two decades and the challenges ahead", *IEEE PAMI*, vol. 22, no. 1, pp. 85-105, January 2000.

- [29] M. Effros, "Zerotree design for image compression: toward weighted universal zerotree coding," *Proceedings of the IEEE International Conference on Image Processing*, Santa Barbara, November 1997.
- [30] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies*, vol. 3, no. 3-4, pp. 231-62, July 1994.
- [31] M. Fleck, D. A. Forsyth, C. Bregler, "Finding naked people," *Proc. 4<sup>th</sup> European Conf on Computer Vision*, UK, vol 2, pp. 593-602, 1996.
- [32] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, "Query by image and video content: the QBIC system," *Computer*, vol. 28, no. 9, pp. 23-32, Sept. 1995.
- [33] G. B. Folland, *Fourier Analysis and Its Applications*, Brooks/Cole Publishing Co., Pacific Grove, Calif., 1992.
- [34] D. A. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, C. Bregler, "Finding pictures of objects in large collections of images," *Proc. Int'l Workshop on Object Recognition*, Cambridge, 1996.
- [35] A. Fournier, "Transfers and fluxes of wind kinetic energy between orthogonal wavelet components during atmospheric blocking," *Wavelets in Physics*, Cambridge, pp. 263-298, 1999.
- [36] G. C. Freeland, T.S. Durrani, "On the use of the wavelet transform in fractal image modelling," *IEE Colloquium on Application of Fractal Techniques in Image Processing (Digest No.171)*, pp. 2/1-4, London, UK, IEE, 1990.
- [37] H. Garcia-Molina, J. Ullman, J. Widom, *Database system implementation*, Prentice Hall, New Jersey, 2000.
- [38] A. Gersho, "Asymptotically optimum block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, no. 4, pp. 373-380, July 1979.

- [39] A. Gersho, R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1991.
- [40] E. Giladi, M. G. Walker, J. Z. Wang, W. Volkmuth, "SST: an algorithm for searching sequence databases in time proportional to the logarithm of the database size," *Currents in Computational Molecular Biology, Proceedings of the Fourth Annual International Conference on Computational Molecular Biology (RECOMB)*, S. Miyano, R. Shamir, T. Takagi, (eds), Universal Academy Press, Inc., Japan, 2000.
- [41] D. Goldberg-Zimring, A. Achiron, S. Miron, M. Faibel, H. Azhari, "Automated detection and characterization of multiple sclerosis lesions in brain MR images," *Magnetic Resonance Imaging*, vol. 16, no. 3, pp. 311-18, Elsevier, April 1998.
- [42] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Co., Mass., 1993.
- [43] S. J. Gortler, P. Schroder, M. F. Cohen, P. Hanrahan, "Wavelet radiosity," *Proceedings of SIGGRAPH 1993, Computer Graphics Proceedings, Annual Conference Series*, pp. 221-230, August, 1993.
- [44] R. M. Gray, J. W. Goodman, *Fourier Transforms : An Introduction for Engineers*, Kluwer Academic Publishers, Boston, 1995.
- [45] S. M. Griffin, "Digital Libraries Initiative - Phase 2: fiscal year 1999 awards," *D-Lib Magazine*, vol. 5, No. 7/8, July/August 1999. <http://www.dlib.org>
- [46] A. Gupta, R. Jain, "Visual information retrieval," *Comm. Assoc. Comp. Mach.*, vol. 40, no. 5, pp. 70-79, May 1997.
- [47] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 729-36, July, 1995.



- [48] D. Hanselman, B. Littlefield, *The student edition of MATLAB : version 5, user's guide*, The MathWorks, Inc., Prentice Hall, Upper Saddle River, NJ, 1997.
- [49] D. Harman, "Relevance feedback and other query modification techniques," *Information retrieval: Data structures & algorithms*, Prentice Hall, New Jersey, 1992.
- [50] J. A. Hartigan, M. A. Wong, "Algorithm AS136: a k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100-108, 1979.
- [51] F. S. Hillier, G. J. Lieberman, *Introduction to Operations Research*, McGraw-Hill, New York, 1990.
- [52] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE (IEEE) Trans. Info. Theory*, vol. IT, no. 8, pp. 179-187, 1962.
- [53] P. R. Hubbs, M. Tsai, P. Dev, P. Godin, J. G. Olyarchuk, D. Nag, G. Linder, T. C. Rindfleisch, K. L. Melmon, "The Stanford Health Information Network for Education (SHINE): integrated information for decision making and learning," *Proc AMIA Annual Fall Symp.*, pp. 505-8, Nashville, Tennessee, October, 1997.
- [54] L. M. Hurvich, D. Jameson, "An opponent-process theory of color vision," *Psychological Review*, vol. 64, pp. 384-390, 1957.
- [55] "Web surpasses one billion documents," *Inktomi Corporation Press Release*, January 18, 2000.
- [56] C. Isaac, D. K. B. Li, M. Genton, et al., "Multiple sclerosis: a serial study using MRI in relapsing patients," *Neurology*, 38:1511-1515, 1988.
- [57] C. E. Jacobs, A. Finkelstein, D. H. Salesin, "Fast multiresolution image querying," *Proceedings of SIGGRAPH 95, in Computer Graphics Proceedings, Annual Conference Series*, pp. 277-286, August 1995.
- [58] A. K. Jain, R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, New Jersey, 1988.

- [59] R. Jain, S. N. J. Murthy, P. L.-J. Chen, S. Chatterjee “Similarity measures for image databases”, *Storage and Retrieval for Image and Video Databases III, Proceedings of the SPIE*, vol. 2420, pp. 58-65, San Jose, CA, Feb. 9-10, 1995.
- [60] B. Jansen, A. Spink, J. Bateman, T. Saracevic, “ Real life information retrieval: a study of user queries on the Web,” *ACM SIGIR Forum*, vol. 32, No. 1, 1998.
- [61] M. J. Jensen, “An alternative maximum likelihood estimator of long-memory processes using compactly supported wavelets,” *Journal of Economic Dynamics and Control*, vol. 24 no. 3, pp. 361-87, March, 2000.
- [62] B. Johnson, M. S. Atkins, B. Mackiewicz, M. Anderson, “Segmentation of multiple sclerosis lesions in intensity corrected multispectral MRI,” *IEEE Tran. on Medical Imaging*, vol. 15, no. 2, pp. 154-169, 1996.
- [63] G. Kaiser, *A Friendly Guide to Wavelets*, Birkhauser, Boston, 1994.
- [64] G. Kaiser, “Physical wavelets and radar: a variational approach to remote sensing,” *IEEE Antennas and Propagation Magazine*, vol. 38, no. 1, pp. 15-24, February, 1996.
- [65] S. Lawrence, C.L. Giles, “Searching the World Wide Web,” *Science*, vol. 280, pp. 98, 1998.
- [66] S. Lawrence, C.L. Giles, “Accessibility of information on the Web,” *Nature*, vol. 400, pp. 107-109, 1999.
- [67] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, Addison-Wesley Publishing Company, Mass., pp. 99-110, pp. 280-287, 1994.
- [68] J. Li, R. M. Gray, “Context based multiscale classification of images,” *Int. Conf. Image Processing*, Chicago, Oct. 1998.
- [69] J. Li, R. M. Gray, “Text and picture segmentation by the distribution analysis of wavelet coefficients,” *Int. Conf. Image Processing*, Chicago, Oct. 1998.

- [70] J. Li, J. Z. Wang, R. M. Gray, G. Wiederhold, "Multiresolution object-of-interest detection of images with low depth of field," *Proceedings of the 10th International Conference on Image Analysis and Processing*, Venice, Italy, 1999.
- [71] J. Li, J. Z. Wang, G. Wiederhold, "Classification of textured and non-textured images using region segmentation," *Proceedings of the Seventh International Conference on Image Processing*, Vancouver, BC, Canada, September, 2000.
- [72] J. Li, J. Z. Wang, G. Wiederhold, "IRM: Integrated Region Matching for image retrieval," *Proceedings of the 2000 ACM Multimedia Conference*, Los Angeles, October, 2000.
- [73] J. Li, J. Z. Wang, G. Wiederhold, "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries," *Proceedings of the Fourth International Conference on Visual Information Systems (VISUAL)*, Lyon, France, November 2-4, 2000.
- [74] A. R. Lindsey, J. Dill, "Wavelet packet modulation: a generalized method for orthogonally multiplexed communications," *Proc. 27th Southeastern Symposium on System Theory*, Starkville, MS, March, 1995.
- [75] Y. Liu, W.E. Rothfus, M.D., T. Kanade, "Content-based 3D neuroradiologic image retrieval: preliminary results," *IEEE International Workshop on Content-based Access of Image and Video Databases*, Bombay, India, January, 1998.
- [76] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, IT-28:127-135, March 1982.
- [77] W. Y. Ma, B. Manjunath, "NaTra: A toolbox for navigating large image databases," *Proc. IEEE Int. Conf. Image Processing*, Santa Barbara, pp. 568-71, 1997.
- [78] W. Y. Ma, B. S. Manjunath, "Edge flow: a framework of boundary detection and image segmentation," *Proceedings of IEEE Computer Society Conference on*

- Computer Vision and Pattern Recognition*, pp. 744-9, San Juan, Puerto Rico, June, 1997.
- [79] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674-693, July 1989.
- [80] S. G. Mallat, "Multiresolution approximations and wavelet orthonormal bases of  $L^2(R)$ ," *Trans. Amer. Math. Soc.*, vol. 315, no. 1, pp. 69-87, 1989.
- [81] E. Mathias, A. Conci, "Comparing the influence of color spaces and metrics in content-based image retrieval," *Proceedings SIBGRAP'98. International Symposium on Computer Graphics, Image Processing, and Vision*, Rio de Janeiro, Brazil, Oct. 20-23, 1998.
- [82] W. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.
- [83] S. Mehrotra, Y. Rui, M. Ortega-Binderberger, T.S. Huang, "Supporting content-based queries over images in MARS," *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pp. 632-3, Ottawa, Ont., Canada 3-6 June 1997.
- [84] Y. Meyer, *Wavelets Algorithms and Applications*, SIAM, Philadelphia, 1993.
- [85] S. Mukherjea, K. Hirata, Y. Hara, "AMORE: a World Wide Web image retrieval engine," *World Wide Web*, vol. 2, no. 3, pp. 115-32, Baltzer, 1999.
- [86] A. Natsev, R. Rastogi, K. Shim, "WALRUS: A similarity retrieval algorithm for image databases," *SIGMOD*, Philadelphia, PA, 1999.
- [87] A. Newell, H.A. Simon, "GPS, a program that simulates human thought," *Lernende Automaten*, H. Billing (ed.), pp. 109-124, R. Oldenbourg, Munich, Germany, 1961.

- [88] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, "The QBIC project: querying images by content using color, texture, and shape," *Proc. SPIE - Int. Soc. Opt. Eng., in Storage and Retrieval for Image and Video Database*, vol. 1908, pp. 173-87, San Jose, February, 1993.
- [89] A. Pentland, R. W. Picard, S. Sclaroff, "Photobook: tools for content-based manipulation of image databases," *SPIE Storage and Retrieval Image and Video Databases II*, vol. 2185, pp. 34-47, San Jose, February 7-8, 1994.
- [90] S. M. Perlmutter, P.C. Cosman, R.M. Gray, R.A. Olshen, D. Ikeda, C.N. Adams, B.J. Betts, M.B. Williams, K.O. Perlmutter, J. Li, A. Aiyer, L. Fajardo, R. Birdwell, B.L. Daniel, "Image quality in lossy compressed digital mammograms," *Signal Processing, Special Issue on Medical Image Compression*, Elsevier, vol. 59, no. 2, pp. 189-210, June 1997.
- [91] R. W. Picard, T. Kabir, "Finding similar patterns in large image databases," *IEEE ICASSP*, Minneapolis, vol. V, pp. 161-64, 1993.
- [92] J. B. Ramsey, D. Usikov, G. M. Zaslavsky, "An analysis of U.S. stock price behavior using wavelets," New York University, C.V. Starr Center *Economic Research Report*, no. 94-06, pp. 18, February, 1994.
- [93] J. B. Ramsey, C. Lampart, "Decomposition of economic relationships by timescale using wavelets: money and income," *Macroeconomic Dynamics*, vol. 2 no. 1 1365-1005, March 1998.
- [94] E. A. Riskin, "Variable rate vector quantization of images," *Ph.D Dissertation*, Stanford University, 1990.
- [95] E. A. Riskin, R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers," *IEEE Trans. Signal Process.*, November 1991.
- [96] Y. Rubner, L. J. Guibas, C. Tomasi, "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval," *Proceedings of the ARPA Image Understanding Workshop*, pp. 661-668, New Orleans, LA, May 1997.

- [97] Y. Rubner, *Perceptual Metrics for Image Database Navigation*, Ph.D. Dissertation, Computer Science Department, Stanford University, May 1999.
- [98] A. Said, W. A. Pearlman, "An image multiresolution representation for lossless and lossy image compression," *IEEE Transactions on Image Processing*, Sept. 1996.
- [99] P. Schroder, W. Sweldens, "Spherical wavelets: efficiently representing functions on the sphere," *Computer Graphics, (SIGGRAPH '95 Proceedings)*, 1995.
- [100] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.
- [101] G. Sheikholeslami, W. Chang, A. Zhang, "Semantic clustering and querying on heterogeneous features for visual data," *ACM Multimedia*, pp. 3-12, Bristol, UK, 1998.
- [102] J. Shi, J. Malik, "Normalized cuts and image segmentation," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 731-7, San Juan, Puerto Rico, June, 1997.
- [103] E. H. Shortliffe, *Computer Based Medical Consultations: MYCIN*, Elsevier, New York, 1976.
- [104] C. R. Shyu, A. Kak, C. Brodley, L. Broderick, "Testing for human perceptual categories in a physician-in-the-loop CBIR system for medical imagery," *Proc. of the IEEE Workshop of Content-Based Access of Image and Video Databases*, Fort Collins, Colorado, June 1999.
- [105] H. A. Simon, "Why should machines learn?" *Machine learning : an artificial intelligence approach*, R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds), Palo Alto, Tioga Pub. Co., Palo Alto, Calif., 1983.

- [106] J. R. Smith, S.-F. Chang, "An image and video search engine for the World-Wide Web," *Storage and Retrieval for Image and Video Databases V (Sethi, I K and Jain, R C, eds)*, *Proc SPIE 3022*, pp. 84-95, 1997.
- [107] J. R. Smith, C. S. Li, "Image classification and querying using composite region templates," *Journal of Computer Vision and Image Understanding*, vol. 75, no. 1-2, pp. 165-74, Academic Press, 1999.
- [108] G. W. Snedecor, W. G. Cochran, *Statistical Methods*, Iowa State University Press, Ames, Iowa, 1989.
- [109] *Special Issue on Wavelets and Signal Processing, IEEE Trans. Signal Processing*, vol. 41, Dec. 1993.
- [110] S. Stevens, M. Christel, H. Wactlar, "Informedia: improving access to digital video," *Interactions*, vol. 1, no. 4, pp. 67-71, 1994.
- [111] M.J. Swain, D. H. Ballard, "Color indexing," *Int. Journal of Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.
- [112] M. Szummer, R. W. Picard, "Indoor-outdoor image classification," *Int. Workshop on Content-based Access of Image and Video Databases*, pp. 42-51, Jan. 1998.
- [113] J. K. Udupa, L. Wei, S. Samarasekera, Y. Miki, M. A. van Buchem, R. I. Grossman, "Multiple sclerosis lesion quantification using fuzzy-connectedness principles," *IEEE Trans Med Imaging*, vol. 16, no. 5, pp. 598-609, Oct. 1997.
- [114] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, no. 11, pp. 1549-1560, Nov. 1995.
- [115] A. Vailaya, A. Jain, H. J. Zhang, "On image classification: city vs. landscape," *Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries*, pp. 3-8, Santa Barbara, CA, 21 June 1998.

- [116] M. Vetterli, J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, New Jersey, 1995.
- [117] J. Villasenor, B. Belzer, J. Liao, "Wavelet filter evaluation for image compression," *IEEE Transactions on Image Processing*, vol. 2, pp. 1053-1060, August 1995.
- [118] J. Z. Wang, G. Wiederhold, O. Firschein, X. W. Sha, "Content-based image indexing and searching using Daubechies' wavelets," *International Journal of Digital Libraries*, vol. 1, no. 4, pp. 311-328, 1998.
- [119] J. Z. Wang, J. Li, G. Wiederhold, O. Firschein, "System for screening objectionable images," *Computer Communications Journal*, vol. 21, no. 15, pp. 1355-60, Elsevier Science, 1998.
- [120] J. Z. Wang, J. Li, R. M. Gray, G. Wiederhold, "Unsupervised multiresolution segmentation for images with low depth of field," *IEEE PAMI*, 2000, to appear.
- [121] J. Z. Wang, G. Wiederhold, J. Li, "Wavelet-based progressive transmission and security filtering for medical image distribution," *Medical Image Databases*, S. Wong (Ed.), Kluwer International Series in Engineering and Computer Science, Secs 465, pp. 303-324, Kluwer Academic, Boston, 1998.
- [122] J. Z. Wang, G. Wiederhold, O. Firschein, X. W. Sha, "Wavelet-based image indexing techniques with partial sketch retrieval capability," *Proceedings of the 4th Forum on Research and Technology Advances in Digital Libraries (ADL'97)*, pp. 13-24, Washington D.C., May 1997.
- [123] J. Z. Wang, G. Wiederhold, O. Firschein, "System for screening objectionable images using Daubechies' wavelets and color histograms," *Interactive Distributed Multimedia Systems and Telecommunication Services, Proceedings of the 4th International Conference (IDMS'97)*, Ralf Steinmetz and Lars C. Wolf (Eds.), Darmstadt, Germany, pp. 20-30, Springer-Verlag LNCS 1309, September 1997.



- [124] J. Z. Wang, M. Bilello, G. Wiederhold, "A textual information detection and elimination system for secure medical image distribution," *Proceedings of the 1997 American Medical Informatics Association (AMIA) Annual Fall Symposium (formerly SCAMC), Journal of AMIA supplement*, Nashville, Tennessee, October 1997.
- [125] J. Z. Wang, J. Li, G. Wiederhold, O. Firschein, "System for classifying objectionable websites," *Interactive Distributed Multimedia Systems and Telecommunication Services, Proceedings of the 5th International Conference (IDMS'98)*, Thomas Plagemann and Vera Goebel (Eds.), Oslo, Norway, pp. 113-124, Springer-Verlag LNCS 1483, September 1998.
- [126] J. Z. Wang, G. Wiederhold, "WaveMark: digital image watermarking using Daubechies' wavelets and error correcting coding," *Proceedings of SPIE Symposium of Voice, Video, and Data Communications*, vol. 3528, pp. 432-9, Boston, November 1998.
- [127] J. Z. Wang, G. Wiederhold, "System for efficient and secure distribution of medical images on the internet," *Proceedings of the 1998 American Medical Informatics Association (AMIA) Annual Fall Symposium, Journal of AMIA supplement*, pp. 907-11, Orlando, Florida, November 1998.
- [128] J. Z. Wang, M. A. Fischler, "Visual similarity, judgmental certainty and stereo correspondence," *Proceedings of DARPA Image Understanding Workshop*, pp. 1237-48, Monterey, 1998.
- [129] J. Z. Wang, J. Nguyen, K.-K. Lo, C. Law, D. Regula, "Multiresolution browsing of pathology imaging using wavelets," *Proceedings of the 1999 American Medical Informatics Association (AMIA '99) Annual Fall Symposium, Journal of AMIA supplement*, pp. 340-344, Washington, D.C., November, 1999.
- [130] J. Z. Wang, J. Li, D. Chan, G. Wiederhold, "Semantics-sensitive retrieval for digital picture libraries," *D-LIB Magazine*, vol. 5, no. 11, DOI:10.1045/november99-wang, November, 1999. <http://www.dlib.org>

- [131] J. Z. Wang, "Pathfinder: multiresolution region-based searching of pathology images using IRM," *Proceedings of the 2000 American Medical Informatics Association (AMIA'00) Annual Fall Symposium, Journal of AMIA supplement*, Los Angeles, November, 2000.
- [132] G. Wiederhold, "Mediators in the architecture of future information systems", *IEEE Computer*, vol. 25, pp. 38-49, 1992.
- [133] G. Wiederhold, *Intelligent Integration of Information*, Kluwer Academic, Boston, 1996.
- [134] S.T.C. Wong (ed.), *Medical Image Databases*, Kluwer Academic, Boston, 1998.
- [135] G. Wyszecki, W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, John Wiley and Sons, New York, NY, 1982.
- [136] S. C. Zhu, A. Yuille, "Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884-900, 1996.
- [137] <http://www.mathsoft.com/wavelets.html>

# Index

- AMORA, 6
- aperiodic signal, 39
- artificial intelligence, 49
- artificial neuron, 49
  
- band, 43
- basis function, 36
- benign, 10, 33
- bioinformatics, 5
- Blobworld, 6, 25, 30, 145
- Boolean query, 20
  
- CART, 13, 56
- CBIR, 3, 16
  - challenges, 17, 31
  - evaluation, 21
- CIE, 27
- CIE Lab, 27
- city vs. landscape, 34
- classification, 9, 16, 32, 48, 56, 94
  - medical, 34
- clinical decision making, 4
- clinical diagnosis, 4
- clinical trial, 4
- clustering, 109
- CMY, 26
- CMYK, 26
- coefficient, 36
- color channel, 25
- color layout, 24, 29
- color space, 25, 66
- compression, 37
  - codec, 37
  - progressive, 46, 108
  - wavelet, 46
- Computed Tomography, 4
- computer vision, 17
- content-based image retrieval, 3, 16
- copy detection, 7
- COREL, 112
- CRT, 31
- CT, 4
  
- database, 2
- Daubechies, 40
- DCT, 37
- DENDRAL, 50
  
- education, 5
- electrophoresis gel, 5
- EMD, 6
- Euclidean distance, 50
- evaluation, 21
  
- face detection, 34
- feature, 24
- fluctuation, 41
- Fourier transform, 37
- frequency, 36
  
- Gaussian mixture model, 56
- gold standard, 21
- GPS, 50
- graph, 9, 33, 97
- gray-scale, 25
  
- Haar, 39
- HCI, 7
- high-pass filter, 43

- histogram, 24, 27, 134
- HLS, 26
- HSV, 26
- HTML, 118
- HTTP, 117
- hue, 26
- HVS, 24, 37, 92
  
- IBCOW, 161
- image retrieval, 2
- indexing, 23
- Informedia, 6
- Integrated Region Matching, 10, 97
- interchange color space, 27
- interface, 115
- IRM, 10, 97
  
- JAVA, 116
- JPEG, 37, 112
  
- k-means, 13, 51, 92
- k-nearest neighbor, 56
- kernel method, 56
  
- learning, 49
- lesion, 5
- lightness, 26
- LISP, 49
- localization, 36
- low-pass filter, 43
- LUV, 27
- LVQ, 54
  
- machine learning, 49
- Magnetic Resonance Imaging, 4
- mammography, 4, 34, 46
- Manhattan distance, 50
- MARS, 6
- mediator, 4
- MRI, 4
- multiple sclerosis, 4, 63
- MYCIN, 50
  
- NeTra, 6, 25, 30
- neuroradiology, 32
  
- objectionable, 5, 10, 33
  
- PACS, 4
- PATHFINDER, 14, 106, 111
- pathology, 5, 31
- patient care digital library, 3
- pattern library, 31
- PBR, 31
- perception, 27
- Photobook, 6
- photograph, 9, 33, 97
- Picture Archive and Communications Systems, 4
- precision, 21, 128
- primitive feature, 18
- progressive, 118
- progressive transmission, 108
- prune, 59
  
- QBIC, 6, 78
- quadratic mirror filter, 41
- query, 18, 115
  - partial, 76
  
- recall, 21, 128
- region-based search, 24, 29
- relevance feedback, 21
- resolution, 17
- retrieval, 23
- RGB, 26
  
- saturation, 26
- security, 5
- segmentation, 10, 29, 34, 92, 107
- semantics, 16, 19, 32
- semantics-sensitive image retrieval, 7
- signature, 23
- similarity measure, 10, 24
- similarity metric, 97

- SIMPLIcity, 8, 86, 111
- spatial region, 36
- speech recognition, 47
- spline, 37
- splitting criterion, 58
- statistical classification, 13, 48
- statistical clustering, 13, 48, 50
- surrounding text, 6
  
- text-based image retrieval, 2
- textured, 9, 94
- time, 36
- transform, 36
- trend, 40
- TSVQ, 13, 53, 109
  
- understandability, 17
- user interface, 20
  
- vector quantization, 54
- video indexing and retrieval, 3
- VIRAGE, 6, 78
- VisualSEEK, 6
- VQ, 54
  
- WALRUS, 29
- wavelet, 12, 64, 108
  - application, 44
  - band, 43
  - father wavelet, 41
  - mother wavelet, 41
  - prototype function, 41
- wavelet transform, 39
- WBIIS, 6, 29, 64
- Web, 6
- WebSEEK, 6
- weighted precision, 129
- WFT, 39
- Windowed Fourier Transform, 39
- WIPE, 33, 153
- World-Wide Web, 6
- WWW, 6
  
- X-ray, 4
- XYZ, 27
  
- zero-tree, 46