Content-based image indexing and retrieval using computational intelligence.

Wu, Kui.

2008

Wu, K. (2008). Content-based image indexing and retrieval using computational intelligence. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41417>

<https://doi.org/10.32657/10356/41417>

Content-Based Image Indexing and Retrieval Using Computational Intelligence

Wu Kui

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2008

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

.....28/02/08.....
Date

.....Wu kui.....
WU KUI

Acknowledgements

First and foremost I would like to express my deepest gratitude to my supervisor, Dr. Yap Kim-Hui, for his insightful guidance, kind encouragement and inspiring support. Without him this thesis would be impossible in many ways. He has always provided his technical knowledge, insightful perspectives and sincere encouragement throughout my research work. His wisdom, broad vision and kindness are invaluable to me.

I also would like to acknowledge Nanyang Technological University for awarding me the research scholarship and providing me with excellent research facilities. In addition, I am grateful to my colleagues in Media Technology Laboratory for their collaboration and discussion. My appreciation is also extended to all my friends for their sincere friendship and great help. Many thanks to all the people who have directly or indirectly helped me in my PhD study.

Last but not least, I would like to thank my parents, my elder sister and my husband. Their endless love, unselfish care, and unconditioned support are the greatest inspiration for me in my life. I dedicate this thesis to all of you.

Summary

The significant growth in the volume of image data has driven the demand for efficient techniques to index and access the image collections. These techniques are used in fields including applications such as online image libraries, e-commerce, biomedicine, military and education, among others. In view of this, content-based image retrieval (CBIR) has been developed as a scheme for managing, searching, filtering, and retrieving the image collections. CBIR is a process of retrieving a set of desired images from the database on the basis of visual content such as color, texture, shape, and spatial relationship that are present in the images. The problem is challenging due to the semantic gap between the low-level visual features and the high-level human perception. With the objective to reduce the semantic gap, this thesis investigates several challenging problems in current CBIR systems. It covers the following three main aspects: relevance feedback in CBIR (Chapters 4 and 5), relevance feedback in region-based image retrieval (Chapter 6), and peer tagging and knowledge propagation (Chapter 7).

The first contribution of this thesis is the development of a new soft relevance framework in interactive CBIR systems. A soft relevance notion is proposed to integrate the users' fuzzy perception of visual content into the framework of relevance feedback. A progressive fuzzy radial basis function network (PFRBFN) is proposed to learn the user information need by optimizing a cost function. A gradient descent-based learning strategy is then employed to estimate the underlying network parameters due to its algorithmic simplicity.

The second contribution of the thesis is the proposal of a new pseudo-label fuzzy support

vector machine (PLFSVM) framework to perform content-based image retrieval. The technique addresses the small sample problem by incorporating pseudo-labeled images to enlarge the training data set. An unsupervised clustering algorithm is used to select pseudo-labeled images by studying the characteristics of the labeled images. The relevance of the pseudo-labeled images is estimated using the fuzzy membership function, and integrated into the fuzzy support vector machine (FSVM) for learning.

The third contribution of this thesis is the development of a new framework that integrates relevance feedback into region-based image retrieval (RBIR) systems. A variable-length radial basis function network (VLRBFN) is constructed and progressively trained to achieve improved retrieval results. A new kernel function of the VLRBFN centered on region-based representation is introduced to handle variable-length computation of image similarity. An unsupervised clustering algorithm is developed for VLRBFN center selection based on the characteristics of region-based representation. The importance of the constituent regions in each RBF unit is estimated using a hybrid of probabilistic estimation and perceptual determination scheme. A gradient descent-based learning strategy is then employed to estimate the underlying network parameters.

The last part of this thesis deals with the issue of semantic gap from the perspective of peer tagging and knowledge propagation. Unique issues associated with peer tagging are discussed including human-computer interface (HCI) for image tagging, tag generation and formation, and tag clustering. In contrast to the conventional labor-intensive process of manual image annotation, peer tagging provides a practical alternative for keyword annotation. Further, a knowledge propagation scheme is presented that automatically propagates keywords from a subset of labeled images to the unlabeled images. It is based on image content analysis and training of keyword classifiers. In particular, genetic algorithm (GA) is utilized to find the salient regions in the labeled images of the same semantic concepts. Since

different regions in an image have unequal importance for computing image similarity, a proper weight has to be assigned to each constituent region in the images. The importance of the unseen regions that haven't been assigned proper weights is then estimated by a one-class support vector machine (OCSVM). Next, radial basis function (RBF)-based classifiers are trained based on the content of the labeled images. Finally, the trained classifiers are used for keywords propagation.

Contents

Acknowledgements	i
Summary	ii
List of Figures	ix
List of Tables	xiii
List of Abbreviations	xiv
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	6
1.2.1 User Perception	6
1.2.2 Small Sample Problem	7
1.2.3 Relevance Feedback in Region-based Image Retrieval (RBIR)	7
1.2.4 Peer Tagging and Knowledge Propagation	9
1.3 Computational Intelligence	10
1.3.1 Neural Networks	11
1.3.2 Support Vector Machines (SVMs)	12
1.3.3 Fuzzy Logic	14
1.3.4 Evolutionary Computation	15
1.4 Contributions of Thesis	15
1.5 Organization of Thesis	18
2 Review of Content-based Image Retrieval	20
2.1 Overview	20
2.2 Feature Extraction and Representation	22
2.3 Similarity Measures in CBIR Systems	25
2.4 Relevance Feedback in CBIR Systems	25

2.4.1	Previous Work on Relevance Feedback	26
2.4.2	Relevance Judgement in CBIR Systems	27
2.4.3	Active Learning.....	28
2.4.4	Incorporating Unlabeled Images into CBIR Systems	28
2.5	Region-based Image Retrieval	29
2.5.1	Similarity Measures in RBIR Systems.....	29
2.5.2	Relevance Feedback in RBIR Systems	31
2.6	Automatic Image Annotation	32
2.7	Performance Evaluation.....	34
3	Backgrounds and Foundations	36
3.1	Image Content Description	36
3.2	Radial Basis Function (RBF) Network.....	41
3.3	Support Vector Machines (SVMs)	46
4	Soft Relevance Framework in Interactive Content-based Image Retrieval	50
4.1	Introduction	50
4.2	Fuzzy Relevance Feedback in CBIR	52
4.2.1	Hierarchical User Information Priority	52
4.2.2	Fuzzy Relevance Feedback.....	54
4.3	Progressive Fuzzy Radial Basis Function Network (PFRBFN).....	56
4.3.1	User Interface	56
4.3.2	Structure of PFRBFN	57
4.3.3	Two-stage Clustering for PFRBFN Subnet Creation	61
4.3.4	Estimation of Soft Relevance Membership Function for Fuzzy Images.....	66
4.3.5	PFRBFN Training Algorithm	69
4.4	Experimental Results	74
4.4.1	Performance Evaluation based on Objective Measure.....	75
4.4.2	Performance Evaluation based on Subjective Measure	80
4.4.3	Computational Efficiency Analysis	83
4.4.4	Illustrations of Retrieval Results Using PFRBFN	83
4.5	Summary	87
5	A Pseudo-Label Fuzzy Support Vector Machine Framework	88
5.1	Introduction	88
5.2	Learning from Small Samples.....	90

5.2.1	Small Sample Problem	90
5.2.2	Active Learning with Pseudo-Labeled Images	91
5.3	Pseudo-Label Fuzzy Support Vector Machine (PLFSVM)	93
5.3.1	Overview of the Proposed Framework.....	93
5.3.2	Formulation of FSVM	94
5.3.3	Unlabeled Image Selection and Pseudo-Label Estimation.....	95
5.3.4	Estimation of Soft Relevance Membership Function for Pseudo-Labeled Images.....	97
5.4	Experimental Results	99
5.5	Summary	103
6	Region-based Image Retrieval Using Variable-Length Radial Basis Function Network	104
6.1	Introduction	104
6.2	Overview of the Proposed Region-based Image Retrieval System.....	106
6.2.1	Overview of the System	106
6.2.2	Image Segmentation and Region-based Image Representation.....	107
6.2.3	Image Similarity Measure.....	110
6.3	Progressive Variable-Length Radial Basis Function Network (VLRBFN).....	111
6.3.1	Structure of Progressive VLRBFN	111
6.3.2	Modified Subtractive Clustering for VLRBFN Center Selection.....	115
6.3.3	Perceptual Determination and Density Estimation for Region Weight Learning	116
6.3.4	VLRBFN Training Algorithm	120
6.4	Experimental Results	122
6.4.1	Performance Comparison based on Precision versus Recall Measure.....	123
6.4.2	Performance Comparison based on Retrieval Accuracy Measure.....	126
6.4.3	Computational Efficiency Analysis	128
6.5	Summary	129
7	Peer Tagging and Knowledge Propagation	130
7.1	Introduction	130
7.2	Peer Tagging	133
7.2.1	Human-Computer Interface (HCI) for Image Tagging	133
7.2.2	Tag Generation and Formation	134
7.2.3	Tag Clustering.....	135

7.3	Knowledge Propagation.....	137
7.3.1	Overview of the Proposed Knowledge Propagation Scheme	137
7.3.2	GA for Salient Region Selection.....	140
7.4	Experimental Results	146
7.5	Summary	149
8	Conclusion and Recommendations	150
8.1	Conclusion	150
8.1.1	Relevance Feedback in CBIR	150
8.1.2	Relevance Feedback in RBIR	151
8.1.3	Peer Tagging and Knowledge Propagation	152
8.2	Recommendations for Further Research.....	154
8.2.1	Content-Based Video Retrieval (CBVR)	154
8.2.2	Knowledge Discovery through Content and Keyword Fusion	155
8.2.3	CBIR in Domain-Specific Applications	155
	Author's Publications	157
	Bibliography	159
	Appendix A	173

List of Figures

Figure 1.1 Manual annotation of an image is highly subjective.2

Figure 1.2 “An image is worth a thousand words”: it is hard to describe all the content in the images using only a few keywords.2

Figure 1.3 An image with complex textures that is hard to describe using keywords.....3

Figure 1.4 CBIR system architecture.4

Figure 1.5 The semantic gap in CBIR systems.....5

Figure 1.6 Two images that are visually dissimilar but semantically similar.5

Figure 1.7 Two images that are visually similar but semantically dissimilar.5

Figure 1.8 Two images containing similar object of eagles but with different background.8

Figure 1.9 The four categories of computational intelligence techniques..... 11

Figure 1.10 Diagram of a typical neural network with one hidden layer..... 12

Figure 1.11 The basic idea of SVM: map the nonlinearly separable training data into a higher dimensional feature space via kernel function Ψ where the training data is separable, and construct an optimal separating hyperplane there. This yields a nonlinear decision boundary in the input space..... 13

Figure 1.12 Optimal separating hyperplane. (a) A separating hyperplane with small margin, (b) The optimal separating hyperplane with maximal margin. A better generalization capability is expected from (b). 14

Figure 1.13 Structured levels of image content analysis. 16

Figure 2.1 Interactive CBIR system architecture.....	21
Figure 2.2 Illustration of relevance feedback process.....	26
Figure 2.3 Interactive RBIR system architecture.....	29
Figure 2.4 Imperfect image segmentation. (a) Over-segmentation, (b) Under-segmentation.	30
Figure 3.1 Illustration of wavelet transform using a three-level decomposition.....	38
Figure 3.2 The architecture of RBF network with a single output.	42
Figure 3.3 The optimal separating hyperplane $\mathbf{w} \cdot \mathbf{z} + b = 0$ with maximal margin $\frac{2}{\ \mathbf{w}\ }$	47
Figure 4.1 Scenario of user ambiguity occurring during the feedback process.....	51
Figure 4.2 Hierarchical structure of user information priorities.	54
Figure 4.3 Illustration of user interface.	57
Figure 4.4 Illustration of image similarity and query characterization in the feature space. (a) Global modeling of image similarity using a single model, (b) Multi-cluster local modeling of image relevance.....	58
Figure 4.5 Example images that represent the semantic “dog”.....	58
Figure 4.6 Schematic diagram of PFRBFN. (a) Recursive structure of PFRBFN, (b) Update network.....	60
Figure 4.7 General overview of the proposed PFRBFN scheme.....	73
Figure 4.8 Selected sample images from the database, which are used in the experiments....	74
Figure 4.9 The average precision versus recall graph of the PFRBFN and ARBFN methods (after 5 feedback iterations).	77
Figure 4.10 Retrieval accuracy of the PFRBFN and ARBFN methods. (a) Retrieval accuracy in top 25 results, (b) Retrieval accuracy in top 50 results.	79

Figure 4.11 Performance comparison of PFRBFN, ARBFN and MARS based on user subjectivity. (a) Retrieval accuracy in top 25 results, (b) Retrieval accuracy in top 50 results.	82
Figure 4.12 Illustrative case study. (a) Initial retrieval results based on K-NN search, (b) Retrieval results with user marking the cat images as irrelevant, (c) Retrieval results with user marking the cat images as relevant, (d) Retrieval results with user marking the cat images as fuzzy.	86
Figure 5.1 The average precision-versus-recall graphs (after the first iteration of active learning). (a) APR for 5 initial labeled images, $l_0 = 5$, (b) APR for 10 initial labeled images, $l_0 = 10$	101
Figure 5.2 Retrieval accuracy of the PLFSVM and SVM methods. (a) Retrieval accuracy in top 10 results, (b) Retrieval accuracy in top 20 results.	102
Figure 6.1 Illustration of feature extraction in the VLRBFN system.....	109
Figure 6.2 Schematic diagram of VLRBFN. (a) Recursive structure of VLRBFN, (b) Update network.	113
Figure 6.3 Examples of segmentation results by the mean shift algorithm. Original images are on the left, the corresponding segmented images are on the right. (a) Tiger, (b) Lion, (c) Woman, (d) Cat.....	123
Figure 6.4 The APR graphs in top 25 returned images. (a) APR after 1 feedback iteration, (b) APR after 5 feedback iterations.	125
Figure 6.5 Retrieval accuracy in top 25 returned images.	127
Figure 7.1 The tag cloud on Flickr.com.	135
Figure 7.2 Illustrative example of tag clusters in Flickr.com.	136
Figure 7.3 Schematic overview of the knowledge propagation scheme.	138
Figure 7.4 Illustration of SUS.	144

Figure 7.5 Single-point crossover operator.	145
Figure 7.6 Mutation operator.	145

List of Tables

Table 2.1 Commonly used features and their representations in CBIR systems.22

Table 3.1 Image features used in the experiment.....40

Table 7.1 Procedure of SGA.....141

Table 7.2 Selection probabilities and fitness values of 10 individuals.144

Table 7.3 Keyword list of the 100 image categories.....147

Table 7.4 Results of average annotation accuracy.149

List of Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
2D MHMM	two-dimensional multi-resolution hidden Markov model
APR	average precision versus recall
ARBFN	adaptive radial basis function network
BPM	Bayes point machine
CBIR	content-based image retrieval
CBSA	content-based soft annotation
CBVR	content-based video retrieval
CDE	confidence-based dynamic ensemble
CMRM	cross-media relevance model
CT	computer tomography
D-EM	Discriminant Expectation Maximization
DWT	discrete wavelet transform
EM	Expectation Maximization
EMD	Earth Mover's Distance
FCH	fuzzy color histogram
FCM	fuzzy C-means
FSVM	fuzzy support vector machine
GA	genetic algorithm
HCI	human-computer interface
IRM	integrated region matching
K-NN	k-nearest neighbors
KT	Kuhn-Tucker
LDA	Latent Dirichlet Allocation
LSA	latent semantic analysis
MIL	multiple instance learning
MSHP	most similar highest priority
OCSVM	one-class support vector machine

PFRBFN	progressive fuzzy radial basis function network
PLFSVM	pseudo-label fuzzy support vector machine
PLSA	probabilistic latent semantic analysis
PR	precision versus recall
QBE	query-by-example
QP	quadratic programming
QPM	query point movement
RA	retrieval accuracy
RBIR	region-based image retrieval
RBF	radial basis function
RAA	relaxed annotation accuracy
RRA	relevant retrieval accuracy
SAA	strict annotation accuracy
SGA	simple genetic algorithm
SRM	structural risk minimization
SUS	stochastic universal sampling
SVMs	support vector machines
TRA	total retrieval accuracy
TSVM	transductive support vector machine
UFM	unified feature matching
VC	Vapnik-Chervonenkis
VLR	variable-length representation
VLRBFN	variable-length radial basis function network

Chapter 1

Introduction

1.1 Motivation

The volume of image data has grown tremendously in recent years due to increasing popularity of imaging devices such as digital cameras, prevalence of low-cost high-capacity storage devices, and the increasing proliferation of image data over the Internet and wireless networks. This is reflected in many applications including online image libraries, e-commerce, biomedicine, military and education, among others. This motivates the development of efficient image indexing and retrieval systems and algorithms.

Traditional image retrieval systems use textual descriptors such as keywords to annotate images. This is, however, exhausting and time consuming as they involve a significant amount of human labor in manual annotations of large-scale image databases. Further, because of human perceptual subjectivity, the annotations are highly individual and subjective. Take the image in Figure 1.1 for example, different annotators may assign different keywords to it. Some may annotate it using keywords such as “lake”, “house” or “tree”, which are the objects in the image. Others may annotate it as “autumn” or “scenery” by interpreting the whole image. Manual annotation also experiences the difficulty of describing the rich content of images using only a few keywords since “an image is worth a thousand words” (Figure 1.2). Therefore, the limited keywords assigned to an image are often incomplete. Another limitation in text-based image retrieval systems comes from the formulation of a query. Often,

there exist queries that are hard to describe using keywords. For instance, a user may be looking for an image with complex textures having a particular combination of colors as shown in Figure 1.3. It would be difficult to describe such an image using a small number of keywords.

“Lake, house, tree, autumn, scenery...”

Figure 1.1 Manual annotation of an image is highly subjective.

Figure 1.2 “An image is worth a thousand words”: it is hard to describe all the content in the images using only a few keywords.

Figure 1.3 An image with complex textures that is hard to describe using keywords.

In view of all these problems, content-based image retrieval (CBIR) systems have been developed to address these shortcomings. CBIR attempts to retrieve a set of desired images from the database using visual features such as color, texture, shape, and spatial relationships that are present in the images. The process of feature extraction and image indexing can be fully automated, hence alleviating the intensive human labor involved in manual image annotations in traditional text-based retrieval systems. Since the images are represented by their own features, the subjectivity involved in the description of images by keywords can be avoided. Further, CBIR offers an alternative approach to perform a query. It allows the users to submit a query to the system in the form of an image, thus reducing the need to provide textual descriptions for a query which may be hard to describe using keywords. Figure 1.4 illustrates how typical CBIR systems work. Users submit their query to the systems. Features are then extracted from the query and compared with the features of the images in the database. After similarity comparison, the systems display the retrieval results to the users.

Figure 1.4 CBIR system architecture.

Many research and commercial CBIR systems have been developed, including QBIC [1], MARS [2], Virage [3], Photobook [4], VisualSEEK [5], PicToSeek [6], and PicHunter [7]. Despite these research efforts, the retrieval performance of these systems remains relatively unsatisfactory. This is mainly due to the semantic gap between the low-level visual features (color, texture, shape) and high-level human perception. Semantic gap is the uncertainty in the correspondence between (i) the information that one can extract from the visual data and (ii) the interpretation that the same data have for a user in a given situation [8]. An illustrative example of the semantic gap is given in Figure 1.5. Users interpret an image based on the high-level concept such as flower, rose or plant. In contrast, the retrieval systems represent the image based on its low-level visual features such as red color, ruffled texture, and round shape. This results in the semantic gap between the low-level features and high-level human perception. Let us have a look at two examples in Figure 1.6 and Figure 1.7, which illustrate the different interpretation criteria utilized by the retrieval systems and the users. Suppose only color features are used in the retrieval systems. The retrieval systems will judge the two images in Figure 1.6 as visually dissimilar (colors are different) and images in Figure 1.7 as visually similar (colors are the same). However, the users will regard images in Figure 1.6 as

semantically similar (both are flowers) while images in Figure 1.7 as semantically dissimilar (flower and tomato).

Figure 1.5 The semantic gap in CBIR systems.

Figure 1.6 Two images that are visually dissimilar but semantically similar.

Figure 1.7 Two images that are visually similar but semantically dissimilar.

1.2 Objectives

The aim of this thesis is to develop efficient and user-friendly image retrieval systems, which are useful for real-life applications. One of the most challenging problems in building a successful image retrieval system lies in bridging the semantic gap. To narrow the semantic gap, relevance feedback has been introduced to learn the associations between the high-level concepts and the low-level features [2], [6]–[7], [9]–[22]. It is an interactive and iterative process to learn from the user’s feedbacks regarding the relevance of currently retrieved results to the original query. In CBIR systems with relevance feedback, several important problems need to be addressed and they are highlighted as follows.

1.2.1 User Perception

An important aspect in the formulation of relevance feedback is the consideration for the potential imprecision of user perception during the feedback process. So far, there have been limited efforts in addressing this issue. Most current relevance feedback methods employ binary labeling approach, namely, the users have to classify an image either as “fully relevant” or “totally irrelevant” [2], [9], [12]–[15], [17], [20]. This hard-decision approach does not reflect the nature of user interpretation and understanding of images, which tends to be uncertain or imprecise due to perceptual subjectivity. For instance, when a user would like to find a flower, particularly red in color, he/she will encounter a dilemma as whether to classify a pink flower as “fully relevant” or “totally irrelevant”. This is because the pink flower satisfies, up to a certain extent, his/her information need. It is, therefore, inappropriate to describe the feedback decision by binary crisp logic without considering the degree of relevance. There are a few other works which provide three-option classification. Although these classification schemes provide three options to the users, namely, “relevant”,

“irrelevant”, and “neutral”, they do not utilize the “neutral” images in learning the user information need, or in other words, they ignore the neutral images. This will lead to potential underutilization of the information available. In contrast, multi-level labeling categorizes the feedback samples into several discrete levels of (ir)relevance [10], [11], [22]. This technique, however, is both cumbersome as well as tedious as the users need to classify every image into one of the multiple levels. Therefore, the users’ imprecise perception should be taken into account in relevance feedback process.

1.2.2 Small Sample Problem

Conventional relevance feedback in CBIR systems uses only the labeled images for learning. Image labeling, however, is a time consuming task and users are often unwilling to label too many images during the feedback process. This gives rise to the small sample problem where learning from a small number of training samples restricts the retrieval performance. Therefore, it is imperative to find solutions to solve the small sample problem faced by relevance feedback.

1.2.3 Relevance Feedback in Region-based Image Retrieval (RBIR)

While users generally look for visual objects in images, the global features extracted from the whole image in CBIR systems sometimes fail to match the users’ object-level perceptions. For instance, suppose a user is looking for images of “eagles” regardless of their background, global image-based CBIR systems may not retrieve images of “eagles” as shown in Figure 1.8 because of their different background. Consequently, region-based image retrieval (RBIR) systems have been developed to represent images at the object level to achieve higher retrieval relevancy [5], [23]–[27]. They utilize image segmentation to decompose images into

regions, and then perform comparison of image similarity based on the local features extracted from the constituent regions.

Figure 1.8 Two images containing similar object of eagles but with different background.

User interpretation of visual content tends to vary with respect to different users and circumstances. In other words, the user perception of image similarity is subjective in nature. Each image, depending on the context, can have many different interpretations. For instance, given the same query image of “a model standing in front a car”, one user may be more interested in finding the images of models, while, another user may look for the images of cars. Even the same user may perceive the same image differently under various situations. However, some existing RBIR systems adopt fixed image similarity metric [5], [23]–[27]. They ignore the contextual dependency of visual content, therefore, failing to capture the user perceptual subjectivity. To address this issue, relevance feedback has been introduced to overcome some of the limitations associated with RBIR systems [28]–[38]. Relevance feedback is traditionally used in CBIR systems that use global features. There are still relatively few works that incorporate it in the RBIR systems. Nevertheless, some preliminary works have been done to combine RBIR with relevance feedback to improve the retrieval performance. Despite the previous works on relevance feedback for RBIR systems, it is still a challenging task to develop effective and efficient interactive mechanisms to yield satisfactory

retrieval performance. Some issues that need to be resolved include: (i) the definition of image similarity measure to decrease the impact of imprecise segmentation, (ii) the learning algorithm for region-based representation to progressively improve the retrieval accuracy through interaction with the users, and (iii) the tradeoff between retrieval effectiveness and realistic real-time requirement.

1.2.4 Peer Tagging and Knowledge Propagation

Retrieving images via low-level features alone cannot achieve satisfactory results. It would be advantageous to build interactive CBIR systems which support high-level semantic query. The main idea is to integrate the strengths of content- and keyword-based image indexing and retrieval algorithms while alleviating their respective difficulties. A major difficulty of conventional manual image annotation is that it is laborious and time consuming especially for large image databases. Recently, a new trend in image annotation has emerged over the past 2-3 years. This involves the paradigm of peer or community tagging where distributed users (particularly Internet users) add tags (keywords) to community-shared images. In peer tagging, individual users can apply arbitrary tags to images that they have uploaded for sharing, or participate in voluntary tagging of images that are of interests to them. In contrast to personal annotation, peer tagging potentially distributes the workload of annotation amongst multiple voluntary contributors, hence making the process easy and flexible. Online media (image and video) tagging systems have emerged recently and become increasingly popular such as Flickr.com [39], Smugmug.com [40], and YouTube.com [41]. Nevertheless, these services do not explore the correlation between the tags and the media content, hence underutilizing the information available. Further, they do not address the problem where only a fraction of the images or videos out of the complete collection is annotated. Therefore, the unique issues associated with the new paradigm of online peer tagging need to be investigated

so that it can be incorporated into CBIR systems to facilitate retrieval. New approaches in knowledge propagation are also required to propagate keywords from the annotated images to the rest of the unannotated images in the collection.

1.3 Computational Intelligence

To address the above-mentioned problems, it is important for the retrieval systems to have some sort of intelligence to simulate and understand the user information need. Computational intelligence, which offers an instrumental mathematical modeling of artificial intelligence, is employed in this thesis due to its effectiveness. Computational intelligence is a field of study that attempts to simulate human intelligence using computational algorithms [42]–[43]. Its primary objective is to incorporate the principles and mechanisms behind the intelligent behaviors both in natural and artificial systems. Computational intelligence hypothesizes that the basic reasoning behind simulated intelligence lies in its computation. Four categories of computational intelligence techniques, namely neural networks, support vector machines, fuzzy logic, and evolutionary computation, are employed in this thesis as shown in Figure 1.9.

Figure 1.9 The four categories of computational intelligence techniques.

1.3.1 Neural Networks

Neural networks have been developed as generalizations of mathematical models of human neural biology [44]–[45]. A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use [44]. It resembles the brain in two respects: (i) knowledge is acquired by the network from its environment through a learning process, (ii) interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge [44]. Each unit in a neural network generates an output signal based on its activation and sends its output value to all other units that are connected to it. Units are connected to each other with connection weights, which multiply the signals transmitted. These network weights specify the strength of the connection. They are adjustable through a training process which enables the network to change its overall behavior according to the nature of specific training examples provided. The diagram of a typical neural network is displayed in Figure 1.10. It consists of units that are organized into three layers: input layer, hidden layer and output layer. The input layer receives signals that are fed into the network, and is connected to the hidden layer. The activation of a hidden unit is determined by the input unit activation and the connection weights between the input and the hidden units. The

hidden layer is connected to the output layer which generates the result of the computations. Since neural networks are designed to model the way in which human brain performs a particular task, and offer the capability of constructing an input-output mapping given the training samples, they can be nicely integrated into CBIR systems to simulate user perception of visual content from feedback information.

Figure 1.10 Diagram of a typical neural network with one hidden layer.

1.3.2 Support Vector Machines (SVMs)

Support vector machines (SVMs) are an implementation of the method of structural risk minimization (SRM) [46]–[48]. This induction principle is based on the fact that the error rate of a learning machine on test data (i.e., the generalization error rate) is bounded by the sum of the training error rate and a term that depends on the *Vapnik-Chervonenkis* (VC) dimension. The basic idea of SVMs involves first transforming data in the original input

space to higher dimensional feature space by utilizing the technique known as “kernel trick” (kernel function Ψ) as illustrated in Figure 1.11. In doing so, nonlinearly separable data can be transformed into a linearly separable feature space. An optimal decision hyperplane can then be constructed in this high dimensional feature space by maximizing the margin of separation between positive and negative samples (Figure 1.12). A linear decision boundary constructed in the feature space corresponds to nonlinear decision boundary in the input space. By the use of a kernel function, it is possible to compute the separating hyperplane without explicitly carrying out the mapping in the feature space. The optimal hyperplane is determined by solving a quadratic programming (QP) problem, which can be converted to its dual problem by introducing Lagrangian multipliers. The training data points that are nearest to the separating hyperplane are called support vectors. The optimal hyperplane is specified only by the support vectors. Because of the nice properties of SVMs, they have been successfully utilized in many real-world applications. SVMs are also good choices in the context of CBIR to learn the user information need by estimating the decision boundary between the positive and negative feedback images.

Figure 1.11 The basic idea of SVMs: map the nonlinearly separable training data into a higher dimensional feature space via kernel function Ψ where the training data is separable, and construct an optimal separating hyperplane there. This yields a nonlinear decision boundary in the input space.

Figure 1.12 Optimal separating hyperplane. (a) A separating hyperplane with small margin, (b) The optimal separating hyperplane with maximal margin. A better generalization capability is expected from (b).

1.3.3 Fuzzy Logic

Fuzzy logic is derived from fuzzy set theory dealing with reasoning that is approximately rather than precisely deduced from classical predicate logic [49]–[51]. It is a tool for embedding human structured knowledge expressed in linguistic terms such as “young”, “big”, “old”, which is approximate and non-quantitative. Fuzzy logic attempts to model this aspect of reasoning that goes in the human brain using a machine. It allows for set membership values between zero and one, which express the degree of membership as opposed to crisp logic. In CBIR systems, user interpretation of visual content is, at times, uncertain or imprecise. Therefore, fuzzy logic is a natural way to model this imprecision of user perception. The relevance of the retrieved image can be mapped into an interval $[0,1]$ through a fuzzy membership function. This is in contrast to the binary logic employed by conventional CBIR systems, where the relevance of the retrieved images is judged as either “fully relevant” or “totally irrelevant”.

1.3.4 Evolutionary Computation

Evolutionary computation is a technique inspired by the process of biological evolution, i.e., “survival of the fittest” [52], [53]. They involve the study and development of algorithms that simulate biological evolution of individuals via processes of selection and reproduction. These processes depend on the so-called evolution fitness of the individuals as defined by the environment. The goal of evolutionary computation is to create individuals that are solutions to a problem of better fitness, in other words, better quality. Usually, they are used to solve analytically nontractable or difficult problems. In the context of an optimization problem, an evolutionary algorithm can be used by creating a population of potential solutions to the optimization problem instead of a single solution as in traditional optimization techniques. At each generation, the evolutionary algorithm involves a competitive selection of solutions based on their fitness associated with the optimization problem. Solutions with high degree of fitness will survive into the next generation and reproduce to generate new individuals. This iterative cycle continues until the termination criteria are met. Evolutionary algorithms are well suited for a wide range of combinatorial and continuous problems. In the case of knowledge propagation for CBIR systems, we would like to find the set of salient regions from each image that shares the same semantic concept. This can be formulated as a combinatorial optimization problem. Evolutionary algorithms thus can be used to find the solutions.

1.4 Contributions of Thesis

The main contributions of this thesis are development of efficient and effective image retrieval systems using computational intelligence techniques, namely, neural networks, support vector machines, fuzzy logic, and evolutionary computation. The frameworks are

developed from low-level (color, texture, shape) to medium-level (regions of attributes) to high-level (keywords, tags), which follow the structured levels of image content analysis as shown in Figure 1.13.

Figure 1.13 Structured levels of image content analysis.

In Chapter 4, we propose a soft relevance framework to integrate the users’ fuzzy perception of image similarity into relevance feedback. It reconciles the dilemma of binary or multi-level labeling by employing soft fuzzy decision instead of hard binary decision. We introduce an a posteriori probability estimator to evaluate the relevance of the fuzzy images labeled by the users. A progressive fuzzy radial basis function network (PFRBFN) is developed to learn the user’s fuzzy perception of image relevance through soft-decision labeling.

In Chapter 5, we develop a unified framework called pseudo-label fuzzy support vector machine (PLFSVM) to perform content-based image retrieval. The PLFSVM integrates the advantages of pseudo-labeling and FSVM for solving the small sample problem. In contrast to most existing feedback approaches in CBIR systems that are concerned with only the

labeled data, the proposed method exploits the unlabeled data to enhance the retrieval performance. Previous studies have shown that some unlabeled images contain a certain degree of statistical correlations with labeled images, and promising results can be achieved when incorporating these unlabeled images for learning. In this work, by studying the characteristics of labeled images, we propose to use an unsupervised clustering algorithm to select unlabeled images, which are called pseudo-labeled images. The relevance of these pseudo-labeled images is estimated using a fuzzy membership function, and integrated into fuzzy support vector machine (FSVM) for learning.

It is worth noting that Chapter 4 focuses on fuzzy perception of user feedback, where a recursive network structure is designed to utilize the unique batch process of feedback to reduce computational cost. In contrast, SVMs are not effective in handling recursive-structured classifier. Chapter 5 focuses on small sample problem, where the adoption of SVMs will not be too computationally intensive as the number of samples is small. Different considerations determine the utilization of different classifiers namely PFRBFN and SVMs in Chapter 4 and Chapter 5 respectively.

Chapter 6 presents a new framework that integrates relevance feedback into RBIR systems. It is designed for solving several important issues arising in RBIR systems such as region-based image similarity measure, effective learning algorithms, and design of region weight estimation schemes. A new variable-length radial basis function network (VLRBFN) with progressive learning capability is proposed. A main feature of VLRBFN lies in its computational efficiency. This is because the proposed recursive network structure reduces the training time as only a small update network and its associated parameters need to be estimated.

Chapter 7 investigates the new paradigm of online peer tagging which can be incorporated into CBIR systems. The ultimate goal is to integrate the advantages of content- and keyword-

based image retrieval algorithms while alleviating their respective difficulties. Related issues are studied including HCI for image tagging, tag generation and formation, and tag clustering. Due to the broad scope of online peer tagging, and the need for significant user participation to perform comprehensive tagging, we mainly focus on knowledge propagation of keywords in this study. It is a process to automatically assign or propagate keywords from a set of manually annotated images to other unannotated images. The proposed knowledge propagation scheme is based on image content analysis and training of keyword classifiers.

1.5 Organization of Thesis

The remainder of this thesis is organized as follows:

In Chapter 2, review of CBIR systems is presented.

In Chapter 3, the image features and two computational intelligence techniques, namely, radial basis function (RBF) network and SVMs are introduced.

In Chapter 4, a soft relevance framework in CBIR systems is described. The proposed notion of fuzzy relevance feedback is introduced. The developed PFRBFN framework is then discussed in detail.

In Chapter 5, a unified framework that integrates the advantages of pseudo-labeling and FSVM is presented to address the small sample problem. The unlabeled image selection and pseudo-label estimation scheme are discussed. A fuzzy membership function is developed to determine the implicit class membership of the pseudo-labeled images.

In Chapter 6, an efficient and effective relevance feedback approach in RBIR systems is proposed. We discuss the structure of the developed VLRBFN and the clustering algorithm used in network construction. The new region weight learning algorithm that determines the relevance of the regions is presented. The learning algorithm of the network is further

explained.

In Chapter 7, we first explore the unique issues in peer tagging including HCI for image tagging, tag generation and formation, and tag clustering. The developed knowledge propagation scheme is then presented.

In Chapter 8, we summarize the contributions of this thesis toward CBIR and discuss future research directions.

Chapter 2

Review of Content-based Image Retrieval

2.1 Overview

The rapid growth of image data has increased the demand for efficient access to them in many applications, ranging from art galleries, digital libraries, biomedicine to military and education. In view of this, content-based image retrieval (CBIR) has been developed as an instrumental approach of accessing image data. It is designed to retrieve a set of desired images from an image collection on the basis of visual content such as color, texture, shape and spatial relationship that are present in the images. An interactive CBIR system involves users' interactions by providing their judgement on the relevance of the retrieved results. These feedbacks are then used by the systems to learn user information need. A typical interactive CBIR system architecture is illustrated in Figure 2.1.

Figure 2.1 Interactive CBIR system architecture.

The main processing of an interactive CBIR system involves offline processing and online querying. Offline processing includes feature extraction, representation, and organization. Online processing is the interaction between the user and the system. A user can browse through the image database and submit his/her initial query to the system through query-by-example (QBE). The system follows some search rules such as k-nearest neighbors (K-NN) search and uses a similarity measure for matching the query with the images in the database, and subsequently retrieves the K most similar images. The user provides his/her evaluations by labeling each displayed image as either ‘relevant’ or ‘irrelevant’. The system learns the user’s interests based on the feedback images. Each image in the database is then assigned a similarity value based on the outputs of the trained classifier. According to the similarity values of all the images in the database, a new ranked list of images which better approximate the user’s preferences is obtained and presented to the user. This feedback process iterates until the user is satisfied with the retrieval results.

We will present major related works in CBIR in the remainder of this chapter. Section 2.2 reviews feature extraction and representation techniques, which are the basis of CBIR. Section 2.3 describes the similarity measures used to assess the resemblance between images.

In Section 2.4, relevance feedback in CBIR systems is reviewed, and the corresponding fundamental issues are discussed. In Section 2.5, issues related to RBIR systems are presented. Section 2.6 provides a brief discussion on automatic image annotation (knowledge propagation) for CBIR systems. Section 2.7 introduces performance evaluation measures in CBIR systems.

2.2 Feature Extraction and Representation

Feature extraction and representation is a fundamental process in CBIR systems. It is concerned with the use of mathematical and quantitative methods for the description and representation of image content. Color, texture, and shape are the most frequently used visual features in current CBIR systems. Each feature may have several representations. No single best representation exists for a given feature due to human perceptual subjectivity. Different representations characterize different aspects of the feature. Table 2.1 lists some of the most commonly used features and their corresponding representations.

Features	Representations
Color	Color histogram, Color moments, Color correlogram, Color sets
Texture	Co-occurrence matrix, Tamura texture, Wavelet transform
Shape	Fourier descriptor, Invariant moments

Table 2.1 Commonly used features and their representations in CBIR systems.

Color is an expressive attribute for describing image content, and is commonly used in CBIR systems. Generally, color features are relatively easy to be extracted and matched. Color histogram is one of the most widely used color feature representations [54]. It is a first-order descriptor for the representation of the global color distribution, derived by counting the

number of pixels of each of given set of color ranges in a typically two-dimensional or three dimensional color space. Most color histogram approaches consider neither the color similarity across different bins nor the color dissimilarity in the same bin, hence are sensitive towards noise. Fuzzy color histogram (FCH) has been proposed to address this concern and demonstrated improved performance over the conventional color histogram methods [55]. It considers the color similarity of each pixel's color associated to all the histogram bins through fuzzy-set membership function. Color moments have been proposed to overcome the quantization effects in color histogram [56]. They characterize the color distribution of an image by its moments (mean, variance and skewness). Color correlogram has been proposed as a kind of spatial extension of the color histogram. It is a second-order representation of local color feature, which expresses how the spatial correlation of color changes with distance [9], [57]. Color sets have been proposed as an alternative to color histogram [58]. They first perform a color space transformation and quantize the transformed color space into bins. A color set is then defined as a selection of the colors from the quantized color space.

Texture is another powerful discriminating feature. The co-occurrence matrix [59] and Tamura representation [60] are two popular texture features. Co-occurrence matrix explores the gray level spatial dependence of texture. Based on the orientation and distance between image pixels, a co-occurrence matrix is constructed, from which meaningful statistics can be extracted as the texture representation. Some commonly used statistical features computed based on the co-occurrence matrix include contrast, entropy, inverse difference moment, energy, correlation, homogeneity, and angular second moment. Tamura representation is based on the psychological measurements corresponding to human visual perception of textures. It characterizes texture by measures of coarseness, contrast, directionality, line-likeness, regularity, and roughness. Tamura representation is desirable in image retrieval as all of its texture measures are visually meaningful. Other texture representations include

Gabor transforms [61], wavelet transforms [62], [63], Markov random fields [64], and so on. Wavelet moments and Gabor wavelet are both multi-resolution features to characterize the image texture in the frequency domain. Wavelet moments describe the global texture properties of images using the energy of discrete wavelet transform (DWT) subbands. Gabor wavelet offers flexibility in controlling the orientation and scale information.

Shape is one of the primitive features for image content description. Shape content description is a challenging task because it is difficult to define perceptual shape features and measure the similarity between shapes. To make the problem more complex, shape is often corrupted with noise, defection, arbitrary distortion and occlusion. Generally, there are two types of shape representations: boundary-based (based on the outer boundary of the shape) and region-based (based on the entire shape region). Common descriptors for boundary-based shape representation include the chain code [65], Fourier descriptor [66], and the UNL¹ descriptor [67]. Region-based representation method is a more general shape description technique than the boundary-based representation method. Some important and widely used region shape descriptors are invariant moments [68], Zernike moments [69], and pseudo-Zernike moments [67]. Fourier descriptor and invariant moments are the two most popular representatives for these two categories, respectively. Fourier descriptor is formed by applying Fourier transform to the object boundary and combining the Fourier transform coefficients as the shape feature. Invariant moments are shape features that use region-based moments which are invariant to translation, rotation and scale. Other shape representations include finite element method [4], turning function [70], wavelet descriptors [71] and the combined representations of several basic representations such as Fourier descriptor and invariant moments.

¹ Universidade Nova de Lisboa

2.3 Similarity Measures in CBIR Systems

Within the feature space, similarity measures are distance functions mapping the images represented by feature vectors into numeric values which indicate the degree of (dis)similarity between them [72]. Images returned at the top of the retrieval results have the highest similarity with the query. The selection of similarity measures depends on the types of features used. Some common similarity measures include the L1 norm (also known as Manhattan distance or city-block distance) [73], L2 norm (or Euclidean distance) [74], the weighted distance metric (or Mahalanobis distance) [75], histogram intersection [54], and Earth Mover's Distance (EMD) [76]. However, all these distance functions do not necessarily match human perceptual similarity at the semantic level. Some research has been done to address this issue. Dynamic partial functions have been proposed to discover a perceptual distance function by extensively mining the image data [77].

2.4 Relevance Feedback in CBIR Systems

Relevance feedback was introduced into CBIR systems to address the problem of mismatch between the high-level concepts and the low-level visual features [2], [6], [7], [9]–[22]. The user is incorporated into the retrieval systems to provide his/her evaluations on the retrieval results. The systems then learn from the user feedback to retrieve a new set of images that better satisfy the user information need. Figure 2.2 illustrates how a typical relevance feedback approach works. First, the retrieval system presents a set of images to the user. The user evaluates which ones are close to what he/she is looking for, marks them as relevant or irrelevant, and returns these feedbacks to the system. In turn, the system learns from user's feedbacks and selects the next set of images to be displayed. We can see that with relevance

feedback, the new set of images shows more relevant ones (in this case, the user is looking for images of lions). This iterative process continues until the user is satisfied with the retrieval results.

Figure 2.2 Illustration of relevance feedback process.

2.4.1 Previous Work on Relevance Feedback

Many relevance feedback algorithms have been adopted in CBIR systems and demonstrated considerable performance improvement [2], [6], [7], [9]–[22]. Some well-known methods include query refinement [2], [12], feature reweighting [9], [10], statistical learning [7], [13], [14], SVMs [15]–[17], and neural networks [18]–[20].

Query refinement and feature reweighting are two widely used relevance feedback methods in CBIR. Query refinement tries to reach an optimal query point by moving it towards relevant images and away from the irrelevant ones. This technique has been implemented in many CBIR systems. The best-known implementation is the MARS (Multimedia Analysis and Retrieval System) [2]. Reweighting aims at emphasizing the feature components that help to retrieve relevant images, while de-emphasizing those that hinder this process by updating the weights of the feature vectors. It uses heuristic

formulation to adjust the weight parameters empirically.

Statistical learning has been developed to incorporate the users' feedbacks by modeling the probability distribution of images in the database [7], [13]. Bayesian classifiers have also been developed to perform retrieval based on the feedback samples in [14]. Positive examples are used to estimate a Gaussian distribution that represents the desired images for a given query, while the negative examples are used to modify the ranking of the retrieved candidates. Another popular method in CBIR is centered on SVMs. SVMs utilize the samples near the boundary as the support vectors to minimize the classification errors.

Radial basis function (RBF) networks have been used to determine the nonlinear relationship between features so that a more accurate similarity comparison between images can be supported [18]. An adaptive radial basis function network (ARBFN) has been proposed for interactive image retrieval [20]. It characterizes the query by multi-class models, an inherent strategy of local modeling, and associates those relevant (positive) samples as the models. The irrelevant (negative) samples are then used to modify the multi-class models in such a way that the models are moved slightly away from the irrelevant samples.

2.4.2 Relevance Judgement in CBIR Systems

Conventional relevance feedback restricts users to binary labeling of feedback images, namely, the images are determined as either "fully relevant" or "totally irrelevant" [2], [9], [12]–[15], [17], [20]. This feedback process is based on hard-decision as to whether the retrieval results satisfy the user requirement. There are some works which provide the ability to have three-option classification by ignoring the "neutral" images. On the other hand, multi-level labeling categorizes the positive and negative examples into several discrete levels of (ir)relevance [10], [11], [22].

2.4.3 Active Learning

Active learning is designed to achieve maximal information gain or minimize uncertainty in decision making [78]. It selects the most informative samples to query the users for labeling. Some recent work on the application of active learning in image retrieval can be found in [7], [15], [79], [80]. Among the various active learning techniques, SVM-based active learning is one of the most promising methods currently available [15], [79]. It aims to select samples that maximally reduce the version space of SVM. It selects samples that are closest to the current SVM decision boundary as the most informative points. Samples that are farthest away from the boundary and on the positive side are considered as the most relevant images.

2.4.4 Incorporating Unlabeled Images into CBIR Systems

Despite the previous works on relevance feedback for CBIR systems, it is still a challenging task to develop effective and efficient interactive mechanisms to yield satisfactory retrieval performance. One of the main issues associated with relevance feedback is the small sample problem. This is because users usually do not have the patience to label a large number of images. Therefore, the performance of relevance feedback methods is often constrained by insufficient training samples. To deal with this problem, some works have been done to incorporate the unlabeled data to improve the learning performance.

Discriminant Expectation Maximization (D-EM) algorithm has been introduced to incorporate the unlabeled samples to estimate the underlying probability distribution [21]. Transductive support vector machine (TSVM) for text classification has been proposed to tackle the problem by incorporating the unlabeled data [81]. It has also been applied for image retrieval [79]. The method proposes to incorporate unlabeled images to train an initial SVM, followed by standard active learning. Incorporating prior knowledge into the SVM has

also been introduced to resolve the small sample problem [82].

2.5 Region-based Image Retrieval

Region-based image retrieval (RBIR) systems have been developed to represent images at the object level to achieve higher retrieval accuracy [5], [23]–[27]. They utilize image segmentation to decompose images into regions, and then perform comparison of image similarity based on the local features extracted from the constituent regions. Relevance feedback is traditionally used in CBIR systems that use global features. There are still relatively few works that incorporate it in the RBIR systems. Nevertheless, some preliminary works have been done to combine RBIR with relevance feedback to improve the retrieval performance. A typical interactive RBIR system architecture is illustrated in Figure 2.3.

Figure 2.3 Interactive RBIR system architecture.

2.5.1 Similarity Measures in RBIR Systems

An important issue in RBIR systems is the definition of image similarity measure, which can

be broadly divided into region-to-region and image-to-image similarity measures. In RBIR systems that use individual region-to-region similarity measures [23], [25], the users are provided with the segmented regions of an image and asked to select (i) the regions used for image comparison, (ii) the features (color, texture, shape) used for region matching, and (iii) sometimes even the relative weights of the features or regions. Such systems obviously place unreasonable workload on the users, and therefore are not widely adopted. Ideally, we would like to have automatic schemes that can identify the significance of each image region, and make meaningful comparisons between two region-based images. However, automatic yet accurate algorithms for extracting meaningful objects from images are difficult to achieve in practice. For instance, inaccurate segmentation may result in an object being partitioned into several regions with none of them being representative of the object (over-segmentation in Figure 2.4(a)). It may also produce insufficient separation of an object from the background (under-segmentation in Figure 2.4(b)).

(a)

(b)

Figure 2.4 Imperfect image segmentation. (a) Over-segmentation, (b) Under-segmentation.

To reduce the amount of inputs from the users and the impact of inaccurate segmentation, image-to-image similarity measures that combine information from all of the image regions have been proposed. This includes integrated region matching (IRM) [24], unified feature

matching (UFM) [26], and EMD [76]. IRM defines the similarity measure as a weighted sum of region-to-region distances. The weighting matrix is determined based on significance constraints and a most similar highest priority (MSHP) principle. UFM is a region-based fuzzy feature matching approach. In UFM, each region is represented as a multidimensional fuzzy feature to characterize the gradual transition between regions (blurry boundaries). The similarity between two images is then defined as the overall similarity between two sets of fuzzy features. EMD computes the distance between two distributions, which are represented by sets of weighted features that capture the distributions. Intuitively, the EMD measures the least amount of work needed to transform one image distribution into the other, which can be solved as a linear optimization problem.

2.5.2 Relevance Feedback in RBIR Systems

Existing feedback methods in RBIR systems can be mainly categorized into query point movement (QPM) [29], [30], feature and region reweighting [31]–[33], SVMs [29], [30], [34], multiple instance learning (MIL) [35], [36], and statistical learning [37], [38].

QPM tries to find the ideal query model by assembling the segmented regions of relevant (positive) samples together, reweighting the regions and forming a pseudo image as the new query. It does not take into account both the irrelevant samples and spatial arrangements of the regions, hence leading to potential underutilization of the information available. Reweighting methods assume unequal emphasis is placed on the regions or features of different objects. Therefore, the contribution of regions to the image similarity should be weighted by their relative importance. Under this consideration, both heuristic [31] and optimal weight learning schemes [32], [33] for determining the region relevance have been developed.

Another class of approaches in RBIR is centered on SVMs. SVMs have been employed

as a binary classifier that separates the relevant images from the irrelevant ones. A new kernel function which is a generalization of the Gaussian kernel is proposed for the SVM-based algorithms [29], [30]. The results are promising, but the computational complexity can be significant for large databases.

In the scenario of MIL in RBIR systems, each image is viewed as a bag, which contains a number of instances corresponding to the regions. Labels (relevant or irrelevant) of individual instances in the bags are unavailable. MIL assumes that a bag is labeled as relevant if at least one of its instances is relevant; otherwise, the bag is labeled as irrelevant. The goal of MIL is to generate a mapping function from the training samples to predict the labels of the unseen images.

Statistical learning has been developed to address the hidden semantic concept discovery in the database [37], [38]. A probabilistic model is constructed and analyzed by the Expectation Maximization (EM) technique to discover the latent semantic concepts. The model is then integrated into the similarity measure for effective image retrieval.

2.6 Automatic Image Annotation

Keywords are the best descriptors for image semantics. However, full manual annotation of a complete database is often tedious and expensive. To address this difficulty, automatic image annotation has been proposed to automatically assign textual descriptors in the form of keywords or tags to unannotated images. It works by first manually annotating a subset of images with keywords. The correlation between image visual features and keywords is then learned from the annotated images. The learned correlation is used to propagate the keywords to the unannotated images.

Previous works on automatic image annotation can be divided into two main categories:

model-based and classifier-based [83]–[91]. A number of model-based approaches use associations between keywords and image regions [83]–[88]. A co-occurrence model is utilized to perform automatic image annotation [83]. It looks at the co-occurrences of words with image regions created by using a regular grid. The translation model views image annotation as a machine translation procedure to translate from a vocabulary of blobs to a vocabulary of keywords [84]. The cross-media relevance model (CMRM) assumes that images may be described from small vocabulary of blobs and learns the joint distribution of blobs and words from a training set of annotated images [85]. Standard latent semantic analysis (LSA) and probabilistic latent semantic analysis (PLSA) have been applied for automatic image annotation by introducing latent variables to link image features with words [86]. Extended from Latent Dirichlet Allocation (LDA) model [87], the correspondence LDA model assumes that a Dirichlet distribution can be used to generate a mixture of latent factors, which is used to generate words and blob features [88].

The classification-based approaches learn a set of classifiers for annotating images [89]–[91]. A content-based soft annotation (CBSA) approach is proposed for image annotation [89]. The procedure starts with annotating a small set of training images with one single keyword provided for each image. An ensemble of binary classifiers is then trained to predict keyword membership for images. The trained classifiers are utilized to propagate soft keywords with associated keyword membership factors to the unannotated images. Bayes point machine (BPM) and SVM have been employed as the binary classifiers for CBSA. A two-dimensional multi-resolution hidden Markov model (2D MHMM) has been proposed to annotate images [90]. Each category of images in the image database is annotated with a few words and is referred to as a concept. A 2D MHMM is employed to profile each category. Given a new image, the similarity between this image and a category is assessed by its likelihood under the profiling model. Categories are then ranked by the likelihoods of the

image under their profiling 2-D MHMMs. Candidate annotation words are selected from those used to describe the top ranked categories. Statistically significant keywords are used to annotate the image. A confidence-based dynamic ensemble (CDE) scheme has been proposed to annotate images [91]. It employs one-class, two-class and multi-class SVMs to perform a three-level classification. At the base level, CDE uses one-class SVM to characterize a confidence factor, which is used to assert the correctness of a class prediction made by a binary SVM classifier. The confidence factor is then propagated to the multi-class classifiers at subsequent levels.

2.7 Performance Evaluation

Performance evaluation is a difficult yet crucial issue in CBIR. Most CBIR systems use performance evaluation techniques developed in information retrieval community. We have adopted the commonly used performance measure called retrieval accuracy (RA) to evaluate the retrieval system [14], [92]:

$$\text{Retrieval accuracy} = \frac{\text{relevant images retrieved in top } T \text{ returns}}{T} \quad (2.1)$$

where T is the number of retrieved images. Further, in order to complement the performance measure of RA, another robust performance metric namely precision-vs-recall (PR) has also been adopted in the thesis [93]. The definitions of precision and recall are given as follows:

$$\text{Precision} = \frac{\text{number of relevant images retrieved}}{\text{total number of images retrieved}} \quad (2.2)$$

$$\text{Recall} = \frac{\text{number of relevant images retrieved}}{\text{total number of relevant images in the database}} \quad (2.3)$$

Precision and recall are usually presented as precision versus recall (PR) graph to evaluate the

retrieval performance. They can indicate the retrieval effectiveness quite well since they include information about images that are not retrieved in top ranked images or images that are not shown to the users.

Other measures include rank of the best match [94], average rank of the retrieved relevant images [95], error rate [96] and so on. Relevance judgement of images could be based on some predefined ground truth, which is considered as objective performance measures. On the other hand, because of subjective human perception of image similarity, subjective measures that integrate users into the evaluation process are useful to assess the performance of a retrieval system.

Chapter 3

Backgrounds and Foundations

In this chapter, we introduce the image features and two computational intelligence techniques, namely, radial basis function (RBF) network and support vector machines (SVMs).

3.1 Image Content Description

The features and their representations used in image retrieval are application dependent. We have explored a large variety of important features (color, texture, and shape) and their corresponding feature representations for our general-purpose image retrieval systems. We have found that color and texture are the most effective features in our studies. For shape features, it has been found that they do not consistently produce good results especially on those images without clear foreground objects. Therefore, only color and texture features are employed in this work. For each feature, multiple representations exist which characterize different aspects of the feature. There is no single representation for a given feature that best represents the human perceptual similarity. Consistent with the studies reported in previous literature, the following feature representations are employed in this work and they have been shown to be effective in our retrieval systems. They are color histogram [54], color moments [56] and color auto-correlogram [9] as the color feature representation, and wavelet moments [62] and Gabor wavelet [61] as the texture feature representation.

Color histogram, representing the first-order color distribution in an image, is one of the most widely used color descriptors. It is easy to compute, invariant to rotation, translation, and viewing axis. We implement the color histogram by first converting the RGB representation of each image into its HSV equivalence. Then, each H, S, V component is uniformly quantized into 8, 2, 2 bins respectively to get a 32-dimensional feature vector.

Color moments have been proposed to overcome the quantization effects in color histogram. It characterizes the color distribution of an image by its moments (mean, variance and skewness). In this study, the first two moments (mean and variance) from the R, G, B color channel are extracted as the color feature representation to form a 6-dimensional feature vector.

Color auto-correlogram is a two-dimensional spatial extension of color histogram. Color histogram does not provide any spatial information, therefore images with similar histograms may have different appearances. Color correlogram integrates spatial information with the color histogram by constructing a color co-occurrence matrix indexed by color pairs and distance, with each entry (i, j) representing the probability of finding a pixel of color j at a distance k from a pixel of color i . The storage requirement for a co-occurrence matrix is significant, therefore only its main diagonal is computed and stored, which is known as color auto-correlogram. The auto-correlogram of the image I for color C_i , is given as:

$$\gamma_{C_i}^{(k)}(I) = \Pr \left[|p_1 - p_2| = k, p_2 \in I_{C_i} \mid p_1 \in I_{C_i} \right] \quad (3.1)$$

where p_1 is a pixel of color C_i in the image I , p_2 is another pixel of the same color C_i with a distance of k away from p_1 . D_8 distance (chessboard distance) is chosen as the distance measure: $D_8(p, q) = \max(|p_x - q_x|, |p_y - q_y|)$, which is the greater distance in the x - or y -direction.

Wavelet moments describe the global texture properties of images using the energy of discrete wavelet transform subbands. It is a simple wavelet-transform feature of an image that is constructed using the mean and standard deviation of the energy distribution at each decomposition level. This in turn corresponds to the distribution of “edges” in the horizontal, vertical, and diagonal directions at different resolutions. In this study, the Daubechies wavelet transform with a three-level decomposition is employed. At each level, it decomposes a 2D signal into four subbands, which are often referred to as LL, LH, HL, and HH according to their frequency characteristics (L = Low, H = High). In this work, the wavelet transform recursively decomposes the LL subband. The LH, HL, and HH subbands are not further decomposed. Thus, a three-level decomposition yields ten subbands as shown in Figure 3.1(i.e., $LLk(m)$, $LHk(m)$, $HLk(m)$, and $HHk(m)$, where $k = 1, \dots, 3$ denotes the decomposition level and $m = 0, \dots, 9$ is the index of the subbands). For each subband, the mean and standard deviation of the transform coefficients are computed. This results in a feature vector of $10 \times 2 = 20$ dimension.

Figure 3.1 Illustration of wavelet transform using a three-level decomposition

Gabor wavelet is widely adopted to extract texture features, and has been shown to be very efficient. Basically, Gabor filters are a group of wavelets, with each wavelet capturing energy at a specific frequency and a specific direction. Expanding a signal using this basis provides a localized frequency description, therefore capturing local features/energy of the signal. A 2-D Gabor function $g(x, y)$ is defined as:

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y} \right) \exp \left[-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\pi j W x \right] \quad (3.2)$$

The self-similar functions are obtained by appropriate dilations and rotations of $g(x, y)$ through the generating function:

$$\begin{aligned} g_{mn}(x, y) &= a^{-m} g(x', y') \\ x' &= a^{-m} (x \cos \theta_n + y \sin \theta_n) \\ y' &= a^{-m} (-x \sin \theta_n + y \cos \theta_n) \end{aligned} \quad (3.3)$$

where $a > 1$, m and n specify the scale and orientation of the wavelet respectively, W is the modulation frequency. The half peak radial bandwidth is chosen to be octave, which determines σ_x and σ_y . In this study, Gabor wavelet filters spanning four scales: 0.05, 0.1, 0.2 and 0.4 and six orientations: $\theta_0 = 0$, $\theta_{n+1} = \theta_n + \pi/6$ are used. For a given image $I(x, y)$, its Gabor wavelet transform is defined by:

$$W_{mn}(x, y) = \int I(x_1, y_1) g_{mn}^*(x - x_1, y - y_1) dx_1 dy_1 \quad (3.4)$$

where $*$ denotes complex conjugation. The mean and standard deviation of the transform coefficient magnitudes are used to form a 48-dimensional feature vector.

The extracted features are summarized in Table 3.1.

Features	Description	Dimension
Color histogram	HSV space is chosen, each H, S, V component is uniformly quantized into 8, 2 and 2 bins respectively.	32
Color auto-correlogram	The chessboard distance is chosen as the distance measure. The image is quantized into $4 \times 4 \times 4 = 64$ colors in the RGB space.	64
Color moments	The first two moments (mean and standard deviation) from the R, G, B color channels are extracted.	6
Gabor wavelet	Gabor wavelet filters spanning four scales: 0.05, 0.1, 0.2 and 0.4 and six orientations: $\theta_0 = 0$, $\theta_{n+1} = \theta_n + \pi/6$ are applied to the image. The mean and standard deviation of the Gabor wavelet coefficients are used to form the feature vector.	48
Wavelet moments	Applying the wavelet transform to the image with a 3-level decomposition, the mean and the standard deviation of the transform coefficients are used to form the feature vector.	20

Table 3.1 Image features used in the experiment.

There exist two main feature fusion methods in the context of CBIR. The first approach consists of two stages. The first stage is to perform similarity matching on individual features (color, texture or shape, etc.). The second stage is to make a fusion by combining the individual matching scores to obtain the overall similarity [10] [11] [18]. Often, weights (linear or nonlinear relationship) are used for combining multiple features. The weights represent the importance of the individual features in the computation of a similarity metric. With user’s relevance feedback, weights can be automatically updated to reflect the user’s perception subjectivity. The second feature fusion method is to concatenate multiple individual features to form an overall new feature vector and performing similarity matching using the generated new feature vector [17] [20] [22]. When combining heterogeneous features into one overall feature vector, feature normalization is necessary. Gaussian normalization is commonly used to provide equal emphasis on each feature component [10]. In this work, we adopt the second feature representation method.

After all the color and texture features have been extracted offline, we concatenate the feature elements from all the individual features (color histogram, color moments, color autocorrelogram, wavelet moments and Gabor wavelet) into an overall feature vector with a dimension of 170. In CBIR systems, the dimensionality of the feature vectors is usually in the order of 10^2 . We have tried employing the well-known dimensionality reduction technique such as principal component analysis (PCA) to reduce the dimension of the feature vectors [44]. However, some degradation in retrieval performance has been observed. The reasons may be that some important discriminating information is lost during the dimensionality reduction. Therefore, the dimension of the feature vectors remains unchanged in this work. This is also consistent with other works that keep the dimensions of the feature vectors unchanged [20][22].

3.2 Radial Basis Function (RBF) Network

Radial basis function (RBF) networks emulate the behavior of certain biological networks [44], [97]–[99]. The standard RBF network consists of an input layer, a hidden layer and an output layer with feed forward architecture. The RBF network with a single output neuron is shown in Figure 3.2.

Figure 3.2 The architecture of RBF network with a single output.

The input layer has a set of P units, which accept the elements of a P -dimensional input feature vector $\mathbf{x} = [x_1, \dots, x_P]^T \in \mathbb{R}^P$. The input units are connected to the hidden layer with r hidden units which are referred to as the RBF units. Connections between the input and the hidden layer have unit weights. The responses of the hidden layer neurons are locally tuned to some region of the input space and decrease as function of the distance of inputs from the neuron's receptive field center. The operation of the hidden layer neuron can be described as $\Phi(\|\mathbf{x} - \mathbf{c}_i\|)$, where $\mathbf{c}_i = [c_{i1}, \dots, c_{iP}]^T \in \mathbb{R}^P$ is the center of neuron i ; $\Phi(\cdot)$ is the basis function of the network defining the activations of the hidden units; $\|\cdot\|$ denotes the Euclidean norm (distance) between the input vector and the center of the RBF unit. Among the possible choices of radial basis functions for RBF networks, Gaussian function is the most common one. Gaussian function is desirable for its capability in generalizing a global mapping and its effectiveness in refining local features without much altering the learned mapping. Gaussian

function is also chosen in this work and defined as follows:

$$\Phi(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \quad (3.5)$$

where σ_i is the width of the i th RBF unit. The RBF units are fully connected to the output layer. The output layer unit implements a linear combination of responses of the hidden layer neurons. In this structure, the output of the RBF neural network can be described by:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^m w_i \Phi(\|\mathbf{x} - \mathbf{c}_i\|) \\ &= \sum_{i=1}^m w_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \end{aligned} \quad (3.6)$$

where m is the number of hidden layer neurons. w_i is the output connection weights that connect the hidden and the output layer.

The parameters of the RBF networks need to be estimated through training algorithms. For a Gaussian function-based RBF network, training of RBF neural networks involves determining the center, width of the hidden layer neurons and output connection weights. Many training algorithms have been designed, most of which correspond to supervised training or a hybrid of unsupervised and supervised training. We will introduce a few schemes in the design of an RBF network depending on how RBF neuron centers are determined [44].

The simplest learning strategy for the RBF network is to randomly select a subset of the training data as the fixed RBF centers. For the radial basis functions themselves, an isotropic Gaussian function may be used whose standard deviation is fixed according to the spread of the centers:

$$\sigma = \frac{d_{\max}}{\sqrt{2N}} \quad (3.7)$$

where $d_{\max}$ is the maximum distance between the selected centers. Such a width determination scheme avoids some extreme conditions such as spiky or flat Gaussian function. Alternatively, we may also use individually scaled centers with broader width in areas of low data density, and narrower width in areas of high density. Once centers have been fixed, the output connection weights can be directly estimated by using the pseudoinverse method.

The random selection method is easy to implement. However, it cannot guarantee a good coverage of the input space unless a large number of training samples is used. One way to overcome this problem is to use a two-stage hybrid learning procedure. In this approach, a self-organized learning process is first used to estimate the locations for the centers of the radial basis functions by placing the centers in only those regions of the input space where significant data are presented. Standard K-means clustering method can be used to associate a representative cluster prototype to each RBF unit. Next, the linear weights of the output layer are estimated by a supervised learning stage, where the least mean square (LMS) algorithm can be used.

Another training algorithm is optimization-based selection of centers. In this approach, the centers of the basis functions and all other free parameters of the network are updated by applying a supervised learning process. Gradient descent-based nonlinear optimization can be used for such a process by minimizing the mean squared error between the desired output and the actual output of the network. First, the criterion to be optimized is defined as:

$$E = \frac{1}{2} \sum_{j=1}^N e_j^2 \quad (3.8)$$

Where N is the number of training samples, and e_j is the error signal between the desired output T_j and actual network output $f(\mathbf{x}_j)$ for an input sample $\mathbf{x}_j$:

$$\begin{aligned} e_j &= T_j - f(\mathbf{x}_j) \\ &= T_j - \sum_{i=1}^m w_i \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \end{aligned} \quad (3.9)$$

By minimizing the cost function E using gradient descent learning, we can obtain all the parameters of the RBF network. The gradient descent-based optimization can be summarized as follows:

- (i) Weight estimation at the n -th learning iteration:

$$\frac{\partial E(n)}{\partial w_i(n)} = -\sum_{j=1}^N e_j(n) \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{c}_i(n)\|^2}{2\sigma_i^2(n)}\right) \quad (3.10)$$

$$w_i(n+1) = w_i(n) - \eta_1 \frac{\partial E(n)}{\partial w_i(n)}, \quad i = 1, 2, \dots, m \quad (3.11)$$

- (ii) Center estimation at the n -th learning iteration:

$$\frac{\partial E(n)}{\partial \mathbf{c}_i(n)} = -w_i(n) \sum_{j=1}^N e_j(n) \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{c}_i(n)\|^2}{2\sigma_i^2(n)}\right) \frac{\mathbf{x}_j - \mathbf{c}_i(n)}{\sigma_i^2(n)} \quad (3.12)$$

$$\mathbf{c}_i(n+1) = \mathbf{c}_i(n) - \eta_2 \frac{\partial E(n)}{\partial \mathbf{c}_i(n)}, \quad i = 1, 2, \dots, m \quad (3.13)$$

- (iii) Width estimation at the n -th learning iteration:

$$\frac{\partial E(n)}{\partial \sigma_i(n)} = -w_i(n) \sum_{j=1}^N e_j(n) \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{c}_i(n)\|^2}{2\sigma_i^2(n)}\right) \frac{\|\mathbf{x}_j - \mathbf{c}_i(n)\|^2}{\sigma_i^3(n)} \quad (3.14)$$

$$\sigma_i(n+1) = \sigma_i(n) - \eta_3 \frac{\partial E(n)}{\partial \sigma_i(n)}, \quad i = 1, 2, \dots, m \quad (3.15)$$

- (iv) Repeat steps (i)-(iii) until convergence or a maximum number of iterations is reached.

The term $e_j(n)$ is the error signal of the j -th training sample $\mathbf{x}_j$ at the n -th learning iteration, and η_1 , η_2 and η_3 are learning parameters for w_i , $\mathbf{c}_i$, and σ_i , respectively.

3.3 Support Vector Machines (SVMs)

Support vector machines (SVMs) are powerful learning machines introduced by Vapnik based on the idea of structural risk minimization (SRM) [46]–[48]. They have been applied to many fields of application including text classification, speech recognition, data mining and communication, and shown to outperform traditional learning machines. The basic idea of SVMs for binary classification is to find an optimal separating hyperplane that maximizes the margin between two classes in a kernel-induced feature space. Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a set of n training samples, where $\mathbf{x}_i \in \mathbb{R}^P$ is a P -dimensional sample in the input space, and $y_i \in \{-1, 1\}$ is the class label of $\mathbf{x}_i$ for $i = 1, \dots, n$. SVMs first map the input data into a high-dimensional feature space through a mapping function $\mathbf{z} = \phi(\mathbf{x})$ and finds the optimal separating hyperplane with minimal classification errors in this space. The hyperplane can be represented as:

$$\mathbf{w} \cdot \mathbf{z} + b = 0 \quad (3.16)$$

where $\mathbf{w}$ is the normal vector of the hyperplane, and b is the bias which is a scalar. In particular, the set S is said to be linearly separable if the following inequalities hold for all training data in S :

$$\begin{cases} \mathbf{w} \cdot \mathbf{z}_i + b \geq 1, & \text{if } y_i = 1 \\ \mathbf{w} \cdot \mathbf{z}_i + b \leq -1, & \text{if } y_i = -1 \end{cases} \quad i = 1, \dots, n \quad (3.17)$$

For linearly separable case, the optimal hyperplane can be obtained by maximizing the margin of separation ρ between the two classes, where $\rho = \frac{2}{\|\mathbf{w}\|}$. The construction of the optimal hyperplane is illustrated in Figure 3.3.

Figure 3.3 The optimal separating hyperplane $\mathbf{w} \cdot \mathbf{z} + b = 0$ with maximal margin $\frac{2}{\|\mathbf{w}\|}$.

Maximizing the margin leads to solving the following constrained optimization problem:

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to } y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1, \quad i = 1, \dots, n \end{aligned} \quad (3.18)$$

This optimization problem can be solved by quadratic programming (QP). However, for the linearly non-separable case where the inequalities in (3.17) does not hold for some data points in S , a modification to the original SVM formulation can be made by introducing nonnegative variables $\{\xi_i\}_{i=1}^n$. In this case, the margin of separation is said to be soft. The constraint in (3.17) is modified to:

$$y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, n \quad (3.19)$$

The $\{\xi_i\}_{i=1}^n$ are called slack variables. They measure the deviation of a data point from the idea condition of pattern separability. Misclassifications occur when $\xi_i > 1$. The optimal separating hyperplane is then found by solving the following constrained optimization problem:

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ & \text{subject to } y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n \end{aligned} \quad (3.20)$$

where C is the regularization parameter controlling the tradeoff between margin maximization and classification error. Larger value of C produces narrower-margin hyperplane with less misclassifications. The optimization problem can be transformed into the following equivalent dual problem using Lagrange multipliers:

$$\begin{aligned} & \text{maximize } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{z}_i \cdot \mathbf{z}_j \\ & \text{subject to } \sum_{i=1}^n y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n \end{aligned} \quad (3.21)$$

where α_i is the Lagrange multiplier associated with the constraints in (3.19). According to the Kuhn-Tucker (KT) condition [100], the following equalities hold:

$$\alpha_i [y_i(\mathbf{w} \cdot \mathbf{z}_i + b) - 1 + \xi_i] = 0, \quad i = 1, \dots, n \quad (3.22)$$

$$(C - \alpha_i) \xi_i = 0, \quad i = 1, \dots, n \quad (3.23)$$

α_i is nonzero only if $y_i(\mathbf{w} \cdot \mathbf{z}_i + b) = 1 - \xi_i$. The data points that correspond with $\alpha_i > 0$ are called support vectors. The optimal solution for the weight vector $\mathbf{w}$ is a linear combination of the training samples which is given by:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{z}_i \quad (3.24)$$

The optimal bias b can be determined from the KT condition in (3.22) by taking any data point $\mathbf{z}_j$ with $0 < \alpha_j < C$. The decision function of the SVMs can then be obtained as:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{z} + b = \sum_{i=1}^n \alpha_i y_i \mathbf{z}_i \cdot \mathbf{z} + b = \sum_{i=1}^n \alpha_i y_i \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}) + b \quad (3.25)$$

It is noted that both the construction of the optimal hyperplane in (3.21) and the evaluation of the decision function in (3.25) only require the evaluation of dot products $\varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$ or $\varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x})$. This implies that we do not necessarily know about φ in explicit form. Instead, a function $K(\cdot, \cdot)$ called kernel function is introduced that can compute the inner product of two data points in the feature space, i.e. $K(\mathbf{x}_i, \mathbf{x}) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x})$. Functions that satisfy the Mercer's theorem can be used as kernels [44]. There are three common types of kernels used in SVMs including polynomial kernel, radial basis function kernel and sigmoid kernel. Using this kernel trick, the dual optimization problem in (3.21) becomes:

$$\begin{aligned} & \text{maximize } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & \text{subject to } \sum_{i=1}^n y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n \end{aligned} \quad (3.26)$$

And we can construct the optimal hyperplane in the feature space without having to know the mapping φ :

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \quad (3.27)$$

Chapter 4

Soft Relevance Framework in Interactive Content-based Image Retrieval

4.1 Introduction

An important aspect in the formulation of relevance feedback in CBIR is the consideration for the potential imprecision of user perception during the feedback process. Current relevance feedback methods employ binary labeling approach, namely, the users have to classify an image either as “fully relevant” or “totally irrelevant” [2], [9], [12]–[15], [17], [20]. This hard-decision approach is inconsistent with the nature of human visual perception, which tends to be uncertain at times. An example of such a scenario occurring in image retrieval can be seen in Figure 4.1, where the user submits the query of a red flower with the intention to locate some flowers and in particular red flowers. The retrieval results contain red flowers, yellow flowers, and other objects. Obviously, the user will mark the red flowers as relevant, the non-flower images as irrelevant. However, uncertainties arise in the cases of yellow flowers since they satisfy, up to a certain extent, the information need of the user. If a hard-decision has to be made on the yellow flowers, the user will face the dilemma whether to classify them as either fully relevant or totally irrelevant. It reflects the uncertainty embedded in the decision-making process. Such a decision can affect the outcome of the subsequent retrieval results. If the user marks the yellow flowers as relevant, what he/she really implies is that they satisfy his/her information need up to a certain extent. Therefore, the matching

degree cannot be specified by the user with binary labeling.

In contrast, multi-level labeling categorizes the feedback samples into several discrete levels of (ir)relevance [10], [11], [22]. This technique, however, is both cumbersome as well as tedious as the users need to classify every image into one of the multiple levels. This conflicts with the uncertainty embedded in human perception. A user is more inclined towards using expressions such as “this image is more or less relevant” or “this image is more relevant than that one”.

Figure 4.1 Scenario of user ambiguity occurring during the feedback process.

In view of this, we propose a soft relevance framework to integrate the users’ fuzzy perception of image similarity into relevance feedback. In addition to the conventional “relevant” and “irrelevant” labels, a third “fuzzy” option is incorporated into the feedback mechanism. Users can select a displayed image as fuzzy if they are ambiguous about its relevance. The new scheme reconciles the dilemma of binary or multi-level labeling by employing soft fuzzy decision instead of hard binary decision. We introduce an *a posteriori* probability estimator to evaluate the relevance of these fuzzy images. A progressive fuzzy radial basis function network (PFRBFN) is developed to learn the user information needs,

where a recursive network structure is designed to utilize the unique batch process of feedback to reduce computational cost. A main feature of PFRBFN lies in its computational efficiency as only the parameters of a small update network need to be estimated in each feedback iteration.

The organization for the rest of this chapter is outlined as follows. Section 4.2 introduces the proposed notion of fuzzy relevance feedback. In Section 4.3, we discuss the proposed PFRBFN framework. The user interface is first described. The structure of PFRBFN and the clustering algorithm used in network construction are then provided. We further explore the proposed *a posteriori* estimator which is developed to determine the soft relevance membership function of the fuzzy images. The learning algorithm of the network is then explained in detail. In Section 4.4, experimental results using the proposed method are discussed and compared with other techniques. Finally, concluding remarks are given in Section 4.5.

4.2 Fuzzy Relevance Feedback in CBIR

4.2.1 Hierarchical User Information Priority

Often users have multiple levels of information needs, which can be abstracted into an information hierarchical structure given in Figure 4.2 [101]. In the figure, we illustrate a three-level hierarchy consisting of the primary, secondary, and tertiary focuses. The primary focus is the general object or topic that a user is looking for, the secondary focus can be considered as important attributes that characterize the objects, while the tertiary focus in turn refers to less prominent properties. Focuses at various levels have different degrees of importance, and therefore do not contribute equally to the user information need. It is worth noting that the hierarchical tree structure of user information need has been adopted in this

work to illustrate potentially different levels of priority, overlapping interests, and fuzzy interpretation of visual contents. It does not imply that user information need can be divided into three levels of interests as this contradicts with the argument that user perception of image similarity is ambiguous. In this case, a three-level structure is adopted simply to demonstrate that the user information need is hierarchical, or in other words, prioritized in nature. There is no specific number of layers in the information-priority structure.

User interpretation of visual content is, at times, uncertain or imprecise as it tends to vary with respect to different users and circumstances. In other words, the human perception of visual content is fuzzy in nature, and maybe nonstationary under different conditions. In most current interactive CBIR systems, the users are asked to make a binary labeling on each feedback image. Binary labeling is a hard decision approach which is inappropriate in addressing the issue of fuzzy perception. Using the same example in Figure 4.1, a user may be looking for red flowers in a database. Based on the perceptual model in Figure 4.2, the user may have a primary focus of “flowers” object, and a secondary focus of “red color” attribute. If binary labeling is adopted, the user would encounter a dilemma as to whether a given yellow flower should be classified as “fully relevant” or “totally irrelevant”, since the yellow flower satisfies the primary but not the secondary need. Therefore, the binary crisp labeling cannot resolve this dilemma adequately.

Figure 4.2 Hierarchical structure of user information priorities.

4.2.2 Fuzzy Relevance Feedback

As pointed out in the earlier discussion, binary labeling cannot handle the inherent uncertainty of visual perception. Fuzzy logic, on the other hand, provides a more natural and flexible way in expressing the user interpretation of image similarity. In view of this, we propose a new soft relevance paradigm which integrates the users’ fuzzy perception of image similarity into the framework of relevance feedback. Apart from the conventional “relevant” and “irrelevant” labeling, a third “fuzzy” option is incorporated into the feedback mechanism. As opposed to making the hard binary decisions, the new approach allows soft-decisions to be made on the similarity of the retrieval results. There are some works which provide the ability to have three-option classification. However, it is worth noting that under this setting, the methods ignore the “neutral” images and do not attempt to evaluate the relevance of these images, hence leading to potential underutilization of the information available.

We develop an *a posteriori* probability estimator to determine the soft relevance membership function for each fuzzy image. These estimates are combined with those

originating from the relevant and irrelevant samples to train a classifier. A PFRBFN is then constructed progressively to learn the user information needs. A two-stage algorithm consisting of subtractive and fuzzy C-means clustering is employed to estimate the neuron number and their centers. A gradient descent-based algorithm is developed to adjust the underlying network parameters by optimizing a carefully selected cost function which will be explained in greater detail in subsection 4.3.4.

It is noted that the proposed method employs similar notion of fuzzy image relevance, when compared with techniques such as PicHunter [7]. The proposed PFRBFN method, however, differs from the PicHunter system in the following aspects: (i) the PFRBFN method is designed for category searching (group of images sharing similar characteristics), while the PicHunter system is developed for single target searching. The PicHunter system adopts a different performance measure which is defined as the average number of images required to converge to the desired specific target. Therefore, it is difficult to make a fair comparison between the two methods as they have different searching objectives, (ii) the PFRBFN method introduces a new fuzzy option in addition to traditional relevant and irrelevant choices during the feedback process. The system then adopts a soft membership function to estimate the relevance of these fuzzy feedback images, hence putting into context the relative importance of these images, and (iii) the PFRBFN method employs a radial basis function network to perform image similarity learning while the PicHunter system uses Bayesian framework to update the probability density distribution of the images in order to retrieve the target.

4.3 Progressive Fuzzy Radial Basis Function Network (PFRBFN)

4.3.1 User Interface

In this section, we provide a general overview on the operation of user interface in our retrieval system as shown in Figure 4.3. Initially, the user can select a query image by browsing through the image database. The selected query is displayed at the top left corner. Next, the user can search the image database by pressing the “Search” button, and the 25 most relevant images are ranked and displayed in a descending order of relevance from left to right, and top to bottom. It is noted that under each displayed image, a pull-down menu is available which enables the user to select three possible choices of feedback (relevant, irrelevant and fuzzy), as illustrated in the figure. The user will simply be asked to select each displayed image as either relevant, irrelevant or fuzzy according to his/her information need. The user can then submit his/her feedback by pressing the “Feedback” button. The system then learns from the feedback images, and presents a new ranked list of images to the user for further feedback. The process continues until the user is satisfied with the retrieved results.

Figure 4.3 Illustration of user interface.

4.3.2 Structure of PFRBFN

In interactive CBIR, global modeling of image similarity (as illustrated in Figure 4.4(a)) may not address the local information contained in the query and feedback images adequately (as illustrated in Figure 4.4(b)). For instance, when the user submits the query image of a dog, he/she may be interested in finding any images containing dogs, as shown in Figure 4.5. Hence, the user information need will be more accurately represented by a variety of models, each of which is correlated to the semantic class “dog” but also has its own local characteristics. To exploit the local characteristics of image relevance, it is more desirable to adopt a multi-cluster local function modeling strategy. In doing so, the local information can be utilized efficiently to achieve better generalization. In addition, the learning rate of retrieval systems should be fast since the users often interact with the systems online. An

ideal choice that satisfies the above-mentioned criteria is the radial basis function (RBF) network. RBF network enjoys the following characteristics that make them a good choice in the context of CBIR: fast learning speed (convergence rate during training), simple network structure, piece-wise local function approximation, and global generalization power [44], [97]–[99].

Figure 4.4 Illustration of image similarity and query characterization in the feature space. (a) Global modeling of image similarity using a single model, (b) Multi-cluster local modeling of image relevance.

Figure 4.5 Example images that represent the semantic “dog”.

In this section, we propose a new PFRBFN to progressively learn the user perception of interested images through continual feedback. The schematic diagram of the network is given in Figure 4.6. The t -th iteration PFRBFN is constructed recursively based on two modules: (i) the $(t-1)$ -th iteration PFRBFN, and (ii) the t -th iteration update network. The output of the t -th iteration PFRBFN $F_t(\mathbf{x})$, for R -dimensional input feature vectors $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_R]^T$, is given as:

$$F_t(\mathbf{x}) = \begin{cases} F_{t-1}(\mathbf{x}) + U_t(\mathbf{x}), & t = 1, 2, \dots, T \\ 0, & t = 0 \end{cases} \quad (4.1)$$

where $F_{t-1}(\mathbf{x})$ is the $(t-1)$ -th iteration PFRBFN output, $U_t(\mathbf{x})$ is the update network output, and T is the total number of feedback iterations.

The update network consists of an input layer, three subnets and an output layer. The input layer receives the R -dimensional feature vectors. It is connected to three subnets which are constructed from the relevant (positive), irrelevant (negative), and fuzzy samples, respectively. The output layer has a single unit whose output value $U_t(\mathbf{x})$ is the weighted combination of the subnet responses. $\{w_{P_i}^t\}_{i=1}^{I_P}$, $\{w_{N_i}^t\}_{i=1}^{I_N}$, and $\{w_{F_i}^t\}_{i=1}^{I_F}$ are the sets of output connection weights associated with the positive, negative and fuzzy samples at the t -th iteration. I_P , I_N and I_F are the number of RBF units for the positive, negative and fuzzy samples, respectively. The detailed network structure and parameters updating algorithm will be described in subsection 4.3.5.

Figure 4.6 Schematic diagram of PFRBFN. (a) Recursive structure of PFRBFN, (b) Update network.

The PFRBFN is developed with a few main considerations: (i) to integrate the unique batch process of user feedback, (ii) to reduce the computational time of training process, and (iii) to integrate the potential fuzzy feedbacks from the users. These objectives are achieved by building the t -th iteration PFRBFN from the $(t-1)$ -th iteration PFRBFN progressively, and combining it with the t -th iteration update network. The main idea is that the previous $(t-1)$ -th iteration PFRBFN has been trained to learn the user information need. Therefore, it is unnecessary to retrain the whole network for each feedback iteration. Instead, we introduce an update network to track the current user information need. This enables a smaller update network to be developed and trained batch-wise. The proposed recursive network structure requires training of a small update network at each iteration. Hence, only the parameters of the small update network need to be estimated at each feedback iteration instead of the whole network. This will lead to significant reduction in training time. The method is algorithmically effective as all feedback samples accumulated over the previous iterations are used to train the update network. The relevance of the fuzzy images is estimated using an *a posteriori* estimator to be explained in subsection 4.3.4. These values, together with the relevant and irrelevant feedbacks, are then incorporated into the supervised learning of the update network.

4.3.3 Two-stage Clustering for PFRBFN Subnet Creation

An important issue in the development of PFRBFN is the number of RBF units and their initial values in three subnets corresponding to relevant, irrelevant, and fuzzy samples. This is vital as image retrieval systems usually require fast interaction. Hence reducing the number of RBF units while maintaining a good representation of image similarity will cut down the training time of the network. In view of this, a more systematic and efficient approach to

estimate the neuron number and their initial values is required. In CBIR, the positive and fuzzy samples tend to form clusters, while the negative samples often scatter around the classes. Taking into account the fast-response requirement and the local multi-cluster nature of the samples, we employ a two-stage clustering process to determine the subnet centers. The feedback samples are clustered according to their types: relevant, irrelevant, and fuzzy. This will then be used to construct the subnets of the PFRBFN.

Fuzzy C-means (FCM) clustering is one of the most widely used clustering algorithms [102]. It considers each data point as potentially belonging to multiple clusters through membership functions. Notwithstanding its attractive features, FCM suffers from sensitivity towards the initial conditions, namely, the number of clusters and their initial estimates. In order to rectify this difficulty, we employ a two-stage clustering strategy in this work. First, subtractive clustering is employed as a preprocessing step to estimate the number and initial values of the clusters [103]. Subtractive clustering is used as it is fast, efficient and does not require the number of clusters to be specified *a priori*. The estimates obtained by subtractive clustering are then employed by FCM as its initial center estimates to perform clustering based on iterative optimization.

Subtractive clustering assumes each sample as a potential cluster center. It computes a potential field which determines the likelihood of a sample being a cluster center. Let $\{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{R}^R$ be a set of R -dimensional samples to be clustered. The initial potential function $P_{k=1}(i)$ of the i -th sample $\mathbf{x}_i$, expressed in terms of the Euclidean distance to the other samples $\mathbf{x}_j$, is defined as:

$$P_{k=1}(i) = \sum_{j=1}^n \exp \left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{r_a^2} \right) \quad i = 1, \dots, n \quad (4.2)$$

where r_a is a positive coefficient defining the range of the field. The potential function has large values at densely populated neighborhoods, suggesting strong likelihood that clusters may exist in these regions. The subtractive clustering algorithm can be summarized as follows:

- (i) Compute $P_{k=1}(i)$ for $i=1, \dots, n$ and select the sample with the highest potential as the first cluster center. Let $\mathbf{x}_1^*$ and P_1^* denote the first cluster center and its potential, respectively.
- (ii) For $k=2, \dots, K$, update the potential of each sample according to:

$$P_k(i) = P_{k-1}(i) - P_{k-1}^* \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_{k-1}^*\|^2}{r_b^2}\right) \quad i=1, \dots, n \quad (4.3)$$

where $\mathbf{x}_{k-1}^*$ and P_{k-1}^* are the $(k-1)$ -th cluster center and its potential value, r_b is a positive coefficient defining the neighborhood radius for potential reduction, and K is the maximum number of potential clusters. Equation (4.3) serves to remove the residual potential of the $(k-1)$ -th cluster center from the current k -th iteration field. The samples that are close to the $(k-1)$ -th cluster center will experience greater reduction in potential, hence reducing their likelihood to be chosen as the next center. Let $\mathbf{x}_k^*$ be the sample data with the maximum potential P_k^* in the current k -th iteration, the following criteria are used to determine whether it should be selected as the current cluster center:

if $\frac{P_k^*}{P_1^*} > \varepsilon_A$, accept $\mathbf{x}_k^*$ as the k -th cluster center

else if $\frac{P_k^*}{P_1^*} < \varepsilon_R$, reject $\mathbf{x}_k^*$ and terminate the clustering process

```

else if  $\frac{P_k^*}{P_1^*} + \frac{d_{\min}}{r_a} \geq 1$ , accept  $\mathbf{x}_k^*$  as a cluster center

else    reject  $\mathbf{x}_k^*$  and set its potential to zero ( $P_k^* \leftarrow 0$ ), and repeat the process
        with the sample with the next highest potential

endif

(iii) Repeat step (ii) until the termination criterion is satisfied or the maximum number of
iteration is reached.

```

In step (ii), ε_A is the acceptance ratio above which a sample will be accepted as a cluster center, ε_R is the rejection ratio below which a sample will be rejected, and $d_{\min}$ is the shortest distance between $\mathbf{x}_k^*$ and all previously found cluster centers. If the potential of the sample falls between the acceptance and rejection ratios, we will accept it only if it achieves a good compromise between having a reasonable potential and being sufficiently far from all existing cluster centers.

After subtractive clustering, we obtain a set of c cluster centers $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k, \dots, \mathbf{v}_c\}$, which is then used as the initial center estimates for FCM. FCM conducts clustering by minimizing the following objective function:

$$J_m(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^n \sum_{k=1}^c (\mu_{ik})^m (d_{ik})^2 \quad (4.4)$$

where $\mathbf{U} = [\mu_{ik}]$ with μ_{ik} representing the membership of $\mathbf{x}_i$ in the k -th cluster $\mathbf{v}_k$, $d_{ik} = \|\mathbf{x}_i - \mathbf{v}_k\|$ is the Euclidian distance between $\mathbf{x}_i$ and $\mathbf{v}_k$, and $m \geq 1$ is the fuzzification factor. In order to minimize $J_m(\mathbf{U}, \mathbf{V})$, it has been shown that the following conditions will be satisfied [102]:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c (d_{ik}/d_{jk})^{2/(m-1)}} \quad (4.5)$$

$$\mathbf{v}_k = \frac{\sum_{i=1}^n (\mu_{ik})^m \mathbf{x}_i}{\sum_{i=1}^n (\mu_{ik})^m} \quad (4.6)$$

under the following constraints:

$$\forall_{\substack{1 \leq i \leq n \\ 1 \leq k \leq c}} \mu_{ik} \in [0, 1] \quad (4.7)$$

$$\sum_{k=1}^c \mu_{ik} = 1 \quad (4.8)$$

$$0 < \sum_{i=1}^n \mu_{ik} < n \quad (4.9)$$

The complete two-stage clustering for the construction of subnets can be summarized as follows:

For each relevant, irrelevant, and fuzzy cluster types:

- (i) Obtain the initial cluster centers $\{\mathbf{v}_k\}_{k=1}^c$ from subtractive clustering.
- (ii) Calculate the fuzzy partition matrix $\mathbf{U} = [\mu_{ik}]$ according to (4.5).
- (iii) Update the fuzzy cluster centers $\{\mathbf{v}_k\}_{k=1}^c$ using (4.6).
- (iv) Compute the cost function $J_m(\mathbf{U}, \mathbf{V})$ in (4.4). Terminate the clustering process if $J_m(\mathbf{U}, \mathbf{V})$ is less than a predefined threshold δ or the maximum number of iterations is reached. Otherwise, repeat steps (ii)-(iv).
- (v) Initialize the RBF centers of the subnet as the centroid of each cluster.

After clustering, three sets of separate clusters are obtained, relevant, irrelevant and fuzzy

sets. Let X be the set of all training images. The representations for the cluster formation of X are given as:

$$X = P \cup N \cup F \quad (4.10)$$

$$P = \bigcup_{i=1}^{I_P} P_i \quad N = \bigcup_{j=1}^{I_N} N_j \quad F = \bigcup_{k=1}^{I_F} F_k \quad (4.11)$$

where P , N , and F denote the positive, negative, and fuzzy cluster types. P_i , N_j , and F_k are the i -th positive cluster, j -th negative cluster, and k -th fuzzy cluster. I_P , I_N and I_F are the number of RBF units for the positive, negative and fuzzy images, respectively.

4.3.4 Estimation of Soft Relevance Membership Function for Fuzzy Images

In this study, relevance feedback is implemented as an online supervised learning process by adjusting the underlying parameters of the network. The error function E_t at the t -th iteration is defined as:

$$E_t = \frac{1}{2} \sum_{j=1}^{N_T} e_{jt}^2 = \frac{1}{2} \sum_{j=1}^{N_T} (Y_t(\mathbf{x}_j) - F_t(\mathbf{x}_j))^2 \quad (4.12)$$

where N_T is the total number of training samples, and e_{jt} is the error signal for the j -th training sample $\mathbf{x}_j$ at the t -th iteration. $F_t(\mathbf{x}_j)$ and $Y_t(\mathbf{x}_j)$ represent the actual and desired network output for $\mathbf{x}_j$, respectively. We set $Y_t(\mathbf{x}_j)$ to 1 and 0 for $\mathbf{x}_j$ associated with the positive and negative feedback, respectively. The issue remains to be addressed is the estimation of the appropriate output values for the fuzzy samples. The desired values should reflect the degree of relevance embedded in the user perception of image similarity. Our objective, therefore, is to determine a soft relevance membership function $s(\mathbf{x}_j): \mathbb{R}^R \rightarrow [0,1]$

that assesses each fuzzy sample and maps it into a proper output value between $[0,1]$.

We propose an *a posteriori* estimator to determine $s(\mathbf{x}_j)$. The problem is formulated into the estimation of the conditional probability $P(\omega_r | \mathbf{x}_j)$, namely, given a fuzzy sample $\mathbf{x}_j$, what is the probability that it belongs to the relevant class ω_r . In CBIR systems, images are indexed and represented as feature vectors $\mathbf{x}_j = [\mathbf{x}_{j1}^T \ \mathbf{x}_{j2}^T \ \dots \ \mathbf{x}_{jM}^T]^T$ consisting of M feature subvectors $\mathbf{x}_{jm}$, $m=1,2,\dots,M$. Each $\mathbf{x}_{jm}$ is a d_m -dimensional feature subvector extracted from color histogram, wavelet moments, among others. The following estimation principle that assumes equal emphasis on each of the M features towards the overall contribution is adopted [104]:

$$s(\mathbf{x}_j) = P(\omega_r | \mathbf{x}_j) = \frac{1}{M} \sum_{m=1}^M P(\omega_r | \mathbf{x}_{jm}) \quad (4.13)$$

where $P(\omega_r | \mathbf{x}_{jm})$ is the *a posteriori* probability of the m -th subvector $\mathbf{x}_{jm}$. It has been demonstrated that the sum rule outperforms other classifier combination schemes such as the product rule, min rule, max rule, median rule, and majority voting [104]. According to the Bayesian theorem, we can rewrite (4.13) as:

$$s(\mathbf{x}_j) = \frac{1}{M} \sum_{m=1}^M \frac{p(\mathbf{x}_{jm} | \omega_r)P(\omega_r)}{p(\mathbf{x}_{jm} | \omega_r)P(\omega_r) + p(\mathbf{x}_{jm} | \omega_i)P(\omega_i)} \quad (4.14)$$

where ω_r and ω_i denote the relevant and irrelevant classes, respectively. $P(\omega_r)$ and $P(\omega_i)$ are the prior probabilities of the relevant and irrelevant classes. They can be estimated from the feedback samples using:

$$P(\omega_r) = \frac{n_r}{n_r + n_i} \quad (4.15)$$

$$P(\omega_i) = 1 - P(\omega_r) \quad (4.16)$$

where n_r and n_i are the total number of relevant and irrelevant samples accumulated over previous feedback iterations, $p(\mathbf{x}_{jm} | \omega_r)$ and $p(\mathbf{x}_{jm} | \omega_i)$ are the class conditional probability density functions (pdf) of $\mathbf{x}_{jm}$ for the relevant and irrelevant classes, respectively. If each feature vector is assumed to be Gaussian-distributed, the class pdf can be expressed as:

$$p(\mathbf{x}_{jm} | \omega_q) = \frac{1}{(2\pi)^{d_m/2} |\Sigma_m^q|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x}_{jm} - \boldsymbol{\mu}_m^q)^T \Sigma_m^q^{-1} (\mathbf{x}_{jm} - \boldsymbol{\mu}_m^q) \right] \quad (4.17)$$

where $\omega_q \in \{\omega_r, \omega_i\}$, $\boldsymbol{\mu}_m^q$ is the mean vector and Σ_m^q is the covariance matrix for the m -th feature vector of class ω_q . They can be estimated using N_q training samples $\mathbf{x}_{jm}^q$ for each relevant or irrelevant class ω_q :

$$\boldsymbol{\mu}_m^q = \frac{1}{N_q} \sum_{j=1}^{N_q} \mathbf{x}_{jm}^q \quad (4.18)$$

$$\Sigma_m^q = \frac{1}{(N_q - 1)} \sum_{j=1}^{N_q} (\mathbf{x}_{jm}^q - \boldsymbol{\mu}_m^q)(\mathbf{x}_{jm}^q - \boldsymbol{\mu}_m^q)^T \quad (4.19)$$

As the number of training samples is usually small in relevance feedback, the matrix Σ_m^q may experience singularity. To address the problem, an approach known as linear shrinkage estimator that incorporates a regularized diagonal matrix is adopted [105], [106]:

$$\hat{\Sigma}_m^q = (1 - \lambda) \Sigma_m^q + \frac{\lambda}{d_m} \text{tr}[\Sigma_m^q] \mathbf{I} \quad (4.20)$$

where $\text{tr}[\Sigma_m^q]$ denotes the trace of Σ_m^q , d_m is the dimension of the feature space, and $0 < \lambda < 1$ controls the amount of shrinkage towards the identity matrix $\mathbf{I}$.

Therefore, the desired output values for the PFRBFN network can be summarized as

follows:

$$Y_t(\mathbf{x}_j) = \begin{cases} 0, & \mathbf{x}_j \in N \\ 1, & \mathbf{x}_j \in P \\ s(\mathbf{x}_j), & \mathbf{x}_j \in F \end{cases} \quad (4.21)$$

which provides the soft relevance membership function for the positive (P), negative (N), and fuzzy (F) feedback images.

4.3.5 PFRBFN Training Algorithm

After two-stage clustering to determine the subnet structure, and estimation of soft relevance membership function for feedback samples, we are ready to discuss the learning process of the PFRBFN. The kernel function of the RBF units in the PFRBFN is given by:

$$f(\mathbf{x}, \mathbf{v}_{\alpha i}^t, \sigma_{\alpha i}^t) = \exp\left(-\frac{(\mathbf{x} - \mathbf{v}_{\alpha i}^t)^T \mathbf{\Lambda} (\mathbf{x} - \mathbf{v}_{\alpha i}^t)}{2(\sigma_{\alpha i}^t)^2}\right), \quad t=1, 2, \dots, T, \quad i=1, 2, \dots, I_{\alpha} \quad (4.22)$$

where $\alpha \in \{P, N, F\}$ is the cluster type characterizing the feedback, i and I_{α} are the index and number of RBF units for feedback type α , t is the iteration number, and T is the total number of feedback iterations. $\mathbf{v}_{\alpha i}^t$ and $\sigma_{\alpha i}^t$ are the center and width for the i -th RBF unit with α subnet type at the t -th iteration. The values $\mathbf{v}_{\alpha i}^t$ are initialized based on the outputs obtained using the two-stage clustering process. The matrix $\mathbf{\Lambda} = \text{diag}[\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_R}]$ is a diagonal matrix that denotes the relative importance of different feature components with σ_r , $r=1, 2, \dots, R$ representing the standard deviation of all relevant samples along the r -th dimension. This approach is based on the observation that consistent values of a particular feature component across all the relevant samples suggest that it is reliable in determining

image similarity. Therefore, the feature components with more consistent values are assigned greater weights [10].

Based on the kernel function in (4.22), the update function $U_t(\mathbf{x})$ is given as:

$$U_t(\mathbf{x}) = \sum_{\alpha \in \{P, N, F\}} \sum_{i=1}^{I_\alpha} w_{\alpha i}^t f(\mathbf{x}, \mathbf{v}_{\alpha i}^t, \sigma_{\alpha i}^t) \quad (4.23)$$

where $w_{\alpha i}^t$ is the output connection weight. Substituting (4.23) into (4.1), the PFRBFN output function $F_t(\mathbf{x})$ can be expressed as:

$$F_t(\mathbf{x}) = \begin{cases} F_{t-1}(\mathbf{x}) + \sum_{\alpha \in \{P, N, F\}} \sum_{i=1}^{I_\alpha} w_{\alpha i}^t f(\mathbf{x}, \mathbf{v}_{\alpha i}^t, \sigma_{\alpha i}^t), & t = 1, 2, \dots, T \\ 0, & t = 0 \end{cases} \quad (4.24)$$

The system training of the CBIR system at each feedback iteration is performed by minimizing the cost function E_t in (4.25) with respect to the network parameters of $U_t(\mathbf{x})$, namely, $\boldsymbol{\theta} = \{w_{\alpha i}^t, \mathbf{v}_{\alpha i}^t, \sigma_{\alpha i}^t \mid \alpha \in \{P, N, F\}, i = 1, 2, \dots, I_\alpha\}$. The learning process can be represented as:

$$\boldsymbol{\theta} = \arg \min_{\boldsymbol{\theta} \in \Theta} (E_t) = \arg \min_{\boldsymbol{\theta} \in \Theta} \left(\frac{1}{2} \sum_{j=1}^{N_T} (Y_t(\mathbf{x}_j) - F_t(\mathbf{x}_j))^2 \right) \quad (4.25)$$

where Θ is the solution space of the parametric vector $\boldsymbol{\theta}$.

The training is achieved using gradient-descent algorithm based on all the feedback samples accumulated over all previous iterations. The update equations for $w_{\alpha i}^t$, $\mathbf{v}_{\alpha i}^t$, and $\sigma_{\alpha i}^t$ are summarized as follows:

- (i) Weight estimation at the k -th learning iteration:

$$\frac{\partial E_t(k)}{\partial w'_{\alpha i}(k)} = -\sum_{j=1}^{N_t} e_{jt}(k) f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}(k), \sigma'_{\alpha i}(k)) \quad (4.26)$$

$$w'_{\alpha i}(k+1) = w'_{\alpha i}(k) - \eta_1 \frac{\partial E_t(k)}{\partial w'_{\alpha i}(k)}, \quad \alpha \in \{P, N, F\}, \quad i = 1, 2, \dots, I_\alpha \quad (4.27)$$

(ii) Center estimation at the k -th learning iteration:

$$\frac{\partial E_t(k)}{\partial \mathbf{v}'_{\alpha i}(k)} = -w'_{\alpha i}(k) \sum_{j=1}^{N_t} e_{jt}(k) f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}(k), \sigma'_{\alpha i}(k)) \frac{\boldsymbol{\Lambda}(\mathbf{x}_j - \mathbf{v}'_{\alpha i}(k))}{(\sigma'_{\alpha i}(k))^2} \quad (4.28)$$

$$\mathbf{v}'_{\alpha i}(k+1) = \mathbf{v}'_{\alpha i}(k) - \eta_2 \frac{\partial E_t(k)}{\partial \mathbf{v}'_{\alpha i}(k)}, \quad \alpha \in \{P, N, F\}, \quad i = 1, 2, \dots, I_\alpha \quad (4.29)$$

(iii) Width estimation at the k -th learning iteration:

$$\frac{\partial E_t(k)}{\partial \sigma'_{\alpha i}(k)} = -w'_{\alpha i}(k) \sum_{j=1}^{N_t} e_{jt}(k) f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}(k), \sigma'_{\alpha i}(k)) \frac{(\mathbf{x}_j - \mathbf{v}'_{\alpha i}(k))^T \boldsymbol{\Lambda}(\mathbf{x}_j - \mathbf{v}'_{\alpha i}(k))}{(\sigma'_{\alpha i}(k))^3} \quad (4.30)$$

$$\sigma'_{\alpha i}(k+1) = \sigma'_{\alpha i}(k) - \eta_3 \frac{\partial E_t(k)}{\partial \sigma'_{\alpha i}(k)}, \quad \alpha \in \{P, N, F\}, \quad i = 1, 2, \dots, I_\alpha \quad (4.31)$$

(iv) Repeat steps (i)-(iii) until convergence or a maximum number of iterations is reached.

The term $e_{jt}(k)$ is the error signal of the j -th training sample $\mathbf{x}_j$ at the k -th learning iteration,

and η_1 , η_2 and η_3 are learning parameters for $w'_{\alpha i}$, $\mathbf{v}'_{\alpha i}$, and $\sigma'_{\alpha i}$, respectively.

It is noted that the gradient-descent algorithm given in (4.26)-(4.31) does not involve back-propagation of the error signal. Thus, it requires less training time to converge when compared with other neural networks such as multilayer perceptron. The size of the training set at the t -th feedback iteration is the total number of feedback images accumulated over the past t iterations. This in turn depends on the number of displayed images shown to the users

for feedback. As a result, the number of training samples will increase with respect to the number of feedback iterations. During the retrieval process, the number of clusters for the network does not remain constant. This is because the training samples are used to generate a new update network at each feedback iteration, which consists of new clusters as shown in Figure 4.6(b). Therefore, the number of clusters for the PFRBFN will increase throughout the retrieval process in order to track the progressive user information need.

The flowchart summarizing the proposed PFRBFN scheme is given in Figure 4.7.

Figure 4.7 General overview of the proposed PFRBFN scheme.

4.4 Experimental Results

The performance of the PFRBFN is evaluated on an image database consisting of 10,000 natural images with 100 different categories obtained from the Corel Gallery product [107]. The categories are predefined by the Corel Photo Gallery based on their semantic concepts. It covers a wide variety of subjects, as shown in Figure 4.8. This image database will be used throughout the whole thesis.

Figure 4.8 Selected sample images from the database, which are used in the experiments.

The proposed PFRBFN is employed in our retrieval system. The two-stage clustering is utilized to determine the subnet structure of the relevant, irrelevant and fuzzy images. During the subtractive clustering, the following parameters are adopted: r_a is set to 0.2, 0.25, and 0.2 for relevant, irrelevant and fuzzy samples, respectively, with $r_b = 4r_a$, $\varepsilon_A = 0.5$, and $\varepsilon_R = 0.15$. The subsequent FCM clustering uses the following parameters: $m = 3$ and $\delta = 0.05$. The

PFRBFN is constructed progressively and trained iteratively using the learning algorithms in (4.26)-(4.31). The learning step-sizes are taken as $\eta_1 = 0.01$, $\eta_2 = 0.0001$, and $\eta_3 = 0.02$. The values of the parameters are determined using the following rule of thumb. The approximate values of the parameters are first selected according to previous literature findings and empirical study. These are followed by experimental fine-tuning. The predefined variables in the proposed method can be classified into two categories: (i) variables used in the clustering stage of PFRBFN creation $\{r_a, r_b, \varepsilon_A, \varepsilon_R, m, \delta\}$, and (ii) those involved in the training stage of PFRBFN $\{\eta_1, \eta_2, \eta_3\}$. The impact of the underlying parameters used in FCM clustering and steepest-descent optimization has been studied in previous literatures such as [102] and [44]. In this work, we have conducted many experiments using different values of predefined variables and more than 100 different query images. Experimental results show that the system produces satisfactory results consistently for different query images and variable values, so long as they are within a reasonable order of magnitude.

There are two parts in the experiments, namely, objective performance evaluation and subjective performance evaluation. For objective evaluation, 100 queries from 100 semantic concepts are selected for evaluation. For subjective evaluation, 6 users are invited to evaluate the retrieval system. A total of 150 queries from 100 semantic concepts are used for evaluation. The size of the training set depends on the number of images shown to the user for feedback. The testing set is the whole image database containing 10,000 images.

4.4.1 Performance Evaluation based on Objective Measure

In our experiment, we employ both the objective and subjective measures to evaluate the performance of the proposed PFRBFN method and compare it with the adaptive radial basis function network (ARBFN) method [20]. The ARBFN method characterizes the query by

multi-class models, an inherent strategy of local modeling, and associates those relevant samples as the models. The irrelevant samples are then used to modify the multi-class models in such a way that the models are moved slightly away from the irrelevant samples.

The objective measure, which is based on the Corel's predefined ground truth, is used when there is no ambiguity in the user perception of image similarity, or in other words, there are no fuzzy feedbacks. In this case, the retrieved images are considered to be relevant if they belong to the same category as the query image. Precision versus recall (PR) curve is adopted in our experiment. The definitions of precision and recall have been given in (2.2) and (2.3). In our experiment, 100 queries with one from each category, as given in Figure 4.8, are selected for evaluation. For each query, the top 25 retrieved images are displayed to the users for feedback. The precision and recall rates are averaged over all the queries. The average precision versus recall (APR) graph after 5 feedback iterations is shown in Figure 4.9. From the figure, we observe that the PFRBFN method outperforms the ARBFN method. The PFRBFN method provides higher recall rate at the same precision level, and higher precision rate at the same recall level. For instance, the PFRBFN method offers a precision rate of 70% compared to 50% offered by the ARBFN method at 30% recall rate. This reflects the effectiveness of the proposed PFRBFN method. The ARBFN method is heuristic as the learning process to optimize the underling network parameters is inadequate.

Figure 4.9 The average precision versus recall graph of the PFRBFN and ARBFN methods (after 5 feedback iterations).

We have also adopted the performance measure of retrieval accuracy (RA) to evaluate the retrieval system, which has been defined in (2.1). The retrieval accuracy performance of the PFRBFN method is given in Figure 4.10. The retrieval accuracy is averaged over 100 test queries, and seven iterations of feedbacks are recorded. It is observed that for both top 25 and 50 returned images, the retrieval accuracy of the PFRBFN method increases quickly in the first few iterations. This is desirable since the user can obtain satisfactory results within a few iterations. Further, they reach steady-state retrieval accuracy of 81% (25 returned images) and 72% (50 returned images) in about 5 feedback iterations. In comparison, the ARBFN method achieves lower steady-state retrieval accuracy in both cases. The PFRBFN method offers improvements of 7% (25 returned images) and 6% (50 returned images) for steady-state retrieval accuracy over the ARBFN method.

(a)

(b)

Figure 4.10 Retrieval accuracy of the PFRBFN and ARBFN methods. (a) Retrieval accuracy in top 25 results, (b) Retrieval accuracy in top 50 results.

4.4.2 Performance Evaluation based on Subjective Measure

Now we study the performance of the retrieval system when the users experience uncertainty in the feedback images. In other words, the users will select fuzzy option in at least some of the feedbacks. Under these circumstances, we cannot employ ground truth-based objective performance measures because the fuzzy images selected by the users may span across different categories. In view of this, the best performance measure will be subjective human evaluation since the perception of image similarity, especially on fuzzy images, is user-dependent. Six users are invited to evaluate the retrieval system. A total of 150 queries are used for evaluation. We introduce the following performance measures: total retrieval accuracy (TRA), and relevant retrieval accuracy (RRA) to evaluate the system performance:

$$\text{Relevant retrieval accuracy} = \frac{\text{relevant images retrieved in top } T \text{ returns}}{T} \quad (4.32)$$

$$\text{Total retrieval accuracy} = \frac{\text{relevant and fuzzy images retrieved in top } T \text{ returns}}{T} \quad (4.33)$$

where T is the number of retrieved images. The RRA is similar to the conventional RA when there are no ambiguities in the user interpretation of feedback images. Unfortunately, this criterion alone is inadequate in evaluating the performance of the system when the users choose some images as fuzzy since they satisfy, up to a certain extent, the user information need. Therefore, the TRA is introduced to incorporate both the relevant as well as fuzzy images as the desired images. An alternative perspective to interpreting the TRA and RRA is that they are the upper and lower bounds of effective retrieval accuracy. These give us an idea of how the PFRBFN method performs when fuzzy relevance feedback is included.

The performance of the PFRBFN method is given in Figure 4.11. The TRA and RRA graphs are averaged over all test queries and users. In the figures, the first iteration denotes the users' assessment of the initial retrieved images based on K-NN search. It is observed that

the RRA and TRA graphs of the PFRBFN method increase quickly for both top 25 and 50 returned images. Further, they reach steady-state RRA of 82% and TRA of 86% (25 returned images), with RRA of 74% and TRA of 80% (50 returned images) in about 5 feedback iterations. The effective retrieval accuracy should lie within the upper bound of TRA and lower bound of RRA. The results imply that there are, on average, 4% (25 returned images) and 6% fuzzy images (50 returned images) in these experiments. When compared with the ARBFN and MARS [10] methods. The MARS method adopts multi-level labeling in relevance feedback. It involves five discrete levels of (ir)relevance, namely, highly relevant, relevant, neutral, irrelevant, and highly irrelevant. The scores associated with each level are: 3, 1, 0, -1, -3, respectively. It is noted that neutral images are assigned the score of 0, which implies that they do not provide any contribution during the training process. In order to provide a comparative performance measure, we define the retrieval accuracy of the MARS method as the ratio of the sum of both “highly relevant” and “relevant” images in top T returned images. It is clear that the PFRBFN method provides superior results. The performance of the MARS system is mediocre because it is based on heuristic updates of intra- and inter-feature weights. Further, the feedback process of the MARS system can be very tedious for the users as they need to classify each image in every feedback iteration into one of the multiple levels. In contrast, the proposed PFRBFN framework simplifies the feedback process, while maintaining the priority of the user information need.

(a)

(b)

Figure 4.11 Performance comparison of PFRBFN, ARBFN and MARS based on user subjectivity. (a) Retrieval accuracy in top 25 results, (b) Retrieval accuracy in top 50 results.

4.4.3 Computational Efficiency Analysis

The experiments are conducted on a PC with the following specifications: Pentium4 2.4GHz processor, 512M RAM, Windows XP, Matlab 6.5. In our experiments, the average retrieval time for each feedback iteration using both the PFRBFN and ARBFN methods is comparable, ranging from 5 to 10 seconds for the top 25 results.

The overall computational complexity of the PFRBFN method during each feedback iteration consists of two main stages: (i) network training (cluster formation, soft membership determination, and network parameter estimation), and (ii) distance metric calculation of all images in the database during similarity matching. It is noted that the computation of network parameter estimation dominates the training process. The computational complexities for network training and distance computation are given by $O(RKN_TN_\alpha)$ and $O(RN_\alpha Q)$, respectively, where R is the dimensionality of the feature vector, K is the number of training iteration, N_T is the total number of training samples, N_α is the number of RBF units, and Q is the number of images in the database. The overall complexity of the algorithm is given by $O(RKN_TN_\alpha + RN_\alpha Q)$. In typical medium to large database retrieval applications where $Q > KN_T$, the effective complexity of the proposed algorithm is $O(RN_\alpha Q)$, which is similar to that of the ARBFN method in [20]. It is observed that the proposed method offers performance improvement over the ARBFN method, while maintaining comparable computational cost.

4.4.4 Illustrations of Retrieval Results Using PFRBFN

In an image retrieval session, a user is interested in finding some home pets, particularly dogs. Given the query image of a dog, the initial retrieval results using the k-nearest-neighbor method are shown in Figure 4.12(a). It can be observed that they consist of 8 dogs, 2 cats and

other non-pet images.

The user will subsequently mark the dogs as relevant and other non-pet images as irrelevant. The ambiguities arise for the categorization of the cats as it depends on individual user subjectivity. If the user marks the cat images as irrelevant, he/she may be concerned that some information will be lost as the cat images satisfy, up to a certain extent, his/her information need. On the other hand, if the user marks the cat images as relevant, he/she may be concerned that his/her information priority, namely the dogs, will be obscured by the cat images. In other words, he/she is worried that the dog images will appear after the cat images. In either case, the user faces a dilemma if a hard-decision has to be made on the relevance of the cat images. Therefore, it is desirable if the system can categorize those cat images as fuzzy and assign appropriate membership function to them accordingly. It is worth noting that in the graphical user interface in Figure 4.12, the displayed images are ranked in decreasing order of priority from left to right, and top to bottom.

Consider the conventional binary labeling approach where the user has to make a hard-decision on the relevance of the cat images. If the user marks the cats as irrelevant, the retrieval results after 1 feedback iteration are given in Figure 4.12(b). It is observed that the retrieved results contain 11 dogs and 0 cats. The retrieval results are unsatisfactory because they disregard the user's partial information need that may include cat images as the candidates. On the other hand, if the user marks the cats as relevant, 8 dogs and 5 cats (see Figure 4.12(c)) are retrieved. The results are undesirable as there are less dog images as compared to Figure 4.12(b). Further, the ranking of the dog and cat images is mixed up. Ideally, we would like to have dogs ranked higher than the cats, which reflects the user information priority. In this case, however, some cats (images 6 and 7) in fact are ranked higher than the dogs (images 8-10). Ranking is particularly important in applications such as image retrieval through bandwidth-limited, display-constrained devices, e.g. PDA, where

images with the highest priority should be transmitted first. Obviously, inappropriate ranking in this case will give rise to the problem where the most desired images may not be shown to the user. In contrast, the proposed fuzzy relevance framework provides the users with the flexibility to classify an image as either relevant, irrelevant or fuzzy. If the user selects the cats as fuzzy, the retrieval results are given in Figure 4.12(d). It is observed that the proposed method manages to retrieve 11 dog and 3 cat images. This is clearly superior to the binary labeling results in Figure 4.12(b) and (c). Further, the ranking of the dog images is higher than the cat images, hence reflecting the priorities of the user information need.

Conventional relevance feedback approaches adopt the binary logic to model the user's perception of image similarity. This is, however, counter-intuitive as user interpretation of visual content is fuzzy and nonstationary in nature. The case study in Figure 4.12 highlights this dilemma, and illustrates that the proposed PFRBFN scheme is instrumental in addressing this issue by utilizing the notion of fuzzy relevance feedback.

Figure 4.12 Illustrative case study. (a) Initial retrieval results based on K-NN search, (b) Retrieval results with user marking the cat images as irrelevant, (c) Retrieval results with user marking the cat images as relevant, (d) Retrieval results with user marking the cat images as fuzzy.

4.5 Summary

This chapter presents a new framework called fuzzy relevance feedback in interactive content-based image retrieval systems. In contrast to conventional approaches that are based on binary labeling, the new method provides a natural way to model the user interpretation of image similarity. The importance or relevance of fuzzy images is estimated based on an *a posteriori* estimator. A PFRBFN is constructed progressively to reduce the computational complexity. A gradient descent-based learning strategy is then employed to estimate the underlying network parameters. Experimental results demonstrate that the PFRBFN method is effective in performing image retrieval, while addressing the priority of user information need.

Chapter 5

A Pseudo-Label Fuzzy Support Vector

Machine Framework

5.1 Introduction

This chapter studies another state-of-the-art computational intelligence technique, namely, support vector machines (SVMs) for image retrieval. SVMs and radial basis function (RBF) networks are both powerful learning machines for many real-world applications. SVMs differ from RBF networks in several ways. The RBF networks minimize a quadratic cost function, regardless of what the learning task is. In contrast, the support vector learning algorithm maximizes the margin of separation between the positive and negative samples [44]. Further, the computational complexity of SVMs does not depend on the dimensionality of the input space. The curse of dimensionality is bypassed by focusing on the dual problem for performing the constrained optimization problem [44]. In terms of running time, SVMs are currently slower than RBF networks for a similar generalization performance.

Conventional relevance feedback in CBIR systems uses only the labeled images for learning. Image labeling, however, is a time consuming task and users are often unwilling to label too many images during the feedback process. This gives rise to the small sample problem where learning from a small number of training samples restricts the retrieval performance. To address this problem, some works have been done to incorporate the

unlabeled data to improve the learning performance. Discriminant Expectation Maximization (D-EM) algorithm has been introduced to incorporate the unlabeled samples to estimate the underlying probability distribution [21]. The results are promising, but the computational complexity can be significant for large databases. Transductive support vector machine (TSVM) for text classification has been proposed to tackle the problem by incorporating the unlabeled data [81]. It has also been applied for image retrieval [79]. The method proposes to incorporate unlabeled images to train an initial SVM, followed by standard active learning. It is, however, observed that the performance of this method may be unstable in some cases. Incorporating prior knowledge into the SVM has also been introduced to resolve the small sample problem [82]. All these proposed methods show some promising outcomes, however few can learn from the labeled and unlabeled data effectively.

In view of this, we propose a technique based on the concept of pseudo-labeling in order to enlarge the training data set. By making use of the potential role of unlabeled images, certain informative unlabeled images which are 'similar' to the labeled ones in terms of the visual features are determined for pseudo-labeling. As the name implies, a pseudo-labeled image is an image not labeled explicitly by the users, but estimated using a fuzzy rule. Therefore, it contains a certain degree of uncertainty or fuzziness in its class information. The main principles of the proposed algorithm can be described concisely as: identification of useful unlabeled images as candidates for pseudo-labeling, integration of fuzziness embedded in the pseudo-labeled images into the scheme, and development of appropriate computational techniques to optimize the learning. The requirements for systematic data identification, intelligent information integration, and robust learning point to computational intelligence techniques. Two categories of computational intelligence, namely SVM and fuzzy logic are employed in this chapter. Fuzzy support vector machine (FSVM), an extended version of SVM, takes into account the fuzzy nature of some training samples during its training [108].

In order to exploit the advantages of pseudo-labeling, active learning and the structure of FSVM, we develop a unified framework called pseudo-label fuzzy support vector machine (PLFSVM) to perform content-based image retrieval. By exploiting the characteristics of the labeled images, unlabeled images are chosen carefully and assigned different pseudo-labels such as ‘relevant’ or ‘irrelevant’. This process will enlarge the training data set. A fuzzy membership function is employed to estimate the class membership of the pseudo-labeled images. The fuzzy information is then integrated into the FSVM for active learning.

Chapter 4 deals with the issue of fuzzy perception of user feedback, where the unique batch process of feedback is utilized to build a recursive network structure to reduce computational cost. In contrast, SVMs are not effective in handling the problems involving recursive-structure training. This chapter utilizes SVMs to address the small sample problem in CBIR systems. Since the number of training samples is small, the adoption of SVMs will not be too computationally intensive. Further, they have been shown to perform well on small samples.

The rest of this chapter is organized as follows. Section 5.2 explains the small sample problem and the formulation of the PLFSVM. In Section 5.3, we describe FSVM and discuss the pseudo-label estimation scheme in detail. We further explore the fuzzy membership function which is developed to determine the implicit class membership of the pseudo-labeled images. In Section 5.4, experimental results using the proposed method are discussed and compared with other techniques. Finally, concluding remarks are given in Section 5.5.

5.2 Learning from Small Samples

5.2.1 Small Sample Problem

In relevance feedback of CBIR systems, the number of training samples is usually small since

it is not user-friendly to let the users label too many images. Thus, the small sample problem arises in interactive CBIR systems. Learning from such a small number of training samples may not produce good retrieval results, even for a powerful learning machine such as SVM. Therefore, it is imperative to find solutions to solve the small sample problem faced by relevance feedback.

Realizing that obtaining a large number of labeled images is labor intensive while unlabeled images are readily available, we propose to utilize the unlabeled images to augment the available labeled images. Unlabeled images should be carefully chosen so that they will be beneficial to the retrieval performance; otherwise, they can degrade the performance if used improperly. Each selected, unlabeled image is assigned a pseudo-label of either 'relevant' or 'irrelevant' based on an algorithm to be explained in subsection 5.3.3. These pseudo-labeled images are fuzzy in nature since they are not explicitly labeled by the users. Therefore the potential imprecision embedded in their class information should be taken into consideration. Fuzzy logic provides a natural and flexible way in modeling the uncertainty of data. We utilize a fuzzy membership function to determine the degree of uncertainty for each pseudo-labeled image, hence putting into context the relative importance of these images. These pseudo-labeled samples are then combined with the labeled samples to train the SVM. The details of the algorithm will be explained in Section 5.3.

5.2.2 Active Learning with Pseudo-Labeled Images

Despite the excellent generalization performance of SVM, it is limited to crisp classification where each training example belongs to either one or the other class with equal importance. However, in many real-world applications, different training data have different degrees of importance and this fuzzy nature should be taken into consideration. Hence, FSVM is proposed to address this problem [108]. FSVM is an extension of SVM. It can take into

account different importance of training data. FSVM exhibits the following properties that motivate us to adopt it in our framework: integration of fuzzy data, strong theoretical foundation, and excellent generalization power.

We develop a unified PLFSVM framework that integrates the advantages of pseudo-labeling and FSVM. In contrast to most existing feedback approaches in CBIR systems that are concerned with only the labeled data, the proposed PLFSVM exploits the unlabeled data to enhance the retrieval performance. It is noted that the proposed PLFSVM differs from the traditional SVM in several ways. The PLFSVM is developed for solving the small sample problem by incorporating pseudo-labeled images, while traditional SVM can only handle labeled images. Further, PLFSVM requires less user workload, thus making it more appealing for practical applications such as image retrieval over bandwidth-limited network. Lastly, PLFSVM can take relative significance of the training samples into consideration, and hence, is more general and flexible.

Under the new framework, a few important issues in the development of the PLFSVM have to be addressed: (i) selection of the right images for pseudo-labeling, (ii) determination of the pseudo-labels, (iii) estimation of the degree of uncertainty associated with the pseudo-labeled images, and (iv) integration of pseudo-labeled images into FSVM-based active learning. To address these issues, the following ideas have been proposed: (i) a two-stage clustering algorithm is developed for choosing the unlabeled image for pseudo-labeling, (ii) a similarity measure analogous to the K-NN is proposed for pseudo-label assignment, (iii) a proper fuzzy membership function is developed to estimate the class membership of the pseudo-labeled images, and (iv) FSVM is employed by emphasizing the labeled images over the pseudo-labeled images during training.

SVM-based active learning selects samples that are closest to the current SVM decision boundary as the most informative points. Samples that are farthest away from the boundary

and on the positive side are considered as the most relevant images. We adopt the same selection strategy in this work. Integrating the merits of PLFSVM into active learning, we can achieve improved retrieval performance with less user labeling.

5.3 Pseudo-Label Fuzzy Support Vector Machine (PLFSVM)

5.3.1 Overview of the Proposed Framework

The proposed PLFSVM is a unified framework that incorporates pseudo-labeling into FSVM in the context of CBIR. The general overview of the framework is outlined as follows.

- (i) Given a user submitted query, the system performs the K-NN search using the Euclidean distance for similarity matching. The top l_0 most similar images are shown to the user for feedback.
- (ii) The user provides the feedback on the l_0 images as either relevant or irrelevant. Based on the l_0 labeled images, an initial SVM classifier is trained.
- (iii) The SVM active learning is employed by selecting l unlabeled images that are closest to the current SVM decision boundary for the user to label.
- (iv) After the user labels the l images, add them to the previously labeled training set.
- (v) A two-stage clustering is performed separately on the labeled relevant and irrelevant images. The formed clusters are then used for unlabeled images selection and pseudo-label assignment.
- (vi) A fuzzy metric is employed to evaluate the soft relevance membership of the pseudo-labeled images.
- (vii) An FSVM is trained using a hybrid of the labeled and pseudo-labeled images.
- (viii) Repeat steps (iii)-(vii) until the retrieval performance is satisfactory.

5.3.2 Formulation of FSVM

Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a set of n training samples, where $\mathbf{x}_i \in \mathcal{R}^P$ is a P -dimensional sample in the input space, and $y_i \in \{-1, 1\}$ is the class label of $\mathbf{x}_i$ for $i = 1, \dots, n$. In FSVM, each training sample is associated with a fuzzy membership value $\{\mu_i\}_{i=1}^n \in [0, 1]$. The membership value μ_i reflects the fidelity of the data, or in other words, how confident we are about the actual class information of the data. The higher its value, the more confident we are about its class label. The optimization problem of the FSVM is formulated as follows [108]:

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \mu_i \xi_i \\ & \text{subject to } y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n \end{aligned} \quad (5.1)$$

where $\mathbf{z} = \varphi(\mathbf{x})$ is the mapping function that maps the input data into a high-dimensional feature space. $\mathbf{w}$ is the normal vector of the hyperplane, b is the bias, and C is the regularization parameter. It is noted that the error term ξ_i is scaled by the membership value μ_i . The fuzzy membership values are used to weigh the soft penalty term in the cost function of SVM. The weighted soft penalty term reflects the relative fidelity of the training samples during training. Important samples with larger membership values will have more impact in the FSVM training than those with smaller membership values.

To solve the optimization problem in (5.1), the following Lagrangian is constructed similar to the standard SVM:

$$Q(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \mu_i \xi_i - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w} \cdot \mathbf{z}_i + b) - 1 + \xi_i) - \sum_{i=1}^n \beta_i \xi_i \quad (5.2)$$

where α_i and β_i are nonnegative Lagrange multipliers. The solution to this optimization problem is given by the saddle point of the Lagrangian. At the saddle points, the following

conditions hold:

$$\frac{\partial Q(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{z}_i = 0 \quad (5.3)$$

$$\frac{\partial Q(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial b} = -\sum_{i=1}^n \alpha_i y_i = 0 \quad (5.4)$$

$$\frac{\partial Q(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \xi_i} = \mu_i C - \alpha_i - \beta_i = 0 \quad (5.5)$$

By substituting (5.3)-(5.5) into the Lagrangian in (5.2), the optimization problem of FSVM can be transformed into its dual problem as follows:

$$\begin{aligned} & \text{maximize } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & \text{subject to } \sum_{i=1}^n y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq \mu_i C, \quad i = 1, \dots, n \end{aligned} \quad (5.6)$$

The KT conditions are now defined as:

$$\alpha_i [y_i (\mathbf{w} \cdot \mathbf{z}_i + b) - 1 + \xi_i] = 0, \quad i = 1, \dots, n \quad (5.7)$$

$$(\mu_i C - \alpha_i) \xi_i = 0, \quad i = 1, \dots, n \quad (5.8)$$

Solving the optimization problem of FSVM in (5.6) will lead to a decision function similar to (3.27), but with different support vectors and corresponding weights α_i .

5.3.3 Unlabeled Image Selection and Pseudo-Label Estimation

Careful incorporation of the unlabeled images is instrumental in achieving good retrieval results. If chosen inappropriately, the unlabeled images can degrade instead of improving the performance. In this work, we propose to exploit the characteristics of the labeled images to select the unlabeled images for pseudo-labeling. The basic idea is to identify certain samples

in the unlabeled image population which are ‘similar’ to the labeled images to perform pseudo-labeling and fuzzy membership estimation. This will enlarge the training data set.

The labeled relevant images usually exhibit local characteristics of image similarity. To exploit this property, it is desirable to adopt a multi-cluster local modeling strategy. Taking into account the local multi-cluster nature of image similarity, we employ a two-stage clustering process to determine the local clusters. The labeled samples are clustered according to their types: relevant or irrelevant. K-means clustering is one of the most widely used clustering algorithms [109]. K-means clustering finds a partition in which data points within each cluster are as close to each other as possible, and as far as possible from data points in the other clusters. It uses an iterative algorithm that minimizes the sum of distances from each data point to its respective cluster centroid for all the clusters. Notwithstanding its attractive features, K-means clustering suffers from sensitivity towards the number of chosen clusters and their initial estimates. In order to rectify this difficulty, we employ a two-stage clustering strategy in this work. First, subtractive clustering is employed as a preprocessing step to estimate the number and initial values of the clusters. These estimates are then employed by K-means to perform clustering based on iterative optimization.

After clustering, two sets of separate clusters are obtained, relevant and irrelevant sets. We use the heuristic argument similar to the K-NN technique, that is, closer data points tend to have similar class labels. We select K nearest unlabeled neighbors for each labeled cluster based on their Euclidean distances to the center of the labeled cluster. The label (relevant or irrelevant) of each labeled cluster is then propagated to the unlabeled neighbors. This is referred to as pseudo-labeling process. As the computational cost will increase with respect to the number of pseudo-labeled images, therefore, only the most ‘similar’ neighbor ($K = 1$) is selected in this work.

5.3.4 Estimation of Soft Relevance Membership Function for Pseudo-Labeled Images

Taking the fuzziness of the pseudo-labeled images into consideration, our objective here is to find a mapping $g(\mathbf{x}_p): \mathcal{R}^m \rightarrow [0,1]$ that assigns a proper relevance value between zero and one to each pseudo-labeled image $\mathbf{x}_p$. Motivated by the concepts of fuzzy logic, we propose to develop a fuzzy membership function as a soft value with respect to the relevance of the class labels.

Since clustering has been performed on each positive and negative class separately to get multiple clusters for each class (relevant or irrelevant), the obtained clusters in each class can be employed to generate the membership value. Assume that the pseudo-labels of the selected images have been determined as explained in the earlier section. Intuitively, the closer a pseudo-labeled image is to the nearest cluster of the same class label, the higher is its degree of relevance. In contrast, the closer a pseudo-labeled image is to the nearest cluster of the opposite class label, the lower is its degree of relevance. Based on this argument, an exponentially-based fuzzy function is selected:

$$w_1(\mathbf{x}_p) = \begin{cases} \exp\left(-a_1 \frac{\min_i (\mathbf{x}_p - \mathbf{v}_{Si})^T (\mathbf{x}_p - \mathbf{v}_{Si})}{\min_j (\mathbf{x}_p - \mathbf{v}_{Oj})^T (\mathbf{x}_p - \mathbf{v}_{Oj})}\right), & \text{if } \frac{\min_i (\mathbf{x}_p - \mathbf{v}_{Si})^T (\mathbf{x}_p - \mathbf{v}_{Si})}{\min_j (\mathbf{x}_p - \mathbf{v}_{Oj})^T (\mathbf{x}_p - \mathbf{v}_{Oj})} < 1 \\ 0, & \text{otherwise} \end{cases} \quad (5.9)$$

where $\mathbf{v}_{Si}$ denotes the center of the i th cluster with the same class label as the pseudo-labeled image $\mathbf{x}_p$, while $\mathbf{v}_{Oj}$ denotes the center of the j th cluster with the opposite class label to the pseudo-labeled image $\mathbf{x}_p$. $\min_i (\mathbf{x}_p - \mathbf{v}_{Si})^T (\mathbf{x}_p - \mathbf{v}_{Si})$ and $\min_j (\mathbf{x}_p - \mathbf{v}_{Oj})^T (\mathbf{x}_p - \mathbf{v}_{Oj})$ represent the distance between $\mathbf{x}_p$ and the nearest cluster centers with the same and opposite class labels, respectively. $a_1 > 0$ is a scaling factor. This membership function is divided into two

scenarios. If the distance ratio is smaller than 1, suggesting that the pseudo-labeled image is closer to the nearest cluster with the same class label, then we will estimate its soft relevance. Otherwise, if the pseudo-labeled image is closer to the nearest cluster with the opposite class label, a zero value is assigned.

Further, the fuzzy membership function of the pseudo-labeled samples will also depend on the agreement between the predicted label obtained in subsection 5.3.3, and the predicted label obtained using the trained SVM. The second factor of the fuzzy function is chosen as a sigmoid function defined as follows:

$$w_2(\mathbf{x}_p) = \begin{cases} \frac{1}{1 + \exp(-a_2 y)}, & \text{pseudo-label is positive} \\ \frac{1}{1 + \exp(a_2 y)}, & \text{otherwise} \end{cases} \quad (5.10)$$

where $a_2 > 0$ is a scaling factor. y is the directed distance of the pseudo-labeled image $\mathbf{x}_p$ to the SVM boundary (the decision function output of SVM for the pseudo-labeled image $\mathbf{x}_p$). We will explain the rationale of the fuzzy expression in (5.10) by first considering that the pseudo-label of the selected image has been determined as positive in subsection 5.3.3. In this case, the upper equation in (5.10) will be used. If y has a large positive value, this will suggest that it is most likely to be a relevant image. Since there is a strong agreement between the predicted pseudo-label from subsection 5.3.3 and the predicted class label using the trained SVM, its fuzzy membership value should be set to a large value close to unity. If y has a large negative value, this will suggest that it is most likely to be an irrelevant image. Since there is a strong disagreement between the predicted pseudo-label from subsection 5.3.3 and the predicted class label using the trained SVM, its fuzzy membership value should be set to a small value close to zero. The same arguments apply when the pseudo-label of the selected image has been determined to be negative in subsection 5.3.3.

Finally, we combine these two measures affecting the fuzzy membership to obtain the final soft relevance estimate, namely:

$$g(\mathbf{x}_p) = w_1(\mathbf{x}_p)w_2(\mathbf{x}_p) \quad (5.11)$$

The estimated soft relevance of the pseudo-labeled images is then used in FSVM training.

5.4 Experimental Results

The proposed method is applied in our retrieval system. Subtractive clustering is utilized to determine the cluster centers of the relevant and irrelevant images. It uses the following parameters: r_a is set to 0.075 and 0.25 for relevant and irrelevant samples, respectively, with $r_b = 1.2r_a$, $\varepsilon_A = 0.5$, and $\varepsilon_R = 0.2$, where r_a , r_b , ε_A , and ε_R have been defined in Chapter 4.

RBF kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$ is used for SVM, where $\sigma = 3$ and the

regularization parameter $C = 100$. Similar parameter selection scheme as explained in Section 4.4 is used in the experiment. For each query, an initial set of l_0 images is shown to the user for labeling. An SVM is trained based on the l_0 labeled images. l_0 is a small number that is set according to the user information need. FSVM-based active learning is then launched by selecting l images to the user to label. l is set to be 20 in the experiment.

In our experiment, the performance of the proposed PLFSVM method is evaluated and compared with active learning using SVM [15]. The objective measure, which is based on the Corel's predefined ground truth, is used. That is, the retrieved images are considered to be relevant if they belong to the same category as the query image. 100 queries with one from each category are selected for evaluation. Retrieval performance is evaluated by ranking the

database images according to their directed distances to the SVM boundary after each active learning iteration. Five iterations of feedbacks are recorded.

The average precision-versus-recall (APR) graphs after the first iteration of active learning are shown in Figure 5.1. We have shown the results for two different numbers of initially labeled images. From the figures, we observe that the PLFSVM method outperforms the standard SVM method in both cases. The PLFSVM method provides higher recall rate at the same precision level, and higher precision rate at the same recall level. For instance, the PLFSVM method offers a precision rate of 55% compared to 38% offered by the SVM method at 10% recall rate for $l_0=10$. This reflects the effectiveness of the proposed PLFSVM method. In our experiments, it is observed that PLFSVM consistently achieves better performance than SVM for different values of l_0 .

We have also adopted the performance measure of retrieval accuracy to evaluate the retrieval system. The retrieval accuracy performance of the PLFSVM method is given in Figure 5.2 for the case of $l_0=10$. It is observed that for both the top 10 and 20 returned images, the retrieval accuracy of the PLFSVM method increases quickly in the first few iterations. This is desirable since the user can obtain satisfactory results within the first few iterations. It is worth emphasizing that the initial retrieval performance is very important since users often expect quick results and are unwilling to provide much feedback. Hence, reducing the amount of user feedback while providing good retrieval results is of great interests for many CBIR systems. It is observed that our method offers an improvement of 16% (10 returned images) and 14% (20 returned images) over the SVM method after the first iteration of active learning. The superiority of our method over the SVM method mainly lies in the incorporation of pseudo-labeled images for effective learning.

(a)

(b)

Figure 5.1 The average precision-versus-recall graphs (after the first iteration of active learning). (a) APR for 5 initial labeled images, $l_0 = 5$, (b) APR for 10 initial labeled images, $l_0 = 10$.

(a)

(b)

Figure 5.2 Retrieval accuracy of the PLFSVM and SVM methods. (a) Retrieval accuracy in top 10 results, (b) Retrieval accuracy in top 20 results.

5.5 Summary

This chapter presents a new PLFSVM framework that integrates pseudo-labeled images into FSVM for CBIR. The technique addresses the small sample problem by incorporating pseudo-labeled images to enlarge the training data set. An unsupervised clustering algorithm is used to select pseudo-labeled images by studying the characteristics of the labeled images. The relevance of the pseudo-labeled images is estimated using the fuzzy membership function, and integrated into the FSVM for effective learning. Experimental results demonstrate that the PLFSVM method can achieve improved performance, while reducing the user workload.

Chapter 6

Region-based Image Retrieval Using Variable-Length Radial Basis Function Network

6.1 Introduction

Region-based image retrieval (RBIR) systems have been developed to represent images at the object level to achieve higher retrieval accuracy [5], [23]–[27]. They utilize image segmentation to decompose images into regions, and then perform comparison of image similarity based on the local features extracted from the constituent regions.

User interpretation of visual content tends to vary with respect to different users and circumstances. In other words, the user perception of image similarity is subjective in nature. Each image, depending on the context, can have many different interpretations. For instance, given the same query image of “a model standing in front a car”, one user may be more interested in finding the images of models, while, another user may look for the images of cars. Even the same user may perceive the same image differently under various situations. However, some existing RBIR systems adopt fixed image similarity metric [5], [23]–[27]. They ignore the contextual dependency of visual content, therefore, failing to capture the user perceptual subjectivity. To address this issue, relevance feedback has been introduced to

overcome some of the limitations associated with RBIR systems [28]–[38]. Relevance feedback is traditionally used in CBIR systems that use global features. There are still relatively few works that incorporate it in the RBIR systems. Nevertheless, some preliminary works have been done to combine RBIR with relevance feedback to improve the retrieval performance.

Despite the previous works on relevance feedback for RBIR systems, it is still a challenging task to develop effective and efficient interactive mechanisms to yield satisfactory retrieval performance. Some issues that need to be resolved include: (i) the learning algorithm for region-based representation to progressively improve the retrieval accuracy through interaction with the users, and (ii) the tradeoff between retrieval effectiveness and realistic real-time requirement.

RBFN has been employed as an effective relevance feedback classifier in global image-based CBIR systems [110], [111]. However, to the best of our knowledge, very limited work has been done to extend it into the RBIR framework. This is because the conventional RBFNs are incapable of handling variable-length feature representation. Images with variable-length representation (VLR) in RBIR systems are described by sets of weighted features that reflect the significance of the constituent regions in the images. The size of the set may vary depending on the number of regions in different images.

To address the above-mentioned issues, we propose a new variable-length radial basis function network (VLRBFN) in this chapter to model and learn the user perception of image similarity in RBIR systems. The input data and RBF units of the VLRBFN have VLR. In contrast to most existing feedback approaches in RBIR systems that are concerned with global characterization of image similarity such as QPM [29], [30] and reweighting methods [31]–[33], the proposed VLRBFN utilizes local function approximation of image similarity to characterize the local context defined by the query and feedback images. Under this

framework, some important issues involved in the development of the VLRBFN have to be addressed, which include: (i) center selection of the VLRBFN, (ii) weight determination of the regions in the images, and (iii) training of the VLRBFN. To address these issues, the following methods have been proposed: (i) an unsupervised subtractive clustering algorithm for VLRBFN center selection, (ii) a region weight learning strategy that offers a hybrid of perceptual determination and density estimation, and (iii) optimization of the underlying network parameters (weight and width) of the VLRBFN using gradient-descent training based on feedback data. The trained VLRBFN is then used in subsequent sessions to retrieve the images.

The organization for the rest of the chapter is outlined as follows. Section 6.2 introduces the image segmentation algorithm, the region-based image representation, and the image similarity measure. In Section 6.3, we discuss the structure of the VLRBFN and the clustering algorithm used in network construction. We further explore the new region weight learning algorithm that determines the relevance of the regions in each cluster center. The learning algorithm of the network is then explained in detail. In Section 6.4, experimental results using the proposed method are discussed and compared with other techniques. Finally, concluding remarks are given in Section 6.5.

6.2 Overview of the Proposed Region-based Image Retrieval System

6.2.1 Overview of the System

The main processing of the VLRBFN system involves the offline and online stages. Offline processing includes image segmentation, feature extraction and representation. Online processing is the interaction between the user and the system. The complete online

processing steps are described as follows.

- (i) User submits his/her initial query to the system through query-by-example. The system performs the K-NN search using the EMD for similarity matching. The top K most similar images are displayed to the user for feedback.
- (ii) The user provides his/her feedback on the display images.
- (iii) Subtractive clustering is performed on the respective relevant and irrelevant feedback samples to determine the relevant and irrelevant cluster centers, which are used to construct the hidden layer of the current update network.
- (iv) The region weights of the obtained cluster centers in step (iii) are updated using the proposed region weight learning strategy.
- (v) The constructed update network is combined with the previously trained VLRBFN and trained using all feedback samples accumulated over the previous iterations. Only the parameters of the update network are updated.
- (vi) Retrieve new images from database based on the trained VLRBFN.
- (vii) Stop if the user is satisfied with the retrieval results; otherwise, go to step (ii).

6.2.2 Image Segmentation and Region-based Image Representation

We use mean shift algorithm as our image segmentation method to partition images into homogenous regions [112]. Mean shift algorithm is a technique for mode seeking in non-parametric distributions. We have explored various image segmentation algorithms and found that the mean shift algorithm produces good segmentation results in our experiment. Therefore, it is adopted in our system.

After image segmentation, for each region in the image, the color and texture descriptors of the region are extracted and used to represent the regions. Color moments are used as the color feature by extracting the first two moments (mean and standard deviation) from each

channel of LUV color space [56], whereas wavelet moments are used as the texture feature [62]. They are formed by applying three-level wavelet decomposition on the image, and extracting the mean and standard deviation of the transform coefficients. A more detailed description of these two features has been given in Chapter 3.

Further, we use region weights to reflect the importance or relevance of the regions. Studies have shown that humans generally focus their attentions on a few prominent areas in an image [113]–[115]. Several factors have been found to influence human visual attention including area [113], location [116], contrast [113]–[115], and shape [114], among others. In our experiment, we explore and find that area and location are among the most effective factors in measuring the perceptual importance of a region. Therefore, these two factors are adopted in our retrieval system to determine the initial region importance. Importance due to the region's area is calculated as the area percentage of the region with respect to the whole image. This is in agreement with the viewpoint that important regions in an image tend to occupy large areas. Importance due to location of a region is taken as inversely proportional to the distance between the center of the region and the center of the image. This is based on the assumption that important regions are usually located near the center of the image. During the initialization process, the weights of area and location factors are summed up to provide the initial region weights. These region weights are then normalized such that the summation of all the region weights within an image is equal to one. During the feedback process, a perceptual determination and density estimation-based region weight learning strategy is designed to adapt the feedback region weights (relevance). The detailed algorithm will be described in subsection 6.3.3. An illustration of feature extraction in the VLRBFN system is provided in Figure 6.1.

Figure 6.1 Illustration of feature extraction in the VLRBFN system.

6.2.3 Image Similarity Measure

We employ the Earth Mover's Distance (EMD) to compare two images with their different constituent regions [76]. The initial retrieved images are based on K-NN search using the EMD. The EMD computes the distance between two distributions, which are represented by signatures. The signatures are sets of weighted features that capture the distributions. The size of the two signatures can be different. That is, the EMD can operate on variable-length representations of the distributions, which is suitable for region-based image similarity comparison. Intuitively, the EMD measures the least amount of work needed to transform one image distribution into the other, which can be solved as a linear optimization problem. The notion of “work” is based on the user-defined ground distance which is the distance between two features. The EMD works as follows.

Let $X_p = \{(\mathbf{C}_{pi}, W_{pi})\}_{i=1}^m$ be an image with index p that consists of m regions $\{R_{pi}\}_{i=1}^m$ where $\mathbf{C}_{pi}$ and W_{pi} represent the feature vector and weight of the region R_{pi} . Another image $X_q = \{(\mathbf{C}_{qj}, W_{qj})\}_{j=1}^n$ with index q consists of n regions $\{R_{qj}\}_{j=1}^n$ where $\mathbf{C}_{qj}$ and W_{qj} represent the feature vector and weight of the region R_{qj} . Our objective is to find a set of optimal flows f_{ij} between $\mathbf{C}_{pi}$ and $\mathbf{C}_{qj}$ that minimize the overall cost $\sum_{i=1}^m \sum_{j=1}^n f_{ij} d(\mathbf{C}_{pi}, \mathbf{C}_{qj})$, subject to the following constraints:

$$f_{ij} \geq 0, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n \quad (6.1)$$

$$\sum_{j=1}^n f_{ij} \leq W_{pi}, \quad 1 \leq i \leq m \quad (6.2)$$

$$\sum_{i=1}^m f_{ij} \leq W_{qj}, \quad 1 \leq j \leq n \quad (6.3)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min\left(\sum_{i=1}^m W_{pi}, \sum_{j=1}^n W_{qj}\right) \quad (6.4)$$

Once we have obtained the optimal flows, the EMD between the images X_p and X_q can be defined as:

$$\text{EMD}(X_p, X_q) = \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij} d(\mathbf{C}_{pi}, \mathbf{C}_{qj})}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \quad (6.5)$$

where $d(\mathbf{C}_{pi}, \mathbf{C}_{qj})$ is the ground distance metric between $\mathbf{C}_{pi}$ and $\mathbf{C}_{qj}$. In general, the ground distance can be any meaningful distance measure. Here it is chosen to be the Euclidean distance. Since the summation of the region weights within each image is normalized to one,

that is, $\sum_{i=1}^m W_{pi} = \sum_{j=1}^n W_{qj} = 1$, therefore, the EMD between X_p and X_q is given as:

$$\text{EMD}(X_p, X_q) = \sum_{i=1}^m \sum_{j=1}^n f_{ij} d(\mathbf{C}_{pi}, \mathbf{C}_{qj}) \quad (6.6)$$

EMD is an image-to-image similarity measure that combines information from all the image regions. Each region in the first image will be compared with every region in the second image. Therefore, as opposed to the individual region-to-region similarity measures where a precise segmentation is essential, the EMD is relatively robust against imperfect segmentation (over-segmentation or under-segmentation).

6.3 Progressive Variable-Length Radial Basis Function Network (VLRBFN)

6.3.1 Structure of Progressive VLRBFN

In interactive RBIR systems, some works such as the QPM [29], [30] and reweighting feedback approaches [31]–[33] are concerned with global characterization of image similarity using a single model. For instance, the QPM method tries to find the single optimal query,

which is associated with a particular location in the feature space. Such a query model yields a global evaluation for similarity learning. However, the image relevance is most appropriately characterized by multiple models, each of which has its own local characteristics. Therefore, global modeling of image similarity may not adapt adequately to the local context defined by the query and feedback images. To fully exploit the local characteristics of image relevance, it is more desirable to adopt a multi-cluster local function modeling strategy to achieve better generalization. In addition, the learning speed of retrieval systems should be fast since the users expect quick interaction. This motivates us to adopt the radial basis function network (RBFN) as a good choice in the context of RBIR due to its following properties: fast learning speed, simple network structure, piece-wise local function approximation, and global generalization power [44], [97]–[99].

Conventional RBFNs use fixed-length kernel which are incapable of handling variable-length feature representation. In view of this, a new variable-length radial basis function network (VLRBFN) with progressive learning capability is proposed. The VLRBFN is based on the characteristics of variable-length region-based image representation in RBIR systems. It progressively learns the user perception of visual content through continual feedbacks. The schematic diagram of the network is given in Figure 6.2. It has a recursive structure (Figure 6.2(a)) in consideration of fast interaction required by the retrieval systems. Since the VLRBFN during the previous iterations has been trained to learn the user information need, retraining the whole network for each feedback iteration is unnecessary and time consuming. Therefore, a small update network (Figure 6.2(b)) is introduced to track the current user information need based on newly retrieved images. This enables a much smaller update network to be developed and trained batch-wise, leading to significant reduction in training time. Only the parameters of the update network need to be estimated in each feedback iteration.

Figure 6.2 Schematic diagram of VLRBFN. (a) Recursive structure of VLRBFN, (b) Update network.

The input data to the network is an image with index i $X_i = \{(\mathbf{C}_k, \mathbf{W}_k)\}_{k=1}^{m_i}$ that consists of m_i regions $\{R_k\}_{k=1}^{m_i}$ with $\mathbf{C}_k$ and $\mathbf{W}_k$ representing the feature vector and weight of the region R_k . To simplify representation of notations, we drop the subscript i from X_i which then becomes $X = \{(\mathbf{C}_k, \mathbf{W}_k)\}_{k=1}^m$ as shown in Figure 6.2. The t -th iteration VLRBFN is built from the $(t-1)$ -th iteration VLRBFN progressively, and combined with the t -th iteration update network. The output of the t -th iteration VLRBFN $F_t(X)$ for an input image X can be expressed as:

$$F_t(X) = \begin{cases} F_{t-1}(X) + U_t(X), & t = 1, 2, \dots, T \\ 0, & t = 0 \end{cases} \quad (6.7)$$

where $F_{t-1}(X)$ is the $(t-1)$ -th iteration VLRBFN output, $U_t(X)$ is the update network output, and T is the total number of feedback iterations. The update network consists of an input layer, a hidden layer and an output layer. The input layer receives images of region-based representation, with one input-layer unit for each region. As a result, the number of input-layer units m varies with the number of regions in the input image X . The input layer is connected to the hidden layer which is constructed from the feedback images. Each RBF unit of the hidden layer has a region-based image representation. The number of RBF units c_t may vary depending on the feedback images at iterations t . For simplification of representation, we drop the subscript t and utilize the symbol c to denote the number of RBF units for $U_t(X)$. It is worth noting that all input-layer units of the input image X interact with each hidden unit that is connected to them. The output layer has a single unit whose output value $U_t(X)$ is the weighted combination of all the responses from each RBF unit. $\{w'_i\}_{i=1}^c$ is the set of output connection weights at the t -th iteration. The method is computationally efficient since only the parameters of the update network need to be estimated. It is also

algorithmically effective as all feedback samples accumulated over the previous iterations are used to train the update network. The detailed network structure and parameters updating algorithm will be described in subsection 6.3.4.

It is noted that the proposed VLRBFN differs from the conventional RBFN in several ways. The VLRBFN is developed for variable-length region-based image representation in RBIR systems, while the conventional RBFN can only handle uniform-length feature representation. Further, traditional RBFNs use a fixed-length kernel in the computation of image similarity, which cannot be applied in VLR. Therefore, a kernel function centered on VLR has been introduced. In particular, a distance metric called the EMD is employed as the distance measure in the kernel function. Under this framework, some important issues involved in the development of the VLRBFN have to be addressed: (i) center selection of the VLRBFN, (ii) weight determination of the regions in the images, and (iii) training of the VLRBFN. We will outline how this study addresses these issues in the following few sections.

6.3.2 Modified Subtractive Clustering for VLRBFN Center Selection

The VLRBFN structure determination involves selecting the appropriate number of RBF units and their initial values. In image retrieval systems, the relevant samples tend to form clusters, while the irrelevant samples often scatter around the classes. An important aspect of image retrieval systems is that they usually require fast interaction. Hence reducing the number of RBF units while maintaining a good representation of image similarity will cut down the training time of the network. In view of this, we develop the utilization of a proper clustering process to determine the VLRBFN centers so that a compact structure of the VLRBFN can be achieved. The feedback samples are clustered according to their types: relevant and irrelevant. This will then be used to construct the hidden layer of the VLRBFN. We have employed subtractive clustering in Chapter 4 as it is fast, efficient and does not require the number of

clusters to be specified *a priori*. Considering the variable-length, multi-region image representations, and many-to-many region mapping between any two images, we propose a modified subtractive clustering algorithm, which calculates the distance between two samples based on the EMD.

6.3.3 Perceptual Determination and Density Estimation for Region Weight Learning

After subtractive clustering, we obtain a set of c cluster centers at the t -th feedback iteration $\{V_i^t\}_{i=1}^c$, where $V_i^t = \{(C_{ik}, W_{ik}) \mid_{k=1}^{m_i}\}$ has a region-based representation, and m_i is the number of regions in it. These estimated centers can belong to either the relevant or irrelevant class. Since different regions in an image have unequal importance (relevance) for computing image similarity, the relevance of the constituent regions in each cluster center has to be learned when feedback information is available. The feedback relevant images belong to the same semantic class, while the irrelevant images usually belong to multiple classes. Therefore, we only employ relevant images for region weight learning since irrelevant images are not representative enough of the negative class and can be misleading in the estimation of region weights. Under this framework, only region weights in the relevant cluster centers need to be learned.

A typical image consists of segmented regions. Generally there exist one or two most salient regions (user's regions of interest) in each image that characterize its primary semantic information. Therefore, the regions should be assigned the largest importance. Often these salient regions can be detected reliably based on the human perceptual model as mentioned in subsection 6.2.2. As for other less salient regions, it is also important to have reliable weight estimation. This is because all the regions have contributions towards the image-to-image similarity (EMD measure) in order to reduce the impact of imperfect segmentation. Further,

considering the fast-interaction requirement, a more systematic and efficient approach to estimate the region weights is required. When we have a set of relevant images sharing the same semantic meaning, the most salient regions from all the relevant images can be combined to represent more accurately the primary semantic information that the user is most interested. Our objective, therefore, is to first determine the most salient region in each relevant image respectively such that these regions jointly capture the semantic class of a user's query at the primary level. We then utilize the set of selected regions from all the relevant images for weight estimation of other unseen regions. In this work, the perceptual importance of the regions can be utilized together with the feedback information in determining the salient regions. In doing so, the salient region selection is based on a set of relevant images instead of one single image, hence making it more robust. Since these salient regions are from a semantic class, we can estimate their density distribution. Then the importance of a test region is evaluated by determining how it differs from the estimated distribution. Because a precise estimate of the underlying region distribution is not required, we make use of one-class SVM (OCSVM) [117], which does not impose any assumption on the data distribution. This then serves as a basis for systematic region weight estimation.

We have employed area and location as two important factors in measuring region importance as described in subsection 6.2.2. In this section, we adopt the same two factors to compute the raw region perceptual saliency for consistency. The raw perceptual saliency of each region within an image refers to an unnormalized saliency as opposed to the normalized region weight. It is used to perform comparison among regions in different images. We first sort regions in each image according to their raw region perceptual saliency. Here, our basic assumption is that the most salient region in each image is usually among the top r regions ranked by their saliency in the image. Since r is much smaller than the total number of regions in the image, the salient region selection among the top r regions instead of all the

regions in each relevant image will cut down the computational time while maintaining a reliable estimation. Let $D = \{X_i\}_{i=1}^n$ be the set of n relevant images, $S = \emptyset$ be an empty set. Our objective is to find the most salient region for each image in D . First, we choose the region with the largest saliency value among the top r salient regions of each image in D as a starting point. Once a salient region is selected, it will be added into S , and the corresponding image that contains it will be removed from D . Next, we select a region from the top r salient regions in the rest of the images in D . The region that is the most similar to the centroid of the selected regions in S is chosen. The similarity measure is based on the Euclidean distance between them. The smallest distance indicates the largest similarity. By comparing all the selected regions so far in S with the unselected regions in D , one salient region from a relevant image in D is chosen at each step. This iterative process terminates until D is empty.

After the perceptual strategy to determine the set of salient regions $S = \{x_i\}_{i=1}^n$, we estimate their density distribution using OCSVM. The basic idea of OCSVM is to learn a function that fits the training data from only one class into a small region. Our strategy used to learn the function is to first map the input data into a high-dimensional feature space through a mapping function, and then separate the data from the origin by a maximal margin hyperplane. Given the training data $S = \{x_i\}_{i=1}^n$, OCSVM finds the optimal hyperplane by solving the following optimization problem [117]:

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu n} \sum_{i=1}^n \xi_i - \rho \\ & \text{subject to } \mathbf{w} \cdot \phi(\mathbf{x}_i) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n \end{aligned} \quad (6.8)$$

where $\mathbf{w}$ is the normal vector of the hyperplane, n is the number of training data, ξ_i is the slack variable and ρ is the offset of the decision function. $\nu \in (0,1)$ is the regularization

parameter. It controls the tradeoff between maximizing the distance from the origin and containing most of the training samples on the side of the hyperplane that does not contain the origin. The optimization problem can be transformed into the following equivalent dual problem:

$$\begin{aligned} & \text{minimize } \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & \text{subject to } 0 \leq \alpha_i \leq \frac{1}{vn}, \sum_{i=1}^n \alpha_i = 1, \quad i = 1, \dots, n \end{aligned} \quad (6.9)$$

where α_i is the Lagrange multiplier. $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function in the input space that computes the inner product of two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in the feature space, that is, $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$. Using the kernel trick, we can construct the optimal hyperplane in the feature space without having to know the mapping φ . The decision function of the OCSVM can be represented as:

$$f(\mathbf{x}) = \mathbf{w} \cdot \varphi(\mathbf{x}) - \rho = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho \quad (6.10)$$

The decision function of OCSVM returns a positive value if a test data belongs to the region that contains most of the training data, otherwise it returns a negative value. Each region in the relevant cluster center is given a score by the decision function of OCSVM. Since the training data are the salient regions that the user have most interest in. Test regions with higher scores are more likely to dwell in the neighborhood of regions of interest, and therefore should be assigned larger weights, while regions with lower scores will be assigned smaller weights. These scores correspond to the degrees of region importance and can be taken as the weights of the associated regions. They are then normalized such that the region weights fall into the range of $[0,1]$ and the summation of region weights within a relevant cluster center is

equal to one.

6.3.4 VLRBFN Training Algorithm

After obtaining the initial VLRBFN center estimates, we are ready to discuss the learning process of the VLRBFN. In this study, relevance feedback is implemented as an online supervised learning process by adjusting the underlying parameters of the network. The error function E_t at the t -th iteration is defined as:

$$E_t = \frac{1}{2} \sum_{j=1}^{N_T} e_{jt}^2 = \frac{1}{2} \sum_{j=1}^{N_T} (Y_t(X_j) - F_t(X_j))^2 \quad (6.11)$$

where N_T is the total number of training samples, and e_{jt} is the error signal for the j -th training sample X_j at the t -th iteration. $F_t(X_j)$ and $Y_t(X_j)$ represent the actual and desired network output for X_j , respectively. $Y_t(X_j)$ is set to 1 and 0 for X_j associated with the positive and negative feedback, respectively.

Traditional RBFNs use fixed-length kernel in the computation of image similarity, which cannot be applied in VLR. Therefore, we utilize the EMD kernel proposed in [29], [30] for the RBF units in the context of RBIR, which is given by:

$$f(X_j, V_i^t, \sigma_i^t) = \exp\left(-\frac{\text{EMD}^2(X_j, V_i^t)}{2(\sigma_i^t)^2}\right), \quad t=1, 2, \dots, T, \quad i=1, 2, \dots, c \quad (6.12)$$

where t is the iteration number, and T is the total number of feedback iterations. V_i^t and σ_i^t are the center and width of the i -th RBF unit at the t -th iteration. The values V_i^t are initialized based on the outputs obtained using the subtractive clustering and region weight learning approaches. Based on the kernel function, the update function $U_t(X_j)$ is given as:

$$U_i(\mathbf{X}_j) = \sum_{t=1}^c w_i^t f(\mathbf{X}_j, \mathbf{V}_i^t, \sigma_i^t) \quad (6.13)$$

where w_i^t is the output connection weight. Substituting (6.13) into (6.7), the VLRBFN output function $F_t(\mathbf{X}_j)$ can be expressed as:

$$F_t(\mathbf{X}_j) = \begin{cases} F_{t-1}(\mathbf{X}_j) + \sum_{i=1}^c w_i^t f(\mathbf{X}_j, \mathbf{V}_i^t, \sigma_i^t), & t = 1, 2, \dots, T \\ 0, & t = 0 \end{cases} \quad (6.14)$$

The system training of the RBIR system at each feedback iteration is performed by minimizing the cost function E_t with respect to the network parameters of $U_i(\mathbf{X})$, namely, $\boldsymbol{\theta} = \{w_i^t, \sigma_i^t \mid i = 1, 2, \dots, c\}$. The learning process can be represented as:

$$\boldsymbol{\theta} = \arg \min_{\boldsymbol{\theta} \in \Theta} (E_t) = \arg \min_{\boldsymbol{\theta} \in \Theta} \left(\frac{1}{2} \sum_{j=1}^{N_T} (Y_t(\mathbf{X}_j) - F_t(\mathbf{X}_j))^2 \right) \quad (6.15)$$

where Θ is the solution space of the parametric vector $\boldsymbol{\theta}$.

The training is achieved using gradient-descent algorithm based on all the feedback samples accumulated over all previous iterations. The update equations for w_i^t and σ_i^t are summarized as follows:

- (i) Weight estimation at the l -th learning iteration:

$$\frac{\partial E_t(l)}{\partial w_i^t(l)} = - \sum_{j=1}^{N_T} e_{jt}(l) f(\mathbf{X}_j, \mathbf{V}_i^t(l), \sigma_i^t(l)) \quad (6.16)$$

$$w_i^t(l+1) = w_i^t(l) - \eta_1 \frac{\partial E_t(l)}{\partial w_i^t(l)}, \quad i = 1, 2, \dots, c \quad (6.17)$$

- (ii) Width estimation at the l -th learning iteration:

$$\frac{\partial E_t(l)}{\partial \sigma'_i(l)} = -w'_i(l) \sum_{j=1}^{N_r} e_{ji}(l) f(X_j, V'_i(l), \sigma'_i(l)) \frac{\text{EMD}^2(X_j, V'_i(l))}{(\sigma'_i(l))^3} \quad (6.18)$$

$$\sigma'_i(l+1) = \sigma'_i(l) - \eta_2 \frac{\partial E_t(l)}{\partial \sigma'_i(l)}, \quad i = 1, 2, \dots, c \quad (6.19)$$

(iii) Repeat steps (i)-(ii) until convergence or a maximum number of iterations is reached.

The term $e_{ji}(l)$ is the error signal of the j -th training sample X_j at the l -th learning iteration, η_1 and η_2 are learning parameters for w'_i and σ'_i , respectively. It is noted that the gradient-descent algorithm given in (6.16)-(6.19) does not involve back-propagation of error signal. Thus, it requires less training time to converge when compared with other neural networks such as multilayer perceptron.

6.4 Experimental Results

The performance of the proposed framework is evaluated on the same image database as shown in Figure 4.8 in Chapter 4. Some selected examples of image segmentation results using the mean shift algorithm are given in Figure 6.3. The proposed method is applied to perform region-based image retrieval. Modified subtractive clustering is utilized to determine the cluster centers of the relevant and irrelevant images. The hybrid of perceptual determination and density estimation strategy is used for region weight learning with $r = 3$ in our experiment. RBF kernel with width $\sigma = 0.2$ is used for OCSVM and the regularization term $\nu = 0.001$. The VLRBFN is constructed progressively and trained iteratively using the learning algorithms in (6.16)-(6.19). The learning step-sizes are taken as $\eta_1 = 0.01$ and $\eta_2 = 0.00001$. Similar parameter selection scheme as explained in Section 4.4 is used.

Figure 6.3 Examples of segmentation results by the mean shift algorithm. Original images are on the left, the corresponding segmented images are on the right. (a) Tiger, (b) Lion, (c) Woman, (d) Cat.

6.4.1 Performance Comparison based on Precision versus Recall Measure

In our experiment, the performance of the proposed VLRBFN method for RBIR is evaluated and compared with the approach using conventional RBFN for global image-based CBIR (GRBFN) [111] (The method described in Chapter 4). To render a fair comparison, the same features of color moments and wavelet moments are computed from the whole images as the global features in the GRBFN method. Further, the proposed scheme is also compared with another two feedback approaches called QPM for RBIR (RQPM) and SVM for RBIR (RSVM) [29]. QPM is a widely used relevance feedback method in CBIR and has recently been extended into RBIR. The RSVM method adopts the powerful learning machine SVM and has achieved promising retrieval results. Experimental results indicate that our method achieves much better retrieval performance than both the GRBFN and RQPM methods, and

comparable retrieval performance to that of the RSVM method. It is worth noting that our method is more computationally efficient than the RSVM method. This is because the proposed recursive network structure in our method reduces the training time as only a small update network and its associated parameters need to be estimated. Further, the proposed region weight determination scheme is efficient. In contrast, the RSVM method involves determination of a new hyperlane at every feedback iteration, which is computationally intensive. In addition, its region weight learning process is very time-consuming. Taking into account the fast-response requirement, computational efficiency is also important for an interactive retrieval system. Therefore, our method is more desirable for an interactive retrieval system than the RSVM method.

We employ the objective measure, which is based on the Corel's predefined ground truth. In our experiment, 100 queries with one from each category, as given in Figure 4.8 in Chapter 4, are selected for evaluation. The precision and recall rates are averaged over all the queries. The average precision-versus-recall (APR) graphs for top 25 retrieved images after 1 and 5 feedback iterations are shown in Figure 6.4. From the figures, we observe that the VLRBFN method outperforms both the GRBFN and RQPM methods, while maintaining comparable performance to that of the RSVM method. The VLRBFN method provides higher recall rate at the same precision level, and higher precision rate at the same recall level than the GRBFN and RQPM methods. For instance, the VLRBFN method offers precision rates of 48% (1 feedback iteration) and 88% (5 feedback iterations) at 20% recall rate. In comparison, at the same recall rate, the GRBFN method offers precision rates of 30% (1 feedback iteration) and 54% (5 feedback iterations), and the RQPM method offers precision rates of 30% (1 feedback iteration) and 31% (5 feedback iterations). This reflects the effectiveness of the proposed method. Its advantage in terms of computational efficiency over the RSVM method will be illustrated later.

(a)

(b)

Figure 6.4 The APR graphs in top 25 returned images. (a) APR after 1 feedback iteration, (b) APR after 5 feedback iterations.

6.4.2 Performance Comparison based on Retrieval Accuracy Measure

We have also adopted the performance measure of retrieval accuracy to evaluate the retrieval system. The retrieval accuracy comparison is given in Figure 6.5. The retrieval accuracy is averaged over 100 test queries, and six iterations of feedbacks are recorded. It is observed that the retrieval accuracy of the VLRBFN method increases quickly in the first few iterations. This is desirable since the users can obtain satisfactory results within a few iterations. Further, it reaches steady-state retrieval accuracy of 90% in about 5 feedback iterations, which is very close to that of the RSVM method (91%). In comparison, the GRBFN and RQPM methods achieve lower steady-state retrieval accuracy. The VLRBFN method offers improvements of 27% over the GRBFN method, and improvements of 45% over the RQPM method for steady-state retrieval accuracy. It is clear that the VLRBFN method consistently provides superior results when compared with the GRBFN and RQPM methods. The superiority of our method over the RQPM method mainly lies in the local modeling of image similarity and effective learning to estimate the underlying network parameters. With integration of region-based representation and effective interactive mechanism, our method can outperform the GRBFN method that employs conventional whole image-based retrieval scheme.

Figure 6.5 Retrieval accuracy in top 25 returned images.

6.4.3 Computational Efficiency Analysis

The experiments are conducted on a PC with the following specifications: Pentium4 2.4GHz processor, 512M RAM, Windows XP, Matlab 6.5. In our experiments, the average retrieval time for each feedback iteration using the VLRBFN method ranges from 10 to 15 seconds for the top 25 results. This is around seven times faster than the RSVM method. The improvement in computational efficiency of our method is more significant as the number of iterations increases.

The overall computational load of the proposed VLRBFN method and the RSVM method during each feedback iteration lies mainly on two stages: (i) classifier training, and (ii) distance metric calculation of all images in the database during similarity matching. Since both methods adopt the same distance metric (EMD), the second stage of similarity matching can be ignored for computational load comparison. For the first stage of classifier training in the VLRBFN method (cluster formation, region weight determination, and network parameter estimation), it is noted that the computation of network parameter estimation dominates the training process. The proposed recursive structure greatly reduces the training time as only a small update network $U_i(\mathbf{X})$ and its associated parameters need to be estimated. The computational load due to cluster formation and region weight determination is small compared to network parameter estimation. On the other hand, the training of the RSVM method (region weight determination and SVM parameter estimation) involves determination of a new hyperplane at every feedback iteration. This process involves significant computational time as the quadratic programming problem of the full SVM needs to be solved at each iteration. Further, the RSVM method utilizes a region weight learning scheme centered on region frequency and inverse image frequency. It assumes that important regions

should appear more often in the positive images than in other images in the database. This involves similarity comparison between each region in the positive images and regions in all other images in the database, hence making it very time-consuming especially for large databases. As a result, our method offers better computational performance than the RSVM method.

6.5 Summary

This chapter presents an efficient VLRBFN-based relevance feedback approach in interactive RBIR systems. The basic idea is to introduce local modeling of image similarity to address the local information contained in the users' feedbacks. During the feedback iterations, a VLRBFN is constructed and progressively trained to achieve improved retrieval results. A new kernel function of the VLRBFN centered on region-based representation is introduced for image similarity. An unsupervised clustering algorithm is developed for VLRBFN center selection based on the characteristics of region-based representation. The importance of the constituent regions in each RBF unit is estimated using a hybrid of perceptual determination and density estimation scheme. A gradient descent-based learning strategy is then employed to estimate the underlying network parameters. Experimental results demonstrate that the proposed method is effective in performing region-based image retrieval.

Chapter 7

Peer Tagging and Knowledge Propagation

7.1 Introduction

CBIR systems have been developed to address the shortcomings encountered in traditional text-based image retrieval systems which use keywords to annotate images. Nevertheless, CBIR systems experience the challenge of semantic gap between the low-level visual features and the high-level semantic concepts. On the other hand, keywords have more direct correspondence with high-level semantics than low-level features. Therefore, it would be advantageous to unify both keywords and visual features into CBIR systems. In doing so, meaningful and efficient retrieval can be achieved by benefiting from the strengths of both content- and keyword-based image indexing and retrieval algorithms.

Recently, a new trend in image annotation has been to use a new paradigm of peer tagging where distributed users (particularly Internet users) add tags (keywords) to images. The main idea is that individual users can apply arbitrary tags to images that they have uploaded for sharing, or participate in voluntary tagging of images that are of interests to them. The notion of peer tagging has transformed the conventional intensive, single-user annotation task into voluntary, simple, and multiple-user process. The power of peer tagging based on free and voluntary manpower can be demonstrated in Wikipedia, a free online encyclopedia. Created in January 2001, Wikipedia has attracted some ten thousand voluntary contributors from all over the world to add or edit information in it. Online media (image and video) tagging

systems have emerged recently and become increasingly popular such as Flickr.com [39], Smugmug.com [40], and YouTube.com [41]. For instance, YouTube.com has an average of 50,000 new video uploads a day, coupled with tags for indexing and retrieval. It also managed to attract more than 20 million visitors in May 2006 alone [118]. Notwithstanding their attractive features, these systems do not explore the correlation between the tags and media content, hence underutilizing the information available. Further, some images or videos in the systems may not properly annotated or even annotated at all.

In this chapter, we propose to incorporate the notion of peer tagging into CBIR systems. To the best of our knowledge, very little research work has been done on peer tagging in the context of CBIR. Online peer tagging can help to improve the effectiveness and efficiency of image annotation, hence facilitating image retrieval. An important potential application of the proposed idea is Internet image search engine. Current Internet image search engines such as Google and Yahoo image search rely on keywords alone [119], [120]. They utilize the clues such as image filename and text surrounding images to index the images, which may be inaccurate, or at times irrelevant. In contrast, the new paradigm of incorporating peer tagging into CBIR systems and developing effective fusion algorithms to combine tags and visual features will lead to new perspective in Internet image search engines.

Due to the broad scope of online peer tagging, and the need for significant user participation to perform comprehensive tagging, we focus on knowledge propagation of keywords in this study. This is the process of automatically propagating keywords from a subset of annotated (labeled) images to the rest of the unannotated (unlabeled) images in the database. One issue that needs to be addressed adequately in the current peer tagging systems is that only a fraction of the images out of the complete collection is annotated. Knowledge propagation thus can be utilized to resolve this problem. Knowledge propagation is beginning to gain popularity in CBIR systems in recent years. It reduces the enormous burden involved

in full manual annotation of large image database, thereby facilitating image retrieval. The proposed knowledge propagation scheme is based on image content analysis and training of keyword classifiers. First, all images in the database are segmented into regions. Image content descriptors (color and texture) are extracted from each region and represented using feature vectors. Region weight is further used to reflect the importance of each region. A subset of images out of the complete database is manually annotated with keywords, which are used as the training set. To perform weight estimation of unseen regions, we first apply genetic algorithm (GA) to select salient regions from the training images. Once the salient regions are determined, OCSVM is employed to estimate their density distribution. The importance of a region in the training images is then evaluated by determining how it differs from the estimated distribution. After estimating the region weights, an RBF-based classifier is trained for each keyword based on the content (region feature vectors and their weights) of the training images. Once these keyword classifiers are fully trained, they are used to propagate keywords with associated soft relevance values to other unannotated images in the collection.

The organization for the rest of the chapter is outlined as follows. Section 7.2 studies the unique issues in peer tagging including HCI for image tagging, tag generation and formation, and tag clustering. In Section 7.3, we present the developed knowledge propagation scheme in detail. In Section 7.4, experimental results using the proposed knowledge propagation method are discussed and compared with other techniques. Finally, concluding remarks are given in Section 7.5.

7.2 Peer Tagging

7.2.1 Human-Computer Interface (HCI) for Image Tagging

The mechanism of HCI for tagging can have great impact on the performance of image tagging. It involves an important issue of tagging granularity. Granularity specifies whether tagging is based on segmented regions in the images or on the global images. Since an image is worth a thousand words, different users tend to have different interests in the rich content embedded in an image. Therefore, it would be useful to provide individual regions of interests to the users for tagging. However, tagging with high granularity requires precise image segmentation, which is very difficult to achieve in practice. For instance, inaccurate segmentation may result in an object being partitioned into several regions, with none of them being representative of the object. Presenting these non-meaningful regions to the users will trouble them, while tagging with low granularity could potentially avoid such problems. Further, high granularity tagging implies tagging many discrete regions of the images, which requires more user participation than tagging the whole images.

Nevertheless, it should be noted that image segmentation may be more accurate and precise in some domain-specific applications as opposed to general-purpose image segmentation. Under this circumstance, high granularity tagging may be more useful than low granularity tagging. Therefore, the utilization of the level of tagging granularity can vary with respect to different applications. A flexible way to achieve a compromise between the two is by offering the users both levels of tagging granularity. Users can choose to tag segmented regions or the whole images according to their preferences. This will provide a user-friendly environment for image tagging by adapting the tagging interface to suit different user needs and expectations.

7.2.2 Tag Generation and Formation

Peer tagging offers a low-cost alternative by allowing the users to add any chosen tags to online images. Nevertheless, the issue of tag reliability may arise because users choose the tags without any restrictions. For example, spelling mistakes may happen in inserting tags. The users may also use inaccurate, irrelevant, or esoteric tags. The results are unreliable tags that do not support effective image retrieval. As a consequence, it will incur extra efforts to identify the most appropriate tags for images. This is in contrast to conventional manual annotation where some expert annotators control the tags about images. Therefore, the issue of tag reliability needs to be considered when generating tags. One possible solution to this problem is a compromise between the two by letting the users collaborate with expert annotators. In doing so, unreliable tags (incorrectly spelled tags, inaccurate, irrelevant or esoteric tags) can be greatly reduced with the supervision and guidance of the experts. The results will be rich and high-quality tags generated by combining the benefits of both.

Other means of generating reliable tags can be used. One possible way is to utilize suggestive tagging that suggests a list of possible tags to the users for the images to be tagged. The suggested tags may be based on tags assigned to the same images by other users. For instance, the popular tags from other users can be employed as suggested tags for the same images. The main idea is that popular tags are used by large number of users and will indicate some form of agreement and coherence amongst the users. Therefore, they are very likely to be used by a new user for the same images. An example of such visualization paradigm is implemented in Flickr.com called tag cloud as shown in Figure 7.1. The popularity of a tag is reflected by its displayed size in the tag cloud. Tags with larger font indicate more popularity, while the displayed order is generally alphabetical. In addition, the suggested tags can also be generated from related tags of other images by exploring the correlation between the existing tags and associated content of the images. Further, suggested tags can be presented to a user

on the basis of the user’s previous tagging behavior. For instance, we can observe how many tags a user have added, analyze the motivations when adding tags, the reasons for choosing particular tags and the decision-making process behind tag selection.

More research is needed to provide a more user-friendly environment so that the users are willing to participate in the tagging process, enter more reliable and useful tags, and are attracted to return to the tagging process again.

All time most popular tags

Figure 7.1 The tag cloud on Flickr.com.

7.2.3 Tag Clustering

Peer tagging provides users the opportunity to assign tags to images according to their preferences. On the other hand, because of human perceptual subjectivity, different users tend to assign slightly different tags to the same image. This will result in a large number of diverse tags that does not support effective search and efficient collection management. Often, these tags are related and do not stand alone, which can be grouped into clusters for better

utility. Tag clustering, therefore, is employed to identify groups of tags sharing similar semantic concepts. The resultant clusters can be used as a thesaurus to facilitate searching. An example of tag clustering can be shown in Flickr.com as illustrated in Figure 7.2. We can see that given the tag “Singapore”, Flickr.com returns several related semantic categories with finer distinctions from each other but associated with the concept of “Singapore”. Both the representative images and the tags that comprise each cluster are displayed to the users. Users can further explore the clusters that are of interest to them by clicking on the displayed tags of the respective clusters and browse through the images shown. Such visual display of clusters through tag clustering offers an opportunity to explore the image collection, hence improving the searchability of the system.

Figure 7.2 Illustrative example of tag clusters in Flickr.com.

Tag clustering finds groups of tags according to some measure of similarity. An

automated tag clustering algorithm has been proposed by first providing a similarity measure among tags and then running clustering techniques on the tag space [121]. It is based on counting the number of co-occurrences of any pair of tags and a cut-off point is selected to determine when the co-occurrence count is significant enough to be used. Often, tags could be grouped into clusters based on the relationships between them. The goal is, therefore, to discover the tag relations by mining the tag space. The spelling and word morphology could be used for determining the tag correlations. For example, similar spelling of “Chinese new year” and “chinesenewyear” indicates a strong relation. In addition, two tags that use the singular and plural of a word such as “cat” and “cats”, “party” and “parties”, the paradigm of a word such as “tag”, “tagging”, and “tagged”, are also strongly related. Further, tags that are of synonyms (different word but same meaning) could also be considered similar. Examples of synonyms are the tags “film” and “movie”, “sick” and “ill”, “fast” and “quick”, and “baby” and “infant”, among others. The process of tag clustering can be fully automated by the system, or involve users’ participation by allowing them to specify the relationships between tags. This user specified knowledge could then be utilized to group tags.

7.3 Knowledge Propagation

7.3.1 Overview of the Proposed Knowledge Propagation Scheme

The proposed knowledge propagation scheme is based on image content analysis and training of keyword classifiers. A schematic overview of the method is given in Figure 7.3.

Figure 7.3 Schematic overview of the knowledge propagation scheme.

A preprocessing step is performed by segmenting images into regions. Each region is represented by a feature vector and a corresponding weight that reflects its importance. The procedure starts with annotating a subset of images from the database with one keyword assigned for each image. This keyword-labeling scheme is simple since it does not involve manual annotation of the segmented regions with multiple keywords. The entire annotated images are used as training set. The remaining unannotated images in the database are test set. For each keyword K_l , $l = 1, 2, \dots, L$, images that are labeled with it are considered as positive images $D_{K_l} = \{\mathbf{X}_i\}_{i=1}^n$ while other images in the training set are negative images $D_{\bar{K}_l} = \{\mathbf{X}_j\}_{j=1}^m$.

Since different regions in an image have unequal importance (relevance) for computing image similarity, the relevance of the constituent regions in each image has to be learned. Similar to the region weight estimation scheme described in subsection 6.3.3 of Chapter 6, our objective is to first determine the most salient region in each image contained in D_{K_l} such that these regions jointly capture the semantic class K_l at the primary level. We then utilize the set of selected salient regions from all the images in D_{K_l} for weight estimation of other unseen regions. It is noted that we have adopted human perceptual model to detect the salient regions. From an optimization point of view, finding the salient regions in each image in D_{K_l} can be formulated as a combinatorial optimization problem. The objective function to be optimized is based on a similarity measure between the regions. In this work, GA is employed to find the salient regions in D_{K_l} . It is chosen due to the following reasons: (i) knowledge propagation is an offline procedure which does not have the real-time requirement, (ii) GA is known to be an effective method to solve combinatorial optimization problem, (iii) GA is easy to implement, and (iv) GA has the characteristics of convergence towards global optima. Instead of a sequence of single potential solution, GA operates on a population of potential solutions to solve the optimization problem. The implicit parallel exploration of the

search space reduces the possibility of being stuck in local optima [122]–[124].

Since the selected salient regions are from one semantic class, we can estimate their density distribution. The importance of a region in the training images is then evaluated by determining how it differs from the estimated distribution. We employ OCSVM to estimate their density distribution. Each unseen region is given a score by the decision function of OCSVM. Regions with higher scores are more likely to dwell in the neighborhood of regions of interest, and therefore should be assigned larger weights, while regions with lower scores will be assigned smaller weights. These scores correspond to the degrees of region importance and can be taken as the weights of the associated regions. After estimating the region weights of the training images, for each keyword K_l , a binary RBF classifier C_{K_l} is trained based on the visual features (regions and corresponding weights) of the positive images in D_{K_l} and negative images in $D_{\bar{K}_l}$. The RBF classifier used here is based on the EMD kernel as introduced in Chapter 6. In total, an ensemble of L binary classifiers $\{C_{K_l}\}_{l=1}^L$ is trained for the L keywords. The trained classifiers are then used to propagate keywords to the test (unannotated) images. Each test image is classified by the L trained classifiers respectively and assigned a set of soft labels according to the decision output of the classifiers. The soft labels with associated keyword membership values reflect the likelihood that a keyword matches the image. As OCSVM and RBF with EMD kernel have been described in Chapter 6, we present the procedure of salient region selection using GA in the following subsection.

7.3.2 GA for Salient Region Selection

GA is a widely used evolutionary algorithm inspired by the procedure of evolution in nature [122]–[124]. It is usually applied for solving optimization problems. Instead of a sequence of

single potential solution, GA operates on a population of potential solutions to the optimization problem. It genetically breeds the population of individuals that are represented by chromosomes. Each individual denotes a potential solution to the problem and is assigned a fitness value that measures the quality of the corresponding solution. GA starts with an initial population of randomly generated individuals. Fitness of each individual in the population is then calculated using a fitness function. Fitter individuals from the current population are selected for reproduction using a selection method. Next, the selected individuals go through the procedure of crossover (recombination) and mutation to produce new population. The genetic operators of selection, crossover, and mutation are iteratively applied to the population until the termination criterion is satisfied. Each iteration of this process is called a generation. A pseudo-code outline of the simple genetic algorithm (SGA) is shown in Table 7.1 to illustrate the basic components of GA [122]. The population of candidate solutions at iteration t is represented by $P(t)$.

begin	
	$t = 0;$
	initialize $P(t)$;
	evaluate $P(t)$;
	while not done do
	$t = t + 1;$
	select $P(t)$ from $P(t-1)$;
	reproduce pairs in $P(t)$;
	evaluate $P(t)$;
	end while
end	

Table 7.1 Procedure of SGA.

The details of the GA used to solve our problem are now given. First, for each keyword K_i , GA generates a random population of chromosomes. A chromosome consists of encoding of the parameter set, with each gene of the chromosome encoding a variable in the parameter set. Often, chromosomes are represented by binary strings of 0s and 1s in classical GA. In

our work, an integer-value representation of the chromosomes is utilized. A region $\mathbf{x}_{ij}$ in image X_i is represented by its index $j \in \{1, \dots, m_i\}$ which is an integer between 1 and m_i , where m_i is the total number of regions in X_i . The region index refers to the unique label that is used to identify a region in an image. Therefore, integer-value representation provides a natural and convenient way of expressing the mapping from the chromosome representation to the problem domain. Each chromosome thus contains the indexes of the selected regions, with each gene of a chromosome being the index of a region from each image. GA then operates on the encoding of the variables (region indexes) rather than the variables themselves (feature vectors of the regions). A chromosome S generated in this case can be illustrated as $S=[1 \ 3 \ 2 \ 4 \ 6]$, which denotes region 1, region 3, region 2, region 4 and region 6 from their respective images. The length of a chromosome is n which is the size of D_{K_t} .

Next, the fitness value of each chromosome in the population is evaluated by a fitness (objective) function, which is the function to be optimized. Higher fitness values correspond to fitter chromosomes. This fitness evaluation process is performed after decoding the chromosomes into the problem domain (candidate regions). Intuitively, salient regions from the same semantic class are both visually similar and perceptually important. Therefore, the fitness function is designed using both visual and perceptual similarity measures. Visual similarity among the set of candidate regions can be measured using the sum of the Euclidean distances from each region to the centroid of all the candidate regions. This is a reflection of the compactness of the regions. Smaller distances indicate larger visual similarity among them. While perceptual importance can be evaluated using human perceptual model. Based on this argument, an exponentially-based fitness function is selected and defined as follows:

$$f(S_{K_t}) = \exp \left(- \sum_{i=1, j \in \{1, \dots, m_i\}, \mathbf{x}_{ij} \in X_i}^n \frac{\|\mathbf{x}_{ij} - \boldsymbol{\mu}_{K_t}\|}{w_{ij}} \right) \quad (7.1)$$

where S_{K_i} is a chromosome in the population for keyword K_i . μ_{K_i} is the centroid of the regions $\{\mathbf{x}_{ij} \mid i=1, \dots, n, j \in \{1, \dots, m_i\}\}$ corresponding to chromosome S_{K_i} . w_{ij} is the raw region perceptual saliency of $\mathbf{x}_{ij}$ as defined in subsection 6.3.3 of Chapter 6, which denotes the perceptual importance of region $\mathbf{x}_{ij}$. $\|\mathbf{x}_{ij} - \mu_{K_i}\|$ is the Euclidean distance between region $\mathbf{x}_{ij}$ and the centroid μ_{K_i} . The distance is further weighted by the respective region perceptual saliency w_{ij} . If the regions are visually similar ($\|\mathbf{x}_{ij} - \mu_{K_i}\|$ is small) and perceptually important (w_{ij} is large), $\frac{\|\mathbf{x}_{ij} - \mu_{K_i}\|}{w_{ij}}$ will be small. Consequently, the corresponding fitness value $f(S_{K_i})$ will be large due to the exponentially decaying function.

Once each chromosome has been assigned a fitness value, a selection operation is performed to choose fitter chromosomes from the population according to their fitness. Chromosomes with higher fitness values are more likely to be selected to reproduce. There are several selection methods in GA such as fitness-proportionate selection, ranking selection, and tournament selection. In this work, we employ the fitness-proportionate selection scheme called stochastic universal sampling (SUS) [125]. In SUS, an interval *sum* is determined as either the sum of the individuals' selection probabilities or the sum of the individuals' raw fitness values. Individuals are then mapped to contiguous segments of a line in the range $[0, \text{sum}]$. The size of each individual's segment corresponds to its fitness value. Considering N number of parents to be selected, N equally spaced pointers are placed over the line. The distance between the pointers is sum/N . The position of the first pointer is given by a randomly generated number in the range $[0, \text{sum}/N]$. Individuals whose segments span the positions of the pointers will be selected. An example of SUS selection method is illustrated in Figure 7.4 given a population of 10 individuals. Their selection probabilities and fitness

values are shown in Table 7.2. Individual 1 is the fittest individual and consequently occupies the largest interval, whereas individual 10 is the least fit one, hence having the smallest interval on the line (Figure 7.4). In this example, *sum* equals to 1 which is the sum of the individual's selection probability. The selection probability of an individual is computed as its raw fitness value normalized by the total fitness values of the population. If 6 parents are to be selected, the distance between the pointers is $1/6 = 0.167$. Suppose a random number generated in the range $[0, 0.167]$ is 0.1, which is the position of the first pointer, SUS will select the individuals 1, 2, 3, 5, 6, 9 for mating.

Individuals	1	2	3	4	5	6	7	8	9	10
Fitness value	2.2	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4
Selection probability	0.17	0.15	0.14	0.12	0.11	0.09	0.08	0.06	0.05	0.03

Table 7.2 Selection probabilities and fitness values of 10 individuals.

Figure 7.4 Illustration of SUS.

The selected chromosomes then go through the process of crossover and mutation to create new chromosomes in the next generation. Crossover involves information exchange between two chromosomes. It combines substructures of two parent chromosomes to form new child chromosomes with certain probability. Some commonly used crossover methods include single-point, two-point and uniform crossover. Single-point crossover is utilized in this work. In single-point crossover, one crossover point is picked within the chromosomes of both parents. At that point, the parent chromosomes are split into sub-chromosomes which

are swapped to generate new offspring. An example of single-point crossover is illustrated in Figure 7.5. Given two parent chromosomes, a crossover point at position 3 is selected along the length of the parent chromosomes. By exchanging all genes on the right side of this crossover point between the two parent chromosomes, two new offspring chromosomes are produced. Mutation is a genetic operator that alters one or more genes in a chromosome with probability to preserve generic diversity within the population. Figure 7.6 shows the mutation process. In this example, the fifth gene of a chromosome is mutated to create a new chromosome. By introducing variations into the population of chromosomes, mutation can help to avoid local minima. Crossover is the dominant operation in GA and is usually applied with a high probability. Mutation in contrast only serves as a background operation and occurs with a low probability. This iterative process proceeds through subsequent generations until the maximum number of generations has been reached. The result is an optimal solution to the problem, that is, a set of salient regions.

Figure 7.5 Single-point crossover operator.

Figure 7.6 Mutation operator.

7.4 Experimental Results

The performance of the proposed knowledge propagation scheme is evaluated on the same image database as shown in Figure 4.8 in Chapter 4. The following preprocessing procedure is applied, which is the same as described in Chapter 6. All images are segmented into regions using mean shift algorithm. Each region is characterized by both color and texture features and is assigned an initial region weight.

The image database contains 100 different semantic categories, which are predefined by the Corel Photo Gallery based on their semantic concepts. Images in the same category share the same semantic meaning. We predefine 100 keywords with each keyword describing the semantics of an image category, which are listed in Table 7.3. The predefined keywords could reduce the diversity of the annotations from different annotators and are utilized to establish the ground-truth annotations. The ground-truth annotation for an image in the database is the keyword describing the category that contains the image. This serves as the basis for annotation performance evaluation. A small percentage of images are sampled from each semantic category, which are used as the training set. Since there are a total of 100 keywords characterizing all images in the database, therefore, each of the training images is manually annotated with one of the 100 keywords. The remaining unannotated images in the database are the test set.

lion	canvas	frost	Pyramid	texture
tiger	car	fruit	rodeo	cobble
horse	card	holiday	Rome	tissue
mountain	cat	herb	rose	tool
plant	college	kungfu	royal-guard	train
cloud	crystal	lighthouse	rural	tribe
mushroom	cuisine	mask	surf	tulip
cave	cyber-art	lizard	sculpture	vegetable
jack-o-lantern	decoy	mineral	seed	warplane
ancestor	design	rock	seashell	waterfall
antique	dog	molecule	skiing	leopard
autumn	drink	mosaic	space	eagle
balloon	Easter-egg	dish	sport	goat
architecture	fashion	doll	stamp	nest
bobsled	firearm	old-post	steam-engine	whale
bonsai	firework	old-works	subsea	woman
kitchen	fitness	penguin	sunset	ship
bus	flag	owl	wave	youth
butterfly	foliage	plant-art	seaside	polo
binary-image	door	leaf	textile	epidermis

Table 7.3 Keyword list of the 100 image categories.

For each keyword K_l , $l=1,2,\dots,100$ as listed in Table 7.3, the integer-valued GA is employed to select salient regions from the positive training images. The population size is 50 chromosomes. Crossover and mutation are applied probabilistically with a crossover probability of 0.7 and a mutation probability of 0.08. The maximum number of generations is 400. OCSVM is applied for density estimation of the salient regions selected by GA. An RBF classifier is then trained using the positive and negative images for keyword K_l . In total, 100 RBF classifiers are trained for the 100 keywords in the experiment. Similar parameter selection scheme as explained in Section 4.4 is used. The parameter settings for OCSVM and the RBF classifiers are the same as those adopted in Chapter 6. Keywords are then automatically propagated to the unannotated images using the 100 trained RBF classifiers. Each keyword assigned to an image is associated with a weight (soft keyword membership value) based on the decision output of the RBF classifier for that particular keyword. As a result, each image is automatically assigned a set of 100 keywords along with weights

reflecting the significance of the corresponding keywords in describing the image semantics. Lastly, for each image, we perform a ranking on the keywords assigned to it according to the keyword weights. Only the top 10 keywords become the final annotations of the images, which we call auto-annotations of the images.

In our experiment, the performance of the proposed knowledge propagation method is evaluated and compared with the model-based method called cross media relevance model (CMRM) [85], and classifier-based method using SVM [89]. To assess the annotation performance, we adopt the measure called annotation accuracy defined as follows:

$$\text{Annotation accuracy} = \frac{\text{total number of test images correctly annotated}}{\text{total number of test images}} \quad (7.2)$$

In the strict sense, a test image is considered to be correctly annotated if and only if the keyword having the largest weight within its auto-annotation matches its ground-truth annotation (strict annotation accuracy (SAA)). In a relaxed sense, an image is regarded as correctly annotated so long as there exists a keyword in its auto-annotation that coincides with its ground-truth annotation (relaxed annotation accuracy (RAA)).

We use both 10% and 20% of the images from the database for training. The remaining 90% and 80% images are used for testing, respectively. The results of SAA and RAA on the testing set are shown in Table 7.4. It is observed that our method consistently outperforms the CMRM method, while maintaining comparable performance to that of the SVM method. For instance, using 10% of images from the database as training set, our method offers SAA of 34% and RAA of 72% on the test data, which is very close to that of the SVM method (SAA of 35% and RAA of 72%). In comparison, the CMRM method achieves much lower annotation accuracy (SAA of 10% and RAA of 28%). Our method offers improvements of 24% (SAA) and 44% (RAA) over the CMRM method. This reflects the effectiveness of the proposed method. Further, the annotation accuracy of our method increases to 44% (SAA)

and 79% (RAA) for 20% of images from the database as training set. This is slightly better than the SVM method which achieves SAA of 41% and RAA of 78%. While the annotation accuracy of the CMRM method has small improvement (SAA of 10% and RAA of 30%). It can also be observed that our method maintains higher annotation accuracy (SAA of 34% and RAA of 72%) even with just 10% of the database images as the training set than that of the CRMR method (SAA of 10% and RAA of 30%) using 20% of the database images as the training set. This is desirable since we can obtain higher annotation accuracy with less number of manually annotated images.

Method	10% of images from the database as training set		20% of images from the database as training set	
	SAA	RAA	SAA	RAA
CMRM	10%	28%	10%	30%
SVM	35%	72%	41%	78%
Proposed method	34%	72%	44%	79%

Table 7.4 Results of average annotation accuracy.

7.5 Summary

This chapter introduces the new concept of peer tagging and explores the related issues including HCI for image tagging, tag generation and formation, and tag clustering. The goal is to incorporate peer tagging into CBIR systems to address the concern of the semantic gap in CBIR systems. We further present a knowledge propagation scheme based on image content analysis and training of keyword classifiers. Experimental results show that the proposed method is effective for image annotation.

Chapter 8

Conclusion and Recommendations

8.1 Conclusion

In many CBIR systems, a critical problem that has not been adequately addressed is the semantic gap between the low-level visual features and high-level semantic concepts. In order to address this intrinsic difficulty, this thesis has investigated several issues in CBIR systems using computational intelligence techniques. This investigation covers three main categories: relevance feedback in CBIR (Chapters 4 and 5), relevance feedback in RBIR (Chapter 6), and peer tagging and knowledge propagation (Chapter 7).

8.1.1 Relevance Feedback in CBIR

An important issue that needs to be considered in the formulation of relevance feedback is the potential imprecision of user perception. Conventional binary labeling scheme in relevance feedback requires a hard-decision to be made on the relevance of each retrieved image. This is inflexible as user interpretation varies with respect to different information needs and perceptual subjectivity. In addition, users tend to learn from the retrieval results to further refine their information priority. It is, therefore, inadequate to describe the users' fuzzy perception of image similarity with crisp logic. In view of this, we propose a fuzzy relevance feedback concept which integrates the fuzzy interpretation into the notion of relevance

feedback in Chapter 4. As opposed to making hard-decisions, it allows soft-decisions to be made with respect to the retrieval results in the form of fuzzy feedbacks. Thus the proposed scheme reconciles the dilemma of binary or multi-level labeling by employing soft-decision instead of hard-decisions on the retrieval results. The relevance of the fuzzy images is evaluated using an *a posteriori* probability estimator. Different images under the fuzzy label are mapped into suitable relevance scores. With the users' feedback, a corresponding PFRBFN is developed to learn the different degrees of relevance embedded in the user interpretation of image similarity.

Another issue associated with relevance feedback in CBIR systems is the small sample problem where only a limited number of labeled images are available for learning. Learning from insufficient training images often constrains the retrieval performance. To address this problem, we propose a new algorithm based on the concept of pseudo-labeling in Chapter 5. It incorporates carefully selected unlabeled images to enlarge the training data set and assigns proper pseudo-labels to them through a label propagation process. This is in contrast to most existing relevance feedback approaches in CBIR systems that are concerned with the use of labeled data only. Further, some fuzzy rules are utilized to automatically estimate class membership of the pseudo-labeled images. FSVM is designed to take into account the fuzzy nature of some training images during its training. In order to exploit the advantages of pseudo-labeling, active learning, and the structure of FSVM, we develop a unified PLFSVM framework to perform content-based image retrieval.

8.1.2 Relevance Feedback in RBIR

For interactive RBIR systems, it is still a challenging task to develop effective and efficient interactive mechanisms to yield satisfactory retrieval performance. As mentioned in Chapter

1, some issues that need to be resolved include: (i) the learning algorithm for region-based representation to progressively improve the retrieval accuracy through interaction with the users, and (ii) the tradeoff between retrieval effectiveness and realistic real-time requirement. To address these issues, we have proposed a new VLRBFN framework that integrates relevance feedback into RBIR systems in Chapter 6. It can be considered as a natural extension of the proposed approach in Chapter 4. However, the unique characteristics of RBIR systems need to be considered. RBIR systems employ variable-length feature representation as opposed to uniform-length feature representation in conventional global-based image retrieval systems. On the other hand, conventional RBF networks use fixed-length kernel which are incapable of handling variable-length feature representation.

The VLRBFN adopts a new kernel function that is instrumental in handling variable-length computation of image similarity. Further, an unsupervised subtractive clustering algorithm is developed for VLRBFN center selection. In order to assess and estimate the relevance of each region in the image, we propose a region weight learning strategy that offers a hybrid of probabilistic estimation and perceptual determination. During the feedback training process, the underlying network parameters (weight and width) of the VLRBFN are optimized using gradient-descent optimization. The trained VLRBFN is then used in the subsequent sessions to retrieve images. A main advantage of the method is its computational efficiency. This is because the proposed recursive network structure in our method reduces the training time as only a small update network and its associated parameters need to be estimated.

8.1.3 Peer Tagging and Knowledge Propagation

A recent trend in image annotation has emerged over the past 2-3 years. This involves the

new paradigm of peer tagging which enables Internet users to annotate images online in a voluntary, simple and effort-sharing basis. In Chapter 7, we propose to incorporate the notion of peer tagging into CBIR systems and develop effective content- and keyword-based fusion algorithm to address the issue of semantic gap. The main idea is to integrate the strengths of content- and keyword-based image indexing and retrieval algorithms while alleviating their respective difficulties. Unique issues related to peer tagging have been studied including HCI for image tagging, tag generation and formation, and tag clustering. This will serve as a foundation for future research on knowledge discovery through content and keyword fusion in CBIR systems.

We further develop a knowledge propagation scheme based on image content analysis and training of keyword classifiers. First all images in the database are segmented into regions. Image content descriptors (color and texture) are extracted from each region and represented using feature vectors. A region weight is further used to reflect the importance of each region. A subset of images out of the complete database is manually annotated with keywords, which are used as the training set. To perform weight estimation of the unseen regions, we first apply GA to select salient regions from the training images. Once the salient regions are determined, OCSVM is employed to estimate their density distribution. The importance of a region in the training images is then evaluated by determining how it differs from the estimated distribution. After region weight estimation, an RBF-based classifier is trained for each keyword based on the content (region feature vectors and their weights) of the training images. Once these keyword classifiers are fully trained, they are used to propagate keywords with associated soft relevance values to the unannotated images in the collection.

8.2 Recommendations for Further Research

CBIR is an exciting research area with many promising future research directions. To extend and continue the research carried in this thesis, a list of possible directions for future research is given below.

8.2.1 Content-Based Video Retrieval (CBVR)

Video is a key element in multimedia applications such as interactive TV, remote video surveillance, and video conferencing, etc. There is a need to efficiently index, store, and retrieve the visual information from video database [126], [127]. Currently, we concentrate on image retrieval systems and algorithms. The frameworks can be extended into content-based video retrieval (CBVR) systems in the future. CBVR can be developed by first segmenting the video stream into a sequence of temporally homogeneous shots. Each shot is then characterized by one or a few key frames extracted from the shot. CBIR techniques can then be modified to exploit these key frames for video retrieval.

Video content analysis and representation is the basis for constructing an effective CBVR system. Since video has both spatial and temporal information, it is important to develop analysis techniques to exploit the spatial and temporal attributes to support effective video retrieval.

Similar to image retrieval, there exists a semantic gap between the low-level audio-visual features and the high-level human perception in CBVR systems. Relevance feedback can be incorporated to narrow the gap. Significant advances on relevance feedback for CBIR systems have been made, while limited progress has been achieved for CBVR systems. Relevance feedback in CBVR is a more challenging problem since there is a temporal

duration associated with the user feedback. Further, much more computing power is needed for processing videos than for images.

8.2.2 Knowledge Discovery through Content and Keyword Fusion

CBIR systems experience the challenge of semantic gap between the low-level features and high-level human perception. Generally, keywords are the best descriptors for image semantics. Therefore, it is desirable to integrate the advantages of content- and keyword-based image indexing and retrieval algorithms while alleviating their respective difficulties. There are relatively few works in this direction, even though it is of practical importance. Some recent works have been done on merging keywords and visual content in CBIR systems [128]–[130].

By extending the research work as described in Chapter 7, a new framework of knowledge discovery through content and keyword fusion can be developed in the future. More specifically, it includes: (i) to fully exploit the unique issues related to peer tagging such as the optimal setting in HCI for image tagging, tag reliability, generation, clustering and verification, and incorporate the paradigm of peer tagging into CBIR or CBVR systems, (ii) to introduce effective fusion scheme to unify keywords and visual content, and (iii) to develop new relevance feedback techniques on joint keywords and visual feature space.

8.2.3 CBIR in Domain-Specific Applications

The current state-of-the-art in CBIR for general-purpose images holds promise to be useful for domain-specific applications such as medical images, art galleries, museum archives, satellite and aerial image database, and surveillance related applications, among others.

Nevertheless, CBIR systems are usually developed for general-purpose images. They are not readily applicable to the domain-specific applications. Consequently, domain knowledge needs to be considered when extending CBIR algorithms to a particular application domain.

Domain-specific image retrieval is getting increasingly important. This is true especially in medical field [131], [132]. In recent years, large volume of medical images including ultrasound, computer tomography (CT), magnetic resonance imaging (MRI) and X-ray images are produced in increasing quantities. Therefore, effective retrieval techniques for medical images are needed to retrieve such visual information. An important application of medical image retrieval is in diagnostics. For instance, a clinician could retrieve similar cases of patients from the medical archive before making a diagnostic decision, hence facilitating clinical diagnosis and therapy.

Medical CBIR is a challenging research area. Medical images differ from general-purpose images in many ways. Many powerful visual features for representing general-purpose images may not be effective discriminators for medical images. For instance, color is often of less importance for medical images than for general-purpose images since most medical images do not contain colors. Often, medical images emphasize more on local features rather than global features. Majority of medical images captures human anatomy in a three dimensional structure. Therefore, image registration becomes necessary before volumetric image comparisons and image feature extraction. Further, the semantics of medical images are usually objective and well defined while the semantics of general-purpose images are diverse and may vary according to different human perception. In conclusion, the use of domain constraints and prior knowledge is important in domain-specific applications.

Author's Publications

Book Chapter

- [1] Kui Wu and Kim-Hui Yap, "A soft-labeling framework for content-based image retrieval," accepted for publication in *Computational Intelligence in Multimedia Processing: Recent Advances*, Springer-Verlag.
- [2] Kui Wu and Kim-Hui Yap, "A perceptual subjectivity notion in interactive content-based image retrieval systems," *Intelligent Multimedia Processing with Soft Computing*, Springer-Verlag, 2005.

Journal

- [1] Kui Wu and Kim-Hui Yap, "Content-based image retrieval using fuzzy perceptual feedback," *Multimedia Tools and Applications*, vol. 43, pp. 235-251, Mar. 2007.
- [2] Kui Wu and Kim-Hui Yap, "Fuzzy SVM for content-based image retrieval - A pseudo-label support vector machine framework," *IEEE Computational Intelligence Magazine*, vol. 1, pp. 10–16, May 2006.
- [3] Kim-Hui Yap and Kui Wu, "A soft relevance framework in content-based image retrieval systems," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1557–1568, Dec. 2005.
- [4] Kui Wu and Kim-Hui Yap, "Region-based image retrieval using variable-length radial basis function network," revised and resubmitted to *IEEE Trans. Neural Networks*.

Conference

- [1] Kui Wu and Kim-Hui Yap, "Knowledge propagation in content-based image retrieval," accepted for publication in *Int. Conf. Information, Communications and Signal Processing*, Singapore, 2007.
- [2] Kim-Hui Yap and Kui Wu, "A pseudo-labeling framework for content-based image retrieval," in *IEEE Symposium on Computational Intelligence in Image and Signal Processing*, Hawaii, USA, 2007.
- [3] Kui Wu and Kim-Hui Yap, "Region-based image retrieval using radial basis function network," in *Proc. IEEE Int. Conf. Multimedia & Expo*, Toronto, Canada, 2006.
- [4] Kui Wu and Kim-Hui Yap, "Fuzzy relevance feedback in content-based image retrieval systems using radial basis function network," in *Proc. IEEE Int. Conf. Multimedia & Expo*, Amsterdam, The Netherlands, 2005.
- [5] Kui Wu and Kim-Hui Yap, "An efficient radial basis function network approach for content-based image retrieval," in *Proc. IEEE Int. Conf. Acoustic, Speech, and Signal Processing*, Montreal, Canada, vol. 3, pp. 437–440, 2004.
- [6] Kim-Hui Yap and Kui Wu, "Fuzzy relevance feedback in content-based image retrieval," in *Proc. Int. Conf. Information, Communications and Signal Processing and Pacific-Rim Conference On Multimedia*, Singapore, vol. 3, pp. 1595–1599, 2003.

Bibliography

- [1] M. Flickher, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, no. 9, pp. 23–32, Sept. 1995.
- [2] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in *Proc. IEEE Int. Conf. Image Processing*, Washington D.C., USA, pp. 815–818, 1997.
- [3] G. Amarnath and J. Ramesh, "Visual information retrieval," *Communications of ACM*, vol. 40, no. 5, pp. 70–79, May 1997.
- [4] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233–254, June 1996.
- [5] J. R. Smith and S. F. Chang, "VisualSEEk: a fully automated content-based image query system," in *Proc. ACM Multimedia*, pp. 87–98, Nov. 1996.
- [6] T. Gevers and A. W. M. Smeulders, "PicToSeek: Combining color and shape invariant features for image retrieval," *IEEE Trans. Image Processing*, vol. 9, pp. 102–119, 2000.
- [7] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papatomas, and P. N. Yianilos, "The Bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments," *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 20–37, 2000.

- [8] A. M. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no.12, pp. 1349–1380, Dec 2000.
- [9] J. Huang, S. R. Kumar, and M. Metra, "Combining supervised learning with color correlograms for content-based image retrieval," in *Proc. ACM Multimedia*, pp. 325–334, Nov. 1997.
- [10] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. Circuits and Video Technology*, vol. 8, no. 5, pp. 644–655, 1998.
- [11] Y. Rui and T. S. Huang, "Optimizing learning in image retrieval," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 236–243, June 2000.
- [12] H. Müller, W. Müller, S. Marchand-Maillet, and D. McG. Squire, "Strategies for positive and negative relevance feedback in image retrieval," in *Proc. IEEE Int. Conf. Pattern Recognition*, Barcelona, Spain, pp. 1043–1046, 2000.
- [13] N. Vasconcelos and A. Lippman, "Learning from user feedback in image retrieval systems," in *Proc. Neural Information Processing Systems*, Denver, Colorado, pp. 977–986, 1999.
- [14] Z. Su, H. J. Zhang, S. Li, and S. P. Ma, "Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning," *IEEE Trans. Image Processing*, vol. 12, pp. 924–937, Aug. 2003.
- [15] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. ACM Int. Conf. Multimedia*, Ottawa Canada, pp. 107–118, 2001.
- [16] Y. Q. Chen, X. S. Zhou, and T. S. Huang, "One-class SVM for learning in image retrieval," in *Proc. IEEE Int. Conf. Image Processing*, Greece, pp. 815–818, Oct. 2001.

- [17] G. D. Guo, A. K. Jain, W. Y. Ma, and H. J. Zhang, "Learning similarity measure for natural image retrieval with relevance feedback," *IEEE Trans. Neural Networks*, vol. 13, no. 4, pp. 811–820, July 2002.
- [18] H. K. Lee and S. I. Yoo, "A neural network-based image retrieval using nonlinear combination of heterogeneous features," *International Journal of Computational Intelligence and Applications*, vol. 1, no. 2, pp. 137–149, 2001.
- [19] J. Laaksonen, M. Koskela, and E. Oja, "PicSom-self-organizing image retrieval with MPEG-7 content descriptions," *IEEE Trans. Neural Network*, vol. 13, no. 4, pp. 841–853, July 2002.
- [20] P. Muneesawang and L. Guan, "Automatic machine interactions for content-based image retrieval using a self-organizing tree map architecture," *IEEE Trans. Neural Networks*, vol. 13, no. 4, pp. 821–834, July 2002.
- [21] Y. Wu, Q. Tian, and T. S. Huang, "Discriminant-EM algorithm with application to image retrieval," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, South Carolina, pp. 222–227, 2000.
- [22] X. S. Zhou and T. S. Huang, "Small sample learning during multimedia retrieval using BiasMap," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Hawaii, pp. 11–17, Dec. 2001.
- [23] W. Y. Ma and B. S. Manjunath, "NeTra: A toolbox for navigating large image databases," *Multimedia System*, vol. 7, no. 3, pp. 184–198, 1999.
- [24] J. Z. Wang, J. Li, and G. Weiderhold, "SIMPLIcity: Semantic-sensitive integrated matching for picture libraries," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no.9, pp.947–963, 2001.
- [25] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation

- using expectation-maximization and its application to image querying,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp.1026–1038, 2002.
- [26] Y. X. Chen and J. Z. Wang, “A region-based fuzzy feature matching approach to content-based image retrieval,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no.9, pp.1252–1267, 2002.
- [27] B. Ko and H. Byun, “FRIP: A region-based image retrieval tool using automatic image segmentation and stepwise Boolean AND matching,” *IEEE Trans. Multimedia*, vol. 7, no.1, pp. 105–113, 2005.
- [28] T. P. Minka and R. W. Picard, “Interactive learning using a society of models,” *Pattern Recognition*, vol. 30, no. 4, pp. 565–581, 1997.
- [29] F. Jing, M. J. Li, H. J. Zhang, and B. Zhang, “Relevance feedback in region-based image retrieval,” *IEEE Trans. Circuits and Systems for Video Technology*, vol.14, no. 5, pp. 672–681, 2004.
- [30] ———, “An efficient and effective region-based image retrieval framework,” *IEEE Trans. Image Processing*, vol. 13, no. 5, pp. 699–709, 2004.
- [31] B. Ko and H. Byun, “Extracting salient regions and learning importance scores in region-based image retrieval,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol.17, no.8, pp. 1349–1367, 2003.
- [32] Z. Stejic, Y. Takama, and K. Hirota, “Relevance feedback-based image retrieval interface incorporating region and feature saliency patterns as visualizable image similarity criteria,” *IEEE Trans. Industrial Electronics*, vol. 50, no. 5, pp. 839–852, 2003.
- [33] C. Y. Li, M. C. Shih, and C. T. Hsu, “Image retrieval with relevance feedback based on graph-theoretic region correspondence estimation,” in *Proc. IEEE Int. Conf. Pattern*

- Recognition*, pp. 842–845, Aug. 2004.
- [34] I. Gondra, and D. R. Heisterkamp, “Learning in region-based image retrieval with generalized support vector machines,” in *Proc. 4th Int. Workshop on Multimedia Data and Document Engineering*, USA, July 2004.
- [35] C. Yang and T. Lozano-Perez, “Image database retrieval with multiple-instance learning techniques,” in *Proc. 16th Int. Conf. Data Engineering*, San Diego, CA, pp. 233–243, 2000.
- [36] C. C. Zhang, S. C. Chen, and M. L. Shyu, “Multiple object retrieval for image databases using multiple instance learning instance learning and relevance feedback,” in *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 775–778, June 2004.
- [37] R. Zhang and Z. Zhang, “Hidden semantic concept discovery in region based image retrieval,” in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 996–1001, June 2004.
- [38] ———, “A Bayesian framework for automatic concept discovery in image collections,” in *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 973–976, Aug. 2004.
- [39] [Online] <http://www.flickr.com/>
- [40] [Online] <http://www.smugmug.com/>
- [41] [Online] <http://www.youtube.com/>
- [42] N. Ansari and E. Hou, *Computational Intelligence for Optimization*. Boston: Kluwer Academic, 1997.
- [43] W. Pedrycz, *Computational Intelligence: an Introduction*. Boca Raton, FL: CRC Press, 1997.
- [44] S. Haykin, *Neural Networks a Comprehensive Foundation*. Upper Saddle River, NJ: Prentice-Hall, 1999.

- [45] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Santa Fe Institute, Addison-Wesley, 1991.
- [46] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [47] C. Cortes and V. N. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [48] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [49] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty and Information*. Englewood Cliffs, NJ: Prentice Hall, 1988.
- [50] B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice Hall, 1992.
- [51] L. A. Zadeh, "Fuzzy sets," *Journal of Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [52] T. Back, *Evolutionary Algorithms in Theory and Practice*. New York: Oxford University Press, 1996.
- [53] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press, 1995.
- [54] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
- [55] J. Han and K-K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Trans. Image Processing*, vol. 11, no. 8, pp. 944–952, August 2002.
- [56] S. Markus and O. Markus, "Similarity of color images," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, pp. 381–392, 1995.

- [57] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Conference*, San Juan, Puerto Rico, pp. 762–768, 1997.
- [58] J. R. Smith and S. F. Chang, "Single color extraction and image query," in *Proc. IEEE Conf. Image Processing*, pp. 528–531, 1995.
- [59] R. Haralick, K. Shanmugan, and I. Dinstein, "Textural features for image classification," *IEEE Trans. System, Man and Cybernetics*, vol. 3, no.6, pp. 610–621, 1973.
- [60] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception, " *IEEE Trans. System, Man and Cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.
- [61] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no.8, pp. 837–842, Aug. 1996.
- [62] J. R. Smith and S. F. Chang "Automated binary texture feature sets for image retrieval," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, GA, pp. 2239–2242, 1996.
- [63] T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree structured wavelet transform," *IEEE Trans. Image Processing*, vol. 2, no. 4, pp. 429–441, 1993.
- [64] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, no.1, pp. 25–39,1983.
- [65] H. Freeman and L. Davis, "A comer-finding algorithm for chain-coded curves," *IEEE Trans. Computers*, vol. 26, no. 3, pp. 297–303, 1977.
- [66] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," *IEEE*

Trans. System, Man and Cybernetics, vol. 7, no.3, pp. 170–179, 1977.

- [67] B. M. Mehtre, M. Kankanhalli, and W. F. Lee, "Shape measures for content based image retrieval: A comparison," *Information Processing and Management*, vol.33, no.3, pp. 319–337, 1997.
- [68] M. K. Hu, "Visual pattern recognition by moment invariants, computer methods in image analysis," *IRE Trans. Information Theory*, vol. 8, no. 2, pp.179–187, 1962.
- [69] A. Khotanzad, Y. H. Hong, "Invariant image recognition by Zemike moments," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no.5, pp. 489–497,1990.
- [70] E. M Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.13, no.3, pp. 209–216, 1991.
- [71] G. C. H. Chuang and C. C. J. Kuo, " Wavelet descriptor of planar curves: Theory and applications," *IEEE Trans. Image Processing*, vol. 5, no.1, pp.56–70, 1996.
- [72] S. Santini and R. Jain, "Similarity measures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 871–883, 1999.
- [73] R. K. Srihari, "Automatic indexing and content-based retrieval of captioned images," *IEEE Computer Magazine*, vol. 28, no. 9, pp. 49–56, 1995.
- [74] D. C. Tseng, Y. F. Li, and C. T. Tung, "Circular histogram thresholding for color image segmentation," in *Proc. 3rd Int. Conf. Document Analysis and Recognition*, 1995.
- [75] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 729–736, 1995.
- [76] Y. Rubner, C. Tomasi, and L. Guibas, "The Earth Mover's Distance as a metric for

- image retrieval,” *International Journal of Computer Vision*, vol. 40, pp. 99–123, 2000.
- [77] B. Li, E. Chang, and Y. Wu, “The discovery of a perceptual distance function for measuring image similarity,” *ACM Multimedia Journal Special Issue on Content-Based Image Retrieval*, vol. 8, no. 6, pp. 512–522, 2003.
- [78] D. Angluin, “Queries and concept learning,” *Journal of Machine Learning*, vol. 2, no. 4, pp. 319–342, 1988.
- [79] L. Wang and K. L. Chan, “Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval,” in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 629–634, 2003.
- [80] B. Li, E. Chang and C. S. Li, “Learning image query concepts via intelligent sampling,” in *Proc. IEEE Int. Conf. Multimedia and Expo*, Tokyo, pp. 1168–1171, Aug. 2001.
- [81] T. Joachims, “Transductive inference for text classification using support vector machines,” in *Proc. Int. Conf. Machine Learning*, pp. 200–209, 1999.
- [82] L. Wang and K. L. Chan, “Incorporating prior knowledge into SVM for image retrieval,” in *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 981–984, 2004.
- [83] Y. Mori, H. Takahashi, and R. Oka, “Image-to-word transformation based on dividing and vector quantizing images with words,” in *Proc. Int. Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.
- [84] P. Duygulu, K. Barnard, N. de Freitas and D. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” in *Proc. European Conf. Computer Vision*, pp. 97–112, 2002.
- [85] J. Jeon, V. Lavrenko and R. Manmatha, “Automatic image annotation and retrieval using cross-media relevance models,” in *Proc. Int. ACM SIGIR Conf. Research and*

- Development in Information Retrieval*, pp. 119–126, 2003.
- [86] F. Monay and D. Gatica-Perez, “On image auto-annotation with latent space models,” in *Proc. ACM Int. Conf. Multimedia*, pp. 275–278, 2003.
- [87] D. Blei, A. Ng, and M. Jordan, “Latent Dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [88] D. Blei and M. Jordan, “Modeling annotated data,” in *Proc. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 127–134, 2003.
- [89] E. Chang, G. Kingshy, G. Sychay, and G. Wu, “CBSA: Content-based soft annotation for multimodal image retrieval using Bayes point machines,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no.1, pp. 26–38, Jan. 2003.
- [90] J. Li and J. Z. Wang, “Automatic linguistic indexing of pictures by a statistical modeling approach,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075–1088, 2003.
- [91] K. Goh and E. Chang, “Using one-class and two-class SVMs for multiclass image annotation,” *IEEE Trans. Knowledge and Data Engineering*, vol. 17, no.10, pp. 1333–1346, 2005.
- [92] X. F. He, O. King, W. Y. Ma, M. J. Li, and H. J. Zhang, “Learning a semantic space from user’s relevance feedback for image retrieval,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, pp. 39–48, Jan. 2003.
- [93] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill Book Company, 1982.
- [94] A. P. Berman and L. G. Shapiro, “Efficient content-based retrieval: Experimental results,” in *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, pp. 55–61, 1999.

- [95] U. Gargi and R. Kasturi, "Image database querying using a multi-scale localized color representation," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pp. 28–32, 1999.
- [96] W. Hwang, J. Weng, M. Fang, and J. Qian, "A fast image retrieval algorithm with automatically extracted discriminant features," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pp. 8–15, 1999.
- [97] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.
- [98] T. Poggio and F. Girosi, "Networks for approximation and learning," in *Proc. IEEE*, vol. 78, pp. 1481–1497, 1990.
- [99] M. J. D. Powell, "Radial basis functions for multivariable interpolation: a review," *Algorithms for Approximation*, pp. 143–167, Oxford: Clarendon Press, 1987.
- [100] R. Fletcher, *Practice Methods of Optimization*. New York: Wiley, 1987.
- [101] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*, ACM Press, New York, 1999.
- [102] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [103] S. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & Fuzzy Systems*, vol. 2, no. 3, pp. 267–278, Sept. 1994.
- [104] J. Kittler, R. Hatef, R. P. W. Dubin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, Mar. 1998.
- [105] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *ACM Multimedia Systems Journal*, vol. 8, no. 6, pp. 536–544, 2003.
- [106] J. Friedman, "Regularized discriminant analysis," *Journal of American Statistics*

Association, vol. 84, no. 405, pp. 165–175, 1989.

[107] (1999) Corel Gallery Magic 65000. [Online] <http://www.corel.com/>

[108] C. F. Lin and S. D. Wang, “Fuzzy support vector machines,” *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[109] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: John Wiley & Sons, 2001.

[110] K. Wu and K. H. Yap, “An efficient radial basis function network approach for content-based image retrieval,” in *Proc. IEEE Int. Conf. on Acoustic, Speech, and Signal Processing*, Montreal, Canada, vol. 3, pp. 437–440, 2004.

[111] K. H. Yap and K. Wu, “A soft relevance framework in content-based image retrieval systems,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1557–1568, Dec. 2005.

[112] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[113] J. Findlay, “The visual stimulus for saccadic eye movement in human observers,” *Perception*, 9:7–21, Sept. 1980.

[114] J. Senders, “Distribution of attention in static and dynamic scenes,” in *Proc. SPIE*, vol. 3016, pp. 186–194, San Jose, Feb. 1997.

[115] A. Yarbus, *Eye Movements and Vision*. Plenum Press, New York NY, 1967.

[116] G. Elias, G. Sherwin and J. Wise, “Eye movements while viewing NTSC format television,” *SMPTE Psychophysics Subcommittee white paper*, Mar 1984.

[117] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” Technical report MSR-

- TR-99-87, Microsoft, Nov. 1999.
- [118] The Strait Times, 12 July 2006.
- [119] [Online] <http://images.google.com/>
- [120] [Online] <http://images.search.yahoo.com/>
- [121] G. Begelman, P. Keller, and F. Smadja, "Automated tag clustering: Improving search and exploration in the tag space," *15th International World Wide Web Conference*, Edinburgh, UK, May 2006.
- [122] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [123] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.
- [124] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge MA: MIT Press, 1996.
- [125] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proc. 2nd Int. Conf. Genetic Algorithms*, pp.14–21, 1987.
- [126] R. Lienhart, W. Effelsberg, and R. Jain, "VisualGREP: A systematic method to compare and retrieve video sequences," *Multimedia Tools and Applications*, vol. 10, no. 1, pp. 47–72, January 2000.
- [127] H. S. Chang, S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1269–1279, Dec. 1999.
- [128] X. S. Zhou and T. S. Huang, "Unifying keywords and visual content in image retrieval," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 23–33, April-June 2002.
- [129] Y. Lu, C. H. Hu, X. Q. Zhu, H. J. Zhang, Q. Yang, "A unified framework for semantics

and feature based relevance feedback in image retrieval systems,” in *Proc. ACM Int. Multimedia Conf.*, pp. 31–38, 2000.

- [130] F. Jing, M. J. Li, H. J. Zhang, and B. Zhang, “A unified framework for image retrieval using keyword and visual features,” *IEEE Trans. Image Processing*, vol. 14, no. 7, pp. 979–989, July 2005.
- [131] H. L. Tang, R. Hanka, and H. H. S. Ip, “A review of intelligent content-based indexing and browsing of medical images,” *Health Informatics Journal*, vol. 5 no.1, pp. 40–49, March 1999.
- [132] —, “Histological image retrieval based semantic content analysis,” *IEEE Trans. Information Technology in Biomedicine*, vol. 7, no.1, pp. 26–36, March 2003.

Appendix A

Gradient-descent algorithm is used to train the PFRBFN at each feedback iteration by minimizing the cost function E_t with respect to the network parameters of $U_t(\mathbf{x})$, namely,

$$\boldsymbol{\theta} = \{w'_{\alpha i}, \mathbf{v}'_{\alpha i}, \sigma'_{\alpha i} \mid \alpha \in \{P, N, F\}, i = 1, 2, \dots, I_{\alpha}\}, \text{ where } E_t = \frac{1}{2} \sum_{j=1}^{N_t} e_{jt}^2 = \frac{1}{2} \sum_{j=1}^{N_t} (Y_t(\mathbf{x}_j) - F_t(\mathbf{x}_j))^2.$$

We now develop in detail the updating equations for $w'_{\alpha i}$, $\mathbf{v}'_{\alpha i}$, and $\sigma'_{\alpha i}$. First, the partial derivative of E_t with respect to $w'_{\alpha i}$ is calculated as follows:

$$\frac{\partial E_t}{\partial w'_{\alpha i}} = - \sum_{j=1}^{N_t} e_{jt} \frac{\partial F_t(\mathbf{x}_j)}{\partial w'_{\alpha i}} \quad (\text{A.1})$$

As only the parameters of the update network $U_t(\mathbf{x}_j)$ need to be estimated in each feedback iteration, we obtain:

$$\begin{aligned} \frac{\partial E_t}{\partial w'_{\alpha i}} &= - \sum_{j=1}^{N_t} e_{jt} \frac{\partial U_t(\mathbf{x}_j)}{\partial w'_{\alpha i}} \\ &= - \sum_{j=1}^{N_t} e_{jt} \frac{\partial \sum_{\alpha \in \{P, N, F\}} \sum_{i=1}^{I_{\alpha}} w'_{\alpha i} f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}, \sigma'_{\alpha i})}{\partial w'_{\alpha i}} \\ &= - \sum_{j=1}^{N_t} e_{jt} f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}, \sigma'_{\alpha i}) \end{aligned} \quad (\text{A.2})$$

Hence, weight estimation at the k -th learning iteration is:

$$\frac{\partial E_t(k)}{\partial w'_{\alpha i}(k)} = - \sum_{j=1}^{N_t} e_{jt}(k) f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}(k), \sigma'_{\alpha i}(k)) \quad (\text{A.3})$$

$$w'_{\alpha i}(k+1) = w'_{\alpha i}(k) - \eta_1 \frac{\partial E_t(k)}{\partial w'_{\alpha i}(k)}, \quad \alpha \in \{P, N, F\}, \quad i = 1, 2, \dots, I_\alpha \quad (\text{A.4})$$

Similarly,

$$\begin{aligned} \frac{\partial E_t}{\partial \mathbf{v}'_{\alpha i}} &= - \sum_{j=1}^{N_t} e_{jt} \frac{\partial U_t(\mathbf{x}_j)}{\partial \mathbf{v}'_{\alpha i}} \\ &= - \sum_{j=1}^{N_t} e_{jt} \frac{\partial \sum_{\alpha \in \{P, N, F\}} \sum_{i=1}^{I_\alpha} w'_{\alpha i} f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}, \sigma'_{\alpha i})}{\partial \mathbf{v}'_{\alpha i}} \\ &= - \sum_{j=1}^{N_t} e_{jt} \frac{\partial \sum_{\alpha \in \{P, N, F\}} \sum_{i=1}^{I_\alpha} w'_{\alpha i} \exp\left(-\frac{(\mathbf{x}_j - \mathbf{v}'_{\alpha i})^T \mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}'_{\alpha i})}{2(\sigma'_{\alpha i})^2}\right)}{\partial \mathbf{v}'_{\alpha i}} \\ &= -w'_{\alpha i} \sum_{j=1}^{N_t} e_{jt} \exp\left(-\frac{(\mathbf{x}_j - \mathbf{v}'_{\alpha i})^T \mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}'_{\alpha i})}{2(\sigma'_{\alpha i})^2}\right) \frac{\mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}'_{\alpha i})}{(\sigma'_{\alpha i})^2} \\ &= -w'_{\alpha i} \sum_{j=1}^{N_t} e_{jt} f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}, \sigma'_{\alpha i}) \frac{\mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}'_{\alpha i})}{(\sigma'_{\alpha i})^2} \end{aligned} \quad (\text{A.5})$$

Hence, center estimation at the k -th learning iteration is:

$$\frac{\partial E_t(k)}{\partial \mathbf{v}'_{\alpha i}(k)} = -w'_{\alpha i}(k) \sum_{j=1}^{N_t} e_{jt}(k) f(\mathbf{x}_j, \mathbf{v}'_{\alpha i}(k), \sigma'_{\alpha i}(k)) \frac{\mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}'_{\alpha i}(k))}{(\sigma'_{\alpha i}(k))^2} \quad (\text{A.6})$$

$$\mathbf{v}'_{\alpha i}(k+1) = \mathbf{v}'_{\alpha i}(k) - \eta_2 \frac{\partial E_t(k)}{\partial \mathbf{v}'_{\alpha i}(k)}, \quad \alpha \in \{P, N, F\}, \quad i = 1, 2, \dots, I_\alpha \quad (\text{A.7})$$

Furthermore,

$$\begin{aligned}
\frac{\partial E_t}{\partial \sigma_{\alpha i}^t} &= -\sum_{j=1}^{N_r} e_{jt} \frac{\partial U_t(\mathbf{x}_j)}{\partial \sigma_{\alpha i}^t} \\
&= -\sum_{j=1}^{N_r} e_{jt} \frac{\partial \sum_{\alpha \in \{P, N, F\}} \sum_{i=1}^{I_\alpha} w_{\alpha i}^t f(\mathbf{x}_j, \mathbf{v}_{\alpha i}^t, \sigma_{\alpha i}^t)}{\partial \sigma_{\alpha i}^t} \\
&= -\sum_{j=1}^{N_r} e_{jt} \frac{\partial \sum_{\alpha \in \{P, N, F\}} \sum_{i=1}^{I_\alpha} w_{\alpha i}^t \exp\left(-\frac{(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t)^T \mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t)}{2(\sigma_{\alpha i}^t)^2}\right)}{\partial \sigma_{\alpha i}^t} \\
&= -w_{\alpha i}^t \sum_{j=1}^{N_r} e_{jt} \exp\left(-\frac{(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t)^T \mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t)}{2(\sigma_{\alpha i}^t)^2}\right) \frac{(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t)^T \mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t)}{(\sigma_{\alpha i}^t)^3} \\
&= -w_{\alpha i}^t \sum_{j=1}^{N_r} e_{jt} f(\mathbf{x}_j, \mathbf{v}_{\alpha i}^t, \sigma_{\alpha i}^t) \frac{(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t)^T \mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t)}{(\sigma_{\alpha i}^t)^3}
\end{aligned} \tag{A.8}$$

Hence, width estimation at the k -th learning iteration is:

$$\frac{\partial E_t(k)}{\partial \sigma_{\alpha i}^t(k)} = -w_{\alpha i}^t(k) \sum_{j=1}^{N_r} e_{jt}(k) f(\mathbf{x}_j, \mathbf{v}_{\alpha i}^t(k), \sigma_{\alpha i}^t(k)) \frac{(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t(k))^T \mathbf{\Lambda}(\mathbf{x}_j - \mathbf{v}_{\alpha i}^t(k))}{(\sigma_{\alpha i}^t(k))^3} \tag{A.9}$$

$$\sigma_{\alpha i}^t(k+1) = \sigma_{\alpha i}^t(k) - \eta_3 \frac{\partial E_t(k)}{\partial \sigma_{\alpha i}^t(k)}, \quad \alpha \in \{P, N, F\}, \quad i = 1, 2, \dots, I_\alpha \tag{A.10}$$