Image retrieval and classification based on MPEG-7 descriptors

Wang, Surong

2008

Wang, S. R. (2008). Image retrieval and classification based on MPEG-7 descriptors. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/2600

https://doi.org/10.32657/10356/2600

NANYANG TECHNOLOGICAL UNIVERSITY

IMAGE RETRIEVAL AND CLASSIFICATION BASED ON MPEG-7 DESCRIPTORS

WANG SURONG

SCHOOL OF COMPUTER ENGINEERING

2008

Image retrieval and classification based on MPEG-7 descriptors

Wang Surong

School of Computer Engineering

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

2008

Acknowledgments

I would like to express my gratitude to my supervisor, Dr. Chia Liang-Tien, for his advice and support throughout the whole process of my Ph.D study. I am sincerely grateful for his guidance, understanding and patience in leading me through this effort. My research could not have been finished without insightful advice from him.

I would like to extend my thanks to Dr. Deepu Rajan and Dr. Manoranjan Dash for their thorough help and supervision. They kindly assisted me in many aspects.

I also would like to thank all my colleagues in Center for Multimedia and Network Technology (CeMNet), for their understanding and supports. In particular, I'd like to thank Xu Min, Xie Jun, Liu Song, Zhou Chen, Yi Haoran, Chen Ling, Chu Yang, Hu Yiqun and Wang Huan.

Abstract

In order to make use of the vast amount of multimedia data, efficient and effective techniques to analyze and retrieve multimedia information based on its content need to be developed. In this thesis, we investigate three issues related to image retrieval and classification based on MPEG-7 descriptors.

Firstly, we propose a new similarity measure for one MPEG-7 color descriptor based on Earth Mover's Distance (EMD). To reduce the computation time, M-tree index and lower bound of EMD are discussed, which can prune the images far from the query image. To combine two or more different MPEG-7 descriptors to improve the retrieval performance, a descriptor-weighting scheme for combining multiple MPEG-7 visual descriptors is discussed. An optimization model is built to find a set of optimal weights for a set of corresponding descriptors. Explicit solutions can be derived by Lagrange multipliers, which are optimal and easy to calculate.

Secondly, in order to minimize the semantic gap, an object category classification model based on regions is proposed. This model can learn and classify objects by training the model with various objects within the same category. Each object category is represented by a constellation of representative parts, i.e., the regions. During the learning procedure, the similarity distance between any two regions is calculated and accumulated as a frequency measure. Then the regions with small frequency of appearance are removed from the image model iteratively. At last, a small set of representative regions with suitable weights are kept as the image model. When this image model is used for classification of object category, the similarity distance based on appearance of single region and the geometric distortion between a pair of regions are both computed. Furthermore, a graph matching algorithm is applied to use the nested relationships between the regions to improve the performance.

Finally, to handle multimedia applications where data size is usually very large, we propose an EASIER sampling algorithm and verify it with various applications. The proposed EASIER is a new and efficient method for sampling of large and noisy multimedia data. It compares the histograms of the sample set and the whole set to estimate the representativeness of the sample. EASIER deals with noise in an elegant manner, which simple random sampling (SRS) and other methods are not able to deal with. We experiment on image and audio datasets. Comparison with SRS and other sampling methods shows that EASIER is vastly superior in terms of sample representativeness particularly for small sample sizes, although time-wise it is comparable to SRS, the least expensive method in computation time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last a few years, with the huge amount increase of digital multimedia data, there is a need for more efficient ways to manage and retrieve multimedia data, based on their content. Although some techniques are presented to meet the requirements, such as content-based image retrieval (CBIR), these techniques are still difficult for the typical end users, especially for those without basic image knowledge. The main difficulties include:

- Lack of standard and powerful description. Identifying and managing the multimedia data efficiently is becoming more difficult. ,
- Lack of semantic object description. It is still hard to depict the object clearly and precisely based on low-level features.
- Lack of efficient management for huge multimedia data. It is difficult to handle large amount of multimedia data for various applications.

The new MPEG-7 standard, formally named "Multimedia Content Description Interface", has been developed to address the issue of audiovisual information description. MPEG-7 standard can improve current multimedia applications and enable new exciting ones such as efficient organization, management, and retrieval of multimedia content. It

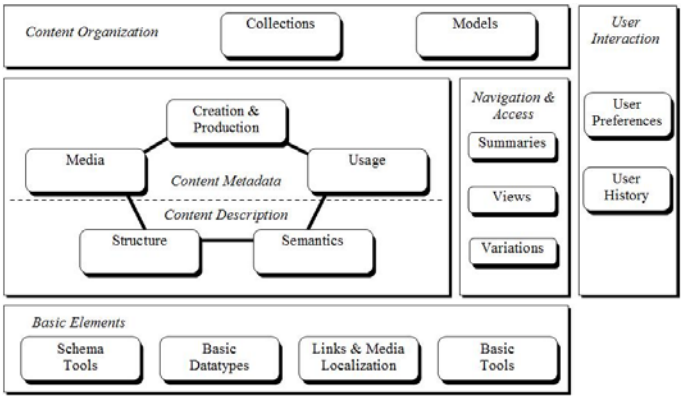


Figure 1.1: The MPEG-7 Multimedia Description Schemes.

provides a rich set of standard description tools to describe multimedia content, such as the description of image content - visual descriptors, and Multimedia Description Schemes (MDS). Figure 1.1 illustrates the structure of MPEG-7 MDS. However, methods of using these description tools to achieve efficient and effective content-based retrieval still need to be further investigated. For image retrieval based on low-level descriptors such as color, texture and shape, similarity measures between images are important and MPEG-7 does not standardize the measures. More advanced search approaches, such as search based on multi-feature, multi-region or the combination need further research.

One of the major issues in content-based image retrieval is the so-called semantic gap - the mismatch between the capabilities of current CBIR systems and the conceptual needs of users, as well as using low-level features to correspond to high-level abstractions [7]. MPEG-7 provides various tools to annotate the images at the semantic level, and it will be helpful to bridge this gap. But the extraction of high-level semantics from low-level features, or the extraction of high level descriptors from low-level descriptors is still a research challenge.

1.1 Objectives

The purpose of our work is to put forward some approaches to increase the effect of image retrieval and classification based on MPEG-7 descriptors, and minimize the semantic gap. Summarizing the issues above, we have the following objectives.

1. Propose and implement an image retrieval system based on MPEG-7 descriptors. Briefly speaking, we want to create a MPEG-7 compliance CBIR system with multiple functions. It is a distributed image retrieval system, which can search based on regions or whole images. Multiple descriptors are used to get the retrieval results. During this procedure, some algorithms about multiple descriptors retrieval and region-based retrieval will be investigated.

2. Try to minimize the semantic gap between low-level features and semantic requirements, such as the object-based image retrieval and classification. Besides the retrieval of whole image based on low-level features, we also want to identify the objects in the images. Currently research works attempt to obtain and use the semantics of image to perform better retrieval. Towards this goal, segmentation of an image into regions has been used in recent years, since local properties of regions can help the matching of objects between images, and thereby contribute towards a more effective CBIR system. Despite the effort of researchers over a number of decades, this objective has remained unsolved for the most part. Although reasonably successful attempts have been made for certain classes of objects, such as human faces, no satisfactory methods exist that work with any category of object. Our goal is to be able to identify object, especially object categories within images. Then user can put semantic labels to the object categories. It may be useful for semantic approach. As regions can represent the objects and MPEG-7 can be applied to the region of images, we will build a region-based object model to represent the objects and the object categories.

3. Efficient management issues related to multimedia content, such as sampling for large and noisy dataset. As the amount of multimedia data is increasing day-by-day, thanks to less expensive storage devices and increasing numbers of information sources, many multimedia applications, such as machine learning algorithms, are faced with large-sized and noisy datasets. Fortunately, the use of a good sampling set can represent the data properly. But using a simple random sample may not obtain satisfactory results, because such a sample may not adequately represent the large and noisy dataset, due to its blind approach in selecting samples. The difficulty is particularly apparent for huge datasets where, due to memory constraints, only very small sample sizes can be used. This is typically the case for multimedia applications, where data size is usually very large. A new and effective method to sample of large and noisy multimedia data is important for various multimedia applications.

1.2 Main contributions

In this thesis, an overall system of image related management and retrieval is proposed and implemented. Figure 1.2 describes the overview structure of the work. The main contributions of the thesis are summarized as follows:

1. Investigate image retrieval based on MPEG-7 descriptors. At the beginning, a better distance measure is applied to a visual descriptor, Dominant Color Descriptor (DCD), with suitable index method. To apply two or more different descriptors to improve the results, a descriptor-weighting scheme for combining multiple MPEG-7 visual descriptors is discussed. An optimization model is built to find a set of optimal weights for a set of descriptors. Explicit solutions can be derived by Lagrange multipliers, which are optimal and easy to calculate. The calculation procedure is fully automatic and no manual work is needed. Experiments show that better retrieval results can be achieved compared with

using single descriptor only. The results are also better than simple average combination of descriptors.

2. Region-based image retrieval and classification. An object category classification model based on regions is proposed in this thesis. The region-based model can learn and classify objects, by training the model with different objects within the same category. Each object category is represented by a constellation of representative parts, i.e., the regions. These regions are detected by salient region detector over suitable scales. The standard conjunction rule is applied to construct the image model. During the learning procedure, the similarity distance between any two regions is calculated and accumulated as a frequency measure. This measure is inversely proportional to the probability of a match. The regions with small frequency of appearance are removed from the image model iteratively. After clustering, a small set of representative regions are kept as the image model. Each representative region in the model has a representation weight, which is normalized based on the number of regions in corresponding clusters. When this image model is used for object classification, the similarity distance based on appearance of single region and the geometric distortion between a pair of regions are both considered. In order to make use of the nested spatial relationships between the regions, we further introduce a graph-based matching algorithm to find the corresponding regions in the image model and images in the database. Experimental results show that the image model based on representative regions is easy to calculate and can obtain efficient results.

3. Propose an EASIER sampling algorithm and verify it with various applications. The proposed EASIER algorithm is used for sampling of large and noisy multimedia data. It compares the histograms of the sample set and the whole set to estimate the representativeness of the sample. EASIER deals with noise in an elegant manner which simple random sampling (SRS) and other methods are not able to deal with. We experiment on image and audio datasets. Comparison with SRS and other sampling methods

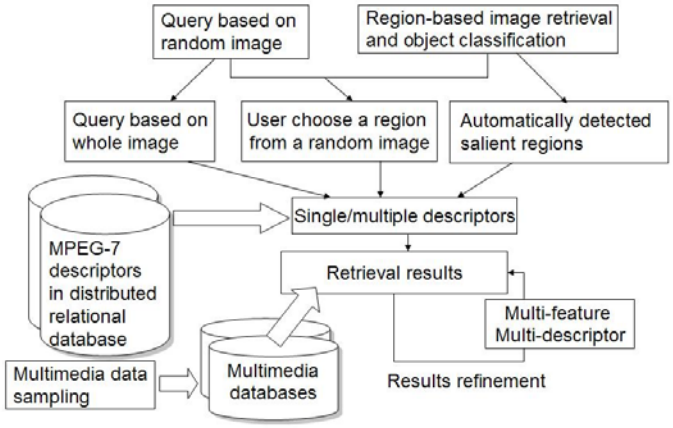


Figure 1.2: The basic structure of whole image retrieval and classification system.

shows that the proposed EASIER is vastly superior in terms of sample representativeness, particularly for small sample sizes although time-wise it is comparable to SRS, the least expensive method in computation time.

1.3 Organization of the thesis

The structure of the thesis is as follows. In Chapter 2, some background knowledge and related techniques in image retrieval and management are reviewed. The overview of MPEG-7 standard and visual descriptors is also presented in this chapter. Chapter 3 introduces several approaches of image retrieval based on MPEG-7 descriptors, including single and combination of multiple descriptors. In Chapter 4, we investigate scale invariant points, features and their applications. A constellation model based on salient regions is proposed for classification of object categories. We describe how the constellation model is trained and applied for classification in detail. In Chapter 5, a sampling algorithm EASIER is proposed to efficiently select representative samples from the large and noisy dataset. The performance of EASIER is evaluated using various applications. Finally, in Chapter 6, we draw conclusions about the approaches related to image retrieval and management and discuss the future work.

Chapter 2

Literature review

In this chapter, literatures related to image retrieval and organization methods are reviewed for three specific areas, including image description and retrieval, object representation and classification, and efficient sampling for multimedia database.

2.1 Content-based image retrieval

With the huge amount of digital images, how to effectively index and retrieve the images becomes a problem. Traditional text-based index methods can be used to retrieve images which are annotated with descriptive text. Although there are some tools for automatic image annotation [8, 9], mostly they only provided simple classes labels. The simple annotations are insufficient to capture all the image content, and cannot be described in a standard way. Creating detail text annotations for the images is usually manual and very time consuming These text-based index methods are not practical for visual information indexing.

As a result, content-based image retrieval (CBIR) systems attempt to overcome these problems of text-based searching. CBIR is the attempt to search for visual content in media databases by deriving meaningful features and measuring the dissimilarity of visual objects based on distance functions [10]. In [11], almost 300 key theoretical and

empirical contributions in the current decade related to image retrieval and automatic image annotation are analyzed.

CBIR provides powerful tools to automatically extract and compare the visual features, and have the capability to retrieve images. Users can specify visual features of images in more direct and natural ways than the text specifications for traditional databases. One powerful approach is to let the users express the query terms with images rather than words.

For content based image retrieval, users can use different methods to specify a visual query, such as query by example, query by sketch, and query by keywords etc. Query by example (QBE) is to search for images based on an existing image [12]. The system automatically compares the query image to those in the database and retrieves the most similar ones. This image-based query is very intuitive and widely used in many image retrieval systems. Another method is known as query by sketch [12]. It uses some alternate specifications to represent a query image (the most natural method is by sketching). Query by sketch has the advantage that no initial image is required to perform a query. These approaches can be used independently or integrated with text-based methods.

In principle, similarity queries have two basic types:

1. Range query: Given a query image, get all images within a given maximum search distance (threshold) to the query image.

2. K-Nearest-Neighborhood (KNN) query: Given a query image and an integer $K > 1$, get the top K nearest neighbors in distance to the query image.

CBIR has two main steps: feature extraction and similarity measurement. Firstly, CBIR system must be able to analyze an image to extract key visual features such as shape, color and texture. Then the system requires similarity measures to determine the similarity between two images. In addition, to improve the retrieval efficiency, the system can build indices based on the extracted features. Currently there are many CBIR

systems, such as the famous QBIC (IBM) [13], ImageScape [14], and imgSeek [15]. They provide different methods to query, including query by image, query by sketch, query by region, and query by combination of several features. These systems retrieve images only by similarity of appearance, using low level visual features such as color, texture or shape. They use their own feature vectors rather than MPEG-7 descriptors. With the MPEG-7 standardization, we are starting to see a few prototypes of MPEG-7 compliant image retrieval systems, such as "MPEG-7 Homogeneous Texture Descriptor Demo" [16], "PicSom" (content-based image retrieval with self-organizing maps) [17].

2.2 MPEG-7 standard

In this section, we briefly introduce the MPEG-7 standard, especially the visual descriptors in MPEG-7.

2.2.1 Overview of MPEG-7 standard

The tremendous growth of multimedia content is driving the need for more effective and efficient methods for storing, filtering and retrieving audiovisual data. MPEG-7 is a multimedia standard, which can further improve content-based retrieval by providing a rich set of standardized descriptors and description schemas for describing multimedia content. The scope of MPEG-7 is shown in Figure 2.1. The normative part of MPEG-7 includes Descriptors and Descriptor Schemas, while how to extract (produce) and use these descriptions for further processing (e.g. retrieval systems) are not standardized. It gives maximum flexibility to various applications. Both automatic systems and human users, which process audiovisual information, are within the scope of MPEG-7 [18].

MPEG-7 provides a comprehensive set of standardized tools to describe multimedia content, such as Descriptors (Ds), Descriptor Schemas (DSs), and Description Definition Language (DDL). Figure 2.2 shows the relationships among Ds, DSs and DDL.

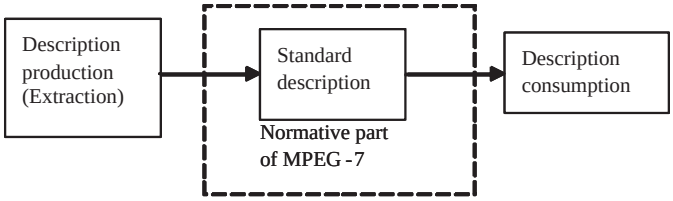


Figure 2.1: The scope of MPEG-7 standard.

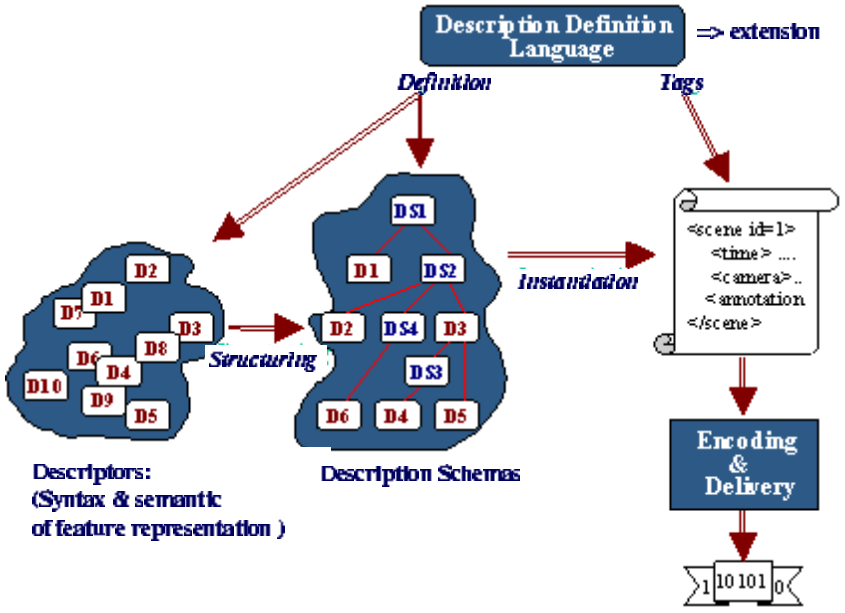


Figure 2.2: The relationship among the MPEG-7 elements.

Descriptors are defined primarily to describe low-level features, and shall be extracted automatically in most applications. They can be represented in textual format (XML), in binary format (BiM, Binary format for Multimedia description streams), or a mixture of the two formats, according to different applications.

2.2.2 Visual descriptors in MPEG-7 standard

In this section, we give a brief introduction for MPEG-7 visual descriptors. The MPEG-7 visual descriptors define a rich set of image and video features, which can describe various aspects of visual content in a compact style. All visual descriptors and basic

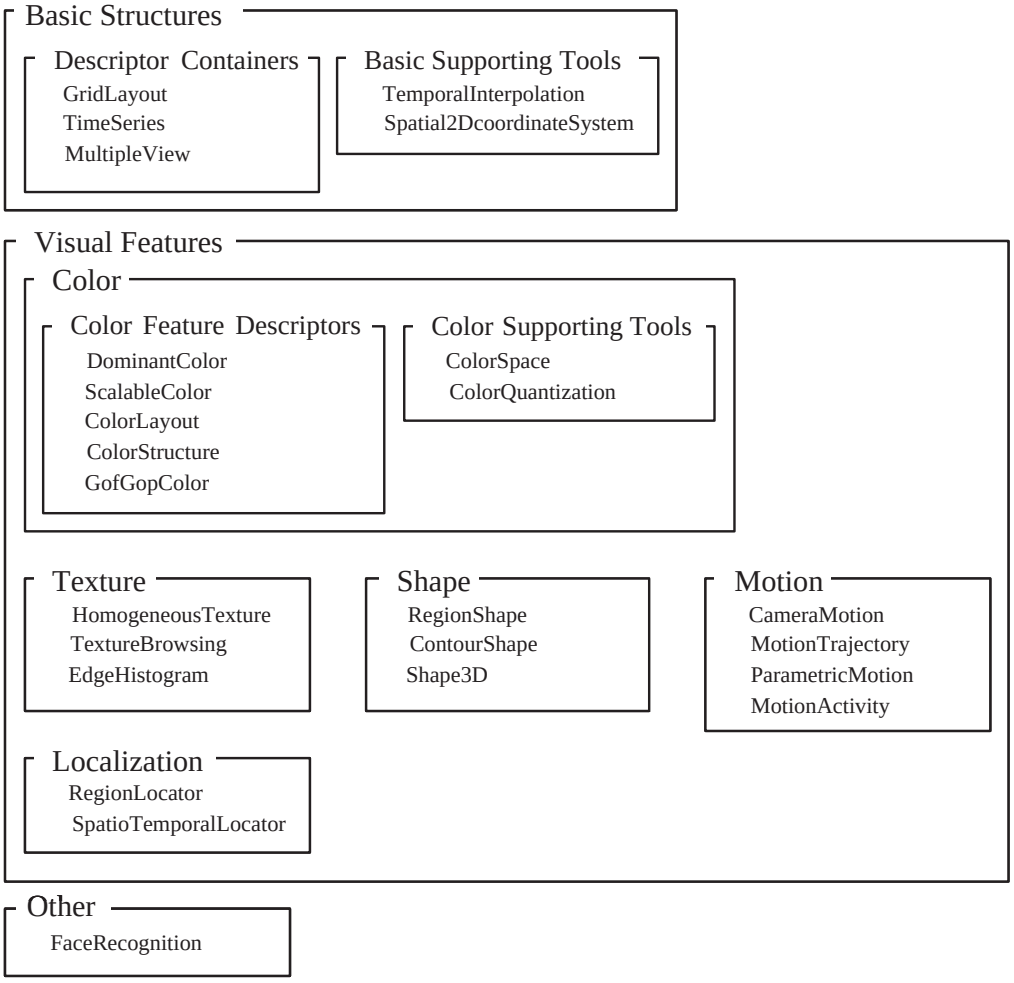


Figure 2.3: An overview of MPEG-7 visual descriptors.

structures in MPEG-7 are shown in Figure 2.3. These low-level descriptors include color, texture, shape, and motion descriptors, which describe different features of visual content, and a face-recognition descriptor, which is application-dependent. More details of these descriptors can be found in the MPEG-7 visual standard [2].

2.2.2.1 Color descriptors

As shown in Figure 2.3, currently MPEG-7 color descriptors include color supporting tools and color feature descriptors.

Color supporting tools include Color Space Descriptor and Color Quantization Descriptor. Color Space Descriptor specifies the selection of a color space to be used in other color descriptors. The color spaces used in MPEG-7 include RGB, YCbCr, HSV (hue-saturation-value), HMMD (Hue-Max-Min-Difference), Monochrome and Linear transformation matrix with reference to RGB. Among them HMMD is a new color space defined for MPEG-7. Color Space Descriptor uses continuous value to define the color components and quantization is necessary to form the discrete representation. The Color Quantization Descriptor specifies the number of quantization levels for each color component in the color space. It is assumed that each of the color components in a given color space uses uniform quantization.

Besides the color supporting tools, color descriptors consist of a Dominant Color Descriptor (DCD), a Color Layout Descriptor (CLD), and some color histogram descriptors, including Scalable Color Descriptor (SCD), Group of Frames/Picture Descriptor (GoF/GoP) and Color Structure Descriptor (CSD). These descriptors represent many different aspects of the color feature, including spatial layout and structure of color and color distribution. Figure 2.4 gives an example of color structure representation.

DCD describes the dominant colors of an image. It can specify a small number of dominant color values and their statistical properties including percentage and variance. DCD provides an effective, compact and intuitive description of the representative colors in an image or region. CSD is identical in form to a color histogram but is semantically different. Both the color distribution of the image (like a color histogram) and the local spatial structure of the color are represented by CSD. CSD can be used to distinguish the images with same color histogram because of the additional spatial information of colors.

CLD captures the spatial layout of the dominant colors on a grid superimposed on the region of interest [18]. It is a very compact descriptor that is effective in high-speed

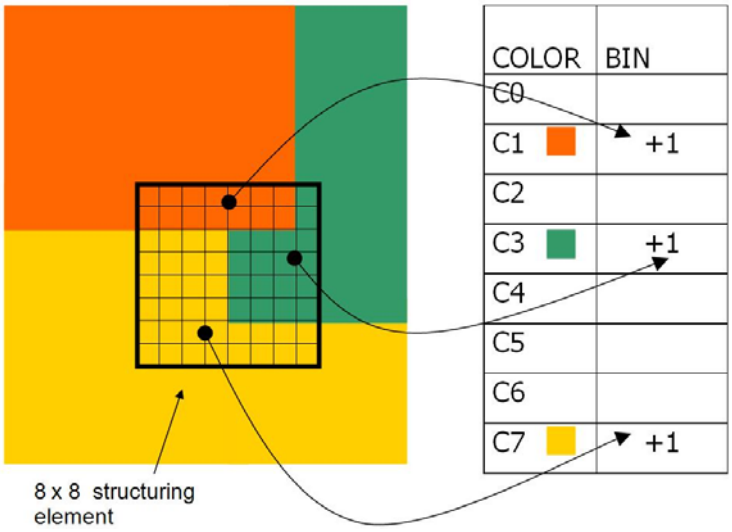


Figure 2.4: Color structure histogram accumulation for CSD, which represents the information of color structure [2].

browsing and search applications. SCD is derived from a color histogram with fixed color space quantization in the HSV color space. A novel Haar transform coefficient is used to encode the color histogram. GoF/GoP descriptor is an extension of SCD to be used for a collection of pictures or a group of frames from a video. It specifies several different ways to construct the color histogram.

2.2.2.2 Texture descriptors

Texture, is a powerful low-level feature for image search and retrieval applications. The texture descriptors in MPEG-7 can be used to browse and retrieve image and video databases. Currently Homogenous Texture Descriptor (HTD), Texture Browsing Descriptor (TBD) and Edge Histogram Descriptor (EHD) are included in MPEG-7. HTD quantitatively characterizes the homogeneous texture regions for similarity retrieval. TBD provides the characterization of perceptual attributes such as directionality, regularity, and coarseness of a texture. It is based on computation of the local spatial-frequency statistics of the texture. EHD is used for the region which is not homogeneous in texture

properties [19]. These three descriptors are suitable for similarity matching and retrieval, of both homogenous (HTD and TBD) and non-homogenous texture (EHD).

2.2.2.3 Shape descriptors

MPEG-7 provides three shape descriptors both for 2-D and 3-D objects, including Contour Shape Descriptor, Region Shape Descriptor and 3-D Shape Descriptor. Contour Shape Descriptor can efficiently describe objects whose contours represent their characteristic shape features. Region Shape Descriptor expresses pixel distribution within an object or region. They are both used for 2-D objects. 3-D Shape Descriptor expresses characteristic features of objects represented as discrete polygonal meshes.

2.2.2.4 Other visual descriptors

1. Motion descriptors

All the color, texture and shape descriptor in MPEG-7 can be used to retrieve or index images of video sequences. Besides these visual descriptors described above, four motion descriptors are specially developed to capture and describe essential motion characteristics in MPEG-7. They are Motion Activity Descriptor, Camera Motion Descriptor, Motion Trajectory Descriptor and Parametric Motion Descriptor.

2. Face Recognition Descriptor

Here is also an application-dependent visual descriptor, Face Recognition Descriptor. The Face Recognition descriptor is not associated with any particular visual feature. It can be used to describe a human face for applications requiring the matching and retrieval of face images [2]. This descriptor is based on the Principal Component Analysis (PCA) technique. The Face Recognition Descriptor is in fact a 48-element vector which represents the projection of a face vector onto a set of 48 basis vectors (face patterns). These basis vectors are extracted from eigenvectors of a set of normalized training face images.

2.3 Image retrieval based on multiple features

Since its advent, content-based image retrieval (CBIR) has attracted great research attention. Early research of CBIR has focused on just one low-level visual feature. As features extracted from different techniques emphasize image attributes in different domains, it would be more accurate to use not only one feature, but also the combination of multiple features. Combination of multiple features within a single model has been investigated as a promising technique to increase the retrieval efficiency. In image retrieval, queries can use several features such as color, shape, texture or text. In such a case, the accuracy and efficiency of retrieval depend much on how a multi-features query is decoupled into search of each individual feature.

The simple and easy way is average combination. In [20] the color, texture and other features are combined directly using specified weights to obtain a better result as in Eq. 2.1, where f_i is the ith feature and w_{f_i} is the corresponding weigh for f_i. $dist(f_i)$ is the distance calculated based on f_i only. One optimization challenge is to guarantee the correct retrieval of the k top-ranked results efficiently, when combining multi-features result lists. Previously significant work in this area is in [21]. The search involves a *sorted access phase* and a *random access phase* over multiple features. In *sorted access phase*, the result lists ordered by ascending distance are collected based on each feature separately (each query is called an atomic query). In next *random access phase*, the distances of objects based on other features are computed. After these two phases, all candidate objects can get a combined distance based on some combination functions and its distance of each feature. The combination function is based on fuzzy mathematic. This algorithm is asymptotically optimal in terms of database size with arbitrarily high probability, however only for uniform score distributions — which very rarely occur in

practice.

$$distance = \sum_i w_{f_i} dist(f_i) \qquad \text{(Eq. 2.1)}$$

In [22] Quick-Combine algorithm is proposed for combining multi-feature result lists. Compared with Fagin's algorithm [21] an improved termination condition is developed in combination with a heuristic control flow adapting itself narrowly to the particular score distribution. Top-ranked results can be computed and output incrementally.

It can be seen that the previous methods focus on how to efficiently terminate the combination procedure, i.e., obtain the retrieval results. The weights of different features are based on fuzzy mathematic and cannot adjust. As in linear combination, feature weighting is an important issue, there are various techniques applied to dynamically adjust the weights of different features. In [23], a neural network model for merging heterogeneous features is presented. This model can be used to determine nonlinear relationship between features. In [24], a vigorous optimization formulation is presented to effectively learn from the users when there are multiple visual features in the retrieval system. Lagrange multipliers are applied to derive explicit solutions, which are both optimal and fast to compute. In [25], a weighted cascading algorithm is applied to optimize the search time when multiple features are used to retrieve. The basic procedure is that each feature is compared in sequence and has a relative importance. The rank created by one feature determines the weight of next feature. The features with lower weight use a subset of the total database determined by the higher weight features. In [26], instead of giving every feature a weight explicitly, the importance of a feature is regulated implicitly by learning a user's perception based on Bayesian Learning. Then the process of feature combination is adaptive. In [27], the weight of every feature is determined gradually according to user's retrieval goal. Overall, it can be seen that

different weights for features represent the corresponding importance. The self-adaptive weights can approximate to a user's perception better.

2.4 Region-based image retrieval and object classification

Recently, region and object based image retrieval has attracted significant research attention. Users usually focus on part of the image, which can be represented as the Region-Of-Interest (ROI). The difficulty in object-based image retrieval is the identification of an object under different viewing conditions. In this case, the query might contain objects at a different scale and orientation from those present in the database. Hence, for object recognition, this factor should also be considered in addition to the common visual features, such as color, texture and shape.

2.4.1 Points focus on local features

Representing an image by local interest points allows efficient retrieval of images, as only local information is required to be stored. Different point detectors have been proposed and applied for image matching, such as Harris points [28], extreme in the normalized scale-space of the Laplacian of the image [29] and Harris-Laplacian, a combination of both [6]. An overview of different interest points is provided in [30]. Besides the classic corner detectors, wavelet-based detector is also widely used for point detection. In [31], a detector based on wavelet transform to detect global variations as well as local ones is presented. The salient point detector can extract points where variations occur in the image, whether they are corner-like or not. In [32] the interest points are estimated with significant luminance variations. A small region around the interest point is located as an image patch. In [33] and [34], the color information is involved in the detection of interest points.

After the detection of scale-space extreme points, corresponding features can be extracted at the points to represent the objects. These scale and orientation invariant features which represent the images are used in image retrieval. In [6] the scale-invariant points are detected by the Harris-Laplacian detector in a multiple scale representation of image, which is built by the convolution with a series Gaussian kernel. The corresponding derivatives at the scale-invariant points are calculated as orientation-invariant feature vectors. Rotation invariance is obtained by selecting the derivatives in the gradient direction. In [35] key points are identified by finding peaks of a Difference-of-Gaussian (DOG) function convolved with different scales of an image. The SIFT (Scale Invariant Feature Transform) keys are created by representing blurred image gradients in multiple orientation planes and at multiple scales.

To evaluate the similarity between images based on the points, the distance between the set of points can be calculate using different methods. The distance can be directly used for comparison, or some alternative measures based on the distance can be evaluated. In [6] a voting algorithm is used to select the most similar images in the database. The distance between two matched points is evaluated by a vote, with the image getting the highest number of votes being the most similar results. In [36] the points are sampled from contours on the shape of objects. To reduce the details of the local points, geometric blur is applied as the features. The matching points are determined by a complex Binary Quadratic Optimization model. The results demonstrate that using one image as training can obtain acceptable results. In [37], sparse feature-based approach is proposed for comparing hierarchical object models to multi-scale features extracted from image data. This measure is applied for evaluating the likelihood of object models. Furthermore, the notion of feature likelihood maps based on dense filter is developed to avoid an explicit feature extraction step and to evaluate models using a function defined directly from the image data.

As the target is to find the identical points by the suitable distance of point matching, these points and their features focus on the local characteristics of the specific part. For example Lowe's SIFT descriptor [35] has been shown in various studies to perform very well particularly at tasks where one is looking for identical points in different images. These methods are difficult to apply for the classification of object category, as the objects in different images may not be the same object.

2.4.2 Recognition of object categories

Recognizing categories of objects in ordinary color images is a challenging problem. Images may contain many different objects, from different viewpoints, and in different arrangements. There are several appearance-based approaches for object class recognition. Previously these methods mostly characterize the object based on the whole image. They are not robust enough to object occlusion or variation. To overcome these problems, object recognition based on some local features are proposed. In these methods the objects are represented by a number of key points or regions.

The collection of representative regions is a powerful representation of object category. In [38], the salient regions over both location and scale are used to model the objects. Shape, appearance, occlusion and relative scale are all represented with a probabilistic model. An entropy-based feature detector is used to select regions and their scale within the image. The expectation-maximization in a maximum-likelihood setting is used to estimated the parameters of the model and Bayesian model is used to classify the images. In [39] the prior knowledge of unrelated categories are incorporated to reduce the number of training images. However, the high computation costs of the joined estimation limits their methods to a small number of object parts. In [40], a class of distinguished regions based on detecting the most salient convex local arrangements of contours in the image is introduced. The regions capture shape which are invariant to scale changes and rotations, and robust against clutter, occlusions and spurious edge detections.

The spatial relationship between the regions is an effective feature for the classification of object categories. As attributed relational graph (ARG) can represent and compare entities and relationships as parts of a global structure that captures mutual dependencies, the representation of object category can be a kind of ARG. So that the ARG matching algorithm naturally follows as the key procedure for the similarity matching of the model of object category. In [41] the random attributed relational graph (RARG) based on the traditional ARG is developed for the detection of part-based static concept (object and scene). The model attaches the ARG with random variables, which are used to capture the statistics of part appearance features and part relational features. RARG can be learned from the training images in an unsupervised way.

Regarding learning of image similarity, there has been much work on similarity learning based on graph-based presentation. The definition of similarity between two graphs mainly includes two types of approaches. One is transformation based similarity definition. Most of the prior work is based on computing the cost of transformation. Typical examples include string or graph edit distance, Earth Mover's Distance (EMD) [3], etc. In [42], EMD is modified as nested EMD (nEMD) to measure the similarity between two graphs. This distance is applied to a MPEG-7 shape descriptor, Perceptual 3D Shape descriptor [43].

The other is probabilistic method for matching the vertices of ARGs. For example, Bayesian methods have been proposed and extended for matching structural or labeled graphs in [41]. The similarity between graphs is defined as the probability ratio (also known as odds) of whether or not one ARG is transformed from the other. The relationships between the nodes of model and images form an association graph. The association graph is used to define an undirected graphical model - Markov Random Field (MRF) - for the computing of the probability ratio. As proved in the paper, the probability ratio is related to the partition function of MRF. Then the partition function or its approximation is calculated to obtain the ratio.

In summary, all these approaches rely on appearance or spatial information of the objects. As we cannot manually describe different object classes with reliable criterion, we have to learn the corresponding characteristic from image data.

2.5 Handling large multimedia database

With the availability of the Internet and the reduction in price of digital cameras and camcorders, we are experiencing a high increase in the amount of multimedia information. The need to efficiently analyze and manage the multimedia data becomes essential. In order to make use of the vast amount of multimedia data, efficient techniques to analyze and organize multimedia information based on its content need to be developed for various related applications. Here we focus on the machine learning area. Usually, machine learning approaches are suitable for the modest-sized datasets. Larger-sized datasets can result in new problems and difficulties.

2.5.1 Multimedia data sampling

There are various algorithms for data sampling, such as the most simple method, simple random sampling (SRS) [44, 45]. SRS is fast and easy to implement. The use of a simple, random sample may, however, lead to unsatisfactory results. The problem is that such a sample may not adequately represent the entire dataset due to random fluctuations in the sampling process. This difficulty is particularly apparent when small sample sizes are needed [1].

Theoretical analysis and practical experience have shown that a classifier can often be built from fewer instances if the learning algorithm is allowed to build some artificial instances to help the learning. *Membership queries* to be labeled by domain expert, are also helpful [46, 47]. A membership query returns some information: whether the queried element is a member of the unknown set. For example, kernel classifiers such

as support vector machines or Bayesian kernel classifiers classify data using the most informative data instances (support vectors). This makes them natural candidates for instance selection procedures. However, like most machine learning algorithms, they are generally trained with a randomly selected classified training set that is classified in advance. Active selection of instances can significantly improve the generalization performance of a training algorithm.

Active learning identifies a subset of the input data used in the learning modelling, as the learning algorithm assumes some control over the training subset of the data [48], that is, an algorithm for choosing which instances to request for subsequent training. It has been proposed in various algorithms [49, 50, 51]. In many settings, pool-based active learning [52] is used. Instead of using a randomly selected training set, the learner has access to a pool of unlabeled instances and can request the labels for some of them. In [53] an algorithm based on "version space" for performing active learning with support vector machines is introduced. Uncertainty sampling [54, 55] is another approach for active learning. This iteratively requests class labels for training instances whose classes are uncertain, despite the previous labeled instances. In [56], an active learning method that uses adaptive resampling in a natural way to significantly reduce the size of required labeled set is proposed.

The learning-curve sampling method [57] is an approach for applying machine learning algorithms to large datasets. This method is based on the observation that the computational cost for training a model increases when the sample size of training data is increased, whereas the accuracy of the model does not improve so much. Therefore, this method tries to find the trade off between training cost and accuracy. It monitors the increasing cost and performance when larger and larger amounts of data are used for training. When future cost outweighs future benefit, the training is terminated.

Distributed and/or parallel learning has also been used to efficiently handle very large datasets. There has been extensive research on clustering and it has been applied to many

domains. Previous work mainly focuses on how to divide the dataset and organize the results of subsets efficiently. Bagging [58] is a technique that uses repeated random samples of dataset, therefore the sum of cardinalities of subsets is greater than the total size of the initial dataset. In addition, in [59] a more intelligent way of partitioning into disjoint subsets using clustering is proposed. After that, bagging technology is used to combine the results. Clustering technique groups similar data together instead of choosing elements randomly. This approach attempts to choose "similar" elements for a partition.

In [1] EASE (Epsilon Approximation Sampling Enabled) is proposed to output a sample set from the original dataset. It starts with a relatively large simple random sample of transactions and iteratively halves this sample to create a final subsample whose "distance" from the complete database is as small as possible. For computational efficiency, it defines the subsample as *close* to the original database if the high-level aggregates of the subsample normalized by the total number of data points are close to the normalized aggregates in the database. These normalized aggregates typically correspond to 1-itemset or 2-itemset supports in the association-rule setting, or, in the setting of a contingency table, to relative marginal or cell frequencies. The key innovation of EASE lies in the method by which the final subsample is obtained. Unlike FAST [60], which obtains the final subsample by trimming away outliers in a process of quasigreedy descent, EASE uses an approximation method to obtain the final subsample by a process of repeated halving. EASE provides a guaranteed upper bound on the distance between the initial sample and final subsample. In addition, EASE can process transactions on-the-fly, that is, a transaction is examined only once to determine whether it belongs to the final subsample. Moreover, the average time needed to process a transaction is proportional to the number of items in that transaction. EASE leads to much better estimation of frequencies than SRS. Experiments in the context of both association-rule

mining and classical contingency-table analysis indicate that EASE outperforms both FAST and SRS.

2.5.2 Noise in multimedia database

In machine learning, the problem of noise cleansing has attracted much attention. For a noisy database, SRS cannot effectively remove noise, as it can only randomly select some samples. The percentage of noise in the sample set remains the same as in the original set. These problems also exist in other traditional sampling algorithms. For example, in stratified sampling, the original dataset is divided into mutually disjoint parts called *strata*. A stratified sample is obtained by applying SRS over each stratum. So, the problems that existed with SRS also remain for stratified sampling. Similarly, another example is cluster sampling. In cluster sampling, the original dataset is grouped into mutually disjoint clusters. Then SRS is applied to each cluster. Considering that these clusters are created based on some database features (e.g., each page in the database is considered a cluster and SRS is applied over each page to get a sample), the problems that existed with SRS continue to remain in the cluster samples.

Many inductive learning algorithms have a mechanism to handle noise in the training samples [61]. For a noise elimination algorithm, there are normally two opposite approaches to achieve the noise cleansing procedure: selecting "good" examples or deleting "bad" examples. To distinguish "bad" instances from normal cases, various strategies have been designed. Among them, the most general techniques are motivated by pruning the decision tree to remove the mislabeled samples. Pruning on a decision tree is designed to reduce the chances that the tree is overfitting to noise. To efficiently remove some noisy data, some outlier preprocessing schemes are employed. To handle class noise from large and distributed datasets, a partitioning filter (PF) was proposed in [62], where noise classifiers learned from small subsets are integrated together to identify noisy examples.

These sampling approaches are usually used for text classification mainly and not applied in multimedia area, to the best of our knowledge. Besides, the procedure of training set selection is dependent on the process of classification. In our work, we focus on how to apply sampling algorithm for efficient training set selection which is separated from the classification process.

2.6 Summary

In this chapter, some image retrieval and object classification related issues are reviewed. MPEG-7 standard offers rich set of descriptors and enables the needed efficient and effective access (search, filtering and browsing) to multimedia content. MPEG-7 visual descriptors are outlined in this chapter, including color, texture, shape and some other descriptors. CBIR system should also involve more rich and efficient image descriptions relating to semantics, scene properties and object recognition. It is related to the development of technologies about image indexing, browsing and classification. The local properties such as points and regions are powerful tools for the object based image retrieval and classification. To deal with the large multimedia database, sampling is an effective method to help find the representative sample set.

Chapter 3

Image retrieval based on MPEG-7 visual descriptors

As MPEG-7 provides a rich set of description tools to describe the image content, image retrieval based on MPEG-7 visual descriptors is a promising approach to obtain better retrieval performance. MPEG-7 standard also gives maximum flexibility to various applications, as the extraction and application of these descriptions are not standardized. In this chapter, we address the issue on how to improve the CBIR efficiency based on MPEG-7 visual descriptors, which are extracted from whole images. Firstly, we propose the use of Earth Mover's Distance (EMD) as the distance measure of Dominant Color Descriptor (DCD), and compare the relative advantages and disadvantages of two index approaches, EMD lower bound and M-tree. After that, an optimization model of image retrieval based on multiple descriptors is introduced. It combines several descriptors using optimal weights sum function to obtain combination results.

3.1 The application of EMD for the distance computation of DCD

In this section, a new distance measure to calculate the similarity of Dominant Color Descriptor is discussed. Using EMD, better retrieval results can be obtained, compared with those obtained from the original MPEG-7 reference software - eXperiment Model

(XM) [4]. XM software is the simulation platform for the MPEG-7 descriptors, description schemes, coding schemes and description definition language. In addition to the normative components, the simulation platform has also some non-normative components, essentially to execute some procedural code to be executed on the data structures. The data structures and the procedural codes together form the applications, i.e., the extraction and comparison of descriptors.

3.1.1 Dominant Color Descriptor and its distance computation

Dominant Color Descriptor is a color descriptor that describes the dominant colors of whole image or any arbitrary shaped region. It provides an effective, compact and intuitive description of the dominant colors in an image or region [18]. It can specify a small number of representative color values and their statistical properties including percentage and variance. Based on single or several dominant color values, users can efficiently browse image database or retrieve similar images. DCD consists of the Color Index (c_i), Percentage (p_i), Color Variance (v_i), and Spatial Coherency (SC, s). The last two parameters are optional. Then the DCD is defined by:

$$DCD = \{(c_i, p_i, v_i), s\}, i = 1, ..., N \qquad \text{(Eq. 3.1)}$$

where N is the number of the colors and $\sum_{i=1}^{N} p_i = 1$. The maximum N is eight. There is one overall Spatial Coherency (s) value for the whole image, and several groups of (c_i, p_i, v_i) for the corresponding dominant colors, which can be used to compute the visual difference between images based on color.

In [4], some informative examples that illustrate the extraction of descriptions from multimedia content are provided, which are non-normative and optional in MPEG-7. For DCD, these representative colors are normally obtained by clustering colors into a small number of representative colors. After that, color quantization is applied to obtain

color values. Unlike the traditional histogram based descriptors, the representative colors of DCD are computed from each image instead of being fixed in the color space, thus allowing the color representation to be accurate and compact.

As described in [4], the dominant colors are extracted as a result of successive divisions of the color clusters with the generalized Lloyd algorithm (GLA) algorithm in each division and then merging of the color clusters. The details of clustering procedure is as follows. At the beginning, the cluster is initialized with one cluster consisting of all pixels, and one representative color computed as the centroid (center of cluster) of the cluster. After that, a sequence of centroid calculation and clustering steps are repeated, until a stopping criterion (minimum distortion or maximum number of iterations) is reached. The cluster with highest distortion is split by adding perturbation vectors to the centroid, until the maximum distortion reduces below a predefined threshold, or the maximum number of clusters is generated. The percentage of pixels in each cluster of the image is then quantized to five bits as the percentage value in descriptor.

Consider the two DCDs, $F_1 = \{(c_{1i}, p_{1i}, v_{1i}), s_1\}, i = 1, ..., N_1$ and $F_2 = \{(c_{2j}, p_{2j}, v_{2j}), s_2\}$, $j = 1, ..., N_2$. The distance between the two images with respect to DCDs in MPEG-7 XM software is defined by [4]:

$$\begin{aligned} &Dist = W_1 * SC\_Diff * DC\_Diff + W_2 * DC\_Diff \\ &W_1 + W_2 = 1 \end{aligned} \qquad \text{(Eq. 3.2)}$$

where DC_Diff is the difference between two set of dominant colors (calculated by c_i and p_i as in Eq. 3.3) and $SC\_Diff = abs(s_1 - s_2)$. W_1 and W_2 are fixed weights with recommended settings of 0.3 and 0.7, respectively. Set W_1 to 0 if Spatial Coherency is not available. Note that this distance function is non-normative part in MPEG-7, i.e., a suggestion only.

DC_Diff can be computed by the following distance function (ignoring optional Color Variance v_i):

$$DC\_Diff^2(F_1, F_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2a_{1i,2j}p_{1i}p_{2j} \qquad \text{(Eq. 3.3)}$$

where

$$a_{k,l} = \begin{cases} 1 - d_{k,l}/d_{max} & d_{k,l} \le T_d \\ 0 & d_{k,l} > T_d \end{cases} \qquad d_{k,l} = ||c_k - c_l|| \qquad \text{(Eq. 3.4)}$$

Here $a_{k,l}$ is the similarity coefficient between two colors c_k and c_l. T_d is the maximum distance for two colors which are considered similar, and $d_{max} = \alpha T_d$. In CIE-LUV color space T_d is usually between 10 to 20 and α is usually from 1.0 to 1.5. Each c_i has three elements $c_{i,0}$, $c_{i,1}$ and $c_{i,2}$.

If Color Variance is present, the following distance can be used as the similarity measure.

$$D_V^2(F_1, F_2) = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} p_{1i}p_{1j}f_{1i1j} + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} p_{2i}p_{2j}f_{2i2j} - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2p_{1i}p_{2j}f_{1i2j} \qquad \text{(Eq. 3.5)}$$

where

$$f_{xiyj} = \frac{1}{2\pi\sqrt{v_{xiyjl}v_{xiyju}v_{xiyjv}}} \exp\left[-\left(\frac{c_{xiyjl}}{v_{xiyjl}} + \frac{c_{xiyju}}{v_{xiyjv}} + \frac{c_{xiyjv}}{v_{xiyjv}}\right)/2\right] \qquad \text{(Eq. 3.6)}$$

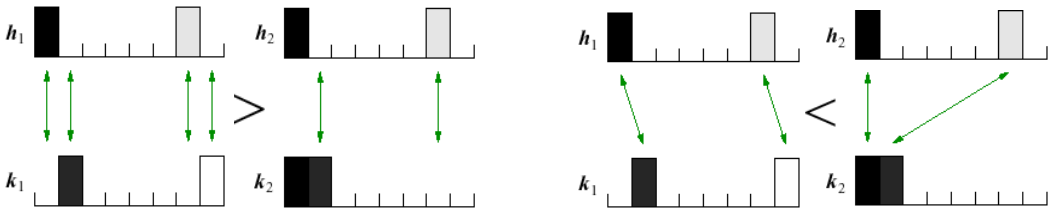
and

$$c_{xiyjl} = (c_{xil} - c_{yjl})^2, v_{xiyjl} = (v_{xil} + v_{yjl}) \qquad \text{(Eq. 3.7)}$$

In the above equations, c_{xil} and v_{xil} are dominant color values and corresponding color variances, respectively.

3.1.2 EMD's computation and application

The Earth Mover's Distance is a perceptual flexible similarity measure between two weighted multi-dimensional distributions. EMD is defined as the minimum work needed to transform from one distribution to the other. The computing of EMD is based on a



3.1a: L_1 distance $\sum_i |x_i - y_i|$ 3.1b: EMD

Figure 3.1: Examples where the L_1 distance (as a representative of bin-by-bin dissimilarity measures) do not match perceptual dissimilarity and the desired correspondences (EMD) [3].

solution to the well-known *transportation problem* [63]. EMD is shown to be a better similarity measure than many other similarity measures for two distributions of mass in a space that is itself endowed with a ground distance. EMD lifts the distance from individual features to full distributions [3]. Figure 3.1 demonstrates the characteristic of EMD. It can be seen that the two histograms h_1 and k_1 are almost the same except for a shift by one bin, but their L_1 distance between them is larger than between the h_2 and k_2. As EMD is a perceptual metric, it will calculate the distance of corresponding bins. So the EMD between h_1 and k_1 is less than EMD between h_2 and k_2.

Formally, let $P = \{(p_1, \omega_{p_1}), \ldots, (p_m, \omega_{p_m})\}$ be the first distribution with m feature vectors, where p_i is the values of features and ω_{p_i} is the weights of the corresponding p_i; $Q = \{(q_1, \omega_{q_1}), \ldots, (q_n, \omega_{q_n})\}$ is the second distribution with n feature vectors, where q_j is the values of features and ω_{q_j} is the weights of the corresponding q_j; and $D = [d_{ij}]$ is the ground distance matrix, where $d_{ij} = d(p_i, q_j)$ is the distance between p_i and q_j. The EMD between distributions P and Q is then defined as:

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d(p_i, q_j) f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \qquad \text{(Eq. 3.8)}$$

where $F = [f_{ij}]$ with $f_{ij} > 0$ is the flow between p_i and q_j. It is the optimal admissible flow

from P to Q that minimizes numerator of Eq. 3.8 subject to the following constraints:

$$\begin{aligned} &\sum_{i=1}^{m} f_{ij} < \omega_{p_i}, 1 \le i \le m \\ &\sum_{j=1}^{n} f_{ij} < \omega_{q_j}, 1 \le j \le n \\ &\sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} = min(\sum_{i=1}^{n} \omega_{p_i}, \sum_{j=1}^{m} \omega_{q_j}) \end{aligned} \qquad \text{(Eq. 3.9)}$$

f_{ij} can be solved efficiently by Simplex method as the transportation problem. Its initial basic feasible solution is computed by Russell's method. For EMD, more details can be found in [3].

3.1.3 Applying EMD to DCD

As described in section 2.2, MPEG-7 standard does not standardized the approach of similarity matching of the descriptors. Therefore, in this section, we apply a new similarity measure, the Earth Mover's Distance, to the Dominant Color Descriptor to improve the performance of similarity comparison.

Since the EMD can be used to calculate the distance between two multi-dimensional distributions where each distribution is represented by sets of weighted features, it is appropriate to use the EMD to determine the similarity between two DCDs, where the basic elements are the dominant colors and their corresponding percentages. Applying EMD to DCD, the EMD distance between two DCDs $F_1 = \{(c_{1i}, p_{1i})\}, i = 1, ..., N_1$ and $F_2 = \{(c_{2j}, p_{2j})\}, j = 1, ..., N_2$ is shown in Eq. 3.10. Here we omit the optional Color Variance v_i and Spatial Coherency s.

$$DC\_Diff(EMD) = \frac{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} d(c_{1i}, c_{2j}) f_{ij}}{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} f_{ij}} \qquad \text{(Eq. 3.10)}$$

where $d(c_{1i}, c_{2j})$ is the ground distance between c_{1i} and c_{2j}. In this work, the Euclidean distance in the CIE-LUV color space was used as the metric to compute the ground distance in EMD. $F = [f_{ij}]$ with $f_{ij} > 0$ is the flow between c_{1i} and c_{2j}. It is the optimal admissible flow from P to Q that minimizes numerator of Eq. 3.10 subject to the same constraints as in 3.1.2.

$$\begin{aligned} &\sum_{i=1}^{N_1} f_{ij} < p_{1i}, 1 \leq i \leq N_1 \\ &\sum_{j=1}^{N_2} f_{ij} < p_{2j}, 1 \leq j \leq N_2 \\ &\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} f_{ij} = min(\sum_{i=1}^{N_1} p_{1i}, \sum_{j=1}^{N_2} p_{2j}) \end{aligned} \qquad \text{(Eq. 3.11)}$$

f_{ij} can be solved efficiently by Simplex method as the transportation problem and its initial basic feasible solution is computed by Russell's method.

When the Color Variance parameters are used, EMD cannot be used directly. This is because Color Variance presents some other visual information unlike color index value and needs specific similarity measures. How to use color variance for EMD may be future work.

3.1.4 Analysis of retrieval complexity

EMD is accurate as a similarity measure, but its computation is still complex. The computation time for random distributions as a function of the number of feature vectors in the distribution is shown in [3]. Here we propose two methods for speeding up the retrieval process, one based on the lower bound of the EMD and the other based on M-tree index.

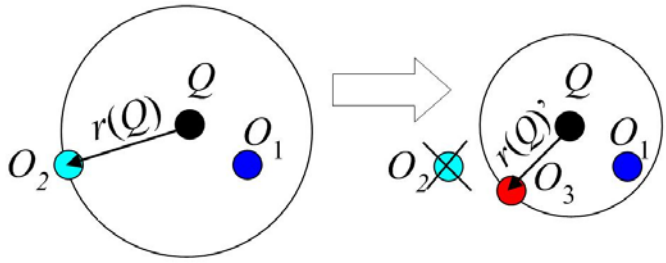


Figure 3.2: The concept of query radius. For a top 2 query, $r(Q)$ is the initial query radius, which contains object O_1 and O_2. $r(Q)'$ is the new query radius after object O_3 coming. $r(Q)' < r(Q)$.

3.1.4.1 Lower bound

For EMD, with equal total weights for distributions and the ground distance is induced by a norm, it has an easy-to-compute lower bound. When an image from database is coming, firstly the lower bound between the new image and query image is calculated. If the lower bound is larger than the current query radius, the true EMD need not to be calculated. These bounds can significantly reduce the number of EMDs that actually calculated by pre-filtering the database and ignoring images which are too far from the query image. The lower bound is the distance between the centroid of distributions. Figure 3.2 shows the concept of query radius. The whole retrieval procedure with lower bound is shown in Figure 3.3.

The computation of lower bound is as follows. It is defined as the distance between their centroid [3]. Given the two distributions $P = (P_i, \omega_{p_i})$ and $Q = (Q_j, \omega_{q_j})$, where ω_{p_i} and ω_{q_j} are the percentages of each feature; P_i and Q_j are the corresponding feature value. Then the corresponding lower bound can be calculated as follows:

$$EMD(P,Q) \geq \|\overline{P} - \overline{Q}\| \qquad \text{(Eq. 3.12)}$$

$$\overline{P} = \frac{\sum_{i=1}^{m} \omega_{p_i} P_i}{\sum_{i=1}^{m} \omega_{p_i}}, \qquad \overline{Q} = \frac{\sum_{j=1}^{n} \omega_{q_j} Q_j}{\sum_{j=1}^{n} \omega_{q_j}} \qquad \text{(Eq. 3.13)}$$

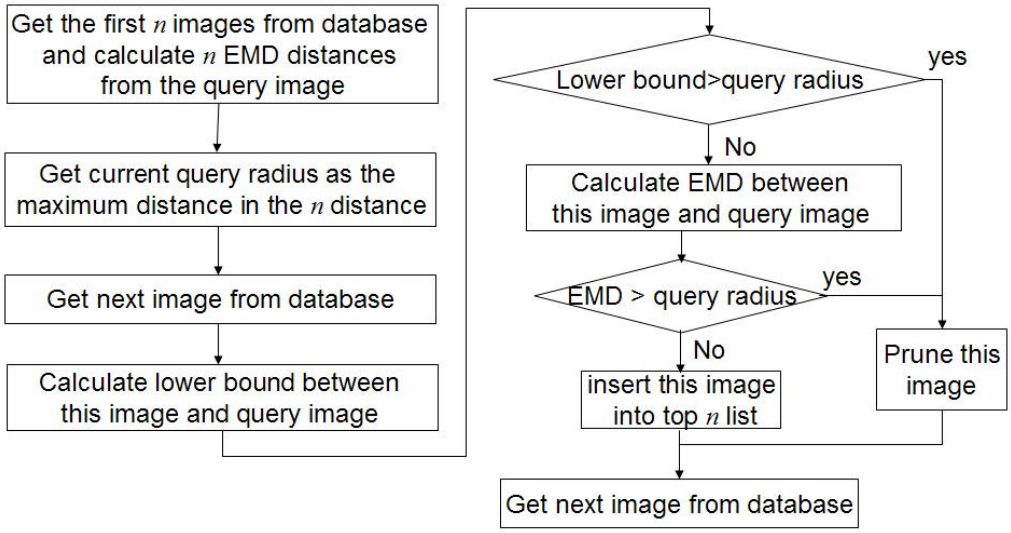


Figure 3.3: The procedure of retrieval with lower bound.

3.1.4.2 Introduction of M-tree

M-tree was proposed as a paged dynamic structure based on metric space [64]. It is used to index multimedia databases where objects have complex features which, consequently, result in time-consuming distance computations. Users can define arbitrary metric distances to compare the objects. This is more general than those based on vector spaces, such as R-tree and R*-tree. A space A is called a metric space if for any of its two elements x and y, there is a function $d(x, y)$, called the distance that satisfies the following properties: [64]

$d(x, y) \geq 0$ (Non-negativity)

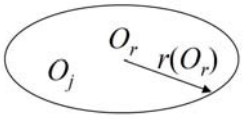
$d(x, y) = 0$ if and only if $x = y$ (identity)

$d(x, y) = d(y, x)$ (Symmetry)

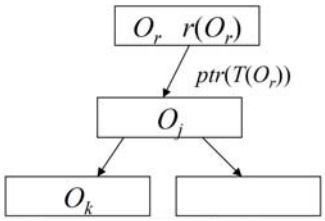
$d(x, z) \leq d(x, y) + d(y, z)$ (Triangle inequality)

When building up the M-tree, both the I/O access cost and the distance computation cost are considered [64]. When the features of similarity queries are multi-dimensional, the distance function can be very complex. The computational cost of distance function is not trivial in the real application and cannot be ignored. M-tree is suitable as an index for database with complex distance calculation. It can improve similarity queries by efficiently pruning the data space which needs to be searched, and at same time guarantee that the query results are exact [65]. This is because the actual distance function is used without any approximation although M-tree saves the distance calculation cost.

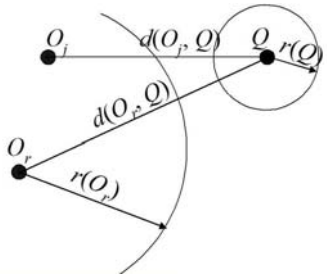
The M-tree partitions a given data search space based on relative distances between objects. Each node of a M-tree is fixed-size and stores an object covering corresponding region of the metric space (as shown in Figure 3.4). The distance function of M-tree is fully parametric as a black box and can be any type of metric distance. It need not fit into a vector space nor must use a L_p metric. Hence M-tree considerably extends the possible number of applications for which efficient queries can be achieved.



3.4a: Routing object O_r has a covering radius $r(O_r)$.



3.4b: O_r references a covering tree $T(O_r)$.



3.4c: A search rule of M-tree.

Figure 3.4: The objects and covering tree of M-tree. Fig 3.4c demonstrates one of search rules. This rule is: If $d(O_r, Q) > r(O_r) + r(Q)$, then $d(O_r, Q) > r(O_r)$.

The objects in M-tree can be classified as *ground object* and *routing object*. Object O_j is defined as a *ground object* if its entry is stored in a leaf node of the M-tree and all non-ground (internal) objects are called *routing objects* (O_r).

The following are two fundamental concepts of the M-tree design:

i. Database objects are recursively organized according to their distances from reference (routing) objects.

ii. Routing objects are database objects, which acquire their routing functions by specific promotion algorithms.

The M-tree organizes the objects into fixed-size nodes (variable-size nodes can also be used). Each M-tree node can store maximum M entries and M is the capacity of nodes. An entry for a routing object O_r is defined as $entry(O_r) = [O_r, ptr(T(O_r)), r(O_r), d(O_r, P(O_r))]$. $ptr(T(O_r))$ is a pointer which references the root of a sub-tree $T(O_r)$. $T(O_r)$ is the covering tree of O_r with a covering radius $r(O_r) > 0$. $d(O_r, P(O_r))$ is the distance from O_r to the parent object $P(O_r)$. The semantic of the covering radius $r(O_r)$ is that all the objects stored in the $T(O_r)$ are within the distance $r(O_r)$ from O_r, i.e., $\forall O_i \in T(O_r), d(O_r, O_i) \leq r(O_r)$. Therefore a routing object O_r defines a region in the metric space, centered on O_r and with radius $r(O_r)$ (see Figure 3.4).

In this way the M-tree organizes the space into a set of (possibly overlapping) regions with the same principle recursively applied. M-tree support both range query and KNN query. When the tree is built, the level at which each data entry has to be located is determined through the following process: each new descriptor entry is compared with the routing objects at each level; the tree traversing path is determined by selecting the routing object which requires the minimum increase of the covering radius to include the descriptor entry in the cluster. At the leaf level, if the leaf node selected is full (node overflow), a split of the leaf cluster is performed. The split can propagate from the leaf to the upper nodes, up to the root.

Different split policies can be used such as mM-Rad (minimum of Maximum of Radii), or MLB-Dist (Maximum Lower Bound on Distance). They differ in the way in which routing objects are promoted when a cluster split occurs. mM-Rad, promotes as routing objects the pair of entries which have the minimum of the maximum of covering radii (thus reducing the extension of covering regions). MLB-Dist, promotes the two entries that are at the greatest distance (thus reducing the overlapped area). The detail algorithms of insert, delete nodes and query based on M-tree are introduced in [64].

3.1.4.3 Building M-tree index to EMD-based computation

When the color features have equal weights such as DCD and the ground distance $d(p_i, q_j)$ in Eq. 3.8 is metric, the EMD is a true metric. Proof is given in [3]. So the DC_Diff based on EMD as Eq. 3.10 can be used as the distance function to build the M-tree. The $Dist$ in 3.1.1 cannot be used, because it is not a true metric for some values of W_1, W_2 and SC_Diff.

We use EMD as the distance to build the M-tree. In the M-tree user's guide [66], several parameters about the M-tree structure are given, including the splitting and balance policy. The M-tree is built in advance and the computation costs of building M-tree are not considered. After testing, the parameter set "0.5, Generalized_Hyperplane,

MIN_RAD" is most efficient in search and are used to build the M-tree. The meaning of these parameters are as follows [66]:

0.5: it is the minimum node utilization. It is used to guarantee a minimum fill factor for tree nodes during the split. It can assume values in the range [0, 0.5]. We use 0.5.

Generalized Hyperplane: it specifies the split policy of how to efficiently partition the given entries N into two subsets, N_1 and N_2 according to two routing objects O_{p1} and O_{p2}. This method assigns each object $O_j \in N$ to the nearest routing object.

MIN_RAD: it specifies the promotion policy is "minimum maximum radius" policy. It uses "minimum (sum of) RAD ii" algorithm which is the most complex in terms of distance computations. It considers all possible pairs of objects and, after partitioning the set of entries, promotes the pair of objects for which the sum of covering radii, $r(O_{p1}) + r(O_{p2})$, is minimum [64].

3.2 Image retrieval based on multiple descriptors

Early CBIR research has focused on just one low-level visual feature. It would be more accurate based on the combination of multiple features. In image retrieval, queries can use several features such as color, texture, shape or text. Our work concentrates on how to efficiently combine multiple MPEG-7 visual descriptors for image retrieval. Each descriptor has a suitable weight for combination. In this section, we propose a new optimization model for the proper selection of weights. The model determines the optimal weights of descriptors directly instead of the calculated distances of each descriptor. The weights are calculated according to the different values of the query descriptors and it provides a simple optimal adjustment when the query image is changed. There is no manual interaction needed, such as labelling.

3.2.1 Combination of multiple descriptors

The retrieval of images based on a single descriptor constrains the range of images that could be explored for meaningful results. Moreover, it is highly unlikely that a human querying an image database would do so on the basis of a single feature, say, color. In order to expand the scope of the query, it is important to consider how retrieval can be done on the basis of multiple descriptors. This calls for a sound and rigorous method to automatically determine the corresponding weights of each of the descriptors so that meaningful retrieval can be obtained. In this section we describe the retrieval model based on combination of descriptors through an optimization framework.

3.2.1.1 Optimization model

The underlying idea is to rank the distances between the query image and the images in the database where the distance refers to either the distance of the combined weighted descriptors, i.e., $d(w_xQ_x + w_yQ_y, w_xD_x + w_yD_y)$ or the sum of the distance of each of the descriptors, i.e., $d(w_xQ_x, w_yQ_y,) + d(w_xD_x, w_yD_y)$. Here x and y are the features. Q and D refer to the query and database images respectively. Setting up the problem in an optimization framework, the task is to minimize the distance subject to the constrains that the descriptors in the retrieved image are close to the corresponding description in the query image, where closeness is represented by normalized weights.

Firstly the definition of our notations is given. The descriptors of query image are defined as set $Q = Q_0, Q_1, ..., Q_I$, where Q_i is the ith query descriptor. The descriptors of the ideal similar images are denoted by a set $D = D_0, D_1, ..., D_I$, where each D_i represents similar descriptor of the corresponding type of Q_i. Each descriptor takes the form of a vector $Q_i = Q_{i0}, Q_{i1}, ..., Q_{iJ}$ or $D_i = D_{i0}, D_{i1}, ..., D_{iJ}$. The elements of vectors are represented by D_{ij} and Q_{ij} separately. The objective function of the optimization

model is defined as:

$$\begin{aligned} minJ &= d(Q,D) && \text{(Eq. 3.17)} \\ s.t.D_{ij} &= w_i Q_{ij}, && \text{(Eq. 3.18)} \\ \sum_{i=0}^{I} \frac{1}{w_i} &= 1, && \text{(Eq. 3.19)} \\ i &= 0,...,I, \\ j &= 0,...,J. \end{aligned}$$

where $d(Q,D)$ is the distance between the query image and the database images in terms of the descriptors Q and D. Q_{ij} represents the jth element of ith query descriptor. w_i is the similarity weight of ith descriptor as shown in Eq. 3.18. Eq. 3.19 is for scaling purpose, otherwise a trivial optimal solution which are all zero can be obtained.

To solve this optimization problem, Lagrange multipliers are used to reduce this constrained problem to an unconstrained one. Then the optimal solution of D_{ij} and w_i will be calculated. The unconstrained problem is as follows:

$$L = d(Q,D) - \sum_{i=0}^{I}\sum_{j=0}^{J} \lambda_{ij}(D_{ij} - w_i Q_{ij}) - \beta(\sum_{i=0}^{I} \frac{1}{w_i} - 1) \qquad \text{(Eq. 3.20)}$$

where λ_{ij} and β are Lagrange multipliers.

3.2.1.2 Optimal solution of w_i and D_{ij}

To obtain optimal solution of w_i and D_{ij}, the partial differentiation are calculated.

$$\begin{aligned} \frac{\partial L}{\partial D_{ij}} &= \frac{\partial d(Q,D)}{\partial D_{ij}} - \lambda_{ij} && \text{(Eq. 3.21)} \\ \frac{\partial L}{\partial w_{ij}} &= \sum_{j=0}^{J} \lambda_{ij} Q_{ij} - \beta \frac{1}{{w_i}^2} && \text{(Eq. 3.22)} \end{aligned}$$

Set the partial differentiation to zero, then the optimal λ_{ij} is obtained.

$$\lambda_{ij} = \frac{\partial d(Q,D)}{\partial D_{ij}} \qquad \text{(Eq. 3.23)}$$

According to [24], we multiply both side of Eq. 3.22 by w_i and summarize over i, then we have:

$$\sum_{i=0}^{I} w_i(\sum_{j=0}^{J} \lambda_{ij} Q_{ij}) + \beta(\sum_{i=0}^{I} \frac{1}{w_i}) = 0 \qquad \text{(Eq. 3.24)}$$

Let $h_{ij} = \sum_{j=0}^{J} \lambda_{ij} Q_{ij}$, the optimal value of β is:

$$\beta^* = -\sum_{i=0}^{I} w_i(\sum_{j=0}^{J} \lambda_{ij} Q_{ij}) = -\sum_{i=0}^{I} w_i h_i \qquad \text{(Eq. 3.25)}$$

This will lead to the optimal solution of w_i:

$$w_i^* = \sum_{i=0}^{I} \sqrt{\frac{h_j}{h_i}} \qquad \text{(Eq. 3.26)}$$

With w_i^*, we can get optimal solution of D_{ij}:

$$D_{ij} = w_i^* * Q_{ij} \qquad \text{(Eq. 3.27)}$$

These are the ideal combination of multiple descriptors for the query image. The image with these descriptors is the most similar one to the query image when multiple feature descriptors are used to query. Although the ideal similar descriptors are obtained, the images in the database may not have the same descriptors. So the retrieval results cannot be obtained. To solve this problem, we have to calculate the distance between descriptors of the images in the database and the ideal descriptors. Same optimal weights are applied. These distances are the similarity measure and they reflect the degree of similarity between these images. After the distances are sorted, we get the results in terms of degree of similarity.

3.2.2 Application of different MPEG-7 visual descriptors

The optimal method is used to combine MPEG-7 visual descriptors for image retrieval. In descriptor selections, two color descriptors and one texture descriptor are included,

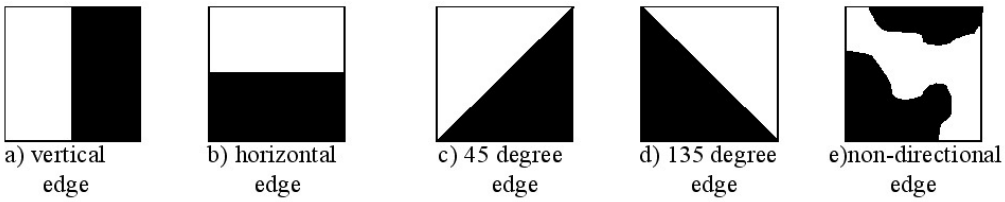


Figure 3.5: Five type of edges for EHD [2].

and they can be applied individually or used together for image retrieval. Because the images usually have non-homogeneous texture, Edge Histogram Descriptor is selected as a texture descriptor. It is designed for image retrieval with non-uniform texture. Two color descriptors in different format are selected. One is a histogram descriptor (Color Structure Descriptor); the other is of a specific format (Dominant Color Descriptor).

3.2.2.1 Introduction of EHD and CSD

Edge Histogram Descriptor (EHD) is a texture descriptor used for the region which is non-homogeneous in texture properties [19]. It is a 80-bins histogram represents local edge distribution in an image by dividing the image into $4 \times 4 = 16$ sub-images. The spatial distribution of 5 kinds of edges as shown in Figure 3.5, including vertical, horizonal, 45° diagonal, 135° diagonal and non-directional edges in each sub-image are categorized to form the vector, which has $16 \times 5 = 80$ bins. Semi-global and global edge distribution histogram can be calculated according to the local histogram. Global edge distribution is the edge distribution for whole image, i.e., the number of edge in each direction calculated in whole image space. Semi-global edge distribution are calculated based on some group of subsets of image space, as shown in Figure 3.6. When calculating the distance, combining the local, semi global and global histogram, a total of 150 bins histogram is constructed for similarity matching.

Color Structure Descriptor (CSD) describes both color content (like color histogram) and the structure of this content by using a structure element. CSD can be used to

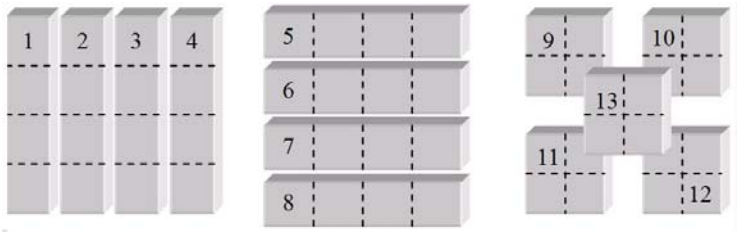
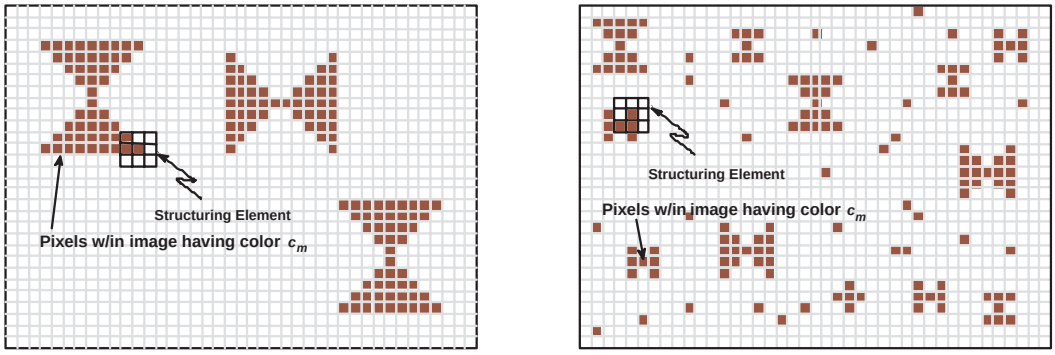


Figure 3.6: Thirteen Clusters of sub-images for semi-global histograms [4].



3.7a: Highly structured color plane.

3.7b: Unstructured color plane.

Figure 3.7: Example of structured and unstructured color.

distinguish between images which have similar traditional color histograms and different color spatial distribution. Figure 3.7 illustrates this using a pair of images [4]. Figure 3.7a is a highly structured color image and Figure 3.7b is an unstructured color image. As the number of foreground color pixels is the same, they cannot be distinguished by traditional color histograms. But using CSDs, these two images can be distinguished very clearly because the distributions of color are different.

Compared with other color descriptors, CSD has detailed color information and can achieve better retrieval results. This is proven in experiments shown in [67]. The format of CSD is identical to a color histogram, but the semantic meaning is different. It is a 1D array of eight bit-quantized values, CSD $= \overline{h_s}(m), m \in 0, 1, ..., M-1$, where s is the scale of the associated square structuring element and M is chosen from the set {256, 128, 64, 32}. So CSD can be 256-, 128-, 64- or 32-bin (array elements). To extract CSD,

an image is presented using HMMD (Hue-Max-Min-Diff) color space.

The HMMD, which is proposed in MPEG-7, is closer to a perceptually uniform color space. It is defined by a nonlinear, reversible transformation from the RGB color space. There are five distinct attributes (components) in the HMMD color space, however only three of them (Hue, Max, Min, or Hue, Diff, Sum) are sufficient to define the color space. The five attributes can be characterized as follows:

- Hue: the same as in HSV.

- Max: indicates how much black color it has, giving a flavor of shade or blackness.

- Min: indicates how much white color it has, giving a flavor of tint or whiteness.

- Diff: indicates how much gray it contains and how close to the pure color, giving a flavor of tone or colorfulness.

- Sum: simulates the brightness of the color.

The transformations for Max, Min and Hue are the same as the equations for Min, Max and Hue in HSV color space. The transformations for Diff and Sum have the following form:

Diff = Max - Min;

Sum = (Max + Min)/2;

The Max, Min and Sum components have values in the range [0,1] and the Diff component has values in the range [0,1]. The Hue component takes values in the range [0,360]. The HMMD color space has a double cone appearance as shown in Figure 3.8.

In order to extract CSD, HMMD color space is non-uniformly partitioned to 32, 64, 128 or 256 cells. The M bins of CSD are *bijective* to the M cells of HMMD. In MPEG-7, $s = 8^2$. Hence, an 8×8-sized structuring element is slid over the whole image and the numbers of positions where the element contains each quantized color are accumulated as the descriptor. In our experiment, 256-bin CSD is used. The distance functions of

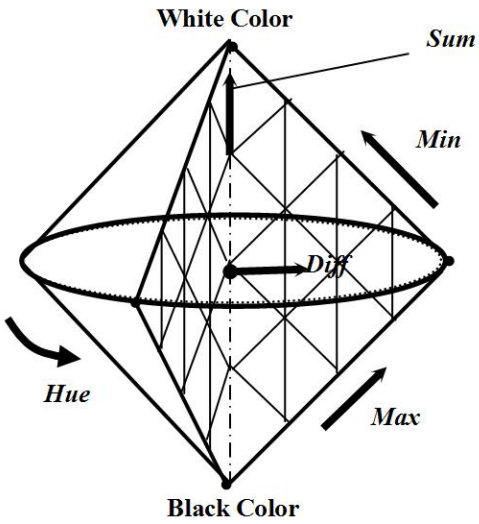


Figure 3.8: Appearance of the HMMD color space [2].

CSD and EHD suggested in MPEG-7 are both L_1-norm. It is shown as follows:

$$d(Q_i, D_i) = \sum_{j=1}^{J} |D_{ij} - Q_{ij}| \qquad \text{(Eq. 3.28)}$$

3.2.2.2 Combination of EHD and CSD

EHD and CSD are combined with the following optimal weights:

The partial derivatives of $d(Q, D)$ with respect to D_i are:

$$\frac{\partial d(Q,D)}{\partial D_{ij}} = \begin{bmatrix} \frac{\partial d}{\partial D_{i,0}} \\ \dots \\ \frac{\partial d}{\partial D_{i,J}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \sum_{j=1}^{J} |D_{ij} - Q_{ij}|}{\partial D_{i,0}} \\ \dots \\ \frac{\partial \sum_{j=1}^{J} |D_{ij} - Q_{ij}|}{\partial D_{i,J}} \end{bmatrix} \qquad \text{(Eq. 3.29)}$$

Then the optimal values of λ_{ij} are obtained according to Eq. 3.23, while the optimal value of w_i and D_{ij} are obtained as described in section 3.2.1.2. For histogram descriptor, such as CSD and EHD, their corresponding histogram bins are directly combined using the optimal weights. Because their dimensions are a little different, zeros are added to

extend the short one. For all the images in database, we can use this optimal weight to combine these two descriptors. Then L_1-norm is used to calculate the distance between the new combined descriptor of query image and image in database. After the distance sorting, the ranked list will be the retrieval results.

3.2.2.3 Combination of EHD and DCD

EHD (D_0) and DCD (D_1) are combined as another example. The partial derivatives of EHD are calculated by Eq. 3.29 as in previous section. The details of DCD is described in section 3.1.1. In section 3.1.3 the Earth mover's distance is applied as the similarity distance of DCD. As EMD is based on a solution to the transportation problem, the partial derivatives of EMD is difficult to calculate. Thus we have to use the original distance function in in MPEG-7 eXperiment Model (XM) to compute the distance between DCDs. Consider the two DCDs, $D_1 = \{c_{D_1 i}, p_{D_1 i}\}, i = 1, ..., N_{D_1}$ and $Q_1 = \{c_{Q_1 i}, p_{Q_1 i}\}, i = 1, ..., N_{Q_1}$. The distance between the two images with respect to DCDs in MPEG-7 eXperiment Model (XM) is defined by (ignoring optional v_i and s): [2]

$$d^2(Q_1, D_1) = \sum_{i=1}^{N_{D_1}} p_{D_1 i}^2 + \sum_{j=1}^{N_{Q_1}} p_{Q_1 j}^2 - \sum_{i=1}^{N_{D_1}} \sum_{j=1}^{N_{Q_1}} 2 a_{D_1 i, Q_1 j} \qquad \text{(Eq. 3.30)}$$

where

$$a_{k,l} = \begin{cases} 1 - dist_{k,l}/d_{max} & dist_{k,l} \leq T_d \\ 0 & dist_{k,l} > T_d \end{cases} \qquad \text{(Eq. 3.31)}$$

T_d is the maximum distance for two colors which are considered similar and $d_{max} = \alpha T_d$. In CIE-LUV color space T_d is usually between 10 to 20 and α is usually from 1.0 to 1.5. Each c_i has three elements $c_{i,0}$, $c_{i,1}$ and $c_{i,2}$. The partial derivatives of $d(Q, D)$ with

respect to DCD are:

$$\frac{\partial d(Q,D)}{\partial D_{ij}} = \begin{bmatrix} \frac{\partial d}{\partial p_{D_1 i}} \\ \dots \\ \frac{\partial d}{\partial D_{i,J}} \end{bmatrix} = \begin{bmatrix} \frac{(2p_{D_1 i} - \sum_{j=1}^{N_{Q_1}} ap_{D_1 i})}{\partial D_{i,0}} \\ \dots \\ \frac{\partial \sum_{j=1}^{J} |D_{ij} - Q_{ij}|}{\partial D_{i,J}} \end{bmatrix} \quad \text{(Eq. 3.32)}$$

Because DCD has special format and distance function, it is inefficient to combine the element of EHD and DCD directly. We have to calculate the distances of DCD and EHD separately according to the descriptors with the optimal weights and then we sum up the two distances as the similarity measure.

More descriptors can also be combined using this optimization model. Furthermore, it is not for MPEG-7 visual descriptors only and can be used to other feature vectors.

3.3 Experimental results

The experimental results of image retrieval based on one and several descriptors are described in this section. At the beginning, we introduce the test set and evaluation measure of MPEG-7, after that the detail results are presented.

3.3.1 Effectiveness evaluation of MPEG-7: CCD, CCQ and ANMRR

MPEG-7 provides a common data set and a common set of queries for experiments to test the effect of color descriptors. There are a total of seven subsets containing about 5,000 images in the Common Color Dataset (CCD). Among them subset 2 includes 2045 images in ppm format. About 50 Common Color Query (CCQs) have been defined as a query image with specified ground truth images. The entire 9 CCQs in subset 2 are used to test the color descriptors. The details of CCD and CCQ can be found in [68].

Unlike the Recall, Precisions and Fallout, a new retrieval accuracy measure Average Normalized Modified Retrieval Rate (ANMRR) is used in MPEG-7. ANMRR gives the formula for computing the retrieval accuracy values. The objective is to determine how many correct images are retrieved and how high they are ranked among the retrieval results. The retrieval accuracy measures are computed as follows: [69]

- Let the number of ground truth images for a query q be $NG(q)$
- Compute $NR(q)$, number of found items in first K retrievals (the top ranked K retrievals), where
- $K = \min(4 * NG(q), 2 * GTM)$, where GTM is $\max\{NG(q)\}$ for all q's of a data set.
- Compute $MR(q) = NG(q) - NR(q)$, number of missed items
- Compute from the ranks $Rank(k)$ of the found items counting the rank of the first retrieved item as one.
- A Rank of $(K * 1.25)$ is assigned to each of the ground truth images which are not in the first K retrievals.
- Compute $AVR(q)$ for query q as follows:

$$AVR(q) = \sum_{k=1}^{NG(q)} \frac{Rank(k)}{NG(q)} \qquad \text{(Eq. 3.33)}$$

- Compute the modified retrieval rank as follows:

$$MRR(q) = AVR(q - 0.5 - \frac{NG(q)}{2}) \qquad \text{(Eq. 3.34)}$$

- Compute the Retrieval Rate as follows:

$$RR(q) = \frac{NR(q)}{NG(q)} \qquad \text{(Eq. 3.35)}$$

- Compute the normalized modified retrieval rank as follows:

$$NMRR(q) = \frac{MRR(q)}{K * 1.25 - 0.5 * (NG(q) + 1)}) \quad \text{(Eq. 3.36)}$$

  Note that the $NMRR(q)$ will always be in the range of [0.0,1.0].

- Compute average of NMRR over all queries

$$ANMRR(q) = \frac{1}{Q} \sum_{q=1}^{Q} NMRR(q)) \quad \text{(Eq. 3.37)}$$

- Provide numbers $NR(q)$, $MR(q)$, $RR(q)$, $AVR(q)$, $MRR(q)$, $NMRR(q)$ for each query, and the average of $ANMRR$ over whole set of queries

If more than one retrieved (ground truth) images have the same similarity measure value, then the rank value assigned to that image is the average ranks of all the retrievals that have the same similarity value.

3.3.2 Image retrieval based on EMD

3.3.2.1 Retrieval results based on EMD

Here ANMRR is used to evaluate the retrieval quality of two different methods. The smaller the ANMRR value, the better the retrieval effect is. The test set is subset 2 of MPEG-7 Common Color Dataset (CCD), including 2045 images in ppm format. Now the entire 9 Common Color Query (CCQ) in subset 2 is used to test the color descriptors. HSV color space is used to store the descriptor and CIE-LUV color space is used to calculate the distance. Before calculating the EMD, all color values of different color space are transformed into the CIE-LUV color space. Therefore which color space is used to store the descriptor is not important. The test results of Dominant Color Descriptor are as follows:

ANMRR results for Dominant Color Descriptor:

XM without Spatial Coherency: ANMRR = 0.2604

EMD without Spatial Coherency: ANMRR = 0.2019

EMD with Spatial Coherency: ANMRR = 0.1914

In [18], the entire MPEG-7 CCD generates an ANMRR of 0.252 (without Spatial Coherency and Color Variance). When we calculate the ANMRR on a subset of 2045 images, we get a value of 0.2604 (XM) which is comparable. It can be seen that a significant improvement can be achieved by using EMD to calculate the distance. The CCQs used for testing are shown in Figure 3.9. Table 3.1 shows an example of calculation procedure of ANMRR.

Figure 3.10 shows a ground truth set of CCQ. The first image is the query image. In Figure 3.11 the top 12 retrieval results by using EMD with SC are shown. It can be seen that the results are very good.

3.3.2.2 Comparison of M-tree and lower bound

As described in section 3.1.4, both M-tree and lower bound are applied to reduce the computation time of EMD. We compare the effect of M-tree and lower bound based on two aspects, the computation complexity of EMD and the runtime for each of the methods. All experiments are based on the CCD subset 2 with 2045 images described in section 3.3.1.

Table 3.2 shows for the five different indexing methods, the average number of EMD computations per query according to the number of top images retrieved. All 2045 images are used as the query image and the table was generated by averaging all the queries.

For M-tree, many parameter sets are tested. Each dominant color forms a 4-dimension vector because it has a percentage value and 3 color index values. To compare the effect of different dimensionality, several dominant colors, including 2-color DCD (8-dimension), 5-color DCD (20-dimension) and the full-color DCD (32-dimension), are used as the

3.9a: Ron Reagan, 9 images

3.9b: Landscape Image 2, 4 images

3.9c: Landscape Image 3, 3 images

3.9d: Containers, 4 images

3.9e: Big pipes, 6 images

3.9f: Indoor Image, 12 images

3.9g: People on the red, 5 images

3.9h: Speaker, 6 images

3.9i: Sunset over lake, 8 images

Figure 3.9: The details of CCQ images in a subset of CCDs. These images are the query images of corresponding CCQs.

index. 2-color DCD selects the two colors with the largest percentage to calculate the distance. 5-color DCD uses the top five dominant colors. Full-color DCD uses the original descriptor. For lower bound, calculating EMD with and without SC of DCD are tested separately.

Table 3.2a shows that the results of M-tree are not as good as lower bound, especially when the dimensionality is high. M-tree can give the exact query results. The results of 2-color DCD and 5-color DCD are approximate because we only use two and five colors of the total DCD. The results of lower bound method are shown in Table 3.2b. This lower bound guarantees that no image is wrongly missed as a result of saving computation. It

Figure 3.10: A whole CCQ with eight ground truth images (sunset over lake).

can be seen that the lower bound can prune at least half of the images and it is therefore very effective. Even without considering Spatial Coherency, the retrieval results are satisfied. When the Spatial Coherency is used, the result is a little better. This is because the SC increases the distance among the EMD distances.

Figure 3.12 shows the average runtime of both the M-tree and lower bound method with the best results and the sequential search is also added for comparison. It can be seen that the lower bound method is much faster than the sequential search. For M-tree, when the dimension is less than 10, its performance is acceptable but still slower than the lower bound method.

3.3.3 Image retrieval using combined descriptors

Because MPEG-7 has no ground truth sets to test combined descriptors, the ground truth sets of color and texture descriptors is used to test the retrieval performance of optimal combination model, including 17 Common Color Query (CCQ) and two standard texture queries. The test sets are subset 1 and 2 of MPEG-7 Common Color Dataset (CCD), including 2045 images in ppm format and 298 images in jpg format. In MPEG-7 these sets are also used to test the effect of non-homogeneous texture descriptor. Firstly Average Normalized Modified Retrieval Rank (ANMRR) is used to evaluate the retrieval

3.11a: 0.000000 3.11b: 11.172273 3.11c: 15.872145 3.11d: 16.767384

3.11e: 17.252483 3.11f: 19.320299 3.11g: 19.446711 3.11h: 19.559317

3.11i: 19.694727 3.11j: 19.909126 3.11k: 20.071930 3.11l: 20.205475

Figure 3.11: The top 12 retrieval results of EMD (with SC). The numbers under the images are the corresponding distances.

quality of different methods. Table 3.3 shows the ANMRR value of the different methods. Average combination means we only average the two distances of the corresponding descriptors as the similarity measure. It can be seen that a significant improvement is achieved by using the optimal combination. When the descriptors are both in histogram

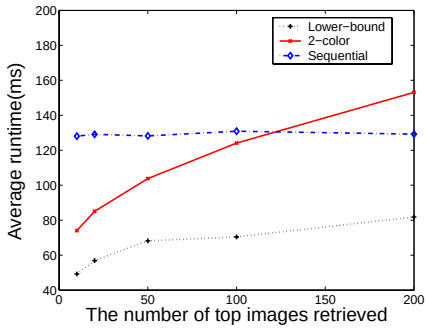


Figure 3.12: The average runtime of different methods.

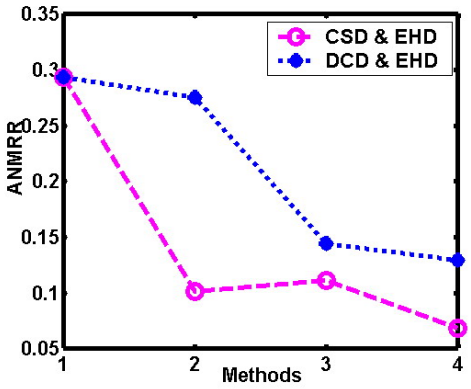


Figure 3.13: ANMRR of different combination methods. The details of methods are according to the index of Table 3.3.

format, the results are better. The combination of CSD and EHD is better than DCD and EHD. It is because, as a whole, the effect of CSD is better than DCD. But for some user queries, the results of DCD are better than CSD. In order to compare with other systems, the common evaluation measure Precision and Recall are also calculated in Table 3.3.

Figure 3.13 shows ANMRR values of different methods explicitly. The index of different methods is the same as the order in Table 3.3. It is shown that the optimal combination (method four) is better than other methods. A retrieval example is presented in Figure 3.14. It is an indoor image set and the top-left image (Figure 3.14a) is the query image. Top 12 images are shown and sorted according to the distance. It is the results of retrieval using optimal combination of CSD and EHD. Currently the optimal combination is applied for whole image database. The calculation procedure is easy and fast. If the image database is very large, single descriptor can be used to obtain the first round results, and optimal combination is used to refine the retrieval results.

3.14a: Query image

3.14b: Rank 1

3.14c: Rank 2

3.14d: Rank 3

3.14e: Rank 4

3.14f: Rank 5

3.14g: Rank 6

3.14h: Rank 7

3.14i: Rank 8

3.14j: Rank 9

3.14k: Rank 10

3.14l: Rank 11

Figure 3.14: A retrieval example using optimal combination of multiple descriptors.

Table 3.1: The calculation procedure of ANMRR.

3.1a: XM.

CLASS	CCQ 1	CCQ 2	CCQ 3	CCQ 4	CCQ 5	CCQ 6	CCQ 7	CCQ 8	CCQ 9
NG(q)	9	5	4	3	12	6	4	8	6
K(q)	24	20	16	12	24	24	16	24	24
NR(q)	9	4	4	3	11	6	1	4	3
MR(q)	0	1	0	0	1	0	3	4	3
AVR(q)	7.2222	7	5.25	2	11.0833	4	15.25	17.125	16.1667
MRR(q)	0	4	2.75	0	4.5833	0.5	12.75	12.625	12.6667
RR(q)	1	0.8	1	1	0.9167	1	0.25	0.5	0.5
NMRR(q)	0.0889	0.1818	0.1571	0	0.1950	0.0189	0.7286	0.4951	0.4780

$ANMRR(q) = \frac{1}{Q}\sum_{q=1}^{Q} NMRR(q) = 0.2604$

3.1b: EMD without Spatial Coherency.

CLASS	CCQ 1	CCQ 2	CCQ 3	CCQ 4	CCQ 5	CCQ 6	CCQ 7	CCQ 8	CCQ 9
NG(q)	9	5	4	3	12	6	4	8	6
K(q)	24	20	16	12	24	24	16	24	24
NR(q)	9	5	4	3	9	6	2	7	2
MR(q)	0	0	0	0	3	0	2	1	4
AVR(q)	5	3	4.75	2	12.9167	3.667	11	10.75	23.83
MRR(q)	0	0	2.25	0	6.4167	0.167	8.5	6.25	20.33
RR(q)	1	1	1	1	0.75	1	0.5	0.875	0.3333
NMRR(q)	0	0	0.1286	0	0.2731	0.0063	0.4875	0.2451	0.7672

$ANMRR(q) = \frac{1}{Q}\sum_{q=1}^{Q} NMRR(q) = 0.2019$

3.1c: EMD with Spatial Coherency.

CLASS	CCQ 1	CCQ 2	CCQ 3	CCQ 4	CCQ 5	CCQ 6	CCQ 7	CCQ 8	CCQ 9
NG(q)	9	5	4	3	12	6	4	8	6
K(q)	24	20	16	12	24	24	16	24	24
NR(q)	9	5	4	3	9	6	2	8	2
MR(q)	0	0	0	0	3	0	2	0	4
AVR(q)	5	3.4	3.75	2	14.5	3.5	11.5	6.125	20.33
MRR(q)	0	0.4	1.25	0	8	0	9	1.625	16.833
RR(q)	1	1	1	1	0.75	1	0.5	1	0.3333
NMRR(q)	0	0.0182	0.0714	0	0.3404	0	0.5143	0.0637	0.6351

$ANMRR(q) = \frac{1}{Q}\sum_{q=1}^{Q} NMRR(q) = 0.1914$

Table 3.2: The average number of EMD computations (the database contains 2045 images).

3.2a: The number of EMDs computed after using M-tree

Number of top images		Top 10	Top 20	Top 50	Top 100	Top 200
Number of EMDs computed	2-color	696	807	994	1168	1348
	5-color	1246	1312	1502	1647	1788
	Full-color	1316	1491	1645	1754	1871

3.2b: The number of EMDs computed after using lower bound

Number of top images		Top 10	Top 20	Top 50	Top 100	Top 200
Number of EMDs computed	Without SC	638	757	943	1115	1329
	With SC	623	740	926	1098	1314

Table 3.3: The performance evaluation of different methods.

3.3a: The ANMRR and Precision/Recall for DCD and EHD.

Methods		ANMRR	Precision	Recall
1	EHD only	0.2937	0.4512	0.7125
2	DCD only	0.2752	0.5243	0.7672
3	Average combination	0.1445	0.6524	0.8130
4	Optimization combination	0.1298	0.6917	0.8627

3.3b: The ANMRR and Precision/Recall for CSD and EHD.

Methods		ANMRR	Precision	Recall
1	EHD only	0.2937	0.4512	0.7125
2	CSD only	0.1017	0.7058	0.9149
3	Average combination	0.1118	0.6967	0.8954
4	Optimization combination	0.0687	0.7426	0.9367

3.4 Summary

In this chapter, we presented the image retrieval based on MPEG-7 descriptors. The Earth Mover's Distance is an effective and flexible measure with serval desirable properties. Compared with the original similarity measure of DCD in XM, better results can be achieved when using EMD as the measure. Furthermore, both lower bound and M-tree can also be used to reduce the number of EMDs calculated when the optional variance parameter and spatial coherency are ignored. The results show that the performance of lower bound is much better than M-tree. It excludes those images that does not need to calculate the EMD distance, therefore reducing the number of the images to half. By itself, it is also very easy to compute. M-Tree structure can also optimize the database structure. Its results are affected by both distribution of the distance and dimensionality of the feature. It can be seen that the 2-color index can meet the same effect as lower bound. But the results is only approximate and can be used for first level filtering for multi-level retrieval or browsing the image database based on single or two colors. When the fixed lower bound cannot be used, M-tree is also a choice. For some features with low dimension, M-tree performs well.

Besides the image retrieval based on single MPEG-7 descriptor, in this chapter we propose a weighted combination method for image retrieval based on multiple features. It is applied to several MPEG-7 visual descriptors and the weight of every descriptor is determined self-adaptively based on optimization technology. From our experiments, query by multiple descriptors with optimal weights can achieve better performance than query by single descriptor or simple average combination. The same optimization structure can be used for many other visual features. It is a unified approach to content-based image retrieval. The optimal solution is explicit and the calculation procedure is not time-consuming.

Chapter 4

Object category classification and retrieval

To bridge the semantic gap, object recognition based on low-level features is an important issue to address for image applications. Detection, representation, and training are the three major issues that need to be discussed in an object recognition or classification system. The difficulty in object-based image retrieval is the identification of an object under different viewing conditions. In this case, the query might contain objects at a different scale and orientation from those present in the database. Therefore, for object retrieval or recognition, this factor should also be considered in addition to the common visual features, such as color, texture and shape. In the last few years, scale and orientation invariant features for CBIR systems has been investigated [70, 71]. These local features focus on the local characteristics of the specific part and suitable for tasks where one is looking for the identical points of a object. To represent object categories, using collection of regions is a possible approach. Each region is a distinctive part of the corresponding object category.

In this chapter, we investigate the object-based image retrieval using scale and orientation invariant features. The method can find an object in different scales and orientations. Then we expand the recognition from the same object to object categories. We present

an efficient method to classify the object categories based on the constellation of representative regions. The regions with highest appearance frequencies are used to build the model. Experimental results show that the image model based on representative regions is easy to calculate and an improved performance can be achieved.

4.1 Robust matching and retrieval based on scale and orientation invariant features

In this section scale and orientation invariant features are applied for region-based object query. An efficient search algorithm based on interest point matching is discussed. Each region represents an object at different scale or orientation. In these regions the interest points and corresponding feature vectors are calculated based on [6]. The points are detected by the Harris-Laplacian detector and the features are the Gaussian derivatives calculated at the corresponding points. Then we use these features to query and improve the retrieval process. Our proposed robust matching and retrieval procedure includes two steps. First, a fuzzy distance measure is evaluated to measure the similarity for interest points. Second, the cross-correlation is calculated to reject some mismatched image and obtain the retrieval results.

4.1.1 The scale and orientation invariant feature of images

The objective of this work is to apply scale and orientation invariant features to region-based image retrieval and to find the most similar images containing a scaled and rotated object. The scale-invariant points are detected by the Harris-Laplacian detector from a multiple scale representation of an image built by the convolution with a series Gaussian kernel. In the multi-scale space, at first the Harris detector is used to detect the robust points in a 2D plane at several scale planes. Then the scale invariant points are selected from the robust points at which a local Laplacian measure is maximal over scales. So a

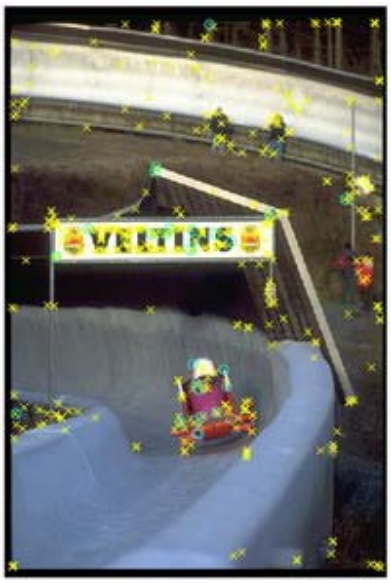


Figure 4.1: An example of detected scale invariant points. '∘' and '×' are the detected points.

Table 4.1: An example of two feature vectors. x and y are the corresponding coordinates of scale-invariant points. Cornerness is a value indicating the degree to which the detector believes this point is a corner. Scale is the detecting scale of the point.

x	y	cornerness	scale	feature vector (12-Dimension)
24	407	804.692	1	0.742243 0.0701232 -1.16625 0.568919 0.748237 0.298251 -0.0539323 1.11958 1.03285 -0.181406 -0.584073 0.771906
252	10	900.561	1.4	0.721299 -0.923366 0.920174 0.622473 -0.294663 0.559273 0.0311264 2.45683 0.089865 0.759739 0.888222 -1.64331

subset of the points computed in scale space are selected. Figure 4.1 shows an example of detected points. The corresponding derivatives at the scale invariant points are calculated as orientation invariant feature vectors [6]. Up to 4th order Gaussian derivatives at the point are computed to form a 12-dimension feature vector. Table 4.1 shows an example of the feature vectors. Rotation invariance is obtained by selecting the derivatives in the gradient direction.

These feature vectors are calculated for the images in database as well as for the query

regions. Our work is mainly about how to compare and find the most similar images with a scalable and rotated region efficiently. The query regions are cropped from the images in database manually and each region represent one or more objects. Querying with region that is cropped from an image makes it difficult to match its interest points with those in the database. This is due to the fact that cropped regions have fewer interest points, which could easily lead to mismatches. At the same time, we would like a point matching technique that is not computationally intensive so that the retrieval is fast as well as accurate.

The proposed region-based image retrieval algorithm consists of identifying the interest points in the query image and determining similar images from the database using a fuzzy distance measure. The results obtained at this stage are further refined using a cross correlation method to obtain the final retrieval images.

4.1.2 Fuzzy measure of matching degree

In [6] a voting algorithm is used to select the most similar images in the database. The distance between two matched points is evaluated to vote, with the image getting highest number of votes being the most similar result. The detailed procedure is as follows. For each point of a query image, its descriptor is compared to the descriptors in the database. If the distance is less than a fixed threshold, the vote of that image in database is increased by one. This will be repeated for each scale-invariant point. The images with the highest number of votes are retrieved as the most similar results. Using this method, the results greatly depend on the selection of distance threshold. In fact it is very difficult to find a suitable fixed threshold for all images. If this method is applied to region-based retrieval, a new problem occurs because a simple region usually does not have enough scale-invariant points. It could be very easy to match the small number of scale-invariant points in the region. When there are several images all have the same high rank, it cannot distinguish the most similar one and get the correct results.

To solve this problem, a matching degree is applied to measure the similarity of each voted point. It is based on the assumption that if the two images are correctly matched, the distance between the matched points of these two images will be very small. The matching degree reflects the real distance between the matched points of images. It is a fuzzy concept to measure the similarity and is of a value in the range $[0,\ 1] \subset \Re$, whereby 1 means these two points are equivalent and 0 means the distance is no less than the threshold. In the retrieval procedure, the matching degree of all matched points is accumulated and the average of accumulated matching degree is the final similarity distance between current image and the query region.

Currently a simple linear function is used to map the distance to a matching degree:

$$f(x_i) = 1 - \frac{x_i}{threshold} \qquad \text{(Eq. 4.1)}$$

where $threshold$ is the fix threshold and x_i is the ith computed distance. In our approach, the common Euclidean distance is used as the distance measure between any two interest points. Note that we have also calculated the distance using EMD, but the results are not satisfied. This may be caused by the large number of feature vectors. As there are hundreds of detected points in an image, it seems EMD cannot match these points efficiently.

With Eq. 4.1, the average matching degree $avgd$ of current image in database will be:

$$\begin{aligned} avgd &= \frac{1}{N}\sum_i f(x_i) \\ &= 1 - \frac{1}{N}\sum_i \frac{x_i}{threshold} \end{aligned} \qquad \text{(Eq. 4.2)}$$

where N is the number of matched points of current image. From our experiment, it can be seen that the results of using a matching degree are better than using vote number only.

4.1.3 Cross correlation

As mentioned in section 4.1.2, there are large number of interest points in the whole image. So it is possible that the average matching degree is large enough but in fact the region is not similar to any part of the whole image. This is because the matched points may be distributed in the whole image and not centralized on a part of the image, i.e., the similar regions. These points are wrongly matched and there will be no similar region. However, the correct selected image always has a similar region with the query region and the correct matched points will concentrate on that region. So the spatial location of matched points can be further detected to reject some mismatched images. To evaluate the spatial location of the matched interest points, cross correlations between the feature vectors of the query region and database images are calculated. Cross correlation is a standard method to estimate the degree to which the query region and the region in database image being sought are correlated.

In our system the features are represented by a series of points sort according to the spatial location in the image or region. The cross correlation between the two series of points of image and region separately is calculated with various delay. The peak value of cross correlation series is determined as the cross correlation between current image and the query region. If the cross correlation is less than a threshold, the image cannot be similar with the query region and will be rejected. Instead of using a fixed threshold, the average of all the cross correlations is used as the threshold. Cross correlation measure is suitable for using a region to query a whole image. It can identify a subset from the whole image which is similar with the query region.

The rejection procedure is shown as follows:

1. Calculate cross correlation series between the query region and the image in database with different delay.

2. Select the peak value of cross correlation series as the cross-correlation value.

3. Calculate the average cross correlation $avgcorr$ for this query region as the threshold.

$$avgcorr = \frac{1}{M}\sum_{i} corr(i) \qquad \text{(Eq. 4.3)}$$

where M is the whole number of images in the database and $corr(i)$ is ith cross correlation.

4. If $corr(i)$ is less than $avgcorr$, the image will be rejected.

4.1.4 The disadvantages of the scale invariant points

As discussed earlier, these scale invariant points and their features focus on the local characteristics of the specific part, because the target of these feature is to find the identical points by the suitable distance of point matching. For example Lowe's SIFT descriptor [35] has been shown in various studies to perform very well particularly at tasks where one is looking for the identical points. These methods are difficult to apply for the classification of object category, as the objects in different images may not be the same object. Another disadvantage is the large number of interest points. It is difficult to handle the large number of points with an approach which is efficient and easy to compute. More efficient methods need to be investigate to represent the object category.

4.2 Object category classification

Object understanding has always been an important tool for an image system to perform an automatic search for similar objects or organize a large database of objects. Object understanding usually attempts to find the same object under different visual conditions. Another related but different problem is describing and classifying object categories, such as [72, 73]. The important thing is to identity the characteristic of an object category and yet be able to adapt it for some minor variations. For example, the color of objects may be different, or some parts of the object may be absent or occluded. Image

retrieval based only on basic low-level visual features may find some visually similar images, but they may not in a same object category. For different categories of objects, the visual features may differ and cannot be easily compared using a generic set of global features. Recently object classification based on a collection of image regions has caught the attention of some research groups [74, 75]. Since each object category must have some distinctive properties, the definite parts of the objects can be used to represent the corresponding category. Another benefit of classification based on regions is the computation complexity. As the region is usually small and simple compared with the whole image, the feature extraction and classification computation is easy and fast.

In our work, the salient regions are used to build the image model. The basic idea is to calculate the matching probability of similar regions statistically. The regions which appear most frequently are selected as the representative regions. After clustering these regions are collected as an image model to represent the object category. After training, the obtained image model is used for object classification. The salient regions and same visual features in a new image are extracted and evaluated using the trained image model.

4.2.1 Selection and preprocessing for representative region

To identify the representative regions, salient regions detected in various scales have been discussed in [76]. Visual saliency is a broad term that refers to the idea that certain parts of a scene are pre-attentively distinctive. The visually attentive areas create immediate significant visual arousal within the first few hundreds of a second when the image is shown to the viewer [76]. These detected salient regions are reasonable representation of the whole image and can be used in image classification and image understanding.

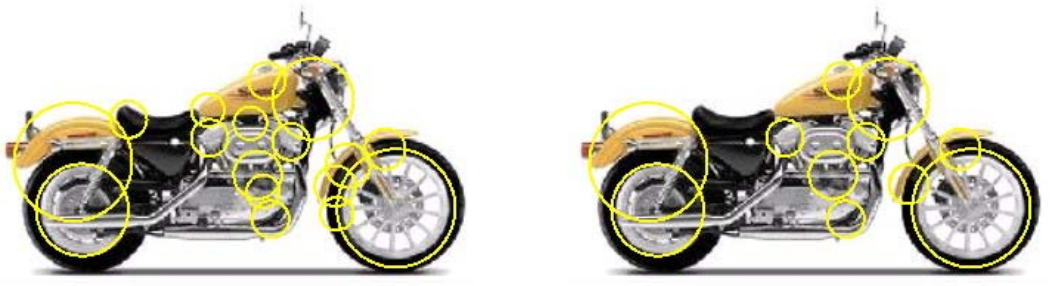
In our work the salient regions over suitable scales are used to represent the objects. The salient regions are identified using the detector provided by Kadir and Brady [76]. It is a affine invariant salient region detector. This region detector searches the saliency in

the scale-space as well as spatial dimensions. It includes two steps to detect the salient region. Firstly, the detector searches for scale localized features with high entropy and obtains the scale with peaked entropy. The value of entropy has been weighted by the sum of absolute difference of the PDFs (Probability density functions) of the local descriptor around the peak entropy. Second, the salient volumes are clustered to form the salient region. With the detector, regions with the highest saliency over the image are selected for training and classification. In practise, this method gives a stable identification of features over a variety of sizes and copes well with intra-class variability. Since this method applies the scale dissimilarity as a weight for each detection, it can correctly capture the most salient scales.

As the detected regions usually overlap each other, and a large number of regions are difficult to handle, a preprocessing step is necessary to select the prominent regions. If any two regions overlap by more than 2/3, these two regions are merged to form a new larger region. Some very small regions are removed if there are no other regions that are close to them. Figure 4.2 shows an example of the detected salient regions before and after preprocessing. Once the regions are detected and pre-processed, they will be cropped from the image using the minimum bounding box.

4.2.2 Feature representation of each region

For each region we need to select efficient features. The basic visual features includes color, texture, shape etc. In our work the color, texture and location information of a region are used to represent the regions and build the image model. For conformance with standard description of visual features, we use Dominant Color Descriptor (DCD) and Homogenous Texture Descriptor (HTD) of the MPEG-7 standard as the color and texture feature respectively. These descriptors and the locations are used to describe the regions.



4.2a: Before preprocessing

4.2b: After preprocessing

Figure 4.2: An output example of the region detector and these regions after preprocessing.

4.2.2.1 Color feature

For color images, color is one of the most expressive and distinguishing visual features. Although the salient regions are detected based on monochrome information, the color in the region is still useful to distinguish different kinds of objects. As the salient regions are usually small and has limited colors, a color histogram with many bins is unnecessarily complex. So a color feature which can represent some main color information of the regions is suitable. MPEG-7 standard provides a set of standard description for multimedia content. To meet the requirement, we select the Dominant Color Descriptor, which describes the dominant colors of an image or region in any shape.

As described in section 3.1.1, DCD can specify a small number of dominant color values and their statistical properties including percentage and variance [18]. The descriptor consists of the color index (c_i), color percentage (p_i), color variance (v_i) and spatial coherency (s); the last two parameters are optional and we have omit these two parts. Therefore the DCD is defined by:

$$F = \{(c_i, p_i)\}, i = 1, ..., N. \qquad \text{(Eq. 4.4)}$$

where N is the number of the colors and $\sum p_i = 1$. The maximum number of colors in DCD is modified to adapt our application. For regions, we set the maximum number

of colors to four instead of the standard eight. The minimum number of colors is two. The color index can be represented by common color space, such as RGB, HSV (hue-saturation-value) or HMMD (Hue-Max-Min-Difference). Here we use the common RGB color space. As the size of regions will not effect the value of color features, for different regions we need not normalized the size of regions when extracting DCD.

4.2.2.2 Texture feature

As can be observed, textures within an image are usually concentrated in various regions of the image. Therefore, texture is likely to be a region property, and the different levels of homogeneity is a result from the presence of multiple colors or intensities within an image. Texture can be applied to distinguish many natural and artificial objects. We also apply a standard MPEG-7 descriptor as the texture feature in our experiments. MPEG-7 Homogeneous Texture Descriptor characterizes the region texture with the mean and deviation of energy of frequency channels [18]. The mean energy and its deviation are computed in each of these 30 frequency channels. Figure 4.3 shows the 2-dimension frequency plane partitioned into 30 channels. In our experiments, HTD is used as the texture feature of a region. The structure of the HTD is shown as follows:

$$HTD = [f_{DC}, f_{SD}, e_1, e_2, ..., e_{30}, d_1, d_2, ..., d_{30}] \quad \text{(Eq. 4.5)}$$

where f_{DC} and f_{SD} are the mean and standard deviation of the image, respectively. e_i and d_i are the nonlinearly scaled and quantized mean energy and energy deviation of the corresponding ith channel in Figure 4.3 , respectively.

4.2.2.3 The location

Besides the standard color and texture features, the geometric distance between the location of two regions is also evaluated. The change in location of corresponding regions

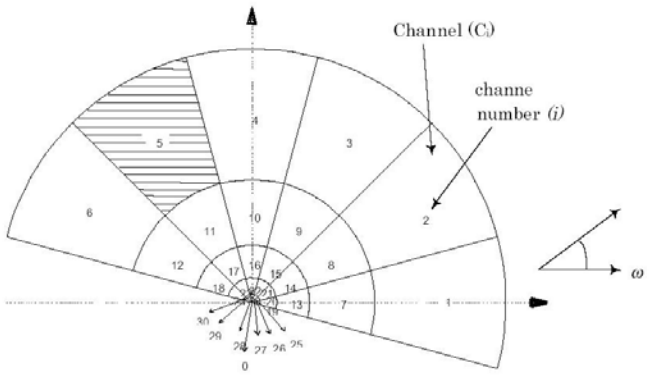


Figure 4.3: Channels used in computing the HTD [2].

represent the geometric distortion between the objects. To measure the geometric distortion of regions, the images are normalized in advance. Then the Euclidean distance between the coordinates of centroid of two regions is calculated as the distance.

4.2.3 Selection of representative regions

In this section how to select the most representative regions is introduced. It mainly includes two steps. Firstly, when a region either has multiple matching regions or a smaller cumulative matching distance, it will have a higher matching probability. The regions with high matching probability are selected and cropped as the candidate representative regions of the current object. After that, the candidate regions are clustered to select the most representative regions.

4.2.3.1 Distance for region comparison

For color and texture features different similarity distances are used for computation. We applied Earth Mover's Distance (EMD) as the similarity distance for the color feature (DCD). Since EMD can be used to calculate the distance between two multi-dimensional distributions where each distribution is represented by sets of weighted features, it is

appropriate to use the EMD to determine the similarity between two DCDs, where the basic elements are the dominant colors and their corresponding percentages. The distance between DCDs of two regions i and i^* is represented by $d_{DCD}(i, i^*)$ in our work. The details can be found in section 3.1.3.

In the training procedure, the similarity distance between two regions based on HTD is measured by summing the weighted absolute difference between two sets of feature vectors, HTD_i and HTD_{i^*}. The function used to calculate the distance between two HTDs is shown in Eq. 4.6 [4].

$$d_{HTD}(i, i^*) = \sum_k \left| \frac{HTD_i(k) - HTD_{i^*}(k)}{\alpha(k)} \right| \qquad \text{(Eq. 4.6)}$$

where $\alpha(k)$ is normalization value and it is usually specified to 1 as in [4]. k is the number of HTD bins.

As the similarity distance of different features cannot be compared directly, the similarity distance is normalized between 0 and 1 before comparison to avoid the difference between the various features. Currently the original distance of features is proportional to the matching degree. A linear function is used to map the original distance to a normalized matching degree:

$$\begin{aligned} degree(d_{DCD}(i, i^*)) &= 1 - \frac{d_{DCD}(i, i^*)}{threshold_{color}} \\ degree(d_{HTD}(i, i^*)) &= 1 - \frac{d_{HTD}(i, i^*)}{threshold_{texture}} \end{aligned} \qquad \text{(Eq. 4.7)}$$

The $threshold_{color}$ and $threshold_{texture}$ are the corresponding thresholds which are defined as the maximum distance selected from all similarity distances of the corresponding features respectively. Then the matching degree is from 0 to 1.

As described in section 4.2.2.3, the Euclidean distance between the coordinates of centroid of two regions are calculated as part of the similarity distance for each feature ($d_{location}(i, i^*)$). So the distance function of DCD and HTD are shown as follows:

$$\begin{aligned} dist(i, i^*)_{color} &= degree(d_{DCD}(i, i^*)) + d_{location}(i, i^*) \\ dist(i, i^*)_{texture} &= degree(d_{HTD}(i, i^*)) + d_{location}(i, i^*) \end{aligned} \qquad \text{(Eq. 4.8)}$$

4.2.3.2 Matching probability

It is difficult to only use one image to represent an object. In our work several images which contain objects of the same category are used to select the representative regions for that object category. The training procedure is described as follows:

At beginning all training images are randomly selected from the image class. Each class is a set of images which contains an object category. Usually for an object category, 10 images are randomly selected for training. The salient regions of each image are detected and preprocessed as described in section 4.2.1. After preprocessing the color and texture features of each salient region are extracted separately.

The image model is built according to the conjunction function [77]. Initially the image model includes all the salient regions, as $R_1 \wedge R_2 \wedge R_3 \wedge ... \wedge R_n$, where R_i represents the regions and n is the total number of regions. The irrelevant items are determined and removed from the function iteratively. For each region in an image, it is compared with all other regions in other training images. The calculated matching degree as Eq. 4.7 between a pair of regions is recorded as a fuzzy matching probability. The matching probability of every region is accumulated during the training. If more images have such a similar region, the matching probability of this region is much higher than others. Then this region is selected as a representative region for the current object category. For a region with small number of matched regions, it will be removed from the initial model function. The matching probability of a region is the summation of all matching degrees of matched regions of this region.

4.2.3.3 Clustering of the selected regions

Using matching probability only, we could end up with numerous similar representative regions. Figure 4.4a shows such an example. It can be seen that the top 10 regions with high probability are almost the same three regions in different images, i.e., parts

4.4a: Before clustering (motorbike), 10 representative regions

4.4b: After clustering (motorbike), 10 representative regions

Figure 4.4: The selected 10 regions before and after clustering for motorbike image class. The regions in the red and yellow cycles are the representative regions. The red cycles demonstrate the regions which are similar to the regions in Figure 4.4a.

of the front wheel and the spokes of the whole front wheel. Because these regions are similar to each other, all their matching probability are increased accordingly during the computation. In order to reduce representative regions being all from just a few similar regions, we use clustering method to select one representative region to represent a set of similar regions.

The centroid of each cluster can be directly used to build the model. But in order to demonstrate the representative regions visually, the regions which are closest to the centroid of each cluster are selected as the represented regions. After that other similar regions are removed from the model. In this thesis the k-means clustering method is

applied for clustering. The algorithm consists of a simple re-estimation procedure as follows. First, the data points are assigned randomly to the clusters. Then the centroid is computed for each cluster. These two steps are alternated until a stopping criterion is met, i.e., when there is no further change in the assignment of the data points. The number of clusters is predefined as the number of representative regions. In our method, the number of cluster is 10 for the model used for basic classification measure. Because in the first step the regions are randomly assigned to clusters, the results of representative regions may be slightly different due to the clustering discrepancies.

4.2.3.4 The weights of representative regions

After clustering, a set of representative regions based on color and texture features are obtained separately. Among these regions, different regions have different representative degrees for the objects. Some representative regions should have a higher weight, as the object which contains these regions in Figure 4.4b is more likely to be the object. The representative regions with high weights in Figure 4.4b are shown using red color. These regions appear more frequently than other regions in the motorbike class. So the weight of region is related to the corresponding appearance frequency of this region. It means if one cluster has more regions, the representative region of this cluster is more important and should have a higher weight.

Therefore, for each region in the model, the number of similar regions in a training set that have been classified into the corresponding cluster is evaluated as the suitable weight. The weight of each region w_i is defined as:

$$w_i = \frac{n_{cluster_i}}{N_{all}} \qquad \text{(Eq. 4.9)}$$

where $n_{cluster_i}$ is the number of regions in cluster i and N_{all} is the total number of regions in all cluster. So the cumulative weight is normalized to 1.

As the color and texture focus on different characteristic and appearance of a region, different regions will be selected according to color and texture features. So the model is a combination of regions selected by color features and a number of regions selected by texture features. Hereby the matching probability consists of two parts, color and texture. When the representative regions selected by color and texture features are combined, the weights in Eq. 4.9 are also applied. For those regions selected by both color and texture features, their matching probability is increased to the weighted sum of the matching probabilities based on two features separately. These regions with high weights will play an important role in the similarity comparison.

4.2.3.5 The whole training algorithm for the image model

Suppose D_i is the matching probability of region R_i and $dist(R_i, R_{i^*})$ is the calculated matching probability between region R_i and R_{i^*}, the complete algorithm is shown in algorithm 1. *threshold* is a predefined value to decide if the region should be removed. k is the number of representative regions, i.e., the number of clusters.

In our experiments, after preprocessing an image usually has 5 to 15 detected regions, so there will be quite a number of comparisons. But due to the simple features and easy matching method, the calculation time and memory requirements are reduced.

4.2.4 Basic classification using the trained image model

Object classification proceeds basically by evaluating the color and texture features (appearance) of other images using the trained image model. The location change of the regions is also considered. Similarity distances based on color and/or texture and location change will become the overall combination distance. This likelihood measure of each image is calculated from the overall combination distance and compared to the threshold of classification. If the likelihood measure is larger than the threshold, the object belongs to this known category. Otherwise the object will be rejected.

Algorithm 1 : Region Selection

Input: R_i, k
Output: R_s, the representative regions

Initialization
$R = R_1 \wedge R_2 \wedge R_3 \wedge ... \wedge R_n$
for each item R_i in R **do**
    $D_i = 0$
end for
Reduce irrelevant items using positive samples
for each item R_i in R **do**
    for each item R_{i^*} in R, $i^* \neq i$ **do**
        $D_i = D_i + dist(R_i, R_{i^*})$
    end for
    if $D_i < threshold$ **then**
        $R = R - \{R_i\}$
    end if
end for
for each item R_i in R **do**
    k-means clustering to produce k clusters
end for
for each center of the cluster **do**
    Find the nearest region R_s in R
end for
Return R_s(k regions)

4.2.4.1 Comparison between regions

The classification based on the model is according to the similarity between the regions of model and the input image. For each region in the model, the corresponding region in the image is selected. The quality of a correspondence is measured in two aspects: how similar appearance is to its corresponding regions, and how much the spatial arrangement of the regions is changed. The former is represented by the calculated matching probabilities based on color and texture, and the later is represented by the geometric distortion of a pairs of regions. So the computation of likelihood measure includes two steps. Firstly a set of regions which have similar appearance to the regions in the model are selected.

Then the mixed similarity measures are calculated between the model regions and selected regions, including the appearance similarity and the geometric distortion. After that the region pairs with maximum mixed similarity measure are selected as the matched region pair. So the likelihood measure of the image is the sum of all similarity measures of matched region pairs as in Eq. 4.10.

$$dist_{total} = \sum_{i=1}^{N} [dist_{color}(i) + dist_{texture}(i) + dist_{distortion}(i)] \qquad \text{(Eq. 4.10)}$$

where i is ith matched region pair and N is the number of total matched region pairs. The details of Eq. 4.10 can be found in next section.

At the beginning, for each region in the model, the matching similarities compared with all the regions in the image are calculated. Using the distance functions in section 4.2.3.1, the similarity distances based on color and texture information are calculated separately, and normalized as Eq. 4.7. Then for each region in the model, the closest regions in the image are selected as the similar region set. This is determined by comparing the similarity with a predefined threshold. If the similarity degree is less than the threshold, we consider there is no matched region for current region. Currently we define the threshold is 0.7. For one region in the model, it may have several similar regions in the image. The most matching region is determined by the geometric distortion as described in next section.

4.2.4.2 Geometric distortion costs

Besides the appearance comparison between the representative regions, the spatial relationship between a pair of regions are also considered. For the change of location we use a flexible and efficient distance, the geometric distortion costs, instead of the direct comparison of coordinates. We consider correspondence between regions in image model and the matched regions in any image of database.

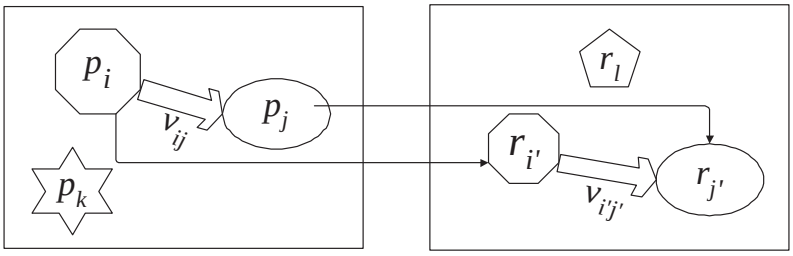


Figure 4.5: An illustration of geometric distortion between pairs of regions. p_i, p_j is a pair of regions, and $r_{i'}, r_{j'}$ is the corresponding region pair. v_{ij} and $v_{i'j'}$ are the two vectors formed by region pairs.

The geometric distortion is computed over pairs of regions in an image. Figure 4.5 demonstrates an example of region pairs and the geometric distortion between them. The centroid of two regions form a vector, so the difference in direction and length of two offset vectors of corresponding regions in two images are measured. We adapted the distortion function in [36] for the regions. The function is shown in Eq. 4.11. The dot product of two vectors is used to calculate the angles.

Suppose the trained image model P has N regions $P = \{p_i, i = 1, ..., N\}$, and for each p_i, it has a set of similar regions $R_i = \{r_{i,i'}, i' = 1, ..., M\}$ in Q, where Q is any image from the database for classification. To reduce notational clutter we abbreviate $r_{i,i'}$ as r'_i. Then for any two regions p_i and p_j in P, they composes a vector $v_{ij} = p_i - p_j$. p_i has a similar region set R_i and p_j has a similar region set R_j. The corresponding vector of v_{ij} in Q is one of vector set $v_{i'j'} = r'_i - r'_j, r'_i \subset R_i, r'_j \subset R_j, r'_i \neq r'_j$. At last the calculation of geometric distortion between two vectors is as follows:

$$
\begin{aligned}
d_{angle} &= \left| \arccos\left(\frac{s_{i'j'} \cdot r_{ij}}{|s_{i'j'}||r_{ij}|} \right) \right| \\
d_{length} &= \frac{s_{i'j'} - |r_{ij}|}{|r_{ij}|} \\
d_{ij,i'j'} &= 1 - \frac{distortion}{threshold} \\
distortion &= \min_{i',j'} d(r_i, r_j; r_{i'}, r_{j'}) \\
&= \min_{i',j'} d(v_{ij}, v_{i'j'}) \\
&= \min_{i,j} \left(w_1 d_{angle} + (1 - w_1) d_{length} \right)
\end{aligned}
\qquad \text{(Eq. 4.11)}
$$

where d_{angle} calculates the angle between two vectors, and d_{length} calculates the change in length between two vectors. The constant w_1 weighs the proportion between angle distortion term and the length distortion term.

The combined distance between two regions includes similarity distance based on appearance and geometric distortion. The proportion of these two parts is adjusted by a weight w_2 as follows:

$$
comb\_dist_{i,j;i',j'} = w_2 * (d_{i,i'} + d_{j,j'}) + (1 - w_2) * d_{ij,i'j'} \qquad \text{(Eq. 4.12)}
$$

where $d_{i,i'}$ (or $d_{j,j'}$) represents the similarity distance between region r_i and $r_{i'}$ (or r_j and $r_{j'}$) based on the appearance features, and $d_{ij,i'j'}$ represents the geometric distortion between the offset vectors. The constant w_2 weighs the similarity distance against the geometric distortion. We assign it to 0.5 in the experiments. Note that these distances are normalized according to the threshold in advance.

For the whole image, the likelihood measure $dist_{total}$ of any images in a database is the sum of distances of all match regions. It is shown as follows:

$$
dist_{total} = \sum_{i,j} comb\_dist_{i,j;i',j'} \qquad \text{(Eq. 4.13)}
$$

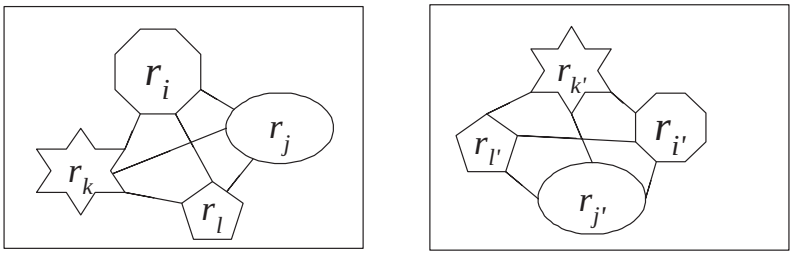


Figure 4.6: An illustration of nested spatial relationships between regions. These two groups of four regions have different positions, but the spatial structures are the same.

4.2.5 Object model matching based on graph structure

In section 4.2.3 we have obtained the representative regions of the object category successfully. In section 4.2.4 the regions in the model are directly used to classify the object, but the relationships between the regions are not utilized adequately. In this section we apply a more sophisticated method to evaluate the correspondences between the regions in the model and images. Instead of only pairs of regions, the nested spatial relationships between multiple regions are adopted to improve the results. Figure 4.6 demonstrates an example.

4.2.5.1 Object representation based on attributed relational graph (ARG)

The attributed relational graph (ARG) is a relational structure which consists of a set of vertices and a set of edges, which are representing the relationships between these vertices. The detailed definition of ARG is given in [78]. A complete ARG can conveniently represent a content model which combines the individual attributes of a set of spatial entities along with their binary relationships. In ARG the spatial entities are represented as vertices (V), each labeled with a unary attribute (v_i), and binary spatial relationships are represented as pairs of vertices ($V \times V$), i.e., the edges (E), each labeled with a spatial feature (e_{ij}).

In our method, the regions that are the centers of all clusters are used to build the model as described in 4.2.3. We use ARG to organize the model and the structure of

model is a full-connected graph. Each vertex of the graph corresponds to a representative region and the feature vector of the vertex is the color or texture feature of the region. The edge reflects the spatial relations between the regions. The feature vector of the edge is the difference of coordinates of regions. For each image in the database, a graph is built based on the detected salient regions.

4.2.5.2 Graph matching based on EMD

As introduced in section 3.1.2, Earth Mover's Distance (EMD) is a perceptual flexible similarity measure between two weighted multi-dimensional distributions. The solution of EMD is based on the well-known transportation problem. Suppose that there are several suppliers required to supply several consumers. Each supplier has a known amount of goods and each consumer has a known capacity. For each supplier-consumer pair, the transportation cost of a single unit of goods is given. The transportation problem can be defined as to find a least-expensive flow of goods from the suppliers to the consumers that satisfies the consumers' demand. Graph matching can be naturally cast as a *transportation problem* by defining one graph as the supplier and the other as the consumer. The cost for a supplier-consumer pair is set as to the ground distance between a vertex in the first graph and a vertex in the second. Intuitively, the solution is the minimum amount of "work" required to transform one graph into the other.

However, EMD cannot be directly applied to graph matching because the ground distance between vertices cannot be computed directly. So the computation procedure can be divided into two steps. In the first step, the ground distances between every pair of vertices in these two graphs are computed. Suppose a vertex in the supplier graph is matched with another vertex in the consumer graph, the minimum cost for the conversion between two graphs under this situation is computed. This cost can be considered as the ground distance between this pair of vertices. After the ground distance of every pair

of vertices is calculated, the minimum cost of converting these two graphs is computed based on the obtained set of ground distances.

The idea is similar to the nested EMD in [42]. In [42] EMD is modified as nested EMD (nEMD) to measure the similarity between two graphs. The nested structure of the EMD consists of inner EMD and outer EMD. The inner EMD reflects the difference of both vertices and edges between a pair of vertices in two ARG's in a perceptual manner, and the outer EMD establishes the correspondence between vertices in the two ARG's in a natural way. Note that in [42] some details of inner EMD is given and the outer EMD is only described briefly. How to computed the outer EMD and the final matching flow is not introduced. As the computation is based on the *transportation problem* and Earth Mover's Distance, we have investigated and finished the computation of outer EMD. The results are verified and applied to the matching of our image model successfully.

To be compliant with the description in [42], we use the similar notation in the thesis. Assume that two ARGs to be matched are $G = (V, E)$, where

$$V = \{v_i | 1 \leq i \leq n\} \quad \text{(Eq. 4.14)}$$

$$E = \{e_{ij} | i \neq j, 1 \leq i \leq n, 1 \leq j \leq n\}, \quad \text{(Eq. 4.15)}$$

and $G' = (V', E')$, where

$$V' = \{v'_{i'} | 1 \leq i' \leq n'\} \quad \text{(Eq. 4.16)}$$

$$E' = \{e'_{i'j'} | i' \neq j', 1 \leq i' \leq n', 1 \leq j' \leq n'\}. \quad \text{(Eq. 4.17)}$$

The completed computation procedure is as follows:

1. 1st cost matrix for EMD (Inner EMD): for a pair of vertices V_i in the model G and V'_i in image G', suppose these two vertices are matched. Then a matrix for the transformation cost of other vertices under this situation is computed as D_{inner}. The cost between any other two vertices includes two parts, the unary part and the binary

part. The unary part is computed based on the features of these two vertices, i.e., the color or texture feature of the regions. The same distance functions as in section 4.2.3.1 are applied to DCD and HTD separately. The binary part is based on the features of the edges, i.e., the difference of spatial coordinates of the regions.

Finally, an inner EMD of the two vertices can be computed using a distance matrix, $D_{inner} = [d_{inner}^{(j,j')}]$, including all differences from their unary features and binary relationships, where j and j' are the vertices in graph G and G' respectively. More specifically, given i and i' , $d_{inner}^{(j,j')}$ is computed as following equation:

$$D_{inner}^{i,i'} = [d_{inner}^{(j,j')}] \quad \text{(Eq. 4.18)}$$

$$d_{inner}^{(j,j')} = (1-\alpha)d(V_j, V'_{j'}) + \alpha d(e_{ij}, e'_{i'j'}) \quad \text{(Eq. 4.19)}$$

where $d(V_j, V'_{j'})$ and $d(e_{ij}, e'_{i'j'})$ means the pre-defined distance between the vertices and edges, respectively. Especially, $e_{ij} = 0$ or $e'_{i'j'} = 0$ when $i = j$ or $i' = j'$, respectively.

2. 2nd cost matrix for EMD (Outer EMD): For every pair of matched V_i and $V'_{i'}$, calculate the optimal cost when G is transferred to G' in this case. Actually this optimal cost is an EMD solution based on the previous cost matrix in Eq. 4.19. Combining with the difference between V_i and V'_i themselves, the minimum cost $d_{outer}^{i,i'}$ for V_i and V'_i is calculated as follows:

$$d_{outer}^{i,i'} = (1-\alpha)d(V_i, V'_{i'}) + \alpha EMD(D_{inner}^{i,i'}) \quad \text{(Eq. 4.20)}$$

where $EMD(D_{inner}^{i,i'})$ is the EMD solution based on the cost matrix $D_{inner}^{i,i'}$. During the computation of EMD, to allow for partial matches, all weights in both inner and outer EMD's are identically provided as follows:

$$w_i = w'_{i'} = \frac{1}{\max(n, n')}, 1 \le i \le n, 1 \le i' \le n' \quad \text{(Eq. 4.21)}$$

Algorithm 2 : Graph matching based on 2-step EMD

Input: $G = \{V, E\}$, $G' = \{V', E'\}$, α

Output: $Flow$, the matching vertices and cost flow between G and G'

```
for each item V_i in G do
    for each item V'_i in G' do
        Suppose V_i and V'_i is matched
        D^{i,i'}_{inner} = [d^{(j,j')}_{inner}]
        d^{(j,j')}_{inner} = (1 − α)d(V_j, V'_j) + αd(e_ij, e'_ij)
    end for
end for
D_outer = [d^{(i,i')}_{outer}]
d^{i,i'}_{outer} = (1 − α)d(V_i, V'_i) + αEMD(D^{i,i'}_{inner})
Flow = EMD(D_outer)
Return Flow
```

For all the pairs of vertices in G and G', the minimum costs $d^{i,i'}_{outer}$ based on the inner EMD are calculated. These minimum costs form a new $n \times n'$ matrix D_{outer}. It is the ground distance matrix to convert G to G'.

$$D_{outer} = [d^{(i,i')}_{outer}] \qquad \text{(Eq. 4.22)}$$

3. Using D_{outer} as the ground distance matrix for the computation of EMD, the EMD to convert G to G' is calculated. At the same time the minimum flow matrix F between vertices in G and G' can be obtained. It establishes the corresponding relations between the vertices of G and G' automatically. The EMD is used to identify whether there is an object in the image.

The completed algorithm of graph matching based on EMD is given in Algorithm 2.

4.2.5.3 A computational example of graph matching based on EMD

In order to demonstrate the graph matching algorithm, we give an example of sub-graph matching to clarify the computation procedure based on EMD. Figure 4.7 shows two fully connected and undirected ARG to be matched, G with five nodes and G' with

four nodes. So it is a sub-graph matching. Euclidean distance is used to calculate the distance between vertices and edges. The weights of all vertices in G and G' are set to 0.2 as determined by Eq. 4.21. In fact, G and G' can be converted into each other by using a permutation matrix P , which is given by

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{(Eq. 4.23)}$$

Now, let us investigate the detailed procedure of the 2-step EMD. First, an inner EMD between every pair of vertices in G and G' requires a corresponding D_{inner}. In case that the 1st node in G and the 1st node G' are matched, i.e., $i = i' = 1$, the inner EMD of the two vertices is given by

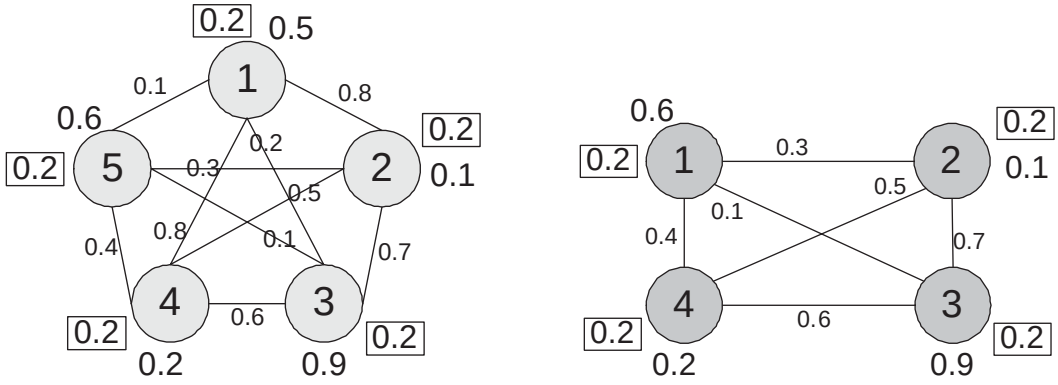
$$D_{inner}^{1,1'} = \begin{bmatrix} 0.05 & 0.35 & 0.25 & 0.35 \\ 0.65 & 0.25 & 0.75 & 0.25 \\ 0.25 & 0.45 & 0.05 & 0.45 \\ 0.6 & 0.3 & 0.7 & 0.2 \\ 0.05 & 0.35 & 0.15 & 0.35 \end{bmatrix} \quad \text{(Eq. 4.24)}$$

where $\alpha = 0.5$.

Using $D_{inner}^{1,1'}$ matrix in Eq. 4.24 as the ground distance, the minimum cost of transferring the graph in this case is 0.1375 computed by EMD. According to Eq. 4.20, $d_{outer}^{1,1'} = (1 - 0.5) \times \sqrt{(0.5 - 0.6)^2} + 0.5 \times 0.1375 = 0.11875$. After all the elements in D_{outer} are calculated, D_{outer} is given by:

$$D_{outer} = \begin{bmatrix} 0.11875 & 0.2875 & 0.23125 & 0.23125 \\ 0.33125 & 0 & 0.49375 & 0.075 \\ 0.2 & 0.5 & 0 & 0.45 \\ 0.29375 & 0.075 & 0.45 & 0 \\ 0 & 0.33125 & 0.2 & 0.29375 \end{bmatrix} \quad \text{(Eq. 4.25)}$$

The matrix in Eq. 4.25 is the ground distance between the vertices to transfer G to G'. So the EMD based on this ground distance matrix can be used to calculate the



4.7a: Graph G with 5 nodes

4.7b: Graph G' with 4 nodes

Figure 4.7: An example of sub-graph matching based on EMD.

minimum work for graph transferring and the flow. The calculated minimum work is 0 as the graph G' is a sub-graph of G. The flow in matrix format is shown in Eq. 4.26. It can be seen that the matrix F is corresponding to the permutation matrix P, which is proved that the correspondence of vertices is correct.

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \\ 0.2 & 0 & 0 & 0 \end{bmatrix} \qquad \text{(Eq. 4.26)}$$

4.2.5.4 Other applications of the image model

Besides object classification, the trained image model can be used for image retrieval. Retrieval proceeds by using the representative regions in the image model as an virtual image directly. The images in the database are compared with the virtual image as an initial comparison. Then the image model can determine whether the current image belongs to the object class.If the current image is belongs to the object class, it is similar to the query virtual image and added to retrieval result set. Other possible methods can be used to improve the initial results. In an image retrieval system with user feedback,

user can help determine a set of positive images and this set of positive images can be used to train the image model. This image model and its representative regions (virtual image) can be used for retrieval and will likely yield better results.

4.3 Experimental results

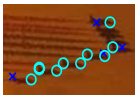
In this section some experimental results are presented to demonstrate the performance of the image retrieval based on the scale invariant points, and the object recognition based on the representative regions.

4.3.1 Image retrieval based on scale invariant points

The image retrieval are based on an image database including 1500 images and 300 regions. The images are randomly selected from the standard COREL Photo Library and we manually crop these 300 regions from our database of 1500 images. Each image has about 800 feature vectors and each region has about 100 feature vectors corresponding to each interest point. Some small regions contain only 10-20 feature vectors.

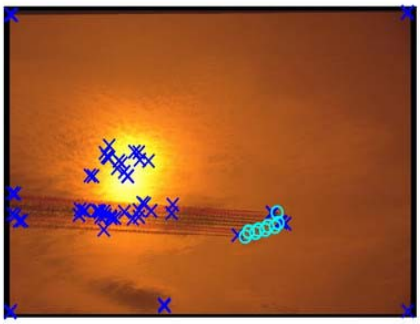
At first a small region without scale change is used to find the original image. 300 regions are tested and more than 290 corresponding original images can be found at the top rank successfully. Others are all in the top 10 results. Figure 4.8 shows some retrieval examples. It can be seen that even when the region is very small and simple, or there are many additional points in target image, the target image also can be found successfully. The matched points and all detected points are also shown and most of them are matched properly. The calculation is very fast because of the small number of interest points.

In the second part of experiments, new regions are formed from existing regions by applying different scales and rotations. Figure 4.9 is an example for query using rotated and scale-change region. The query region and top five results are shown here. The query region is rotated in 20 degree and changed with a 2.4 scale factor. It is a simple

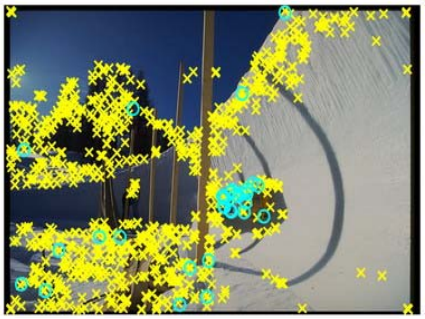


4.8a: 14 interest points

4.8b: 47 interest points



4.8c: 75 interest points, 10 matched

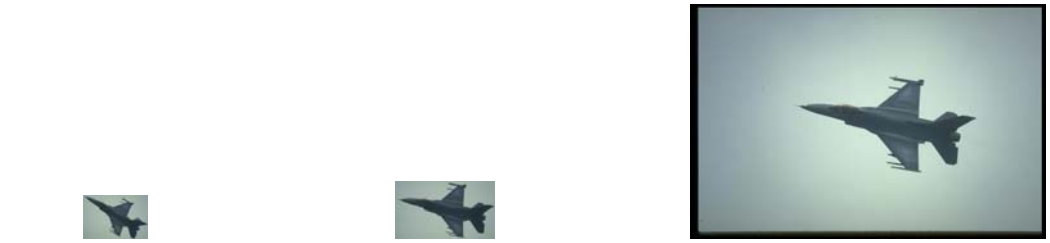


4.8d: 930 interest points, 33 matched

Figure 4.8: Query with small regions. 4.8a and 4.8b are the query regions and 4.8c and 4.8d are the corresponding top rank results. The numbers shown are the whole number of interest points detected in the region or image. 'o' means matched points and '×' means unmatched points.

region and only has 39 interest points. The original image is correctly found and some regions in other scale are also found in high rank. Rank 3 and 4 are two very similar image and region regardless of the color information. Currently the feature vectors are not represent any color information. Another example is given in Figure 4.10.

Table 4.2 shows the summarized retrieval results under different scale factor, "1.4", "1.8", "2.4" and "2.8". A total of 100 regions are used for test and each region is changed with these four scale factors. Some regions also rotated up to ±20 degree. The performance is evaluated as the average percentage when the original image is present as the first image or in the top five or in the top 10 of the result set. It can be seen that even if the scale is changed up to 2.8, the results are good. We compare our results with the voting algorithm of [6] and the results of the voting algorithm are shown in Table 4.3. Table 4.3a shows the results presented in [6] which are querying with the whole images and Table 4.3b shows the query results of using the same regions and images of



4.9a: Query region

4.9b: Rank 1

4.9c: Rank 2

4.9d: Rank 3

4.9e: Rank 4

4.9f: Rank 5

Figure 4.9: Top 5 results of querying with scale and orientation changed region. 4.9a is the query region, scale=2.4, rotated 20°, 39 interest points. 4.9b scale=1.8, 62 interest points, 17 matched. 4.9c is the original image, 126 interest points, 19 matched. 4.9d is a similar image, 209 interest points, 20 matched. 4.9e is a similar region, 181 interest points, 16 matched. 4.9f scale=1, rotated 15°, 112 interest points, 21 matched.

4.10a: Query region

4.10b: Rank 1

4.10c: Rank 2

4.10d: Rank 3

4.10e: Rank 4

4.10f: Rank 5

Figure 4.10: Top 5 results of querying with scale and orientation changed region. 4.10a is the query region, scale=2.8, 213 interest points. 4.10b scale=2.4, 299 interest points, 167 matched. 4.10c is the original image, 1761 interest points, 149 matched. 4.10d scale=1.8, 484 interest points, 137 matched. 4.10e scale=1.4, 699 interest points, 148 matched. 4.10f is a wrong matched image, 1362 interest points, 152 matched.

Table 4.2: Retrieval results based on regions in different scales. All 100 regions are rotated with 10°-20° separately.

	Scale factor (rotation 10°-20°)				
Retrieved rank	1	1.4	1.8	2.4	2.8
Top rank	97%	85%	76%	64%	52%
Top 5	100%	100%	90%	74%	66%
Top 10	100%	100%	94%	81%	70%

our image database. Here only the percentage of results in top 10 rank is evaluated. It can be seen that the results of region-based retrieval is not satisfied. Compared with the results of vote algorithm, it can be seen that the results of our method are better.

4.3.2 Recognition of object categories

The experiments were carried out as follows: the test sets are selected from the COREL image set, 101 Object Categories [1], Pond's image set [79] and the test set of [38], including

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

Table 4.3: Retrieval results using vote algorithm [6].

4.3a: The results of using whole images presented in [6].

	Scale factor (rotation 10°-20°)			
Retrieved rank	1.4	1.8	2.4	2.8
Top rank	60%	60%	60%	50%
Top 5	100%	90%	60%	80%
Top 10	100%	100%	90%	90%

4.3b: The results of using the same Corel regions and images in our image database.

	Scale factor (rotation 10°-20°)				
Retrieved rank	1	1.4	1.8	2.4	2.8
Top 10	50%	30%	20%	None	None

eagle, military aircraft, airplane (side), motorbike, car (side and rear), butterfly, bobsled, watch and face. Each class includes about 100 to 300 images. Totally there are about 1200 images in the database. Figure 4.11 shows some sample images from the image sets. From each class 10 images are randomly selected to train the image model.

4.3.2.1 Representative regions in image model

As described, 10 images of an object category are randomly selected for training. The results of region detection show that usually each image includes 5 to 15 regions. Therefore there are about 100 regions for comparison. The number of representative regions can be determined by a predefined threshold, i.e., the number of clusters. Here we use 10. Figure 4.12 and Figure 4.13 demonstrate the initial region set and the representative regions based on color and texture separately. In order to demonstrate the exact cropped region, rectangles are used to identify the regions. As the clustering procedure can affect the results, each time the representative regions in the model may be slightly different. Because in the initial step of clustering the regions are randomly assigned to clusters, finally the results of representative regions may be slightly different.

The training time for different classes is shown in Figure 4.14. The platform is a PC with 512M RAM and Intel 1.8G CPU, visual studio environment. For most classes,

the training time is less than 0.5s. The three classes which use more training time are butterfly, face and airplane (side). Due to a larger number of detailed regions, training time is higher. The training based on texture feature is faster than color feature. The representative regions in the model based on color and texture features are determined separately. Eventually, these regions are combined to form the combination image model.

4.3.2.2 Basic object category recognition

The trained image model are used to classify the objects. Precision and recall are used to evaluate the results.

$$\begin{aligned} Precision_k &= \frac{A_k}{A_k + B_k} \\ Recall_k &= \frac{A_k}{A_k + C_k} \end{aligned} \qquad \text{(Eq. 4.27)}$$

where k is the number of total results, A_k is the number of correct results, B_k is the number of false results and C_k is the number of missed results.

The results of unweighted for regions in the model are shown in Table 4.4. The results of weighted regions are shown in Table 4.5. It can be seen that for most object classes the results are improved after the weights are applied. The classification results based on DCD and location are shown in Table 4.4a and Table 4.5a. The results based on HTD and location are shown in Table 4.4b and Table 4.5b. The classification results using a combination of all features, including color, texture and location of the regions are shown in Table 4.4c and Table 4.5c. For the car (side) class, as the image is monochrome, only the results of texture feature is presented.

It can be seen that the model can achieve better results for the objects with standard structures, such as face and motorbike. Although the trained image model contains only a few representative regions, the military aircraft and eagle have similar background but they can be easily differentiated. For the bobsled class, the results is not optimal due to the very large variation of direction and scale inside the class. For some object categories

with different colors of objects, the texture feature can obtain better results. When the combined distance is used, the results of all classes show an improvement.

Figure 4.15 shows the average time of classification for one image. Because the numbers of regions in the model for all classes are all 10, the classification time of different classes for one feature is very similar. To classify one image, the time is less than 0.01s. When the texture feature is used, the classification is only about 0.001s. As the classification based on combination features is the combination of the results based on color and texture separately, the time of combination is not evaluated.

Figure 4.16 shows some examples of the representative regions and the matching images. The regions in top two line of images are the representative regions. The image in last line is the matching image. The regions which are corresponding to the representative regions are marked in the corresponding colors. For two regions in a matched region pair, the same color is applied. Figure 4.17 show a match of an object with a more complex background. It can be seen that although the background affects the result, most of the regions are correctly detected and matched.

Table 4.4: Object classification results without weights of regions for all image categories.

4.4a: The results of using color features and location.

Category	Precision	Recall	Category	Precision	Recall
Military aircraft	82.2%	79.0%	Eagle	73.5%	71.0%
Car (rear)	76.9%	71.0%	Face	87.8%	79.2%
Motorbike	93.3%	82.3%	Bobsled	73.3%	70.0%
Airplane (side)	76.8%	72.0%	Watch	86.7%	76.0%
Butterfly	82.0%	79.0%	Car (side)	(grey images)	

4.4b: The results of using texture features and location.

Category	Precision	Recall	Category	Precision	Recall
Military aircraft	84.7%	81.0%	Eagle	76.8%	75.0%
Car (rear)	80.5%	78.7%	Face	91.5%	86.3%
Motorbike	94.5%	86.3%	Bobsled	77.3%	72.0%
Airplane (side)	81.0%	78.5%	Watch	82.5%	73.3%
Butterfly	74.0%	72.0%	Car (side)	83.3%	74.8%

4.4c: The results of combination.

Category	Precision	Recall	Category	Precision	Recall
Military aircraft	86.3%	81.6%	Eagle	79.4%	73.5%
Car (rear)	83.9%	82.3%	Face	92.5%	93.3%
Motorbike	95.5%	89.0%	Bobsled	78.2%	74.5%
Airplane (side)	84.0%	82.3%	Watch	88.3%	81.0%
Butterfly	82.6%	80.0%	Car (side)	(grey images)	

4.3.2.3 Object category recognition based on graph matching

The experiments of graph-based matching for the image model use the same dataset as in section 4.3.2.2. In order to reduce the computation complexity and demonstrate performance of graph-based matching, we reduce the number of regions in the model from 10 to 6. Figure 4.18 shows some examples of the six representative regions selected as the image model. Figure 4.19 shows some matching examples based on the graph structure. It can be seen that the corresponds of regions in the image model and images are correct and efficient. For the 10 categories of objects, the precision and recall based on color and texture information are shown in Table 4.6a and Table 4.6b separately. Although the number of regions in the model are reduced to 6, the results are still better

Table 4.5: Object classification results with the weights of regions for all image categories.

4.5a: The results of using color features and location.

Category	Precision	Recall	Category	Precision	Recall
Military aircraft	82.4%	79.5%	Eagle	72.7%	70.5%
Car(rear)	76.2%	70.3%	Face	89.5%	82.6%
Motorbike	94.6%	83.2%	Bobsled	71.8%	70.0%
Airplane (side)	83.2%	76.5%	Watch	81.2%	75.6%
Butterfly	81.0%	76.6%	Car(side)	(grey images)	

4.5b: The results of using texture features and location.

Category	Precision	Recall	Category	Precision	Recall
Military aircraft	84.9%	81.5%	Eagle	77.2%	75.0%
Car(rear)	82.0%	79.4%	Face	93.5%	89.3%
Motorbike	94.9%	86.5%	Bobsled	77.6%	72.0%
Airplane (side)	86.2%	79.5%	Watch	83.5%	74.2%
Butterfly	75.5%	72.0%	Car(side)	87.4%	79.5%

4.5c: The results of combination.

Category	Precision	Recall	Category	Precision	Recall
Military aircraft	89.2%	82.3%	Eagle	80.5%	75.2%
Car(rear)	84.1%	83.0%	Face	94.8%	93.7%
Motorbike	95.7%	89.6%	Bobsled	79.5%	74.9%
Airplane (side)	89.3%	84.5%	Watch	85.4%	78.0%
Butterfly	83.2%	79.2%	Car(side)	(grey images)	

than the simple comparison as shown in Table 4.4 and Table 4.5. Note that even if we can define different suitable weights for the representative regions in the model according to the corresponding appearance frequency, we cannot pre-define the weights of regions in the images properly. Thus we have to assign the same weights to the regions both in model and images. How to effectively improve the performance based on the suitable weights needs more investigation. For one image, it takes about 2-3 seconds to obtain the matching results in the same platform as described before. It depends on the number of regions in the images.

Table 4.6: Object classification results based on graph matching for all image categories.

4.6a: The results of using color features and location.

Category	Precision	Recall	Category	Precision	Recall
Military aircraft	83.1%	79.4%	Eagle	76.4%	73.2%
Car (rear)	80.5%	75.3%	Face	92.8%	83.5%
Motorbike	96.3%	84.6%	Bobsled	77.1%	74.2%
Airplane (side)	78.7%	74.8%	Watch	88.4%	79.3%
Butterfly	85.6%	81.2%	Car (side)	(grey images)	

4.6b: The results of using texture features and location.

Category	Precision	Recall	Category	Precision	Recall
Military aircraft	86.7%	83.2%	Eagle	79.8%	76.5%
Car (rear)	84.5%	81.3%	Face	93.5%	89.2%
Motorbike	95.8%	88.5%	Bobsled	79.3%	72.5%
Airplane (side)	88.5%	82.1%	Watch	88.2%	81.3%
Butterfly	83.6%	75.5%	Car (side)	89.3%	82.0%

4.11a: Class 1: Military aircraft

4.11b: Class 2: Motorbike

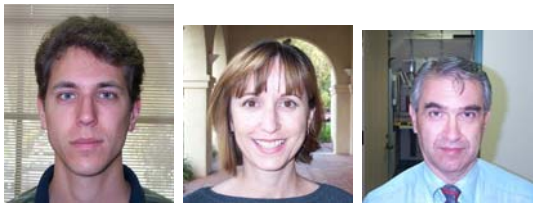
4.11c: Class 3: Bobsled

4.11d: Class 4: Butterfly

4.11e: Class 5: Car (rear)

4.11f: Class 6: Car (side)

4.11g: Class 7: Eagle

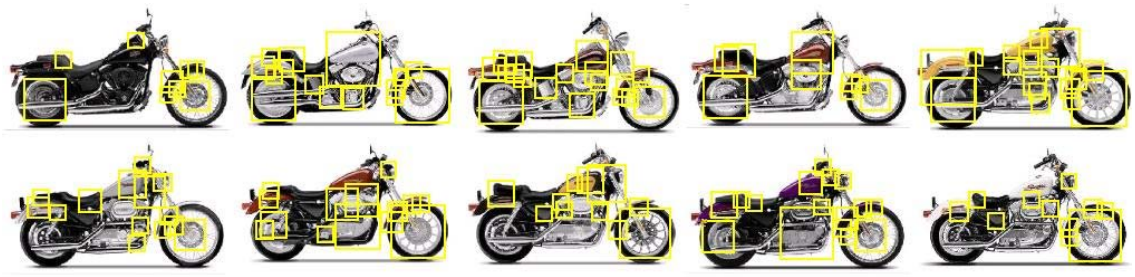


4.11h: Class 8: Face

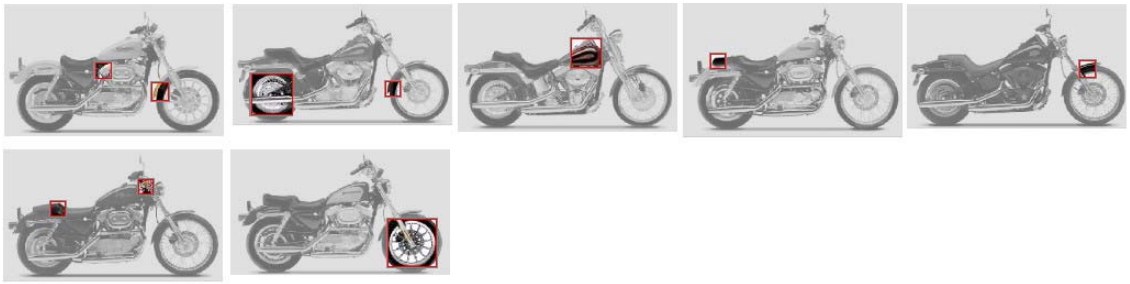
4.11i: Class 9: Airplane (side)

4.11j: Class 10: Watch

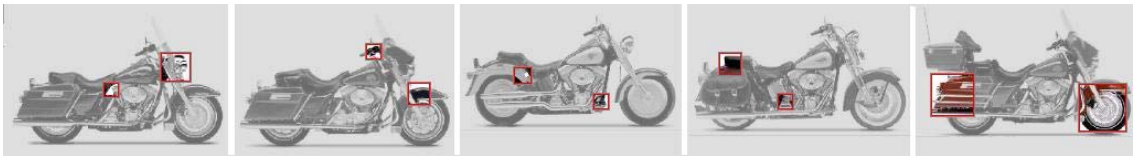
Figure 4.11: Some sample images from the datasets.



4.12a: The detected regions of the training images

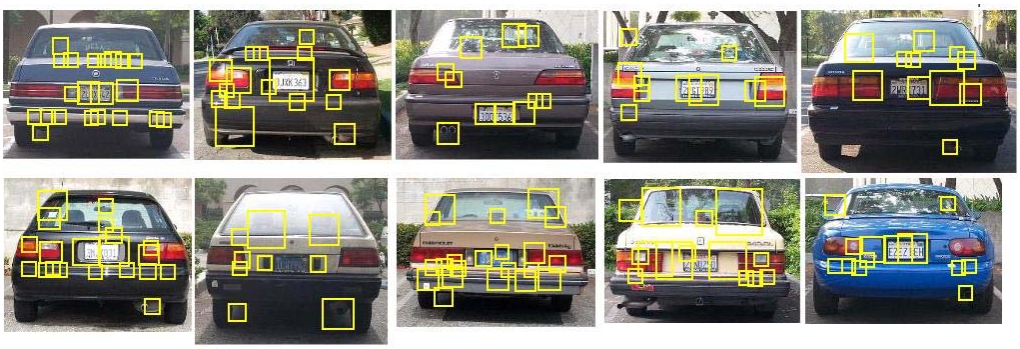


4.12b: The selected 10 representative regions based on color information

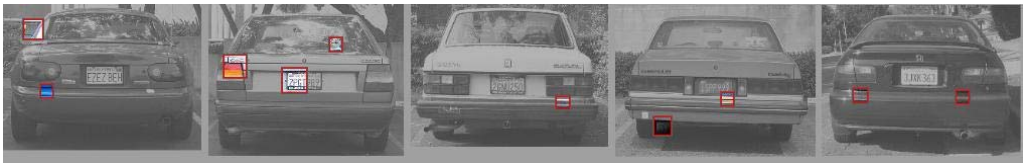


4.12c: The selected 10 representative regions based on texture information

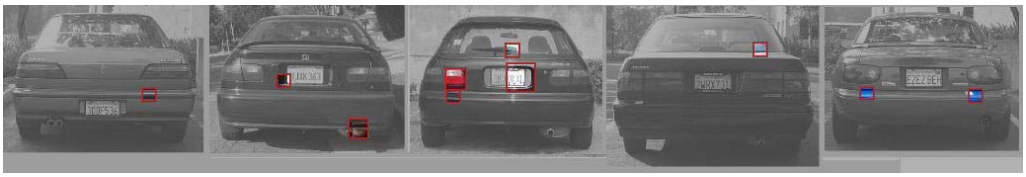
Figure 4.12: The original detected regions of motorbike and the representative regions based on color and texture information. The areas in the red blocks are cropped from the images as representative regions.



4.13a: The detected regions of the training images



4.13b: The selected 10 representative regions based on color information



4.13c: The selected 10 representative regions based on texture information

Figure 4.13: The original detected regions of car (rear) and the representative regions based on color and texture information. The areas in the red blocks are cropped from the images as representative regions.

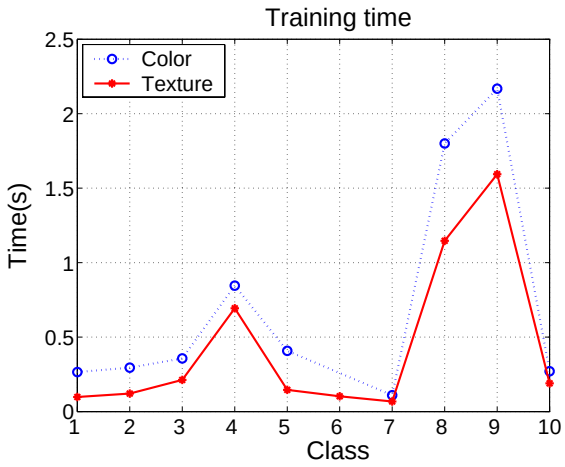


Figure 4.14: The training time for different classes. The numbers in x axis represent the 10 classes.

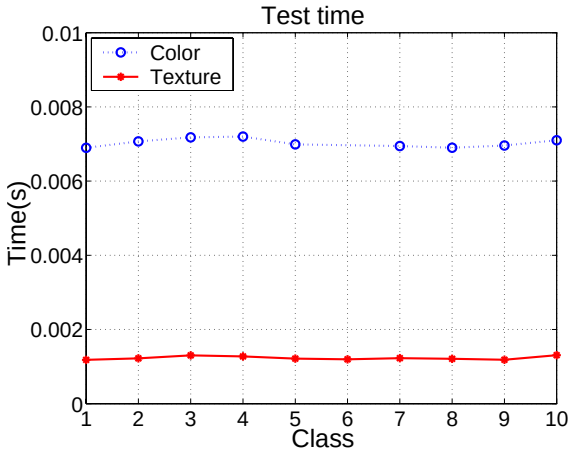
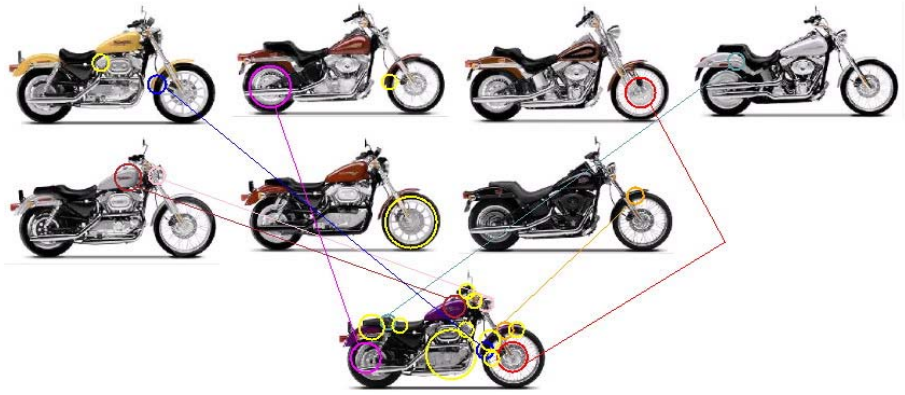
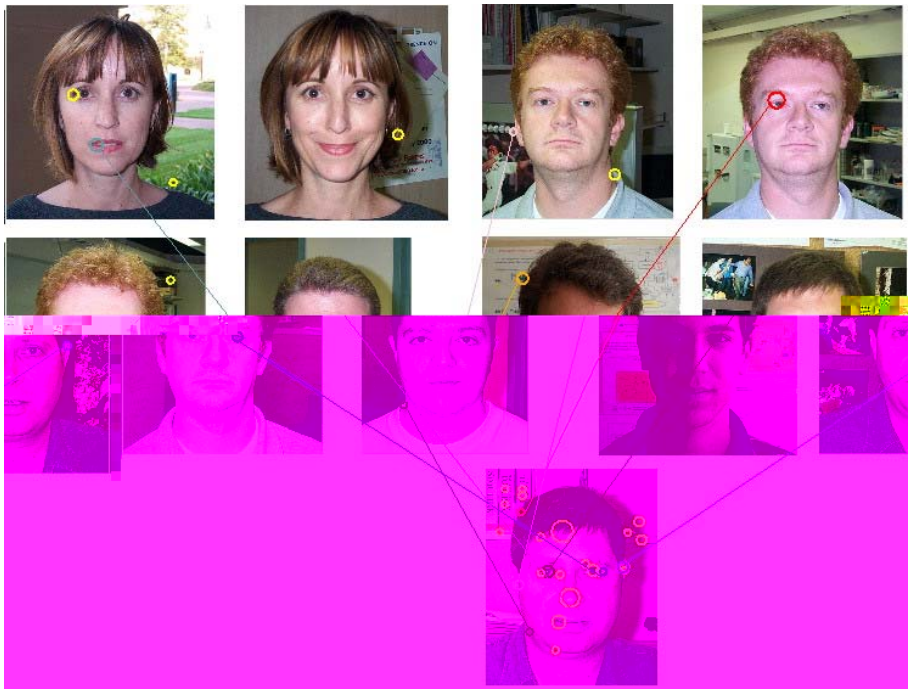


Figure 4.15: The classification time using the trained model for different classes. The numbers in x axis represent the 10 classes.

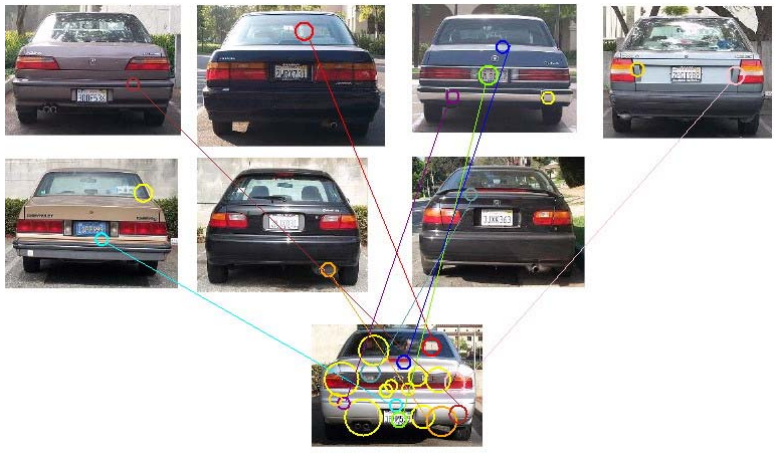


4.16a: The representative regions and a match example of motorbike



4.16b: The representative regions and a match example of face

Figure 4.16: The 10 representative regions in the model and matched examples. The image in last row is the matching image. The different colors of regions and lines demonstrate different corresponding matches.

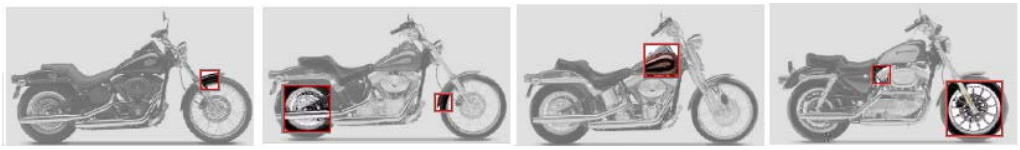


4.16c: The representative regions and a match example of car (rear)

Figure 4.16: The 10 representative regions in the model and matched examples. The image in last row is the matching image. The different colors of regions and lines demonstrate different corresponding matches. (con't)

Figure 4.17: The image model and matched example of a motorbike with a complex background. The different color regions and lines indicate the corresponding regions.

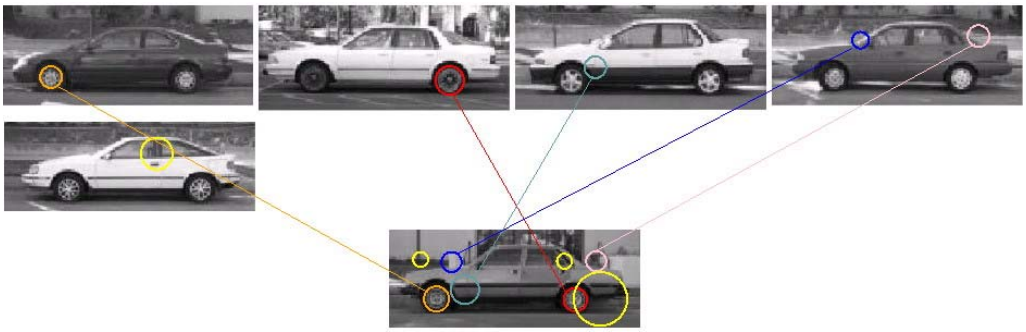
4.18a: The selected 6 representative regions based on color information of car (rear).



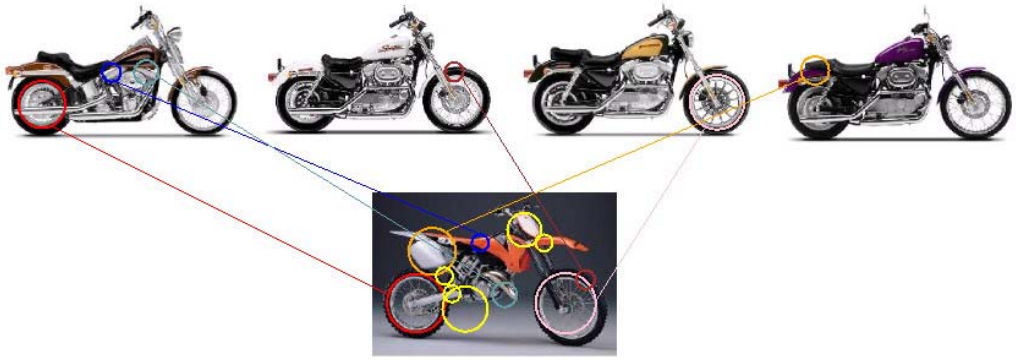
4.18b: The selected 6 representative regions based on texture information of motorbike.

4.18c: The selected 6 representative regions based on texture information of car (side).

Figure 4.18: The 6 representative regions based on color and texture information for some categories. The areas in the red blocks are cropped from the images as representative regions.



4.19a: The representative regions and a match example of car (side) based on texture features.



4.19b: The representative regions and a match example of motorbike based on color features.

Figure 4.19: The 6 representative regions in the model and matched examples based on graph. The image in last row is the matching image. The different colors of regions and lines demonstrate different corresponding matches.

4.4 Summary

Besides the whole image retrieval based on the low-level features, we want to identify the objects in the images. In this chapter, an initial work to use scale-invariant information in region-based retrieval is discussed. From the experimental results, it can be seen that fuzzy matching degree measure can greatly improve the retrieval performance, especially for small set of interest points, because the matching degree is more important when there are inadequate interest points to vote. Cross-correlation can further reject some mismatched images. How to obtain the matching points efficiently from very different numbers of points still needs more investigation.

After that a fast classification method of object category based on salient regions is introduced in this chapter. The classification is based on the representative regions of object categories. Firstly the salient regions in images containing objects are extracted. Then the most representative regions are selected according to the matching probability. These regions with suitable weights are used to build an image model. After training, this image model can be used for object classification and image retrieval. The spatial relationships between pairs of regions are calculated to improve the results.

In order to use the nested spatial relationships between the regions, we further introduce a graph-based matching algorithm to find the corresponding regions in the image model and images in the database. Each region is a vertex in the graph and the spatial relations between pairs of regions form the edges in the full-connected graph. An efficient graph matching algorithm based on EMD is investigated. EMD is applied in 2 steps to find the minimum cost, i.e., the distance between the two graphs. The graph based matching can fully make use of the relationships between the regions and better results can be achieved with less regions. For objects in a stable structure, the results are even better.

Chapter 5

Efficient sampling and its application to multimedia data

Currently the increasing size and diversity of multimedia information has stimulated the requirements for intelligent methods to analyze and manage data effectively. For a large image database, a good sampling algorithm can help us to efficiently manage and browse the images. Besides, the images in sample set can be easily tailored to different applications, such as finding more suitable training samples for machine learning and classification.

Another important issue related to the selection of training set is to reduce outliers or noise in the dataset. As real-world data is never perfect, the classifier can often suffer from corruptions (noise) that may impact the results based on the data. For example: (a) The feature values maybe erroneous or missing, and/or (b) the samples can be labeled with wrong classes. Noise can reduce system performance from several aspects, such as classification accuracy, training time, and the size of classifier. Accordingly, most existing training algorithms have applied various approaches to enhance their training abilities on noisy dataset, but the existence of noise can still introduce serious negative impacts [80]. A more reasonable solution is to employ some noise filter mechanisms to handle noisy data before a classier is trained. It is better if a sampling algorithm is able to distinguish the noise while selecting the training set from the original dataset.

In this chapter, firstly we briefly review the ε-approximation method and Epsilon-approximation method and Epsilon Approximation Sampling Enabled (EASE) algorithm. Then we introduce the new EASIER algorithm and analyze the performance *vis-a-vis* EASE. Finally the applications of EASIER for image and audio dataset are discussed.

5.1 Epsilon-approximation method

In this section we introduce the EASE sampling algorithm. We also analyze the limitations of EASE.

5.1.1 Notation

Firstly we introduce the notations used in describing the sampling algorithm. To be compliant with the predecessor EASE, the notation is according to the context of association rule mining.

Denote by D the database of interest, by S a simple random sample drawn without replacement from D, and by I the set of all items that appear in D. Let $N = |D|$, $n = |S|$, and $m = |I|$. Here $|\cdot|$ means the number of data or item, in the corresponding dataset or itemset. Also, denote by $\mathcal{I}(D)$ the collection of itemsets that appear in D; a set of items A is an element of $\mathcal{I}(D)$ if and only if the items in A appear jointly in at least one transaction $t \in D$. If A contains exactly $k(\geq 1)$ elements, then A is sometimes called a k-itemset. In particular, 1-itemsets are simply the original items. The collection $\mathcal{I}(S)$ denotes the itemsets that appear in S; of course, $\mathcal{I}(S) \subseteq \mathcal{I}(D)$. For $k \geq 1$, we denote by $\mathcal{I}_k(D)$ and $\mathcal{I}_k(S)$ the collection of k-itemsets in D and S, respectively.

For an itemset $A \subseteq I$ and a transactions set T, let $n(A;T)$ be the number of transactions in T that contain A. The support of A in D and in S is given by $f(A;D) = n(A;D)/|D|$ and $f(A;S) = n(A;S)/|S|$, respectively. Given a threshold $s > 0$, an item is frequent in D (respectively, in S) if its support in D (respectively, in S) is no less than

s. We denote by $L(D)$ and $L(S)$ the frequent itemsets in D and S, and $L_k(D)$ and $L_k(S)$ the collection of frequent k-itemsets in D and S, respectively.

For data sampling, the following notations are applied. Specifically, denote by S^i the set of all transactions in S that contains item A_i, and by r_i and b_i the number of red and blue transactions in S^i, respectively. Red means that the transactions will be kept in the final subsample and blue that means the transactions will be deleted. Q is the penalty function of r_i, and b_i. f_r denotes the ratio of red transactions, namely, the sample ratio. Then the ratio of blue transactions is given by $f_b = 1 - f_r$.

5.1.2 Epsilon-approximation method

From mathematic point of view, if for any instance of the problem, it delivers a solution with a relative discrepancy not exceeding epsilon, an approximation algorithm for an extremum problem is called ε-approximation algorithm. In order to obtain a good representation of a huge database, an ε-approximation method can be used to find a small subset, so that the supports of 1-itemset are close to those in the entire database. The sample S_0 of S is an ε-approximation if its discrepancy satisfies:

$$Dist(S_0, S) \leq \varepsilon. \qquad \text{(Eq. 5.1)}$$

The discrepancy is computed as the distance of 1-itemset frequencies between any subset S_0 and the superset S. It can be based on L_p-norm distances, for example [60]:

$$Dist_{L_1}(S_0; S) = \sum_{A \in I_1(S)} |f(A; S_0) - f(A; S)| \qquad \text{(Eq. 5.2)}$$

$$Dist_{L_2}(S_0; S) = \sum_{A \in I_1(S)} (f(A; S_0) - f(A; S))^2 \qquad \text{(Eq. 5.3)}$$

In this thesis $Dist_\infty$ metric is used as the distance metric and the discrepancy is calculated as follows:

$$Dist_\infty(S_0, S) = \max_{A \in I_1(S)} |f(A; S_0) - f(A; S)|. \qquad \text{(Eq. 5.4)}$$

5.1.3 EASE algorithm and its limitations [1]

Given an $\varepsilon > 0$, Epsilon Approximation Sampling Enabled (EASE) algorithm [1] is proposed to efficiently obtain a sample set S_0 which is an ε-approximation of S. The procedure of sampling is deterministically halves the data to get sample S_0. It will apply halving repeatedly ($S = S_1 => S_2 => \ldots => S_t(= S_0)$) until $Dist(S_0, S_1) \leq \varepsilon$.

Each halving step will introduce a discrepancy $\varepsilon_i(n_i, m)$ where n_i is the size of sub-sample S_i, and m is the total number of items in database.

Halving stops with the maximum t such that

$$\varepsilon_t = \sum_{i \leq t} \varepsilon_i(n_i, m) < \varepsilon \qquad \text{(Eq. 5.5)}$$

Finally, S_0 maintains the property that at any point S_0 is an ε-approximation for the set of currently seen transactions in S.

Specifically, S is obtained from the entire dataset D by using simple random sampling (SRS). SRS is the basic sampling technique where a group of subjects (a sample) is selected from a larger group (a dataset). Each individual is chosen entirely by chance and each member of the dataset has an equal chance of being included in the sample. Every possible sample of a given size has the same chance of selection [44, 45]. A repeated halving method keeps about half of the transactions in each round. Each halving iteration of EASE works as follows:

1. In the beginning, uncolor all transactions.

2. Color each transaction in S as red or blue. Red means that the transaction is selected in sample S_0 and blue means that the transaction is rejected.

3. The coloring decision is based on a penalty function Q_i for item A_i. The penalty function is based on hyperbolic cosine and has the shape depicted in Figure 5.1. Note that Q_i is low when $r_i = b_i$ approximately, otherwise Q_i increases exponentially in $|r_i - b_i|$.

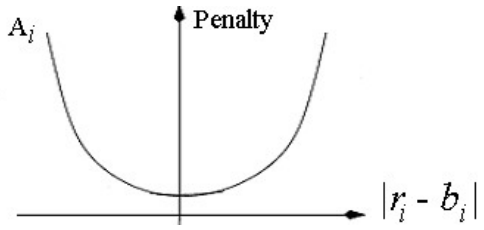


Figure 5.1: The penalty function for the halving method: penalty as a function of $|r_i - b_i|$.

The penalty function Q_i for each item A_i is converted into an exponential equivalent, shown as follows:

$$Q_i = 2\cosh(\delta_i(r_i - b_i)) \quad \text{(Eq. 5.6)}$$

$$= \exp(\delta_i(r_i - b_i)) + \exp(-\delta_i(r_i - b_i)) \quad \text{(Eq. 5.7)}$$

As for a small δ_i, $\exp(\delta_i)$ and $\exp(-\delta_i)$ can be replaced by $(1+\delta_i)$ and $(1-\delta_i)$, Eq. 5.7 can be modified to:

$$Q_i = Q_i^{(j)} = Q_{i,1}^{(j)} + Q_{i,2}^{(j)} \quad \text{(Eq. 5.8)}$$

$$Q_{i,1}^{(j)} = (1+\delta_i)^{r_i}(1-\delta_i)^{b_i} \quad \text{(Eq. 5.9)}$$

$$Q_{i,2}^{(j)} = (1-\delta_i)^{r_i}(1+\delta_i)^{b_i} \quad \text{(Eq. 5.10)}$$

where $Q_i^{(j)}$ means the penalty of ith item in jth transaction and δ_i controls the steepness of the penalty plot. The initial values of $Q_{i,1}$ and $Q_{i,2}$ are both 1.

Supposing the $(j+1)$-th transaction is colored as red (or blue), the corresponding penalty function $Q_i^{(j||r)}$ (or $Q_i^{(j||b)}$) is

$$Q_{i,1}^{(j||r)} = (1+\delta_i)Q_{i,1}^{(j)} \quad \text{(Eq. 5.11)}$$

$$Q_{i,2}^{(j||r)} = (1-\delta_i)Q_{i,2}^{(j)} \quad \text{(Eq. 5.12)}$$

$$Q_{i,1}^{(j||b)} = (1-\delta_i)Q_{i,1}^{(j)} \quad \text{(Eq. 5.13)}$$

$$Q_{i,2}^{(j||b)} = (1+\delta_i)Q_{i,2}^{(j)} \quad \text{(Eq. 5.14)}$$

The penalty function of the current transaction is the summation of penalties for all items. If $Q^{(j||b)} = \sum_i Q_i^{(j||b)}$ is less than $Q^{(j||r)} = \sum_i Q_i^{(j||r)}$, the $(j+1)$-th transaction will be colored blue and rejected. Otherwise, it will be colored red and added to the sample. The initial value of δ_i is $\sqrt{1 - \exp\left(-\ln(2m)/n\right)}$, where m is the number of items in original dataset and n is the initial sample size. EASE maintains the property that S_0 is always an ε-approximation of the current transaction set S. The details can be found in [1].

However, EASE has some limitations.

- Due to its halving nature, EASE has certain granularity in sample ratio, i.e., the sample ratio must be 0.5, 0.25 or 0.125 etc. In [1] an ad hoc solution was proposed where the size of the initial sample is so chosen that by repeated halving of several rounds, one obtains the required size. It means that if a sample ratio 0.3 is wanted, a initial sample set of sample ratio 0.6 will be randomly selected from original dataset firstly. Then EASE is applied to initial sample set to obtain a half sample set, i.e., sample ratio 0.3 of the original dataset.

- In order to obtain a desired sample ratio, EASE may run several iterations of halving procedure. This will increase the computation time. The idea of choosing a large simple random sample initially, before applying the halving method of EASE, is motivated by the total time taken to produce the final sample. If the halving method is applied on the whole data, the accuracy will be better, but time taken will be more than when halving is done starting from an initial simple random sample.

- EASE was built especially for categorical count data, for example, market basket data (the purchase records of a supermarket). It was natural to extend it to continuous data. A simple quantization can be applied to transform the original continuous data of both image and audio applications to binary format.

- EASE cannot handle noise. Experimental results in Section 5.5 show that the percentage of noise remains the same for the sample set as for the original set. The reason for this behavior is that due to its halving nature, EASE keeps half of everything, including the noise.

5.2 New and modified EASE: EASIER

EASE is a good sampling algorithm that outperforms SRS, but it has some disadvantages. In this section we analyze the problems of EASE in detail and propose the new algorithm EASIER to avoid these problems.

5.2.1 EASIER sampling: without halving

As mentioned in previous section, in EASE, the halving process has certain granularity. It can only compute a subset that is approximately half the size of S. If a different sample ratio is wanted, we have to run the halving procedure several times with a proper initial random sample set S of dataset D. Moreover, this will consume more time and memory due to multiple halving iterations.

In order to directly obtain a sample set of any sample ratio in one pass, the halving round is modified to select red transactions with a probability which is proportional to the desired final sample size. This will remove the need to store several levels of penalties. If we want to obtain a sample set from S with sample ratio r_s directly, the ratio of red transactions is $f_r = r_s$ and the ratio of blue transactions is $f_b = 1 - r_s$. Then we have $r_i = f_r \cdot |S^i|$ and $b_i = f_b \cdot |S^i|$. So $\frac{r_i}{f_r} = \frac{b_i}{f_b} = |S^i|$. As $\frac{r_i}{f_r} + \frac{b_i}{f_b} = 2|S^i|$, we use $\frac{r_i}{2f_r} = \frac{b_i}{2f_b}$ and thus $\frac{r_i}{2f_r} + \frac{b_i}{2f_b} = |S^i|$. As the objective of the halving method is to minimize $|r_i - b_i|$, and $r_i + b_i = |S^i|$ for each item i, our new method will be modified to minimize

$$\left| \frac{r_i}{2f_r} - \frac{b_i}{2f_b} \right|$$

instead of $|r_i - b_i|$.

The modified penalty Q_i of jth transaction will be

$$Q_i = Q_i^{(j)} = Q_{i,1}^{(j)} + Q_{i,2}^{(j)} \quad \text{(Eq. 5.15)}$$

$$Q_{i,1}^{(j)} = (1+\delta_i)^{\frac{r_i}{2f_r}}(1-\delta_i)^{\frac{b_i}{2f_b}} \quad \text{(Eq. 5.16)}$$

$$Q_{i,2}^{(j)} = (1-\delta_i)^{\frac{r_i}{2f_r}}(1+\delta_i)^{\frac{b_i}{2f_b}}. \quad \text{(Eq. 5.17)}$$

Supposing that the $(j+1)$th transaction is colored as r (or b), the corresponding penalty function $Q_i^{(j||r)}$ (or $Q_i^{(j||b)}$) in Eq. 5.11 is changed to

$$\begin{aligned} Q_{i,1}^{(j||r)} &= (1+\delta_i)^{\frac{r_i+1}{2f_r}}(1-\delta_i)^{\frac{b_i}{2f_b}} \\ &= (1+\delta_i)^{\frac{1}{2f_r}}(1+\delta_i)^{\frac{r_i}{2f_r}}(1-\delta_i)^{\frac{b_i}{2f_b}} \\ &= (1+\delta_i)^{\frac{1}{2f_r}} Q_{i,1}^{(j)} \end{aligned} \quad \text{(Eq. 5.18)}$$

$$\begin{aligned} Q_{i,2}^{(j||r)} &= (1-\delta_i)^{\frac{1}{2f_r}} Q_{i,2}^{(j)} \\ Q_{i,1}^{(j||b)} &= (1-\delta_i)^{\frac{1}{2f_b}} Q_{i,1}^{(j)} \\ Q_{i,2}^{(j||b)} &= (1+\delta_i)^{\frac{1}{2f_b}} Q_{i,2}^{(j)}. \end{aligned} \quad \text{(Eq. 5.19)}$$

The computation process of $Q_{i,1}^{(j||r)}$is given in Eq. 5.18. Other penalty functions are computed with a similar procedure and the results are shown in Eq. 5.19. The overall penalty is calculated as described in section 5.1.3.

For $Q_i^{(final)}$, we cannot guarantee that $Q_i^{(final)} \leq 2m$. As δ_i is a very small value, $(1+\delta_i)^{\frac{1}{2f_r}}$ and $(1+\delta_i)^{\frac{1}{2f_b}}$ are both close to 1. So $Q_i^{(final)}$ is close to $2m$. According to [1] the value of $\left|\frac{r_i}{2f_r} - \frac{b_i}{2f_b}\right|$ is close to

$$\frac{\ln(2m)}{\ln(1+\delta_i)} + \frac{|S^i|\ln(1/(1-\delta_i^2))}{\ln(1+\delta_i)}.$$

Therefore, the same δ_i, as in Section 5.1.3, is used in the new algorithm. In Algorithm 3 the completed EASIER algorithm is given. The penalty for each item i of a transaction is calculated only once. So, it does not need to store the penalty for each halving iteration,

and thus results in a reduction of memory from $O(mh)$ for EASE to $O(m)$. The time for processing one transaction is bounded by $O(T_{max})$ for EASIER, whereas EASE requires $O(hT_{max})$, where T_{max} denotes the maximal transaction length in T. Thus, unlike EASE, EASIER is independent of sample size.

5.2.2 Handling noise

Here, we consider one type noise that often occurs in transactional data. In transactional data, each transaction has several items. A noisy transaction has some of its actual (nonnoisy) items deleted and some noisy items inserted. Typically, the frequencies of these noisy items are low compared to those of the nonnoisy items. This scenario of transactional data can be easily extended to other multimedia data, such as image and audio, as is verified in our experiments.

Due to the halving nature of EASE, in each halving the items are evenly divided into two parts. As half of the noisy items are also kept in the sample set during each halving, EASE is not able to discriminate between noisy and nonnoisy items. The experiments in Section 5.5 demonstrate this phenomenon. SRS has a similar problem. As SRS randomly selects the sample set, the noisy data in the sample set has almost the same percentage as the original data when the experiment repeats several times.

However, the proposed EASIER is able to avoid this problem in an elegant manner. For simplicity purposes, consider that each transaction holds only one item. This assumption can be generalized to multiple items. In Eq. 5.18 and Eq. 5.19, the sample ratio f_r is typically very small, that is, $f_r \ll f_b$. The associated penalties for painting a transaction red or blue are influenced by the values of f_r and f_b. This effect can be offset only by the two other variables in Eq. 5.18 and Eq. 5.19, that is, r_i and b_i. That means the transactions have to be painted in such a manner that the effect $f_r \ll f_b$ is offset. This is possible only by painting a sufficient number of transactions as blue before painting a transaction as red. When the item is nonnoisy, there will be a sufficient number

of transactions so as to ensure that some transactions holding that item are eventually painted red, i.e., selected. But when the item is noisy, usually there is not sufficient number of transactions so as to ensure that even one transaction holding that item is painted red. This is particularly so when the sample ratio f_r is very small.

5.2.3 Comparison of other EASE-related algorithms

In [81] two extensions of EASE are presented, Biased-EA and Biased-L2. Unlike EASIER, Biased-EA attempts to minimize the difference between the current item count ($r_i = f_r \cdot |S^i|$) in the sample obtained so far and the expected count of items in the sample $f_r \cdot n_i$, where $n_i = r_i + b_i$ is the count of the item encount red so far. Thus

$$\begin{aligned} \left|\frac{r_i}{2f_r} - \frac{b_i}{2f_b}\right| &= \left|\frac{f_b \cdot r_i - f_r \cdot b_i}{2f_r \cdot f_b}\right| = \left|\frac{(1-f_r)\cdot r_i - f_r \cdot b_i}{2f_r \cdot (1-f_r)}\right| \\ &= \left|\frac{r_i - f_r \cdot (r_i + b_i)}{2f_r \cdot (1-f_r)}\right| = \frac{|r_i - f_r \cdot n_i|}{2f_r \cdot (1-f_r)}. \end{aligned} \qquad \text{(Eq. 5.20)}$$

Therefore the idea of Biased-EA is similar to EASIER, but the detail implementation is different. Another extension is Biased-L2. Compared with Biased-EA, Biased-L2 uses an L_2-norm instead of the L_1-norm in Biased-EA and the δ_i is reduced by adjusting the format of the penalty function.

5.2.4 Various applications of EASIER

EASIER can efficiently select a representative sample set from a large database based on a set of features dynamically. It is a suitable sampling algorithm to select the training set for multimedia classification. We validate and compare the output samples of EASE[1], SRS, and EASIER by performing classification of continuous multimedia data. For image data, a support vector machine (SVM) is used as the image classifier due to

[1]There exist a few variants of EASE. Owning to time constraint and the goal of generalization, we mainly use EASE for comparison.

its high classification accuracy and strong theoretical foundation [82]. The results of the SVM classifier show that EASIER samples outperform SRS samples in the accuracy and EASE samples in the time to perform sampling. EASIER achieves the same or even better accuracy than EASE. For audio data, audio event identification is considered as a classification problem. In [5, 83], the research on audio event identification by using Hidden Markov Models has been presented. Previously the SRS samples are selected to train an audio event identifier without considering sample efficiency and computation time. Experimental results shows that EASIER outperforms SRS and EASE greatly, especially when the sample ratio is very small.

Besides the classification of multimedia data, another important data mining task, namely association rule mining, is performed to evaluate the performance of EASIER. The association rule mining results for IBM QUEST dataset [84] are reported to compare with the earlier reported results [1]. For the small sample ratio, the accuracy of EASIER is better than EASE and the computation is very fast.

Algorithm 3 : EASIER Sampling

Input: D, n, m, f_r
Output: S_0, the transactions in red color

for each item i in D **do**
  $\delta_i = \sqrt{1 - \exp\left(-\frac{\ln(2m)}{n}\right)}$;
  $Q_{i,1} = 1$;
  $Q_{i,2} = 1$;
end for
for each transaction j in S **do**
  color transaction j red;
  $Q^{(r)} = 0$;
  $Q^{(b)} = 0$;
  for each item i contained in j **do**
    $Q_{i,1}^{(r)} = (1 + \delta_i)^{\frac{1}{2f_r}} Q_{i,1}$;
    $Q_{i,2}^{(r)} = (1 - \delta_i)^{\frac{1}{2f_r}} Q_{i,2}$;
    $Q_{i,1}^{(b)} = (1 - \delta_i)^{\frac{1}{2f_b}} Q_{i,1}$;
    $Q_{i,2}^{(b)} = (1 + \delta_i)^{\frac{1}{2f_b}} Q_{i,2}$;
    $Q^{(r)} += Q_{i,1}^{(r)} + Q_{i,2}^{(r)}$;
    $Q^{(b)} += Q_{i,1}^{(b)} + Q_{i,2}^{(b)}$;
  end for
  if $Q^{(r)} < Q^{(b)}$ **then**
    $Q_{i,1} = Q_{i,1}^{(r)}$;
    $Q_{i,2} = Q_{i,2}^{(r)}$;
  else
    color transaction j blue;
    $Q_{i,1} = Q_{i,1}^{(b)}$;
    $Q_{i,2} = Q_{i,2}^{(b)}$;
  end if
  if transaction j is red **then**
    set $S_0 = S_0 + \{j\}$;
  end if
end for

5.3 EASIER for image application

In this section we describe our approach in applying EASIER to image application. We apply EASIER to select representative images based on color structure descriptor. The selected samples are used as the training data of object classification.

5.3.1 Color Structure Descriptor

As described in Section 3.2.2.1 Color Structure Descriptor (CSD) is one color descriptor in MPEG-7. It is defined to represent images by both their color distribution (like color histogram) and the local spatial structure of the color [18]. We apply EASIER to huge image databases to select the representative samples. The image feature we used are CSD. In our experiments, 256-bin CSD is used. Each CSD descriptor has 256 bins and each bin has an 8-bits numerical value (0-255). Because EASIER is based on calculation of the frequency of each item, the format of CSD is changed as shown in Figure 5.2. Firstly, all numerical values of CSD are binarized. The binary vector for a CSD value has a length of 256 and contains all 0's, except for a 1 in the position corresponding to the numerical value. After that, each nonzero value in this binary vector is converted back into a new discrete value depending on its position. This new vector represents the data and is used for sampling with EASIER. It is found that this vector is usually length 100 (it indirectly corresponds to about 100 nonzero bins for each CSD descriptor). After mapping, there will be a total of $256 \times 256 = 65,536$ (one-bit) items.

In order to reduce the number of items in the data set, the 8-bit bin value of the original CSD is requantized into a 4-bit-quantized representation. The requantization is nonuniform and we use the suggested CSD amplitude quantization table in the MPEG-7 standard [4]. This will effectively reduce the number to $16 \times 256 = 4,096$ items for each CSD descriptor. A smaller number of bits to represent the data will result in a significant reduction in the number of items for each CSD descriptor. In order to test the efficiency

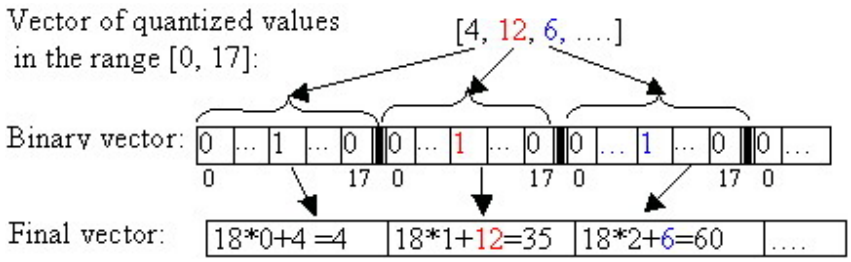


Figure 5.2: An example of the format modification of features.

of 16D CSD, CSDs in both format are used to retrieve the same query image in the same image database. On average, there are 16 images that are in the top 20 images obtained using both quantized data and original data. Experimental results have shown that the retrieval accuracy of the quantized data is close to that using the original CSD data.

5.3.2 EASIER for image classification

To verify the representativeness of the samples, a classification algorithm is needed to test the classification performance based on different training sample sets. For the choice of classification algorithm, SVM is a good candidate for image classification applications. As SVM maximizes the classification margin to obtain the results, the problem of overfitting generated by the huge number of image features is alleviated. Furthermore, the kernel functions of SVM can be complicated nonlinear functions [85]. The SVM is trained by the sample sets selected by EASE, EASIER, and SRS. The remaining images are used for testing the SVM. We use COIL-100[2] as the image database because it has the predefined ground-truth set. The database includes 7,200 color images of 100 objects: For each object there are 72 images where each image is taken at pose intervals of 5 degrees.

[2]http://www1.cs.columbia.edu/CAVE/research/softlib/coil-100.html

5.4 EASIER for audio application

In this section we present an audio application to demonstrate the effect of EASIER sampling. Audio, including voice, music, and various kinds of environmental sounds, is an important type of media and also a significant part of video. Recently, people have begun to realize the importance of effective audio content analysis which provides important cues for semantics [86, 87, 88]. Effective audio analysis techniques provide convincing results. In consideration of computational efficiency, some research efforts have been made for audio content analysis [86, 89]. SRS samples may not represent the characteristics of all audio sequence particularly for small sample ratios. Experiments show that even when 10% random samples are used for training, the classification performance is still very low for SRS, especially for small classes. Therefore, to obtain stable results and achieve better performance using a small size of training data, we propose EASIER sampling algorithm to select representative samples.

5.4.1 Audio event identification

Audio events are defined as some specific audio sounds which have strong hints to interesting video events or highlights. Especially in sports video, some game-specific audio sounds (e.g., excited audience sounds, excited commentator speech, etc.) have strong relationships to the actions of players, referees, commentators, and audience. In this section, we use basketball audio to demonstrate how effectively and efficiently the EASIER work.

In [5], the research on audio event identification using Hidden Markov Models (HMM) is presented. The samples to train audio event identifier are randomly selected without considering sample efficiency and computation time for training. Since the audio database is large, we have to consider the following two issues: 1) When audio data comes from various audio sequences, SRS may only select part of the audio sequences. It is not a

good representative of all sequences; 2) There is a tradeoff between the size of training dataset and accuracy, as the learning-curve sampling method described in Section 2.5.

5.4.1.1 Basketball audio events identification

Basketball games have compact structure. Generally, the offence and defense actions, which are the highlights of a basketball game, take place on an alternating basis. These highlights, which attract most audience's interests, are significant and should be detected for future basketball video editing. Fortunately, excited commentator speech and audience sounds play important roles in highlight detection of sports video. Therefore, basketball audio event identification focuses on identifying excited commentator speech (EC) and excited audience sounds (EA). Besides EC and EA, there are two other basketball audio events: plain commentator speech (PC) and plain audience sounds (PA). These four kinds of events almost cover the full basketball game. Out of these four, EA and EC are smaller classes. A classification task is to classify audio samples into these four predefined classes. There are some other audio events in basketball games, such as whistling, etc., that have small number of samples and are easy to identify. In order to test the efficiency of the proposed sampling algorithm, we use only these four classes.

5.4.1.2 HMM-based audio event identification

Audio signal exhibits consecutive changes in values over a period of time, where variables may be predicted from earlier values. In other words, strong context exists in audio data. In consideration of the success of HMM in speech recognition, we propose our HMM-based audio event generation system. The proposed system includes three stages, namely feature extraction, data preparation and HMM learning, as illustrated in Figure 5.3. Selected low-level features are extracted from audio streams and tokens are added to create observation vectors. This data is then separated into two sets for training and

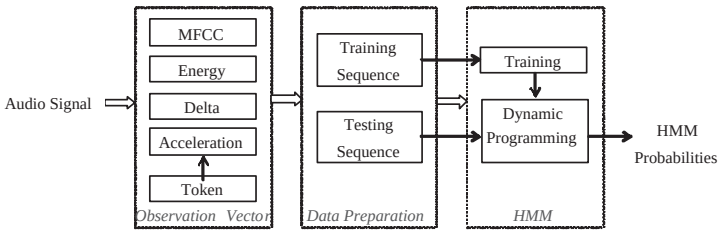


Figure 5.3: Audio events generation system [5].

testing. After that, HMM is trained and then reestimated by using dynamic programming. Finally, according to the maximum posterior probability, the audio keyword with the largest probability is selected to label the corresponding testing data. Details can be found in [5, 90].

5.4.2 EASIER for audio event identification

We apply EASIER to find representative training samples according to the low-level features. In our experiments, we segment audio signal at 20ms per frame, which is the basic unit for feature extraction. Mel-Frequency Cepstral Coefficient (MFCC) and energy are selected as the low-level audio features as they are successfully used in speech recognition and further have be proven efficient for audio keyword generation. Delta and acceleration are further used to accentuate signal temporal characters for HMM. Therefore, 39-dimension (39D) feature vectors are composed. Because the amplitude of an audio signal is continuous and EASIER is based on calculation of the frequency of each item, the format of the features is changed. Firstly, the continuous values are nonuniformly quantized to a range [0, 17]. Then, this discretized data is binarized as shown in Figure 5.2. Similar to CSD, the binary vector has length 18 and it contains all zeros except for a one in the position corresponding to the discretized value. After that, each nonzero value in this binary vector is converted back into a new discrete value

considering its position. This new vector represents the data and is used for sampling with EASIER.

Although experimental results show that using all features gives high accuracy, we experimented with reduced dimensionality that requires less space. In order to reduce the number of items, the most dominant 13 dimensions (13D) of feature vectors are used. These 13 dimensions represent MFCC and energy of the audio signal. Their efficacy in audio analysis has already been shown in [83]. Note that we only use 13D data to select the samples and the original 39D data of the corresponding 13D samples is used in the training and test of HMM. Our experimental results have shown that the classification performance of 13D data is close to that of the original data.

5.5 Experimental results

In this section we compare the performance of EASIER, EASE, and SRS in the context of classification and association rule mining. Both real-world data (CSDs of image database and audio features) and synthetic data (IBM QUEST data) are used for testing.

Primary metrics used for the evaluation are: (a) performance metric of different applications, (b) sampling time, and (c) memory requirements. Sampling time is the time taken to obtain the final samples. For the performance of image classification, since the classifier is SVM, the correct rate of SVM classification is used as the performance metric. For audio event identification, the performance of is measured using precision and recall.

For association rule mining, since it is focused on finding frequent itemsets, namely, the itemsets satisfying the minimum support [84, 91, 92], we use the following performance metric to measure the accuracy of the sampling algorithms:

$$accuracy = 1 - \frac{|L(D) - L(S)| + |L(S) - L(D)|}{|L(D)| + |L(S)|}, \qquad \text{(Eq. 5.21)}$$

where, as before, $L(D)$ and $L(S)$ denote the frequent itemsets from the database D and sample S, respectively. We used Apriori [84] to compare the three algorithms in a fair manner. It computes the frequent item sets for related experiments. A publicly available version of Apriori, written by Christian Borgelt[3], is used.

5.5.1 Image application

As mentioned before, a COIL image database which is composed from 72 class images is used to extract CSD. Thus there are totally 7200 descriptors. To verify the effect of processing outliers, 5% and 10% noisy data are added to the CSD dataset separately. The noise is a series of Gaussian white noise generated by Matlab.

All three sampling algorithms, EASIER, EASE, and SRS, start from the whole dataset to obtain the samples. As the halving process has a certain granularity and EASE cannot achieve the specific sample ratio in halving, in each iteration we first ran EASE with a given number of halving times. Then we use the actual sample ratio from the EASE samples to generate EASIER samples. As EASIER is probabilistic, it does not guarantee the exact sample size. Hence, the actual EASIER sample size is used to generate the SRS sample. Note that although EASE and EASIER sample sizes are not exactly the same, the difference is very little.

For image classification, because EASE is based on halving, the final sample ratios of training sets are predetermined to be 0.5, 0.25, 0.125, and 0.0625. Accuracy of classification is used as the classification metric. We also compare the samples using the accuracy of association rule mining. As one to seven times halving are applied, the sample ratios are 0.5, 0.25, 0.125, 0.0625, 0.03125, 0.015625, and 0.0078125.

Since the results of EASIER and EASE change with the input sequence of data, for each sample ratio we run EASIER and EASE 50 times, and in each iteration the

[3]http://fuzzy.cs.uni-magdeburg.de/~borgelt/apriori.html

input descriptors are shuffled randomly. SRS is also run 50 times over this shuffled data. The association rule mining results of EASIER, EASE, and SRS are computed as an average over these 50 runs. The minimum support value for Apriori is set to 0.77% and we evaluated only the 1-item frequent set for CSD data, as otherwise there are too many frequent itemsets. However, for IBM QUEST transaction data, we use all frequent itemsets. The image classification results are the average of 10 runs.

Table 5.1 and 5.2 straightforwardly demonstrate the effect of removing noise by applying the three methods, EASE, EASIER, and SRS. The whole dataset includes 7,200 CSDs and several subsets are also used for test. It can be seen that when the sample rate is 0.5, about half of the noise is kept in the training sample. As the sample ratio of EASE is always 0.5 for each halving, 50% noise is kept in the sample set of EASE. The results of SRS are similar. For EASIER, when the sample ratio is 0.5, it works identically to EASE. When the sample ratios are reduced, the amount of noisy data in the training sample is reduced greatly. Almost none of the noisy data is included in the training sample. Hence, EASIER can reduce noisy data efficiently when the training set is selected from the whole dataset.

The results of original CSD data (65,536 items) are shown in Figure 5.4. Figures 5.4a, 5.4b and 5.4c show the correct classification rate of image classification with nonnoisy, 5% and 10% noisy data, respectively. Figures 5.4d, 5.4e and 5.4f show the different accuracy of association rule mining respectively. EASIER achieves better accuracy than EASE when the dataset has noise. For data with noisy, the results of EASIER are much better than EASE and SRS for both classification and association rule mining. For example, for a sample ratio of 0.0625 with 10% noise, the proposed sampling method achieves 82.5% classification rate, while the correct rate of EASE is 77.3%, and SRS achieves only 51.7%. For nonnoisy data, EASIER achieves similar accuracy as EASE which is better than SRS for both classification and association rule mining, especially when the sample

Table 5.1: The average number of noise in selected training samples, and the corresponding percentage against total number of noisy data of original CSD data, which is extracted from COIL image set. The sample ratios are 0.5, 0.25, 0.125 and 0.0625, respectively.

5.1a: EASIER

Number of dataset	Noise	The number and percentage of noise for different ratios							
		0.5		0.25		0.125		0.0625	
7200	720 (10%)	358.4	49.78%	59	8.19%	0	0.00%	0	0.00%
7200	360 (5%)	176.3	48.97%	17	4.72%	0	0.00%	0	0.00%
1440	144 (10%)	71.48	49.64%	5.5	3.82%	0	0.00%	0	0.00%
1440	72 (5%)	35.46	49.25%	0	0.00%	0	0.00%	0	0.00%
288	29 (10%)	14.4	49.66%	0	0.00%	0	0.00%	0	0.00%
288	15 (5%)	7.3	48.67%	0	0.00%	0	0.00%	0	0.00%

5.1b: EASE

Number of dataset	Noise	The number and percentage of noise for different ratios							
		0.5		0.25		0.125		0.0625	
7200	720 (10%)	356.4	49.5%	179.2	24.69%	88.9	12.35%	44.3	6.15%
7200	360 (5%)	178.3	49.53%	89.3	24.81%	43.8	12.17%	22.5	6.25%
1440	144 (10%)	70.7	49.10%	35.9	24.93%	17.7	12.29%	9.2	6.39%
1440	72 (5%)	35.8	49.72%	18.1	25.14%	9	12.50%	4.5	6.25%
288	29 (10%)	14.3	49.31%	7.1	24.48%	3.3	11.38%	1.4	4.83%
288	15 (5%)	7.4	49.33%	3.6	24.00%	1.4	9.33%	0.7	4.67%

5.1c: SRS

Number of dataset	Noise	The number and percentage of noise for different ratios							
		0.5		0.25		0.125		0.0625	
7200	720 (10%)	361.0	50.17%	179.2	24.69%	88.9	12.35%	44.3	6.15%
7200	360 (5%)	180.4	50.11%	89.3	24.81%	43.8	12.17%	22.5	6.25%
1440	144 (10%)	72.8	50.56%	36.4	25.28%	17.5	12.15%	9.5	6.60%
1440	72 (5%)	36.1	50.14%	17.7	24.58%	9.2	12.78%	4.4	6.11%
288	29 (10%)	14.5	50.00%	7.3	25.17%	3.7	12.76%	1.8	6.21%
288	15 (5%)	7.6	50.67%	3.8	25.33%	1.9	12.67%	0.9	6.00%

Table 5.2: The average number of noise in selected training samples, and the corresponding percentage against total number of noisy data of requantized CSD data. The sample ratios are 0.5, 0.25, 0.125 and 0.0625, respectively.

5.2a: EASIER

Number of dataset	Noise	The number and percentage of noise for different ratios							
		0.5		0.25		0.125		0.0625	
7200	720 (10%)	355.3	49.32%	20	5.56%	0	0.00%	0	0.00%
7200	360 (5%)	179.7	49.92%	0	0.00%	0	0.00%	0	0.00%
1440	144 (10%)	71.0	49.31%	0	0.00%	0	0.00%	0	0.00%
1440	72 (5%)	35.8	49.72%	0	0.00%	0	0.00%	0	0.00%
288	29 (10%)	14.2	48.96%	0	0.00%	0	0.00%	0	0.00%
288	15 (5%)	7.4	49.33%	0	0.00%	0	0.00%	0	0.00%

5.2b: EASE

Number of dataset	Noise	The number and percentage of noise for different ratios							
		0.5		0.25		0.125		0.0625	
7200	720 (10%)	355.2	49.32%	175.6	24.38%	88.4	12.28%	44.7	6.21%
7200	360 (5%)	179.3	49.81%	88.6	24.61%	42.9	11.97%	21.8	6.06%
1440	144 (10%)	70.8	49.17%	35.6	24.72%	17.1	11.89%	8.9	6.18%
1440	72 (5%)	35.5	49.31%	17.6	24.44%	8.9	12.36%	4.4	6.11%
288	29 (10%)	14.1	48.62%	7.2	24.82%	3.5	12.07%	1.5	5.17%
288	15 (5%)	7.3	48.66%	3.7	24.67%	1.5	10.00%	0.8	5.33%

5.2c: SRS

Number of dataset	Noise	The number and percentage of noise for different ratios							
		0.5		0.25		0.125		0.0625	
7200	720 (10%)	358.1	49.74%	178.5	24.72%	85.5	11.82%	44.5	6.18%
7200	360 (5%)	178.3	49.53%	89.6	24.89%	45.3	12.58%	22.6	6.28%
1440	144 (10%)	72.6	50.42%	36.2	25.14%	17.7	12.29%	9.2	6.39%
1440	72 (5%)	36.4	50.56%	17.9	24.86%	9.1	12.64%	4.5	6.25%
288	29 (10%)	14.6	50.34%	7.4	25.52%	3.5	12.07%	1.9	6.55%
288	15 (5%)	7.5	50.00%	3.9	26.00%	1.8	12.00%	0.9	6.00%

ratio is very small. For example, for a sample ratio of 0.0625, EASIER achieves 83.5% classification rate, while the correct rate of EASE is 82.3% and SRS achieves only 59.7%.

Figure 5.4g shows the sampling time of original CSD data. EASIER requires a fixed amount of time, even as the sample ratio falls. For EASIER, the computation time for sample ratios 0.125, 0.0625, and 0.015625 is 0.1300s, 0.14427s, and 0.1321s, respectively, whereas the computation time of EASE for these sample ratios is 0.26069s, 0.26927s, and 0.26992s, respectively, which increases even as the sample ratio falls.

As described in section 5.4.2, we reduced the number of items in the CSD dataset to 4,096 through requantization. The results of requantized CSD data (4,096 items) are shown in Figure 5.5. Figures 5.5a, 5.5b and 5.5c shows the correct classification rate of image classification with different percentages of noisy data, respectively. Figures 5.5d, 5.5e and 5.5f show the accuracy of association rule mining. For nonnoisy data, the accuracy of EASIER is similar to EASE, with less running time. For example, for a sample ratio of 0.000785, EASIER achieves 85.7% accuracy in association rule mining, while the accuracy of EASE is 85.4%, and SRS achieves only 71.5%. For data with noise, the results of EASIER are much better.

Figure 5.5g shows the sampling time of re-quantized CSD data. The sampling time of EASIER does not change with the sample ratio. For the sample ratio 0.125, 0.0625, and 0.015625, the time taken by EASIER is 0.0717s, 0.0717s, and 0.0713s, respectively, whereas the time taken by EASE for these sample ratios is 0.1831s, 0.2004s, and 0.2166s, respectively. As the total number of items in dataset is reduced, the running time of EASIER is reduced from about 0.13s to 0.07s.

For the memory consumption comparison, in EASE, the memory required for storing the penalty function increases with the number of halving times. For example, when we applied seven halvings to CSD data, the penalty functions of each halving procedure are stored. The required memory for storing the penalty of CSD items is about $7+2 = 9$MB.

Table 5.3: The average number of noise in the selected training samples and the corresponding percentage against total number of noisy data. The dataset is the original 39D data.

5.3a: Results of EASIER. The sample ratios are 0.6, 0.3 and 0.1 separately.

Number of dataset	Noise	The number and percentage of noise for different ratios					
		0.6		0.3		0.1	
3600	360 (10%)	215.1	59.74%	63.4	17.61%	10.6	2.96%
3600	180 (5%)	106.5	59.17%	26.8	14.91%	0	0.00%
360	36 (10%)	21	58.35%	5.8	16.22%	1	2.68%
360	18 (5%)	10.6	58.61%	2.5	13.80%	0	0.00%

5.3b: Results of EASE. The sample ratios are 0.3 and 0.1 separately.

Number of dataset	Noise	The number and percentage of noise for different ratios			
		0.3		0.1	
3600	360 (10%)	105.2	29.21%	33.6	9.32%
3600	180 (5%)	51.7	28.71%	16.4	9.11%
360	36 (10%)	10.7	29.77%	3.5	9.6%
360	18 (5%)	5.4	29.82%	1.7	9.67%

5.3c: Results of SRS. The sample ratios are 0.6, 0.3 and 0.1 separately.

Number of dataset	Noise	The number and percentage of noise for different ratios					
		0.6		0.3		0.1	
3600	260 (10%)	216.4	60.11%	105.6	29.33%	36.7	10.20%
3600	180 (5%)	109.3	60.71%	53.9	29.96%	17	9.47%
360	36 (10%)	21.8	60.67%	10.5	29.05%	3.5	9.72%
360	18 (5%)	10.8	60.15%	5.3	29.41%	1.8	10.00%

But for EASIER, only one set of the penalty functions needs to be stored, so the memory required is only about $1 + 2 = 3$MB. Although a memory requirement of 9MB is not very large, when the number of items is greatly increased, the memory requirement can significantly affect the performance of the algorithm.

5.5.2 Audio application

EASIER and SRS are compared using an audio database that contains one hour of basketball audio in 3,600 samples (one sample for each second). Both algorithms run five times and the results are computed as an average over five samples. As earlier discussed, we use HMM as the audio event identifier. Audio data with different ratios of noisy data is classified into four classes namely are EA, EC, PA, and PC. EASIER, EASE, and SRS are used for sampling. The precision and recall of nonnoisy and different ratios of noisy

Table 5.4: The average number of noise in the selected training samples and the corresponding percentage against total number of noisy data. The dataset is the 13D data.

5.4a: Results of EASIER. The sample ratios are 0.6, 0.3 and 0.1 separately.

Number of dataset	Noise	The number and percentage of noise for different ratios					
		0.6		0.3		0.1	
3600	360 (10%)	212.7	59.08%	60.6	16.84%	4.6	1.27%
3600	180 (5%)	106.7	59.25%	25.9	14.38%	0	0.00%
360	36 (10%)	21.5	59.64%	6.1	16.84%	0.5	1.39%
360	18 (5%)	10.4	57.58%	2.6	14.40%	0	0.00%

5.4b: Results of EASE. The sample ratios are 0.3 and 0.1 separately.

Number of dataset	Noise	The number and percentage of noise for different ratios			
		0.3		0.1	
3600	360 (10%)	102.9	28.6%	33.3	9.24%
3600	180 (5%)	51.1	28.36%	17.2	9.55%
360	36 (10%)	10.1	28.04%	3.2	8.96%
360	18 (5%)	5.2	28.92%	1.6	8.89%

5.4c: Results of SRS. The sample ratios are 0.6, 0.3 and 0.1 separately.

Number of dataset	Noise	The number and percentage of noise for different ratios					
		0.6		0.3		0.1	
3600	360 (10%)	217.6	60.43%	108.2	30.06%	37.3	10.37%
3600	180 (5%)	108.4	60.24%	54.5	30.26%	19.3	10.71%
360	36 (10%)	21.7	60.15%	10.9	30.33%	3.6	10.00%
360	18 (5%)	10.6	59.13%	5.2	29.15%	1.9	10.55%

data are shown in Figures 5.6, 5.7 and 5.8. The sampling ratios for EASIER and SRS include 0.6, 0.3, and 0.1. For EASE only 0.3 and 0.1 are used, as EASE cannot achieve every sample ratio. The data in the sample set is used for training and the other data is used to test the results. For example, if the sample ratio is 0.1, this means 10% samples are selected from the data set for training and other 90% data are test data. For sample ratio 1, all data is used for both training and test.

Tables 5.3 and 5.4 demonstrate the effect of removing noise for audio data. The whole dataset includes 3,600 audio features and two subsets are used for test. It can be seen that EASIER can remove the most of noisy data when the sample ratio is small. EASE and SRS always keep the same percentage of noisy data in the sample set. The results are not as good as image data because in each audio feature, the number of items is more than 3,000. It is too large.

As demonstrated in Figure 5.6, compared with SRS, sampling with EASIER can achieve high performance with less training samples, especially for the smaller classes EA and EC and the data with noisy. By using EASIER, identification improves in two aspects: 1) To achieve similar recall and precision, EASIER sampling needs relatively less training data than SRS. For example, for a precision of 85% for EA, SRS needs 60% training data, whereas EASIER needs only 30%; and 2) For the same training set, EASIER gives higher performance. For sample ratio 0.1 and "Excited Audience" class, the precision of EASIER is 90%, which is significantly higher than that of SRS (54%). Although the expected precision and recall values for 13D data are a little smaller than those for 39D, they are significantly larger than those for SRS.

The performance of EASIER is better than EASE with noisy data, while EASE takes longer sampling time. Figure 5.9 shows the computation time taken by EASIER, EASE, and SRS. For 39D data, the sampling time of EASIER is about 1.8s. This is acceptable, considering the long training and classification time. EASE takes up to 2.8s when the

sample ratio is smaller. For 13D data, the sampling time of EASIER is reduced greatly to only 0.4s. It can be seen that for different sample ratios, EASIER requires almost a fixed amount of time, whereas EASE requires varying time.

Note that for the identification of an event in basketball, the smaller classes of EA and EC are more important than PA and PC. EASIER gives higher accuracy for these two important classes, signifying its preferability over SRS. In addition, EASIER's performance vis-a-vis SRS improves further as we reduce the sample ratio. This has been shown in other experiments not reported here. Due to small size of EA and EC (approximately 10% each of the whole data) classes, we could not show results of EASIER vis-a-vis SRS for sample ratios of less than 10%, since otherwise the training set is too small and cannot achieve satisfactory performance.

5.5.3 Association rule mining

In order to further compare the performance of the three algorithms, the IBM QUEST transaction data of [1] is also used to test the accuracy in the context of association rule mining. The dataset has total 98,040 transactions and the total number of items is 1000. The average length of these transactions is 10, and the average length of potentially frequent item sets is 4. The minimum support value is set to 0.77%.

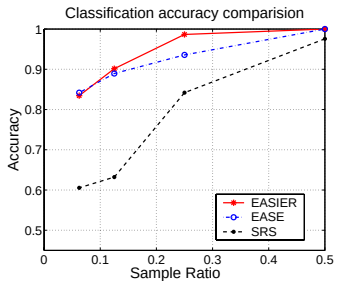
All three algorithms start from a 20% simple random sample S of the original database. One to five halvings are applied to EASE. Thus we get final sample ratios as 0.1, 0.05, 0.025, 0.0125 and 0.00625 of the whole database. The three algorithms generate samples using the described setting. All three algorithms run 50 times for each sample and the results are computed as an average over these 50 runs. For EASIER and EASE, in each iteration a different random sample is used as the initial 20% sample.

The accuracy is shown in Figure 5.10a. It can be seen that the accuracy of EASIER is better than EASE in small sample ratio. For sample ratio 0.00625, the accuracy of

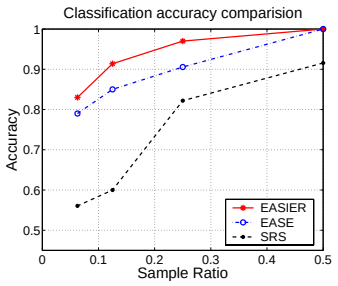
EASIER is about 86.2% while EASE has only 71.2% accuracy. The SRS gives the worst accuracy of 41.2%.The sampling time of the three methods are very similar as shown in Figure 5.10b.

5.5.4 Summary of experimental results

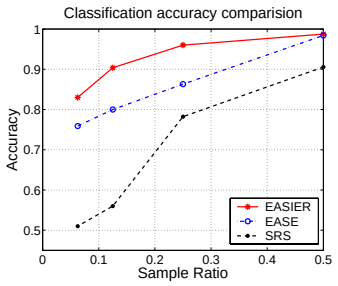
The experimental results in this section have shown that EASIER applies well to a varied range of applications. The image and audio applications have continuous features, whereas the market basket data (IBM QUEST data) has discrete features. Furthermore, for both classification and association rule mining, EASIER samples outperformed SRS significantly. A simple quantization is applied to transform the original continuous data of both image and audio applications to binary format. After that, to reduce the elements in the whole dataset, different approaches are used for image and audio data. For image data, the 8-bit bin values are requantized to 4-bit representations. For audio data, the main 13D features are directly extracted from the original 39D features. Although quantization or discretization invariably results in some information loss, it has only slight impact on performance in comparison with original data because most of the dominant information is retained. In the literature there are more sophisticated discretization methods, such as entropy-based discretization [93]. This is a future direction of the current work, that is, to see the effect of discretization on the performance of EASIER.



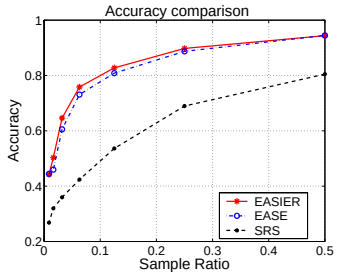
5.4a: Classification accuracy



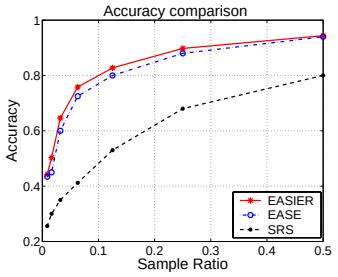
5.4b: Classification accuracy with 5% noise



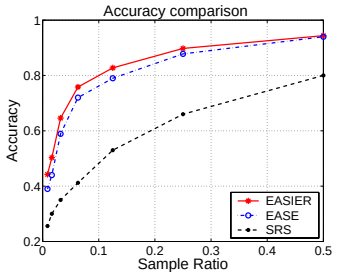
5.4c: Classification accuracy with 10% noise



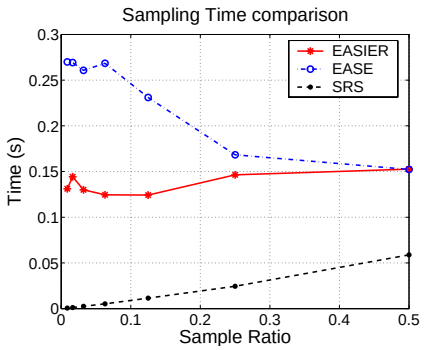
5.4d: Association rule mining



5.4e: Association rule mining with 5% noise

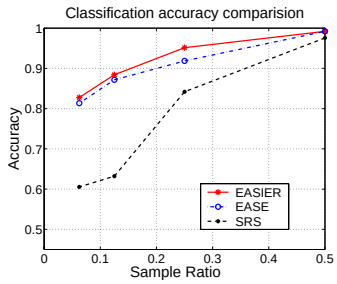


5.4f: Association rule mining with 10% noise

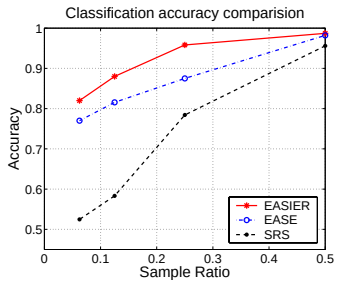


5.4g: Sampling Time

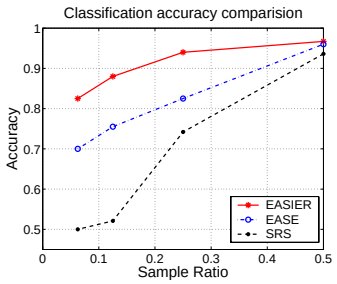
Figure 5.4: The performance of original CSD data extracted from COIL image set.



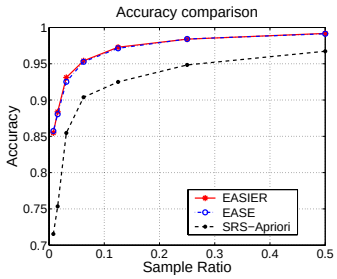
5.5a: Classification accuracy



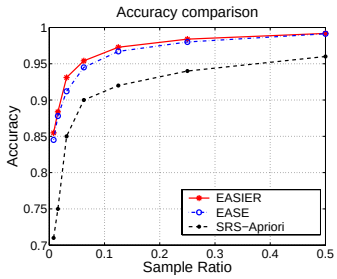
5.5b: Classification accuracy with 5% noise



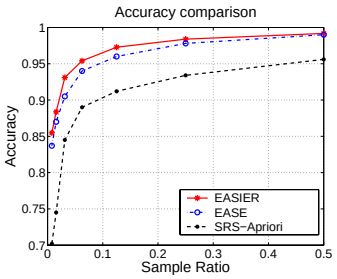
5.5c: Classification accuracy with 10% noise



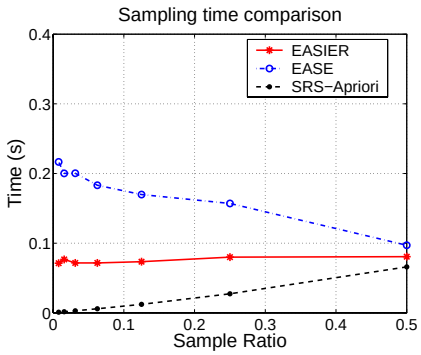
5.5d: Association rule mining



5.5e: Association rule mining with 5% noise

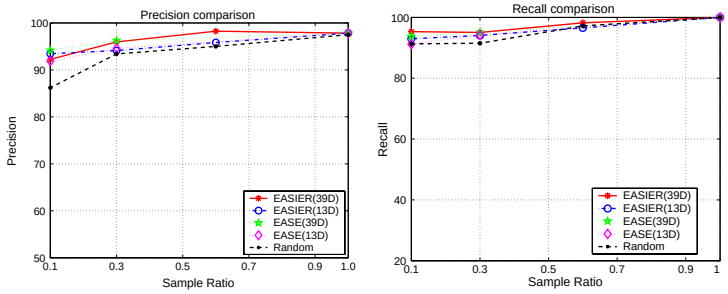


5.5f: Association rule mining with 10% noise

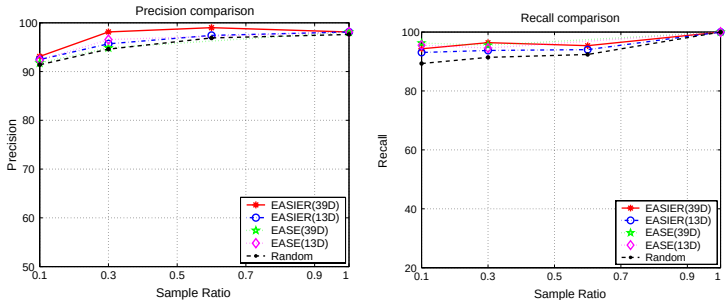


5.5g: Sampling Time

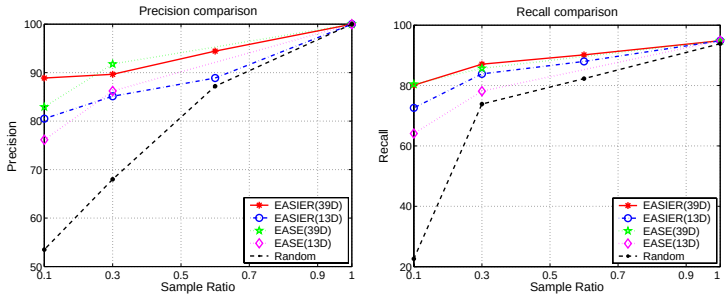
Figure 5.5: The performance of re-quantized CSD data extracted from COIL image set.



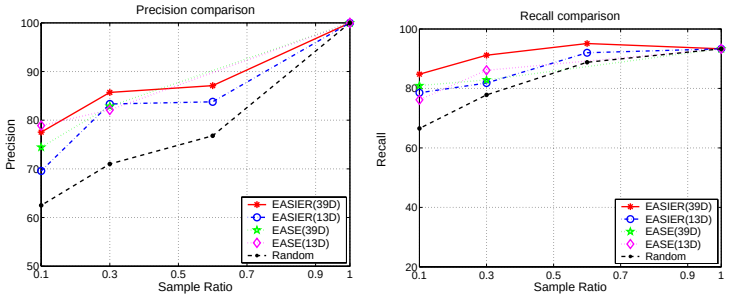
5.6a: Plain Audience



5.6b: Plain Commentator

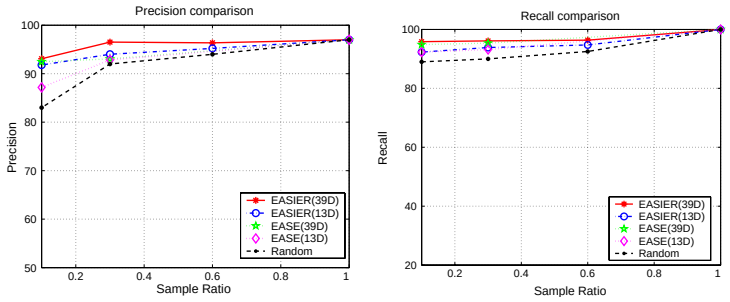


5.6c: Excited Audience

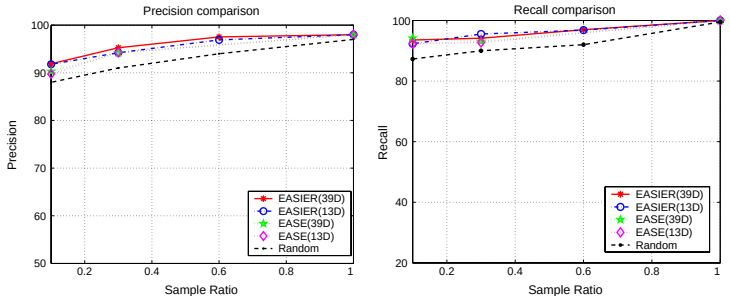


5.6d: Excited Commentator

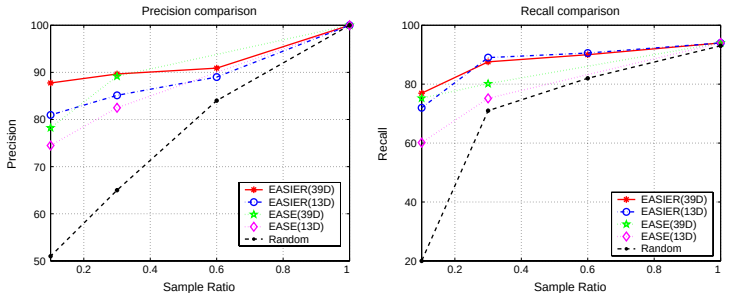
Figure 5.6: The performance of audio event identification based on sample set selected by SRS and EASIER.



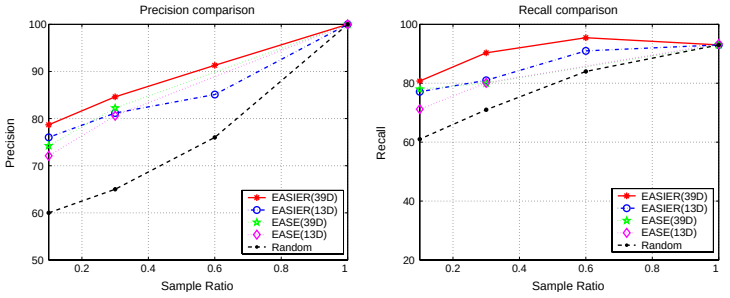
5.7a: Plain Audience



5.7b: Plain Commentator

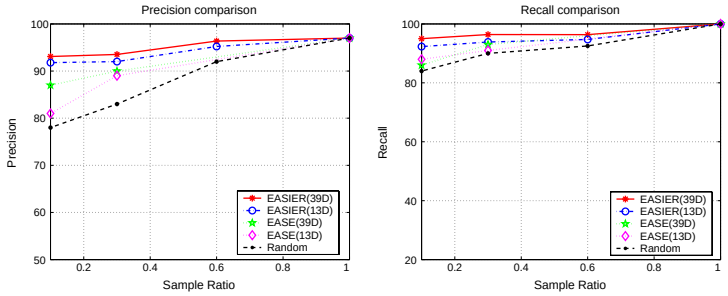


5.7c: Excited Audience

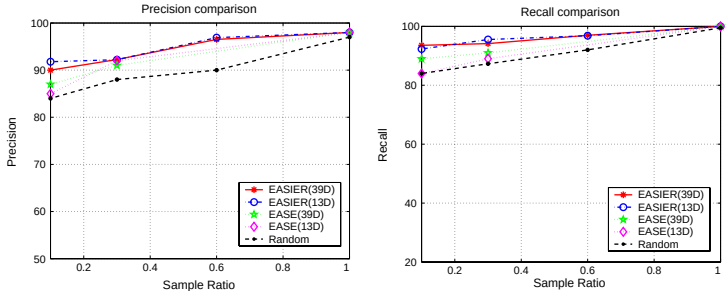


5.7d: Excited Commentator

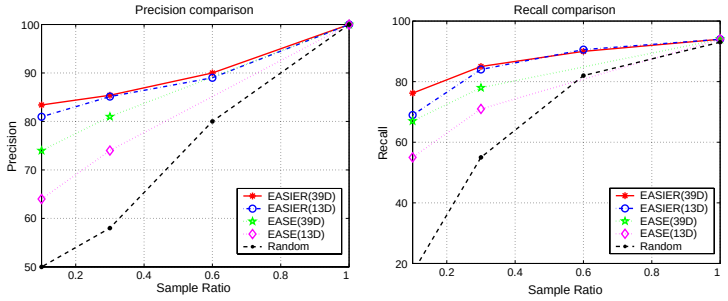
Figure 5.7: The performance of audio event identification based on sample set selected by EASIER, EASE and SRS. The noisy rate is 5%.



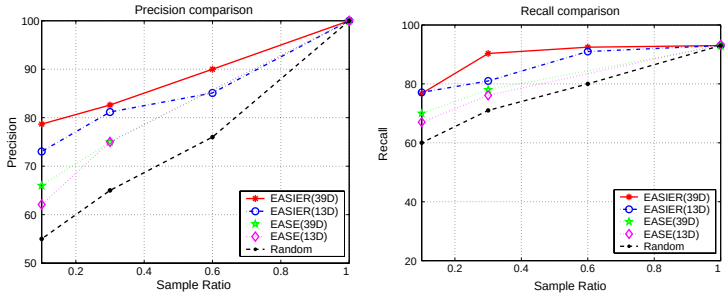
5.8a: Plain Audience



5.8b: Plain Commentator



5.8c: Excited Audience



5.8d: Excited Commentator

Figure 5.8: The performance of audio event identification based on sample set selected by EASIER, EASE and SRS. The noisy rate is 10%.

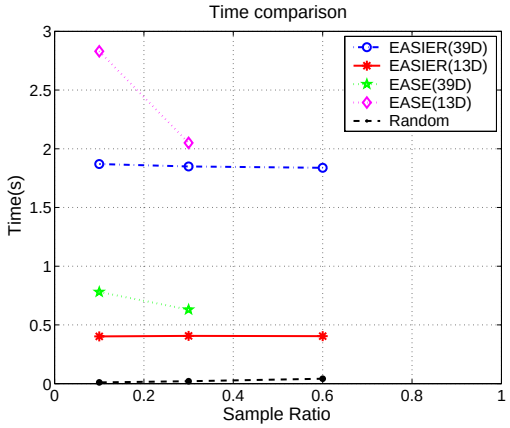
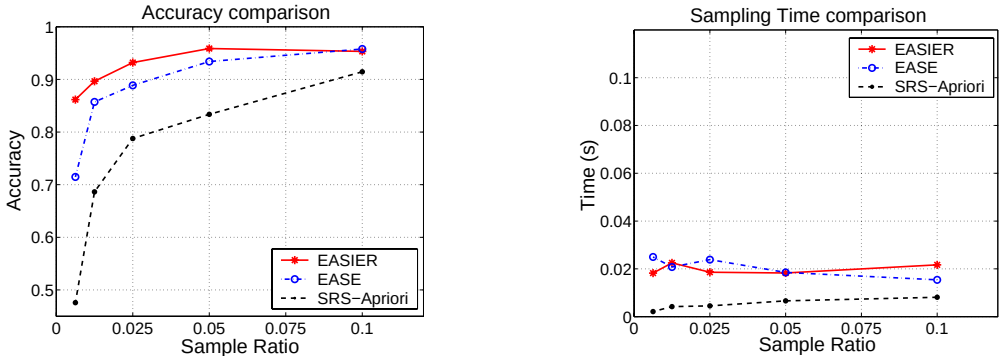


Figure 5.9: The sampling time for different sample ratios.



5.10a: Accuracy

5.10b: Sampling Time

Figure 5.10: The performance of IBM QUEST transaction data.

5.6 Summary

In this chapter, we proposed a new sampling algorithm, EASIER. We applied it to the selection of concise training set for multimedia classification. EASIER is similar to its predecessor EASE, but reduces the requirements for time and memory. In EASE, sampling time and memory increases when the sample ratio is reduced. However, in EASIER the running time is almost fixed and the memory independent of the sample ratio. Another improvement is that due to its halving nature, in EASE we must change the size of the initial sample to obtain some specific ratios. However, using EASIER, any sample ratio can be obtained directly from the same initial set. We have evaluated the performance of EASIER using both real-world and synthetic data. It can be seen that EASIER is a good approximation algorithm which can obtain better sampling results with almost fixed time and even better accuracy than EASE.

We successfully applied EASIER to image and audio applications that have continuous features and marketing basket application that has categorical features. Some Gaussian noisy data was added to verify the efficacy in reducing noise by EASIER. As EASIER can flexibly generate representative samples of a huge image database on-the-fly, it is used to select the training samples for a SVM classifier for image classification and an for audio event identification. The performance study shows that an EASIER sample represents the original data much better than a simple random sample. Compared with SRS, EASIER algorithm improves classification performance in the following two aspects: 1) EASIER effectively finds the relative representative data for training, and consequently improves the classification rate significantly; and 2) it provides a feasible way to train a classifier or identifier for a large multimedia database by only using a small training dataset. Especially for small sample ratios, EASIER can achieve significantly better results than SRS. Currently, in multimedia applications, we test with images and basketball audio.

EASIER is an online algorithm where the incoming transactions are processed once and a decision is taken regarding its participation in the final sample. This scheme is very conducive to stream data processing. The idea is to maintain a sample for the stream data dynamically. Just like reservoir sampling [94], each incoming transaction is processed in a given amount of time. But unlike reservoir sampling, where this decision is made based solely on probability, EASIER makes informed decisions. The initial idea is to maintain a ranking among the selected transactions in the reservoir sample. When a new streaming transaction arrives, its rank is determined by calculating the change in distance of the reservoir sample from the actual data. If its rank is higher than the lowest rank among the reservoir transactions, it is selected.

Chapter 6

Conclusion and Future Work

Multimedia indexing and description is important and useful when we face huge multimedia dataset. The target of this thesis is to find an appropriate representation and indexing scheme based on image content, as well as efficient management of large multimedia database. Different from the traditional database or the text-based retrieval, images need some low-level features to describe the content, such as the MPEG-7 visual descriptors. The semantic gap between low-level features and high level concept is important and needs further research, especially when the image dataset is very large.

6.1 Summary of contributions

In this thesis, an image retrieval system based on MPEG-7 descriptors is proposed and implemented. This system starts with image retrieval based on low-level visual features, and goes to semantic object classification. Besides efficient multimedia management is investigated in several aspects. The main contributions include:

1. Basic image retrieval based on some MPEG-7 color descriptors. A more efficient similarity measure, Earth Mover's Distance, is 385lied to improve the retrieval performance based on Dominant Color Descriptor. M-tree index and lower bound of EMD are discussed to prune the images far from the query image and improve the retrieval efficiency.

2. Extend image retrieval based on single descriptor to combination of multiple descriptors. We have proposed a weighted combination model and the model is applied to several MPEG-7 visual descriptors. The weight of every descriptor is determined self-adaptively based on optimization technology. The same optimization structure can be used for many other features. It is a unified approach for content-based image retrieval. Lagrange multiplier is applied to solve the optimal solutions. The optimal solution is explicit and the calculation procedure is time-efficient.

3. Extend image retrieval based on whole image to local features of points and regions. As discussed earlier, image retrieval based on low-level descriptors can only find some images with visual similarity. For the same object with different scales and orientations in images, the local features are more efficient to determine the object. This is because the local features focus on finding the identical points of an object. As an initial research, scale invariant points and corresponding features are applied to find the objects in different scales and orientations. Using more accurate similarity measures the retrieval efficiency is improved.

4. Extend the search for same object in different views to classification of object category. It is necessary to make the distinction between the recognition of specific instances of objects and classes of objects. Different from local characteristics which are only suitable for the same object in different scale and orientation, a high performance and easy calculated classification model is proposed in this thesis for classification of object category. The model is based on salient regions and multiple MPEG-7 descriptors are applied to represent the regions. When this image model is used for object classification, the similarity distance based on appearance of single region and the geometric distortion between a pair of regions are both taken into consideration. In order to efficiently make use of the nested spatial relationships between the regions, a graph-based matching algorithm is investigated to determine the corresponding regions in the image

model and images in the database. Experimental results show the image model based on representative regions is easy to calculate and efficient results can be obtained. This model can be used to classify the object categories automatically. After that, user can put semantic labels to the object categories, which may be helpful to bridge the semantic gap.

5. Efficient sampling for large and noisy multimedia dataset - from random selection to EASIER sampling. EASIER can directly obtain sample set with any ratio from the whole dataset. As EASIER can flexibly generate representative samples of huge multimedia database on the fly, it can be applied to the selection of concise training set for multimedia classification. We have successfully applied EASIER to image and audio applications that have continuous features. Some Gaussian noisy data are added to verify the effect of reducing noise by the proposed EASIER. The performance study shows that the sample set of EASIER represents the original data much better than a simple random sample. The noisy data in the dataset can be reduced greatly in the sample set. Compared with SRS, EASIER effectively finds the relative representative data for training and consequently improves the classification rate significantly. It provides a feasible way to train classifier or identifier for large multimedia database by only using a small training dataset. Especially for small sample ratios, EASIER can achieve significantly better results than SRS.

From the theoretical analysis to experimental results, it can be seen that MPEG-7 visual descriptors are efficient description tools to describe images as low level features. With representative regions and suitable features, object categories can be distinguished based on the classification model. Efficient sampling can help to find the most representative set of the whole database. Applying good training set selected by sampling algorithm can achieve better classification results.

6.2 Future work

In this thesis we address on image retrieval and classification based on MPEG-7 visual descriptors. We have developed serval methods towards semantic objective, however, there are a number of issues need to be further investigated. We suggest some possible research directions as follows:

1. More efficient combination of multiple descriptors and other features.

Currently the optimal combination is self-adapted according to the image descriptors, i.e., user does not to need to manually adjust or label the results to determine the weights. Although it is an advantage to reduce user's effort, relevance feedback is a powerful tool to iteratively improve a query without increasing the computational requirements.

Another possible approach is to combine the low-level image features with other content-aware information. For example, huge number of images are available in the Internet and there usually exits text related to the images. The text is usually greatly related to the semantic meaning of the images and very useful to refine the results. It is helpful to automatically extract semantic information in high level.

2. The improvement of classification model for object category.

Currently the classification model is based on several individual features and it can be improved with the combination of multiple features. As the spatial locations of a pair of regions cannot sufficiently represent the relationship between regions and the fully-connected graph is a little time-consuming, more research is needed to use the relationship between regions to represent the characteristic of the object category, such as the spanning tree for a graph. Since the effect of region selection greatly influences the retrieval results, more efficient methods for region selection and combination can be investigated in the future. Some negative training sets could also be added to further improve the retrieval results.

As the model is easy to compute and apply, the multi-view representation of objects can be built in one model in a time-efficiency way. Currently the model performs well for the objects with stable structure, such as the motorbike. But plenty of objects are flexible in structure, for example, the butterfly or eagle. Their wings can be open or close. It is can also be included in a multi-view model. To make the current classification model more useful and applicable, an attempt should be made to address these issues.

3. Apply EASIER sampling to various applications.

It is proved that EASIER sampling is efficient in selecting samples of multimedia data. As EASIER is a common sampling method, it is suitable for most of the data reduction issues and can be easily extended to various applications.

References

[1] H. Bronnimann, B. Chen, M. Dash, P. Haas, and P. Scheuermann. Efficient data reduction with ease. In *Proceedings of 9th International Conference on Knowledge Discovery and Data Mining*, pages 59–68, 2003.

[2] *ISO/IEC 15938-3/FDIS Information Technology - Multimedia content description interface - Part 3: Visual.*

[3] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[4] *ISO/IEC15938-8/FDIS Information Technology - Multimedia Content Description Interface - Part 8: Extraction and use of MPEG-7 descriptions.*

[5] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C.-S. Xu, and Q. Tian. Hmm-based audio keyword generation. In *Proceedings of Pacific Conference on Multimedia*, volume 3, pages 566–574, 2004.

[6] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision*, pages 525–531, Vancouver, Canada, 2001.

[7] R. Zhao and W. I. Grosky. *Bridging the Semantic Gap in Image Retrieval, Distributed Multimedia Databases: Techniques and Application.* T. K. Shih (Ed.), Idea Group Publishing, Hershey, Pennsylvania.

[8] G Carneiro, A B Chan, P Moreno, and N Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pages 394–410, 2006.

[9] A Yavlinsky, E Schofield, and S Ruger. Automated image annotation using global features and robust nonparametric density estimation. In *International Conference on Image and Video Retrieval*, July 2005.

[10] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.

[11] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, To appear, 2008.

[12] K. Hirata and T. Kato. Query by visual example - content based image retrieval. In *Advances in Database Technology (EDBT '92, 3rd International Conference on Extending Database Technology)*, volume 580, pages 56–71, 1992.

[13] QBIC™ – IBM's query by image content. `http://wwwqbic.almaden.ibm.com/~qbic/`, 1998.

[14] ImageScape. `http://skynet.liacs.nl/`, 2002.

[15] imgSeek. `http://www.imgseek.net/`.

[16] MPEG-7 homogeneous texture descriptor demo. `http://nayana.ece.ucsb.edu/M7TextureDemo/Demo/client/M7TextureDemo.html`, 2002.

[17] Picsom. `http://www.cis.hut.fi/picsom/`.

[18] B. S. Manjunath, Philippe Salembier, and Thomas Sikora. *Introduction to MPEG-7*. John Wiley & Sons, Ltd, 2002.

[19] B. S. Manjunath, Jens-Rainer Ohm, Vinod V. Vasudevan, and Akio Yamada. Color and texture descriptors. *IEEE transactions on circuits and systems for video technology*, 11(6), 2001.

[20] Qasim Iqbal and J. K. Aggarwal. Combining structure, color and texture for image retrieval: A performance evaluation. In *Proceedings of the International Conference on Pattern Recognition*, pages 438–443, 2002.

[21] R. Fagin. Fuzzy queries in multimedia database systems. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, pages 1–10, June 1998.

[22] Ulrich Guntzer, Wolf-Tilo Balke, and Werner Kiesling. Optimizing multi-feature queries for image databases. In *Proceedings of the 26th international conference on very large databases*, pages 419–428, 2000.

[23] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. Neuromerge: An approach for merging heterogeneous features in content-based image retrieval systems. In *Proceedings of International Workshopon Multi-Media Database Management Systems*, 1998.

[24] Y. Rui and T. S. Huang. Optimizing learning in image retrieval. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, June 2000.

[25] P. Piamsa-nga and N. A. Alexandridis. Multi-feature content-based image retrieval. In *Proceedings of International Conference on Computer Graphics and Imaging*, June 1998.

[26] Lei Wang and Kap Luk Chan. Bayesian learning for image retrieval using multiple features. In *Proceedings of the Second International Conference on Intelligent Data Engineering and Automated Learning, Data Mining, Financial Engineering, and Intelligent Agents*, pages 473–478, London, UK, 2000. Springer-Verlag.

[27] Yumin Tian and Lixia Mei. Image retrieval based on multiple features using wavelet. In *Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications*, page 137, Washington, DC, USA, 2003. IEEE Computer Society.

[28] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of Conference 4th Alvey Vision*, pages 189–192, 1988.

[29] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 2(21):224–270, 1994.

[30] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[31] E. Loupias, N. Sebe, S. Bres, and J.-M. Jolion. Wavelet-based salient points for image retrieval. In *Proceedings of International Conference on Image Processing*, pages 518–521, 2000.

[32] Chiou-Ting Hsu and Ming-Chou Shih. Content-based image retrieval by interest points matching and geometric hashing. In *Proceedings of SPIE Photonics Asia Conference*, pages 80–90, 2002.

[33] V. Gouet and N. Boujemaa. On the robustness of color points of interest for image retrieval. In *Proceedings of IEEE International Conference on Image Processing*, pages II/377–II/380, 2002.

[34] Konik H. Da Rugna, J. Color interest points detector for visual information retrieval. In *Proceedings of SPIE - The International Society for Optical Engineering*, pages 139–146, 2002.

[35] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, September 1999.

[36] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[37] I. Laptev and T. Lindeberg. A distance measure and a feature likelihood map concept for scale-invariant model matching. *International Journal of Computer Vision*, 52(2-3):97–120, 2003.

[38] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, June 2003.

[39] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages 1134–1141, October 2003.

[40] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II90–II96, 2004.

[41] Dongqing Zhang. *Statistical Part-Based Models: Theory and Applications in Image Similarity, Object Detection and Region Labeling*. PhD Thesis Graduate School of Arts and Sciences, Columbia University, 2005.

[42] Duck Hoon Kim, Il Dong Yun, and Sang Uk Lee. A new attributed relational graph matching algorithm using the nested structure of earth mover's distance. In *17th International Conference on Pattern Recognition (ICPR'04)*, pages 48–51, 2004.

[43] Duck Hoon Kim, Il Dong Yun, and Sang Uk Lee. A comparative study on attributed relational gra matching algorithms for perceptual 3-d shape descriptor in MPEG-7. In *ACM Multimedia*, pages 700–707, 2004.

[44] B. Gu, F. F. Hu, and H. Liu. Sampling and its applications in data mining: A survey. Technical report, School of Computing, National University of Singapore, Singapore, 2000.

[45] F. Olken. *Random Sampling from Databases*. PhD thesis, Department of Computer Science, University of California, Berkely, 1993.

[46] M. Plutowski and H. White. Selecting concise training sets from clean data. *IEEE Transactions on Neural Networks*, 4(2):305–318, 1993.

[47] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[48] Les Atlas, David Cohn, Richard Ladner, M. A. El-Sharkawi, and II R. J. Marks. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems*, volume 2, pages 566–573, 1990.

[49] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. In *Advances in neural information processing systems*, volume 7, pages 705–712. The MIT Press, 1995.

[50] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of 8th ACM International Conference on Knowledge Discovery and Data Mining*, 2002.

[51] M. Saar-Tsechansky and F. Provost. Active learning for class probability estimation and ranking. In *Proceedings of 17th International Joint Conference on Artificial Intelligence*, pages 911–920, 2001.

[52] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[53] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. In *Proceedings of 17th International Conference on Machine Learning*, pages 999–1006, 2000.

[54] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of 11th International Conference on Machine Learning*, pages 148–156, 1994.

[55] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden Markov models for information extraction. In *Proceedings of the International Symposium on Intelligent Data Analysis*, 2001.

[56] Vijay S. Iyengar, Chidanand Apte, and Tong Zhang. Active learning using adaptive resampling. In *Proceeding of Intenational Conference on Knowledge Discovery and Data Mining*, pages 92–98, 2000.

[57] Christopher Meek, Bo Thiesson, and David Heckerman. The learning-curve sampling method applied to model-based clustering. *Journal of Machine Learning Research*, 2(3):397–418, 2002.

[58] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[59] Nitesh Chawla, Steven Eschrich, and Lawrence O. Hall. Creating ensembles of classifiers. In *Proceedings of International Conference on Data Mining*, pages 580–581, 2001.

[60] B. Chen, P. Haas, and P. Scheuermann. A new two-phase sampling based algorithm for discovering association rules. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2002.

[61] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

[62] X. Zhu, X. Wu, and S. Chen. Eliminating class noise in large datasets. In *Proceedings of the 20th ICML International Conference on Machine Learning*, pages 920–927, 2003.

[63] F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal on Mathematical Physics*, 20:224–230, 1941.

[64] P. Ciaccia, M. Patella, and P. Zezula. M-tree: an efficient access method for similarity search in metric spaces. In *Proceedings of International Conference on Very Large Databases*, pages 420–435, August, 1997.

[65] P. Zezula, P. Ciaccia, and F. Rabitti. M-tree: a dynamic index for similarity queries in multimedia databases. In *TR 7, HERMES ES-PRIT LTR Project*, 1996.

[66] M. Patella. *M-tree (Version 0.911) User's Guide*. 2000.

[67] T. Ojala, M. Aittola, and E. Matinmikko. Empirical evaluation of MPEG-7 XM color descriptors in content-based retrieval of semantic image categories. In *Proceedings of 16th International Conference on Pattern Recognition*, pages 1021–1024, Quebec, Canada, 2002.

[68] Detlef Zier and Jens-Rainer Ohm. Common datasets and queries in MPEG-7 color core experiments. October 1999.

[69] *ISO/IEC JTC1/SC29/WG11 N2929, Description of Core Experiments for MPEG-7 Color/Texture Descriptors.*

[70] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(27):1615–1630, 2005.

[71] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[72] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2000.

[73] Yali Amit and Donald Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.

[74] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[75] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 2006.

[76] Timor Kadir and Michael Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, November 2001.

[77] Edward Chang and Beitao Li. MEGA—the maximizing expected generalization algorithm for learning complex query concepts. *ACM Transactions on Information Systems*, 21(4):347–382, 2003.

[78] M.A. Eshera and K.S. Fu. A graph distance measure for image analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 14(3):398–408, 1984.

[79] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Semi-local affine parts for object recognition. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 959–968, September 2004.

[80] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: a quantitative study of their impacts. *Artificial Intelligence Review*, 22(3):177–210, 2004.

[81] Alexander Astashyn. *Deterministic data reduction methods for transactional data sets*. Master thesis, 2004.

[82] O. Chapelle, P. Halffiner, and V. N. Vapnik. Support vector machine for histogram based image classification. *IEEE Transactions on Neutral Network*, 10(5):1055–1064, 1999.

[83] M. Xu, L.-Y. Duan, L.-T. Chia, and C.-S.Xu. Audio keywords generation for sports video analysis. In *Proceedings of ACM Multimedia*, 2004.

[84] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of International Conference on Very Large Databases*, 1994.

[85] R. Jin, R. Yan, and A. Hauptmann. Image classification using a bigram model. In *AAAI Spring Symposium on Intelligent Multimedia Knowledge Management*, 2003.

[86] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for tv baseball programs. In *Proceedings of ACM Multimedia*, pages 105–115, 2000.

[87] Chunru Wan and Mingchun Liu. Content-based audio retrieval with relevance feedback. *Pattern Recognition Letter*, 27(2):85–92, 2006.

[88] Regunathan Radhakrishnan, Ajay Divakaran, Ziyou Xiong, and Isao Otsuka. A content-adaptive analysis and representation framework for audio event discovery from "unscripted" multimedia. *EURASIP Journal on Applied Signal Processing*, 2006(1):191–191, 2006.

[89] S. Nepal, U. Srinivasan, and G. Reynolds. Automatic detection of goal segments in basketball videos. In *Proceedings of ACM Multimedia*, Los Angeles, CA, 2001.

[90] S. Young and et al. *The HTK Book (for HTK Version 3.1)*. Cambridge University Engineering Department, 2002.

[91] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of International Conference on Management of Data*, 1993.

[92] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of International Conference on Management of Data*, 2000.

[93] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of International Conference on Machine Learning*, pages 194–202, 1995.

[94] J.S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, March 1985.

List of Publications

Journals

1. Surong Wang, Manorajan Dash, Liang-Tien Chia, and Min Xu, Efficient data reduction in multimedia data, *Applied Intelligence*, 25(3):359-374, 2006.

2. Surong Wang, Manorajan Dash, Liang-Tien Chia, and Min Xu, Efficient sampling of training set in large and noisy multimedia data, *ACM Transactions on Multimedia Computing, Communications and Applications*, 3(3):14, 2007.

Conferences

1. Surong Wang and Liang-Tien Chia, Image model based on salient regions and its applications, In *Proceedings of International Conference on Multimedia Modelling*, 2006.

2. Surong Wang, Min Xu, Manorajan Dash, and Liang-Tien Chia, EASIER sampling for audio event identification, In *Proceedings of International Conference on Multimedia and Expo*, 2005.

3. Surong Wang, Manorajan Dash, and Liang-Tien Chia, Efficient sampling: application to image data, In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2005.

4. Surong Wang, Deepu Rajan, and Liang-Tien Chia, Optimization-based multiple MPEG-7 descriptors for image retrieval, In *Proceedings of International Workshop on Advanced Image Technology*, 2005.

5. Surong Wang, Liang-Tien Chia and Deepu Rajan, Region-based image retrieval with scale and orientation invariant features, *In Proceedings of Pacific-Rim Conference on Multimedia*, 2004.

6. Surong Wang, Liang-Tien Chia and Deepu Rajan, Efficient image retrieval using MPEG-7 descriptors, In *Proceedings of International Conference on Imaging Processing*, 2003.

7. Surong Wang, Liang-Tien Chia and Deepu Rajan, Image retrieval using Dominant Color Descriptor, In *Proceedings of International Conference on Imaging Science, Systems, and Technology*, 2003.