FROM REGION FEATURES TO SEMANTIC LABELS: A PROBABILISTIC APPROACH

Li Rui

NATIONAL UNIVERSITY OF SINGAPORE
2002

Name: Li Rui
Degree: Master of Science
Dept: Computer Science
Thesis Title: From Region Features to Semantic Labels: A Probabilistic Approach

Abstract

Content-based image retrieval has advanced from the initial stage of feature-based approach towards the semantic approach. Existing semantics-based methods typically classify an image or an image region into exactly one of several classes. Due to the presence of noise and ambiguity in images, it is practically impossible to derive classifiers that can accurately classify all the images or regions into a large variety of classes. Therefore, some image retrieval methods have captured the uncertainty of region classification in the region labels and used image structures for disambiguation during image matching. This thesis presents a novel method of semantic labeling that can assign multiple semantic labels to a region along with the confidence measures of the assignment. Unlike existing classification methods, it can learn to perform semantic labeling incrementally. Test results show that the method is effective and accurate in labeling a wide variety of regions.

Keywords: Content-based image retrieval
Semantic labeling
Confidence measure
Region classification

FROM REGION FEATURES TO SEMANTIC LABELS: A PROBABILISTIC APPROACH

Li Rui
(B. Sc. (Hon.) in Computer and Information Sciences, NUS)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE
2002

Acknowledgments

I would like to thank many people who made my time at NUS a period I will treasure.

First and foremost, I would like to thank my advisor, Professor Leow Wee Kheng, for guiding through the research. From him I learned how to think critically, how to select problems, how to solve them, and how to present their solutions. His drive for scientific excellence has pushed me to aspire for the same. I thank him also for the seemingly impossible task of teaching me how to write clearly.

I would like to express my gratitude to Dr. Ang Chuan Heng for guiding me through the honors year. The experience I gained in the year has helped me in many ways in my current research. I would also like to thank Professor Mohan, Dr. Phillipe and Dr. Hadi for inspiring discussions on feature analysis, statistical theory and general topics in image retrieval and classification.

I would like to thank my honors year friends, Indri, Kah Yee, Wai Lup, Kai Song, Wee Hyong, Patrick, Hak Yong, Pei Yuen and Fun Siong, they never fail to cheer me up when I feel down, they are great friends. I am fortunate to be in the same research

lab with Wai Keong, Bryan, Wang Jun, Prospero, Zhang Qiu Ying and Ji Yi. Many thanks to Wai Keong for interesting and fruitful discussions on mutual information and its applications.

Last but no least, I would like to thank my family. I am grateful to my parents who instilled in me the basic interest in science. Without their encouragement and support, I would never be able to go this far. Above all I am grateful to my husband, Tai Peng, for his love, support, patience and encouragement. To him I dedicate this thesis.

Publications

Wee Kheng Leow and Rui Li. Adaptive binning and dissimilarity measure for image retrieval and classification. In *Proceedings IEEE CVPR*, 2001.

Wee Kheng Leow and Rui Li. The Analysis and Applications of Adaptive-Binning Color Histograms". Submitted to *Computer Vision and Image Understanding, special issue on Colour for Image Indexing and Retrieval, 2003.*

Rui Li and Wee Kheng Leow. From Region Features to Semantic Labels: A Probabilistic Approach. Accepted for publication in *International Conference on Multimedia Modeling*, 2002.

Contents

List of Figures

List of Tables

Summary

Image region labeling is an important stage in high level image categorization and retrieval.

We present a novel framework for assigning probabilistic labels to image regions by making use of multiple types of features. In particular, we address the following questions:

- What features describe the content of an image/region well?
- How to measure the similarity between features?
- How to find the best feature subset for labeling?
- How to assign probabilistic semantic labels to a region?

In this thesis, we use color, texture and edge features as the main features for labeling. We did extensive tests on adaptive color histograms and weighted correlation, the dissimilarity measure for adaptive color histograms. We found that adaptive color histograms together with weighted correlation gave the best overall performance in image retrieval, classification and histogram clustering.

To select the best feature subset for the labeling, a feature-based clustering is performed. Different feature types are clustered separately using appropriate dissimilarity measure. Based on the clustering result, different feature types are combined through a probabilistic approach and information about the best feature type combination is derived too. The feature subset, instead of every feature type, are used to label a region. The labeling algorithm is independent of the types of features used and it can adapt to more general or special labeling tasks easily. The labeling algorithm gives multiple fuzzy labels together with corresponding confidence values to a region.

Test results have shown that this framework can find the salient features of respective semantic classes and use the salient features to achieve good labeling performance. Even for those regions with lower confidence which cannot be labeled very accurately, the multiple semantic labels and the corresponding confidence values allow other higher-level algorithm, such as fuzzy conceptual graph matching and attributed relational graph matching, to disambiguate them using information about image structures. In summary, this semantic labeling method is expected to contribute significantly in bridging up the gap between low-level features and high-level semantics for effective image categorization and retrieval.

Chapter 1

Introduction

1.1 Background

Recent technological advancements in various fields of endeavor have combined to make large databases of digital images accessible. These advancements include:

- Availability of image acquisition devices, such as scanners and digital cameras.
- Affordability of large storage units at much lower costs.
- Access to a large numbers of images via the internet, and the rapid growth of the World-Wide Web where images can be easily added and accessed.

Searching through the large databases and finding a particular picture is painstakingly time-consuming. This problem brought active research in efficient image retrieval techniques based on image content. Early content-based image retrieval (CBIR) systems follow the paradigm of representing images by low-level features, such as color, texture

and shape. Image retrieval is performed by matching the features of the query image with those of the database images. This approach of matching global image features is effective for retrieving simple images and images that contain single distinct objects. CBIR systems such as QBIC [22], Virage [21], ImageRover [47], Photobook [40] and VisualSEEK [50] all belong to this category.

For more complicated images with multiple objects and regions local features are extracted from segmented regions or fixed-sized blocks. Images are then retrieved by matching their region or block features with those in the query image. Such region-based CBIR systems include Netra [30], Blobworld [11], and SIMPLIcity [56].

All the above methods use low-level feature-based methods to match images. They are poor at capturing high-level concepts that are more natural to human users. They will fail for queries such as "children playing in a park". To satisfy such a query, the systems must be able to automatically recognize children and parks in the images. That is, it must understand the meaning or semantics of the image contents. In recent years, CBIR research has shifted its focus towards bridging the gap between low-level features and high-level semantics [48]. The general idea is to assign semantic labels to parts of an image or the whole image. This thesis will focus on the problem of assigning semantic labels to parts of an image.

1.2 Preliminary Results

During our initial research of image categorization, we realize that it is extremely difficult to assign a semantic label to an image, especially when the image contains multiple objects. At the same time, we find that we can assign semantic labels to the regions of an image. The visual features of regions usually are more coherent compared to image features and they have a relatively high correlation with region semantics. If we can solve the semantic labeling problem at the region level, deriving semantic labels at image level will become a possible task.

An interesting fact is that the assignment of region labels are fuzzy rather than crisp. Though semantic understanding at the region level is not as complicated as that at the image level, there is still uncertainty in the assignment of region labels. For example, a green region can be either grass or foliage. We would like to preserve this ambiguity by giving multiple labels together with confidence measures to each region. This information can used by high-level algorithms such as fuzzy conceptual graph methods [34, 37, 41], which can disambiguate the fuzzy labels using image structure information.

Almost all existing labeling approaches assign single crisp labels to the regions. Most of them have been shown to work for only 10 or fewer semantic labels (e.g., [10, 16, 54]). Moreover, the assignment is based on classification of the features in a Euclidean space, which turns out to be not a reliable space for analyzing semantics.

For our research, we focus on exploring the correlation between features and semantic classes, selecting a best feature subset for a given semantic class and solving the fuzzy

region labeling problem using a novel method.

1.3 Research Objectives

This thesis work aims at developing a general framework for performing semantic labeling of image regions based on multiple region features. It consists of three major objectives:

1. Analyze the salient features for a specific semantic class. Salient features are features that are highly correlated with a semantic class. In practice, a semantic class is often associated with a specific combination of different types of features. Instead of using all the features extracted for image/region labeling, we can just make use of the salient features.

2. Realize fuzzy labeling by using a probabilistic approach.

3. Apply the method to label complex images.

1.4 Road Map

Figure 1.1 shows the major components of the fuzzy region labeling system. It consists of four components: feature extraction, feature-based region clustering, probability estimation and probabilistic labeling. In the feature extraction module, different types of features are extracted from the regions. What features to extract and how to compare these features are important for region and image classification. This will be presented

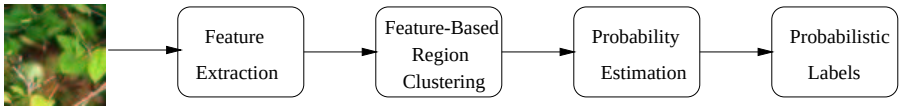


Figure 1.1: *System Overview.*

in Chapter 4. After feature extraction, feature-based region clustering is performed. Different feature types are clustered separately using appropriate dissimilarity measure. In the probability estimation module, different feature types are combined through a probabilistic approach. At this stage, information about best feature subset is derived. The best feature subset, instead of every feature type, are used to label a region. Since the labeling algorithm is independent of the types of features used, it is presented as a generic algorithm in Chapter 3. The experimental results for feature subset selection and labeling performance are described in Chapter 5. Based on the experimental results and findings, conclusions and directions are discussed in Chapter 6.

Before we move on to the detailed algorithms and implementation of the fuzzy labeling system, let us review the existing methods that are related to the proposed labeling method in Chapter 2 first.

Chapter 2

Related Work

There are two levels of semantic labeling. At the image level, it is called global semantic labeling. At the image region level, it is called local semantic labeling. In the following sections we will discuss the related research work according to these two levels separately.

2.1 Global Semantic Labeling

Three well-known methods have been proposed for global semantic labeling. Szummer and Picard [52] divided an image into blocks. Color and texture features were extracted from each block. A k-nearest neighbor classifier was used to classify blocks into indoor or outdoor class based on single image feature. Classification of images into indoor or outdoor class is done by a majority vote of the individual single-feature block classifiers. The classification rate was reported around 90% when evaluated on an image

database of 1,300 images by Kodak. Szummer and Picard claimed that the relationship between two features in block classification is almost certainly non-linear. Hence any linear combination of different features is not expected to produce satisfactory classification accuracy. This is consistent with our research findings.

Vailaya et al. [55] performed a quantitative study of various features for the city vs. landscape classification problems. Those features include: color histogram, color coherence vector, DCT coefficients, edge direction histogram and edge direction coherence vector. Edge direction-based features was chosen as they have the highest discriminative power for the city vs. landscape classification problem. A weighted k-nearest neighbor classifier was used for the classification which resulted in an accuracy of 93.9% when evaluated on an image database of 2,716 images using the leave-one-out method. This approach has been extended to further classify 528 landscape images into forests, mountains, and sunset/sunrise images. First, the input images are classified as sunset/sunrise images vs. forest & mountain images (94.5% accuracy) and then the forest & mountain images are classified as forest images or mountain images (91.7% accuracy). Their final goal is to combine multiple 2-class classifiers into a single hierarchical classifier.

Belongie et al. [2] made use of region blobs for classification. A blob represents a localized region of coherent color and texture. An Expectation-Maximization algorithm was used to find the coherent color and texture for a region and to combine them using mixture model. A naive Bayes classifier was used for image classification based on the presence or absence of region blobs. It classified images into categories such as air shows,

brown and black bears, polar bears, elephants, tigers, cheetahs, bald eagles, mountains, fields, night scenes, deserts and sunsets.

In these three research works, the classification is done at image level based on features of the entire image or of the regions or blocks. They only work for a small set of semantic classes. As many images in practical applications are complex images containing multiple objects, global semantic classification will become extremely difficult. An alternative approach is to perform global semantic classification based on local semantics of local features.

2.2 Local Semantic Labeling

Existing local semantic labeling or region labeling methods include [10, 16, 54]. Campbell et al. [10] trained a neural network to classify regions based on features such as color, texture, shape, size, rotation, and centroid. The regions were classified into 11 categories, such as sky, vegetation, road marking, road, pavement, building, fence/wall, road sign, signs/poles, shadow and mobile objects. The neural network achieved 83.9% accuracy over 3,000 test regions segmented from images in the Bristol Database.

Town and Sinclair [54] also applied a neural network to classify regions into semantic classes based on region features that include area, boundary length, color mean and covariance matrix, texture orientation and density, moment, etc. They obtained a classification accuracy of about 90% but the test was performed on only 11 classes that contain well-defined textures such as brick, cloud, fur, grass, road, and sand.

Fung and Loe [16] used supervised clustering of color features of image blocks to group them into a large number of elementary clusters, which were further grouped into conglomerate clusters. Each conglomerate cluster was associated with a semantic region class. Fixed-sized image blocks were assigned to the clusters using k-nearest-neighbor algorithm, and were then assigned the semantic labels of the majority clusters. The number of region classes and the accuracy of region classification were not reported in the paper.

The common shortcomings of the existing region labeling methods include the following:

- Only one crisp label is assigned to a region. Due to the presence of the noise and ambiguity, it is very difficult to derive very accurate classifiers for a large variety of region classes.

- They classify region features in a Euclidean or linear vector space that combines various feature types linearly. This vector-space approach is convenient but requires the assumption that the features types are independent of each other and, thus, can be regarded as forming orthogonal dimensions of the vector space. This assumption is generally false and has been shown to lead to poorer classification and clustering results in general [26]. Further discussion of this issue is presented in Chapter 3.

- Existing methods have been demonstrated to work on only a small number of about 10 region classes. Moreover, images in the region classes usually have well-defined texture patterns.

- The classification methods cannot learn from new examples incrementally. Addition of new training samples, feature types, and semantic classes entails the re-training of methods on the entire collection of old and new training samples.

In contrast, our method does not combine different feature types linearly in a Euclidean space. Instead, it adopts a probabilistic approach that captures the correlation between feature combinations and semantic classes. It adopts an incremental learning algorithm that can make uses of the dissimilarity measure that is appropriate for each feature type. Finally, it has been tested on 30 semantic classes that span a wide variety of region types that are not restricted to images that contain well-defined textures.

Chapter 3

Probabilistic Semantic Labeling

3.1 Overview

Semantic labeling can be framed as a problem of classifying an image region (or image patch, fixed-sized block) R to one of several semantic classes $C_i, i = 1, \ldots, M$. In practice, it is not possible to achieve perfect classification due to the presence of noise and ambiguity in the region features. A better approach is to represent the uncertainty of classification in the semantic labels and to defer the final decision to a latter stage at which the image structure can be taken into account using, for instance, the fuzzy conceptual graph matching method [37]. Let us denote by $Q_i(R)$ the confidence of classifying region R into semantic class C_i. Then, the semantic labeling problem can be defined as follows:

Semantic Labeling

> Given a region R and M semantic classes $C_i, i = 1, \ldots, M$, compute the confidence $Q_i(R)$ that R belongs to C_i for each i.

A region R contains a set of features F_t of type $t = 1, \ldots, m$, each having a value v_t. Each feature F_t can contain more than one component, as for color histograms, Gabor texture features, wavelet features, etc. The symbols F_t and v_t denote the *whole* feature and feature value instead of the individual component.

The standard method of computing $Q_i(R) = Q_i(v_1, \ldots, v_m)$ is the vector space approach: Regard each set of feature values as a vector $\mathbf{v} = [v_1, \ldots, v_m]^T$ in a linear vector space, and estimate Q_i using vector space classification or function approximation. It is well-known that the various feature types are not independent of each other and the scales of the feature types are different. So, forming a linear vector space by assigning each feature type (or, more commonly, each feature component of each feature type) to an orthogonal dimension of the vector space is not expected to produce reliable results in general [52].

The usual method of dealing with this problem is to represent a region as a linear combination of feature values, and define $Q_i(R)$ as a function of the linear combination:

$$Q_i(R) = Q_i\left(\sum_t w_t v_t\right) = Q_i(\mathbf{w}^T\mathbf{v}) \tag{3.1}$$

for some weight vector $\mathbf{w} = [w_1, \ldots, w_m]^T$. Another method is to generalize the weighted sum to the quadratic form [46]:

$$Q_i(R) = Q_i\left((\mathbf{v} - \mu)^T\mathbf{X}^{-1}(\mathbf{v} - \mu)\right) \tag{3.2}$$

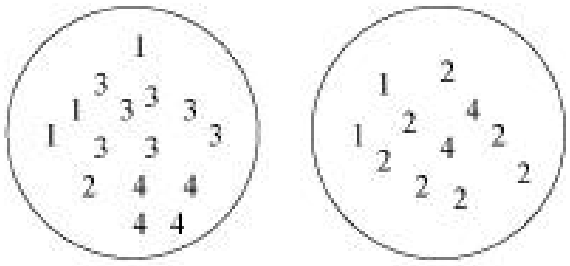
where μ is a $m\times1$ weight matrix and $\mathbf{X}$ is a $m\times m$ weight matrix. The matrices μ and $\mathbf{X}$ can be taken as the mean vector and the covariance matrix of $\mathbf{v}$, but this would require the assumption of a linear vector space, which is not desirable as discussed above. So, more appropriate weight matrices should be obtained. The main difficulty is that it is not known *a priori* what are the appropriate values of the weights. It is also difficult to apply a learning method to obtain the weight values (as in [46]) because the desired values of $Q_i(R)$ are also unknown *a priori*, and they depend very much on the type of classifier used.

3.2 Probabilistic Labeling

To resolve the problems highlighted above, we present a probabilistic method of computing $Q_i(R)$ by estimating the conditional probability $P(C_i \mid v_t)$. The approach encompasses the following characteristics:

- It can make use of the dissimilarity measures that are appropriate for the various types of features [26, 43] instead of the Euclidean distance.
- It does not require the use of weights to combine the various feature types.
- It adopts a learning approach that can adapt incrementally to the inclusion of new training samples, feature types, and semantic classes.

The probabilistic labeling method consists of two stages: (1) semantic class learning and (2) region labeling.



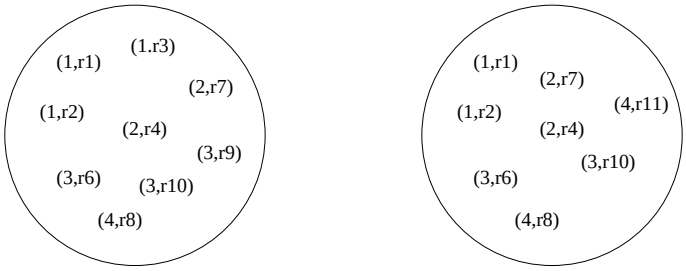
Cluster Ω_{t1} Cluster Ω_{t2}

Figure 3.1: *Clustering of training samples for a feature type t. The labels 1, 2, 3, 4 denote the semantic classes to which the training samples belong.*

3.2.1 Semantic Class Learning

The goal of the semantic class learning stage is to determine the conditional probabilities associated with each semantic class. It first clusters a set of training sample regions R_j, each assigned a pre-defined semantic class C_i, according to each feature type using the dissimilarity measure that is appropriate for the feature type (see Section 3 for details). This process produces a set of clusters Ω_{tk}, for each feature type t (Figure 3.1). After clustering, the conditional probability $P(C_i \,|\, \Omega_{tk})$ for semantic class C_i given cluster Ω_{tk} is estimated. Assuming that the distribution within each cluster is uniform, then $P(C_i \,|\, \Omega_{tk})$ can be estimated from the number of regions in the cluster:

$$P(C_i \,|\, \Omega_{tk}) = \frac{P(C_i, \Omega_{tk})}{P(\Omega_{tk})} = \frac{|C_i \cap \Omega_{tk}|}{|\Omega_{tk}|} \tag{3.3}$$



Cluster k of Feature Type t Cluster k' of Feature Type t'

Figure 3.2: *Combining clusters of different feature types.*

where $|\Omega_{tk}|$ denotes the number of regions in cluster Ω_{tk}, and $|C_i \cap \Omega_{tk}|$ the number of regions in Ω_{tk} that belong to semantic class C_i. Figure 3.1 shows an example of how to calculate $P(C_i \,|\, \Omega_{tk})$ for feature type t. For example, $P(C_1 \,|\, \Omega_{t1}) = 3/15 = 0.20$, where 3 is the number of data that come from class C_1 and 15 is the total number of data being clustered into cluster 1. Similarly, $P(C_1 \,|\, \Omega_{t2}) = 2/11 \simeq 0.18$. To combine multiple feature types, we can determine the cluster combinations $\Psi(\tau, \kappa, n) = \{\Omega_{\tau(1),\kappa(1)}, \ldots, \Omega_{\tau(n),\kappa(n)}\}$ that have high probabilities of associating with some semantic classes C_i:

$$\begin{aligned} P(C_i \,|\, \Psi(\tau,\kappa,n)) &= P(C_i \,|\, \Omega_{\tau(1),\kappa(1)}, \ldots, \Omega_{\tau(n),\kappa(n)}) \\ &= \frac{\left| C_i \cap \bigcap_l \Omega_{\tau(l),\kappa(l)} \right|}{\left| \bigcap_l \Omega_{\tau(l),\kappa(l)} \right|}. \end{aligned} \tag{3.4}$$

The functions $\tau(l)$, $l = 1, \ldots, n$, denote a combination of feature types and $\kappa(l)$ a combination of cluster indices. Figure 3.2 shows an example of how to compute $P(C_i \,|\, \Psi(\tau, \kappa, n))$. In this example, the pairs $(1, r1), (2, r_2), etc.$ refer to region $r1$ with class label 1, region $r2$ with class label 2, and so on. So, from Figure 3.2, $P(C_2, \,|\, \Omega_{tk}, \Omega_{t'k'}) = 2/7 \simeq 0.29$,

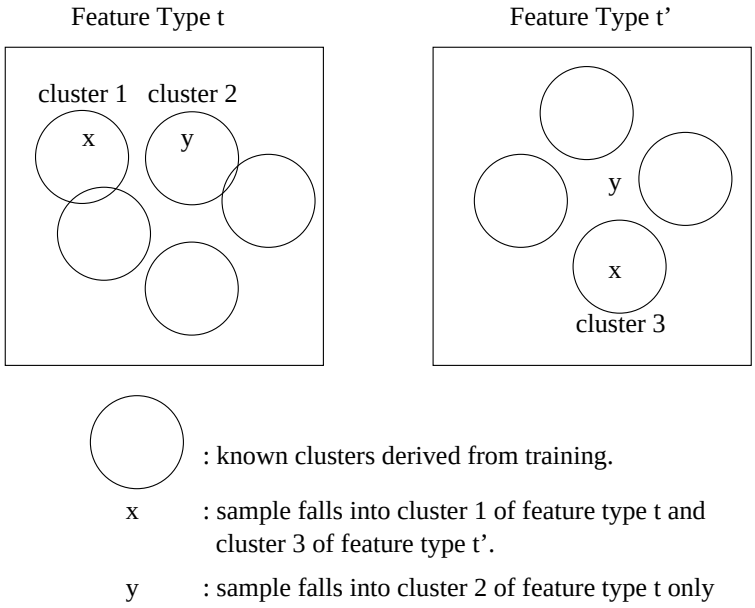


Figure 3.3: *An example to illustrate the labeling process.*

this value is larger than both $P(C_2 \mid \Omega_{tk}) = 2/9 \simeq 0.22$ and $P(C_2 \mid \Omega_{t'k'}) = 2/8 = 0.25$.

In practice, a semantic class is often associated with a specific combination of feature types. So, it is necessary to compute only the conditional probabilities $P(C_i \mid \Psi(\tau, \kappa, n))$ that are significantly larger than zero; those close to zero can be regarded as zero. The cluster combination $\Psi(\tau, \kappa, n)$, the associated semantic classes C_i, and the corresponding conditional probability values $P(C_i \mid \Psi(\tau, \kappa, n))$ are stored for region labeling.

3.2.2 Region Labeling

After the learning stage, a region R can be labeled by determining the associated semantic classes. Given the region R which contains a set of feature values v_t, the clusters that are nearest to the feature values v_t, for each feature type t, are determined. Next, the nearest clusters found are matched with the stored cluster combinations, obtained during the learning stage, that are associated with some semantic classes (Figure 3.3 shows an example of this labeling process, data sample y is labeled based on feature t only.).

The confidence measure $Q_i(R)$ can now be computed from the conditional probabilities of the matching cluster combinations $\Psi(\tau, \kappa, n)$:

$$Q_i(R) = \max_{\tau,\kappa,n} P(C_i \mid \Psi(\tau, \kappa, n)) \,. \tag{3.5}$$

Note that $Q_i(R)$ as defined in Eq. 3.5 is no longer a conditional probability:

- The C_i's in Eq. 3.5 may be conditioned on different sets of feature types.
- While $\sum_i P(C_i \mid \Psi(\tau, \kappa, n)) = 1$ for each cluster combination $\Psi(\tau, \kappa, n)$, the sum $\sum_i Q_i(R) \neq 1$ in general.

Nevertheless, $Q_i(R)$ is well-founded on probability theory and is, thus, a good measure of the confidence that region R belongs to class C_i.

3.2.3 Discussion

The semantic labeling method presented above computes $Q_i(R)$ from the probabilities conditioned on the clusters nearest to the feature values v_t of R instead of the probabilities conditioned on the feature values v_t. In principle, it is possible to estimate $P(C_i \mid v_t)$ using more sophisticated methods such as Gaussian mixture, e.g.,

$$P(C_i \mid v_t) = \sum_k w_k \exp\left(-\frac{\|v_t - \mu_{tk}\|^2}{2\,\sigma_{tk}^2}\right) \tag{3.6}$$

where w_k is a weighting factor that can be optimized during learning, μ_{tk} and σ_{tk} are derived from the location and radius of cluster Ω_{tk}, and $\|v_t - \mu_{tk}\|$ is an appropriate dissimilarity measure for feature type t. However, estimation of probability density that is conditioned on a combination of feature types is more complex. Let $\mathbf{v} = [v_1, \ldots, v_m]^T$ denote a vector that combines the feature values v_t, $t = 1, \ldots, m$. Then, a natural extension of Eq 3.6 to the multi-dimensional case is

$$P(C_i \mid \mathbf{v}) = \sum_k w_k \exp\left(-\frac{1}{2}(\mathbf{v} - \mu)^T \mathbf{X}^{-1} (\mathbf{v} - \mu)\right) \tag{3.7}$$

where $\mu = [\mu_{1,\kappa(1)}, \ldots, \mu_{m,\kappa(m)}]^T$ is the vector of the centroids of the clusters in which the feature values v_t lie, and $\mathbf{X}$ is an appropriate weight matrix. That is, this method defines the conditional probability in terms of a quadratic form of $\mathbf{v}$, which is undesirable (as discussed in Section 3.1). Therefore, in our current formulation, the probability distribution is assumed to be uniform within each cluster and 0 outside.

It is easy to see that the semantic class learning method discussed above is incremental. In some cases, feature types and semantic classes can be added without having to

re-run the learning algorithm on the entire set of training samples. Let us consider the following cases. Suppose that clusters Ω_{tk} have been derived for feature type $t = 1, \ldots, m$ and cluster index k.

- Add a new feature type $m + 1$.

  In this case, we have to cluster the training samples only with respect to the new feature type $m + 1$ to obtain clusters $\Omega_{m+1,k}$. Given these new clusters, a new set of conditional probabilities $P(C_i \mid \Psi(\tau, \kappa, n)$ can be computed to associate the semantic classes C_i with the cluster combinations $\Psi(\tau, \kappa, n)$.

- Add a new semantic class C' to existing training samples.

  Adding a new semantic class to existing training samples does not change their cluster memberships because the samples are clustered based on their feature values in the feature space. Therefore, the new semantic class can be added to the samples without the need to cluster them again, and a new set of conditional probabilities $P(C' \mid \Psi(\tau, \kappa, n))$ can be computed.

- Add a new semantic class with new training samples.

  Even for this case, it is still possible to avoid clustering existing training samples all over again. For example, the centroid of an existing cluster can represent the existing training samples that fall within the cluster. If the existing samples have been well clustered, then further clustering does not change their cluster membership significantly. So, the learning algorithm needs to cluster only the new training samples together with the existing cluster centroids, with more weights

given to the cluster centroids because each centroid represents a set of existing training samples instead of a single sample.

In comparison, commonly used classifiers such as neural network and support vector machines are not incremental. Addition of new feature types, semantic classes, or training samples always entails the re-training of the classifiers on the entire set of training samples.

3.3 Algorithms

The semantic labeling method described in previous section consists of several main algorithms: region clustering, probability estimation and region labeling. The first two algorithms are used for semantic class learning.

3.3.1 Region Clustering

Training sample regions R_j are clustered according to individual feature type t to obtain clusters Ω_{tk}. An adaptive k-means clustering algorithm is used so that the appropriate number of clusters can be determined automatically. The algorithm can be summarized as follows:

Adaptive Clustering

Repeat

For each feature value v_t of each region,

Find the nearest cluster Ω_{tk} to v_t.

If no cluster is found or distance $d(\Omega_{tk}, v_t) \geq s_t$,

create a new cluster with feature v_t;

Else, if $d(\Omega_{tk}, v_t) \leq r_t$,

add feature value v_t to cluster Ω_{tk}.

For each cluster Ω_{ti},

If cluster Ω_{ti} has at least N_m feature values,

update centroid of cluster Ω_{ti};

Else, remove cluster Ω_{ti}.

The centroid of cluster Ω_{ti} is a generalized mean of the feature values in the cluster (see Chapter 4 for more discussion). The function $d(\Omega_{tk}, v_t)$ is a dissimilarity measure appropriate for the feature type t [26, 43] (see Chapter 4 for more details). N_m is the minimum number of feature values in a cluster so that a cluster will not be too small.

The adaptive clustering algorithm groups a feature value v_t into its nearest cluster if it is near enough ($d(\Omega_{tk}, v_t) \leq r_t$). On the other hand, if the feature value is far enough ($d(\Omega_{tk}, v_t) \geq s_t$) from its nearest cluster, then a new cluster is created. Otherwise, it is left unclustered and will be considered again in the next iteration. This clustering algorithm, thus, ensures that each cluster has a *maximum radius* of r_t and that the clusters are separated by the distance of approximately s_t called the *nominal cluster separation*.

The maximum radius r_t, for feature type t, is computed by measuring the average distance between the samples in a class:

$$r_t = \frac{1}{M} \sum_i \frac{1}{N(i)} \sum_{R_j, R_k \in C_i} d(v_{tj}, v_{tk}) \tag{3.8}$$

where M is the number of classes, $N(i)$ is the number of sample pairs in class C_i, and v_{tj} and v_{tk} are the type-t feature values of regions R_j and R_k in class C_i. The same algorithm is applied to different feature types separately. The value of s_t is defined as a multiple γ of r_t, i.e., $s_t = \gamma r_t$. Reasonable values of γ range from 0 (for complete overlapping of the clusters) to 2 (for non-overlapping of clusters). Therefore, the algorithm can automatically determine the number of clusters required to effectively represent the clusterings of feature values. It also ensures that each cluster has a significant number of (at least N_m) feature values; otherwise, the cluster is removed.

The clustering algorithm terminates after running for several iterations (usually 10, for efficiency sake). At this stage, some training samples may still be unclustered. To handle this case, the following algorithm is used:

Adjustment of r_t

For each feature value v_t of each region,

Find the nearest cluster Ω_{tk} to v_t.

If $d(\Omega_{tk}, v_t) \geq r_t$,

update r_t to $r_{tk} = d(\Omega_{tk}, v_t)$.

This adjustment permits all training samples to be clustered by enlarging the cluster radius r_t to r_{tk}, which can be different for different clusters. If the number of unclustered

training samples is small, updating r_t to r_{tk} is not necessary.

3.3.2 Probability Estimation

After clustering the regions according to individual feature types, the conditional probability $P(C_i \mid \Omega_{tk})$ of each cluster Ω_{tk} is estimated. In addition, the conditional probabilities $P(C_i \mid \Psi(\tau, \kappa, n))$ of various cluster combinations $\Psi(\tau, \kappa, n)$ are also estimated.

In order to fully assess the accuracy of the semantic labeling approach, the conditional probability for all possible combinations of 1 to 4 features are computed in the current implementation so that the combinations with the highest probabilities can be identified. In actual applications, a cluster selection procedure can be performed to select candidate cluster combinations that are likely to yield significant probabilities. This method would remove the need to compute the probabilities for all possible combinations.

Probability Estimation

For each semantic class C_i,

For $n = 1, \ldots, 4$,

For each cluster combination $\Psi(\tau, \kappa, n)$,

Compute $P(C_i \mid \Psi(\tau, \kappa, n))$.

If $P(C_i \mid \Psi(\tau, \kappa, n))$ is larger than 0,

store C_i, $\Psi(\tau, \kappa, n)$, and $P(C_i \mid \Psi(\tau, \kappa, n))$.

The cluster combinations $\Psi(\tau, \kappa, n)$, their associated semantic classes C_i, and the corre-

sponding conditional probabilities $P(C_i \mid \Psi(\tau, \kappa, n))$ that are larger than zero are stored for region labeling.

For each combination $\Psi(\tau, \kappa, n)$, an efficient set intersection algorithm is used to calculate $P(C_i \mid \Psi(\tau, \kappa, n))$ according to Eq. 3.4.

Set Intersection

Given Sets Ω_1 and Ω_2, sort the elements in each set in increasing order of element index.

Set L_1 to be the index of the smallest element in Ω_1,

Set L_2 to be the index of the smallest element in Ω_2.

While both sets Ω_1 and Ω_2 are not exhausted,

if $\Omega_1[L_1] < \Omega_2[L_2]$,

$L_1 + +$,

else if $\Omega_1[L_1] > \Omega_2[L_2]$,

$L_2 + +$,

else

add the common element, i.e., $\Omega[L_1] = \Omega[L_2]$, to the resulting Set Ω_r,

increment L_1 and L_2

The complexity of this set intersection algorithm is $O(n\ logn)$, n is the size of the smaller set between the two.

3.3.3 Region Labeling

The region labeling algorithm determines the combinations of feature values of a region that are associated with some semantic classes with high probabilities. These semantic classes and the probabilities are assigned to the region as its semantic labels. In general, a region can be assigned multiple semantic classes.

Region Labeling

Given a region R

Find the nearest cluster of type t in which each feature value v_t of R falls within the the radius r_{tk} of the corresponding nearest cluster k.

Find the combinations Ψ_j of these clusters that match the stored cluster combinations.

Retrieve the classes C_i and probabilities $P(C_i \mid \Psi_j)$ associated with the matching cluster combinations Ψ_j.

For each retrieved class C_i,

Find the Ψ_k with the largest probability:

$$P(C_i \mid \Psi_k) = \max_j P(C_i \mid \Psi_j)$$

Assign C_i and $P(C_i \mid \Psi_k)$ to R,

i.e., $Q_i(R) = P(C_i \mid \Psi_k)$.

For testing regions, it is possible that a feature value v_t may not fall within any known clusters of type t. This is because the testing region might be an outlier which never occurs in the training region. In this case, no cluster of type t is found.

3.3.3 Region Labeling

For the purpose of assessing the effectiveness of the semantic labeling method, region classification is also performed to assign the semantic label with the highest confidence to a region.

Region Classification

Given a region R

Perform region labeling.

Find the C_k with the largest confidence:

$$Q(C_k \mid R) = \max_i Q(C_i \mid R)$$

If $Q(C_k \mid R) >$ threshold Γ,

assign C_k to R;

else assign "unknown class" to R.

When the probability is low, it is better to assign the "unknown class" label to a region than to assign it a wrong label.

Chapter 4

Features and Dissimilarity Measures

In the current implementation, four different types of features are extracted for each image block. They are adaptive color histograms, Gabor features, multi-resolution simultaneous autoregressive (MRSAR) features and edge histograms. These features are known as good features for image classification and retrieval [26, 29, 32, 39] in general.

4.1 Color Histograms

Color information vary substantially over an image or an image part. Such information can be more fully described by a distribution of features instead of using a single feature. Histograms are often used to estimate the distributions. There are two methods of generating histograms: *fixed binning* and *adaptive binning*. Typically, a fixed-binning method induces histogram bins by partitioning the color space into rectangular bins [13, 14, 35, 38, 47, 50, 55]. Once the bins are derived, they are fixed and

the same binning scheme is applied to all images. On the other hand, adaptive binning adapts to the actual distributions of colors in image [4, 15, 17, 19, 24, 43, 45]. As a result, different binnings are induced for different images. It is a common understanding that adaptively-binned histograms can represent the distributions of colors in images more efficiently than do histograms with fixed binning [4, 15, 17, 43, 45]. However, existing systems ([13, 14, 22, 35, 50, 47, 55]) almost exclusively adopt fixed-binning histograms because among existing well-known dissimilarity measures, only the Earth Mover's Distance (EMD) can compare histograms with different binnings [43, 45]. EMD is computationally more expensive than other dissimilarity measures because it requires an optimization process.

4.1.1 Fixed-Binning Histogram

There are two types of *fixed binning* schemes: *regular partitioning* and *clustering*. The first method simply partitions the axes of a target color space into regular intervals, thus producing rectangular bins. Typically, one of the three color axes is regarded as conveying more important information and is partitioned into more intervals than are the other two axes. For example, VisualSeek [50] partitions the HSV space into 18×3×3 color bins and 4 grey bins, producing 166 bins. PicHunter [14] also partitions the HSV space in a similar manner. The CIELAB and CIELUV spaces have been also used [35, 47] because they are more perceptually uniform [3]. In partitioning these spaces, bins that correspond to illegal RGB colors are usually discarded.

The second method partitions a color space into a large number of cells, which are then clustered by a clustering algorithm such as the k-means. For example, QBIC [22] partitioned the RGB space into 16×16×16 cells, mapped the cells to a modified Munsell HVC space, and then clustered the cells into k clusters. Vailaya et al. [55] applied a similar method but mapped the RGB cells into the HSV space, where 64 bins were produced. Quicklook [13] mapped sRGB [1] cells into CIELAB and clustered them into 64 bins.

4.1.2 Adaptive Color Histogram

In [27, 26], we formulated a dissimilarity measure for color histogram based on weighted correlation (WC) of bin similarity. Here we just give a summary of the formulation.

An adaptive histogram $H = (n, \mathcal{C}, \mathcal{H})$ is defined as a 3-tuple consisting of a set $\mathcal{C}$ of n bins $\mathbf{c}_i$, $i = 1, \ldots, n$, and a set $\mathcal{H}$ of corresponding bin counts $h_i \geq 0$. The weighted correlation between histograms $G = (m, \{\mathbf{b}_i\}, \{g_i\})$ and $H = (n, \{\mathbf{c}_i\}, \{h_i\})$, denoted as $G \cdot H$, is defined as

$$G \cdot H = \sum_{i=1}^{m} \sum_{j=1}^{n} w(\mathbf{b}_i, \mathbf{c}_j)\, g_i, h_j \ . \tag{4.1}$$

The similarity $w(\mathbf{b_i}, \mathbf{c_j})$ between bins $\mathbf{b_i}$ and $\mathbf{c_j}$ is defined in terms of the volume of intersection V_s between the bins:

$$w(\mathbf{c}, \mathbf{c}') = w(\alpha) = \frac{V_s}{V} = \begin{cases} 1 - \dfrac{3}{4}\alpha + \dfrac{1}{16}\alpha^3 & \text{if } 0 \leq \alpha \leq 2 \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

where $\alpha = d/R$ is the ratio of the cluster separation d and the bin radius R.

A histogram H can be normalized into $\overline{H}$ by dividing each bin count by the histogram norm $\|H\| = \sqrt{H \cdot H}$. The similarity $s(G, H)$ between G and H is defined as the weighted correlation between their normalized forms: $s(G, H) = \overline{G} \cdot \overline{H}$, and the dissimilarity $d(G, H)$ between them is defined as $d(G, H) = 1 - s(G, H)$.

The mean histogram is defined in terms of histogram merging operation. Let histogram $G = X \cup Y$ and $H = X' \cup Z$ such that X and X' have the same set of bins and X, Y, and Z have disjoint sets of bins. The the merged histogram $G \uplus H = (X \cup Y) \uplus (X' \cup Z) = (X + X') \cup Y \cup Z$. That is, two histograms are merged by collecting all the bins and add the bin counts of identical bins. Note that it is always possible to express two histograms G and H in the form given previously for histogram merging to be well-defined. It is shown in [27] that the merging of histograms H_i is equivalent to the mean of histogram M:

$$M = \biguplus_i \overline{H}_i \tag{4.3}$$

The usual definition of mean divides the sum by the number of items that are added together. For adaptive histograms, this division is not necessary because histogram similarity and dissimilarity are defined in terms of normalized histograms (E.q 4.3) ([27]).

4.1.3 Adaptive Binning

Adaptive binning can be achieved by the k-means clustering algorithm or its variants. To group the pixels into clusters, the same algorithm used for region clustering (See

Chapter 3) is used with different ways of calculating the distance from a pixel p to cluster centroid $\mathbf{c}_k$ of cluster k and cluster centroid. In this context, the cluster centroid is the average color of all the pixels within the same cluster.

The distance d_{kp} between the centroid $\mathbf{c}_k$ of cluster k and pixel p with color $\mathbf{c}_p$ is defined as the CIE94 color-difference equation:

$$d_{kp} = \left[\left(\frac{\Delta L^*}{k_L S_L} \right)^2 + \left(\frac{\Delta C^*_{ab}}{k_C S_C} \right)^2 + \left(\frac{\Delta H^*_{ab}}{k_H S_H} \right)^2 \right]^{1/2} \tag{4.4}$$

where ΔL^*, ΔC^*_{ab}, and ΔH^*_{ab} are the differences in lightness, chrome, and hue between $\mathbf{c}_k$ and $\mathbf{c}_p$, $S_L = 1$, $S_C = 1 + 0.045\,\bar{C}^*_{ab}$, $S_H = 1 + 0.015\,\bar{C}^*_{ab}$, and $k_L = k_C = k_H = 1$ for reference conditions. The variable $\bar{C}^*_{ab}$ is the geometric mean between the chrome values of $\mathbf{c}_k$ and $\mathbf{c}_p$, i.e., $\bar{C}^*_{ab} = \sqrt{C^*_{ab,k} C^*_{ab,p}}$. The CIE94 color-difference equation is used instead of the simpler Euclidean distance in CIELAB space because recent psychological studies show that CIE94 is more perceptually uniform that Euclidean distance [20, 36, 51].

This adaptive clustering algorithm is similar to that of Gong et al. [17]. Both algorithms ensure that the clusters are not too large in volume and not too close to each others. However, our adaptive algorithm is simpler than that in [17], does not require seed initialization, and can automatically determine the appropriate number of clusters.

In practice, for efficiency sake, the algorithm is repeated for only 10 iterations. When the algorithm terminates, some colors may still be unclustered. During color quantization or histogram generation, these unclustered colors are quantized to the colors of their nearest clusters. Empirical tests show that having a small amount of unclustered colors do not produce significant error in the color quantization results. For instance, 5%

unclustered colors contribute only a 1% increase in the mean error of color quantization compared to the case in which all the colors are clustered. In fact, leaving some colors unclustered makes the algorithm more robust against noise colors that differ significantly from other main colors in the image. These noise colors typically occur at abrupt color discontinuities in the images.

4.1.4 Quantitative Evaluation of Adaptive Color Histogram

In the initial feature evaluation, we did a quantitative evaluation of the adaptive color histograms and weighted correlation. The quantitative evaluation consists of three tests. The first test evaluated the accuracy of adaptive clustering in retaining color information. The second and third tests evaluated the combined performance of adaptive color histograms and weighted correlation in image retrieval and classification.

Color Retention

In this test, the performance of the adaptive color histograms was compared with the color histograms generated by regular partitioning and color space clustering. The colors of the images were assumed to be represented in the sRGB space [1], and the target color space was CIELAB.

Test Setup

The adaptive color histograms were tested with cluster radius R ranging from 7.5 to 22.5 and nominal cluster separator factor γ ranging from 1.1 to 1.5. With this

set of parameter values, about 95% or more of the colors in every image could be clustered during the clustering process.

For regular partitioning, the L^*-axis of the CIELAB space was partitioned into l equal intervals (l = 8, 10, 12, 14, 16), and the a^*- and b^*- axes were partitioned into m equal intervals (m = 5, 8, 10 and $m \leq l$). The centroids of the bins were mapped back to the sRGB space and bins with illegal sRGB values were discarded. For color space clustering, the CIELAB space was partitioned into 32×32×32 equal partitions and the bin centroids were clustered using the same adaptive clustering algorithm, with $7.5 \leq R \leq 20$ and $1.1 \leq \gamma \leq 1.5$.

As the test images, 100 visually colorful images were randomly selected from the Corel 50,000 photo collection. The images had sizes of either 256×384 or 384×256.Color histograms were generated for each image using the three binning methods. For color space clustering and adaptive clustering, all the colors in the images, including those that were unclustered during the clustering process, were quantized to the colors of their nearest clusters.

The performance of the three binning methods were measured by three indicators, namely, the number of bins or clusters produced, the number of empty bins, and the mean color error measured as the mean difference between the actual colors and the quantized colors (in CIE94 units). These performance indicators were averaged over all the images.

Color Error

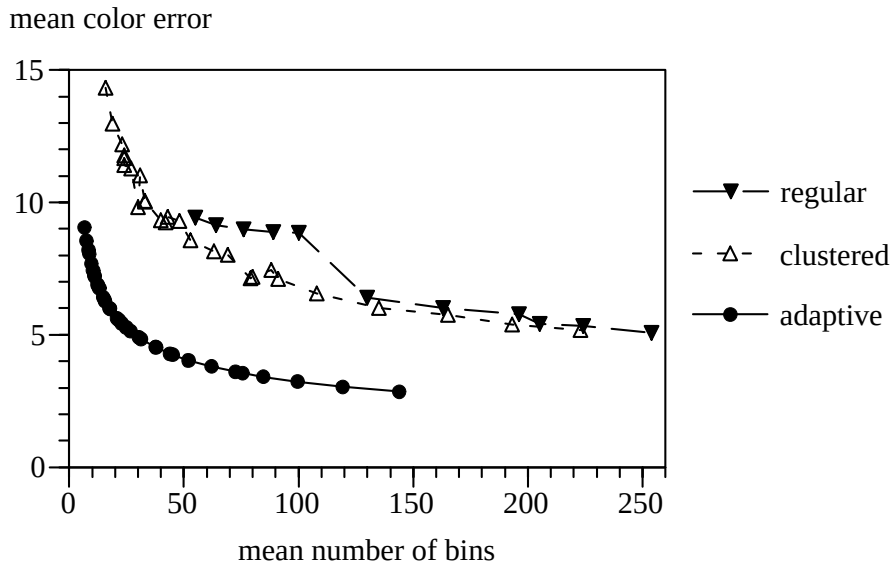


Figure 4.1: *Comparison of mean color errors of regular, clustered, and adaptive histograms.*

Experimental results show that the larger the bin volume (or cluster radius R) and the larger the bin separation γ, the smaller is the number of bins and the larger is the mean color error. Figure 4.1 shows that regular partitioning produced slightly larger mean color error compared to color space clustering, while adaptive clustering produced the smallest error. Given a fixed number of bins, regular and clustered histograms have errors that are about twice those of adaptive histograms.

Empty Bins

Figure 4.2 shows the average percentage of empty bins in the regular and clustered histograms. With a large number of bins, both types of histograms have 50% or more empty bins. With a small number of bins, clustered histograms have as few as

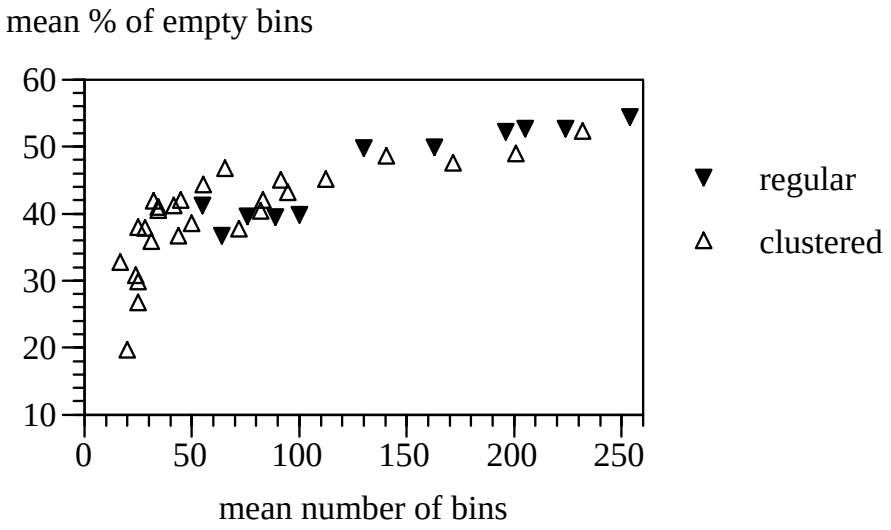
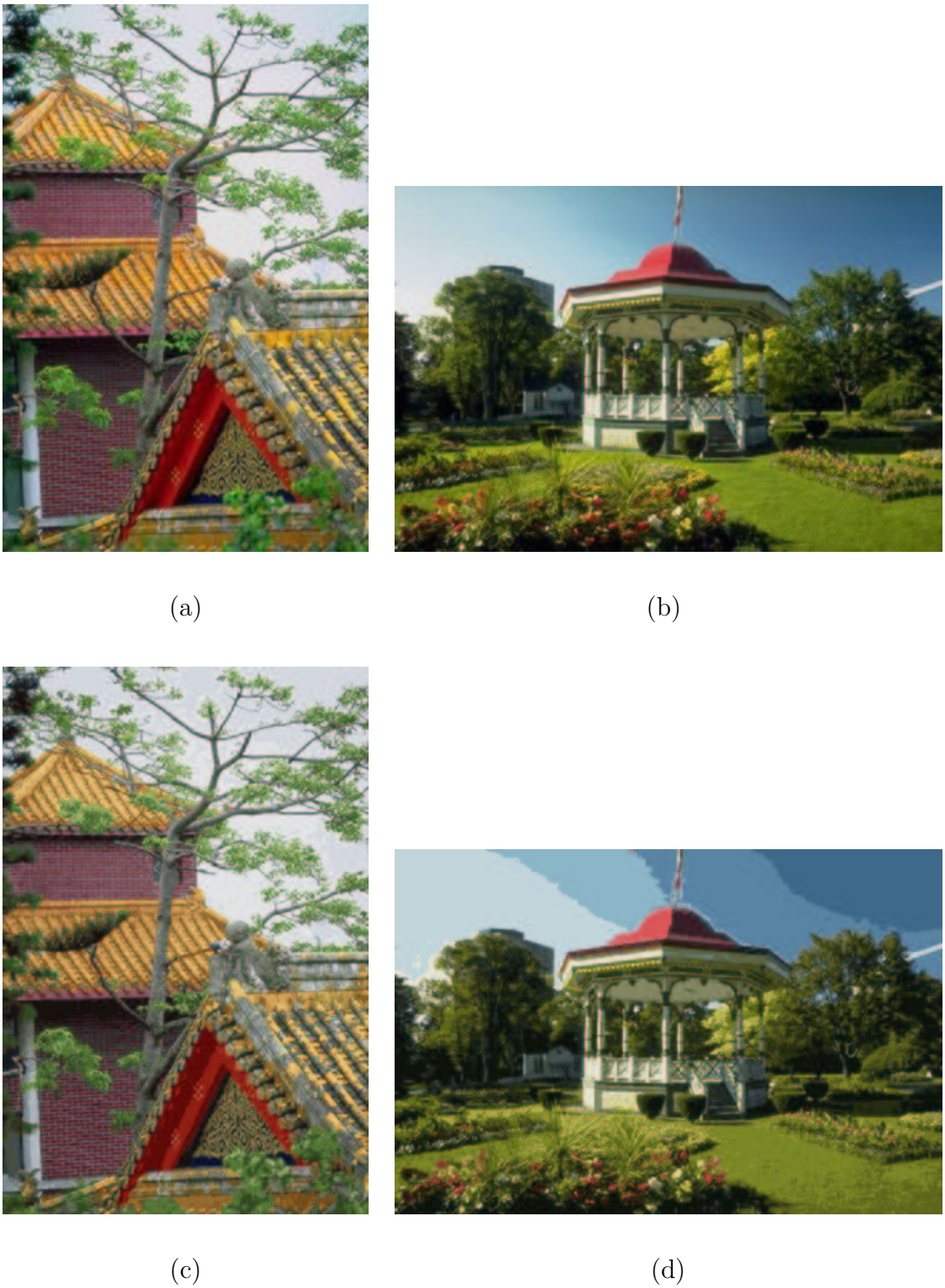


Figure 4.2: *Average percentage of empty bins in regular and clustered histograms.*

20% empty bins. The adaptive histograms have no empty bins. These test results show that adaptive histograms can retain color information more accurately with fewer bins than regular and clustered histograms.

Visual Quality

Visual inspection of the color-quantized images (Fig. 4.3) reveals that images with a color error of 5 or less look almost indistinguishable from the original images except at regions where banding occurs such as clear blue sky. This is the result of quantizing the gradually varying colors into discrete bins. This observation matches recent psychological study [51] very well, which shows that that human's color acceptability threshold is 4.5. That is, two colors with a color difference of less than 4.5 are regarded as practically identical. Note that the acceptability



(a) (b)

(c) (d)

Figure 4.3: *Color quantization results. The original images contain (a) 71599 colors and (b) 46218 colors. The color-quantized images contain only (c) 39 colors and (d) 31 colors.*

threshold is slightly larger than the perceptibility threshold of 2.2 [51], which is the threshold below which two colors are perceptually indistinguishable .

Discussion

Existing systems typically use 64-bin clustered histograms or more than 150 bins for regular histograms. Their respective mean color errors are about 8 and 6, with 45% and 50% empty bins. In comparison, 64-bin adaptive histograms can achieve a color error of about 3.5, lower than human acceptability threshold, with no empty bins.

In the subsequent tests, the parameter values of clustered and adaptive binning methods were fixed at $R = 10$ and $\gamma = 1.5$ because this combination yielded good color retention with small number of bins. With these parameter values, the adaptive binning method produced an average of 37.8 bins with a mean color error of 4.53, and the color space clustering method produced 80 bins, a mean color error of 7.19, and 42% empty bins. In principle, the mean color error of color space clustering can be reduced to, say, below 5 so that it is comparable to that of adaptive binning. However, this will require the clustered histograms to have much more than 250 clusters—a value that is both impractical and beyond our experimental range. It was not necessary to test regular partitioning further because its performance was similar to that of color space clustering.

Image Retrieval

This test assessed the combined performance of binning schemes and dissimilarity measures in image retrieval.

In the image retrieval test of Puzicha et al. [43], random samples of pixels were extracted from the test images. Samples that were drawn from the same image should have similar distributions and were regarded as belonging to the same class. This kind of test samples are useful for testing the performance of various similarity measures in computing global similarity between two images.

Test Setup

A different kind of test samples were prepared for our tests. Each of the 100 images used in the color retention test (Section 4.1.4) were regarded as forming one query class. These images were scaled down and each embedded into 20 different host images, giving a total of 2000 composite images at each scaling factor. The scaled images were used as query images, and the composite images that contained the same embedded images were regarded as relevant. A different kind of test samples were prepared for our tests. Each of the 100 images used in the color retention test (Section 4.1.4) were regarded as forming one query class. These images were scaled down and each embedded into 20 different host images, giving a total of 2000 composite images at each scaling factor. The scaled images were used as query images, and the composite images that contained the same embedded images were regarded as relevant. This test paradigm should be useful for testing the combined

performance of binning schemes and dissimilarity measures in retrieving images that c ontain a particular target region or color distribution of interest. We feel that this test more closely resembles the retrieval of complex images containing one or more regions of interests compared to that in [43]. In the test, scaling factors for image width/height of 1/4, 1/2, and 3/4 were used. These values gave rise to embedded images with area scaling factors of 1/16, 1/4, and 9/16 compared to the original images.

In the experimental tests, the weighted correlation dissimilarity (WC) given in Section 4.1.2 as compared with three existing dissimilarity measures, namely L_2 (Euclidean), Jessen Difference Divergence (JD) [1], and Earth Mover's Distance (EMD).

- L_2 (Euclidean) distance:

$$d(G, H) = \left(\sum_i (g_i - h_i)^2 \right)^{1/2} \tag{4.5}$$

- Jessen difference divergence (JD):

$$d(G, H) = \sum_i \left(g_i \log \frac{g_i}{m_i} + h_i \log \frac{h_i}{m_i} \right) \tag{4.6}$$

  where $m_i = (g_i + h_i)/2$.

[1] The formula that Puzicha et al. [43] called "Jeffreys divergence" is more commonly known as "Jessen difference divergence" in Information Theory literature [7, 8, 53]. Jeffreys divergence, as given in the literature [7, 8, 23, 25, 53], takes the form $\sum_i (g_i - h_i) \log(g_i/h_i) = \sum_i [g_i \log(g_i/h_i) + h_i \log(h_i/g_i)]$.

- Earth Mover's distance (EMD) [44]:

$$d(G,H) = \frac{\sum_{i,j} f_{ij}\, d(\mathbf{b}_i, \mathbf{c}_j)}{\sum_{i,j} f_{ij}} \tag{4.7}$$

where $d(\mathbf{b}_i, \mathbf{c}_j)$ denotes the dissimilarity between bins $\mathbf{b}_i$ and $\mathbf{c}_j$, and f_{ij} is the optimal flow between G and H such that the total cost $\sum_{i,j} f_{ij}\, d(\mathbf{b}_i, \mathbf{c}_j)$ is minimized, subject to the constraints:

$$f_{ij} \geq 0\ ,\ \ \sum_i f_{ij} \leq h_j\ ,\ \ \sum_j f_{ij} \leq g_i\ ,\ \ \sum_i \sum_j f_{ij} = \min(\sum_i g_i, \sum_j h_j)\ . \tag{4.8}$$

The dissimilarity $d(\mathbf{b}_i, \mathbf{c}_j)$ between two bins is typically defined as a monotonic increasing function of the ground distance between the bins.

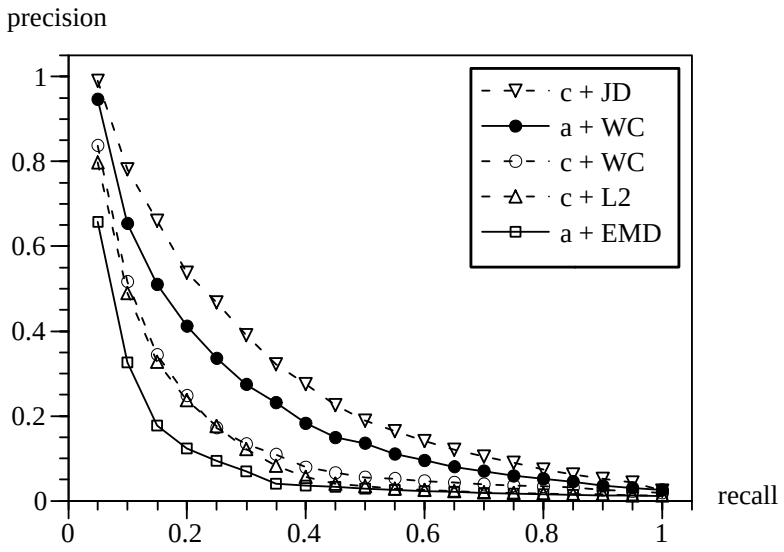
Among these dissimilarity measures, L_2 served as the base case of the performance evaluation. JD and EMD are reported in [43] to yield good performance, respectively, for large and small sample sizes. Other dissimilarity measures evaluated in [43] are expected to yield similar results and are therefore omitted. WC is tested with both clustered and adaptive histograms whereas L_2 and JD could be tested only with clustered histograms. The program for EMD was downloaded from Rubner's web site (`http://robotics.stanford.edu/~rubner`), and was tested only with adaptive histograms due to its longer execution time. The CIE94 distance was used as EMD's ground distance because it is more perceptual uniform than Euclidean distance in the CIELAB space, and was taken as the dissimilarity between two bins.

Results and Discussion

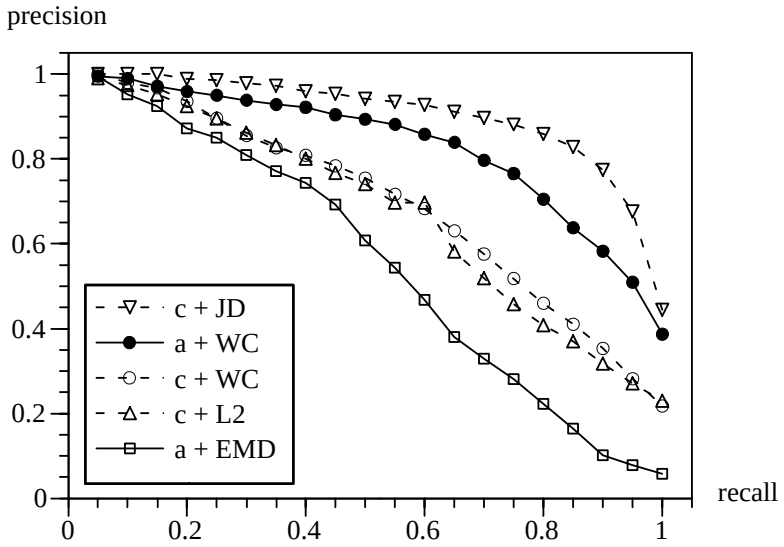
Figure 4.4 plots the precision-recall curves of the image retrieval results for scaling factors 1/2 and 3/4 of both image height and image width. The curves for scaling factor of 1/4 are not shown because all combinations of binnings and dissimilarity measures performed poorly. They all had very low precision of less than 0.2 at recall rate of 0.1, and their precision dropped to about 0.01 at recall rate of 0.3 and above.

All five combinations performed significantly better for the larger scaling factor of 3/4 than for 1/2. For both scaling factors, clustered histograms together with JD (c + JD) performed best, with the adaptive histograms and WC (a + WC) combination following closely behind. The a + WC combination performed significantly better than c + WC, which had roughly the same performance as c + L_2. These results show that, given the same dissimilarity measure, adaptive histograms perform better than clustered histograms because they can describe color information more accurately and yet use fewer bins (Section 4.1.4).

Somewhat surprisingly, EMD (with adaptive histograms) performed poorer than L_2. Compared to the results in [43], which show that EMD performed better for small sample sizes, it is noted that our smallest scaling factor of 1/4 corresponds to an image size of 6144 pixels, which is far larger than the sample sizes used in [43]. Moreover, the adaptive histograms have an average of 37.8 bins, and they correspond to medium sized histograms in [43]. These parameter values may have



(a)



(b)

Figure 4.4: *Precision-recall curves of various combinations of binning methods (c: clustered, dashed line; a: adaptive, solid line) and dissimilarity measures (JD: Jeffrey divergence, WC: weighted correlation, L2: L_2 or Euclidean distance, EMD: Earth Mover's Distance). (a) Scaling = 1/2, (b) scaling = 3/4.*

obscured the strengths of EMD in extreme cases of small sample sizes and small number of bins. On the other hand, our choice of the number of bins, which was supported by the color retention test (Section 4.1.4), and the sample sizes should better resemble the retrieval of complex images with multiple regions in practice.

Image Classification

This test assessed the combined performance of binning schemes and dissimilarity measures in image classification.

Test Setup

The composite images generated in the retrieval tests (Section 4.1.4) were used for image classification test. The composite images that contained the same embedded image were considered as belonging to the same class. This would correspond to the practical application in which images containing the same region or color distribution are considered as identical.

The k-nearest-neighbor classifier with leave-one-out procedure was applied on each of the 2000 composite images. Odd values of k ranging from 1, 3, 5, 7, and 9 were chosen to remove the possibility of ties. Classification error, averaged over all 2000 images, were computed for each combination of binning scheme, dissimilarity measure, and k value

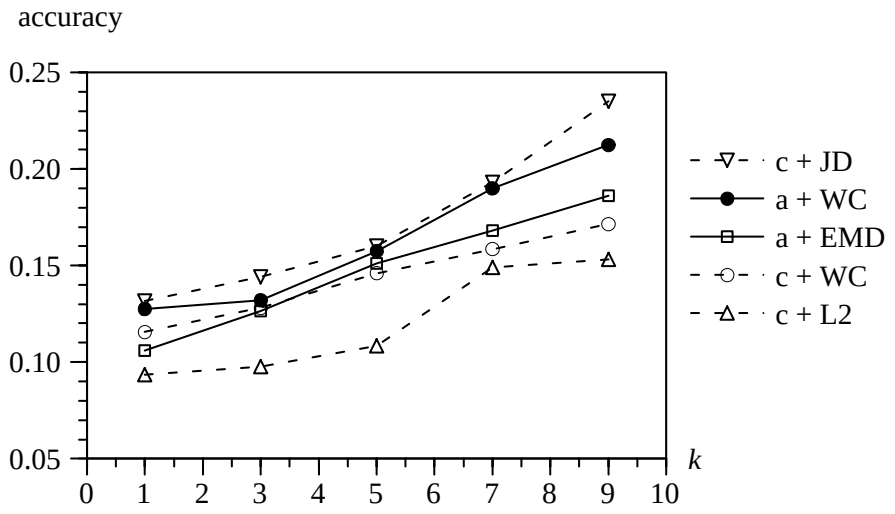
Results and Discussion

Figure 4.5 shows the classification performance for width/height scaling factors

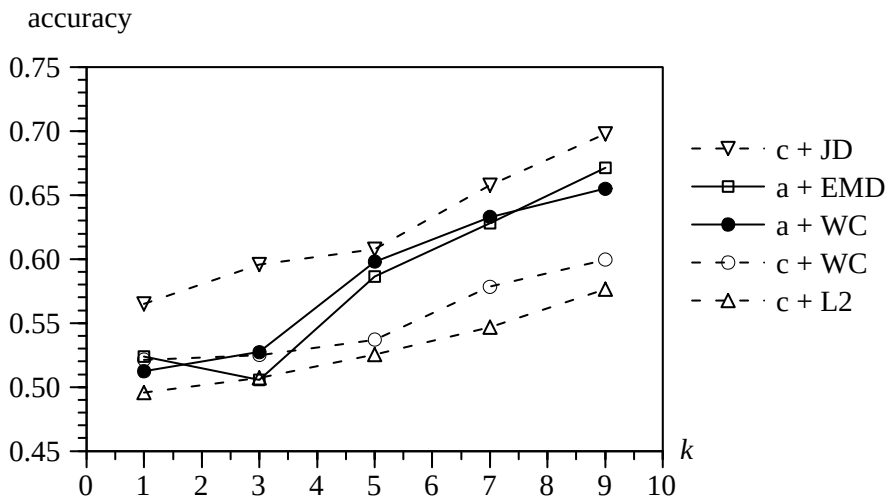
of 1/2 and 3/4. The curves for scaling factor of 1/4 are not shown because all combinations of binnings and dissimilarity measures performed poorly. All five combinations performed significantly better for the larger scaling factor of 3/4 than for 1/2. Moreover, their classification accuracies increased with increasing number of nearest neighbors k. Similar to the image retrieval results, c + JD performed best for both scaling factors, with a + WC following closely behind. The a + WC combination performed better than c + WC, and c + L_2 again had the lowest accuracy. These results again show that, given the same dissimilarity measure, adaptive histograms perform better than clustered histograms. Unlike in the retrieval tests, the performance of a + EMD was very good in the classification tests. The classification accuracy of a + EMD closely matched that of a +WC, especially for the larger scaling factor of 3/4.

Spatial Precision

To further investigate the cause of EMD's inconsistent performance, another performance index called *spatial precision* was computed. Spatial precision measures, for a given image I, the proportion of images among its k nearest neighbors that belong to the same class as I. Figure 4.6 plots the spatial precision averaged over all 2000 images for each k. The spatial precisions of the dissimilarity measures are smaller for a smaller image scaling factor and decrease with increasing value of k. The result shows that as the value of k (i.e., the neighborhood size) increases, more negative samples that belong to other classes are included in the

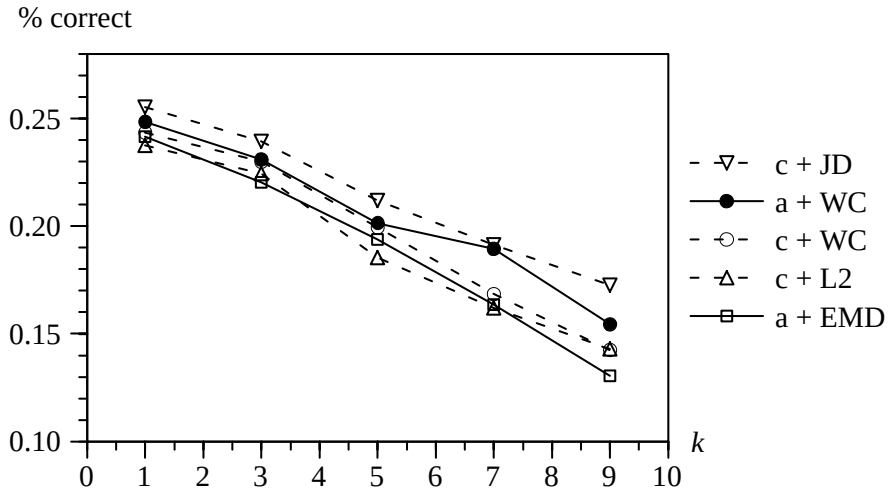


(a)

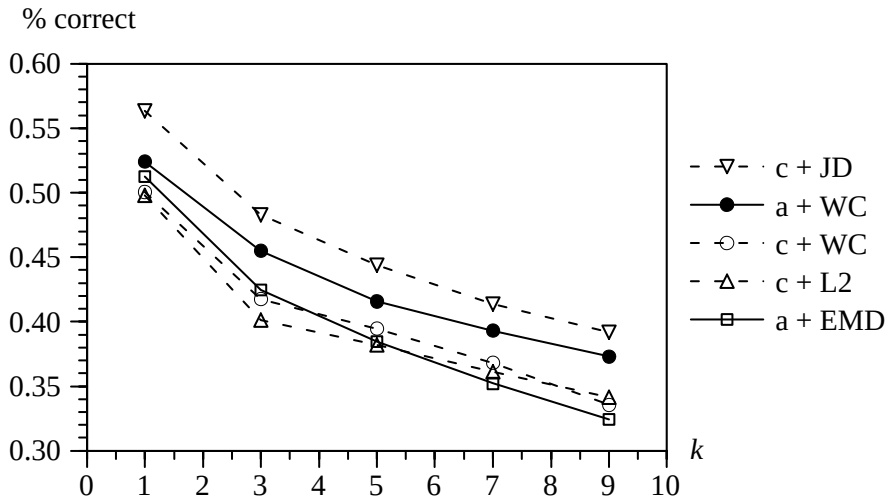


(b)

Figure 4.5: *Classification accuracy of various combinations of binning methods (c: clustered, dashed line; a: adaptive, solid line) and dissimilarity measures (JD: Jeffrey divergence, WC: weighted correlation, L2: L_2 or Euclidean distance, EMD: Earth Mover's Distance). (a) Scaling = 1/2, (b) scaling = 3/4.*



(a)



(b)

Figure 4.6: *Spatial precision of various combinations of binning methods (c: clustered, dashed line; a: adaptive, solid line) and dissimilarity measures (JD: Jeffrey divergence, WC: weighted correlation, L2: L_2 or Euclidean distance, EMD: Earth Mover's Distance). (a) Scaling = 1/2, (b) scaling = 3/4.*

neighborhood. However, given the large number of classes (100) in the test, it is possible that only a small number of negative samples from each class is included. As a result, the majority class can still be the correct class even when there are many negative samples. This is especially true for EMD since its spatial precision decreases faster than those of other dissimilarity measures.

Histogram Clustering

Given well defined dissimilarity measure and mean histogram, the traditional k-means clustering algorithm is reformulated by using weighted correlation as the distance measure and mean histogram as the cluster centroid. This experiment was conducted to measure the performance of histogram clustering.

Test Setup

400 composite images from 20 classes (20 from each class) were randomly chosen from the images generated from the retrieval test. The composite images that contained the same embedded image should be closer to each other than to the other images. Three tests were performed using the following combinations of color histograms and dissimilarity measures: (1) fixed clustered histograms and Euclidean distance (c + L2), (2) fixed clustered histograms with JD for cluster assignment and Euclidean for computing cluster centroid (c + JD/L2), and (3) adaptive histogram and weighted correlation dissimilarity (a + WC). For the first two cases, an ordinary k-means clustering was used. For the third case, the k-means clustering for adaptive histograms was used. For each case, separate clustering tests were

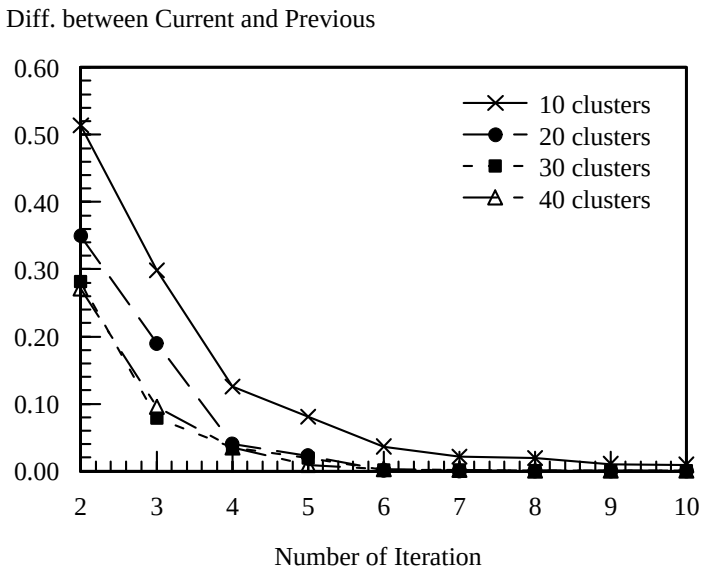


Figure 4.7: *Convergence test. (The difference between current and previous mean is calculated using weighted correlation.)*

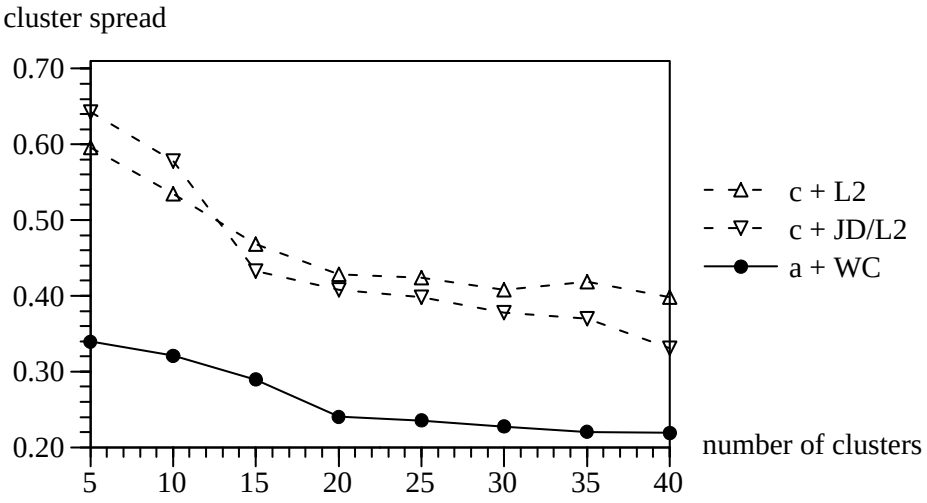
conducted with number of clusters ranging from 5 to 40.

Convergence Property of the Clustering Algorithm

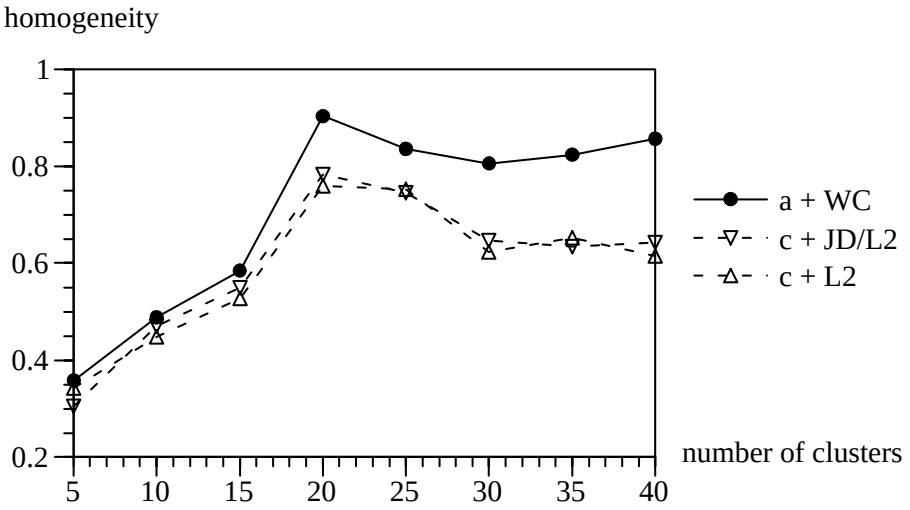
The rigorous proofs of the convergence properties of the traditional k-means algorithms are presented in [5]. To show that the algorithm does converges, we did a convergence test. From figure 4.7, we can see that the algorithm converges in fifth or sixth iteration and then becomes stabilized.

Cluster Spread and Cluster Homogeneity

Clustering performance is measured in terms of the *cluster spread* and *cluster homogeneity*.The cluster spread Ω is the effective radius of a cluster normalized by



(a)



(b)

Figure 4.8: *Comparison of (a) cluster spread and (b) cluster homogeneity between the three test cases.*

its distance to the nearest neighboring cluster:

$$\Omega = \frac{1}{k}\sum_{i=1}^{k}\omega_i \tag{4.9}$$

$$\omega_i = \frac{\frac{1}{|C_i|}\sum_{H_j \in C_i} d(M_i, H_j)}{\min_{j \neq i} d(M_i, M_j)} \tag{4.10}$$

where M_i is the mean histogram of cluster C_i, $d(\cdot)$ is the CIE94 distance, and k is the number of clusters. It measures the compactness of the clusters and the amount of overlaps between the clusters. The smaller the cluster spread, the more compact are the clusters and the less are the overlaps between them.

The cluster homogeneity Θ measures the proportion of histograms in a cluster that belong to the majority class of the cluster:

$$\Theta = \frac{1}{k}\sum_{i=1}^{k} P(L(C_i) \,|\, C_i) \tag{4.11}$$

where $L(C_i)$ denotes the majority class of cluster C_i and $P(L(C_i) \,|\, C_i)$ is the conditional probability of $L(C_i)$ given C_i. If the cluster homogeneity is less than $1/n$, then the cluster must contain histograms that belong to at least $n + 1$ classes. Therefore, the smaller the n, the more homogeneous is the cluster.

In figure 4.8, for all three cases, clustering performance improved significantly when the number of clusters k increased from 5 to 20. At $k > 20$, the cluster spreads of c + L_2 and c + JD/L_2 improved slightly with increasing k but their cluster

homogeneity decreased. Notice that performing cluster assignment with JD did not improve clustering performance significantly because the computation of mean histogram was based on L_2 instead of the more reliable JD.

In contrast, the cluster spread and homogeneity of a + WC stabilized at $k > 20$, and were better than those of c + L_2 and C + JD/L_2 for all k. In other words, a + WC produced more compact and more homogeneous clusters that were more widely spaced out than did c + L_2 and c + JD/L_2. Moreover, its performance is more stable than those of the other two cases. This result indicates that a + WC is more effective and reliable for practical applications in which the optimal number of clusters k is often unknown.

4.2 Gabor Feature

The Gabor texture features and weighted-mean-variance (WMV) as defined by Ma and Manjunath [31] have been shown to produce good texture discrimination, particularly for structured and oriented textures. Following sections give a brief description of Gabor texture features and WMV.

4.2.1 Gabor Functions and Wavelets

A two dimensional Gabor function $g(x, y)$ and its Fourier transform $G(u, v)$ can be written as:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right], \tag{4.12}$$

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\}, \tag{4.13}$$

where $\sigma_u = (2\pi\sigma_x)^{-1}$ and $\sigma_v = (pi\sigma_y)^{-1}$. Gabor functions form a complete but nonorthogonal basis set. Expanding a signal using this basis provides a localized frequency description.

A class of self-similar functions, referred to as *Gabor wavelets* in the following discussion, is now considered. Let $g(x,y)$ be the mother Gabor wavelet, then this self-similar filter dictionary can be obtained by appropriate dilation and rotations of $g(x,y)$ through the generating function:

$$g_{mn}(x,y) = a^{-m} g(x',y'), \tag{4.14}$$

where $a > 1$, m,n = integer and

$$x' = a^{-m}(x\cos\theta + y\sin\theta), \text{ and } y' = a^{-m}(-x\sin\theta + y\cos\theta),$$

$\theta = n\pi/K$ and K is the total number of orientations. The scale factor a^{-m} in 4.14 is meant to ensure that the energy is independent of m.

4.2.2 Gabor Filter Dictionary Design

The nonorthogonality of the Gabor wavelets implies that there is redundant information in the filtered images, and the following strategy is used to reduce this redundancy. Let U_l and U_h denote the lower and upper center frequencies of interest. Let K be the number of orientations and S be the number of scales in the multi-resolution decomposition. Then the design strategy is to ensure that the half-peak magnitude support of the

filter response in the frequency spectrum touch each other. This results in the following formulas for computing the filter parameters σ_u and σ_v (and thus σ_x and σ_y).

$$a = (U_h/U_l)^{\frac{1}{S-1}},\ \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}},$$
$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)\left[U_h - 2\ln\left(\frac{\sigma_u^2}{U_h}\right)\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}}, \tag{4.15}$$

where $W = U_h$ and $m = 0, 1, \ldots, S-1$. In order to eliminate sensitivity of the filter response to absolute intensity values, the real (even) components of the $2D$ Gabor filters are biased by adding a constant to make them zero mean (This can be done by setting $G(0,0)$ in Eq. 4.13 to zero).

4.2.3 Feature Representation

Given an image $I(x,y)$, its Gabor wavelet transform is then defined to be

$$W_{mn}(x,y) = \int I(x_1,y_1)g^*_{mn}(x-x_1, y-y_1)\,dx_1\,dx_2 \tag{4.16}$$

where $*$ indicates the complex conjugate. It is assumed that the local texture regions are spatially homogeneous, and the mean μ_{mn} and the standard deviation σ_{mn} of the magnitude of the transform coefficients are used to represent the region for classification and retrieval purposes:

$$\mu_{mn} = \iint |W_{mn}(xy)|\,dx\,dy, \text{ and } \sigma_{mn} = \sqrt{\iint (|W_{mn}(x,y)| - \mu_{mn})^2\,dx\,dy}. \tag{4.17}$$

A feature vector is now constructed using μ_{mn} and σ_{mn} as feature components. In the experiments, four scales $S = 5$ and six orientations $K = 6$, resulting in a feature vector

$$\mathbf{f} = [\mu_{00}\ \sigma_{00}\ \mu_{01}\ \sigma_{01} \ldots\ \mu_{30}\ \sigma_{30}]^T.$$

Distance Measure

Consider two image patterns i and j, and let $\mathbf{f}^{(i)}$ and $\mathbf{f}^{(j)}$ represent the corresponding feature vectors. Then the distance between the two patterns in the feature space is defined to be

$$d(i,j) = \sum_m \sum_n \left(\left| \frac{\mu_{mn}^{(i)} - \mu_{mn}^{(j)}}{\alpha(\mu_{mn})} \right| + \left| \frac{\sigma_{mn}^{(i)} - \sigma_{mn}^{(j)}}{\alpha(\sigma_{mn})} \right| \right), \tag{4.18}$$

where $\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ are the standard deviations of the respective features over the entire database, and are used to normalize the individual feature components. The distance measure is called weighted-mean-variance (WMV).

4.3 MRSAR Feature

Gabor features are good for structured textures. As for random textures which appear in natural images more often, we use the MRSAR model given in [33]. It is shown in [29] that MRSAR features are good for capturing the characteristics of random textures.

The MRSAR model is a second-order noncausal model described by five parameters at each resolution level. A symmetric MRSAR to applied to the L^* component of the $L^*u^*v^*$ image data. The pixel value $L^*(\mathbf{x})$ at a certain location $\mathbf{x}$ is assumed to linearly depend on the neighboring pixel values $L^*(\mathbf{y})$ and a zero-mean additive independent Gaussian noise term $\epsilon(\mathbf{x})$

$$L^*(\mathbf{x}) = \mu + \sum_{\mathbf{y} \in \nu} \theta(\mathbf{y}) L^*(\mathbf{y}) + \epsilon(\mathbf{x}) \ . \tag{4.19}$$

In equation 4.19, μ is the bias dependent on the mean value of L^*, ν is the set of

neighbors of pixel at location $\mathbf{x}$, and $\theta(\mathbf{y})$ with $\mathbf{y} \in \nu$ are the model parameters. The set of neighbors are defined for a window size of 5×5, 7×7 and 9×9. In [29, 42], it is shown that the MRSAR features computed with these window sizes provide the best overall retrieval performance over the entire Brodatz database.

The model is symmetric, i.e., $\theta(\mathbf{y}) = \theta(-\mathbf{y})$. Hence for a given neighborhood, four parameters representing 4 are estimated through least squares. Thus, the model parameters and the estimation error define a 5-dimensional feature vector. The procedure is repeated for the three chosen window sizes and the vectors are concatenated to form a 15-dimensional multi-resolution feature vector.

To extract the MRSAR feature given an image, a 21×21 overlapping window is moved over the image at steps of two pixels in both the horizontal and vertical directions. In each window, a multi-resolution feature vector is obtained. The mean vector $\mathbf{t}$ and the covariance matrix $\mathbf{S}$ over all windows inside a given image region are the MRSAR features associated with that image region.

The texture dissimilarity is then measured by the distance between two multivariate distributions with known mean vectors and covariance matrices. Given two image patterns i and j, Mahalanobis distance between the MRSAR feature vectors $\mathbf{t}_i$ and $\mathbf{t}_j$ are used to express this dissimilarity:

$$d(i,j) = \sqrt{(\mathbf{t}_i - \mathbf{t}_j)^T \mathbf{S}_j^{-1} (\mathbf{t}_i - \mathbf{t}_j)}, \tag{4.20}$$

where $\mathbf{S}_j$ represents the covariance matrix of $\mathbf{t}_j$.

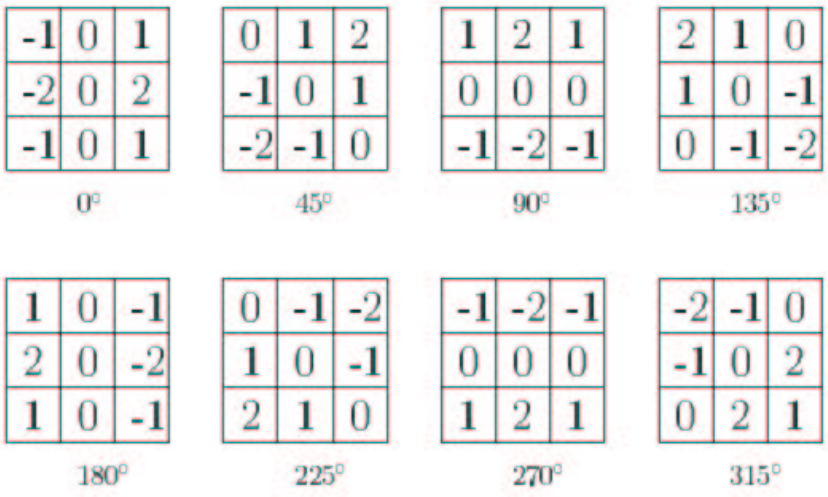


Figure 4.9: *Sobel operators and the corresponding gradient directions.*

4.4 Edge Direction and Magnitude Histogram

Normalized edge direction and magnitude histograms as given in [6, 49] are extracted from the images. First, the image region is transformed to the HSI (hue, saturation, intensity) space [18] so as to handle of the color information in a perceptually meaningful way. Each channel of the HSI representation is convolved with the eight Sobel operators shown in figure 4.9 [6] in order to determine the edges and their directions in the image region. For each pixel, the largest magnitude of the responses of the Sobel operators is used as the magnitude of the gradient and the quantized direction of the corresponding operator is the direction of the gradient.

The gradients at all pixels are thresholded to binary values by an appropriate threshold value for each channel. Finally the edge histogram is computed by summing up the edge pixels in each direction with corresponding quantized magnitude. The magnitude

is quantized to 8 levels, thus forming an 8×8 edge histogram. For edge histograms, the Euclidean distance defined in Eq. 4.5 is used as the dissimilarity measure [6, 49].

Chapter 5

Evaluation of the Probabilistic Labeling Algorithms

Extensive tests were performed to evaluate the following aspects of the semantic labeling method:

- Is the confidence value estimated by the method a reliable measure of classification accuracy?
- Can the method improve the confidence value by combining the most salient feature types?
- How accurate is the labeling method compared to the traditional approach?

Figure 5.1: *Sample images of the semantic classes used in the tests.*

5.1 Test Setup

A wide variety of 30 semantic classes (Fig. 5.1) were randomly identified by browsing the images in the Corel 50,000 photo collection. For each class, 250 image blocks of size 64×64 pixels were cropped from the images. Out of the 250 blocks, 200 were randomly selected for semantic class learning and the remaining 50 for region classification test. In total, 6,000 blocks were used for training and 1500 blocks for testing.

5.2 Feature-Based Region Clustering

Different nominal cluster separation values affect the clustering results. In the clustering stage for learning, two different nominal cluster separation values were used in region clustering, $s_t = 1.5\, r_t$ and $s_t = 2.0\, r_t$. Adjustment of r_t to r'_t is required when the number of unclustered data samples during learning is more than 10% of the total training data.

Table 5.1 shows the number of clusters generated and r'_t, the adjusted r_t (refer to Chapter 3 for detail). There are more color clusters than other clusters because there are more color variations than texture and edge variations in the images. Adjustment of r_t to r'_t is needed for color clusters only as more than 600 training data samples (10% of the training data) were unclustered before the adjustment. When $s = 1.5\ r_t$, more clusters were created and longer computing time was needed. For both nominal cluster separations, color clustering took the much longer time to run than cluster of other feature types because of the need to perform mean histograms compression.

Clustering performance was also measured by the maximum confidence $Q_M(\Psi(\tau, \kappa, n))$ of a cluster or cluster combination $\Psi(\tau, \kappa, n)$, where

$$Q_M(\Psi(\tau, \kappa, n)) = \underset{i}{Max}\ P(C_i|\Psi(\tau, \kappa, n)),\ for\ n\ =\ 1,\ 2.$$

Figure 5.2 show a plot of the average Q_M for clusters of single feature type and clusters of combinations of two feature types. In general, when $s = 1.5\ r_t$, more clusters have higher maximum confidence values, especially for combinations of two feature types. Though it takes much longer to do the clustering as there are more clusters, $s = 1.5\ r_t$ should still be used as time is not a critical issue in the learning stage.

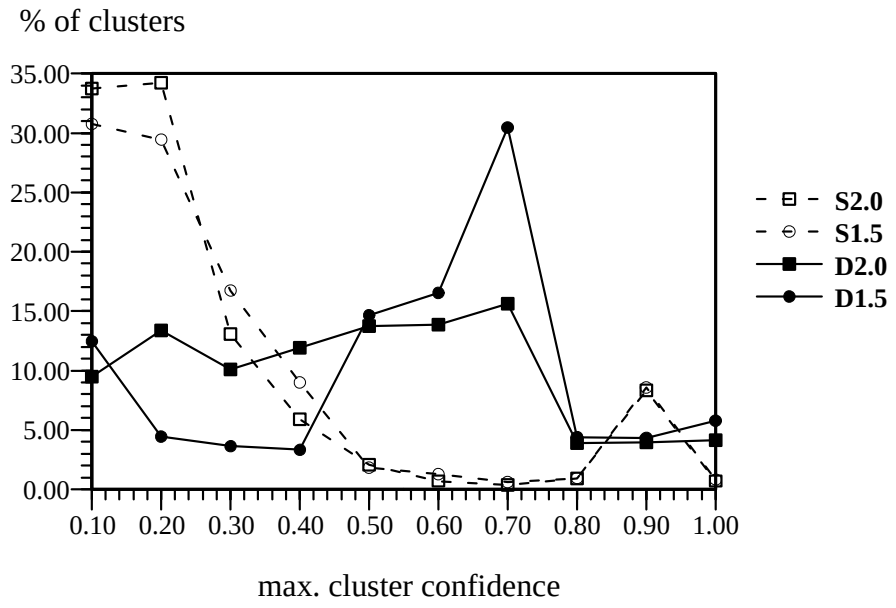


Figure 5.2: *Maximum cluster confidence. The maximum confidence is averaged over single clusters (***S***) and dual-cluster combinations (***D***) for nominal cluster separation* $s = 1.5\ r_t$ *and* $s_t = 2.0\ r_t$.

Table 5.1: *Information of Feature-Based Region Clustering with* $s_t = 1.5\ r_t$ *and* $2.0\ r_t$.

	Color	Gabor	MRSAR	Edge
num. of clusters	137	31	37	28
num. of unclustered data	1171	85	166	261
avg. increase of r_t (%)	24.85	N.A.	N.A.	N.A.
comp. time for clustering (second)	20144.28	164.50	182.70	910.40

(a) $s_t = 1.5\ r_t$

	Color	Gabor	MRSAR	Edge
num. of clusters	68	18	15	23
num. of unclustered data	1678	293	465	482
avg. increase of r_t (%)	33.33	N.A.	N.A.	N.A.
comp. time for clustering (second)	16320.15	139.60	153.67	770.60

(b) $s_t = 2.0\ r_t$

5.3 Salient Features

Salient features are features that are highly correlated with a semantic class. During semantic class learning, cluster combinations with high probability of associating with various semantic classes are identified. The feature values of the cluster centroids constitute the salient features of the classes.

Table 5.2 tabulates the confidence measures of a semantic class C_i averaged over all samples that belong to C_i, i.e.,

$$Q_i(\Psi(\tau, n)) = \frac{1}{|C_i|} \sum_{\kappa} |C_i \cap \Psi(\tau, \kappa, n)| \; P(C_i \,|\, \Psi(\tau, \kappa, n)) \,. \tag{5.1}$$

The average confidence gives an overall assessment of how strongly a feature type correlates with a semantic class. With a single feature, almost all semantic classes have very low confidence values. This confirms the expected results that single features are not enough to identify the semantic classes of image regions. An interesting surprise is that MRSAR, Gabor and edge histograms are highly correlated with clear sky. This is due to the fact that clear sky regions have almost no texture and no edge whereas all other image regions have some textures and edges. Therefore, the learning method can associate not only the *presence* but also the *absence* of features to semantic classes. Whatever the case may be, the learning method always chooses the one with the highest confidence.

Table 5.2 also shows that using a combination of only two feature types can already improve the mean confidence values of all the semantic classes significantly. The mean confidence value of all classes are above 0.5, and the overall average is 0.7. Increasing

Table 5.2: *Salient features. Columns 2–5 give the average confidence measures of the semantic classes using single features (Color (**C**), MRSAR (**M**), Gabor (**G**) and Edge (**E**)) . Numbers in bold are the highest average confidence among the four feature types. Column 6 lists the salient feature pairs (S.F.) and column 7 lists the corresponding improved average confidence (Conf.) using salient feature pairs.*

S/No.	Class	C	M	G	E	S.F.	Conf.
1	grass	**0.229**	0.096	0.058	0.081	color, Gabor	0.746
2	foliage	**0.321**	0.085	0.123	0.132	color, MRSAR	0.801
3	flower	**0.191**	0.097	0.098	0.156	color, edge	0.796
4	cloud	0.285	0.339	0.365	**0.381**	color, Gabor	0.741
5	clear sky	0.452	0.737	0.758	**0.847**	color, Gabor	0.860
6	rock	**0.082**	0.082	0.048	0.050	color, MRSAR	0.755
7	mountain	0.098	0.039	0.038	**0.120**	color, edge	0.697
8	sand	0.109	**0.144**	0.041	0.054	color, MRSAR	0.724
9	calm water	**0.139**	0.042	0.039	0.087	color, MRSAR	0.708
10	choppy sea	**0.159**	0.043	0.042	0.062	color, MRSAR	0.733
11	fur	0.082	**0.132**	0.034	0.048	color, MRSAR	0.740
12	face	**0.229**	0.099	0.115	0.101	color, edge	0.755
13	pebbles	0.094	0.108	**0.141**	0.055	MRSAR, edge	0.697
14	snow	0.202	0.053	0.045	0.078	color, MRSAR	0.442
15	roof tiles	0.094	0.078	**0.252**	0.079	color, Gabor	0.786
16	paved road	**0.116**	0.050	0.045	0.095	color, edge	0.715
17	unpaved road	**0.095**	0.059	0.046	0.066	color, edge	0.698
18	brick wall	0.088	0.059	**0.239**	0.181	color, Gabor	0.635
19	tree trunks	0.090	0.052	0.051	**0.157**	color, edge	0.736
20	wooden surface	**0.155**	0.102	0.046	0.103	color, MRSAR	0.768
21	window	0.077	0.052	0.048	**0.197**	color, edge	0.742
22	fence	0.089	0.054	0.046	**0.166**	color, edge	0.750
23	fire	**0.141**	0.090	0.076	0.068	color, MRSAR	0.768
24	firework	**0.131**	0.047	0.048	0.120	color, MRSAR	0.785
25	grass field	**0.236**	0.035	0.028	0.069	color, MRSAR	0.704
26	building	0.079	0.055	0.047	**0.154**	color, edge	0.765
27	house	0.072	0.051	0.042	**0.116**	color, edge	0.755
28	dirt ground	**0.150**	0.054	0.051	0.081	color, MRSAR	0.658
29	scale	0.141	0.104	**0.152**	0.057	color, Gabor	0.756
30	pillars	**0.143**	0.072	0.087	0.121	color, edge	0.697

the number of feature types to 3 or 4 does not produce higher confidence values. For our data set of 30 semantic classes, combinations of two feature types are enough.

It is interesting to see that a salient pair of features may not be individually salient. For example, for the grass images, the Gabor feature is less salient than MRSAR. Nevertheless, the combination of color and Gabor is the most salient pair for grass. Another interesting example is the class of pebbles. For this class, MRSAR and edge features constitute the salient pair, but individually both features are less salient than Gabor. Color is not found to be a salient feature because the pebble images in the training set contain large variation of colors.

The above results are consistent with those of Szummer and Picard's for indoor vs. outdoor classification of images [52]. They observed that a pair of features is more accurate for image classification than a single feature. Moreover, combining two weak features consistently produced more accurate classification than a single good feature.

5.4 Labeling Performance

In Sections 5.2 and 5.3, we have analyzed the results of learning stage and discussed the findings in the learning stage. In this section, we will evaluating the labeling performance of testing samples.

5.4.1 Confidence and Classification Accuracy

To test whether the confidence measure estimated by the method correlate with classification accuracy, a region classification test was performed on $1,500$ testing image regions, 50 per semantic class. The region classification method described in Chapter 3 was executed at various threshold values. Figure 5.3 plots the classification accuracy vs. the maximum confidence of a region R_i, i.e., $Q_M(R_j) = \underset{i}{Max} Q_i(R_j)$. To compute the classification accuracy, a recursive algorithm was applied to group the samples into groups containing samples with similar $Q_M(R_j)$.Test results in Figure 5.3 shows that above confidence value of 0.75, the accuracy is above 0.9. Figure 5.4 plots the data distribution at various confidence levels, about 77% of data have confidence values above 0.75. 0.75 was chosen as the threshold for classification.

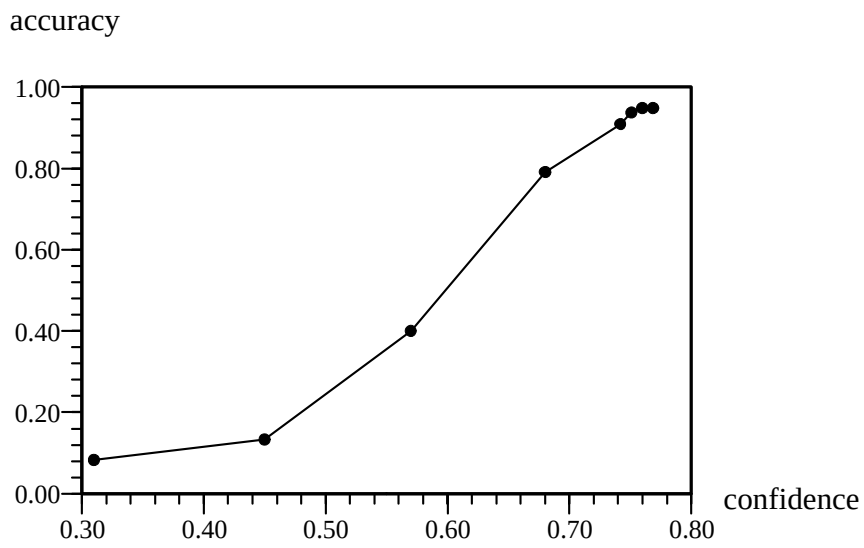


Figure 5.3: *Region classification accuracy at various confidence levels.*

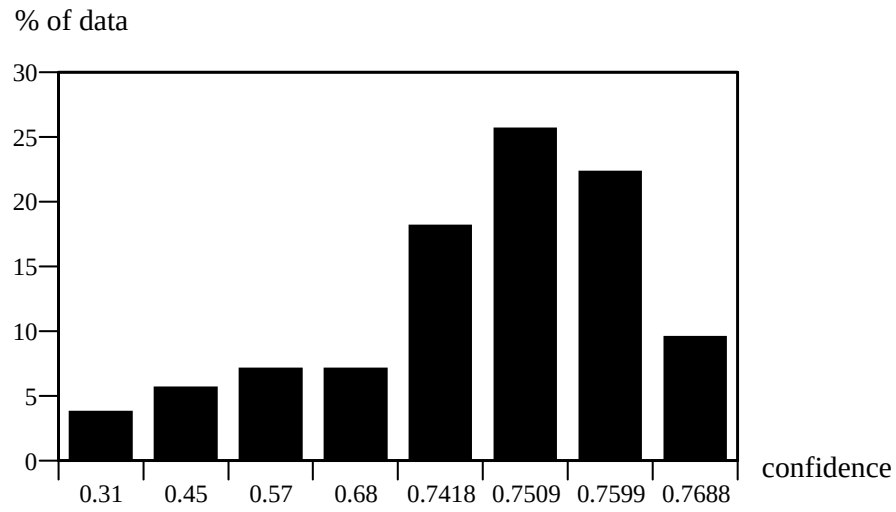


Figure 5.4: *Data distribution at various confidence levels.*

Regions labeled with semantic classes having low confidence values are ambiguous. In applications where image structures are used for image matching, such as [34, 37, 41], image structures provide additional information that can be used for disambiguation. So, regions labeled with "unknown class" should be regarded as ambiguous and should not be classified prematurely by the semantic labeling algorithm. If these regions are rejected and not classified due to low confidence in classification, then the classification accuracy of the remaining regions improves significantly from 0.70 to 0.91 (Fig. 5.5). The amount of rejection for the threshold value of 0.75 is 23%. The confusion matrix in Figure 5.7(a) further confirms the improved classification accuracy. From the above test results, we can conclude that the confidence values estimated by the semantic labeling algorithm are very reliable: a confidence value greater or equal to 0.75 translates to a

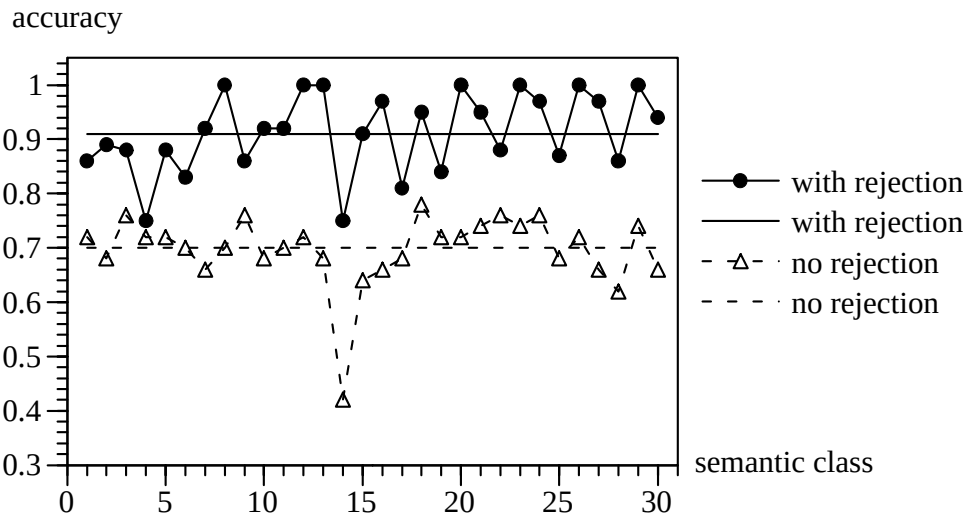


Figure 5.5: Region classification accuracy at confidence of 0.75 (solid lines: with rejection, dotted lines: without rejection, horizontal lines: mean accuracy

mean classification accuracy of 91% or higher. For regions with low confidence, multiple labels are assigned to them together with the corresponding confidence values. These information are much more valuable than single class labels for higher-level modules such as image retrieval and image classification. Those modules can regard regions with high confidence to be correctly classified into the respective semantic classes indicated by their labels. On the other hand, the multiple labels and confidence measures of regions with low confidence can allow the modules to disambiguate between various possibilities using image structures, e.g., by applying fuzzy conceptual graph matching [34, 37] or attributed relational graph matching [41]. This kind of disambiguation would be impossible without the multiple labels and confidence measures.

5.4.2 Performance Comparison with Support Vector Machine

To compare the performance of our labeling method with traditional approach, support vector machine (SVM) was used for the region classification problem.

SVM Training

The SVM was downloaded from the web site of SvmFu3.0 (`http://five-percent-nation.mit.edu/SvmFu/index.html`). Gaussian kernel was chosen and SVM was trained with different Gaussian standard deviation σ's. The input feature vector to SVM is the concatenated feature vector of fixed-binning histogram (Chapter 4), Gabor features, MRSAR features and edge histogram. The input feature vector has 274 dimensions, 165 for color histogram, 30 for Gabor feature (without standard deviation), 15 for MRSAR feature and 64 for edge histogram. The reason for not using adaptive color histograms is that SVM requires input data with a fixed number of dimensions [9, 12].

For a successful SVM training, normalization of the input data is necessary as they are of different scales. It is difficult to perform the normalization across different feature types. In the test, principal component analysis (PCA) was used to determine the normalization factor. The data were normalized so that the largest eigenvalues for each feature type are the same. This normalization avoided accidental biasing of SVM training towards feature types with large feature values.

It is a usual practice to apply PCA to reduce the dimensionality of the input feature vectors. However, we found that SVM failed to find satisfactory decision planes for

Table 5.3: *SVM training with input data of reduced dimensionality. The criterion of reducing dimensionality is to choose the eigenvectors that account for a certain percentage of the total eigenvalues.*

% of Eigenvalue kept	Num. of Classes Separated
90%	12
95%	18
98%	20
100%	30

the dimensionality-reduced training data (Table 5.3). The failure can be attributed to incorrect elimination of important dimensions by PCA, which makes the data inseparable in the space of reduced dimensionality.

SVM Testing

In the training for SVM, different standard deviation σ values were used for the Gaussian kernel. To select the best σ for the labeling task, we started with $1/4\lambda_{max}$ where λ_{max} is the biggest eigenvalue. The same 1500 regions described in Section 5.1 were used for the testing. Figure 5.4.2 shows that different sigma values can greatly affect the classification accuracy. At $\sigma = 17.5$ for performance comparison as it gave the highest classification accuracy of 61.6%.

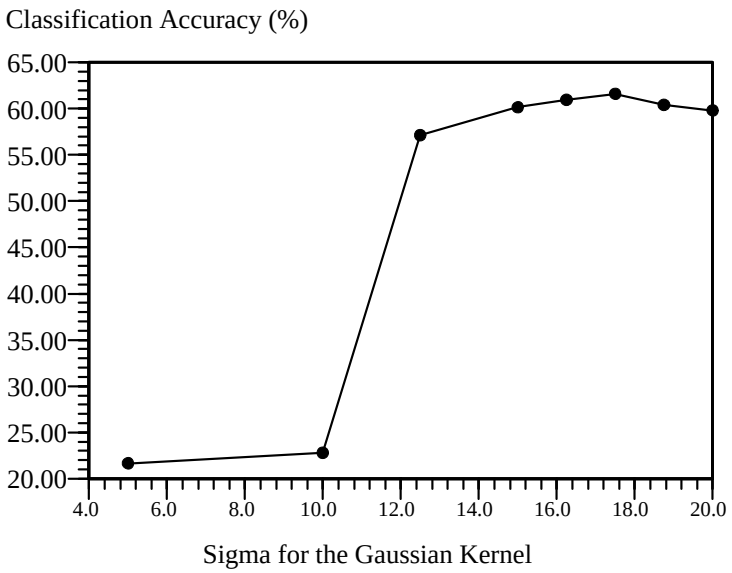


Figure 5.6: *Classification accuracy vs. different sigma (σ) values.*

Performance Comparison

For probabilistic labeling method without rejection of low confidence samples, a classification accuracy of 70% was achieved (Table 5.4). With rejection, the accuracy increased to 90%. The usual SVM implementation does not perform rejection, and is has the lowest classification accuracy of 61.6%. In principle, it is possible to include rejection for SVM, but this would require the SVM to measure some form of confidence.

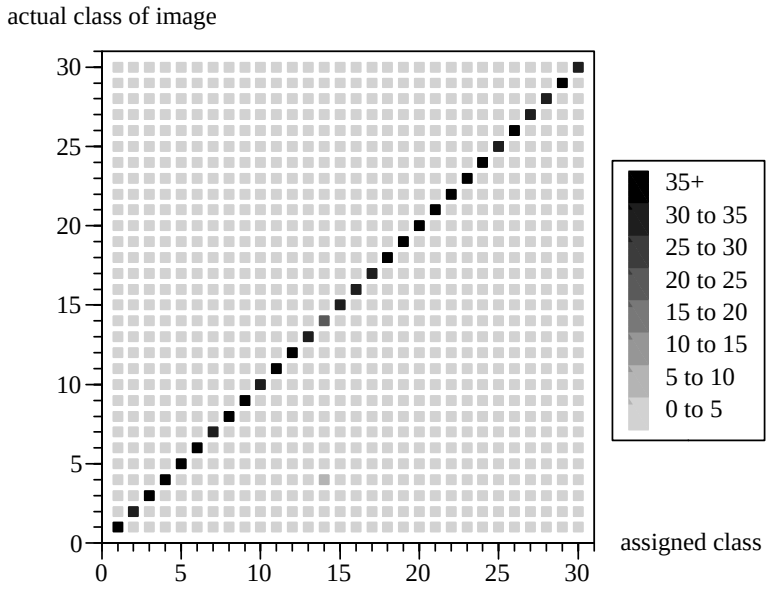
Figure 5.7 shows the comparison of classification accuracy in the form of confusion matrix. In the confusion matrix for SVM (Fig. 5.7(a)), there are a few points that are not on the diagonal line, which means some data are classified into the wrong class. In the confusion matrix for the probabilistic labeling method (with rejection), almost all the

Table 5.4: *Classification Accuracy Comparison (SVM vs. probabilistic labeling).*

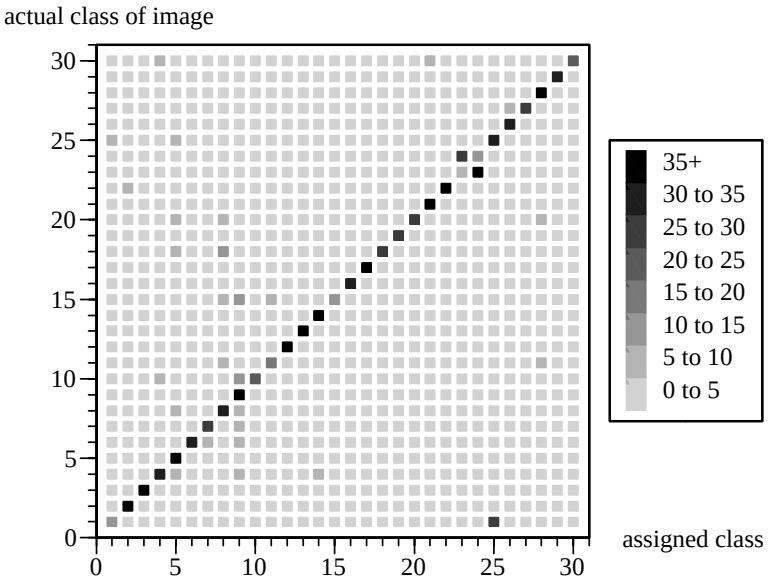
SVM Labeling	Probabilistic Labeling	
	with rejection	without rejection
61.6%	70%	91%

points are along the diagonal line, which means that almost all the data can be classified correctly.

For probabilistic labeling method, different salient feature types are used for different semantic classes. This result has also been confirmed in [55]. However, for SVM labeling method, all four feature types are used and are assumed to be equally important. Assigning weights to different feature types is not a practical solution because different weights will be required to classify the samples into different classes.



(a)



(b)

Figure 5.7: *Confusion matrix of region classification. (a) Probabilistic labeling system with rejection, (b) SVM without rejection.*

Chapter 6

Conclusion and Future Work

This chapter discusses the contributions of the existing work and presents suggestions for possible further development and application of the current work.

6.1 Conclusion

This thesis presented a probabilistic approach for semantic labeling of image regions which opens real possibilities for higher level image categorization and retrieval.

In Chapter 1, we presented the realization process of this approach in a system overview (Figure 1.1). We now go through this process step by step again and summarize our contribution in each step:

Feature Extraction

In Chapter 4 we described various features used in the systems. We focused on adaptive color histograms especially. For color, we used CIELAB space, which was

designed so that distances between single colors conform to perceptual similarity. We presented a dissimilarity measure called *weighted correlation* for comparing two adaptive color histograms. We provided an extensive comparison of various dissimilarity measure that are used for comparing color histograms. It is shown that adaptive color histograms and weighted correlation achieves the best performance in general.

Feature-Based Region Clustering

An adaptive k-means clustering algorithm is used for feature-based region clustering. For clustering of adaptive color histograms, the algorithm used a well-founded definition of mean histogram [27] for the calculation of cluster centroid.

Probability Estimation

A probability estimation algorithm is implemented to combine multiple features together instead of combining them linearly. Through this approach, salient features, features that have high correlation with a semantic class, are selected automatically. In our experiment, we found that not all feature types, but only a subset of all feature types are salient features. Only salient features are used for probabilistic labeling.

Probabilistic Labels

In the probabilistic labeling, an image region is associated with a set of labels with corresponding confidence values. It is found in our classification tests that the confidence values have a high correlation with the classification accuracy. In

particular, it is found that regions with confidence value of greater or equal to 0.65 can be classified into the correct semantic classes with an average accuracy of 90%. For regions with lower confidence values, the multiple semantic labels and the corresponding confidence values allow a higher-level algorithm, such as fuzzy conceptual graph matching and attributed relational graph matching, to disambiguate them using information about image structures. In summary, the semantic labeling method is expected to contributes significantly to bridge up the gap between low-level features and high-level semantics for image categorization and retrieval.

6.2 Future Work

The weighted correlation can be applied to other modalities besides color as long as a ground distance can be defined in the appropriate feature space. Examples include texture (Some work has been done in [28]), shape, compositions of objects, eigenimage similarity, etc. A large ensemble of features from different modalities can improve the overall performance of image categorization and retrieval.

The fuzzy labeling system is a framework for general labeling problem. It can be applied for specialized problem like face detection, 3-D object labeling, by using special features other than color, texture and edge.

For two regions with low confidence values but similar distribution, an interesting question to ask is "what can we say about them? " Can we say they are similar because

they have similar confidence value distribution? Or we cannot conclude anything since the confidence values are low? This is an interesting problem for future research.

Bibliography

[1] IEC 61966-2.1. *Default RGB Colour Space - sRGB*. International Electrotechnical Commission, Geneva, Switzerland, 1999. see also www.srgb.com.

[2] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Recognition of images in large databases using a learning framework. Technical Report 97-939, Computer Science Division, University of California at Berkeley, 1997.

[3] R. S. Berns. *Billmeyer and Saltzman's Principles of Color Technology*. John Wiley & Sons, 3rd edition, 2000.

[4] E. Binaghi, I. Gagliardi, and R. Schettini. Image retrieval using fuzzy evaluation of color similarity. *Int. J. PR and AI*, 8:945–968, 1994.

[5] Léon Bottou and Yoshua Bengio. Convergence properties of the kmeans algorithm. In *Advances in Neural Information Processing Systems*, volume 7, Denver, 1995. MIT Press.

[6] Sami Brandt. Use of shape features in content-based image retrieval. Master's thesis, Helsinki University of Technology, Finland, 1999.

[7] J. Burbea and C. R. Rao. Entropy differential metric, distance and divergence measures in probability spaces: A unified approach. *J. Multivariate Analysis*, 12:575–596, 1982.

[8] J. Burbea and C. R. Rao. On the convexity of some divergence measures based on entropy functions. *IEEE Trans. Information Theory*, 28(3):489–495, 1982.

[9] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[10] N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko. Interpreting image databases by region classification. *Pattern Recognition*, 30(4):555–563, 1997.

[11] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Region-based image querying. In *Proc. CVPR Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[12] O. Chapelle, P. Haffner, and V.N.Vapnik. Support vector machines for histogram based image classification. In *IEEE transactions on Neural Networks*, volume 10, 1999.

[13] G. Ciocca and R. Schettini. A relevance feedback mechanism for content-based image retrieval. *Infor. Proc. and Management*, 35:605–632, 1999.

[14] I. J. Cox, M. L. Miller, S. O. Omohundro, and P. N. Yianilos. PicHunter: Bayesian relevance feedback for image retrieval. In *Proc. ICPR '96*, pages 361–369, 1996.

[15] Y. Deng, B.S. Manjunath, C. Kenny, M.S. Moore, and H. Shin. An efficient color representation for image retrieval. *IEEE Trans. on Image Processing*, 10(1):140–147, 2001.

[16] C. Y. Fung and K. F. Loe. Learning primitive and scene semantics of images for classification and retrieval. In *Proc. ACM Multimedia*, pages II: 9–12, 1999.

[17] Y. Gong, G. Proietti, and C. Faloutsos. Image indexing and retrieval based on human perceptual color clustering. In *Proc. CVPR '98*, 1998.

[18] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison–Wesley Publishing Company, 1993.

[19] G.P.Babu, B.M. Mehtre, and M.S. Kankanhalli. Color indexing for efficient image retrieval. *Multimedia Tools Applicat.*, 1:327–348, 1995.

[20] S.-S. Guan and M. R. Luo. Investigation of parametric effects using small colour differences. *Color Research and Application*, 24(5):331–343, 1999.

[21] A. Gupta and R. Jain. Visual information retrieval. *Comm. of the ACM*, 40(5), 1997.

[22] J. Hafner, H. S. Sawhney, W. Esquitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. PAMI*, 17:729–736, 1995.

[23] H. Jeffreys. *Theory of Probability*. Oxford, 2nd edition, 1948.

[24] M.S. Kankanhalli, B.M. Methre, and J.K. Wu. Cluster-based color matching for image retrieval. *Pattern Recognition*, 29:701–708, 1996.

[25] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[26] W. K. Leow and R. Li. Adaptive binning and dissimilarity measure for image retrieval and classification. In *Proc. IEEE CVPR*, 2001.

[27] W. K. Leow and R. Li. The analysis and applications of adaptive-binning color histograms. *Computer Vision and Image Understanding, special issue on Colour for Image Indexing and Retrieval (submitted)*, 2003.

[28] F. S. Lim and W. K. Leow. Adaptive histograms and dissimilarity measure for texture retrieval and classification. In *Proc. Int. Conf. on Image Processing*, 2002.

[29] F. Liu and R. W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. PAMI*, 18(7):722–733, 1996.

[30] W. Y. Ma and B. S. Manjunath. NeTra: A toolbox for nagivating large image databases. In *Proc. ICIP*, pages 568–571, 1997.

[31] B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. PAMI*, 8(18):837–842, 1996.

[32] W. Y. Masand B. S. Manjunath. Texture features and learning similarity. In *Proc. IEEE CVPR '96*, pages 425–430, 1996.

[33] J. C. Mao and A. K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25:173–188, 1992.

[34] S. Medasani and R. Krishnapuram. A fuzzy approach to content-based image retrieval. In *Proc. IEEE Conf. on Fuzzy Systems*, pages 1251–1257, 1999.

[35] B. M. Mehtre, M. S. Kankanhalli, A. Desai, and G. C. Man. Color matching for image retrieval. *Pattern Recognition Letters*, 16:325–331, 1995.

[36] M. Melgosa. Testing CIELAB-based color-difference formulas. *Color Research and Application*, 25(1):49–55, 2000.

[37] P. Mulhem, W. K. Leow, and Y. K. Lee. Fuzzy conceptual graph for matching images of natural scenes. In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 1397–1402, 2001.

[38] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content using color, texture, and shape. In *Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–181, 1993.

[39] S. J. Park, D. K. Park, and C. S. Won. Core experiments on MPEG-7 edge histogram descriptor. *MPEG document M5984*, 2000.

[40] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. *Int. J. Computer Vision*, 18(3):233–254, 1996.

[41] G. M. Petrakis and C. Faloutsos. Similarity searching in large image databases. *IEEE Trans. on Knowledge and Data Engineering*, 9(3):435–447, 1997.

[42] R. W. Picard, T. Kabir, and F. Liu. Real-time recognition with the entire brodatz texture database. In *Proc. IEEE CVPR*, 1993.

[43] J. Puzicha, J. M. Buhmann, Y. Rubner, and C. Tomasi. Empirical evaluation of dissimilarity for color and texture. In *Proc. ICCV '99*, pages 1165–1172, 1999.

[44] Y. Rubner. *Perceptual Metrics for Image Database Navigation.* PhD thesis, Computer Science Dept., Stanford U., 1999.

[45] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proc. ICCV '98*, pages 59–66, 1998.

[46] Y. Rui and T. Huang. Optimizing learning image retrieval. In *Proc. IEEE CVPR*, 2000.

[47] S. Sclaroff, L. Taycher, and M. La Cascia. Image-Rover: A content-based image browser for the world wide web. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[48] 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 Trans. On PAMI*, 22(12):1349–1379, 2000.

[49] J. R. Smith, S. Basu, G. lyengar, C.-Y. Lin, M. Naphade, B. Tseng, S. Srinivasan, A. Amir, and D. ponceleon. Integrating features, models, and semantics for trec video retrieval. In *Proc. of The Tenth TREC*, 2001.

[50] J. R. Smith and S.-F. Chang. Single color extraction and image query. In *Proc. ICIP*, 1995.

[51] T. Song and R. Luo. Testing color-difference formulae on complex images using a CRT monitor. In *Proc. of 8th Color Imaging Conference*, 2000.

[52] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *Proc. ICCV Workshop on Content-based Access of Image and Video Databases*, pages 42–51, 1998.

[53] I. J. Taneja. New developments in generalized information measures. In P. W. Hawkes, editor, *Advances in Imaging and Electron Physics*, volume 91. Academic Press, 1995.

[54] C. Town and D. Sinclair. Content based image retrieval using semantic visual categories. Technical Report 2000.14, AT&T Laboratories Cambridge, 2000.

[55] A. Vailaya, A. Jain, and H. J. Zhang. On image classification: City images vs. landscapes. *Pattern Recognition*, 31:1921–1935, 1998.

[56] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics sensitive integrated matching for picture libraries. *IEEE Trans. on PAMI*, 23(9):947–963, 2001.