Image Classification for Content-Based Indexing

Aditya Vailaya, *Associate Member, IEEE*, Mário A. T. Figueiredo, *Member, IEEE*, Anil K. Jain, *Fellow, IEEE*, and Hong-Jiang Zhang, *Senior Member, IEEE*

***Abstract*—Grouping images into (semantically) meaningful categories using low-level visual features is a challenging and important problem in content-based image retrieval. Using binary Bayesian classifiers, we attempt to capture high-level concepts from low-level image features under the constraint that the test image does belong to one of the classes. Specifically, we consider the hierarchical classification of vacation images; at the highest level, images are classified as indoor or outdoor; outdoor images are further classified as city or landscape; finally, a subset of landscape images is classified into sunset, forest, and mountain classes. We demonstrate that a small vector quantizer (whose optimal size is selected using a modified MDL criterion) can be used to model the class-conditional densities of the features, required by the Bayesian methodology. The classifiers have been designed and evaluated on a database of 6931 vacation photographs. Our system achieved a classification accuracy of 90.5% for indoor/outdoor, 95.3% for city/landscape, 96.6% for sunset/forest & mountain, and 96% for forest/mountain classification problems. We further develop a learning method to incrementally train the classifiers as additional data become available. We also show preliminary results for feature reduction using clustering techniques. Our goal is to combine multiple two-class classifiers into a single hierarchical classifier.**

***Index Terms*—Bayesian methods, content-based retrieval, digital libraries, image content analysis, minimum description length, semantic indexing, vector quantization.**

I. Introduction

CONTENT-BASED image retrieval has emerged as an important area in computer vision and multimedia computing. Many organizations have large image and video collections (programs, news segments, games, art) in digital format, available for on-line access. Organizing these libraries into categories and providing effective indexing is imperative for "real-time" browsing and retrieval. With the development of digital photography, more and more people are able to store vacation and personal photographs on their computers. As an example, travel agencies are interested in digital archives of photographs of holiday resorts; a user could query these databases to plan a vacation. However, in order to make these databases more useful, we need to develop schemes for indexing and categorizing the humungous data.

Several content-based image retrieval systems have been recently proposed: QBIC [5], Photobook [26], SWIM [44], Virage [10], Visualseek [36], Netra [17], and MARS [20]. These systems follow the paradigm of representing images using a set of attributes, such as color, texture, shape, and layout, which are archived along with the images. Retrieval is performed by matching the features of a query image with those in the database. Users typically do not think in terms of low-level features, i.e., user queries are typically semantic (e.g., "show me a sunset image") and not low-level (e.g., "show me a predominantly red and orange image"). As a result, most of these image retrieval systems have poor performance for (semantically) specific queries. For example, Fig. 1(b) shows the top-ten retrieved images (based on color histogram features) from a database of 2145 images of city and landscape scenes, for the query in Fig. 1(a). While the query image has a monument, some of the retrieved images have mountain and coast scenes. Recent research in human perception of image content [21], [24], [27], [31] suggests the importance of semantic cues for efficient retrieval. One method to decode human perception is through the use of relevance feedback mechanisms [33]. A second method relies on grouping the images into semantically meaningful classes [42]. Fig. 1(c) shows the top-ten results (again based on color histograms) on a database of 760 *city* images for the same query; clearly, filtering out landscape images improves the retrieval result.

As shown in Fig. 1(a)–(c), a successful indexing/categorization of images greatly enhances the performance of content-based retrieval systems by filtering out irrelevant classes. This rather difficult problem has not been adequately addressed in current image database systems. The main problem is that only low-level features (as opposed to higher level features such as objects and their inter-relationships) can be reliably extracted from images. For example, color histograms are easily extracted from color images, but the presence of sky, trees, buildings, people, etc., cannot be reliably detected. The main challenge, thereby, lies in grouping images into semantically meaningful categories based on low-level visual features. One attempt to solve this problem is the hierarchical indexing scheme proposed in [45], [46], which performs clustering based on color and texture, using a *self-organizing map*. This indexing scheme was further applied in [16] to create a texture thesaurus for indexing a database of aerial photographs. However, the success of such clustering-based schemes is often limited, largely due to the low-level feature-based representation of image content. For example, Fig. 2(a)–(d) shows two images and their corresponding edge direction coherence feature vectors (see [42]). Although,

Manuscript received February 8, 1999; revised August 11, 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tsuhan Chen.

A. Vailaya is with Agilent Technologies, Palo Alto, CA 94303-0867 USA (e-mail: aditya_vailaya@agilent.com).

M. Figueiredo is with the Instituto de Telecomunicações and Instituto Superior Técnico, 1049-001 Lisboa, Portugal (e-mail: mtf@lx.it.pt).

A. K. Jain is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: jain@cse.msu.edu).

H.-J. Zhang is with Microsoft Research China, Beijing 100 080, China (e-mail: hjzhang@microsoft.com).

Publisher Item Identifier S 1057-7149(01)00098-7.

1057–7149/01$10.00 © 2001 IEEE

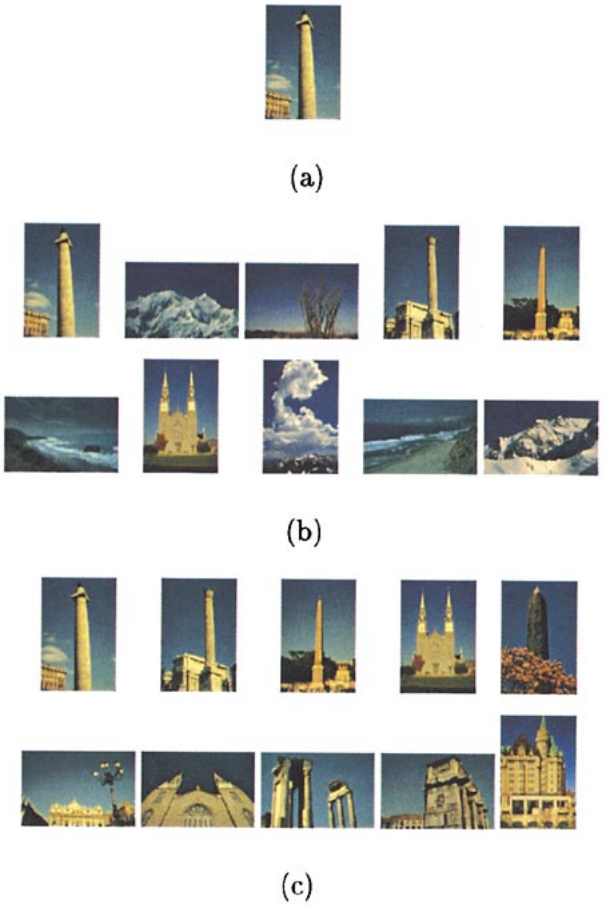


Fig. 1. Color-based retrieval. (a) Query image, (b) top-ten retrieved images from 2145 city and landscape images, and (c) top-ten retrieved images from 760 city images; filtering out landscape images prior to querying clearly improves the retrieval results.

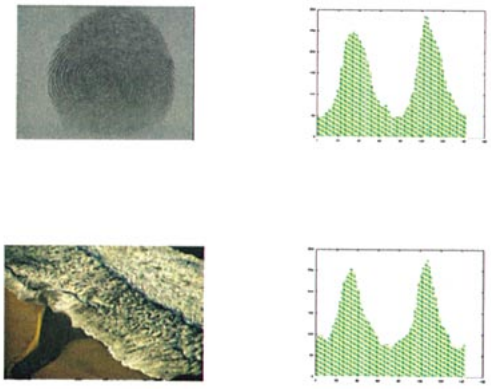


Fig. 2. Edge direction coherence vector features for (a) fingerprint and (c) landscape image.

these are semantically very different concepts, their edge direction histograms are highly similar, illustrating the limitations of this low-level feature in capturing semantic content. Yet, we shall show that these same features are sufficiently discriminative for city/landscape classification. That is, specific low-level

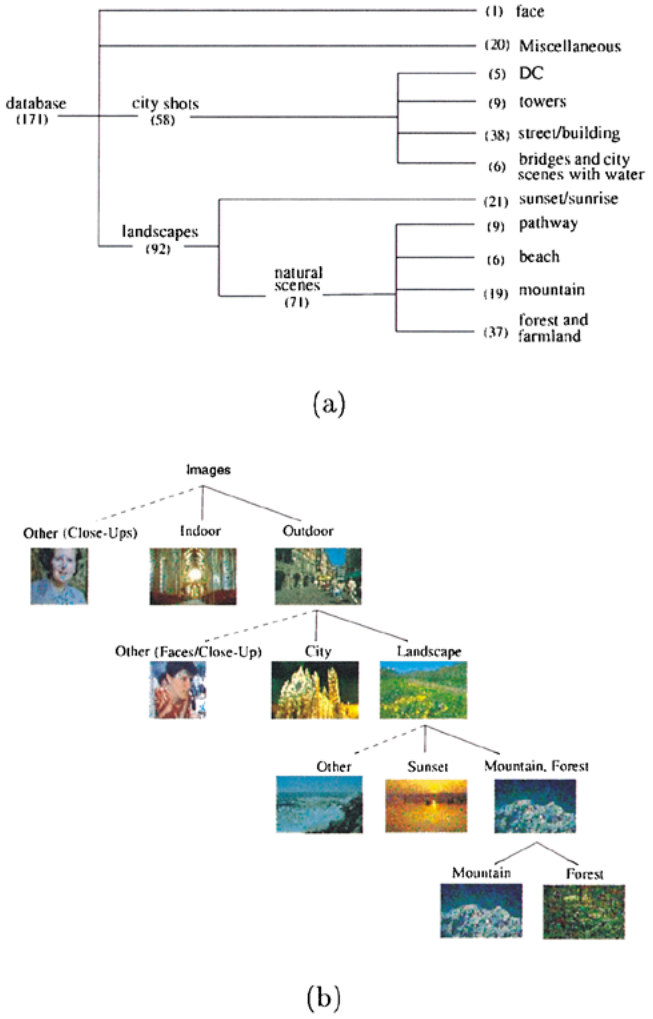


Fig. 3. (a) Hierarchy of the 11 categories obtained from human provided grouping [42] and (b) simplified semantic classification of images; solid lines show the classification problems addressed in this paper.

features can be used in constrained environments to discriminate between certain conceptual image classes. To achieve automatic categorization/indexing in a large database, we need to develop robust schemes to identify salient image features capturing a certain aspect of the semantic content. This necessitates an initial specification of meaningful classes, so that the database images can be organized in a *supervised* fashion.

In this paper, we address the problem of image classification from low-level features. Specifically, we classify vacation photographs into a hierarchy of high-level classes. Photographs are first classified as *indoor* or *outdoor*. Outdoor images are then classified as *city* or *landscape*. A subset of landscape images is further classified into *sunset*, *forest*, and *mountain* classes. The above hierarchy was identified based on experiments with human subjects on a small database of 171 images [42] (as briefly described in Section II). These classification problems are addressed using Bayesian theory. The required class-conditional probability density functions are estimated, during a training phase, using *vector quantization* (VQ) [9]. An MDL-type principle [30] is used to determine the optimal

codebook size from the training samples. Advantages of the Bayesian approach include

1) small number of codebook vectors represent each class, thus greatly reducing the number of comparisons necessary for each classification;
2) it naturally allows for the integration of multiple features through the class-conditional densities;
3) in addition to a classification rule, we have degrees of confidence which may be used to incorporate a reject option into the classifiers.

The paper is organized as follows. Section II briefly mentions psychophysical studies which are the basis of our work in identifying the global scene represented in an image. We also describe our experiments with human subjects to identify conceptual classes in a database of vacation images. After reviewing the Bayesian framework for image classification in Section III, Section IV addresses VQ-based density estimation and the MDL principle for selecting codebook sizes. Section V discusses implementation issues. We report the classification accuracies in Section VI. Sections VII and VIII discuss approaches for using incremental learning and automatic feature selection. Finally, Section IX concludes the paper and presents directions for future research.

II. High-Level Classes Identified by Humans

Psychophysical and psychological studies have shown that scene identification by humans can proceed, in certain cases, without any kind of object identification [1], [2], [34]. Biederman [1], [2] suggested that an arrangement of volumetric primitives (geons), each representing a prominent object in the scene, may allow rapid scene identification independently of local object identification. Schyns and Oliva [34] demonstrated that scenes can be identified from low spatial-frequency images that preserve the spatial relations between large-scale structures in the scene, but which lack the visual detail to identify local objects. These results suggest the possibility of coarse scene identification from global low-level features before the identity of objects is established. Based on these observations, we address the problem of scene identification as the first step toward building semantic indices into image databases.

The first step toward building a classifier is to identify meaningful image categories which can be automatically identified by simple and efficient pattern recognition techniques. For this purpose, we conducted a simple small-scale experiment in which eight human subjects classified 171 vacation images [42]. Our goal was to identify a hierarchy of classes into which the vacation images can be organized. Since these classes match human perception, they allow organizing the database for effective browsing and retrieval.

Our experiments revealed a total of 11 semantic categories: forests and farmlands, mountains, beach scenes, pathways, sunset/sunrise images, long distance city shots, streets/buildings, monuments/towers, shots of Washington, DC, miscellaneous images, and faces. We organized these 11 categories into the hierarchy shown in Fig. 3(a). The first four classes (forests, mountains, beach scenes, and pathways) are grouped into the class *natural scenes*. Natural scenes and sunset images were further grouped into the *landscape* class. City shots, monuments, and shots of Washington DC were grouped into the *city* class. Finally, the miscellaneous, face, landscape, and city classes were grouped into the top-level class of *vacation* scenes. We conducted additional experiments to verify that the above hierarchy is reasonable: we used a multidimensional scaling algorithm to generate a three-dimensional (3-D) feature space to embed the 171 images from the 171×171 dissimilarity matrix used above (generated from user groupings). We then applied a K-means clustering algorithm to partition the (3-D) data. Our goal was to verify if the main clusters in this representation space agreed with the hierarchy shown in Fig. 3(a). For $K = 2$, we obtained two clusters of 62 and 109 images, respectively. The first cluster consisted of predominantly city images, while the second cluster contained landscape images. The following clusters were obtained with $K = 4$

1) city scenes (70 images);
2) sunrise/sunset images (21 images);
3) forest and farmland scenes and pathways (49 images);
4) mountain and coast scenes (31 images).

These groupings motivated us to study a hierarchical classification of vacation images.

In order to make the problem more tractable, we simplified the classification hierarchy as shown in Fig. 3(b). The solid lines show the classification problems addressed in this paper. This hierarchy is not complete, e.g., a user may be interested in images captured in the evening or images containing faces. However, it is a reasonable approach to simplify the image retrieval problem.

Another limitation of the proposed hierarchy is that the leaf nodes are not mutually exclusive. For example, an image can belong to both the city and sunset categories. One way to address this issue is to develop individual classifiers such as city/non-city or sunset/non-sunset, instead of a hierarchy. However, this would drastically increase the complexity of the classification task (now we will have to identify city scenes from all possible scenes, rather than differentiate between city and landscape scenes).

Most images can be classified as representing indoor or outdoor scenes. Exceptions include close-ups and pictures of a window or door. Outdoor images can be further divided into city or landscape [40], [42]. City scenes can be characterized by the presence of man-made objects and structures such as buildings, cars, roads. Natural scenes, on the other hand, lack these structures. A subset of landscape images can be further classified into one of the sunset, forest, and mountain classes. Sunset scenes are characterized by saturated colors (red, orange, or yellow), forest scenes have predominantly green color distribution, and mountain scenes can be characterized by long distance shots of mountains (either snow covered, or barren plateaus).

We assume that the input images do belong to one of the classes under consideration. This restriction is imposed because automatically rejecting images that do not belong to any of the classes, based on low-level image features alone, is in itself a

very difficult problem (see Fig. 2). However, for images belonging to the classes of interest, the Bayesian methodology can be used to reject ambiguous images based on the confidence values associated with the images (images that belong to both the classes of interest, such as an image of a city scene at sunset). We briefly discuss incorporating the reject option in Section VI-F.

III. Bayesian Framework

Bayesian methods have been successfully adopted in many image analysis and computer vision problems. However, its use in content-based retrieval from image databases is just being realized [43].

We now review the Bayesian framework for image classification. The set of possible images is partitioned into K classes $\Omega = \{\omega_1, \cdots, \omega_K\}$; any image belongs to one and only one class. The images from class ω_n are modeled as samples of a random variable, $\mathbf{x}$, whose class-conditional probability density function is $f(\mathbf{x}|\omega_n)$. Each class has an *a priori* probability, $\{p(\omega_1), p(\omega_2), \cdots, p(\omega_K)\}$, with $\sum_{i=1}^{K} p(\omega_i) = 1$. A loss function, $\mathcal{L}(\omega, \hat{\omega})$: $\Omega \times \Omega \rightarrow \mathcal{R}$, specifies the loss incurred when class $\hat{\omega}$ is chosen and the true class is ω. As is common in classification problems, we adopt the "0/1" loss function: $\mathcal{L}(\omega, \omega) = 0$, and $\mathcal{L}(\omega, \hat{\omega}) = 1$, if $\omega \neq \hat{\omega}$.

In most image classification problems, the decision is based on, say m, feature sets, $\mathbf{y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \cdots, \mathbf{y}^{(m)}\}$, rather than directly on the raw pixel values. Of course, $\mathbf{y}$ is a function of the image $\mathbf{x}$. We will then have class-conditional densities for the features, rather than for the raw images. It is often assumed that the feature sets are class-conditionally independent, that is

$$f(\mathbf{y}|\omega) = \prod_{i=1}^{m} f\left(\mathbf{y}^{(i)}|\omega\right), \qquad \text{for } \omega \in \Omega. \tag{1}$$

The classification problem can be stated as: "given the feature sets $\mathbf{y}$, classify the image into one of the classes in Ω."

The decision rule resulting from the "0/1" loss function is the *maximum a posteriori* (MAP) criterion [4], [29],

$$\hat{\omega} = \arg\max_{\omega \in \Omega} \{p(\omega|\mathbf{y})\} = \arg\max_{\omega \in \Omega} \{f(\mathbf{y}|\omega)p(\omega)\}. \tag{2}$$

In addition to the MAP classification, we also have a degree of confidence which is proportional to $p(\hat{\omega}|\mathbf{y})$.

IV. Density Estimation by Vector Quantization

The performance of a Bayes classifier depends critically on the ability of the features to discriminate among the various classes. Moreover, since the class-conditional densities have to be estimated from data, the accuracy of these estimates is also critical. Choosing the right set of features is a difficult problem to which we return in Section V-A. In this section, we focus on estimating the class-conditional densities, adopting a *vector quantization* approach [9].

A. Introduction to Vector Quantization

For compression and communication applications, a *vector quantizer* (VQ) is described as a combination of an encoder and a decoder [8]. A p-dimensional VQ consists of two mappings: an encoder $\gamma(\mathbf{y})$: $\mathbf{A} \rightarrow \mathbf{M}$, mapping the input alphabet $\mathbf{A}$ to the channel symbol set $\mathbf{M}$, and a decoder $\beta(\mathbf{v})$: $\mathbf{M} \rightarrow \hat{\mathbf{A}}$ which maps $\mathbf{M}$ to the output alphabet $\hat{\mathbf{A}}$ (or *codebook*). A distortion measure $\mathcal{D}(\mathbf{y}, \hat{\mathbf{y}})$ specifies the cost associated with quantization, where $\hat{\mathbf{y}} = \beta(\gamma(\mathbf{y}))$. An optimal quantizer minimizes the average distortion under a size constraint on $\mathbf{M}$ [8]. The generalized Lloyd algorithm (GLA) is an iterative algorithm for obtaining a (locally) optimal VQ. Under a mean square error (MSE) distortion criterion, GLA is equivalent to the K-means ($K = p$) clustering algorithm [11]. Any given input vector $\mathbf{y} \in \mathbf{A}$ is quantized into the closest (in $\mathcal{D}$) of the p codebook vectors. This defines a partition of the space $\mathbf{A}$ into the so-called Voronoi cells $\{S_i, i = 1, 2, \cdots, K\}$ [8]. A comprehensive study of VQ can be found in [3], [8].

B. Vector Quantization for Density Estimation

Vector quantization provides an efficient tool for density estimation [9]. Consider n training samples from a class ω. In order to estimate the class-conditional density of the ith feature vector, $f(\mathbf{y}^{(i)}|\omega)$, VQ is used to obtain q (with $q < n$, usually $q \ll n$) codebook vectors, $\mathbf{v}_j^{(i)}$ ($1 \leq j \leq q$), from the training data.[1] In the so-called *high-resolution* approximation (i.e., for small Voronoi cells), this density can be approximated by a piecewise-constant function over each cell $S_j^{(i)}$, with value

$$f\left(\mathbf{y}^{(i)}|\omega\right) \approx \frac{m_j^{(i)}}{Vol\left(S_j^{(i)}\right)}, \qquad \text{for } \mathbf{y}^{(i)} \in S_j^{(i)} \tag{3}$$

where $m_j^{(i)}$ and $Vol(S_j^{(i)})$ are the ratio of training samples falling into cell $S_j^{(i)}$ and the volume of cell $S_j^{(i)}$, respectively, (see [9]). This approximation fails if the cells are not sufficiently small, for example, when the dimensionality of $\mathbf{y}^{(i)}$ is large. In that case, the class-conditional densities can be approximated using a mixture of Gaussians [9], [43], each centered at a codebook vector. The MSE criterion is the sum of the Euclidean distances of each training sample from its closest codebook vector. From a mixture point of view, this is equivalent to assuming covariance matrices of the form $\sigma^2\mathbf{I}$ (where $\mathbf{I}$ is the identity) [43], leading to

$$f\left(\mathbf{y}^{(i)}|\omega, \boldsymbol{\theta}_{(q)}\right) \propto \sum_{j=1}^{q} m_j^{(i)} \exp\left(-\frac{||\mathbf{y}^{(i)} - \mathbf{v}_j^{(i)}||^2}{2\sigma^2}\right) \tag{4}$$

where $\boldsymbol{\theta}_{(q)}^{(i)} = \{\mathbf{v}_1^{(i)}, \cdots \mathbf{v}_q^{(i)}, m_1^{(i)}, \cdots, m_{q-1}^{(i)}\}$ (note that $\sum_j m_j^{(i)} = 1$). The value of σ is not estimated by the VQ algorithm, and so we empirically choose it for each feature. Alternatively, we could use the EM algorithm to directly find *maximum likelihood* (ML) estimates of the mixture parameters, under a diagonal covariance constraint [19]. This choice is computationally demanding, and we have found that the value of σ is not crucial; it simply affects the number of codebook vectors that influence classification. Unless σ is exceptionally

[1] Actually, learning vector quantization (LVQ) is used to select the codebook vectors. LVQ does not run the GLA separately for each class; in this algorithm, the codebook vectors are also "pushed away" from incorrectly classified samples (see [14], [29]).

large, only a few codebook vectors close to the input pattern influence the class-conditional probabilities.

C. Selecting Codebook Size

Selecting q is a key issue in using a VQ, or a mixture, for density representation. We start by noting that GLA approximately looks for the *maximum likelihood* (ML) estimates of the parameters of the mixture in (4). In fact, the EM algorithm becomes exactly equivalent to the GLA when the variance σ^2 goes to zero [29]. We will therefore apply an MDL criterion to select q, since MDL allows extending *maximum likelihood* (ML) estimation to situations where the dimension of the model is unknown [30].

Consider a training set of n independent samples $\{\mathbf{y}(1), \cdots, \mathbf{y}(n)\}$, from the class ω. These are, of course, samples of one of the features, although here we omit this from the notation to keep it simpler. A direct application of the standard MDL criterion would lead to the following criterion to select q [the size of the mixture in (4)]

$$\hat{q} = \arg\min \left\{ -\sum_{s=1}^{n} \log f(\mathbf{y}(s)|\omega, \hat{\boldsymbol{\theta}}_{(q)}) + \frac{\zeta(q)}{2} \log(n) \right\}$$

where $\hat{\boldsymbol{\theta}}_{(q)}$ is the ML estimate assuming size q, and $\zeta(q) = q - 1 + q\dim(\mathbf{y})$ is the number of real-valued parameters needed to specify a q-component mixture (with $\dim()$ denoting "dimension of") [30]. Notice that the additional term proportional to $(1/2)\log(n)$ grows with q, thus counterbalancing the unbounded increase, with q, of the likelihood. The penalty $(1/2)\log(n)$ paid by each additional real parameter has an asymptotical justification (see [30]). For a mixture, however, it can be argued that each center does not "see" n data points, but only (on average) $n * m_j$ (for the jth center) (see [15] and [6], for details). This leads to the following *modified* MDL (MMDL) criterion

$$\hat{q} = \arg\min_{q} \left\{ -\sum_{s=1}^{n} \log f\left(\mathbf{y}(s)|\omega, \boldsymbol{\theta}_{(q)}\right) + \frac{q-1}{2} \log n + \frac{\dim(\mathbf{y})}{2} \sum_{j=1}^{q} \log(m_j\ n) \right\}. \quad (5)$$

V. Implementation Issues

Experiments were conducted on two databases (both independently and combined) of 5081 (indoor/outdoor classification) and 2716 (city/landscape classification and further classification of landscape images) images. The two databases, henceforth referred to as D1 and D2, have 866 images in common, leading to a total of 6931 distinct images, collected from various sources (Corel library, scanned personal photographs, key frames from TV serials, and images downloaded from the Web) and are of varying sizes (from 150×150 to 750×750). The color images are stored with 24-bits per pixel in JPEG format. The ground truth for all the images was assigned by a single subject.

A. Image Features

Outdoor images tend to have uniform spatial color distributions, such as the sky is on top and is typically blue. Indoor images tend to have more varied color distributions and have more uniform lighting (most are close up shots). Thus, it seems logical that spatial color distribution can discriminate between indoor and outdoor images. On the other hand, shape features may not be useful because objects with similar shapes can be present in both indoor and outdoor scenes. Therefore, we use spatial color information features to represent these qualitative attributes. Specifically, first- and second-order moments in the LUV color space were used as color features (it was pointed out in [7] that LUV moments yield better results in image retrieval than other spaces). The image was divided into 10×10 subblocks and six features (three means and three standard deviations) were extracted [37], [41]. As another set of features for indoor/outdoor classification, we extract subblock MSAR texture features as described in [18], [39].

We looked for similar qualitative attributes for city/landscape classification, and further classification of landscape images. City images usually have strong vertical and horizontal edges due to the presence of man-made objects. Non-city images tend to have randomly distributed edge directions. The edge direction distribution seems then as a natural feature to discriminate between these two categories [42]. On the other hand, color features would not have sufficient discriminatory power as man-made objects have arbitrary colors. In the case of further classification of landscape images as sunset, forest, or mountain, global color distributions seem to adequately describe these classes. Sunset pictures typically have saturated colors (mostly yellow and red); mountain images tend to have the sky in the background (typically blue); and forest scenes tend to have more greenish distributions. Based on the above observations, we use edge direction features (histograms and coherence vectors) for city/landscape classification and color features (histograms, coherence vectors, and spatial moments) in HSV and LUV color space for further classification of landscape images [25], [38], [42]. Table I summarizes the qualitative attributes of the various classes and the features used to represent them.

B. Vector Quantization

We used the LVQ_PAK package [14] for vector quantization. Half of the database was used to train the LVQ for each of the image features. The MMDL criterion (Section IV-C) was used to determine the codebook sizes. For the indoor and outdoor classes, with the spatial color moment features, Fig. 4(a)–(c) plots the MMDL cost function [(5)] versus the codebook size q. These plots show that $q \sim 10$ and $q \sim 15$ are the MMDL choices for the indoor and outdoor classes, respectively. For the combination of the two classes, $q \sim 30$ minimizes the MMDL criterion. To confirm this choice from a classification point of view, Fig. 5 plots the accuracy of the indoor/outdoor classifier (on an independent test set of size 2540) as a function of the total codebook size q. As q is initially increased, the classifier accuracy improves. However, it soon stabilizes and further increasing q beyond 30 does not improve the accuracy. This con-

TABLE I
QUALITATIVE ATTRIBUTES OF THE SEVERAL CLASSIFACTION PROBLEMS AND ASSOCIATED LOW-LEVEL FEATURES

Classification Problem	Qualitative Attributes	Low-level Features
Indoor/outdoor	spatial color and intensity distributions	10×10 sub-block color moments in LUV space
City/landscape	distribution of edges	edge direction histograms and coherence vectors
Sunset/forest/mountain	global color distributions and saturation values	spatial moments, color histograms, and coherence vectors in HSV and LUV space

clusion (and similar ones for city/landscape classification) supports the use of MMDL for codebook size selection.

Based on similar analysis (see [40]), 20 codebook vectors were extracted for each of the city and landscape classes. For further classification of landscape images, a codebook of five vectors was selected for each class. These vectors were then stored as representatives of each class. Table II shows the number and dimensionality of the codebook vectors for the various classification problems.

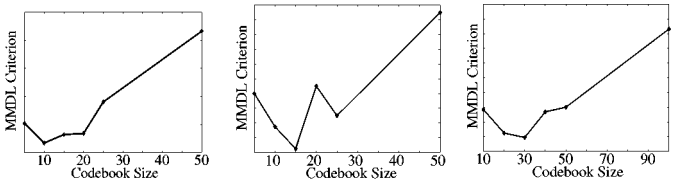
VI. EXPERIMENTAL RESULTS

Given an input image, the classifier computes the class-conditional probabilities for each of the features using (4). These probabilities are then used to obtain the MAP classification [(2)]. We present classification accuracies on a set of independent test patterns as well as on the training patterns. We have done classifications based on individual features and also based on combinations of features [assumed independent, (1)]. As we show later, each of the individual features chosen for the classification problems has sufficient discrimination power for that particular classification problem, and introducing other features does not significantly improve the results.

A. Indoor/Outdoor Classification

Database D1 (2470 indoor and 2611 outdoor images) was used to train the indoor/outdoor classifier. Apart from the color moment features, we also considered the subblock MSAR texture features [39], edge direction features, and color histograms. MSAR features yielded an accuracy of around 75% on the test set. A higher classification accuracy (using a K-NN classifier and leave-one-out testing) of 84% on a database of 1324 images was reported in [39]. We attribute this discrepancy to differences in the database (our database of 5081 images is larger) and mode of testing (we report results on an independent test set). Edge direction and coherence vector features yielded an accuracy of around 60%, while the color moment features lead to a much higher accuracy of around 90%. These results show that the spatial color distribution (probably capturing illumination changes) is suited for indoor/outdoor classification. A combination of color and texture features did not yield a better accuracy than color moment features alone.

Table III shows the classification results with the color moment features for indoor/outdoor classification. The classifier showed an accuracy of 94.2% and 88.2% on the training set and an independent test set (Test Set 1 in Table III), respectively. On a different test set (Test Set 2 in Table III) of 1850 images from database D2, the classifier accuracy was 88.7%. An accuracy of 90.5% was obtained on the entire database of 6931 images. Szummer *et al.* [39] use a K-NN classifier and report 90% accuracy using leave-one-out testing, for the indoor/outdoor classification on a database of 1324 images. Thus, our classifier's performance is comparable to those reported in the literature. A major advantage of the Bayesian classifier over K-NN classifier is its efficiency due to the small number of codebook vectors needed to represent the training data.



(a) Indoor Class (b) Outdoor Class (c) Combined

Fig. 4. Determining codebook size for spatial color moment features for the indoor/outdoor classification problem. (a) Indoor class, (b) outdoor class, and (c) indoor and outdoor classes combined.

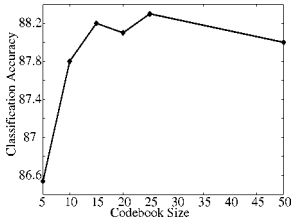


Fig. 5. Accuracy of the indoor/outdoor classifier with increasing codebook size (trained on 2541 images and tested on an independent test set of 2540 images).

TABLE II
DIMENSIONALITIES AND CODEBOOK SIZES FOR EACH CLASSIFIER

Classification Problem	# of Codebook Vectors / Class	Feature Dimensionality
Indoor/Outdoor	15	600
City/Landscape	20	145
Sunset/Forest/Mountain	5	640

TABLE III
ACCURACIES (IN PERCENT) FOR INDOOR/OUTDOOR CLASSIFICATION USING COLOR MOMENTS; TEST SET 1 AND TEST SET 2 ARE INDEPENDENT TEST SETS

Test Data	Database Size	Accuracy (%)
Training Set	2,541	94.2
Test Set 1	2,540	88.2
Test Set 2	1,850	88.7
Entire Database	6,931	90.5

Fig. 6 shows a representative subset of the misclassified indoor/outdoor scenes. Presence of bright spots either from some light source or from sunshine through windows and doors seems to be a main cause of misclassification of indoor images. The main reasons for the misclassification of outdoor images are 1) uniform lighting on the image mostly as a result of a close-up shot and 2) low-contrast images (several of the indoor images used in the training set were low contrast digital images and hence most low contrast outdoor images were classified as indoor scenes). The results show that spatial color distribution captured in the subblock color moment features has sufficient discrimination power for indoor/outdoor classification.

B. City/Landscape Classification

The city versus landscape classification problem and further classification of landscape images as sunset, forest, or mountain using the Bayesian framework has been addressed in detail in [40]. We summarize the results here. Table IV shows the results for the city/landscape classification problem using database D2. Edge direction coherence vector provides the best individual accuracy of 97.0% for the training set and 92.9% for the test set. A total of 126 images were misclassified (95.3% accuracy) when the edge direction coherence vector was combined with the color histogram. Fig. 7 shows a representative subset of misclassified images. Most of the misclassifications for city images could be attributed to the following reasons:

1) long distance city shots at night (difficulty in extracting edges);
2) top view of city scenes (lack of vertical edges);
3) highly textured buildings;
4) trees obstructing the buildings.

Most of the misclassified landscape images had strong vertical edges from tree trunks, close-ups of stems, fences, etc., that led to their assignment to the city class.

We also computed the classification accuracy using the edge direction coherence vector on an independent test set of 568 outdoor images from database D1. A total of 1177 images of the 4181 outdoor images in database D1 contained close ups of human faces. We removed these images for the city/landscape test. Recent advances show that faces can be detected rather reliably [32]. Of the remaining test images, we extracted 568 that were not part of database D2. The edge direction features yielded an accuracy of 90.0% (57 misclassifications out of the 568 images). Combining color histogram features with edge direction coherence vector features reduced the misclassification

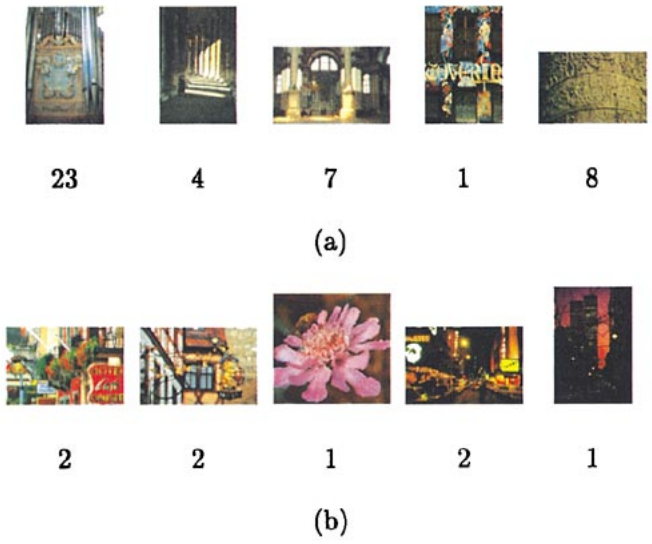


Fig. 6. Some misclassified (a) indoor and (b) outdoor images using color moment features; the corresponding confidence values (in percent) associated with the true class are presented.

in the above experiment to 56, confirming that edge directions are sufficient to discriminate between city and landscape.

C. Further Classification of Landscape Images

While our limited experiments on human subjects [42] revealed classes such as sunset and sunrise, forest and farmland, mountain, pathway, water scene, etc., these groups were not consistent among the subjects in terms of the actual labeling of the images. We found it extremely difficult to generate a semantic partitioning of landscape images. We thus restricted classification of landscape images into three classes that could be more unambiguously distinguished: sunset, forest, and mountain. Of these 528 images, a human subject labeled 177, 196, and 155 images as belonging to the forest, mountain, and sunset classes, respectively. A two-stage classifier was constructed. First, we classify an image into either sunset or the forest and mountain class. The above hierarchy was based on the human study, as shown in Fig. 3(a), where the sunset cluster seemed to be more compact and well separated from the other categories in the landscape class.

Table V shows the results for the classification of landscape images into sunset vs. forest and mountain classes. The color coherence vector provides the best accuracy of 99.2% for the training set and 93.9% for the test set. Color features do much better than the edge direction features here, since color distributions remain more or less constant for natural images (blue sky, green grass, trees, plants, etc). A total of 18 images were misclassified (a classification accuracy of 96.6%) when the color coherence vector feature was used. We find that combining features does not improve the classification accuracy. This shows that color coherence vector has sufficient discrimination ability for the problem at hand.

Table VI shows the classification results for the individual features for forest and mountain classes (373 images). Spatial color moment features provide the best accuracy of 98.4% for the training set and 93.6% on the test set. A total of 15 images

TABLE IV
CLASSIFICATION ACCURACIES (IN PERCENT) FOR CITY/LANDSCAPE CLASSIFICATION; THE FEATURES ARE ABBREVIATED AS FOLLOWS: EDGE DIRECTION HISTOGRAM (EDH), EDGE DIRECTION COHERENCE VECTOR (EDCV), COLOR HISTOGRAM (CH), AND COLOR COHERENCE VECTOR (CCV)

Test Data	EDH	EDCV	CH	CCV	EDH & CH	EDH & CCV	EDCV & CH	EDCV & CCV
Training Set	94.7	97.0	83.7	83.5	94.8	95.4	96.4	96.9
Test Set	92.0	92.9	75.4	76.0	92.5	92.8	93.4	93.8
Entire Database	93.4	95.0	79.6	79.8	93.7	94.1	94.9	95.3

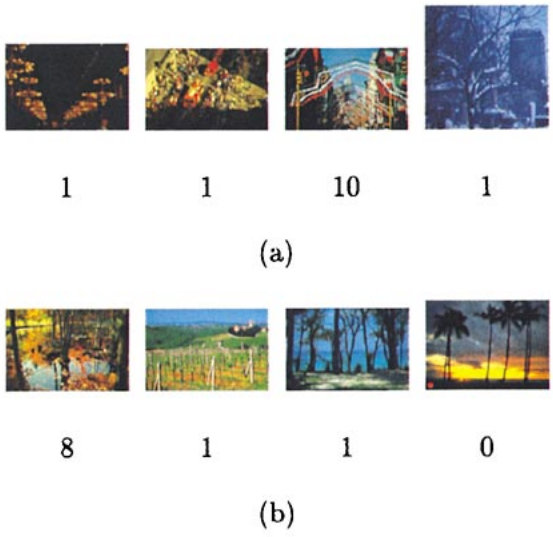


Fig. 7. Subset of the misclassified (a) city images and (b) landscape images using a combination of edge direction coherence vector and color histogram features. The corresponding confidence values (in percent) associated with the true class are indicated.

were misclassified (a classification accuracy of 96%) when the spatial color moment features were used. Again, the combinations of features did not perform better than the color features, showing that these features are adequate for this problem. Note that the spatial color moment features and the color coherence vector features yield similar accuracies for the classification of landscape images. However, the database of 528 images is very small to identify the best color feature for the classification of landscape images. Using color coherence vector features increases the complexity of the classifiers.

D. Error Propagation in Hierarchical Classification

The goal of hierarchical classification is to break a complex problem into simpler problems. However, since each classifier is not perfect, the errors from a classifier located higher up in the tree are propagated to the lower levels.

The indoor/outdoor image classifier yielded an accuracy of 90.5% on the entire database of 6931 images (658 images were misclassified). Of these, 269 images were indoor images out of which 229 were close-ups of people and pets. Out of the remaining 40 images, three were classified as landscape images and 37 were classified as city images. Fig. 8 shows these three indoor images that were misclassified as landscapes. If a face detector is not available and we submit all the 269 images to the city/landscape classifier, it classifies 199 images as city images (most indoor images have man-made structures with strong vertical and horizontal edges) and 70 as landscape. Since we have not yet developed a classifier to identify sunset, forest, and mountain images from other landscape images, in the worst case, all 70 of these images will be fed to the sunset/forest/mountain classifier and hence, degrade the overall classification accuracy. Fig. 8(a) and (b) was classified as sunrise/sunset images and Fig. 8(c) was classified as a forest image.

E. Feature Saliency

The accuracy of the individual classifiers depends on the underlying low-level representation of the images. For example, the edge direction and coherence vector features yield accuracies of about 60% for the indoor/outdoor problem, yet they yield approximately 95% accuracy for the city/landscape problem. This shows the importance of feature definition and selection. We have empirically determined that

1) spatial color moment features are better for indoor/outdoor classification;
2) edge direction histograms and coherence vector features have sufficient discrimination power for city/landscape classification;
3) color moments, histograms, and coherence vectors are more suited for the classification of landscape images.

F. Reject Option

Introducing a reject option is useful, yet a difficult problem in image classification. For Bayesian classifiers, the simplest strategy is to reject images whose maximum *a posteriori* probability is below a threshold T. Table VII shows the accuracies for the *indoor/outdoor* and *city/landscape* image classifiers with reject option, for $T = 0.6$. The indoor/outdoor classifier used spatial color moment features and was trained on 2541 images from database $D1$ and tested on the entire set (6931 images). The classification accuracy improved from 90.5% (no rejection) to 92.1% at 5.4% reject rate. The city/landscape classifier used edge direction coherence vector features; it was trained on 1358 images from database $D2$ and tested on the complete database $D2$ (2716 images). The classification accuracy improved from 95.0% (no rejection) to 95.7% at 2.1% reject rate. There

TABLE V
CLASSIFICATION ACCURACIES (IN PERCENT) FOR SUNSET/FOREST/MOUNTAIN CLASSIFICATION; *SPM* STANDS FOR "SPATIAL COLOR MOMENTS"

Test Data	EDH	EDCV	CH	CCV	SPM	EDH & CH	EDH & CCV	EDCV & CH	EDCV & CCV
Training Set	88.3	88.3	96.2	99.2	98.9	95.9	96.6	95.5	97.0
Test Set	86.3	89.0	89.7	93.9	93.9	90.1	95.4	90.5	95.1
Entire Database	87.4	88.7	93.0	96.6	96.4	93.0	96.0	93.0	96.1

TABLE VI
CLASSIFICATION ACCURACIES (IN PERCENT) FOR FOREST/MOUNTAIN CLASSIFICATION

Test Data	EDH	EDCV	CH	CCV	SPM	EDH & CH	EDH & CCV	EDCV & CH	EDCV & CCV
Training Set	83.4	78.1	92.0	98.9	98.4	94.1	98.4	93.6	98.4
Test Set	87.1	77.2	91.4	91.9	93.6	93.0	92.5	93.5	91.9
Entire Database	85.3	77.7	91.7	95.5	96.0	93.6	95.5	93.6	95.2

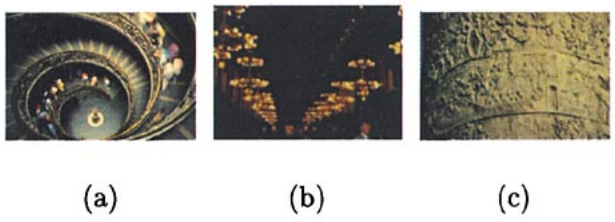


Fig. 8. Indoor images misclassified as landscape.

TABLE VII
CLASSIFIER PERFORMANCE UNDER A REJECT OPTION

Classification Problem	Training Set Size	Test Set Size	Reject Rate (%)	Accuracy (%)
Indoor/Outdoor	2,541	6,931	5.4	92.1
City/Landscape	1,358	2,716	2.1	95.7

is a clear accuracy/reject tradeoff; too much rejection may be needed to further reduce the error rate.

VII. INCREMENTAL LEARNING

It is well-known that the classification performance depends on the training set size: the more comprehensive a training set, the better the classification performance. Table VIII compares the classification accuracies of the indoor/outdoor image classifier (based on spatial color moment features) as the training set size is increased. As expected, increasing the training set size improves the classification accuracy. When we trained the LVQ with all the available 5081 images using the color moment features, a classification accuracy of 95.7% (resubstitution accuracy) was obtained. This shows that the classifier still has the capacity to learn, provided additional training samples are available. The above observations illustrate the need for an *incremental* learning method for Bayesian classifiers.

Collecting a large and representative training set is expensive, time consuming, and sometimes not feasible. Therefore, it is not realistic to assume that a *comprehensive* training set is initially available. Rather, it is desirable to incorporate learning techniques in a classifier [22], [29]. As additional data become available, the classifier should be able to adapt, while retaining what it has already learnt. Since the training set can become extremely large, it may not be feasible to store all the previous data. Therefore, instead of retraining the classifier on the entire training set every time new samples are collected, it is more desirable to *incrementally* train the classifier on the new samples. For the Bayesian classifier proposed above, the initial training set is represented in terms of the codebook vectors ($\mathbf{v}_j^{(i)}$). Learning involves incrementally updating these codebook vectors as new training data become available.

TABLE VIII
CLASSIFICATION ACCURACIES AS A FUNCTION OF TRAINING SET SIZE ON THE INDOOR/OUTDOOR CLASSIFIER

Training Set Size	Ind. Test Set Size	Accuracy (%)
700	2,540	75.3
1,418	2,540	79.8
1,768	2,540	86.0
2,192	2,540	86.4
2,541	2,540	88.2

One simple method to retrain the classifier is to train it with the new data, i.e., start with the previously learnt codebook vectors and run the LVQ with the new data. This straightforward method, however, does not assign an appropriate weight to the previously learnt codebook. In other words, if a classifier was trained on a large number of samples and then a small number of new samples are used to further train the classifier using the

above learning paradigm, the new data will unduly influence the current value of the codebook vectors. Learning with this small amount of new data will in fact lead to unlearning of the distribution based on previous samples. Table IX demonstrates the results of training the indoor/outdoor classifier using only the new data. The indoor/outdoor classifier was initially trained on 1418 images and yielded an accuracy of 79.8% on an independent test set of 2540 images. When the classifier is further trained with 350 new images, the performance on the independent test set deteriorates to 63.7%. When the classifier is further trained on an additional 773 samples using the naive approach, the accuracy on the test set slightly recovers to 72.5%. Note that when all the available data were used ($1418 + 350 + 773 = 2541$ images), the accuracy on the independent test set was 88.2% (Table VIII). These results show that any robust incremental learning scheme must assign an appropriate weight to the already learnt distribution.

TABLE IX
NAIVE APPROACH TO INCREMENTALLY TRAINING A CLASSIFIER. ACCURACIES ARE REPORTED ON AN INDEPENDENT TEST SET OF SIZE 2540

Initial classifier (1,418 training samples)	79.8% accuracy
350 additional samples	63.7% accuracy
773 additional samples	72.5% accuracy

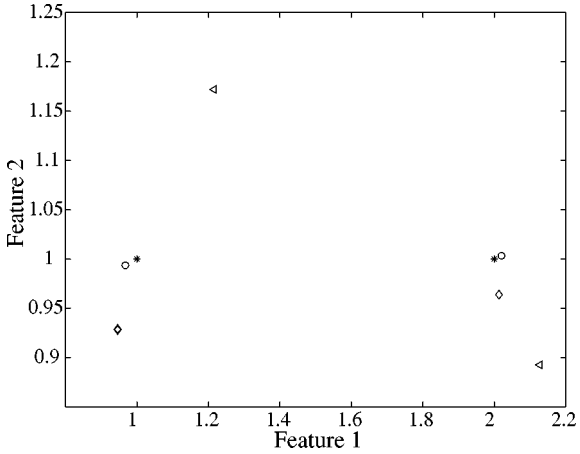


Fig. 9. Incremental learning with synthetic Gaussian data; ($*$) represents the true means; ($\triangleleft$) represents the initial codebook vectors learnt from 100 samples per class; ($\diamondsuit$) represents the codebook vectors after an additional 400 samples per class; and ($\circ$) represents the codebook vectors after 500 more samples per class.

A. Proposed Incremental Learning Scheme

The idea behind the proposed scheme is to try to generate the original samples from the codebook vectors and then augment these estimated samples to the new training set. The combined training set is then used (starting at the current codebook vectors) to determine the new set of codebook vectors. This method differs from traditional bootstrapping [11] which assumes that the original training samples are available for sampling with replacement. In our case, the new samples representing the original training set are generated based on the number of training samples, the proportion of these samples assigned to each codebook vector ($m_j^{(i)}$), and the codebook vectors themselves. Fig. 9 illustrates this learning paradigm for synthetic data where two-dimensional samples are generated from two i.i.d. Gaussian distributions with mean vectors $[1, 1]^T$ and $[2, 1]^T$, respectively, and identity covariance matrices. We see that as the classifier is incrementally trained with additional data, the new codebook vectors approach the true mean vectors.

We have used the following methods to generate (independent) samples from a codebook vector.

- *Case 1*: Using duplicates of the codebook vectors as the samples (this is, by far, the least computationally demanding case, since no samples have to be actually generated).
- *Case 2*: Sampling from a multivariate Gaussian, with covariance $\sigma^2\mathbf{I}$, centered at the codebook vectors.
- *Case 3*: Same as *Case 2*, except that we use a diagonal covariance matrix. The diagonal elements correspond to the individual variances of features of the training samples assigned to the respective codebook vector.
- *Case 4*: Sampling from a multivariate Gaussian with covariance $\sigma^2\mathbf{I}$, centered at the mean of the training patterns assigned to the codebook vector. Note that each codebook vector need not be the mean of the samples assigned to it, as both positive and negative examples influence the codebook vectors (see footnote 1, Section V-B).
- *Case 5*: Same as *Case 4*, except that we use a diagonal covariance matrix. The diagonal elements correspond to the individual variances of features of the training samples assigned to the respective codebook vector.

The last four methods do not enforce the condition that the generated samples be closest to the codebook vector they are estimated from. The above criterion is satisfied in *Case 1* since the generated samples are all identical to the codebook vector. The number of samples generated from each codebook vector are the same as the number of original training samples assigned to that codebook vector. If we had chosen to use the EM algorithm to estimate mixture representations of the class-conditional densities, instead of LVQ, then, incremental learning could be achieved by using an on-line version of EM, such as the one in [35].

B. Experimental Results

We have tested the proposed incremental learning method with the Bayesian indoor/outdoor and city/landscape classifiers. Initially, half the images from the database were used to train a classifier. The classifier was then incrementally trained (all the five methods described above were tested) using the remaining images. The performance of a classifier trained on the entire set of database images (nonincremental learning) was also measured. Table X shows the classification accuracies for the various methods. The best classification accuracies achieved for each of the classifiers were 95.9% for the city/landscape classifier (on 2716 images) and 94.6% for the indoor/outdoor classifier (on 5081 images), versus 97.0% and 95.7%, respectively, for the classifiers trained on the entire database. These

results show that a classifier trained incrementally achieves almost similar accuracies as one trained with all the data. The five methods used to regenerate "training" samples perform equally well. Since the first method (*Case 1*) requires, by far, the least additional storage (only one number denoting the total number of training samples used to train the classifier so far) and computation (no random number generation), it clearly has the best cost/performance tradeoff.

VIII. Feature Subset Selection

Automatic feature subset selection is an important issue in designing classifiers. In fact, one usually finds that the performance of a classifier trained on a finite number of samples starts deteriorating as more features are added beyond a certain number (the *curse of dimensionality* [4], [12], [29]). Can the classification be improved using feature subset selection methods? Selecting the optimal features is a problem of exponential time complexity and various suboptimal heuristics have been proposed [12], [13].

Jain and Zongker [13] studied the merits of various feature subset selection methods. While the branch-and-bound algorithm proposed in [23] is "optimal," it requires the feature selection criterion function to be monotone (i.e., it cannot decrease when new features are added). The above requirement may not be true for small samples. It is thus desirable to use approximate methods that are fast and also guarantee near optimal solutions. Therefore, we tested the sequential floating forward selection (SFFS) method, which was shown to be a promising alternative where the branch-and-bound method cannot be used [28].

We have also applied a simple heuristic procedure based on clustering the features (using K-means [11]), trying to remove redundancy. The feature components assigned to each cluster are then averaged to form the new feature. Thus, the number of clusters determines the final number of features. Although this method does not guarantee an optimal solution, it does attempt to eliminate highly correlated features in high-dimensional feature spaces. We refer to this method as the feature cluster (FC) method.

A. Experiments Using SFFS

We have experimented with feature subset selection on the indoor/outdoor classifier using the implementation of SFFS provided in [13]. We found the algorithm to be very slow over the entire training set of 2541 training samples from database $D1$. We hence took 700 samples each from the training and test sets for the feature subset selection process. Our results using SFFS are summarized as follows.

- It took the program 12 days on a Sun Ultra 2 Model 2300 (dual 300-MHz processors) processor with 512 MB memory to select up to 67 features from the 600-dimensional feature vector for the small training set of 700 samples.
- The best accuracy of 87% on the independent test set of 700 samples was provided by a subset of 52 features, compared to the 88.2% accuracy using all the 600 features.
- Training a new classifier, with the 52 features selected by SFFS, on the 2541 samples from the training set of database $D1$ yielded 82.2% accuracy on the test set (2540 samples). The lower accuracy on larger sets agrees with the observations in [13] on the pitfalls of using feature subset selection on sparse data in a high-dimensional space.

TABLE X
Classification Accuracies (Percent) with and without Incremental Learning; *Case i* Represents one of the Incremental Methods; in *Non-Incremental*, the Whole Database was Used in Training

Method	City/Landscape	Indoor/Outdoor
Case 1	95.9	94.1
Case 2	95.8	94.3
Case 3	95.8	94.5
Case 4	95.8	94.3
Case 5	95.9	94.6
Non-Incremental	97.0	95.7

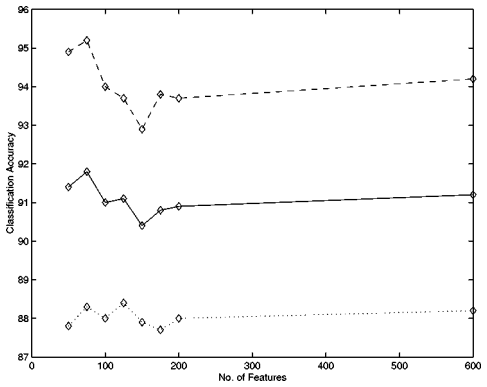


Fig. 10. Accuracies for the indoor/outdoor classifier trained on varying sized feature vectors generated by FC (from the 600-dimensional spatial color moment features); the dashed, dotted, and solid lines represent, respectively, the accuracies of the training set (2541 images), test set (2540 images), and the entire database (5081 images).

B. Experiments Using FC

The spatial color moment features used for indoor/outdoor classification (feature dimensionality of 600) were clustered to generate new feature vectors of sizes 50, 75, 100, 125, 150, 175, and 200. The components assigned to each cluster were averaged to define a new feature. This approach is incomparably faster than SFFS, taking only a few seconds on a training set size of 2541, from database $D1$. The classification accuracies for the various feature set sizes are plotted in Fig. 10. A codebook size of 30 (optimal for the spatial color moments features) was used for all the features. The best classification accuracy of 91.8% on the entire database of 5081 images (95.2% on the training set and 88.3% on an independent test set of 2540 images) was obtained with feature vectors of 75 components. Note that these accuracies are marginally better than those obtained from training the classifier on the 600-dimensional spatial color moment features (accuracy of 88.2% on an independent test set of 2540 images and an accuracy of 94.2% on the training set). On examining the feature components that were clustered to-

TABLE XI
ACCURACIES FOR INDOOR/OUTDOOR CLASSIFICATION WITH THE FEATURES OBTAINED BY FEATURE CLUSTERING

Test data	new feature set; 75 components, $q = 50$	spatial color moments; 600 components, $q = 30$
Training Set	96.0	94.2
Test Set	88.8	88.2
Entire Database	92.4	91.2

gether, we found that all groupings were formed within features of neighboring image regions. These preliminary results show that clustering the features (linear combination of features) is more efficient and accurate than the SFFS feature subset selection method for very high-dimensional feature vectors.

We used MMDL to select the optimal codebook size for the new feature set. The criterion selected $q \sim 50$, for the indoor/outdoor classifier based on the 2541 training samples. Therefore, we extracted 25 codebook vectors each for the indoor and outdoor image classes under the new feature set of 75 components. This illustrates how a reduction in feature size (from 600 spatial color moment features to the new set of 75 features) leads to the generation of a larger codebook (50 vectors represent the underlying density as opposed to 30 for the full spatial color moment features).

Table XI shows the accuracies for the classifier trained on these new features compared against those of the classifier trained on the full spatial color moment features. The FC method for feature selection improved the classifier performance from 91.2% to 92.4% for the indoor/outdoor problem (on a database of 5081 images), while reducing the feature vector dimensionality from 600 components to 75 components. Recall that the low-level features used for the indoor/outdoor image classification problem are extracted over 10×10 subblocks in the image. Usually, neighboring subblocks in an image have similar features as various objects span multiple subblocks (e.g., sky, forest, etc., may span a number of subblocks in many images). Other linear and nonlinear techniques for feature extraction (PCA, Discriminant Analysis, Sammon's nonlinear projection) may be as effective as FC in reducing feature dimensionality.

IX. CONCLUSION AND FUTURE WORK

User queries in content-based retrieval are typically based on semantics and not on low-level image features. Providing high-level semantic indices for large databases is a challenging problem. We have shown that certain high-level semantic categories can be learnt using specific low-level image features under the constraint that the images do belong to one of the classes under consideration. Specifically, we have developed a hierarchical classifier for vacation images. At the top level, vacation images are classified as indoor or outdoor. The outdoor images are then classified as city or landscape (we assume a face detector that separates close-up images of people in outdoor images) and finally, a subset of landscape images are classified as sunset, forest, or mountain. We have adopted a Bayesian classification approach, using vector quantization (LVQ) to learn the class-conditional probability densities of the features. This approach has the following advantages:

1) small number of codebook vectors represent a particular class of images, regardless of the size of the training set;
2) it naturally allows for the integration of multiple features through the class-conditional densities;
3) it not only provides a classification rule, but also assigns a degree of confidence in the classification, which may be used to build a reject option.

Classifications based on local color moments, color histograms, color coherence vectors, edge direction histograms, and edge direction coherence vectors have shown promising results.

The accuracy of the above classifiers depends on the features used, the number of training samples, and the classifier's ability to learn the true decision boundary from the training data. We have developed methods for incremental learning and feature subset selection. Another challenging issue is to introduce a reject option. In the simplest form, the a posteriori class probabilities can be used for rejection (rejecting images whose maximum a posteriori probability is less than a threshold, T—say 0.6). We are looking at other means of adding the reject option into the system. Finally, we will introduce other binary classifiers into the system for categories such as day/night, people/nonpeople, text/nontext, etc. These classifiers can be added to the present hierarchy to generate semantic indices into the database.

REFERENCES

[1] I. Biederman, "On the semantics of a glance at a scene," in *Perceptual Organizations*, M. Kubovy and J. R. Pomerantz, Eds. Hillsdale, NJ: Lawrence Erlbaum, 1981, pp. 213–253.
[2] I. Biederman, "Aspects and extensions of a theory of human image understanding," in *Computational Processes in Human Vision: An Interdisciplinary Perspective*, Z. W. Pylyshyn, Ed. Norwood, NJ: Ablex, 1988, pp. 370–428.
[3] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M. Gray, "Using vector quantization for image processing," *Proc. IEEE*, vol. 81, pp. 1326–1341, Sept. 1993.
[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[5] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *J. Intell. Inform. Syst.*, vol. 3, pp. 231–262, 1994.
[6] M. Figueiredo and A. K. Jain, "Unsupervised selection and estimation of finite mixture models," in *Proc. Int. Conf. Pattern Recognition*, Barcelona, Spain, 2000.
[7] B. Furht, Ed., "Content-based image indexing and retrieval," in *The Handbook of Multimedia Computing*. Boca Raton, FL: CRC, 1998, ch. 13.
[8] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, vol. 1, pp. 4–29, Apr. 1984.
[9] R. M. Gray and R. A. Olshen, "Vector quantization and density estimation," in *Proc. SEQUENCES97*, 1997.
[10] A. Hampapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J. Bach, M. Gorkani, and R. Jain, "Virage video engine," in *Proc. SPIE Storage Retrieval Image Video Databases V*, San Jose, CA, Feb. 1997, pp. 188–197.
[11] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[12] A. K. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 4–38, Jan. 2000.
[13] A. K. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 153–158, Feb. 1997.

[14] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola, "LVQ_PAK: A program package for the correct application of learning vector quantization algorithms," in *Proc. Int. Joint Conf. Neural Networks*, Baltimore, MD, June 1992, pp. 725–730.
[15] J. L. M. Figueiredo and A. K. Jain, "On fitting mixture models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, E. Hancock and M. Pellilo, Eds. Berlin, Germany: Springer-Verlag, 1999.
[16] W. Y. Ma and B. S. Manjunath, "Image indexing using a texture dictionary," in *Proc. SPIE Conf. Image Storage Archiving Systems*, vol. 2606, Philadelphia, PA, October 1995, pp. 288–298.
[17] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Santa Barbara, CA, Oct. 1997, pp. 568–571.
[18] J. Mao and A. K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern Recognit.*, vol. 25, no. 2, pp. 173–188, 1992.
[19] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.
[20] S. Mehrotra, Y. Rui, M. Ortega, and T. S. Huang, "Supporting content-based queries over images in MARS," in *Proc. IEEE Int. Conf. Multimedia Computing Systems*, ON, Canada, June 3–6, 1997, pp. 632–633.
[21] T. P. Minka and R. W. Picard, "Interactive learning using a society of models," *Pattern Recognit.*, vol. 30, no. 4, p. 565, 1997.
[22] T. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
[23] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vol. 26, pp. 917–922, Sept. 1977.
[24] T. V. Papathomas, T. E. Conway, I. J. Cox, J. Ghosn, M. L. Miller, T. P. Minka, and P. N. Yianilos, "Psychophysical studies of the performance of an image database retrieval system," in *Proc. IS&T/SPIE Conf. Human Vision Electronic Imaging III*, San Jose, CA, July 1998, pp. 591–602.
[25] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proc. 4th ACM Conference on Multimedia*, Boston, MA, Nov. 1996, http://simon.cs.cornell.edu/Info/People/rdz/rdz.html.
[26] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *Proc. SPIE Storage Retrieval Image Video Databases II*, pp. 34–47, Feb. 1994.
[27] R. W. Picard and T. P. Minka, "Vision texture for annotation," *Multimedia Syst.*, vol. 3, pp. 3–14, 1995.
[28] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, pp. 1119–1125, Nov. 1994.
[29] B. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996.
[30] J. Rissanen, *Stochastic Complexity in Stastistical Inquiry*. Singapore: World Scientific, 1989.
[31] B. E. Rogowitz, T. Frese, J. Smith, C. A. Bouman, and E. Kalin, "Perceptual image similarity experiments," in *Proc. IS&T/SPIE Conf. Human Vision Electronic Imaging III*, San Jose, CA, July 1998, pp. 576–590.
[32] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 23–38, Jan. 1998.
[33] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 644–655, Sept. 1998.
[34] P. G. Schyns and A. Oliva, "From blobs to boundary edges: Evidence for time and spatial scale dependent scene recognition," *Psychol. Sci.*, vol. 5, pp. 195–200, 1994.
[35] Y. Singer and M. Warmuth, "A new parameter estimation method for Gaussian mixtures," in *Advances in Neural Information Processing Systems 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA: MIT Press, 1999.
[36] J. R. Smith and S. F. Chang, "Visualseek: A fully automated content-based image query system," in *Proc. ACM Multimedia*, Boston, MA, Nov. 1996, pp. 87–98.
[37] M. Stricker and A. Dimai, "Color indexing with weak spatial constraints," in *Proc. SPIE Storage Retrieval Image Video Databases IV*, San Jose, CA, Feb. 1996, pp. 29–41.
[38] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.
[39] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," in *IEEE Int. Workshop Content-Based Access Image Video Databases (in conjunction with ICCV'98)*, Bombay, India, Jan. 1998.
[40] A. Vailaya, M. Figueiredo, A. Jain, and H.-J. Zhang, "A Bayesian framework for semantic classification of outdoor vacation images," in *Proc. SPIE Storage Retrieval Image Video Databases VII*, vol. 3656, San Jose, CA, Jan. 1999, pp. 415–426.
[41] A. Vailaya, M. Figueiredo, A. Jain, and H.-J. Zhang, "Content-based hierarchical classification of vacation images," in *Proc. IEEE Multimedia Systems'99*, vol. 1, Florence, Italy, June 7–11, 1999, pp. 518–523.
[42] A. Vailaya, A. K. Jain, and H. J. Zhang, "On image classification: City images vs. landscapes," *Pattern Recognit.*, vol. 31, no. 12, pp. 1921–1936, 1998.
[43] N. Vasconcelos and A. Lippman, "Library-based coding: A representation for efficient video compression and retrieval," in *Data Compression Conf. '97*, Snowbird, UT, 1997.
[44] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu, "Video parsing retrieval and browsing: An integrated and content-based solution," in *Proc. ACM Multimedia '95*, San Francisco, CA, Nov. 5–9, 1995, pp. 15–24.
[45] H. J. Zhang and D. Zhong, "A scheme for visual feature based image indexing," in *Proc. SPIE Conf. Storage Retrieval Image Video Databases*, San Jose, CA, February 1995, pp. 36–46.
[46] D. Zhong, H. J. Zhang, and S.-F. Chang, "Clustering methods for video browsing and annotation," in *Proc. SPIE Storage Retrieval Image Video Databases IV*, San Jose, CA, February 1996, pp. 239–246.

Aditya Vailaya (A'00) received the B.Tech degree from the Indian Institute of Technology, Delhi, in 1994 and the M.S. and Ph.D. degrees from Michigan State University, East Lansing, in 1996 and 2000, respectively.

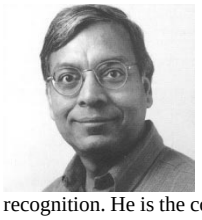
He joined Agilent Laboratories, Palo Alto, CA, in May 2000, where he is currently applying pattern recognition techniques for decision support in bioscience research. His research interests include pattern recognition and classification, machine learning, image and video databases, and image understanding.

Dr. Vailaya received the Best Student Paper Award from the IEEE International Conference on Image Processing in 1999.

Mário A. T. Figueiredo (S'87–M'95) received the E.E., M.S. and Ph.D. degrees in electrical and computer engineering, all from the Higher Institute of Technology [Instituto Superior Tecnico (IST)], Technical University of Lisbon, Lisbon, Portugal, in 1985, 1990, and 1994, respectively.

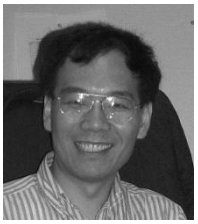
Since 1994, he has been an Assistant Professor with the Department of Electrical and Computer Engineering, IST. He is also a Researcher with the Communication Theory and Pattern Recognition Group, Institute of Telecommunications, Lisbon. In 1998, he held a visiting position with the Department of Computer Science and Engineering, Michigan State University, East Lansing. His scientific interests are in the fields of image analysis, computer vision, statistical pattern recognition, and information theory.

Dr. Figueiredo received the Portuguese IBM Scientific Prize in 1995.



Anil K. Jain (S'70–M'72–SM'86–F'91) is a University Distinguished Professor with the Department of Computer Science and Engineering, Michigan State University, Ann Arbor. He served as the Department Chair from 1995 to 1999. His research interests include statistical pattern recognition, Markov random fields, texture analysis, neural networks, document image analysis, fingerprint matching and 3-D object recognition. He is the co-author of *Algorithms for Clustering Data* (Englewood Cliffs, NJ: Prentice-Hall, 1988), edited the book *Real-Time Object Measurement and Classification* (Berlin, Germany: Springer-Verlag, 1988), and co-edited the books *Analysis and Interpretation of Range Images* (Berlin, Germany: Springer-Verlag, 1989), *Markov Random Fields* (New York: Academic, 1992), *Artificial Neural Networks and Pattern Recognition* (Amsterdam, The Netherlands: Elsevier, 1993), *3D Object Recognition* (Amsterdam, The Netherlands: Elsevier, 1993), and *BIOMETRICS: Personal Identification in Networked Society* (Norwell, MA: Kluwer, 1999).

Dr. Jain received the Best Paper Awards in 1987 and 1991 and certificates for outstanding contributions in 1976, 1979, 1992, and 1997, from the Pattern Recognition Society. He also received the 1996 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (1990–1994). He is a Fellow of the IAPR. He received a Fulbright Research Award in 1998.



Hong-Jiang Zhang (S'90–M'91–SM'97) received the B.S. degree from Zhengzhou University, China, in 1982 and the Ph.D degree from the Technical University of Denmark, Lyngby, in 1991, both in electrical engineering.

In 1999, he joined Microsoft Research China, Beijing, as a Senior Researcher/Research Manager. He was previously with Hewlett-Packard Labs, Palo Alto, CA, where he was a Research Manager, performing research and development in the areas of multimedia content retrieval and management technologies, intelligent image processing and video coding, and Internet media. Before joining Hewlett-Packard Labs Labs, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval, computer vision, and multimedia information systems. He was with the Massachusetts Institute of Technology Media Lab in 1994 as a Visiting Researcher. He has authored two books, about 100 papers and book chapters, and numerous special issues of professional journals in multimedia processing, content-based retrieval, and Internet media. He has served on committees of more than 40 international conferences. He was the Program Committee Co-Chair of the ACM Multimedia Conference in 1999. His interests are in the areas of video and image analysis, processing and retrieval, media compression and streaming, Internet multimedia, computer vision and their applications.

Dr. Zhang currently serves on the editorial boards of five international journals, including IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE MULTIMEDIA MAGAZINE.