SUBMITTED FOR PUBLICATION TO: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, JUI

A Probabilistic Similarity Framework for Content-Based Image Retrieval

Selim Aksoy, *Member, IEEE,* and Robert M. Haralick, *Fellow, IEEE*

Abstract

Previous research on image retrieval focused on using low-level features like color and texture with a geometric framework of distances for similarity. A challenging problem is to find a well-defined formulation that also fuses information from multiple features and similarity measures. We pose the retrieval problem in a probabilistic framework where the goal is to minimize the classification error in a setting of two classes: the relevance and irrelevance classes of the query. We propose solutions to different levels of the retrieval process within this framework. Feature extraction and normalization is done by maximizing class separability. Similarity is measured using likelihood of two images being similar or dissimilar. A key aspect of our framework is a two-level modeling of probability. The first level maps high-dimensional feature spaces to two-dimensional probability spaces using parametric density models for features. The second level uses combinations of classifiers trained in multiple probability spaces and corresponds to a modeling of "probability of probability" to compensate for errors in modeling probabilities in feature spaces. The Bayesian formulation also provides a unified framework for fusion of information from different features as well as support for relevance feedback. Extensive experiments on two ground truth databases show that the proposed framework performs significantly better than the geometric framework and two competing algorithms, and achieves almost perfect retrieval.

Index Terms

Image databases, image classification, retrieval models, similarity measures, probabilistic algorithms, sensor fusion, relevance feedback

S. Aksoy is with Insightful Corporation, 1700 Westlake Ave. N., Suite 500, Seattle, WA 98109. E-mail: saksoy@insightful.com.

R. Haralick is with the Computer Science Department, Graduate Center, City University of New York, 365 Fifth Avenue, New York, NY 10016. E-mail: haralick@gc.cuny.edu.

I. Introduction

CONTENT-BASED image retrieval (CBIR) has become one of the most popular research areas in computer vision. The retrieval process in a CBIR system can be summarized in three levels: pre-processing level (feature extraction and normalization), similarity level (similarity computation between query and database images), and post-processing level (iterative retrievals to improve the performance). Initial work on CBIR focused on using low-level features like color and texture for image representation [1], [2]. More recent approaches developed region-based query systems that compute features for individual regions found by segmentation methods like parametric clustering in the color and texture space [3], identifying the direction of changes in feature values [4], graph-based approaches [5], [6], and non-parametric clustering [7].

After each image (or region) is associated with a feature vector, the next step is to find similarities between images (or regions). Despite the enormous amount of work on developing features, similarity measures have not received significant attention. The most common approach has been to use the nearest neighbor rule with a distance measure to rank database images in ascending order of their distances to the query image, which is assumed to correspond to a descending order of similarity. Popular distance measures have been the Euclidean (L_2) [8], [9], city-block (L_1) [10], weighted Euclidean [11], [12], Cauchy [13], and Mahalanobis [14] distances. However, the nearest neighbor rule suffers from the fact that low-level features cannot always map visually similar images into nearby locations in the feature space and images that are quite irrelevant to the query image can be easily retrieved simply because they are close to it in the feature space.

Another important observation is that different features and different similarity measures perform differently for different types of images. Therefore, developing a framework

to combine features and similarity measures looks promising to improve the overall performance. Early approaches appended different feature vectors together and treated the result as a big global feature vector [9]. Other methods include letting the user weight different features [8], [15], taking linear or Boolean combinations of distances computed using different features [16], training neural networks in the high-dimensional feature space formed by all features [17], and boosting multiple classifiers that are trained on individual features [18]. Relevance feedback was also used to iteratively update the weights for individual features and distances [12], [19]. However, most of the weight assignment and updating has been done heuristically, and training classifiers and doing estimation in high-dimensional feature spaces suffered from the curse of dimensionality and required a lot of training data.

We developed likelihood-based similarity measures [20], [21] to overcome the limitations of distance-based approaches. Moghaddam and Pentland [22] also showed that maximum likelihood measures can perform much more effectively than the distance-based approaches in face recognition. Vailaya *et al.* [23] used Bayesian classifiers to hierarchically classify images into binary groups. However, this approach assumes that the defined classes (city, landscape, forest, mountain, sunset, etc.) form a partition of the database and each image is required to be a member of only one of these groups. When the database gets larger and becomes more complex, the number of classes to be defined increases, the limit being a class for each image. Vasconcelos and Lippman [24] proposed such an approach where each image in the database was assumed to be a class and the goal was to assign the query image to one of these classes. However, this method is only applicable to certain features that can be computed for small neighborhoods in an image to have enough data to estimate a feature distribution for each image separately. Despite the rapidly growing literature, most of the CBIR algorithms include unjustified heuristics and there is no well-defined formulation

neither to combine multiple features and similarity measures nor to tune the algorithms in terms of selecting models, choosing thresholds, etc.

II. Problem Definition

We pose the retrieval problem in a classification framework. The goal is to minimize the classification error in a setting of two classes: the relevance class and the irrelevance class of the query. Unlike other approaches where an ambiguity exists about the images that do not belong to any of the defined classes (city, forest, etc.) [23], or where there are as many classes as the number of images in the database [25], [24], the binary setting of the relevance and irrelevance classes unambiguously partitions any database given the query image because the classes are defined relative to the query image, not to the images in the database. Therefore, this setting does not require an initial partitioning of the database. Assuming that similar images have similar feature values and dissimilar images have relatively different feature values, we base similarity between two images in terms of feature differences. For example, two images of trees or two images of sunsets can be modeled by the relevance class if they have similar feature values (i.e. small feature differences). On the other hand, an image of a tree and a sunset image, or an image of a horse and a building image belong to the irrelevance class because of large differences in their feature values. Hence, the two-class modeling can intuitively describe pairwise similarities between different images.

Given the relevance class $\mathcal{A}$, the irrelevance class $\mathcal{B}$, the query image ξ_i and an image ξ_j from the database, the classification error for the image pair (ξ_i, ξ_j) is computed as

$$\begin{aligned} P(\text{error}) = & 0.5\, P((\xi_i, \xi_j) \text{ assigned to } \mathcal{B}, (\xi_i, \xi_j) \text{ belongs to } \mathcal{A}) + \\ & 0.5\, P((\xi_i, \xi_j) \text{ assigned to } \mathcal{A}, (\xi_i, \xi_j) \text{ belongs to } \mathcal{B}). \end{aligned} \tag{1}$$

Pattern recognition literature provides many choices for a classifier. Since the Bayes classifier gives the theoretical minimum classification error [26], it is the ideal choice for the classifier.

Since it uses posterior probabilities to make the decision, the posterior probabilities are the ideal features for classification. In the two-class problem, the discriminant function can be represented in the posterior ratio form

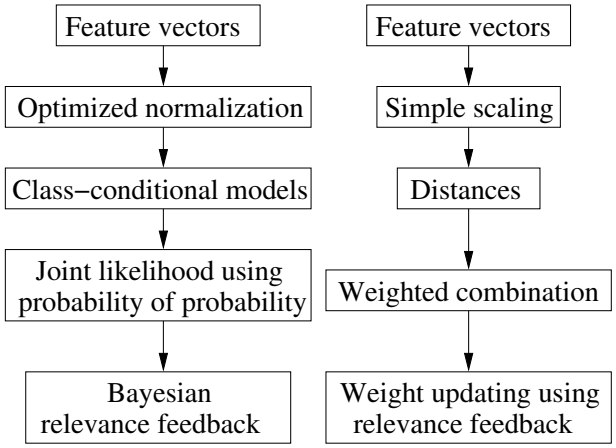
$$\Delta(\xi_i, \xi_j) = \frac{P(\mathcal{A}|(\xi_i, \xi_j))}{P(\mathcal{B}|(\xi_i, \xi_j))} = \frac{P((\xi_i, \xi_j)|\mathcal{A})P(\mathcal{A})}{P((\xi_i, \xi_j)|\mathcal{B})P(\mathcal{B})} \tag{2}$$

which gives the decision rule

$$\text{assign } (\xi_i, \xi_j) \text{ to } \begin{cases} \text{class } \mathcal{A} & \text{if } \Delta(\xi_i, \xi_j) > 1 \\ \text{class } \mathcal{B} & \text{if } \Delta(\xi_i, \xi_j) \leq 1. \end{cases} \tag{3}$$

Similarity can then be computed using posterior ratios in the probabilistic setting instead of distances in the feature space in the geometric setting.

After defining the decision rule, the problem becomes finding solutions to different levels of the retrieval process. First, feature extraction and normalization is done by maximizing class separability (pre-processing). Second, similarity is measured in terms of the likelihood of two images being similar or dissimilar, one being the query image and the other one being an image in the database. We propose a two-level modeling of probability to estimate the posterior probabilities in (2). In the first level, class-conditional probabilities for feature difference vectors are computed using simple parametric models. This can also be interpreted as a mapping from high-dimensional feature spaces to two-dimensional probability spaces. Then, classifiers are trained in these two-dimensional spaces instead of the high-dimensional feature spaces. Since these probabilities are only estimates of the true probabilities, the classifiers trained in the probability spaces implicitly perform a second level modeling of these probabilities to compensate for errors in modeling probabilities in the feature spaces. Furthermore, the Bayesian formulation provides a unified framework to combine models estimated for multiple feature spaces and multiple classifiers trained on these models. Finally,



(a) Probabilistic framework (b) Geometric framework

Fig. 1. Main steps of processing in the proposed probabilistic framework and the commonly used geometric framework.

relevance feedback in terms of user's labeling of retrieval results as relevant and irrelevant is incorporated into the Bayesian framework by automatically updating the posterior probability estimates (post-processing).

Retrieval processes in the proposed probabilistic framework and the commonly used geometric framework are summarized in Fig. 1. The rest of the paper is organized as follows. Section III describes feature normalization and a class separability-based criterion to select a normalization method for a given dataset. Section IV provides models to estimate the class-conditional probabilities. Section V compares operating in the probability space to operating in the feature space. Section VI describes a framework to combine multiple features and similarity models and to support user relevance feedback. Classification and retrieval experiments on ground truth databases are presented in Section VIII where the proposed probabilistic setting performs significantly better than the commonly used geometric framework and two competing algorithms. Conclusions are given in Section IX.

III. Feature Extraction and Normalization

The first step in the retrieval process is image representation. Low-level features have been the most popular choice because they can be easily computed from raw image data. Each image in our system is represented by multiple texture and color feature vectors like line-angle-ratio statistics [27], co-occurrence variances [27], Gabor features [10], moments features [28], Tamura features [8], and color histograms [29] in the HSV color space. Our main goal is to develop similarity models so only global low-level features are considered in this paper. However, all of the algorithms proposed here can be directly applicable to features computed for image regions.

Each component in these feature vectors represents a different quantity and most of these components have different ranges. Furthermore, popular distance measures like the Euclidean distance implicitly assign more weighting to features with large ranges than those with small ranges. Feature normalization is required to approximately equalize the ranges of these features and remove the bias in the computation of similarity. In most of the database retrieval literature, normalization methods were usually not mentioned or only the Gaussian assumption (linear scaling to unit variance) was used [10], [9], [12]. The Mahalanobis distance [26] also involves normalization in terms of the covariance matrix.

In [20], [21], we studied the effects of linear scaling to unit range, linear scaling to unit variance, normalization using the cumulative distribution function, rank normalization, and normalization by fitting distributions on retrieval performance in terms of precision and recall. Each feature component was independently normalized to the $[0, 1]$ range and the best normalization method was empirically chosen as the one that gave the largest precision. In the following, we use a class separability criterion as part of the classification framework.

Two classical approaches to linear data transformations, the principal components anal-

ysis and the discriminant analysis, use scatter matrices to find projections that efficiently represent or efficiently separate the data, respectively. Therefore, scatter information is an important measure of class separability [26]. Assume we have n data vectors $\mathbf{x_1}, \ldots, \mathbf{x_n} \in \mathbb{R}^{(q\times 1)}$ and m classes $\mathcal{C}_1, \ldots, \mathcal{C}_m$. Let $n_1, \ldots, n_m$ be the number of data vectors assigned to these m classes. The within-class scatter matrix $\mathbf{S_W} \in \mathbb{R}^{(q\times q)}$ measures the scatter of data vectors around their respective class means and is computed as

$$\mathbf{S_W} = \sum_{i=1}^{m} \sum_{\mathbf{x}\in\mathcal{C}_i} (\mathbf{x} - \boldsymbol{\mu_i})(\mathbf{x} - \boldsymbol{\mu_i})^T \tag{4}$$

where $\boldsymbol{\mu_i} = \frac{1}{n_i} \sum_{\mathbf{x}\in\mathcal{C}_i} \mathbf{x}$. On the other hand, the between-class scatter matrix $\mathbf{S_B} \in \mathbb{R}^{(q\times q)}$ is the scatter of the class means around the total mean and is computed as

$$\mathbf{S_B} = \sum_{i=1}^{m} n_i(\boldsymbol{\mu_i} - \boldsymbol{\mu_0})(\boldsymbol{\mu_i} - \boldsymbol{\mu_0})^T \tag{5}$$

where $\boldsymbol{\mu_0} = \frac{1}{n} \sum_{i=1}^{m} n_i \boldsymbol{\mu_i}$ is the total mean.

Possible measures for the "size" of a scatter matrix are the trace and the determinant. Both measures often give the same results but the latter is invariant to changes in the scale of the axes [26] so it is preferable over the former. It is also known that the eigenvalues of $\mathbf{S_W}^{-1}\mathbf{S_B}$ are invariant under non-singular linear transformations of the data. Therefore, an appealing criterion becomes

$$\varsigma = \log |\mathbf{S_W}^{-1}(\mathbf{S_W} + \mathbf{S_B})| \tag{6}$$

where $|\cdot|$ represents the determinant. ς gives a number which is large when the between-class scatter is large and the within-class scatter is small, and the normalization method that gives the largest ς is chosen as the best method. ($\mathbf{S_W}^{-1}\mathbf{S_B}$ cannot be used directly because $\mathbf{S_B}$ will be singular if either m or $n-m$ is less than or equal to the dimensionality of the feature space.)

IV. Similarity Measures

After computing and normalizing the feature vectors for all images in the database, the next step for retrieval is similarity computation. This section discusses probabilistic and geometric similarity measures.

A. Probabilistic Similarity Measures

As mentioned in Section II, the choice for the Bayes classifier comes from the fact that it minimizes the classification error given that we know the true class-conditional distributions and prior probabilities. However, these distributions are not exactly known in practice but can be estimated from training data.

Given two images with q-dimensional feature vectors $\mathbf{x}$ and $\mathbf{y}$, and their feature difference vector $\mathbf{d} = \mathbf{x} - \mathbf{y} \in \mathbb{R}^{(q\times 1)}$, the posterior ratio in (2) becomes the likelihood ratio

$$\Delta(\mathbf{d}) = \frac{p(\mathbf{d}|\mathcal{A})}{p(\mathbf{d}|\mathcal{B})} \tag{7}$$

under the equal priors assumption. The motivation for estimating $p(\mathbf{d}|\mathcal{A})$ and $p(\mathbf{d}|\mathcal{B})$ in terms of feature difference vectors comes from the assumption that similarity between images can be based on the closeness of their feature values, i.e. similar images have similar feature values (therefore, difference vectors have a small scatter) and dissimilar images have relatively different feature values (difference vectors have a larger scatter).

A.1 Multivariate Gaussian

The first model assumes that feature difference vectors for the relevance and irrelevance classes have multivariate Gaussian densities

$$p(\mathbf{d}|\mathcal{A}) = p(\mathbf{d}|\boldsymbol{\mu}_{\mathcal{A}}, \boldsymbol{\Sigma}_{\mathcal{A}}) = \frac{1}{(2\pi)^{q/2}|\boldsymbol{\Sigma}_{\mathcal{A}}|^{1/2}}\, e^{-(\mathbf{d}-\boldsymbol{\mu}_{\mathcal{A}})^T\boldsymbol{\Sigma}_{\mathcal{A}}^{-1}(\mathbf{d}-\boldsymbol{\mu}_{\mathcal{A}})/2} \tag{8}$$

$$p(\mathbf{d}|\mathcal{B}) = p(\mathbf{d}|\boldsymbol{\mu}_{\mathcal{B}}, \boldsymbol{\Sigma}_{\mathcal{B}}) = \frac{1}{(2\pi)^{q/2}|\boldsymbol{\Sigma}_{\mathcal{B}}|^{1/2}}\, e^{-(\mathbf{d}-\boldsymbol{\mu}_{\mathcal{B}})^T\boldsymbol{\Sigma}_{\mathcal{B}}^{-1}(\mathbf{d}-\boldsymbol{\mu}_{\mathcal{B}})/2} \tag{9}$$

respectively. Given training feature difference vectors $\mathbf{d_1}, \ldots, \mathbf{d_n} \in \mathbb{R}^{(q\times 1)}$, the maximum likelihood estimates (MLEs) for the sample means $\boldsymbol{\mu}_{\mathcal{A}}, \boldsymbol{\mu}_{\mathcal{B}}$ and sample covariance matrices $\boldsymbol{\Sigma}_{\mathcal{A}}, \boldsymbol{\Sigma}_{\mathcal{B}}$ are computed using the formulas

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{d_i} \qquad \text{and} \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d_i}-\hat{\boldsymbol{\mu}})(\mathbf{d_i}-\hat{\boldsymbol{\mu}})^T \tag{10}$$

(q is 20 for line-angle-ratio statistics and co-occurrence variances, 60 for Gabor features, 36 for moments features, 4 for Tamura features, and 64 for color histograms). To simplify the likelihood ratio in (7), we take its logarithm, eliminate some constants, and use

$$\Delta'(\mathbf{d}) = (\mathbf{d}-\boldsymbol{\mu}_{\mathcal{A}})^T\boldsymbol{\Sigma}_{\mathcal{A}}^{-1}(\mathbf{d}-\boldsymbol{\mu}_{\mathcal{A}}) - (\mathbf{d}-\boldsymbol{\mu}_{\mathcal{B}})^T\boldsymbol{\Sigma}_{\mathcal{B}}^{-1}(\mathbf{d}-\boldsymbol{\mu}_{\mathcal{B}}) \tag{11}$$

to rank the database images in ascending order of these values which corresponds to a descending order of similarity.

A.2 Independently Fitted Distributions

Another probability model can be constructed by treating feature components as independent, fitting parametric models using the Kolmogorov-Smirnov test statistic [30] as the goodness-of-fit criterion, and taking the product of these fitted marginal distributions to compute the joint distributions for relevance and irrelevance classes.

Given training vectors $\mathbf{d_i} = (\mathbf{d_{i1}}, \ldots, \mathbf{d_{iq}})^T \in \mathbb{R}^{(q\times 1)}, i = 1, \ldots, n$, we use symmetric[1] distributions like Gaussian $\frac{1}{\sqrt{2\pi}\sigma}e^{-(d-\mu)^2/2\sigma^2}$ with MLEs

$$\hat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n}\mathbf{d_{i}}_j \qquad \text{and} \qquad \hat{\sigma}_j^2 = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d_{i}}_j - \hat{\mu}_j)^2, \tag{12}$$

Double Exponential $\frac{1}{2\lambda}e^{-|d-\mu|/\lambda}$ with MLEs

$$\hat{\mu}_j = \text{median}(\mathbf{d_1}_j, \ldots, \mathbf{d_n}_j) \qquad \text{and} \qquad \hat{\lambda}_j = \frac{1}{n}\sum_{i=1}^{n}|\mathbf{d_{i}}_j - \hat{\mu}_j|, \tag{13}$$

[1] Similarity relationship is symmetric. I.e. if image ξ_i is similar to image ξ_j, image ξ_j is equally similar to image ξ_i. Therefore, the feature difference values are symmetric around zero.

or Logistic $\frac{e^{-(d-\mu)/\tau}}{\tau(1+e^{-(d-\mu)/\tau})^2}$ with method of moments estimators (MOM)

$$\hat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{d}_{\boldsymbol{i}j} \qquad \text{and} \qquad \hat{\tau}_j^2 = \frac{3}{\pi^2}\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{d}_{\boldsymbol{i}j}^2 - \hat{\mu}_j^2\right) \tag{14}$$

to estimate the marginal distributions for components $j = 1, \ldots, q$. After computing the joint distributions by multiplying the corresponding marginals, the likelihood ratio can be simplified by taking logarithms and eliminating constants as

$$\Delta'(\mathbf{d}) = \frac{1}{2}\sum_{t\in\mathcal{T}_\mathcal{A}} \frac{(\mathbf{d}_t - \mu_{\mathcal{A}t})^2}{\sigma_{\mathcal{A}t}^2} + \sum_{u\in\mathcal{U}_\mathcal{A}} \frac{|\mathbf{d}_u - \mu_{\mathcal{A}u}|}{\lambda_{\mathcal{A}u}} + \sum_{v\in\mathcal{V}_\mathcal{A}} \left[\frac{(\mathbf{d}_v - \mu_{\mathcal{A}v})}{\tau_{\mathcal{A}v}} + 2\log(1 + e^{-(\mathbf{d}_v - \mu_{\mathcal{A}v})/\tau_{\mathcal{A}v}})\right]$$
$$- \frac{1}{2}\sum_{t\in\mathcal{T}_\mathcal{B}} \frac{(\mathbf{d}_t - \mu_{\mathcal{B}t})^2}{\sigma_{\mathcal{B}t}^2} - \sum_{u\in\mathcal{U}_\mathcal{B}} \frac{|\mathbf{d}_u - \mu_{\mathcal{B}u}|}{\lambda_{\mathcal{B}u}} - \sum_{v\in\mathcal{V}_\mathcal{B}} \left[\frac{(\mathbf{d}_v - \mu_{\mathcal{B}v})}{\tau_{\mathcal{B}v}} + 2\log(1 + e^{-(\mathbf{d}_v - \mu_{\mathcal{B}v})/\tau_{\mathcal{B}v}})\right] \tag{15}$$

where $\mathcal{T}_c, \mathcal{U}_c, \mathcal{V}_c, c = \mathcal{A}$ or $\mathcal{B}$ are the sets of indices for components with best fits as Gaussians, Double Exponentials and Logistics for the relevance and irrelevance classes, respectively.

The assumption of independent features is a very strong assumption because features are usually correlated. Linear transformations, like whitening that makes the components of a vector uncorrelated or independent components analysis that tries to find the axes on which data is independent in as many statistical orders as possible, can be used for pre-processing before fitting distributions to each component. We studied both models but often had over-fitting problems for high-dimensional feature vectors. This is another example that shows the problems that can arise when one is operating in the feature space and will be further discussed in Section V.

A.3 Mixtures of Gaussians

Yet another set of probability models includes mixture densities. We use Gaussian mixtures with the Expectation-Maximization algorithm [31] to obtain analytical solutions for parameter estimates and the Minimum Description Length (MDL) principle [32] to find the covariance structure and the number of components in the mixture model.

Table I summarizes the MLEs for the parameters of a mixture of m Gaussians

$$p(\mathbf{d}|\mathbf{\Theta}) = \sum_{j=1}^{m} \alpha_j p_j(\mathbf{d}|\boldsymbol{\theta_j}), \quad \alpha_j \geq 0, \quad \sum_{j=1}^{m} \alpha_j = 1 \tag{16}$$

where $\boldsymbol{\theta_j} = (\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})$, $\mathbf{\Theta} = (\alpha_1, \ldots, \alpha_m, \boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_m})$, and

$$p_j(\mathbf{d}|\boldsymbol{\theta_j}) = \frac{1}{(2\pi)^{q/2}|\boldsymbol{\Sigma_j}|^{1/2}} e^{-(\mathbf{d}-\boldsymbol{\mu_j})^T \boldsymbol{\Sigma_j}^{-1}(\mathbf{d}-\boldsymbol{\mu_j})/2}, \quad j = 1, \ldots, m. \tag{17}$$

The best m can be found as

$$m^* = \arg\min_m \left[\frac{1}{2}\kappa_{\mathcal{M}} \log n - \sum_{i=1}^{n} \log \left(\sum_{j=1}^{m} \alpha_j p_j(\mathbf{d_i}|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}) \right) \right] \tag{18}$$

where $\kappa_{\mathcal{M}}$ is the summation of the number of free parameters for mixture weights $\{\alpha_j\}_{j=1}^m$, means $\{\boldsymbol{\mu_j}\}_{j=1}^m$ and covariance matrices $\{\boldsymbol{\Sigma_j}\}_{j=1}^m$. MDL can also be used to choose the most appropriate covariance model to a given data. Since we are using both multivariate Gaussian and Gaussian mixture models to estimate the distribution of the same training data, we want these two models to have equal number of parameters both for a fair comparison and also to avoid over-fitting for the mixture model. Equating the number of free parameters in the multivariate Gaussian model ($q+q(q+1)/2$) to the number of free parameters in the mixture model with dimensionality reduced to k using principal components analysis gives

$$k = \begin{cases} \min\{\frac{-2m+2+3q+q^2}{4m}, q\} & \text{if } \boldsymbol{\Sigma_j} \text{ is different and diagonal } (\sigma_j^2\mathbf{I}) \\ -\frac{1}{2} - m + \frac{1}{2}\sqrt{9 - 4m + 4m^2 + 12q + 4q^2} & \text{if } \boldsymbol{\Sigma_j} \text{ is same and full } (\boldsymbol{\Sigma}) \\ \frac{-3m+\sqrt{m^2+8m+12mq+4mq^2}}{2m} & \text{if } \boldsymbol{\Sigma_j} \text{ is different and full.} \end{cases} \tag{19}$$

After estimating the mixtures for the relevance and irrelevance classes,

$$\Delta'(\mathbf{d}) = \log \sum_{j=1}^{m_{\mathcal{B}}} \alpha_{\mathcal{B}j} p_{\mathcal{B}j}(\mathbf{d}|\boldsymbol{\mu_{\mathcal{B}j}}, \boldsymbol{\Sigma_{\mathcal{B}j}}) - \log \sum_{j=1}^{m_{\mathcal{A}}} \alpha_{\mathcal{A}j} p_{\mathcal{A}j}(\mathbf{d}|\boldsymbol{\mu_{\mathcal{A}j}}, \boldsymbol{\Sigma_{\mathcal{A}j}}) \tag{20}$$

can be used to rank database images where $m_{\mathcal{A}}$ and $m_{\mathcal{B}}$ are the number of components in the mixtures for the relevance and irrelevance classes, respectively.

TABLE I

MAXIMUM LIKELIHOOD ESTIMATES AND THE NUMBER OF FREE PARAMETERS FOR A MIXTURE OF m GAUSSIANS $p(\mathbf{d}|\boldsymbol{\Theta}) = \sum_{j=1}^{m} \alpha_j p_j(\mathbf{d}|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})$ FOR THE SAMPLE $\mathbf{d_1}, \ldots, \mathbf{d_n} \in \mathbb{R}^{(q\times 1)}$ WHERE $\boldsymbol{\Theta} = (\alpha_1, \ldots, \alpha_m, \boldsymbol{\mu_1}, \ldots, \boldsymbol{\mu_m}, \boldsymbol{\Sigma_1}, \ldots, \boldsymbol{\Sigma_m})$.

Variable	Estimate (M-step)	# of free parameters
$p(j\|\mathbf{d_i}, \boldsymbol{\Theta})$	$\frac{\alpha_j p_j(\mathbf{d_i}\|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})}{\sum_{k=1}^{m} \alpha_k p_k(\mathbf{d_i}\|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})}$	
α_j	$\frac{\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})}{n}$	$(m-1)$
$\boldsymbol{\mu_j}$	$\frac{\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})\mathbf{d_i}}{\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})}$	mq
$\boldsymbol{\Sigma_j} = \sigma^2\mathbf{I}$	$\sigma^2 = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})\|\mathbf{d_i} - \boldsymbol{\mu_j}\|^2}{nq}$	1
$\boldsymbol{\Sigma_j} = \sigma_j^2\mathbf{I}$	$\sigma_j^2 = \frac{\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})\|\mathbf{d_i} - \boldsymbol{\mu_j}\|^2}{q\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})}$	m
$\boldsymbol{\Sigma_j} = \mathrm{diag}(\{\sigma_{jk}^2\}_{k=1}^{q})$	$\sigma_{jk}^2 = \frac{\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})(\mathbf{d}_{ik} - \boldsymbol{\mu}_{jk})^2}{\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})}$	mq
$\boldsymbol{\Sigma_j} = \boldsymbol{\Sigma}$	$\boldsymbol{\Sigma} = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})(\mathbf{d_i} - \boldsymbol{\mu_j})(\mathbf{d_i} - \boldsymbol{\mu_j})^T}{n}$	$\frac{q(q+1)}{2}$
$\boldsymbol{\Sigma_j}$, full	$\boldsymbol{\Sigma_j} = \frac{\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})(\mathbf{d_i} - \boldsymbol{\mu_j})(\mathbf{d_i} - \boldsymbol{\mu_j})^T}{\sum_{i=1}^{n} p(j\|\mathbf{d_i}, \boldsymbol{\Theta})}$	$m\frac{q(q+1)}{2}$

B. Geometric Similarity Measures

The most commonly used similarity measure in image retrieval has been the nearest neighbor rule with the Minkowsky L_p metric which retrieves the nearest neighbors of the query feature vector as the most relevant ones to the query. Commonly used forms of the L_p metric are the city-block (L_1) distance and the Euclidean (L_2) distance. Note that the L_1 metric implicitly assumes that the components of the feature difference vector are i.i.d. Double Exponentials and the L_2 metric implicitly assumes that they are i.i.d. Gaussians. Within our classification framework we use a linear classifier to choose the best p value for the L_p metric [21]. Given training sets of feature vector pairs for the relevance and irrelevance classes, distances between these pairs are computed and their histograms are used to select a threshold for classification. This corresponds to a likelihood ratio test where the class-conditional densities are estimated by the distance histograms. The p value that gives the smallest classification error is used as the order of the L_p metric.

V. Feature Space vs. Probability Space

Sample size is very important in building classifiers. Duin [33] discussed the effects of the curse of dimensionality and sample size on classification error. Although it was traditionally thought that it is necessary to fill the feature space with more objects than its dimensionality to obtain a classifier that can generalize well, he argued that it is possible to build reliable classifiers in very small sample size problems. He used kernel mapping [33] and prototype objects [34] to map the high-dimensional feature space to low-dimensional spaces and showed that it was possible to build reliable classifiers in these new spaces even though the feature space was sparse.

As discussed in Section II, the proposed probabilistic setting can also be interpreted as a mapping from the high-dimensional feature space to a two-dimensional probability space. Therefore, classification can be done either in the feature space using the feature difference vector $\mathbf{d}$, or in the probability space using the class-conditional probabilities $p(\mathbf{d}|\mathcal{A})$ and $p(\mathbf{d}|\mathcal{B})$ as new features. The class-conditional probabilities computed using parametric density models in the high-dimensional feature space are only estimates of the true probabilities and include uncertainty themselves (because of imperfect density modeling, quantization, dimensionality, etc.). However, classifiers trained in the two-dimensional space of class-conditional probabilities impose a second level modeling of probability, i.e. "probability of probability", to compensate for errors in modeling probabilities in the feature space. This two-level modeling is illustrated with one-dimensional synthetic data in Fig. 2. Classification error is first computed in the original signal space using a linear Gaussian classifier. Then, this error is compared to the error computed using a new classifier in the space formed by the class-conditional probabilities. Even though the final probability of error was still higher than the true Bayes error, it was always smaller in the probability space than in the original

signal space in experiments done using different sample sizes.

The pattern recognition literature provides many choices for a classifier. We use Gaussian linear, Gaussian quadratic, Logistic linear, scaled nearest mean, nearest neighbor, Parzen window, decision tree, and feed-forward neural network classifiers [35], [36]. In a system with I feature representations (feature vectors), J probability models and K classifiers, there are $I \times J \times K$ possible configurations for classification in the probability space and $I \times K$ possible configurations for classification in the feature space as summarized in Figs. 3 and 4.

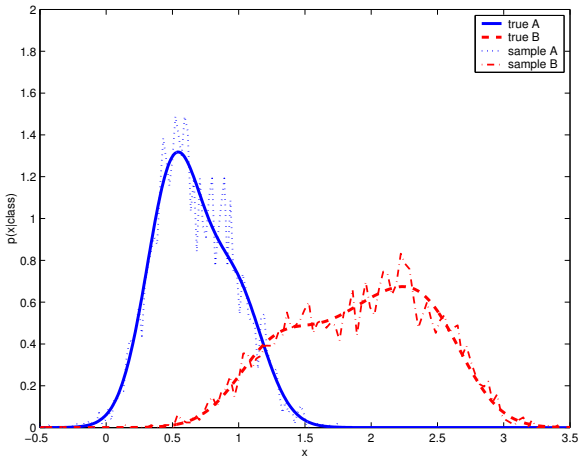
VI. Feature and Similarity Combination

Although most of the classifiers may have similar error rates, sets of image pairs misclassified by different classifiers do not necessarily overlap. Classification performance can be further improved by not relying on a single decision but rather by combining the decisions made by the individual classifiers. There has been a lot of research on classifier combination in the handwriting and speech recognition areas. Popular approaches include majority voting [37], class ranking [38], weighted combination of classifiers [39], and hierarchical multiple classifiers [40]. However, none of these approaches are applicable when the image retrieval problem is set as ranking images according to their distances to the query image in the feature space.

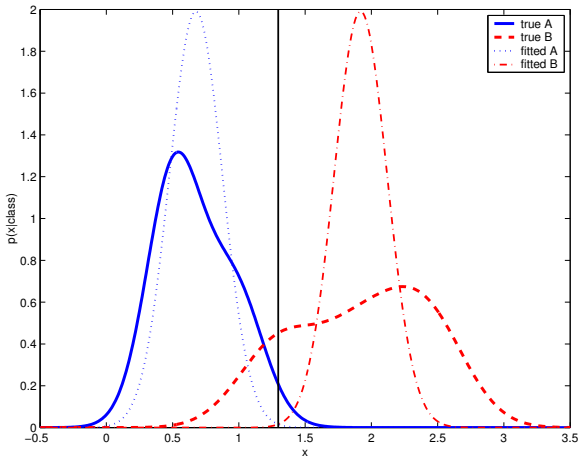
The proposed Bayesian framework provides a natural way to combine multiple measurements on images. Assume that n classifiers with measurement vectors $\mathbf{x_1}, \ldots, \mathbf{x_n}$ are available in our two-class setting. The Bayesian classifier makes the decision as

$$\text{assign } (\xi_i, \xi_j) \text{ to } \arg \max_{c \in \{\mathcal{A}, \mathcal{B}\}} p(c|\mathbf{x_1}, \ldots, \mathbf{x_n}). \tag{21}$$

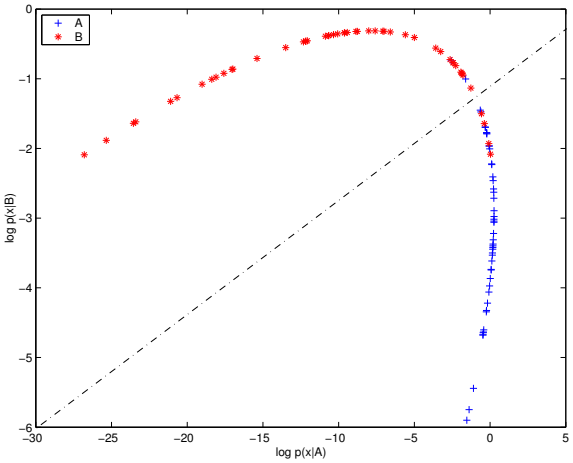
Computing the joint posterior probability $p(c|\mathbf{x_1}, \ldots, \mathbf{x_n})$ may become difficult in a practical situation with limited training data. Using the equal priors assumption and some approxi-



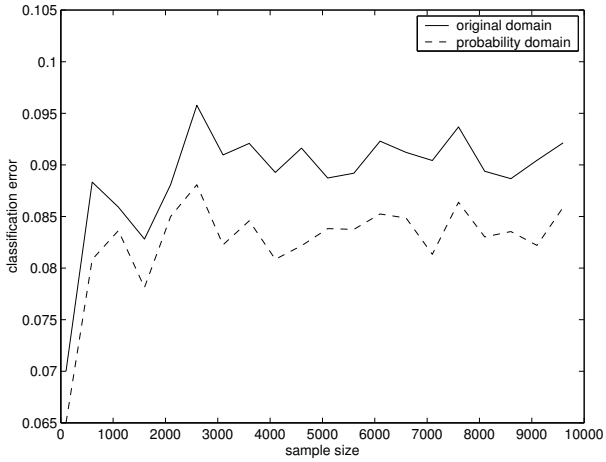
(a) True distributions and 2,500 samples for each of relevance (blue solid) and irrelevance (red dashed) classes (Bayes error, $P_e = 0.0828$)



(b) True and estimated distributions for relevance (blue solid) and irrelevance (red dashed) classes for a linear Gaussian classifier (black dash-dot) ($P_e = 0.0919$)



(c) A linear Gaussian classifier (black dash-dot) trained in the log-probability space for relevance (blue plus) and irrelevance (red star) classes ($P_e = 0.0879$)



(d) Classification error (y-axis) vs. sample size (x-axis) for linear Gaussian classifiers trained in signal (solid) and log-probability (dashed) spaces

Fig. 2. Classification in signal and probability spaces for synthetic data generated using mixtures of three univariate Gaussians for both the relevance and irrelevance classes. A linear Gaussian classifier is used for classification in both spaces. Classification errors (P_e) for true or estimated distributions are given in parentheses. Two-level probability modeling always gave a smaller error.

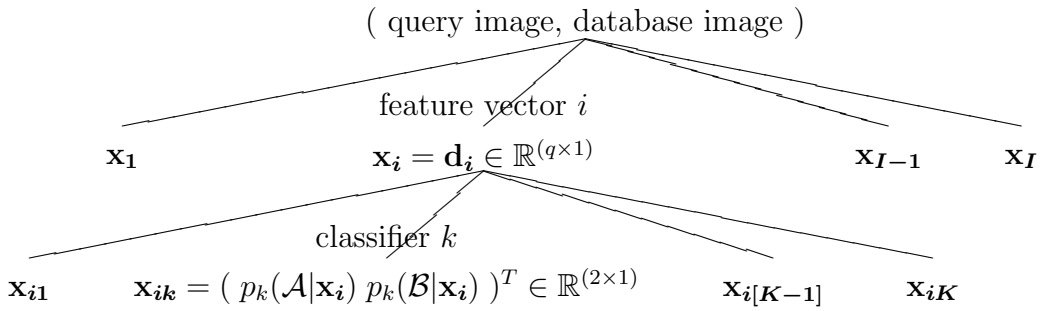


Fig. 3. Levels of classification in the feature space for a system with I feature representations and K classifiers. $\mathbf{x}$ represents measurements and $\mathbf{d}$ represents feature difference vectors. For each measurement $\mathbf{x_i}, 1 \leq i \leq I$, in the feature vector level, K classifiers output the posterior probabilities in $\mathbf{x_{ik}}, 1 \leq k \leq K$.

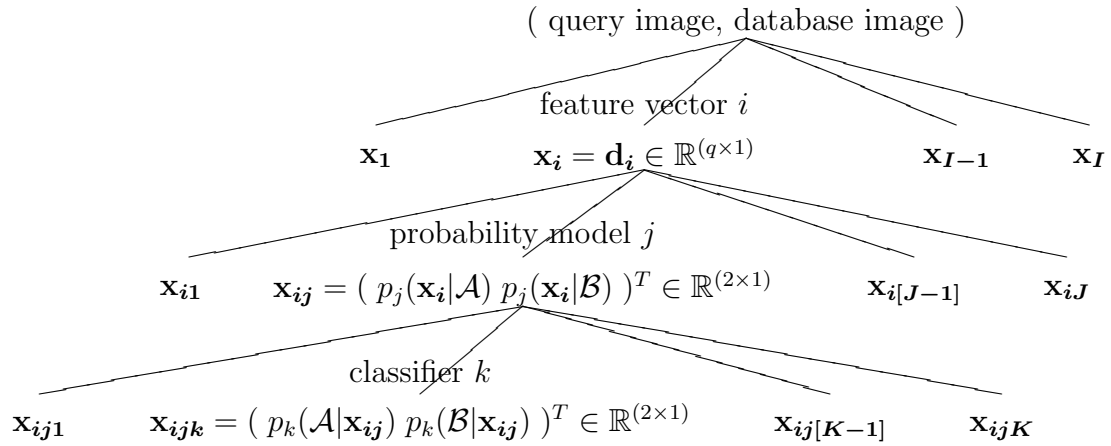


Fig. 4. Levels of classification in the probability space for a system with I feature representations, J probability models and K classifiers. $\mathbf{x}$ represents measurements and $\mathbf{d}$ represents feature difference vectors. In the feature vector level there are I measurements $\mathbf{x_i}, 1 \leq i \leq I$. Then, J probability models estimate class-conditional probabilities and map each $\mathbf{x_i}$ to two-dimensional spaces of measurements $\mathbf{x_{ij}}, 1 \leq j \leq J$. Finally, for each $\mathbf{x_{ij}}$, K classifiers output the posterior probabilities in $\mathbf{x_{ijk}}, 1 \leq k \leq K$.

mations [37], the decision rule in (21) can be simplified as follows:

- Product rule: Assuming that the measurements $\mathbf{x_1}, \ldots, \mathbf{x_n}$ are conditionally statistically independent given the class, the decision rule becomes

$$\text{assign } (\xi_i, \xi_j) \text{ to } \arg \max_{c\in\{\mathcal{A},\mathcal{B}\}} \prod_{i=1}^{n} p(c|\mathbf{x_i}). \tag{22}$$

  The conditional independence assumption may not always hold but it gives a practical approximation and the errors caused by this assumption will not be too severe if different feature vectors and different classifiers are used in the combination [41].

- Sum rule: Assuming that the posterior probabilities from individual classifiers will not deviate dramatically from the corresponding prior probabilities, the decision rule becomes

$$\text{assign } (\xi_i, \xi_j) \text{ to } \arg \max_{c \in \{\mathcal{A}, \mathcal{B}\}} \sum_{i=1}^{n} p(c|\mathbf{x_i}). \tag{23}$$

  This assumption may be unrealistic in some cases but has a low sensitivity to estimation errors [37].

- Max rule: Approximating the sum in (23) by the maximum of the posterior probabilities gives the decision rule

$$\text{assign } (\xi_i, \xi_j) \text{ to } \arg \max_{c \in \{\mathcal{A}, \mathcal{B}\}} \max_{i=1}^{n} p(c|\mathbf{x_i}). \tag{24}$$

- Min rule: Approximating the product in (22) by the minimum of the posterior probabilities gives the decision rule

$$\text{assign } (\xi_i, \xi_j) \text{ to } \arg \max_{c \in \{\mathcal{A}, \mathcal{B}\}} \min_{i=1}^{n} p(c|\mathbf{x_i}). \tag{25}$$

- Median rule: Using the fact that median is a robust estimate of the mean, approximating the sum in (23) by the median of the posterior probabilities gives the decision rule

$$\text{assign } (\xi_i, \xi_j) \text{ to } \arg \max_{c \in \{\mathcal{A}, \mathcal{B}\}} \operatorname*{median}_{i=1}^{n} p(c|\mathbf{x_i}). \tag{26}$$

- Majority vote rule: If we set each classifier to make a binary decision and vote for one of the classes, the decision rule becomes

$$\text{assign } (\xi_i, \xi_j) \text{ to } \arg \max_{c \in \{\mathcal{A}, \mathcal{B}\}} \#\{i \,|\, p(c|\mathbf{x_i}) > 0.5, i = 1, \ldots, n\}. \tag{27}$$

All of these combination methods are based on the conditional independence assumption. Furthermore, individual classifiers should not be strongly correlated in their misclassification, i.e. they should not agree with each other when they misclassify a sample, or at least they

should not assign the same incorrect class to a particular sample. This requirement can be satisfied to a certain extent by using different feature vectors and different classifiers [37], [41]. Since each possible combination of feature vectors, probability models and classifiers gives a set of posterior probabilities (the final level in Fig. 4), the classifier combination methods listed above can be directly used to compute the posterior ratio in (2) to arrive at a final decision about the similarity between images.

VII. Relevance Feedback

The Bayesian framework can also be extended to support the case when positive and negative feedback from the user is available. Given the original query, the initial search is done by computing the feature difference vectors between the query image and all images in the database, and then ranking images according to the posterior ratios $\Delta = \frac{p(\mathcal{A}|\xi^{(0)})}{p(\mathcal{B}|\xi^{(0)})}$ where $\xi^{(0)}$ represents the measurements based on the initial query image.

A. Positive Feedback

When the user labels an image as relevant, new feature difference vectors between the labeled image and all images in the database are computed and the images are re-ranked according to the updated posterior ratios $\Delta = \frac{p(\mathcal{A}|\xi^{(0)},\xi_+^{(1)})}{p(\mathcal{B}|\xi^{(0)},\xi_+^{(1)})} = \frac{p(\xi_+^{(1)}|\mathcal{A})p(\mathcal{A}|\xi^{(0)})}{p(\xi_+^{(1)}|\mathcal{B})p(\mathcal{B}|\xi^{(0)})}$ where $\xi_+^{(1)}$ represents the new measurements based on the first positive feedback image. Given a sequence of n images labeled as relevant, the updated posteriors are incrementally computed using the conditional independence assumption as

$$p(\mathcal{A}|\xi^{(0)},\xi_+^{(1)},\ldots,\xi_+^{(n)}) \propto p(\xi_+^{(n)}|\mathcal{A})p(\mathcal{A}|\xi^{(0)},\xi_+^{(1)},\ldots,\xi_+^{(n-1)}) \tag{28}$$

$$p(\mathcal{B}|\xi^{(0)},\xi_+^{(1)},\ldots,\xi_+^{(n)}) \propto p(\xi_+^{(n)}|\mathcal{B})p(\mathcal{B}|\xi^{(0)},\xi_+^{(1)},\ldots,\xi_+^{(n-1)}) \tag{29}$$

where $\xi_+^{(n)}$ represents the measurements based on the n'th positive feedback image.

B. Negative Feedback

When the user labels an image as irrelevant, search proceeds by computing new feature difference vectors as above but the posteriors are updated differently. The strength of the evidence of two images being relevant is a negative evidence that they are irrelevant. Therefore, the likelihood of a database image being relevant to a negative example also represents its likelihood of being irrelevant to the user's desired image. Given an additional sequence of m images labeled as irrelevant, the posteriors are updated as

$$p(\mathcal{A}|\xi^{(0)},\xi_+^{(1)},\ldots,\xi_+^{(n)},\xi_-^{(1)},\ldots,\xi_-^{(m)}) \propto p(\xi_-^{(m)}|\mathcal{B})p(\mathcal{A}|\xi^{(0)},\xi_+^{(1)},\ldots,\xi_+^{(n)},\xi_-^{(1)},\ldots,\xi_-^{(m-1)}) \quad (30)$$

$$p(\mathcal{B}|\xi^{(0)},\xi_+^{(1)},\ldots,\xi_+^{(n)},\xi_-^{(1)},\ldots,\xi_-^{(m)}) \propto p(\xi_-^{(m)}|\mathcal{A})p(\mathcal{B}|\xi^{(0)},\xi_+^{(1)},\ldots,\xi_+^{(n)},\xi_-^{(1)},\ldots,\xi_-^{(m-1)}) \quad (31)$$

where $\xi_-^{(m)}$ represents the measurements based on the m'th negative feedback image.

Vasconcelos and Lippman [42] and Cox *et al.* [25] proposed similar feedback algorithms but the former used only one kind of feature vector and the latter used only positive feedback with feature vector combination performed as a weighted sum of L_1 distances. Furthermore, the former used as many classes as there are images in the database and having too many classes caused estimation problems for the likelihood based on negative examples.

VIII. Experiments

The proposed classification framework was evaluated using two ground truth databases. The first database contained 736 texture images from the MIT Media Laboratory's VisTex Database with a ground truth of 46 categories with 16 images in each category. Smaller parts of this dataset were used by many researchers (e.g. [12]). The second database came from the COREL Photo Stock Library with a total of 1,575 images divided into 18 categories including animals, nature scenes, buildings, airplanes, cars, etc. Approximately one-third of all images were used for training and the remaining two-thirds were used for testing by finding random

pairings for both relevance and irrelevance classes. We used classification error (defined in (1)), precision (defined as the percentage of retrieved images that are actually relevant) and recall (defined as the percentage of relevant images that are retrieved) as quantitative evaluation criteria. (Databases and experiments are described in detail in [35].)

A. Classification Performance

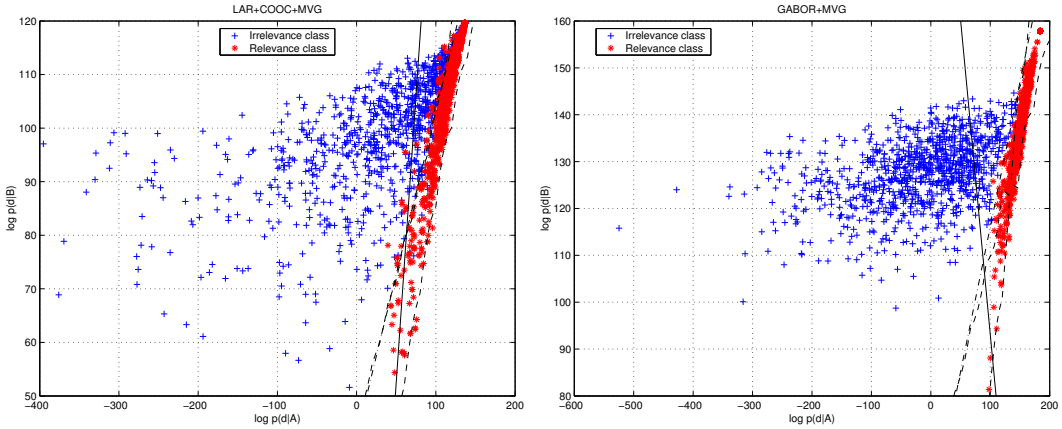
The first set of experiments was done to find the best performing normalization method. The methods in Section III were used to normalize the components of all feature vectors with class separability computed for each case. In these experiments each ground truth group was considered a class for a particular database. Even though there was no single best method, normalization after fitting distributions (done by fitting Gaussian, Lognormal, Exponential and Gamma densities to random samples from feature difference distributions, and scaling and truncating the features at the 99% cut-off values of the fitted distributions so that each feature component has the same range [21]) was usually among the best. These classification results were also consistent with the retrieval results using different normalization methods with individual feature vectors. Therefore, class separability appears to be an effective measure for choosing the normalization method that gives the best retrieval performance and this fact strengthens our motivation for a classification-based framework for image retrieval.

After all feature vectors were normalized, we did experiments to evaluate performances of using classifiers trained in high-dimensional feature spaces vs. ones trained in two-dimensional probability spaces. Example plots of the class-conditional log-probabilities for the relevance and irrelevance classes and the decision boundaries for some of the classifiers are shown in Fig. 5. These plots correspond to the measurements denoted by $\mathbf{x}_{ij}$ in Fig. 4. We can see that these mappings to the two-dimensional probability spaces result in a good separation of the relevance and irrelevance classes and this separation can be easily captured by simple

linear or quadratic classifiers. Classification results using testing data are summarized in the top-left sections of Tables II and III and in Table IV. Gabor and color histograms performed better than other feature vectors. Simple classifiers (like logistic linear or Gaussian quadratic classifiers) trained in probability spaces performed much better than the non-linear classifiers (like Parzen, decision tree and neural network classifiers) in the feature spaces. This is a very useful result because it allows us to do effective classification by training only simple classifiers in the probability space. (These results also agree with those of Duin [33].)

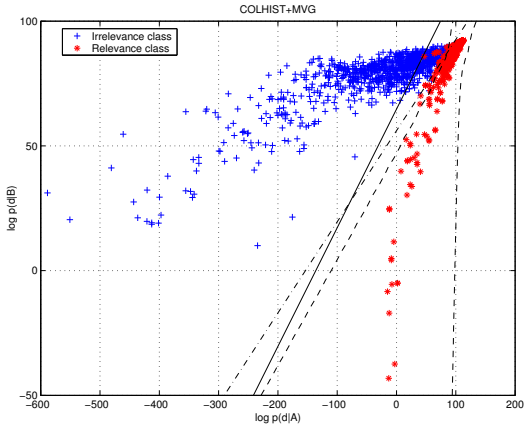
Using mixtures of Gaussians did not give an improvement over the single multivariate Gaussian case. This was because of the difficulties in estimating multivariate distributions in high-dimensional spaces from small amounts of data where one component usually dominated the others. The multivariate Gaussian model also performed better than the independently fitted distributions model because of its handling of the correlations between features. Its significant performance shows that simple models are worth trying before using any of the more complex models because they are often quite effective, do not require extra effort to tune in too many parameters, and do not suffer from the local extrema and convergence problems that may exist in the estimation of more complex models.

As the final set of classification experiments, we evaluated the performances of combinations of classifiers trained on multiple probability spaces corresponding to multiple feature vectors and probability models. Results are summarized in Tables II and III. Combining outputs of a particular classifier trained on multiple probability spaces corresponding to different feature vectors performed better than the cases without combination or when outputs of different classifiers trained on the same probability space were used (there is a higher risk of violating the conditional independence assumption in the latter case). The most successful combination rule was the product rule with logistic linear or Gaussian quadratic classifiers.

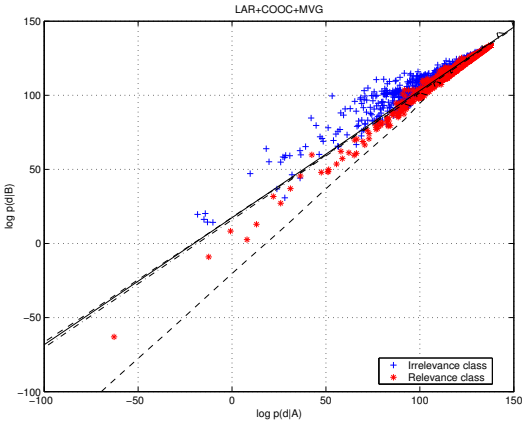


(a) VisTex: LC + MVG

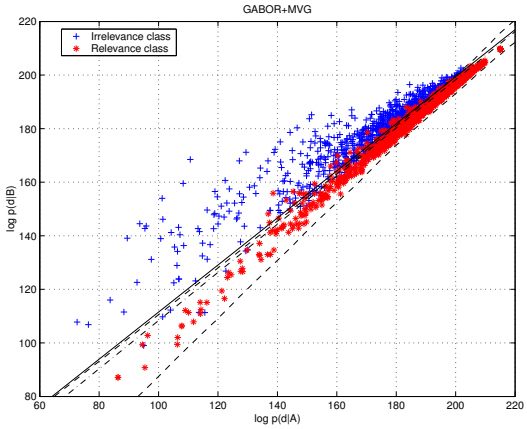
(b) VisTex: G + MVG



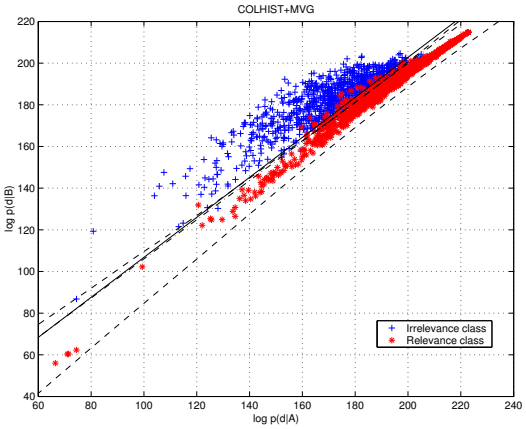
(c) VisTex: CH + MVG



(d) COREL: LC + MVG



(e) COREL: G + MVG



(f) COREL: CH + MVG

Fig. 5. Class-conditional log-probabilities for different feature vectors (LC: line-angle-ratio statistics and co-occurrence variances, G: Gabor, M: moments, T: Tamura, CH: color histograms). Multivariate Gaussian model (MVG) was used to map features to probability spaces. X-axis shows log-probabilities for the relevance class and y-axis shows log-probabilities for the irrelevance class. Red star symbols represent training data for the relevance class and blue plus symbols represent training data for the irrelevance class. Solid line is the Gaussian linear classifier, dashed line is the Gaussian quadratic classifier, and dash-dot line is the logistic linear classifier.

TABLE II

CLASSIFICATION IN THE PROBABILITY SPACE FOR THE VISTEX DATABASE. EACH NUMBER CORRESPONDS TO THE PERCENT CLASSIFICATION ERROR FOR THE TESTING DATASET BASED ON THE MEASUREMENTS DENOTED BY $\mathbf{x}_{ijk}$ IN FIG. 4. TOP-LEFT SECTION SHOWS CLASSIFIERS TRAINED ON INDIVIDUAL FEATURE VECTORS MAPPED USING THE MULTIVARIATE GAUSSIAN MODEL. TOP-RIGHT SECTION SHOWS COMBINING A PARTICULAR CLASSIFIER FOR ALL FEATURE VECTORS. BOTTOM-LEFT SECTION SHOWS COMBINING ALL CLASSIFIERS FOR A PARTICULAR FEATURE VECTOR. BOTTOM-RIGHT SECTION SHOWS COMBINING ALL CLASSIFIERS FOR ALL FEATURE VECTORS. THE BEST MODEL IS UNDERLINED FOR EACH CASE.

	Classifiers	Feature vectors					Combination rules					
		LC	G	M	T	CH	Prod	Sum	Max	Min	Med	Maj
Basic classifiers	Gaussian linear	24.78	9.45	24.18	21.47	14.22	5.66	8.58	5.50	5.50	11.57	9.73
	Gaussian quadratic	15.76	9.76	15.90	19.46	6.78	5.25	3.41	10.11	10.11	5.38	4.78
	Logistic linear	15.18	6.02	15.25	16.49	5.84	3.11	3.41	5.61	5.61	4.33	4.59
	Scaled nearest mean	29.39	12.33	29.53	21.35	16.97	8.62	8.80	11.86	11.86	12.14	10.91
	Nearest neighbor	22.45	9.25	20.85	22.87	8.51	5.98	6.68	6.67	6.67	7.42	9.20
	Parzen	17.00	8.78	16.03	16.76	6.42	7.73	4.34	9.26	8.92	4.83	5.42
	Binary decision tree	22.86	9.14	21.01	23.53	9.22	5.97	6.33	9.05	9.05	7.00	9.38
	Neural network	15.63	6.91	15.50	16.29	5.70	3.32	3.43	4.92	4.92	4.52	4.90
Comb. rules	Product	16.80	6.08	16.11	18.23	6.64	2.88					
	Sum	17.16	6.13	16.35	18.25	6.53		4.53				
	Max	16.29	7.50	15.74	17.88	6.59			8.87			
	Min	16.29	7.50	15.74	17.88	6.59				8.87		
	Median	19.20	6.11	17.04	18.91	6.73					6.83	
	Majority vote	17.78	7.60	16.55	17.65	6.14						6.65

TABLE III

CLASSIFICATION IN THE PROBABILITY SPACE FOR THE COREL DATABASE. STRUCTURE OF THE RESULT MATRIX WAS DESCRIBED IN TABLE II.

	Classifiers	Feature vectors					Combination rules					
		LC	G	M	T	CH	Prod	Sum	Max	Min	Med	Maj
Basic classifiers	Gaussian linear	32.59	26.77	31.86	36.04	18.45	15.57	16.75	16.52	16.52	21.27	19.33
	Gaussian quadratic	34.05	26.63	33.00	37.62	17.32	15.11	14.71	17.32	17.32	23.96	21.70
	Logistic linear	30.25	24.93	30.59	35.80	17.23	14.48	14.62	17.35	17.35	17.75	17.67
	Scaled nearest mean	42.90	40.09	40.62	35.02	35.19	24.51	24.01	27.79	27.79	26.88	26.27
Comb. rules	Product	32.12	25.87	31.52	35.34	17.79	14.91					
	Sum	32.15	25.87	31.51	35.36	17.80		15.54				
	Max	31.41	25.50	31.02	34.73	17.93			17.07			
	Min	31.41	25.50	31.02	34.73	17.93				17.07		
	Median	32.54	26.21	31.86	36.04	18.55					21.25	
	Majority vote	32.54	26.21	31.86	36.04	18.55						20.69

TABLE IV

CLASSIFICATION IN THE FEATURE SPACE. EACH NUMBER CORRESPONDS TO THE PERCENT CLASSIFICATION ERROR FOR THE TESTING DATASET BASED ON THE MEASUREMENTS DENOTED BY $\mathbf{x}_{ik}$ IN FIG. 3. THE BEST MODEL IS UNDERLINED FOR EACH CASE.

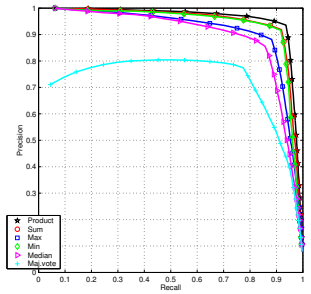
Classifiers	VisTex database					COREL database				
	LC	G	M	T	CH	LC	G	M	T	CH
Gaussian linear	46.34	46.79	47.41	46.38	45.18	49.34	49.52	49.44	49.45	49.36
Gaussian quadratic	21.01	14.80	19.36	17.15	9.43	33.52	27.92	32.72	34.03	20.23
Logistic linear	46.37	46.82	47.39	46.38	45.20	49.34	49.51	49.44	49.45	49.36
Scaled nearest mean	45.26	46.77	46.85	46.53	43.66	49.23	49.35	49.28	49.34	49.08
Nearest neighbor	20.00	6.36	16.73	22.38	6.42					
Parzen	19.75	6.29	15.23	16.12	6.33					
Binary decision tree		9.10	18.70		9.50					
Neural network	16.57	11.70	15.43	17.96	10.23					

B. Retrieval Performance

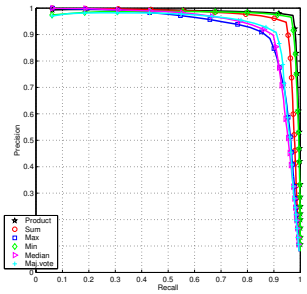
The first set of retrieval experiments evaluated using individual feature vectors with the probabilistic and geometric similarity measures of Section IV. The results were consistent with those of the classification experiments. Likelihood-based measures always performed significantly better than the Minkowsky metrics. The most successful similarity model was again the multivariate Gaussian. The best features were the Gabor and color histograms.

The second set of experiments consisted of using combinations of classifiers trained on multiple probability spaces. Results, as summarized in Fig. 6, showed that the classifier combination models that performed the best in classification experiments consistently gave better results than other models in retrieval experiments. The reasons for relatively low precision in the low recall parts of some COREL precision-recall curves were the relatively small training dataset and the small number of classifiers used during combination. Since the testing and training sets were different, the query images could not always be retrieved as the very top images in the retrieval set and we could not have a perfect retrieval when only a few images were retrieved. However, the precision-recall curves stayed flat for a large range of recall because the classifiers consistently retrieved more relevant images compared to the cases without combination. Using two-thirds of the whole data for training and one-third for testing slightly improved the results. The best performing combinations were the product and max rules with logistic linear classifiers. Combining the outputs of all classifiers for all feature vectors did not give much improvement and was not worth the heavier computation.

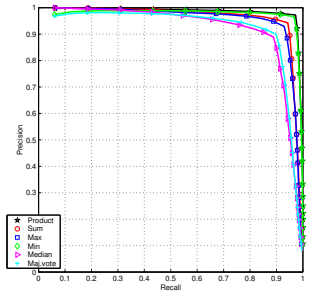
We also compared the proposed framework to other methods. Since our goal was to combine multiple features and incorporate relevance feedback for interactive retrieval, we chose the MARS [12] and ETHZ [43] models as the competing algorithms. MARS uses weighted linear combinations of multiple feature vectors and Euclidean distance values where



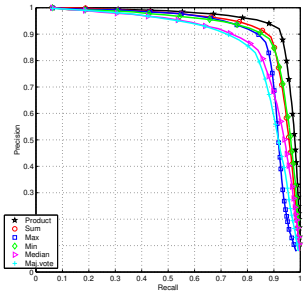
(a) VisTex: Gauss. lin.



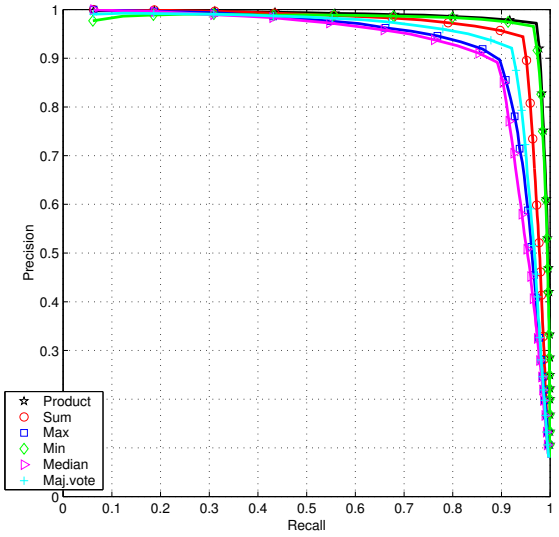
(b) VisTex: Gauss. quad.



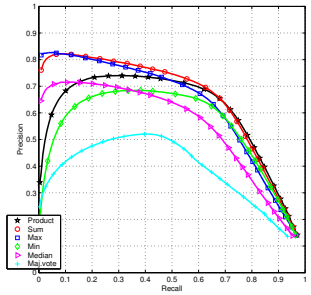
(c) VisTex: Logistic lin.



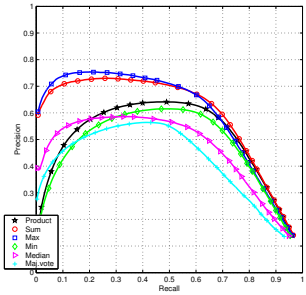
(d) VisTex: Near. neigh.



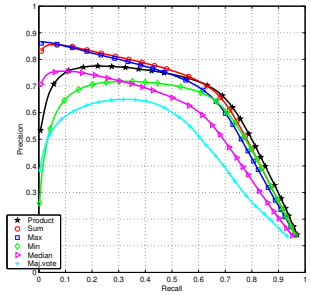
(e) VisTex: All combined



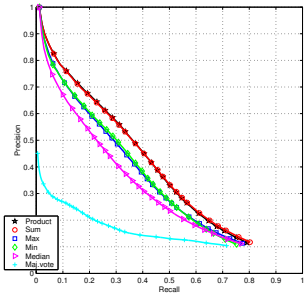
(f) COREL: Gauss. lin.



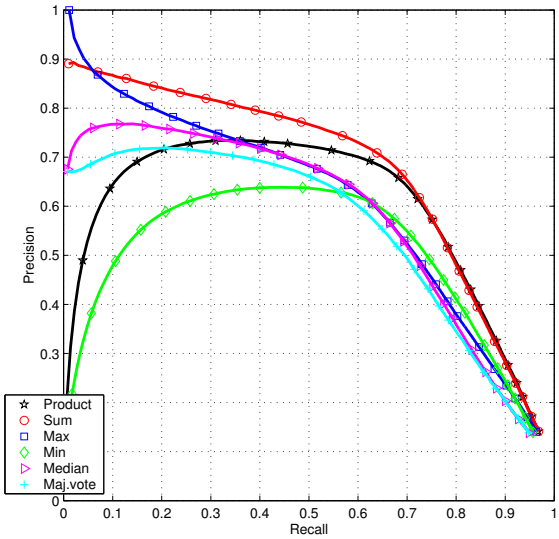
(g) COREL: Gauss. quad.



(h) COREL: Logistic lin.

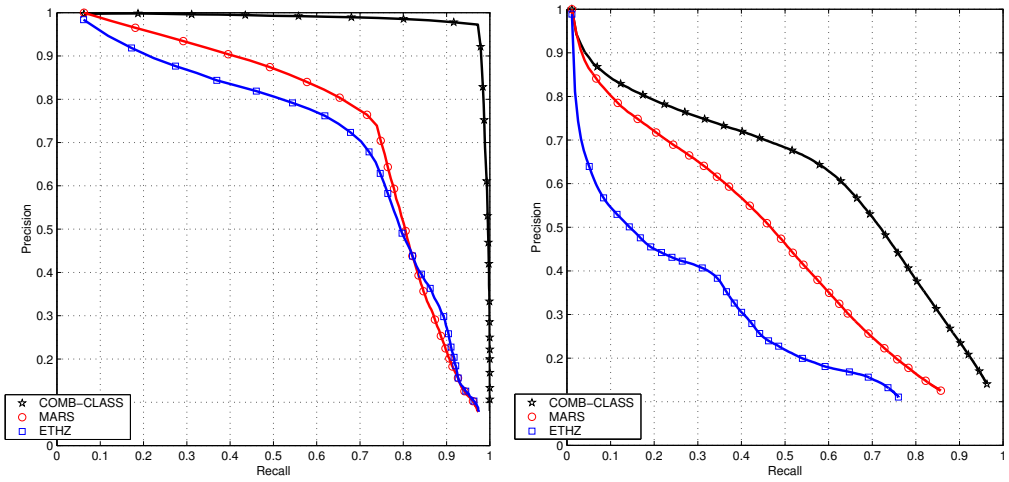


(i) COREL: Nearest mean



(j) COREL: All combined

Fig. 6. Retrieval performance in terms of precision (y-axis) and recall (x-axis) by combining the outputs of a particular classifier for all feature vectors (Figs. (a)-(d) and (f)-(i)) and combining the outputs of all classifiers for all feature vectors (Figs. (e) and (j)). Multivariate Gaussian was used as the similarity model for all features. Different curves within the same plot represent the classifier combination methods: product rule (black pentagram), sum rule (red circle), max rule (blue square), min rule (green diamond), median rule (magenta triangle), and majority vote rule (cyan plus).



(a) VisTex: Comparison (b) COREL: Comparison

Fig. 7. Retrieval performance in terms of precision (y-axis) and recall (x-axis) by the combined classifiers model (black pentagram), the MARS model (red circle) and the ETHZ model (blue square). The classification framework proposed in this paper performed significantly better than the competing algorithms.

the weights can be updated using user's feedback. The ETHZ model clusters vectors in each feature space and uses a naive Bayes network with discrete probability tables to compute the probability of an image being similar to the query given the cluster labels for all features for that image. ([43] used feature vectors computed for each pixel for pixel level classification and we used the feature vectors listed in Section III for image level classification.) Searching starts with uniform priors and relevance feedback is used to update the conditional probabilities using relative frequencies. Precision-recall curves are given in Fig. 7. Our probabilistic framework performed significantly better than the competing algorithms.

The final set of experiments was done to simulate relevance feedback iterations for all test images. Since our user interface shows the top 12 matches on the first screen, we used the feedback available from only the top 12 images in iterative retrievals. Each test image was used as a query and the retrieved images that belonged to its ground truth group were fed back as positive matches and the rest of the 12 were fed back as negative matches. Table V shows that each iteration improved over the case without feedback while the first

TABLE V

AVERAGE PRECISION WHEN 12 IMAGES WERE RETRIEVED USING DIFFERENT FEEDBACK MODELS. "n R.F." REPRESENTS THE n'TH FEEDBACK ITERATION. IMPROVEMENTS FOR EACH ITERATION OVER THE CASE WITHOUT FEEDBACK (0 R.F.) ARE GIVEN IN PARENTHESES. BAYESIAN RELEVANCE FEEDBACK ACHIEVED ALMOST PERFECT RETRIEVAL.

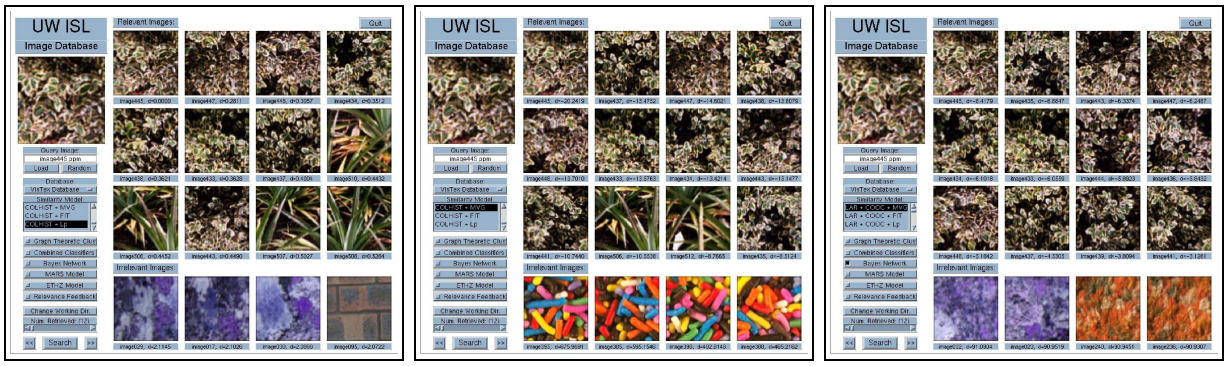
Database	Method	0 r.f.	1 r.f.	2 r.f.	3 r.f.	4 r.f.
VisTex	Bayesian	0.9879	0.9945 (0.66%)	0.9946 (0.68%)	0.9946 (0.68%)	0.9946 (0.68%)
	MARS	0.8225	0.9078 (10.38%)	0.9220 (12.10%)	0.9206 (11.94%)	0.9230 (12.22%)
	ETHZ	0.7773	0.8946 (15.09%)	0.9134 (17.51%)	0.9293 (19.56%)	0.9348 (20.26%)
COREL	Bayesian	0.8342	0.9113 (9.24%)	0.9363 (12.24%)	0.9407 (12.77%)	0.9421 (12.93%)
	MARS	0.7860	0.7441 (-5.34%)	0.7612 (-3.16%)	0.7716 (-1.83%)	0.7894 (0.42%)
	ETHZ	0.5757	0.7492 (30.12%)	0.7809 (35.63%)	0.8081 (40.36%)	0.8282 (43.85%)

iteration had the largest improvement. This is a desired situation because many relevant images are already available to the user after the first feedback. We achieved almost perfect retrieval (precision above 99%) for the VisTex Database and a significant improvement for the COREL Database. Both MARS and ETHZ models also showed improvements for the VisTex Database but the former gave worse results for the COREL Database. The ETHZ model gave large improvements in subsequent iterations but required more iterations than others. However, it was more robust than the MARS model because it also used probabilities instead of heuristic weight assignments in the geometric similarity framework.

Example queries are given in Figs. 8–11. More examples and the details of the experiments can be found in [35].

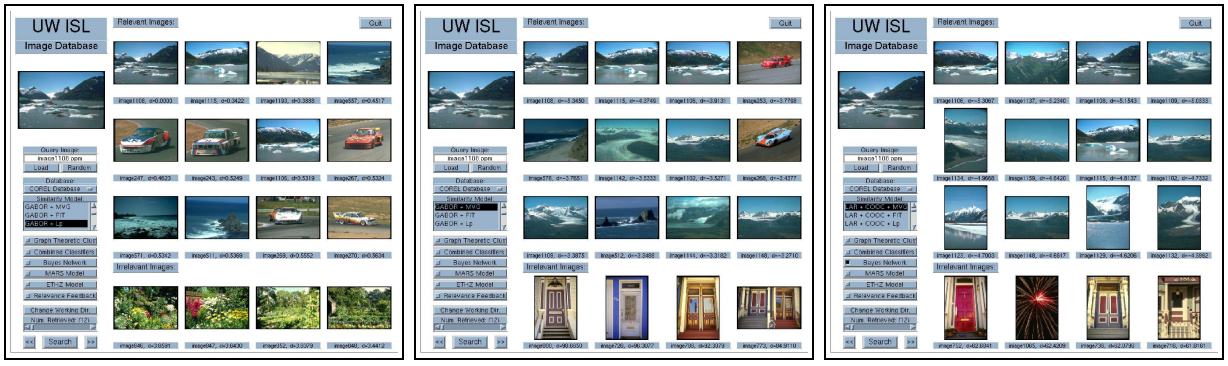
IX. CONCLUSIONS

Numerous feature extraction methods and similarity measures have been proposed in the literature but there is currently no generally applicable, well-defined and effective methodology to design a content-based image retrieval system. We posed the retrieval problem in a two-class classification framework where the goal was to minimize the classification error between the relevance and irrelevance classes of the query. We developed solutions to different levels of the retrieval process within this framework. Feature extraction and normalization



(a) Color histograms and L_p metric (8/12) (b) Color histograms and multivariate Gaussian (10/12) (c) Combined classifiers (12/12)

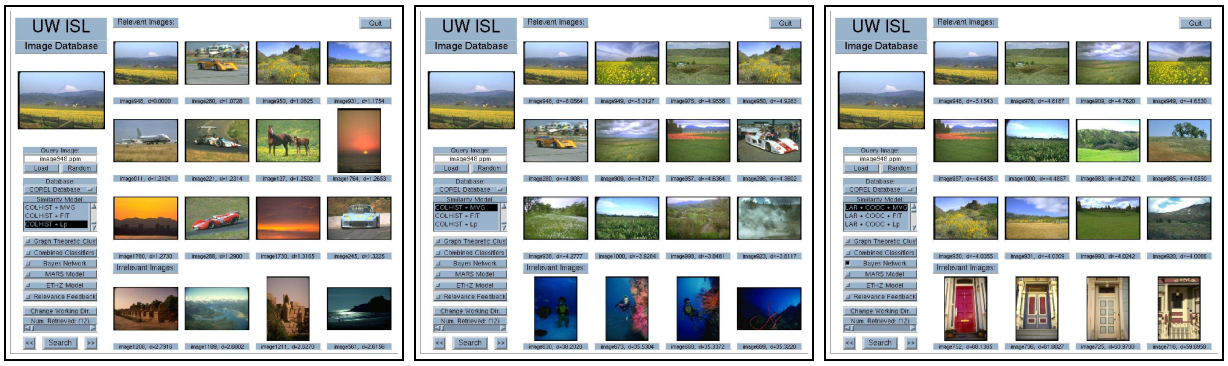
Fig. 8. An example query of leaves from the VisTex Database. The first three rows in the user interface show the best 12 matches and the last row shows the worst 4 matches. Numbers in parentheses in sub-captions show the number of correct matches for each case. Different types of leaves could not be distinguished when only color features were used but combined classifiers achieved perfect retrieval.



(a) Gabor features and L_p metric (4/12) (b) Gabor features and multivariate Gaussian (8/12) (c) Combined classifiers (12/12)

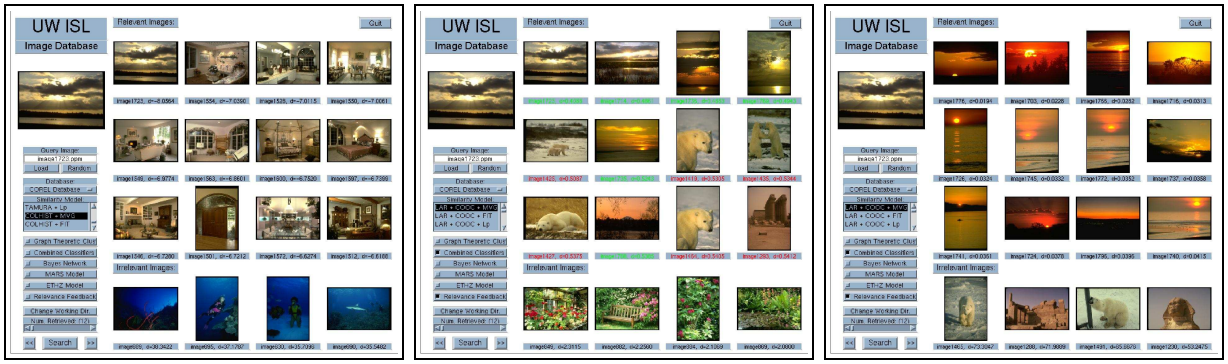
Fig. 9. An example query of glaciers and mountains from the COREL Database. Glaciers, beaches and auto racing images were retrieved together when only Gabor texture features were used but combined classifiers achieved perfect retrieval.

was done by maximizing class separability (pre-processing). Similarity was measured in terms of the likelihood of two images being similar, one being the query image and the other one being an image in the database. Class-conditional probabilities for feature difference vectors were estimated using parametric models like multivariate Gaussian, independently fitted distributions, and Gaussian mixtures. This setting could also be interpreted as a mapping from high-dimensional feature spaces to two-dimensional probability spaces where



(a) Color histograms and L_p metric (3/12) (b) Color histograms and multivariate Gaussian (9/12) (c) Combined classifiers (12/12)

Fig. 10. An example query for fields from the COREL Database. Fields and auto racing images were retrieved together when only color features were used but combined classifiers achieved perfect retrieval.



(a) Color histograms and multivariate Gaussian (1/12) (b) Combined classifiers (no feedback) (6/12) (c) After the first feedback (12/12)

Fig. 11. An example query for sunsets from the COREL Database. Sunsets and residential interiors were retrieved together when only color features were used. Combined classifiers improved the results but retrieved some polar bears instead. Perfect retrieval was achieved with the first feedback iteration after submitting polar bears in (b) as negative examples. Note that the new row of 4 worst matches in (c) includes polar bears as the most irrelevant images to the query.

the Bayes classifier achieves the theoretical minimum classification error. However, the estimated probabilities also had uncertainty due to factors like imperfect density modeling, quantization, high dimensionality, etc. To compensate for errors in modeling probabilities in feature spaces, we proposed a second level modeling as the "probability of probability". Simple linear or quadratic classifiers were trained in two-dimensional probability spaces with the class-conditional probabilities being new features. These classifiers performed much better

than the complex non-linear classifiers trained in the original feature spaces. Furthermore, we used Bayesian combination rules to compute joint posterior probabilities for relevance and irrelevance classes based on classifiers trained in multiple probability spaces corresponding to multiple features. This Bayesian formulation provided a unified and effective framework for fusion of information from different features and similarity measures. Finally, relevance feedback in terms of user's labeling of retrieval results as relevant and irrelevant was incorporated into the Bayesian framework by automatically updating the posterior probability estimates (post-processing).

Extensive experiments on two ground truth databases showed that the proposed probabilistic framework performed significantly better than the commonly used geometric framework of distances and two other competing algorithms. Effectiveness of simple classifiers and Bayesian relevance feedback in improving the retrieval results illustrated the power of the probabilistic framework which simplified the problem and allowed the estimation of less complex yet more powerful models in multiple levels.

Acknowledgments

This work was done when both authors were with the Intelligent System Laboratory at the University of Washington, Seattle. The first author would like to thank Prof. Linda G. Shapiro from the University of Washington for valuable discussions and providing funding during the final stages of this work.

References

[1] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, no. 1, pp. 39–62, March 1999.

[2] 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 Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, December 2000.

[3] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Region-based image querying," in *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[4] W. Y. Ma and B. S. Manjunath, "NETRA: A toolbox for navigating large image databases," in *IEEE Intl. Conf. on Image Processing*, 1997.

[5] P. Felzenszwalb and D. Huttenlocher, "Image segmentation using local variation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Santa Barbara, CA, June 1998, pp. 98–104.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.

[7] E. J. Pauwels and G. Frederix, "Finding salient regions for image segmentation and grouping," *Computer Vision and Image Understanding, Special Issue on Content-Based Access of Image and Video Libraries*, vol. 75, no. 1/2, pp. 73–85, July/August 1999.

[8] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "The QBIC project: Querying images by content using color, texture and shape," in *SPIE Storage and Retrieval of Image and Video Databases*, 1993, pp. 173–181.

[9] C. S. Li and V. Castelli, "Deriving texture set for content based retrieval of satellite image database," in *IEEE Intl. Conf. on Image Processing*, 1997, pp. 576–579.

[10] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, August 1996.

[11] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color- and texture-based image segmentation using EM and its application to content-based image retrieval," in *IEEE Intl. Conf. on Computer Vision*, 1998.

[12] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Segmentation, Description, and Retrieval of Video*

Content, vol. 8, no. 5, pp. 644–655, September 1998.

[13] N. Sebe, M. Lew, and D. P. Huijsmans, "Toward improved ranking metrics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1132–1143, October 2000.

[14] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," in *SPIE Storage and Retrieval of Image and Video Databases II*, February 1994, pp. 34–47.

[15] A. Vailaya, A. Jain, and H. J. Zhang, "On image classification: City images vs. landscapes," *Pattern Recognition*, vol. 31, pp. 1921–1936, December 1998.

[16] A. P. Berman and L. G. Shapiro, "A flexible image database system for content-based retrieval," *Computer Vision and Image Understanding, Special Issue on Content-Based Access of Image and Video Libraries*, vol. 75, no. 1/2, pp. 175–195, July/August 1999.

[17] N. Haering and N. de Vitoria Lobo, "Features and classification methods to locate deciduous trees in images," *Computer Vision and Image Understanding, Special Issue on Content-Based Access of Image and Video Libraries*, vol. 75, no. 1/2, pp. 133–149, July/August 1999.

[18] K. Tieu and P. Viola, "Boosting image retrieval," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, Hilton Head Island, South Carolina, June 2000, pp. 228–235.

[19] Y. Rui and T. Huang, "Optimizing learning in image retrieval," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, Hilton Head Island, South Carolina, June 2000, pp. 236–243.

[20] S. Aksoy and R. M. Haralick, "Probabilistic vs. geometric similarity measures for image retrieval," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, Hilton Head Island, South Carolina, June 2000, pp. 357–362.

[21] ——, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 563–582, May 2001.

[22] B. Moghaddam, T. Jebera, and A. Pentland, "Efficient MAP/ML similarity matching for visual recognition," in *14th IAPR Intl. Conf. on Pattern Recognition*, vol. 1, 1998,

pp. 876–881.

[23] A. Vailaya, M. Figueiredo, A. Jain, and H. J. Zhang, "Content-based hierarchical classification of vacation images," in *IEEE Intl. Conf. on Multimedia Computing and Systems*, Florence, Italy, June 1999.

[24] N. Vasconcelos and A. Lippman, "A probabilistic architecture for content-based image retrieval," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, Hilton Head Island, South Carolina, June 2000, pp. 216–221.

[25] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, "The Bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments," *IEEE Trans. on Image Processing*, vol. 9, no. 1, pp. 20–37, January 2000.

[26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, Inc., 2000.

[27] S. Aksoy and R. M. Haralick, "Using texture in image similarity and retrieval," in *Texture Analysis in Machine Vision*, ser. Series in Machine Perception and Artificial Intelligence, M. Pietikainen, Ed. World Scientific, 2000, vol. 40, pp. 129–149.

[28] F. A. Cheikh, B. Cramariuc, C. Reynaud, M. Qinghong, B. D. Adrian, B. Hnich, M. Gabbouj, P. Kerminen, T. Makinen, and H. Jaakkola, "MUVIS: A system for content-based indexing and retrieval in large image databases," in *SPIE Storage and Retrieval of Image and Video Databases VII*, San Jose, CA, January 1999, pp. 98–106.

[29] M. Swain and D. Ballard, "Color indexing," *Intl. Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[30] K. V. Bury, *Statistical Models in Applied Science*. John Wiley & Sons, Inc., 1975.

[31] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & Sons, Inc., 1997.

[32] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[33] R. P. W. Duin, "Classifiers in almost empty spaces," in *15th IAPR Intl. Conf. on Pattern Recognition*, vol. 2, Barcelona, Spain, September 2000, pp. 1–7.

[34] R. Pekalska and R. P. W. Duin, "Classifiers for dissimilarity-based pattern recognition," in *15th IAPR Intl. Conf. on Pattern Recognition*, vol. 2, Barcelona, Spain, September 2000, pp. 12–16.

[35] S. Aksoy, "A probabilistic similarity framework for content-based image retrieval," Ph.D. dissertation, University of Washington, Seattle, WA, June 2001.

[36] R. P. W. Duin, "PRTools 3.0, A Matlab toolbox for pattern recognition," 2000, online: http://www.ph.tn.tudelft.nl/~bob/PRTOOLS.html.

[37] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, March 1998.

[38] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, January 1994.

[39] J. Kittler and S. A. Hojjatoleslami, "A weighted combination of classifiers employing shared and distinct representations," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1998, pp. 924–929.

[40] Y.-Y. Chou and L. G. Shapiro, "A hierarchical multiple classifier learning algorithm," in *15th IAPR Intl. Conf. on Pattern Recognition*, vol. 2, Barcelona, Spain, September 2000, pp. 152–155.

[41] R. P. W. Duin and D. M. J. Tax, "Experiments with classifier combining rules," in *First Intl. Workshop on Multiple Classifier Systems*, Cagliari, Italy, June 2000, pp. 16–29, as Lecture Notes in Computer Science, vol. 1857.

[42] N. Vasconcelos and A. Lippman, "Learning from user feedback in image retrieval systems," in *Neural Information Processing Systems*, Denver, CO, 1999.

[43] M. Schroder, H. Rehrauer, K. Siedel, and M. Datcu, "Interactive learning and probabilistic retrieval in remote sensing image archives," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 38, no. 5, pp. 2288–2298, September 2000.