Bayesian Models for Visual Information Retrieval

by

Nuno Miguel Borges de Pinho Cruz de Vasconcelos

M.S., Massachusetts Institute of Technology, (1993)
Licenciatura, Electrical and Computer Engineering
Universidade do Porto, Portugal (1988)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Massachusetts Institute of Technology

June 2000

Author: ______________________________
Program in Media Arts and Sciences
May 11, 2000

Certified by: ______________________________
Andrew B. Lippman
Associate Director, MIT Media Laboratory
Thesis Supervisor

Accepted by: ______________________________
Stephen A. Benton
Chairperson, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Bayesian Models for Visual Information Retrieval

by

Nuno Miguel Borges de Pinho Cruz de Vasconcelos

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on May 11, 2000
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Abstract

This thesis presents a unified solution to visual recognition and learning in the context of visual information retrieval. Realizing that the design of an effective recognition architecture requires careful consideration of the interplay between feature selection, feature representation, and similarity function, we start by searching for a performance criteria that can simultaneously guide the design of all three components. A natural solution is to formulate visual recognition as a decision theoretical problem, where the goal is to minimize the probability of retrieval error. This leads to a Bayesian architecture that is shown to generalize a significant number of previous recognition approaches, solving some of the most challenging problems faced by these: joint modeling of color and texture, objective guidelines for controlling the trade-off between feature transformation and feature representation, and unified support for local and global queries without requiring image segmentation. The new architecture is shown to perform well on color, texture, and generic image databases, providing a good trade-off between retrieval accuracy, invariance, perceptual relevance of similarity judgments, and complexity.

Because all that is needed to perform optimal Bayesian decisions is the ability to evaluate beliefs on the different hypothesis under consideration, a Bayesian architecture is not restricted to visual recognition. On the contrary, it establishes a universal recognition language (the language of probabilities) that provides a computational basis for the integration of information from multiple content sources and modalities. In result, it becomes possible to build retrieval systems that can simultaneously account for text, audio, video, or any other content modalities.

Since the ability to learn follows from the ability to integrate information over time, this language is also conducive to the design of learning algorithms. We show that learning is, indeed, an important asset for visual information retrieval by designing both short and long-term learning mechanisms. Over short time scales (within a retrieval session), learning is shown to assure faster convergence to the desired target images. Over long time scales (between retrieval sessions), it allows the retrieval system to tailor itself to the preferences of particular users. In both cases, all the necessary computations are carried out through Bayesian belief propagation algorithms that, although optimal in a decision-theoretic sense, are extremely simple, intuitive, and easy to implement.

Thesis Supervisor: Andrew B. Lippman, Associate Director, MIT Media Laboratory

Bayesian Models for Visual Information Retrieval

by

Nuno Miguel Borges de Pinho Cruz de Vasconcelos

Reader: ____________________________

Aaron Bobick
Associate Professor
College of Computing
Georgia Institute of Technology

Reader: ____________________________

Robert M. Gray
Professor and Vice Chair
Department of Electrical Engineering
Stanford University

Reader: ____________________________

Murat Kunt
Professor
Department of Electrical Engineering
Swiss Federal Institute of Technology

To my parents,
Assunção and Manuel.

Contents

Chapter 1

Introduction

If there is a defining characteristic of the new digital communications media, that characteristic is the potential for interactivity [116, 117]. In the digital world, communication is synonymous with computation: fast processors are required at both ends of the pipe to transform the massive amounts of audio, video, and textual information characteristic of modern applications into a compact and error-resilient bitstream which can be transmitted rapidly and reliably. Digital decoders are therefore intelligent, giving the user the ability to actively search, browse through, or even publish information, instead of passively "tuning in" to what is going on a broadcast channel.

Such a shift with regards to the control over the communications process unveils a new universe of requirements and possibilities for representing audio-visual content. Because digital media are characterized by ubiquitous networking, computation, storage capacity, and absence of geographical barriers, the user has instantaneous access to a virtually unlimited amount of information. But while the ability to tap into such an infinite source of resources is exciting, it can also lead, in the absence of appropriate indexing and navigation mechanisms, to a significant degree of frustration and helplessness.

A significant challenge is, therefore, to design representations that can support not only efficient transmission and storage of information but also filtering, sorting, classification, retrieval, summarization, browsing, and manipulation of content. The challenge is particularly strong when the content to represent does not lend itself to unambiguous textual descriptions. Today, we simply cannot understand the structure contained in a sound recording or an image, and this limits our ability in various areas.

The issues of image understanding and representation are central to this thesis, where we address the problem of how to design systems for retrieving information from large image repositories. While text can be a powerful aid, experience reveals that traditional text-based search engines fall short of fulfilling all the requirements of visual retrieval. The problem is that images can have multiple interpretations and text annotations usually say more about the interpretation of the person that created them than the one that may be relevant for someone else's search. The alternative that we pursue here is to design *content-based image retrieval* (CBIR) systems, i.e. systems that allow users to express their queries directly in terms of visual attributes.

While, ideally, we would like CBIR systems to understand natural language scene descriptions, e.g. "show me all the pictures of a tiger running in the wild," we simply do not yet understand images well enough for this to be feasible. The only viable alternative, in the short-term, is to request less from the machine and more from the users. An increasingly popular solution [129, 43, 118, 32, 7, 164] is to build systems that can make judgments of *visual similarity* and place on the user the burden of guiding the search. This is accomplished through an iterative process where the user provides examples, the machine suggests matches and, from those, the user selects the next round of examples. The obvious advantage is that the CBIR system is now much simpler to design. The main drawback is that, because visual similarity is not the same as *semantic similarity*, the matches returned by the machine will not always be what the user is looking for. Since this increases the risk of user frustration, the next step is to give retrieval systems the ability to react intelligently to the user feedback, i.e. to *learn* from the user interaction [132].

There are, therefore, two fundamental problems to be addressed. First, the design of the visual recognition architecture itself and, second, the design of learning mechanisms to facilitate the user interaction. Obviously, the two problems cannot be solved in isolation since the careless selection of the recognition architecture will make learning more difficult and vice-versa.

1.1 Contributions of the thesis

This thesis presents a unified solution to visual recognition and learning by formulating recognition as a decision theoretical problem, where the goal is to *minimize the probability of retrieval error*. Besides providing an objective performance criteria for the design and evaluation of retrieval systems, the new formulation also leads to solutions that are optimal in a well-defined sense, and can be derived from the well understood principles of Bayesian inference. The resulting Bayesian recognition architecture has several attractives. First, it is based on a universal recognition language (the language of probabilities) that provides a computational basis for the integration of information from multiple content sources and modalities. In result, it becomes possible to build systems that simultaneously account for text, audio, video, or any other modalities. Second, because learning is a consequence of the ability to integrate information over time, this language also provides a basis for designing learning algorithms. It therefore becomes possible to design retrieval systems that can rely on user-feedback both to retrieve images faster and to adapt themselves to their users' preferences. Third, all the integration can be performed through belief propagation algorithms that, although optimal in the decision-theoretic sense, are extremely simple, intuitive, and easy to implement. Finally, as an architecture for visual recognition, it generalizes current retrieval solutions, solving some of the most challenging problems faced by these: joint modeling of color and texture, objective guidelines for controlling the trade-off between feature transformation and feature representation, and unified support for local and global queries without requiring image segmentation.

1.1.1 An architecture for visual recognition

An architecture for visual information retrieval is composed by three fundamental building blocks: 1) a transformation from the image pixel space into a feature space that provides sufficient discrimination between the image classes in the database, 2) a feature representation that describes how each image populates the feature space, and 3) a retrieval metric that relies on this feature representation to infer similarities between images. Even though significant attention has been recently devoted to each of these individual components, there have been significantly fewer attempts to investigate the interrelationships among them and how these relationships may affect the performance of retrieval systems.

In fact, current retrieval solutions can be grouped into two major disjoint sets: on one hand, a set of representations that evolved from texture analysis and are tailored for texture and, on the other hand, a set that grew out of object recognition and is tailored for color. We refer to the former as *texture-based retrieval* and to the latter as *color-based retrieval*. Retrieval approaches in these two classes vary widely with respect to the emphasis placed on the design of the individual retrieval components. For example, because most texture databases consist of homogeneous images, it is reasonable to assume that the associated features will be Gaussian distributed. The Gaussian density is thus commonly used for feature representation (although usually in implicit form), and simple metrics such as the Euclidean or the Mahalanobis distance serve as a basis to evaluate similarity. Given these feature representation and similarity function, the main goal of texture analysis is to find the set of features that allow best discrimination between the texture classes in the database. This goal can be made explicit in the formulation of the problem [176, 183, 40] or implicit [134, 94, 104, 96, 102, 109].

Unlike texture-based retrieval, feature selection has not been a critical issue for color-based retrieval, where the features are usually the pixel colors themselves. Instead a significant amount of work has been devoted to the issue of feature representation, where several approaches have been proposed. The majority of these are variations on the color histogram initially proposed for object recognition [172], e.g. the color coherence vector [126], the color correlogram [66], color moments [167], etc. While each feature representation may require a specific similarity function, the most commonly used are L^p distance norms in feature representation space. Among these, the L^1 distance between color histograms, also known as histogram intersection [172], has become quite popular.

While they have worked well in their specific domains, these representations break down when applied to databases of generic imagery. The main problem for texture-based solutions is that, since generic images are not homogeneous, their features cannot be accurately modeled as Gaussian and simple similarity metrics are no longer sufficient. On the other hand, color-based solutions are plagued by the exponential complexity of the histogram on the dimension of the feature space, and are applicable only to low-dimensional features (e.g.

pixel colors). Hence, they are unable to capture the spatial dependencies that are crucial for texture characterization.

In the absence of solutions that can account for both color and texture, retrieval systems must resort to different features, representations, and similarity functions to deal with the two image attributes [43, 118, 164, 44, 175], making it difficult to perform joint inferences with respect to both. The standard solution is to evaluate similarity according to each of the attributes and obtain an overall measure by weighting linearly the individual distances. This opens up the question of how to weigh different representations on different feature spaces, a problem that has no easy solution.

Ideally, one would like to avoid these problems altogether by designing a retrieval architecture capable of accounting for both color and texture. An obvious, but often overlooked, statement is that carefully designing any of the individual retrieval modules is not sufficient to achieve a good overall solution. Instead, the design must start from an unambiguous performance criteria, and all modules designed with the goal of optimizing the overall performance. In this context, we start by posing the retrieval problem as one of optimal decision-making. Given a set of database image classes and a set of query features, the goal is to find the map from the latter to the former that *minimizes the probability of retrieval error*.

This is shown to be interesting in two ways. First, it leads to a Bayesian formulation of retrieval and a probabilistic retrieval criteria that either generalizes or improves upon the most commonly used similarity functions (Mahalanobis distance, L^p norms, and minimum discrimination information, among others). Second, it provides objective guidelines for the selection of both the feature transformation and representation. The first of these guidelines is that the most restrictive constraints on the retrieval architecture are actually imposed on the feature transformation. In fact, optimal performance can only be achieved under a restricted set of *invertible transformations* that leaves small margin for feature optimization. The second guideline is that performance quality is directly related to the quality of density estimates, which is in turn determined by the feature representation.

A corollary, of great practical relevance, of these guidelines is that there is less to be gained from the feature transformation than from accurate density estimation. This, and the fact that the difficulty of the latter increases with the dimensionality of the space, motivates us to restrict the role of the former to that of dimensionality reduction; i.e. we seek the feature transformation that achieves the optimal trade-off between invertability and dimensionality reduction. This leads to the well known principal component analysis which is, in turn, well approximated by perceptually more justifiable multi-resolution transformations that are common in the compression literature, e.g. the discrete cosine transform (DCT).

On the other hand, we devote significant attention to the issue of feature representation. We notice that a good representation should be 1) expressive enough to capture the details of multi-modal densities that characterize generic imagery, and 2) compact enough to be tractable in high-dimensional spaces. Viewing the standard Gaussian and histogram representations as particular cases of the generic family of mixture models reveals that insufficient number of basis functions, poor kernel selection, and inappropriate partitioning of the space are the major reasons behind their inability to meet these requirements. The Gaussian mixture then emerges as a unifying feature representation for both color and texture, eliminating the problems associated with the standard approaches. Further observation that a mixture model defines a family of embedded densities leads to the concept of embedded multi-resolution mixtures (EMM). These are a family of embedded densities ranging over multiple image scales that allow explicit control of the trade-off between spatial support and invariance.

Overall, the retrieval architecture composed by the Bayesian similarity criteria, the DCT feature transformation, and the embedded mixture representation provides a good trade-off between retrieval accuracy, invariance, perceptual relevance of similarity judgments, and complexity. We illustrate all these properties with an extensive experimental evaluation on three different databases that stress different aspects of the retrieval problem[1]: the Brodatz texture database, the Columbia object database, and the Corel database of stock photography. In all cases, the new approach outperforms the previous best retrieval solutions both in terms of objective (precision/recall) and subjective (perceptual) evaluation.

[1] The experimental set up is discussed in detail in Appendix A.

1.1.2 Learning from user interaction

While a sophisticated architecture for evaluating image similarity is a necessary condition for the design of successful retrieval systems, it is by no means sufficient. The fact is that the current understanding of the image analysis problem is too shallow to guarantee that any retrieval system (no matter how sophisticated) will always find the desired images in response to a user query. In result, retrieval is usually an interactive process where 1) the user guides the search by rating the systems suggestions, and 2) the system refines the search according to those ratings. Conceptually, a retrieval system is nothing more than an interface between an intelligent high-level system (the user's brain) that can perform amazing feats in terms of visual interpretation but is limited in speed, and a low-level system (the computer) that has very limited visual abilities but can perform low-level operations very efficiently. Therefore, the more successful retrieval systems will be those that make the user-machine interaction easier.

The goal is to exploit as much as possible the strengths of the two players: the user can provide detailed feedback to guide the search when presented with a small set of meaningful images, the machine can rely on that feedback to quickly find the next best set of such images. To enable convergence to the desired target image, the low-level system cannot be completely dumb, but must know how to *integrate* all the information provided by the user over the entire course of interaction. If this were not the case, it would simply keep oscillating between the image sets that best satisfied the latest indication from the user, and convergence to the right solution would be difficult.

This ability to learn by integrating information must occur over various time scales. Some components maybe hard-coded into the system from the start, e.g. the system may contain a specialized face-recognition module. However, hard-coding leaves small room for *personalization*. Not all users are interested in the same visual concepts and retrieval systems should be able to respond to the individual user requirements. Therefore, most components should, instead, be learned over time. Since users tend to search often for the concepts that are of greatest interest to them, examples of these concepts will be available. Hence, it is in principle possible for the system to build internal concept representations and become progressively more apt at recognizing specific concepts as time progresses. We refer to such mechanisms as *long-term learning* or *learning between retrieval sessions*, i.e. learning that does not have to occur on-line, or even in the presence of the user.

Information must also be integrated over short-time scales, e.g. during a particular retrieval session. In the absence of *short-term* or *in-session learning*, the user would have to keep repeating the information provided to the retrieval system from iteration to iteration. This would be cumbersome and extremely inefficient since a significant portion of the computation performed by the latter would simply replicate what had been done in previous iterations. Unlike long-term learning, short-term learning must happen on-line and therefore has to be fast.

In this thesis, we show that the Bayesian formulation of the retrieval problem leads to very natural procedures for inference and learning. This is not surprising since probabilistic representations are the soundest computational tool available to deal with uncertainty and the laws of probability the only principled mechanism for making inferences in its presence. We illustrate this point by designing both short- and long-term learning mechanisms that can account for both positive and negative user-feedback and presenting experimental evidence that illustrates the clear benefits of learning for CBIR.

1.2 Organization of the thesis

The standard thesis format asks for an introduction chapter, a chapter of literature review, and then a sequence of chapters describing the contributions, experimental validation, and conclusions. In this document, we deviate from this standard format in at least two ways.

First, because one of the points of the thesis is to demonstrate that a significant part of what has been proposed in the retrieval literature are sub-optimal special cases of the Bayesian formulation now introduced, we do not include a standard review chapter. Instead, we establish connections to previous work as we discuss Bayesian retrieval. We believe that this will be easier for the reader than simply including a large review section and referring to it in the chapters ahead. Since the organization of the thesis follows the fundamental structure of a retrieval system, a reader interested in reviewing particular sub-topics (e.g. feature sets commonly used to characterize texture) can simply skip ahead to the corresponding chapter. Second, instead of having a chapter entirely devoted to experimental validation, we include experimental results as we go along. This allows us to build on the experimental results of the earlier chapters to motivate the ideas introduced in subsequent ones.

The organization of the thesis is as follows. In Chapter 2, we introduce the Bayesian retrieval criteria and show that it generalizes or outperforms most of the similarity measures in common use in the retrieval literature. In Chapter 3, we discuss how Bayesian similarity provides guidelines for feature selection and representation and discuss previous retrieval strategies in light of these guidelines. We conclude that the two most prevalent strategies have strong limitations for retrieval from generic image databases and devise an alternative strategy. This strategy is then implemented in Chapters 4 and 5 where we present solutions for feature transformation and representation.

The issue of local vs. global similarity is addressed on Chapter 6, where we show that Bayesian retrieval provides a natural solution to local queries. However, we also point out that for global queries the straightforward implementation of the Bayesian criteria is usually too expensive. To correct this problem, we devise efficient approximations that are shown to achieve similar performance to that of the exact Bayesian inferences. The ability to account for local queries is a requirement for learning, which is then discussed in Chapters 7 and 8. In Chapter 7, we present a short-term learning algorithm that can exploit both positive and negative user feedback to achieve faster convergence to the target images. In Chapter 8, we present a long-term learning algorithm that gives retrieval systems the ability to, over time, tailor themselves to the interests of their users.

Finally, in Chapter 9 we describe the practical implementation of all the ideas in the "Retrieval as Bayesian Inference" (RaBI) image retrieval system, and we present conclusions and directions for future work in Chapter 10. A discussion of the experimental set up used to evaluate retrieval performance is presented in Appendix A.

Chapter 2

Retrieval as statistical inference

The central component of an architecture for content-based image retrieval (CBIR) is a criteria for evaluating image similarity. Typically, this is achieved by defining a similarity function that maps the space of image classes that compose a database into the space of possible orderings for those classes. In this chapter, we argue that a natural goal for a retrieval system is to minimize the probability of retrieval error[1]. This leads to a new formulation of the retrieval problem, derived from Bayesian decision theory, and a probabilistic criteria for the evaluation of image similarity.

In addition to minimizing retrieval error the Bayesian solution unifies a large body of similarity functions in current use. In particular, it is shown that most of these functions can be derived from Bayesian retrieval by 1) making assumptions with respect to the densities of the image classes or 2) approximating the quantities involved in Bayesian inference. This suggests that, even if minimizing probability of error is not the desired goal for the retrieval system, there is no apparent reason to prefer those functions to the Bayesian counterpart. The theoretical claims are validated through retrieval experiments that confirm the superiority of Bayesian similarity.

[1] A more generic performance criteria is the Bayes risk [10] where different types of errors are assigned different costs. Because we currently do not have good strategies to define such costs, we simply assign a unitary cost to all errors (and zero cost to all correct decisions), in which case Bayes risk is equivalent to the probability of error. It would, however, be straightforward to extend the retrieval formulation presented in the thesis to the minimization of Bayes risk, if more detailed costs were available.

2.1 Terms and notation

We start by defining some terms and notation. An *image* I is a map from a two-dimensional pixel lattice of size $P \times Q$

$$\mathcal{L} = \{1, \ldots, P\} \times \{1, \ldots, Q\} \tag{2.1}$$

into the space $\mathcal{A}$ of all $P \times Q$ arrays of pixel colors

$$I : \mathcal{L} \to \mathcal{A}.$$

The *color* of pixel $(i, j) \in \mathcal{L}$ is denoted by $I_{i,j}$ and can be a scalar (for gray-scale images) or a 3-D vector (for color images). In the former case, the pixel color is also referred to as *intensity*. The number of color channels in an image is denoted by c.

We define two indicator functions. For any set E, the *set indicator* function is

$$\chi_E(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

For any two integers i and j, the *Kronecker delta* function is defined by

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

A *partition* of a set E is a collection of subsets (also known as *partition cells* or *regions*) $\{E_1, \ldots, E_R\}$ that are disjoint and cover E, i.e.

$$\cup_{i=1}^{R} E_i = E \text{ and } E_i \cap E_j = \emptyset, \forall i \neq j. \tag{2.4}$$

An *image database* $\mathbf{D}$ is a collection of images

$$\mathbf{D} = \{I_1, \ldots, I_S\}$$

where S is the *database size*. Within a database, images are organized into M *image classes*

$$\mathbf{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_M\}$$

where the $\mathbf{D}_i$ are a partition for $\mathbf{D}$.

In general, a classification of the images in the database is available. If that is not the case, two alternatives can be pursued. The first is to assume that each image defines a class by its own. This solution reflects the absence of any prior knowledge about the database content and leads to as many classes as the cardinality of the database. We denote this type of structure as a *flat database*. The second is to try to generate the classification either automatically or manually. Since individual images can always be seen as subclasses inside the classes $\mathbf{D}_i$ we call this organization a *hierarchical database*. Of course, there can be multiple levels in the hierarchical organization of a database. *In all the theoretical derivations of the thesis we assume that the images are already classified. For experiments we always rely on a flat database structure*. The issue of automatically grouping the images in the database, or *indexing*, is not addressed.

Associated with an image database there is a space $\mathcal{Z} \subset R^n$ of *image observations*. An image observation $\mathbf{z} = \{z_1, \ldots z_n\}$ is a vector containing n pixel colors extracted from an image. The *region of support* of observation $\mathbf{z}$ is the set of pixels in $\mathcal{L}$ whose colors are represented in $\mathbf{z}$. It can be a single pixel ($n = c$) or any number b of them ($n = cb$). When $b > 1$, the regions of support of different observations can overlap and, consequently, there can be as many observations as there are pixels in the image. A *feature transformation* is a map

$$T : \mathcal{Z} \rightarrow \mathcal{X}$$

from the space of image observations into some other space $\mathcal{X}$ deemed more appropriate to the retrieval operation. We call $\mathcal{X}$ the *feature space*, and $\mathbf{x} = T(\mathbf{z})$ a *feature vector*. *Features* are the elements of a feature vector and feature vectors inherit the region of support of the observations from which they are derived. If the feature transformation is the identity, then $\mathcal{Z}$ and $\mathcal{X}$ are the same.

A *feature representation* is a probabilistic model for how each of the image classes populates the feature space $\mathcal{X}$. We introduce a *class indicator* variable $Y \in \{1, \ldots, M\}$ and denote the *class-conditional probability density function (pdf)* or *class-conditional likelihood* associated with class i by $P_{\mathbf{X}|Y}(\mathbf{X} = \mathbf{x}|Y = i)$. This can be any non-negative function integrating to one. Throughout the thesis, we use upper case for random variables and lower case for particular values, e.g. $\mathbf{X} = \mathbf{x}$ denotes that the random variable $\mathbf{X}$ takes the value $\mathbf{x}$. When the meaning is clear from context, we usually omit one of the symbols. For example, $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ is commonly used instead of $P_{\mathbf{X}|Y}(\mathbf{X} = \mathbf{x}|Y = i)$. Boldface type is used to represent vectors.

One density that we will encounter frequently is the Gaussian, defined by a mean vector μ and a positive-definite covariance matrix $\mathbf{\Sigma}$ according to

$$\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}|}} e^{-\frac{1}{2}||\mathbf{x}-\mu||^2_{\mathbf{\Sigma}}} \tag{2.5}$$

where

$$||\mathbf{x} - \mu||_{\mathbf{\Sigma}} = (\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu) \tag{2.6}$$

is the quadratic norm defined by $\mathbf{\Sigma}^{-1}$. The Euclidean norm is the particular case in which $\mathbf{\Sigma} = \mathbf{I}$. When $\mathbf{\Sigma} = \sigma \mathbf{I}$ and $\sigma \to 0$ the Gaussian converges to the Dirac function [123] defined by

$$\int \delta(\mathbf{x} - \mathbf{x}_0) f(\mathbf{x}) d\mathbf{x} = f(\mathbf{x}_0), \tag{2.7}$$

for all continuous functions $f(\mathbf{x})$.

Together, a *feature transformation* and a *feature representation* determine an *image representation*. An *image representation* and a *similarity function* define a *retrieval system*. This is a system that accepts queries from a user and searches a database for images that best match those queries. A *visual query* $\mathbf{x}$ is a collection of N feature vectors $\{\mathbf{x}_j\}_{j=1}^N$ extracted from a *query image*. If the the union of the regions of support of these feature vectors covers the entire lattice $\mathcal{L}$ the query is denoted as *global*. Otherwise, it is denoted as *local*. Local queries can be assembled through a graphical interface, by allowing a user to select a region or collection of regions from the query image. Throughout the thesis we rely on the following independence assumptions.

Assumption 1 *The feature vectors* $\{\mathbf{x}_j\}_{j=1}^N$ *included in a visual query are independent and* *identically distributed (iid)*

$$P_{\mathbf{X}_1,\ldots,\mathbf{X}_N}(\mathbf{x}_1,\ldots,\mathbf{x}_N) = \prod_{j=1}^{N} P_{\mathbf{X}}(\mathbf{x}_j).$$

Assumption 2 *Given the knowledge of the true image class the query feature vectors $\{\mathbf{x}_j\}_{j=1}^N$ are independent*

$$P_{\mathbf{X}_j|Y,\mathbf{X}_1\ldots\mathbf{X}_{j-1},\mathbf{X}_{j+1},\ldots,\mathbf{X}_N}(\mathbf{x}_j|i,\mathbf{x}_1\ldots\mathbf{x}_{j-1},\mathbf{x}_{j+1},\ldots,\mathbf{x}_N) = P_{\mathbf{X}|Y}(\mathbf{x}_j|i)$$

By application of the chain rule of probability, Assumption 2 is equivalent to

$$P_{\mathbf{X}_1\ldots\mathbf{X}_N|Y}(\mathbf{x}_1\ldots\mathbf{x}_N|i) = \prod_{j=1}^{N} P_{\mathbf{X}|Y}(\mathbf{x}_j|i) \qquad (2.8)$$

Given these definitions, we are now ready to address the questions posed by the design of a retrieval system. We start by considering the question of image similarity.

2.2 A Bayesian criteria for image similarity

In the image retrieval context, image similarity is naturally formulated as a problem of statistical classification. Given the feature space $\mathcal{X}$, a retrieval system is simply a map

$$\begin{aligned} g: \quad \mathcal{X} &\rightarrow \{1,\ldots,M\} \\ \mathbf{x} &\mapsto y \end{aligned}$$

from $\mathcal{X}$ to the index set of the M classes in the database. It is relatively common, in the vision and retrieval literatures, to define this map up-front without a clear underlying justification. For example, the most popular retrieval solution is to minimize the distance between color histograms[2] [172, 72, 49, 96, 2, 121, 193, 168, 149, 126, 43, 167, 163]. It is not clear that when confronted with the question "what would you like a retrieval system to do?" a naive user would reply "minimize histogram distance." In this work we define a more intuitive goal, the *minimization of probability of retrieval error*; i.e. we design systems that strive to be wrong as rarely as possible.

Definition 1 *A retrieval system is a map*

$$g : \mathcal{X} \rightarrow \{1, \ldots, M\}$$

that minimizes

$$P_{\mathbf{X},Y}(g(\mathbf{X}) \neq Y)$$

i.e. the system that has the minimum probability of returning images from a class $g(\mathbf{x})$ *different than that to which the query* $\mathbf{x}$ *belongs.*

Formulating the problem in this way has various advantages. First, the desired goal is stated explicitly, making clear what the retrieval operation is trying to achieve. Second, the criteria is objective leading to concrete metrics for evaluating the retrieval performance. Finally, it allows us to build on a relatively good theoretical understanding of the properties of various types of solutions (e.g. if their performance converges or not to that of the optimal solution and how quickly it does so) that are already in place for similar problems. In fact, once the problem is formulated in this way, the optimal solution is well known [38, 39, 48].

Theorem 1 *Given a feature space* $\mathcal{X}$ *and a query* $\mathbf{x}$*, the similarity function that minimizes the probability of retrieval error is the Bayes or maximum a posteriori (MAP) classifier*

$$g^*(\mathbf{x}) = \arg\max_i P_{Y|\mathbf{X}}(i|\mathbf{x}). \qquad (2.9)$$

Furthermore, the probability of error is lower bounded by the Bayes error

$$L^* = 1 - E_{\mathbf{X}}[\max_i P_{Y|\mathbf{X}}(i|\mathbf{x})], \qquad (2.10)$$

[2] We will give a precise definition of the term color histogram later on.

where $E_{\mathbf{X}}$ means expectation with respect to $P_{\mathbf{X}}(\mathbf{x})$.

Proof: The proof can be found in various textbooks (see [38, 48] among many others). We include it here because 1) it is simple, and 2) provides insights for some later results.

The probability of error associated with the decision rule $g(\mathbf{x})$ is

$$P_{\mathbf{X},Y}(g(\mathbf{X}) \neq Y) = \int P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x})P_{\mathbf{X}}(\mathbf{x})d\mathbf{x} = E_{\mathbf{x}}[P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x})], \tag{2.11}$$

where

$$\begin{aligned} P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x}) &= \sum_i P(Y \neq g(\mathbf{X})|\mathbf{X} = \mathbf{x}, Y = i)P_{Y|\mathbf{X}}(i|\mathbf{x}) \\ &= \sum_i (1 - \delta_{g(\mathbf{x}),i})P_{Y|\mathbf{X}}(i|\mathbf{x}) \\ &= 1 - \sum_i \delta_{g(\mathbf{x}),i}P_{Y|\mathbf{X}}(i|\mathbf{x}) \end{aligned} \tag{2.12}$$

and $\delta_{i,j}$ is the Kronecker delta function defined in (2.3). It follows that

$$\begin{aligned} P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x}) &\geq 1 - \max_i P_{Y|\mathbf{X}}(i|\mathbf{x}) \\ &= 1 - P_{Y|\mathbf{X}}(Y = g^*(\mathbf{x})|\mathbf{x}) \\ &= P_{Y|\mathbf{X}}(Y \neq g^*(\mathbf{x})|\mathbf{x}) \end{aligned}$$

and, consequently

$$E_{\mathbf{x}}[P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x})] \geq E_{\mathbf{x}}[P_{Y|\mathbf{X}}(Y \neq g^*(\mathbf{x})|\mathbf{x})].$$

I.e., any other decision rule will have a larger probability of error than the Bayes classifier. Since, from (2.11),

$$P_{\mathbf{X},Y}(g^*(\mathbf{X}) \neq Y) = 1 - E_{\mathbf{x}}[P_{Y|\mathbf{X}}(Y = g^*(\mathbf{x})|\mathbf{x})] = 1 - E_{\mathbf{x}}[\max_i P_{Y|\mathbf{X}}(i|\mathbf{x})] = L^*$$

the probability of error can never be smaller than the Bayes error. □

The *posterior probabilities* $P_{Y|\mathbf{X}}(i|\mathbf{x})$ are in general not easy to compute, making the direct implementation of this theorem difficult. To cope with this difficulty, several alternative approaches to the classification problem have been proposed in the now extensive classification literature. At the coarsest level, one can divide them into two major categories: *discriminant classifiers* and *classifiers based on generative models*.

Discriminant classifiers strive to find the surfaces in $\mathcal{X}$ that better separate the regions associated with the different classes in the sense of Theorem 1, classifying each point according to its position relative to those surfaces. Examples in this set are linear discriminant classifiers [39], neural networks [13], decision trees [18], and support vector machines [184], among others. From the retrieval point of view, discriminant classifiers have very limited interest because they must be retrained every time an image class is added to or deleted from the database. This is a strong restriction in the retrieval scenario, where databases can change daily or at an even faster pace.

Instead of dealing directly with (2.9), classifiers based on generative models take the alternative route provided by Bayes rule,

$$P_{Y|\mathbf{X}}(i|\mathbf{x}) = \frac{P_{\mathbf{X}|Y}(\mathbf{x}|i)P_Y(i)}{P_{\mathbf{X}}(\mathbf{x})}, \tag{2.13}$$

which leads to

$$g^*(\mathbf{x}) = \arg\max_i P_{\mathbf{X}|Y}(\mathbf{x}|i)P_Y(i)$$

When the query feature vectors $\{\mathbf{x}_j\}$ are iid, from (2.8)

$$\begin{aligned} g^*(\mathbf{x}) &= \arg\max_i \prod_{j=1}^{N} P_{\mathbf{X}|Y}(\mathbf{x}_j|i)P_Y(y=i) \\ &= \arg\max_i \sum_{j=1}^{N} \log P_{\mathbf{X}|Y}(\mathbf{x}_j|i) + \log P_Y(i), \end{aligned} \tag{2.14}$$

where $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ is the class-conditional likelihood for the i^{th} class and $P_Y(i)$ a *prior probability* for this class.

In the recent past, this similarity function has become prevalent for the evaluation of speech similarity and achieved significant success in tasks such as speech recognition and speaker identification [140, 145]. This is interesting because, if we can show that it also has good properties for visual similarity, we will have a common framework for dealing with images and sound. Also, because the individual likelihood functions $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ are learned for each image class independently, these classifiers can adapt easily to class additions and deletions. We denote (2.14) by *Bayesian retrieval criteria* and will refer to image retrieval based on it as *Bayesian retrieval*, *probabilistic retrieval*, or retrieval based on *Bayesian similarity*.

In practice, the probability of error of Bayesian retrieval is usually larger than the Bayes error. This is due to the fact that we do not know the true likelihood function or prior for each classes, and these have to be estimated from 1) images available in the database and 2) prior knowledge about the retrieval problem. We will return to this point in Chapter 3. For now, we analyze the relationships between Bayesian similarity and the similarity functions that are commonly used for image retrieval.

2.3 A unified view of image similarity

Figure 2.1 illustrates how various similarity functions commonly used for image retrieval are special cases of the Bayesian retrieval. While these functions do not exhaust the set of decisions rules that can be derived from or shown to be sub-optimal when compared to the Bayesian criteria (see chapter 3 of [38] for several others), we concentrate on them for two reasons: 1) they *have been* proposed as similarity functions, and 2) when available, derivations of their relationships to Bayesian similarity are scattered around the literature.

The figure illustrates that, if an upper bound on the Bayes error of a collection of two-way classification problems is minimized instead of the probability of error of the original problem, the Bayesian criteria reduces to the *Bhattacharyya distance* (BD). On the other hand, if the original criteria is minimized, but the different image classes are assumed to be equally likely a priori, we have the *maximum likelihood* (ML) retrieval criteria. As the number of query vectors grows to infinity the ML criteria tends to the *minimum discrimination information* (MDI), which in turn can be approximated by the χ^2 test by performing a simple first order Taylor series expansion. Alternatively, MDI can be simplified by assuming that the underlying probability densities belong to a pre-defined family. For *auto-regressive sources* it reduces to the *Itakura-Saito* distance that has received significant attention in the speech literature. In the Gaussian case, further assumption of orthonormal covariance matrices leads to the *quadratic distance* (QD) frequently found in the compression literature. The next possible simplification is to assume that all classes share the same covariance matrix, leading to the *Mahalanobis distance* (MD). Finally, assuming identity covariances results in the square of the *Euclidean distance* (ED). We next derive in more detail all these relationships.

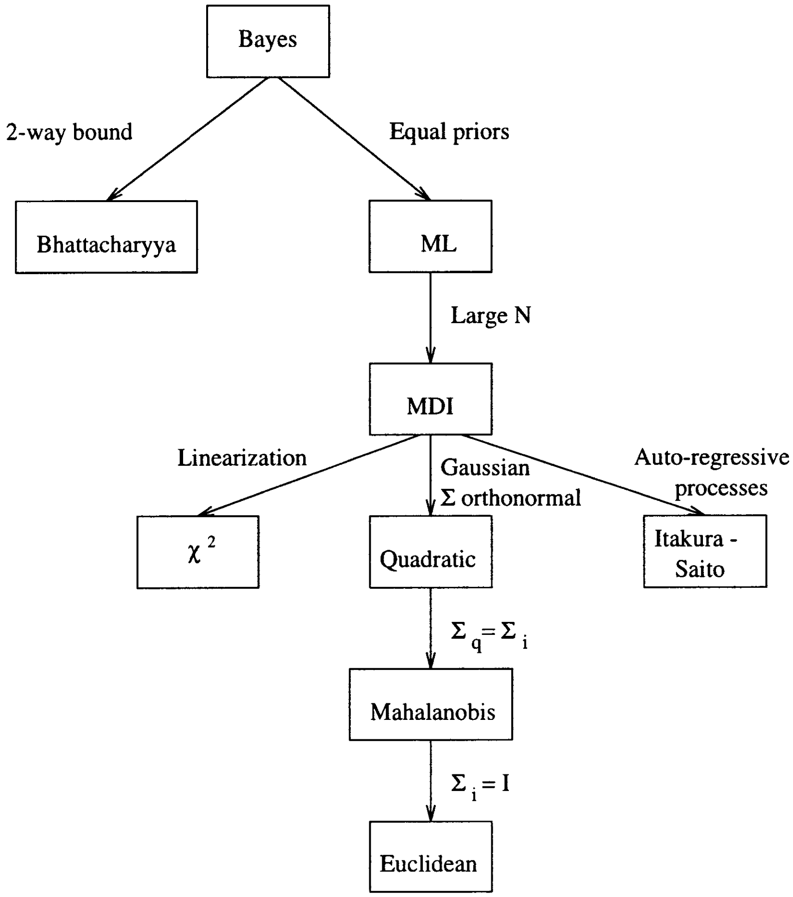


Figure 2.1: Relations between different image similarity functions.

2.3.1 Bhattacharyya distance

If there are only two classes in the classification problem, (2.10) can be written as [48]

$$L^* \;=\; E_{\mathbf{X}}[\min(P_{Y|\mathbf{X}}(0|\mathbf{x}), P_{Y|\mathbf{X}}(1|\mathbf{x}))]$$

$$
\begin{aligned}
&= \int P_{\mathbf{X}}(\mathbf{x}) \min[P_{Y|\mathbf{X}}(0|\mathbf{x}), P_{Y|\mathbf{X}}(1|\mathbf{x})] d\mathbf{x} \\
&= \int \min[P_{\mathbf{X}|Y}(\mathbf{x}|0) P_Y(0), P_{\mathbf{X}|Y}(\mathbf{x}|1) P_Y(1)] d\mathbf{x} \\
&\leq \sqrt{P_Y(0) P_Y(1)} \int \sqrt{P_{\mathbf{X}|Y}(\mathbf{x}|0) P_{\mathbf{X}|Y}(\mathbf{x}|1)} d\mathbf{x},
\end{aligned}
$$

where we have used the bound $\min[a, b] \leq \sqrt{ab}$. The last integral is usually known as the Bhattacharyya distance between $P_{\mathbf{X}|Y}(\mathbf{x}|0)$ and $P_{\mathbf{X}|Y}(\mathbf{x}|1)$ and has been proposed (e.g. [111, 30]) for image retrieval where, for a query density $P_{\mathbf{X}}(\mathbf{x})$, it takes the form

$$g(\mathbf{x}) = \arg\min_i \int \sqrt{P_{\mathbf{X}}(\mathbf{x}) P_{\mathbf{X}|Y}(\mathbf{x}|i)} d\mathbf{x}. \tag{2.15}$$

The resulting classifier can thus be seen as the one which finds the lowest upper-bound on the Bayes error for the collection of two-class problems involving the query and each of the database classes.

Whenever it is possible to solve the minimization of the error probability on the multiclass retrieval problem it makes small sense to replace it by the search for the two class problem with the smallest error bound. Consequently, the above interpretation of the BD makes it clear that, in general, there is small justification to prefer it to Bayesian retrieval.

2.3.2 Maximum likelihood

It is straightforward to see that when all image classes are equally likely a priori, $P_Y(i) = 1/M$, (2.14) reduces to

$$g(\mathbf{x}) = \arg\max_i \frac{1}{N} \sum_{j=1}^{N} \log P_{\mathbf{X}|Y}(\mathbf{x}_j|i). \tag{2.16}$$

This decision rule is usually referred to as the maximum likelihood classifier. While, as we will see after Chapter 6, class priors $P_Y(i)$ provide a useful mechanism to 1) account for the context in which the retrieval operation takes place, 2) integrate information from multiple content modalities that may be available in the database, and 3) design learning algorithms, in Chapters 2-6 we assume that there is no a priori reason to prefer any given image over the rest. In this case, Bayesian and maximum likelihood retrieval are equivalent and we will use the two terms indiscriminately.

2.3.3 Minimum discrimination information

If $H_i, i = 1, 2$, are the hypotheses that $\mathbf{x}$ is drawn from the statistical population with density $P_i(\mathbf{x})$, the *Kullback-Leibler divergence* (KLD) or *relative entropy* [83, 31]

$$KL[P_2(\mathbf{x})||P_1(\mathbf{x})] = \int P_2(\mathbf{x}) \log \frac{P_2(\mathbf{x})}{P_1(\mathbf{x})} d\mathbf{x} \tag{2.17}$$

measures the mean information per observation from $P_2(\mathbf{x})$ for discrimination in favor of H_2 against H_1. Because it measures the difficulty of discriminating between the two populations, and is always non-negative and equal to zero only when $P_1(\mathbf{x}) = P_2(\mathbf{x})$ [83], the KLD has been proposed as a measure of similarity for various compression and signal processing problems [59, 42, 86, 41].

Given a density $P_1(\mathbf{x})$ and a family of densities $\mathcal{M}$ the minimum discrimination information criteria [83] seeks the density in $\mathcal{M}$ that is the "nearest neighbor" of $P_1(\mathbf{x})$ in the KLD sense

$$P_2^*(\mathbf{x}) = \arg \min_{P_2(\mathbf{x}) \in \mathcal{M}} KL[P_2(\mathbf{x})||P_1(\mathbf{x})].$$

If $\mathcal{M}$ is a large family, containing $P_1(\mathbf{x})$, this problem has the trivial solution $P_2(\mathbf{x}) = P_1(\mathbf{x})$, which is not always the most interesting. In other cases, a sample from $P_2(\mathbf{x})$ is available but the explicit form of the distribution is not known. In these situations it may be more useful to seek for the distribution that minimizes the KLD subject to a stricter set of constraints. Kullback suggested the problem

$$P_2^*(\mathbf{x}) = \arg \min_{P_2(\mathbf{x}) \in \mathcal{M}} KL[P_2(\mathbf{x})||P_1(\mathbf{x})]$$

subject to

$$\int T(\mathbf{x}) P_2(\mathbf{x}) = \theta$$

where $T(\mathbf{x})$ is a measurable statistic (e.g. the mean when $T(\mathbf{x}) = \mathbf{x}$) and θ can be computed from a sample (e.g. the sample mean). He showed that the minimum is 1) achieved by

$$P_2^*(\mathbf{x}) = \frac{1}{Z} e^{-\lambda T(\mathbf{x})} P_1(\mathbf{x})$$

where Z is a normalizing constant, $Z = \int e^{-\lambda T(\mathbf{x})} P_1(\mathbf{x}) d\mathbf{x}$, and λ a Lagrange multiplier [11] that weighs the importance of the constraint; and 2) equal to

$$KL[P_2^*(\mathbf{x})||P_1(\mathbf{x})] = -\lambda\theta - \log Z.$$

Gray and his colleagues have studied extensively the case in which $P_1(\mathbf{x})$ belongs to the family of *auto-regressive moving average* (ARMA) processes [59, 42] and showed, among other things, that in this case the optimal solution is a variation of the Itakura-Saito distance commonly used in speech analysis and compression. Kupperman [84, 83] has shown that when all densities are members of the exponential family (a family that includes many of the common distributions of interest such as the Gaussian, Poisson, binomial, Rayleigh and exponential among others [39]), the constrained version of MDI is equivalent to maximum likelihood.

The KLD has only been recently considered in the retrieval literature [192, 189, 70, 139, 16], where attention has focused on the unconstrained MDI problem

$$g(\mathbf{x}) = \arg\min_i KL[P_{\mathbf{X}}(\mathbf{x})||P_{\mathbf{X}|Y}(\mathbf{x}|i)], \tag{2.18}$$

where $P_{\mathbf{X}}(\mathbf{x})$ is the density of the query and $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ that of the i^{th} image class. Similarly to the constrained case, it is possible to derive a connection between unconstrained MDI and maximum likelihood. However, the connection is much stronger in the unconstrained case since there is no need to make any assumptions regarding the type of densities involved. In particular, by simple application of the law of large numbers to (2.16),

$$\begin{aligned} g(\mathbf{x}) &= \arg\max_i E_{\mathbf{X}}[\log P_{\mathbf{X}|Y}(\mathbf{x}|i)] \quad \text{as } N \to \infty \\ &= \arg\max_i \int P_{\mathbf{X}}(\mathbf{x}) \log P_{\mathbf{X}|Y}(\mathbf{x}|i) d\mathbf{x} \\ &= \arg\min_i \int P_{\mathbf{X}}(\mathbf{x}) \log P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} - \int P_{\mathbf{X}}(\mathbf{x}) \log P_{\mathbf{X}|Y}(\mathbf{x}|i) d\mathbf{x} \end{aligned}$$

$$
\begin{aligned}
&= \arg\min_i \int P_{\mathbf{X}}(\mathbf{x}) \log \frac{P_{\mathbf{X}}(\mathbf{x})}{P_{\mathbf{X}|Y}(\mathbf{x}|i)} d\mathbf{x} \\
&= \arg\min_i KL[P_{\mathbf{X}}(\mathbf{x})||P_{\mathbf{X}|Y}(\mathbf{x}|i)],
\end{aligned}
$$

where $E_{\mathbf{X}}$ is the expectation with respect to the query density $P_{\mathbf{X}}(\mathbf{x})$. This means that, independently of the type of densities, MDI is simply the asymptotic limit of the ML criteria as the cardinality of the query tends to infinity[3]. This relationship is important for various reasons. First, it confirms that the Bayesian criteria converges to a meaningful global similarity function as the cardinality of the query grows. Second, it makes it clear that while ML and MDI perform equally well for global queries, the Bayesian criteria has the added advantage of also enabling local queries. Third, while the Bayesian criteria has complexity $O(N)$, as we will see in Chapter 6, for most densities of practical interest MDI either has a much reduced complexity or can be approximated by functions with that property. In practice, by switching to MDI when the size of the query exceeds a given threshold, this allows the complexity of Bayesian retrieval to always remain manageable. Finally, it establishes a connection between the Bayesian criteria and several similarity functions that can be derived from MDI.

2.3.4 χ^2 test

The first of such similarity functions is the χ^2 statistic. Using a first order Taylor series approximation for the logarithmic function about $x = 1$, $\log(x) \approx x - 1$, we obtain[4]

$$
\begin{aligned}
KL[P_1(\mathbf{x})||P_2(\mathbf{x})] &= \int P_1(\mathbf{x}) \log \frac{P_1(\mathbf{x})}{P_2(\mathbf{x})} d\mathbf{x} \\
&\approx \int \frac{P_1(\mathbf{x})^2 - P_1(\mathbf{x})P_2(\mathbf{x})}{P_2(\mathbf{x})} d\mathbf{x} \\
&= \int \left(\frac{P_1(\mathbf{x})^2 - P_1(\mathbf{x})P_2(\mathbf{x})}{P_2(\mathbf{x})} - P_1(\mathbf{x}) + P_2(\mathbf{x}) \right) d\mathbf{x} \\
&= \int \frac{(P_1(\mathbf{x}) - P_2(\mathbf{x}))^2}{P_2(\mathbf{x})} d\mathbf{x},
\end{aligned}
$$

[3] Notice that this result only holds when the true distribution is that of the query. The alternative version of the divergence, where the distribution of the database image class is assumed to be true, does not have an interpretation as the asymptotic limit of a local metric of similarity.

[4] This result is stated without proof in [31].

where we have used the fact that $\int P_i(\mathbf{x})d\mathbf{x} = 1, i = 1, 2$. In the retrieval context, this means that MDI can be approximated by

$$g(\mathbf{x}) \approx \arg\min_i \int \frac{(P_{\mathbf{X}}(\mathbf{x}) - P_{\mathbf{X}|Y}(\mathbf{x}|i))^2}{P_{\mathbf{X}|Y}(\mathbf{x}|i)} d\mathbf{x}. \tag{2.19}$$

The integral on the right is known as the χ^2 statistic and the resulting criteria a χ^2 test [124]. It has been proposed as a metric for image similarity in [157, 16, 139]. Since it results from the linearization of the KLD, it can be seen as an approximation to the asymptotic limit of the ML criteria. Obviously, this linearization can discard a significant amount of information and there is, in general, no reason to believe that it should perform better than Bayesian retrieval.

2.3.5 The Gaussian case

Several similarity functions of practical interest can be derived from the Bayesian retrieval criteria when the class likelihoods are assumed to be Gaussian. We now analyze the relationships for three such functions: the quadratic, Mahalanobis, and Euclidean distances. Given the asymptotic convergence of ML to MDI, these results could also been derived from the expression for the KLD between two Gaussians [83], by replacing expectations with respect to the query distribution by sample means.

Quadratic distance

When the image features are Gaussian distributed, (2.16) becomes

$$\begin{aligned} g(\mathbf{x}) &= \arg\min_i \log|\mathbf{\Sigma}_i| + \frac{1}{N}\sum_n (\mathbf{x}_n - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}_n - \mu_i) \\ &= \arg\min_i \log|\mathbf{\Sigma}_i| + \hat{\mathcal{L}}_i, \end{aligned} \tag{2.20}$$

where

$$\hat{\mathcal{L}}_i = \frac{1}{N}\sum_n (\mathbf{x}_n - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}_n - \mu_i)$$

is the *quadratic distance* (QD) commonly found in the perceptually weighted compression literature [53, 89, 119, 92]. As a retrieval metric, the QD can thus be seen as the result of imposing two stringent restrictions on the generic ML criteria. First, that all image sources are Gaussian and, second, that their covariance matrices are orthonormal ($|\mathbf{\Sigma}_i| = 1, \forall i$).

Mahalanobis distance

Furthermore, because

$$
\begin{aligned}
\hat{\mathcal{L}}_i = \frac{1}{N}\sum_n (\mathbf{x}_n - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}_n - \mu_i) & \\
= & \frac{1}{N}\sum_n (\mathbf{x}_n - \hat{\mathbf{x}} + \hat{\mathbf{x}} - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}_n - \hat{\mathbf{x}} + \hat{\mathbf{x}} - \mu_i) \\
= & \frac{1}{N}\sum_n (\mathbf{x}_n - \hat{\mathbf{x}})^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}_n - \hat{\mathbf{x}}) - 2(\hat{\mathbf{x}} - \mu_i)^T \mathbf{\Sigma}_i^{-1}\frac{1}{N}\sum_n (\mathbf{x}_n - \hat{\mathbf{x}}) + (\hat{\mathbf{x}} - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T \\
= & \frac{1}{N} trace[\mathbf{\Sigma}_i^{-1}\sum_n (\mathbf{x}_n - \hat{\mathbf{x}})(\mathbf{x}_n - \hat{\mathbf{x}})^T] + (\hat{\mathbf{x}} - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T \\
= & trace[\mathbf{\Sigma}_i^{-1}\hat{\mathbf{\Sigma}}_{\mathbf{x}}] + (\hat{\mathbf{x}} - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T \\
= & trace[\mathbf{\Sigma}_i^{-1}\hat{\mathbf{\Sigma}}_{\mathbf{x}}] + \mathcal{M}_i, \qquad (2.21)
\end{aligned}
$$

where

$$\hat{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n$$

is the sample mean of $\mathbf{x}_n$

$$\hat{\mathbf{\Sigma}}_{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n - \hat{\mathbf{x}})(\mathbf{x}_n - \hat{\mathbf{x}})^T$$

the sample covariance and

$$\mathcal{M}_i = (\hat{\mathbf{x}} - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T$$

the Mahalanobis distance, we see that the MD results from complementing Gaussianity with the assumption that all classes have the same covariance ($\mathbf{\Sigma}_{\mathbf{x}} = \mathbf{\Sigma}_i = \mathbf{\Sigma}, \forall i$).

Euclidean distance

Finally, if this covariance is the identity ($\mathbf{\Sigma} = \mathbf{I}$), we obtain the square of the Euclidean distance (ED) or *mean squared error*

$$\mathcal{E}_i = (\hat{\mathbf{x}} - \mu_i)^T(\hat{\mathbf{x}} - \mu_i). \tag{2.22}$$

The MD, the ED, and variations on both, have been widely used in the retrieval literature [163, 24, 96, 43, 166, 153, 118, 158, 134, 102, 193, 129, 65, 15, 160, 139, 72, 195, 175, 150, 96, 7].

Some intuition for the advantages of Bayesian retrieval

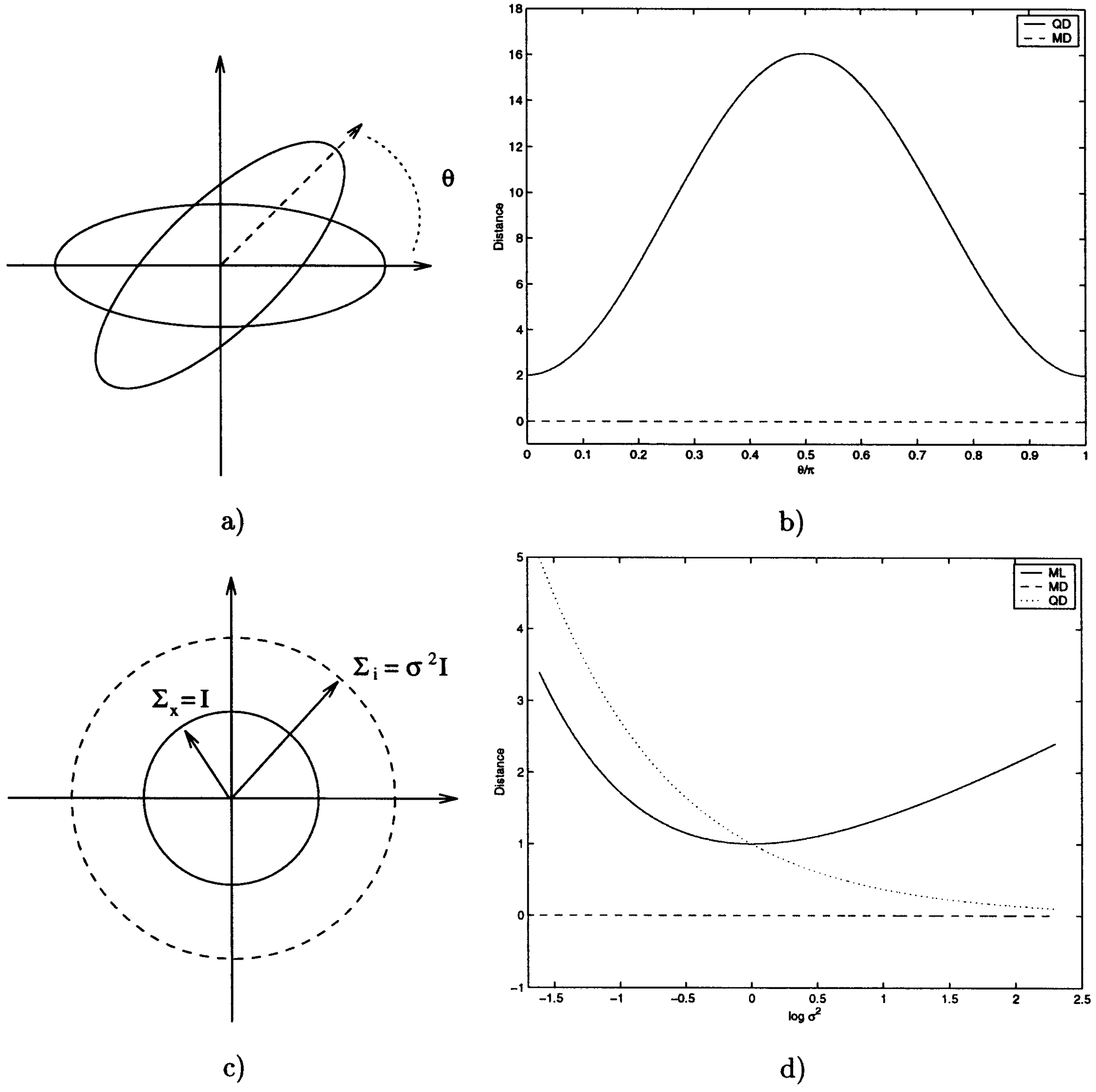
The Gaussian case is a good example of why, even if minimization of error probability is not considered to be the right goal for an image retrieval system, there seems to be little justification to rely on any criteria for image similarity other than the Bayesian. Recall that, under Bayesian retrieval, the similarity function is

$$g(\mathbf{x}) = \arg\min_i \log|\mathbf{\Sigma}_i| + \overbrace{trace[\mathbf{\Sigma}_i^{-1}\hat{\mathbf{\Sigma}}_{\mathbf{x}}] + \underbrace{(\hat{\mathbf{x}} - \mu_i)^T\mathbf{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T}_{\text{MD}}}^{\text{QD}} \tag{2.23}$$

and all three other criteria are approximations that arbitrarily discard covariance information.

As illustrated by Figure 2.2, this information is important for the detection of subtle variations such as rotation and scaling in feature space. In a) and b), we show the distance, under both QD and MD between a Gaussian and a replica rotated by $\theta \in [0, \pi]$. Plot b) clearly illustrates that while the MD has no ability to distinguish between the rotated Gaussians, the inclusion of the $trace[\mathbf{\Sigma}_i^{-1}\hat{\mathbf{\Sigma}}_{\mathbf{x}}]$ term leads to a much more intuitive measure of similarity: minimum when both Gaussians are aligned and maximum when they are rotated by $\pi/2$.

As illustrated by c) and d), further inclusion of the term $\log|\mathbf{\Sigma}_i|$ (full ML retrieval) penalizes mismatches in scaling. In plot c), we show two Gaussians, with covariances



a) b)

c) d)

Figure 2.2: a) A Gaussian with mean $(0,0)^T$ and covariance $diag(4, 0.25)$ and its replica rotated by θ. b) Distance between the Gaussian and its rotated replicas as a function of θ/π under both the QD and the MD. c) Two Gaussians with different scales. d) Distance between them as a function of $\log \sigma^2$ under ML, QD, and MD.

$\mathbf{\Sigma_x} = \mathbf{I}$ and $\mathbf{\Sigma}_i = \sigma^2\mathbf{I}$, centered on zero. In this example, MD is always zero, while $trace[\mathbf{\Sigma}_i^{-1}\hat{\mathbf{\Sigma}}_\mathbf{x}] \propto 1/\sigma^2$ penalizes small σ and $\log|\mathbf{\Sigma}_i| \propto \log\sigma^2$ penalizes large σ. The total distance is shown as a function of $\log\sigma^2$ in plot d) where, once again, we observe an intuitive behavior: the penalty is minimal when both Gaussians have the same scale ($\log\sigma^2 = 0$), increasing monotonically with the amount of scale mismatch. Notice that if the $\log|\mathbf{\Sigma}_i|$ term is not included, large changes in scale may not be penalized at all.

2.3.6 L^p norms

Despite all the nice properties discussed above, probabilistic retrieval has received small attention in the context of CBIR. An overwhelmingly more popular metric of global similarity is the L^p norm of the difference between densities

$$g(\mathbf{X}) = \arg\min_i \left(\int_{\mathcal{F}} |P_\mathbf{X}(\mathbf{x}) - P_{\mathbf{X}|Y}(\mathbf{x}|i)|^p d\mathbf{x} \right)^{\frac{1}{p}}. \tag{2.24}$$

These norms are particularly common in the color-based retrieval literature as metrics of similarity between color histograms.

The *histogram* of a collection of feature vectors $\mathbf{X}$ is a vector $\mathbf{f} = \{f_1, \ldots, f_R\}$ associated with a partition of the feature space $\mathcal{X}$ into R regions $\{\mathcal{X}_1, \ldots, \mathcal{X}_R\}$, where f_r is the number of vectors in $\mathbf{X}$ landing on cell $\mathcal{X}_r$. Assuming a feature space of dimension n and rectangular cells of size $h_1 \times \ldots \times h_n$, the histogram provides an estimate of the feature probability density of the form

$$P(\mathbf{X}) = \sum_k \frac{f_k}{F}\mathcal{K}(\mathbf{x} - \mathbf{c}_k), \tag{2.25}$$

where $\mathbf{c}_k$ is the central point of the k^{th} cell, F the total number of feature vectors, and $\mathcal{K}(\mathbf{x})$ a pdf such that

$$\begin{aligned}
&\mathcal{K}(\mathbf{x}) > 0, \text{ if } |\mathbf{x}_1| < \frac{h_1}{2}, \ldots, |\mathbf{x}_n| < \frac{h_n}{2}, \\
&\mathcal{K}(\mathbf{x}) = 0, \text{ otherwise}, \\
&\int \mathcal{K}(\mathbf{x})d\mathbf{x} = 1.
\end{aligned}$$

Defining $\mathbf{q}$ to be the histogram of Q query vectors, $\mathbf{p}^i$ the histogram of P^i vectors from the i^{th} image class, and substituting (2.25) into (2.24)

$$\begin{aligned}
g(\mathbf{X}) &= \arg\min_i \left(\int_{\mathcal{X}} |\sum_r \left(\frac{q_r}{Q} - \frac{p_r^i}{P^i} \right) \mathcal{K}(\mathbf{x} - \mathbf{c}_r)|^p d\mathbf{x} \right)^{\frac{1}{p}} \\
&= \arg\min_i \left(\int_{\mathcal{X}} \sum_r \left| \frac{q_r}{Q} - \frac{p_r^i}{P^i} \right|^p \mathcal{K}(\mathbf{x} - \mathbf{c}_r) d\mathbf{x} \right)^{\frac{1}{p}} \\
&= \arg\min_i \left(\sum_r \left| \frac{q_r}{Q} - \frac{p_r^i}{P^i} \right|^p \int_{\mathcal{X}_r} \mathcal{K}(\mathbf{x} - \mathbf{c}_r) d\mathbf{x} \right)^{\frac{1}{p}} \\
&= \arg\min_i \left(\sum_r \left| \frac{q_r}{Q} - \frac{p_r^i}{P^i} \right|^p \right)^{\frac{1}{p}}, \qquad (2.26)
\end{aligned}$$

where we have used the fact that the cells $\mathcal{X}_r$ are disjoint and $\mathcal{K}(\mathbf{x})$ integrates to one. As shown in [172], assuming that the histograms are normalized ($\sum_r q_r/Q = \sum_r p_r^i/P^i = 1, \forall i$), the minimization of the L^1 distance is equivalent to the maximization of the *histogram intersection* (HI)

$$g(\mathbf{X}) = \arg\max_i \frac{\sum_r \min(q_r, p_r^i)}{Q}, \qquad (2.27)$$

a similarity function that has become the de-facto standard for color-based retrieval [172, 139, 149, 96, 72, 150, 163, 164, 125, 68, 44, 167, 43, 168, 17].

It is clear that, while (2.16) minimizes the classification error, (2.24) implies that minimizing pointwise similarity between density estimates should be the ultimate retrieval criteria. Clearly, for any of the two criteria to work, it is necessary that the estimates be close to the true densities. However, it is known (e.g. see Theorem 6.5 of [38]) that the probability of error of rules of the type of (2.16) tends to the Bayes error orders of magnitude faster than the associated density estimates tend to the right distributions. This implies that accurate density estimates are not required everywhere for the classification criteria to work.

In fact, accuracy is required only in the regions near the boundaries between the different classes, because these are the only regions that matter for the classification decisions. On the other hand, the criteria of (2.24) is clearly dependent on the quality of the density estimates all over $\mathcal{X}$. It, therefore, places a much more stringent requirement on the quality of these estimates and, since density estimation is know to be a difficult problem [184, 162], there seems to be no reason to believe that it is a better retrieval criteria than (2.16). We next validate these theoretical claims through retrieval experiments on real image databases.

2.4 Experimental evaluation

A series of retrieval experiments was conducted to evaluate the performance of the ML criteria as a global similarity function. Since implementing all the similarity functions discussed above was an extensive amount of work, we selected the two most popular representatives: the Mahalanobis distance for texture-based and the histogram intersection for color-based retrieval. In order to isolate the contribution of the similarity function from those of the features and the feature representation, the comparison was performed with the feature sets and representations that are commonly used for each of the domains: color-based retrieval was implemented by combining the color histogram with (2.16) and texture-based retrieval by the combination of the features derived from the *multi-resolution simultaneous auto-regressive* (MRSAR) model[5] [104] with (2.23).

The MRSAR features were computed using a window of size 21 × 21 sliding over the image with increments of two pixels in both the horizontal and vertical dimensions. Each feature vector consists of 4 SAR parameters plus the error of the fit achieved by the SAR model at three resolutions, in a total of 15 dimensions. This is a standard implementation of this model [104, 94, 102]. For color histogramming, the 3D YBR color space was quantized by finding the bounding box for all the points in the query and retrieval databases and then dividing each axis in b bins. This leads to b^3 cells. Experiments were performed with different values of b.

Figure 2.3 presents precision/recall curves for the Brodatz and Columbia databases. As expected, texture-based retrieval (MRSAR/MD) performs better on Brodatz while color-based retrieval (color histogramming) does better on Columbia. Furthermore, due to their lack of spatial support, histograms do poorly on Brodatz while, being a model specific for texture, MRSAR does poorly on Columbia[6].

[5] See the appendix for a more detailed justification for the use of the MRSAR features as a benchmark.

[6] Notice that this would not be evident if we were only looking at classification accuracy, i.e. the percentage of retrievals for which the first match is from the correct class.

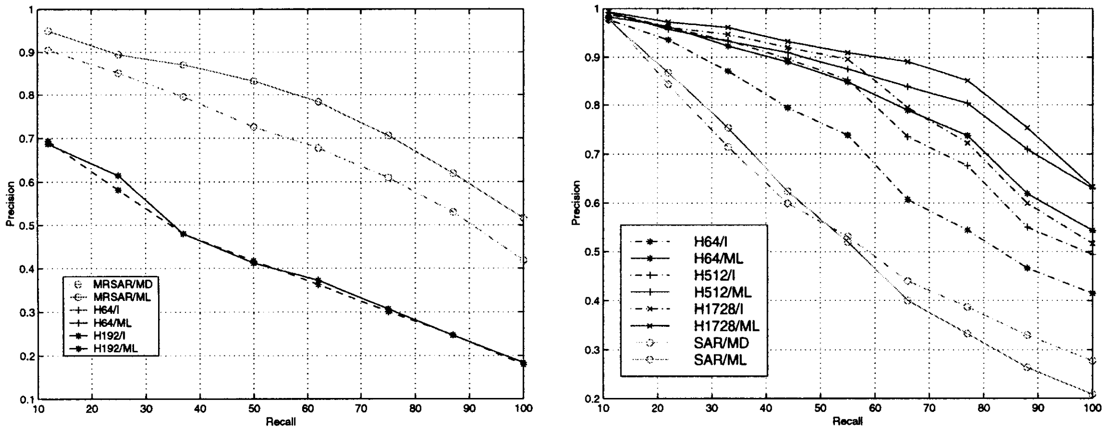


Figure 2.3: Precision/recall curves for Brodatz (left) and Columbia (right). In the legend, MRSAR means MRSAR features, H color histograms, ML maximum likelihood, MD Mahalanobis distance, and I intersection. The total number of bins in each histogram is indicated after the H.

More informative is the fact that, when the correct features and representation are used for the specific database, the ML criteria always leads to a clear improvement in retrieval performance. In particular, for the texture database, combining ML with the MRSAR features and the Gaussian representation leads to an improvement in precision from 5 to 10% (depending on the level of recall) over that achievable with the Mahalanobis distance. Similarly, on Columbia, replacing histogram intersection by the ML criteria leads to an improvement that can be as high as 20%[7].

The latter observation validates the arguments of section 2.3.6, where we saw that, while the ML criteria only depends on the class boundaries, HI measures pointwise distances between densities. This means that whenever there is a change in the imaging parameters (lighting, shadows, object rotation, etc) and the densities change slightly, the impact on HI will be higher than on ML. An example is given in Figure 2.4 where we present the results of the same query under the two similarity criteria. Notice that as the object is rotated, the relative percentages of the different colors in the image change. HI changes accordingly

[7] Notice that, for these databases, 100% recall means retrieving the 8 or 9 images in the same class as the query, and it is important to achieve high precision at this level. This may not be the case for databases with hundreds of images in each class, since it is unlikely that users may want to look at that many images.

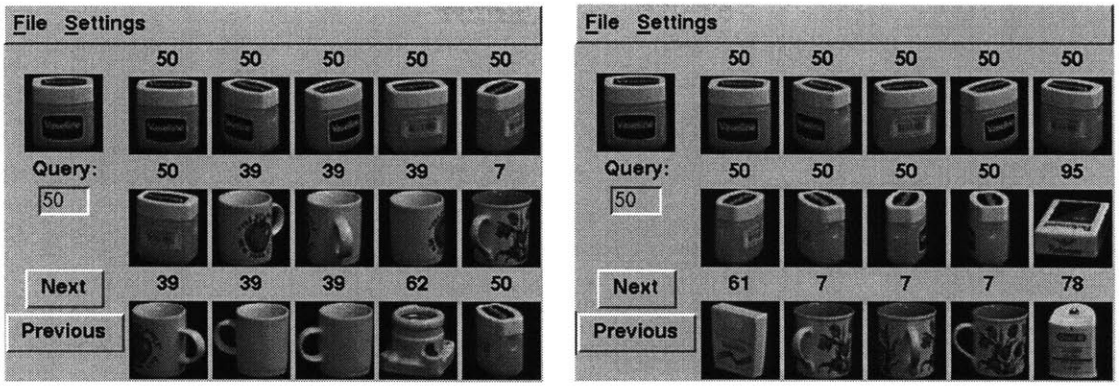


Figure 2.4: Results for the same query under HI (left) and ML (right). In both images, the query is shown in the top left corner, and the returned images in raster-scan order (left to right, top to bottom) according to their similarity rank. The numbers displayed above the retrieved images indicate the class to which they belong.

and, when the degree of rotation is significant, views of other objects are preferred. On the other hand, because the color of each individual pixel is always better explained by the density of the rotated object than by those of other objects, ML achieves a perfect retrieval. This increased invariance to changes in imaging conditions explains why, for large recall, the precision of ML is consistently and significantly higher than that of HI.

Chapter 3

Image representations for retrieval

Numerous image representations have been proposed for image compression [74, 56, 128, 63, 73], object recognition [180, 62], texture analysis [144, 178, 152, 62, 61, 21] and, more recently, content-based retrieval [149, 2, 133, 132, 131]. Because we are interested in generic imagery (i.e. we want to make as few assumptions as possible regarding the content of the images under analysis) and it is still too difficult to segment such images in a semantically meaningful way, we will not consider here any representations that require segmentation either implicitly or explicitly. This includes many representations that are common in vision [115, 179, 77, 130, 194] and most of the ones used for shape-based retrieval [149].

In order to simplify the understanding of the remaining representations, it is useful to further decompose them into the two main components discussed in section 2.1: a feature transformation and a feature representation. In this chapter, we show that minimization of the probability of error, and the resulting Bayesian solution to the retrieval problem, provide us with concrete guidelines for the selection of feature spaces and representations. Interpretation of the strategies in current use according to these guidelines leads to insights about their major limitations and lays the ground for a better solution, that we will pursue in subsequent chapters.

3.1 Bayesian guidelines for image representation

In Chapter 2, we saw that one of the interesting properties of Bayesian retrieval is that it is optimal with respect to the minimization of error probability. In practice, however, good results can only be guaranteed if it is possible to achieve a probability of error close to the Bayes error. In this section, we look for theoretical guidelines that can help us achieve this goal.

3.1.1 Feature transformation

We start by analyzing the impact of a feature transformation on the overall probability of error.

Theorem 2 *Given a retrieval system with observation space $\mathcal{Z}$ and a feature transformation*

$$T : \mathcal{Z} \rightarrow \mathcal{X},$$

the Bayes error on $\mathcal{X}$ can never be smaller than that on $\mathcal{Z}$. I.e.,

$$L^*_{\mathcal{X}} \geq L^*_{\mathcal{Z}}$$

where $L^*_{\mathcal{Z}}$ and $L^*_{\mathcal{X}}$ are, respectively, the Bayes errors on $\mathcal{Z}$ and $\mathcal{X}$. Furthermore, equality is achieved if and only if T is an invertible transformation.

Proof: The following proof is a straightforward extension to multiple classes of the one given in [38] for the two-class problem. From (2.10),

$$\begin{aligned}
L^*_{\mathcal{X}} &= 1 - E_{\mathbf{X}}[\max_i P_{Y|\mathbf{X}}(i|\mathbf{x})], \\
&= 1 - E_{T(\mathbf{z})}[\max_i P_{Y|\mathbf{X}}(i|T(\mathbf{z}))], \\
&= 1 - E_{T(\mathbf{z})}[\max_i \int P_{Y|\mathbf{Z},\mathbf{X}}(i|\mathbf{z}, T(\mathbf{z}))P_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|T(\mathbf{z}))d\mathbf{z}], \\
&= 1 - E_{T(\mathbf{z})}[\max_i \int P_{Y|\mathbf{Z}}(i|\mathbf{z})P_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|T(\mathbf{z}))d\mathbf{z}], \\
&= 1 - E_{T(\mathbf{z})}[\max_i E_{\mathbf{z}|\mathbf{X}}[P_{Y|\mathbf{Z}}(i|\mathbf{z})|\mathbf{X} = T(\mathbf{z})]],
\end{aligned}$$

$$\begin{aligned}&\geq 1 - E_{T(\mathbf{z})}[E_{\mathbf{z}|\mathbf{X}} \max_i [P_{Y|\mathbf{Z}}(i|\mathbf{z})|\mathbf{X} = T(\mathbf{z})]],\\ &= 1 - E_{\mathbf{z}}[\max_i P_{Y|\mathbf{Z}}(i|\mathbf{z})] = L^*_{\mathcal{Z}},\end{aligned}$$

where we have used Jensen's inequality [31], and equality is achieved if and only if T is an invertible map.□

This theorem tells us that the choice of feature transformation is very relevant for the performance of a retrieval system. In particular, 1) any transformation can only increase or, at best, maintain the Bayes error achievable in the space of image observations, and 2) the only transformations that maintain the Bayes error are the invertible ones.

3.1.2 Feature representation

While a necessary condition, low Bayes error is not sufficient for accurate retrieval since the actual error may be much larger than the lower bound. The next theorem provides an upper bound for this difference.

Theorem 3 *Given a retrieval system with a feature space $\mathcal{X}$, unknown class probabilities $P_Y(i)$ and class conditional likelihood functions $P_{\mathbf{X}|Y}(\mathbf{x}|i)$, and a decision function*

$$g(\mathbf{x}) = \arg\max_i \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)\hat{p}_Y(i), \tag{3.1}$$

the actual probability of error is upper bounded by

$$P(g(\mathbf{X}) \neq Y) \leq L^*_{\mathcal{X}} + \sum_i \int |P_{\mathbf{X}|Y}(\mathbf{x}|i)P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)\hat{p}_Y(i)|d\mathbf{x}. \tag{3.2}$$

Proof: From (2.11),

$$P_{\mathbf{X},Y}(g(\mathbf{X}) \neq Y) - L^*_{\mathcal{X}} = \int [P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x}) - P_{Y|\mathbf{X}}(Y \neq g^*(\mathbf{x})|\mathbf{x})]P_{\mathbf{X}}(\mathbf{x})d\mathbf{x} \tag{3.3}$$

and since, $\forall \mathbf{x} \in \mathcal{X}$ such that $g(\mathbf{x}) = g^*(\mathbf{x})$, we have

$$P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x}) = P_{Y|\mathbf{X}}(Y \neq g^*(\mathbf{x})|\mathbf{x}),$$

this is equivalent to

$$P_{\mathbf{X},Y}(g(\mathbf{X}) \neq Y) - L^*_{\mathcal{X}} = \int_E [P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x}) - P_{Y|\mathbf{X}}(Y \neq g^*(\mathbf{x})|\mathbf{x})] P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \qquad (3.4)$$

where

$$E = \{\mathbf{x}|\mathbf{x} \in \mathcal{X}, P_{\mathbf{X}}(\mathbf{x}) > 0, g(\mathbf{x}) \neq g^*(\mathbf{x})\}.$$

Letting

$$\Delta(\mathbf{x}) = P_{Y|\mathbf{X}}(Y \neq g(\mathbf{x})|\mathbf{x}) - P_{Y|\mathbf{X}}(Y \neq g^*(\mathbf{x})|\mathbf{x})$$

and defining the sets

$$E^*_i = \{\mathbf{x}|\mathbf{x} \in E, g^*(\mathbf{x}) = i\}$$

$$E_i = \{\mathbf{x}|\mathbf{x} \in E, g(\mathbf{x}) = i\},$$

it follows from (2.12) that, $\forall \mathbf{x} \in E^*_i \cap E_j$,

$$\Delta(\mathbf{x}) = P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}).$$

Since, from (2.9),

$$P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}) \geq 0 \ \forall \mathbf{x} \in E^*_i, \forall j \neq i$$

from (3.1) and the fact that $P_{\mathbf{X}}(\mathbf{x}) > 0 \, \forall \mathbf{x} \in E$,

$$\frac{\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|j)\hat{p}_Y(j)}{P_{\mathbf{X}}(\mathbf{x})} - \frac{\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)\hat{p}_Y(i)}{P_{\mathbf{X}}(\mathbf{x})} \geq 0 \ \forall \mathbf{x} \in E_j, \forall i \neq j,$$

defining

$$\hat{p}_{Y|\mathbf{X}}(i|\mathbf{x}) = \frac{\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)\hat{p}_Y(i)}{P_{\mathbf{X}}(\mathbf{x})},$$

we have, $\forall \mathbf{x} \in E^*_i \cap E_j$,

$$\begin{aligned} \Delta(\mathbf{x}) &= P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}) \\ &\leq P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}) + \hat{p}_{Y|\mathbf{X}}(j|\mathbf{x}) - \hat{p}_{Y|\mathbf{X}}(i|\mathbf{x}) \end{aligned}$$

$$\begin{aligned} &= |P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}) + \hat{p}_{Y|\mathbf{X}}(j|\mathbf{x}) - \hat{p}_{Y|\mathbf{X}}(i|\mathbf{x})| \\ &\leq |P_{Y|\mathbf{X}}(i|\mathbf{x}) - \hat{p}_{Y|\mathbf{X}}(i|\mathbf{x})| + |P_{Y|\mathbf{X}}(j|\mathbf{x}) - \hat{p}_{Y|\mathbf{X}}(j|\mathbf{x})| \end{aligned}$$

and

$$\begin{aligned} \int_{E_i^* \cap E_j} \Delta(\mathbf{x}) P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \;\leq\; & \int_{E_i^* \cap E_j} |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} \\ + & \int_{E_i^* \cap E_j} |P_{\mathbf{X}|Y}(\mathbf{x}|j) P_Y(j) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|j) \hat{p}_Y(j)| d\mathbf{x}. \end{aligned}$$

Using the fact that both collections of sets E_i^* and E_j partition E, we obtain

$$\begin{aligned} \int_E \Delta(\mathbf{x}) P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} &= \sum_{i,j} \int_{E_i^* \cap E_j} \Delta(\mathbf{x}) P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \\ &\leq \sum_i \int_{E_i^*} |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} + \\ &\quad \sum_j \int_{E_j} |P_{\mathbf{X}|Y}(\mathbf{x}|j) P_Y(j) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|j) \hat{p}_Y(j)| d\mathbf{x} \\ &= \sum_i \left[\int_{E_i^*} |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} \right. \\ &\quad \left. + \int_{E_i} |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} \right] \\ &\leq \sum_i \int |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} \end{aligned}$$

where we have also used the fact that $E_i^* \cap E_i = \emptyset$. □

This theorem states that, if the Bayes error is small, accurate density estimation is a sufficient condition for high retrieval accuracy. In particular, good density estimation will suffice to guarantee optimal performance when the feature transformation is the identity.

3.2 Strategies for image representation

Together the two theorems are a convenient tool to analyze the balance between feature transformation and representation achieved by any retrieval strategy. We now proceed to do so for the two predominant strategies in the literature.

3.2.1 The color strategy

The theorems suggest that all that really matters for accurate retrieval is good density estimation. Since no feature transformation can reduce the Bayes error, there seems to be no advantage in using one. This is the rationale behind Strategy 1 (S1): *avoid feature transformations altogether and do all the estimation directly in $\mathcal{Z}$.* As Figure 3.1 illustrates, the main problem with this strategy is that density estimation can be difficult in $\mathcal{Z}$. Significant emphasis must therefore be given to the feature representation which is required to rely on a sophisticated density model. One possible solution, that has indeed become a de-facto standard for color-based retrieval [172, 139, 149, 96, 72, 150, 163, 164, 125, 68, 44, 167, 43, 168, 17], is the histogram. This solution is illustrated in Figure 3.1 b).

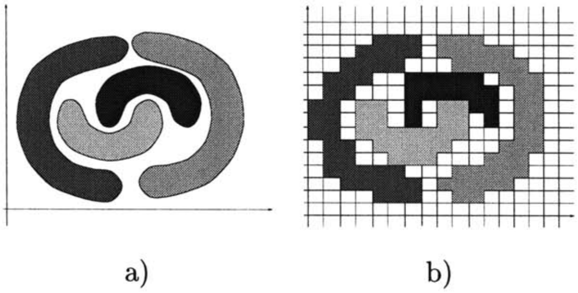


Figure 3.1: Example of a retrieval problem with four image classes. a) In the space of image observations, the class densities can have complicated shapes. b) Strategy 1 is to simply model the class densities as accurately as possible.

3.2.2 The texture strategy

Since accurate density estimation is usually a difficult problem [184, 162, 39], a feature transformation can be helpful if it makes estimation significantly easier in $\mathcal{X}$ than what it is in $\mathcal{Z}$. The rationale behind Strategy 2 (S2) is to exploit this as much as possible: *find a feature transformation that clearly separates the image classes in $\mathcal{X}$, rendering estimation trivial.* Ideally, in $\mathcal{X}$, each class should be characterized by a simple parametric density, such as the Gaussians in Figure 3.2, and a simple classifier should be able to guarantee performance close to the Bayes error.

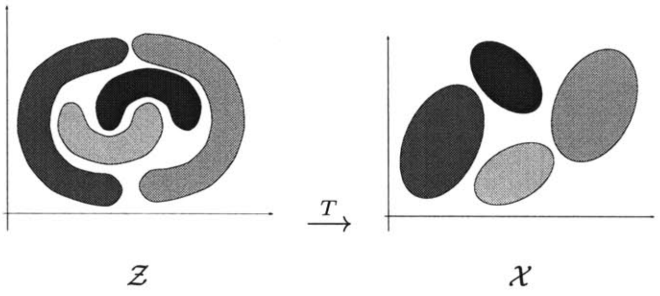


Figure 3.2: Example retrieval problem with four image classes. Strategy 2 is to find a feature transformation such that density estimation is much easier in $\mathcal{X}$ than in $\mathcal{Z}$.

Strategy S2 has become prevalent in the texture literature, where numerous feature transformations have been proposed to achieve *good discrimination* between different texture classes [163, 24, 96, 134, 102, 137, 40, 104, 178, 144, 174, 21, 176]. These transformations are then combined with simple similarity functions, like the Mahalanobis and Euclidean distances or variations of these, that assume Gaussianity in $\mathcal{X}$. More recently it has also been embraced by many retrieval systems [15, 118, 175, 150, 139, 153, 163, 96, 129, 7].

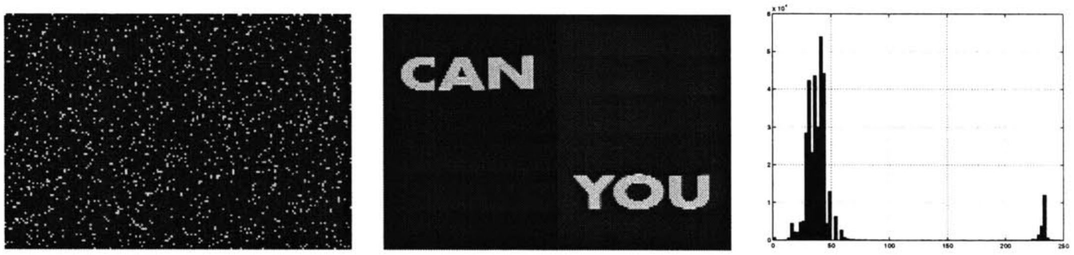
3.2.3 A critical analysis

Overall, none of the two strategies is consistently better than the other. While S1 has worked better for object recognition and color-based retrieval, S2 has proven more effective for the databases used by the texture community. Unfortunately, none of the two strategies is viable when the goal is to jointly model color and texture in the context of generic image databases.

Limitations of strategy S1

While it works reasonably well when $\mathcal{Z}$ is a low-dimensional space, e.g. the 3-D space of pixel colors, S1 is of very limited use in high dimensions. This is a consequence of the well known curse of dimensionality: in higher dimensions, modeling requires more parameters and more data is required to achieve accurate estimation. Typically these relationships are non-linear. For example, the number of elements in the covariance matrix of a Gaussian is quadratic in the dimension of the space, and the number of cells in the histogram model increases exponentially with it[1].

In particular, for c color channels and observations with b pixels, the dimension of $\mathcal{Z}$ is $n = cb$. Hence, the complexity is at least linear and, in the case of the histogram exponential, in the size of the region of support of the observations. Consequently, accurate joint density estimates can only be obtained over very small spatial neighborhoods and the resulting representations cannot capture the spatial dependencies that are crucial for fine image discrimination. This is illustrated by Figure 3.3 where we present two images that, although visually very dissimilar, are characterized by the same histogram [shown in c)]. In order to distinguish between these images, the representation must capture the fact that while on b) the white pixels cluster spatially, the same does not happen on a). This is an impossible task if the measurements do not have spatial support, e.g. the pixel colors commonly used under S1.



a) b) c)

Figure 3.3: a) An homogeneous and b) a non-homogeneous image that are visually dissimilar but have the same color histogram, shown in c).

Of course, there is no law stating that histograms cannot be computed in high dimensions, but in practice it is impossible to guarantee that the upper bound of Theorem 3 remains close to the Bayes error.

[1] Assuming that the number of divisions in each coordinate axis is held constant.

Limitations of strategy S2

For strategy S2, the main problem is the assumption that it is always possible to find a transformation that maps a collection of complicated densities in $\mathcal{Z}$ into a collection of simple densities in $\mathcal{X}$, without compromising Bayes error. The following theorem shows that, for multi-modal class-conditional densities, this is not possible with a generic feature transformation.

Theorem 4 *Consider a retrieval system with observation space $\mathcal{Z}$. If there exists a feature transformation T*

$$T : \mathcal{Z} \rightarrow \mathcal{X}$$

that preserves the Bayes error

$$L^*_{\mathcal{Z}} = L^*_{\mathcal{X}} \tag{3.5}$$

and maps a multi-modal density $P_{\mathbf{Z}|Y}(\mathbf{z}|i)$ on $\mathcal{Z}$ into a unimodal density $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ on $\mathcal{X}$ then 1) T is non-linear, and 2) T depends on $P_{\mathbf{Z}|Y}(\mathbf{z}|i)$.

Proof: From Theorem 2, (3.5) only holds if T is invertible, in which case [114]

$$det\left[J(\mathbf{z})\right] \neq 0 \;\; \forall \mathbf{z}$$

where $J(\mathbf{z})$ the Jacobian of T evaluated at $\mathbf{z}$

$$J_{i,j}(\mathbf{z}) = [D_{\mathbf{z}}T(\mathbf{z})]_{i,j} = \frac{\partial T_i}{\partial \mathbf{z}_j}(\mathbf{z}) \tag{3.6}$$

and $D_{\mathbf{z}}T(\mathbf{z})$ the vector derivative[2] of $T(\mathbf{z})$ with respect to $\mathbf{z}$. It follows, from the change of variables theorem [124], that the densities in $\mathcal{Z}$ and $\mathcal{X}$ are related by

$$P_{\mathbf{X}|Y}(T(\mathbf{z})|i) = det\left[J^{-1}(\mathbf{z})\right] P_{\mathbf{Z}|Y}(\mathbf{z}|i). \tag{3.7}$$

[2] Several definitions have been proposed for the vector derivative. The one adopted here, equation (3.6), is that used in [114].

If T is linear $T(\mathbf{z}) = \mathbf{Az}$ then $J(\mathbf{z}) = \mathbf{A}$ and, up to a scale factor, the two densities are equal

$$P_{\mathbf{Z}|Y}(T(\mathbf{z})|i) = det\left[\mathbf{A}^{-1}\right] P_{\mathbf{Z}|Y}(\mathbf{z}|i).$$

Hence, if $P_{\mathbf{Z}|Y}(\mathbf{z}|i)$ is multi-modal then so is $P_{\mathbf{X}|Y}(T(\mathbf{z})|i)$. This proves the first part of the theorem. If T is non-linear, by taking derivatives on both sides of (3.7)

$$\begin{aligned} D_{\mathbf{z}}\left[det[J^{-1}(\mathbf{z})]P_{\mathbf{Z}|Y}(\mathbf{z}|i)\right] &= D_{\mathbf{z}}P_{\mathbf{X}|Y}(T(\mathbf{z})|i) \\ &= D_{\mathbf{x}}P_{\mathbf{X}|Y}(\mathbf{x}|i)\Big|_{\mathbf{x}=T(\mathbf{z})} J(\mathbf{z}) \end{aligned}$$

and

$$D_{\mathbf{x}}P_{\mathbf{X}|Y}(\mathbf{x}|i)\Big|_{\mathbf{x}=T(\mathbf{z})} = D_{\mathbf{z}}\left[det[J^{-1}(\mathbf{z})]P_{\mathbf{Z}|Y}(\mathbf{z}|i)\right] J^{-1}(\mathbf{z}), \tag{3.8}$$

from which $D_{\mathbf{x}}P_{\mathbf{X}|Y}(\mathbf{x}|i)|_{\mathbf{x}=T(\mathbf{z})} = \mathbf{0}$ if and only if $D_{\mathbf{z}}\left[det[J^{-1}(\mathbf{z})]P_{\mathbf{Z}|Y}(\mathbf{z}|i)\right]$ is in the null space of $J^{-1}(\mathbf{z})$. Since $J(\mathbf{z})$ has full rank,

$$D_{\mathbf{x}}P_{\mathbf{X}|Y}(\mathbf{x}|i)\Big|_{\mathbf{x}=T(\mathbf{z})} = \mathbf{0} \Leftrightarrow D_{\mathbf{z}}\left[det[J^{-1}(\mathbf{z})]P_{\mathbf{Z}|Y}(\mathbf{z}|i)\right] = \mathbf{0}.$$

It follows that, if $\mathbf{x} = T(\mathbf{z})$ is the maximum of $P_{\mathbf{X}|Y}(\mathbf{x}|i)$, then

$$D_{\mathbf{z}}(det[J^{-1}(\mathbf{z})])P_{\mathbf{Z}|Y}(\mathbf{z}|i) + det[J^{-1}(\mathbf{z})]D_{\mathbf{z}}P_{\mathbf{Z}|Y}(\mathbf{z}|i) = 0,$$

$$\frac{1}{det[J^{-1}(\mathbf{z})]}D_{\mathbf{z}}(det[J^{-1}(\mathbf{z})]) = -\frac{1}{P_{\mathbf{Z}|Y}(\mathbf{z}|i)}D_{\mathbf{z}}P_{Z|Y}(\mathbf{z}|i),$$

$$D_{\mathbf{z}}\left[\log\frac{det[J(\mathbf{z})]}{P_{\mathbf{Z}|Y}(\mathbf{z}|i)}\right] = \mathbf{0}.$$

Since the log is a monotonic function, this means that $\frac{det[J(\mathbf{z})]}{P_{\mathbf{Z}|Y}(\mathbf{z}|i)}$ has a critical point at $T^{-1}(\mathbf{x})$. For most parametric densities, $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ only has one critical point, implying that this will be the only critical point of $\frac{det[J(\mathbf{z})]}{P_{\mathbf{Z}|Y}(\mathbf{z}|i)}$. In any case, it follows that T depends on $P_{\mathbf{Z}|Y}(\mathbf{z}|i)$. □

The theorem explains why most texture retrieval approaches work well on databases of homogeneous images (like that of Figure 3.3 a)), but clearly fail when this is not the case. Since the pixel colors of non-homogeneous images (like that of Figure 3.3 b)) have different statistics according to their spatial location, the associated densities are inherently multi-modal. It is therefore impossible to find a generic transformation mapping them into a set of unimodal densities without compromising the Bayes error.

Yet, the vast majority of texture retrieval methods are based on a feature transformations that does not depend on the class conditional pdfs and the Gaussian representation (implicit in quadratic metrics like the Mahalanobis distance) [137, 21, 144, 178, 163, 96, 134, 102, 104, 174]. It is therefore not surprising that they cannot guarantee low Bayes error in $\mathcal{X}$. While data-dependent transformations have been proposed in the literature [40, 176], these usually imply finding a set of *discriminant* features that can only be computed by considering all the image classes simultaneously. This is impossible in the CBIR context since 1) there may be too many classes, and 2) the feature transformation has to be recomputed every time the database changes.

Putting it plainly, the theorem states that there is no such thing as a "free lunch". If we want to rely on simple models for density estimation, we will necessarily have to rely on a complicated feature transformation. And, in the end, the complexity of finding such a transformation may very well be orders of magnitude greater than that required by more sophisticated density estimation. Why then has the texture community been so focused on the question of finding good features for texture characterization? One possible explanation is that this is an historical consequence of the assumption that different textures can always be cleanly segmented and a texture classifier will operate on homogeneous texture patches[3].

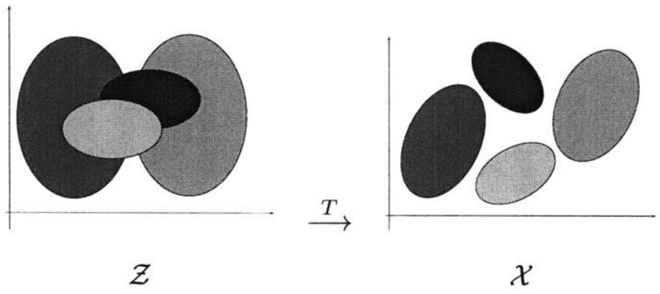


Figure 3.4: When the classes are Gaussian in $\mathcal{Z}$, a feature transformation can help by reducing their overlap in $\mathcal{X}$.

[3] Most of the databases used to evaluate texture recognition are indeed composed of homogeneous images.

Since, by definition, homogeneous images have similar statistics everywhere, the densities of their observations are close to unimodal and any sensible feature transformation will generate unimodal densities in $\mathcal{X}$. For example, any linear transformation will generate a collection of Gaussians in $\mathcal{X}$ if the class-conditional pdfs are already Gaussian in $\mathcal{Z}$. In this case, as illustrated by Figure 3.4, a feature transformation can allow significant improvements in classification accuracy by making the classes in $\mathcal{X}$ more clearly separated than they are in $\mathcal{Z}$.

In practice, however, it is arguable that the segmentation problem can be cleanly solved before recognition. In fact, it it may never be possible to guarantee that the classifier will process samples from unimodal distributions. In this case, Theorem 4 shows that strategy S2 is hopeless as long as one insists on preserving the Bayes error. Unfortunately, unless this is the case, there is no guarantee that good performance in $\mathcal{X}$ will imply good accuracy in $\mathcal{Z}$, the ultimate goal of the retrieval system.

3.3 An alternative strategy

In the context of minimizing probability of error, the two standard strategies can be seen as two ends of a continuum: while strategy S1 is intransigent with respect to any loss in Bayes error and therefore asks too much from the feature representation; strategy S2 constrains the representation to trivial models, expecting the feature transformation to do the impossible.

It seems that a wiser position would be to stand somewhere in between the two extrema. Since the overall probability of error is upper bounded by the sum of the Bayes and estimation errors, we need to consider the two *simultaneously*. While the crucial requirement for low Bayes error is *invertability* of the feature transformation, the crucial requirement for low estimation error is *low-dimensionality* in $\mathcal{X}$. Since we want $\mathcal{Z}$ to be high-dimensional, the two requirements are conflicting and a trade-off between invertability and dimensionality is required. This means that both the feature transformation and representation have an important role in the overall representation.

On one hand, the feature transformation should provide the *dimensionality reduction* necessary for density estimation to be feasible (but no more). On the other hand, the feature representation should be expressive enough to allow accurate estimates without requiring the dimension of $\mathcal{X}$ to be too low, therefore allowing the transformation to be close to invertible. This is the main idea behind our strategy.

Like strategy S2, we rely on a feature transformation. However, we limit its role to enabling dimensionality reduction; i.e. if we define a feature transformation to be of dimensionality reduction level $n - k$ when

$$T : R^n \rightarrow R^k,\ k \leq n,$$

then the *the optimal feature transformation is the one that, for a given level of dimensionality reduction, is as close to invertible as possible*. The idea of *close to invertible transformation* is intimately related to the idea of *semantics-preserving compression* advocated in the design of the Photobook system [129]. Here, we replace the idea of preserving semantics with the simpler and more generic goal of preserving information. It is very difficult to define semantics-preserving transformations without restricting databases to a specific domain or assuming the existence of a perfect segmentation algorithm.

Like strategy S1, we also place strong emphasis on the feature representation. Here, the goal is to guarantee that we will be operating as close to the Bayes error as possible for all levels of dimensionality reduction. In particular, as illustrated by Figure 3.5, we look for the representation that simultaneously satisfies the following requirements:

- like the Gaussian, is computationally tractable in high dimensions;
- like the histogram, can capture the details of multi-modal densities.

In the next chapter, we study the issue of dimensionality reduction. Feature representation is addressed in Chapter 5.

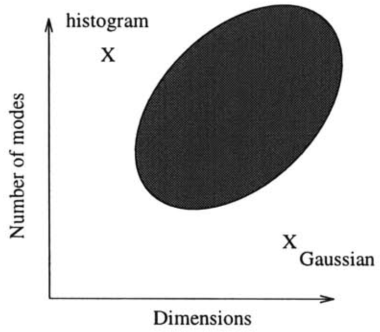


Figure 3.5: The space of feature representations. The histogram can account for multimodal distributions, but is infeasible to compute in high dimensional feature spaces. The Gaussian is unimodal, but can be computed in high dimensions. The shaded area represents the region of the space where new feature representations are needed for the implementation of generic retrieval systems.

Chapter 4

Feature transformation

In addition to making estimation easier, there are a few reasons why dimensionality reduction is a good idea in the context of evaluating image similarity. While, as seen in section 3.2.3, it is important to allow $\mathcal{Z}$ to be high-dimensional, it is also common for the interesting image structure to lie on a lower dimensional manifold [115, 179, 161]. The role of a feature transformation is to expose this manifold, allowing everything else to be discarded. If done right, this can be helpful in various ways.

First, if there is noise associated with the image capture process and the noise is uncorrelated with the image, the signal-to-noise ratio of the representation is usually improved. This happens because most of the noise energy tends to be in the dimensions that are eliminated, while most of the signal energy is in those that are retained. Second, the invariance of the representation to image transformations tends to improve since large regions of the original space are mapped into the same point in the manifold. Finally, because all points in the manifold provide a valid interpretation of the underlying scene, the low-dimensional projection can lead to judgments of similarity that are perceptually more relevant (i.e. intuitive for humans) than what is possible in the high-dimensional space.

For these reasons, most of the feature spaces used for image retrieval involve some form of dimensionality reduction. The interesting question is how to discard dimensions in a way that compromises as little as possible the achievable Bayes error. In this chapter, we argue that multi-resolution feature transformations that are prevalent in the compression literature also have very good properties for retrieval.

4.1 Terms and notation

A *linear transform* is a map

$$A : R^n \to R^n$$

$$\mathbf{z} \mapsto \mathbf{Az}$$

where $\mathbf{A} \in R^{n \times n}$. In the context of this thesis, $\mathbf{z} \in \mathcal{Z} \subseteq R^n$. The row vectors $\mathbf{a_i}$ of $\mathbf{A}$ are known as the *basis functions* of the transform. Since

$$x_i = \mathbf{a}_i \mathbf{z} = \sum_{j=1}^{n} a_{ij}\, z_j,\; i = 1, \ldots, n, \tag{4.1}$$

the components of the transformed vector $\mathbf{x}$ (known as the *transform coefficients*) are nothing more than the projections of $\mathbf{z}$ into these basis functions. When the basis vectors satisfy

$$\mathbf{a}_i^T \mathbf{a}_j = \delta_{i,j}, \tag{4.2}$$

where $\delta_{i,j}$ is the Kronecker delta function (2.3), the transform is said to be *orthonormal.* Orthonormality is a desirable property because the inverse of an orthonormal transformation is very simple to compute. This follows directly from (4.2), since

$$\mathbf{AA}^T = \mathbf{I},$$

where $\mathbf{I}$ is the identity, and thus

$$\mathbf{A}^{-1} = \mathbf{A}^T,$$

i.e. a *unitary* matrix. The *linear projection* of R^n into R^k, for $k < n$, is the map

$$\pi_k : R^n \to R^k$$

$$\mathbf{x} \mapsto \mathbf{\Pi}_k \mathbf{x} \tag{4.3}$$

where $\mathbf{\Pi}_k = [\mathbf{I}_k \, \mathbf{0}_{n-k}]$, $\mathbf{I}_k$ is the $k \times k$ identity matrix and $\mathbf{0}_{n-k}$ a $k \times (n-k)$ matrix of zeros. The *embedding* of R^k into R^n is the map

$$\rho_k : R^k \to R^n$$

$$\mathbf{x} \mapsto \mathbf{\Pi}_k^T \mathbf{x}. \tag{4.4}$$

4.2 Previous approaches

A popular strategy in the retrieval literature for selecting a feature transformation is to simply decide what image properties are important (e.g. textureness, color, edginess, shape, etc.) and define an arbitrary set of features to capture these properties [118, 32, 125, 72, 164, 65, 153, 173, 195, 43, 80]. Other times, the features are selected in a more principled way but predominantly on the basis of a few of the above requirements, e.g. invariance [158, 156, 103, 49] or perceptual relevance [174, 94, 102, 175, 15, 57, 7]. Finally, many times the transformation is selected to provide good discrimination on a specific domain, e.g. texture, object, or faces databases [176, 19, 104, 179, 115, 129, 112].

All these strategies have flaws that are relevant in the context of CBIR. In the first case, it is difficult to know how important is the information that was left out and why other features would not perform better than the ones selected. The answer to this question can only be obtained through extensive experimental evaluation, but so far few exhaustive studies have been conducted [134, 96]. In the second case, it is usually unclear how the selected features perform under the requirements that were not considered for their selection. For example, a representation that has good invariance properties for the smooth surfaces that characterize most object databases may be discarding information that is crucial to characterize texture. Or a representation that captures perceptually relevant attributes for the characterization of texture may be discarding information that is crucial for the perception of faces. In the third case, the resulting features do not even make sense outside the domains for which they were developed.

One solution to these problems is to assemble different feature sets optimized for different criteria and different domains, and build a "society of models"[131, 110, 150]. The combination of multiple models has indeed become prevalent for the design of retrieval systems that can account for both color and texture [43, 118, 44, 164, 153, 175, 32, 101, 72]. However, it has serious drawbacks. First, it implies a significant increase in retrieval complexity since similarity has to be evaluated according to all the models. Second, it is usually not clear how to combine the different representations in order to achieve global inferences. In practice, it frequently requires users to specify weights for the different image attributes, a process that can be extremely non-intuitive. We take the alternate route of modeling the joint density of the image observations over a spatial neighborhood exactly because it avoids these problems.

4.3 Minimizing the reconstruction error

We have already seen that such modeling requires a feature transformation that, for a given level of dimensionality reduction, is as close to invertible as possible. This, in turn, requires a precise definition of "as close to invertible as possible." Following a long tradition in image compression [74, 56, 29], we rely on *linear transformations* and use the minimization of the mean squared reconstruction error as a fidelity criterion.

Definition 2 *A feature transformation T_k provides dimensionality reduction of level $n-k$ if T_k is a map*

$$T_k : R^n \rightarrow R^k$$

defined by

$$T_k = \pi_k \circ A, \tag{4.5}$$

where A is an invertible linear transform.

Definition 3 *The mean squared reconstruction error for a feature transformation T_k defined by (4.5) is*

$$\mathcal{E}_k(\mathbf{z}) = E\left[||\mathbf{z} - (\mathbf{A}^{-1} \circ \rho_k \circ T_k(\mathbf{z}))||^2\right], \tag{4.6}$$

where $||\mathbf{z}||$ is the Euclidean norm of $\mathbf{z}$.

It is well known that this error is minimized by *principal component analysis* (PCA), also known as the *Karhunen-Loeve transform* (KLT), a dimensionality reduction technique that has found wide application in the vision and compression literatures [179, 115, 170, 129, 74, 29, 185]. It consists of finding the eigenvectors $\mathbf{e}_i$ and the eigenvalues λ_i, $i = 1, \ldots, n$, of the sample covariance of $\mathbf{z}$, i.e. the solution to

$$\hat{\mathbf{\Sigma}}_{\mathbf{z}} \mathbf{e}_i = \lambda_i \mathbf{e}_i, \; i = 1, \ldots, n \tag{4.7}$$

where

$$\hat{\mathbf{\Sigma}}_{\mathbf{z}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \hat{\mu}_{\mathbf{z}})(\mathbf{z}_i - \hat{\mu}_{\mathbf{z}})^T,$$

$$\hat{\mu}_{\mathbf{z}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i,$$

and $\{\mathbf{z}_i\}_1^N$ a sample of observations from $\mathbf{z}$, projecting $\mathbf{z}$ onto the eigenvalue basis and discarding the dimensions associated with the smallest eigenvalues. If the eigenvalues are sorted in decreasing order

$$\lambda_1 > \ldots > \lambda_n,$$

this leads to

$$T_k^*(\mathbf{z}) = \pi_k(\mathbf{S}\mathbf{z}) \tag{4.8}$$

where $\mathbf{S} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]^T$ and T_k^* is the optimal feature transformation for a dimensionality reduction of level $n - k$.

In addition to providing optimal dimensionality reduction, PCA has the advantage of generating *decorrelated* coefficients. Notice that, if $\mathbf{x}_i = \mathbf{S}\mathbf{z}_i$, then

$$\hat{\mu}_{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i = \mathbf{S}\,\hat{\mu}_{\mathbf{z}}, \tag{4.9}$$

$$\hat{\mathbf{\Sigma}}_{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{S}(\mathbf{z}_i - \hat{\mu}_{\mathbf{z}})(\mathbf{z}_i - \hat{\mu}_{\mathbf{z}})^T \mathbf{S}^T = \mathbf{S}\hat{\mathbf{\Sigma}}_{\mathbf{z}}\mathbf{S}^T, \tag{4.10}$$

and, from (4.7), $\hat{\mathbf{\Sigma}}_{\mathbf{x}} = diag(\lambda_1, \ldots, \lambda_n)$. In practice, this means a reduction in the complexity of the subsequent density estimation that is larger than the simple dimensionality reduction from n to k. If, for example, a Gaussian model is used, decorrelation is equivalent to diagonal instead of full covariance matrices. This means that there will be k covariance parameters to estimate in $\mathcal{X}$, as opposed to n^2 parameters in $\mathcal{Z}$.

4.4 Discrete Cosine Transform

For simple statistical image models commonly used to evaluate the decorrelating abilities of a feature transformation, such as the first-order Gauss-Markov model, PCA is well approximated by an alternative transformation of lower implementation complexity: the *discrete cosine transform* (DCT). This approximation is particularly good for signals that, like most of the images that we are interested in, exhibit strong pixel correlations [74, 73, 1, 71].

Several definitions of the DCT have been presented in the literature. In this thesis, we will use the one given in [74].

Definition 4 *The 1-D DCT is an orthonormal transform*

$$T : R^n \rightarrow R^n,$$

described by

$$[T(\mathbf{z})]_k = \sqrt{\frac{2}{n}}\alpha_k \sum_{i=0}^{n-1} z_i \cos\frac{(2i+1)k\pi}{2n}, \; k = 0, \ldots, n-1, \tag{4.11}$$

where

$$\alpha_k = \begin{cases} 1/\sqrt{2}, & \text{if } k = 0 \\ 1, & \text{if } k \neq 0. \end{cases} \tag{4.12}$$

By writing (4.11) in matrix form, it can easily be seen that the basis functions of the DCT are

$$\mathbf{t}_k = \sqrt{\frac{2}{N}}\alpha_k \cos\frac{(2i+1)k\pi}{2N}, \; i = 0, \ldots, N-1. \tag{4.13}$$

It is clear that the DCT performs a decomposition of the input signal into a sum of cosines of increasing frequency. In particular, the value of coefficient $T_0(\mathbf{z})$ is the mean or DC value of the input vector and commonly known as the *DC coefficient.* The remaining coefficients, associated with basis vectors of zero-mean, are known as *AC coefficients.* The extension of the 1-D DCT to the 2-D DCT is straightforward: the 2-D DCT is obtained by the separable application of the 1-D DCT to the rows and to the columns of the input image. In this case, the n basis vectors (4.13) are extended to a set of n^2 2-D basis functions that can be obtained by performing the outer products between all the 1-D basis vectors [29]. Figure 4.1 presents these basis functions for $n = 8$.

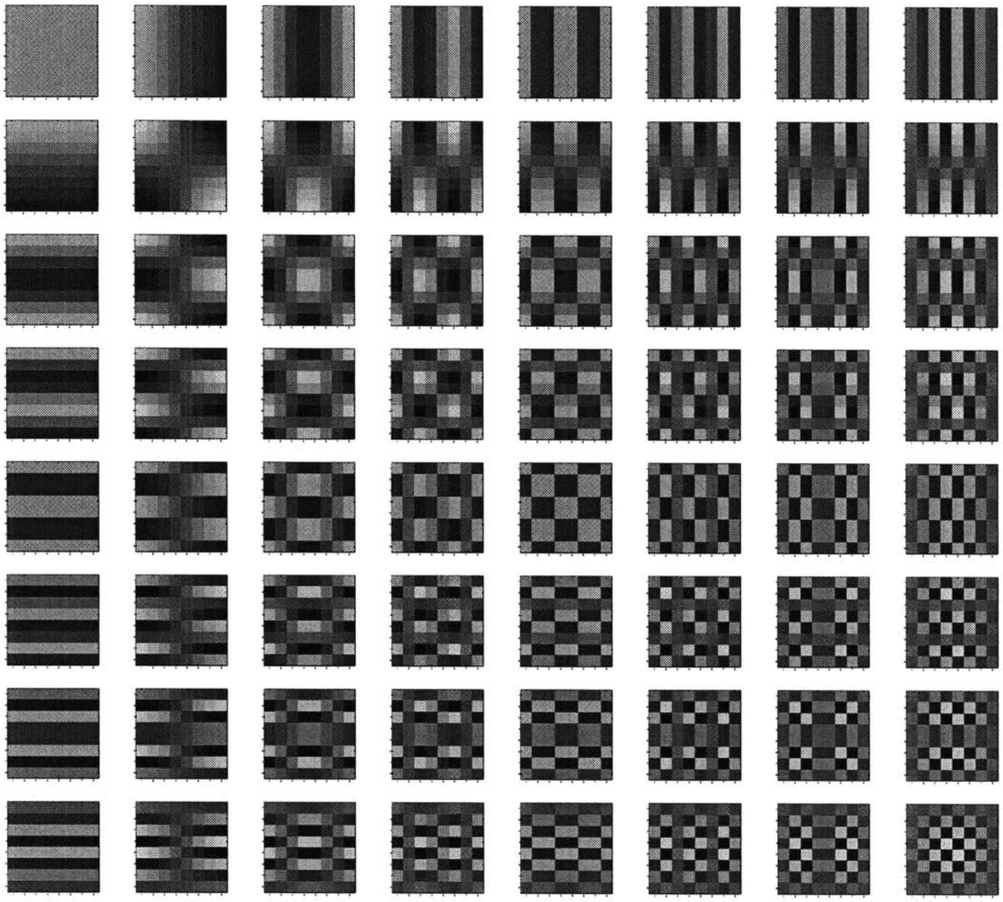


Figure 4.1: Basis Functions of the 2-D DCT of dimension 8.

4.5 Perceptual relevance

Few universal results are known, at this point, about the mechanisms used by the human brain to evaluate image similarity. Ever since the seminal work of Hubel and Wiesel [67], it has been established that 1) processing is local, and 2) different groups in primary visual cortex (i.e. area V1) are tuned for detecting different types of stimulus (e.g. bars, edges, and so on). This indicates that, at the lowest level, the architecture of the human visual system can be well approximated by a multi-resolution representation localized in space and time, and several "biologically plausible" models of early vision are based on this principle [152, 98, 8, 46, 171, 9]. All these models share a basic common structure consisting of three layers: a *space/space-frequency* decomposition at the bottom, a middle stage introducing a non-linearity, and a final stage pooling the responses from several non-linear units.

A space/space-frequency representation is obtained by convolving the image with a collection of elementary filters of reduced spatial support and tuned to different spatial frequencies and orientations. Several elementary filters have been proposed, including *differences of Gaussians* [98], *Gabor functions* [137, 46], and *differences of offset Gaussians* [98]. While Gabor functions seem to provide a better match to actual measurements from the visual cortex [35], the Gabor representation is also the most complex to design [34]. However, there seems to be some agreement in that the exact shape of the filters is not crucial, as long as the representation is localized in space and frequency.

With respect to the non-linearity, at least three different types have been proposed. Denoting by *channel* the output of each filter, these involve intra-channel processing only. If $g(x)$ is the nonlinearity, possible functions are *energy* ($g(x) = x^2$) [9, 46], *full-wave rectification* ($g(x) = |x|$) [8, 171, 46], and *half-wave rectification* ($g(x) = (\max(x, 0), -\min(x, 0))$) [98]. Finally, there is small agreement on the implementation of the final stage other than that it should involve pooling from the individual channel responses.

While biological plausibility is not a constraint for our representation, it is important that it can capture the fundamental characteristics of human visual processing since this is likely to lead to *perceptually* more relevant similarity judgments. The use of the DCT as feature transformation satisfies these requirements because, as illustrated by Figure 4.1, the DCT is localized in both space and frequency. The linear projection to achieve dimensionality reduction is even biologically plausible, since it simply consists of eliminating the filters associated with the frequency/orientation channels to be disregarded.

On the other hand, under the goal of preserving Bayes error, it makes little sense to include a non-invertible linearity, like energy or full-wave rectification, in the model. Malik and Perona have shown that such non-linearities are also not likely to be implemented by the human visual system (by constructing examples of texture pairs where the sign of filter responses is the only property that allows their discrimination by humans) [98]. They suggest half-wave rectification which, being invertible, presents no evidence against our principles. In this case, the need for a non-linearity is simply a consequence of the implementation constraints of neural hardware. This point is important because several authors have argued for the use of the average channel energy as a feature for texture retrieval [102, 163, 40, 24, 137, 15]. Both the Bayesian principles and psychophysical evidence indicate that this is a bad idea. Finally, a third stage of pooling different channels may be consistent with a representation based on density estimates of the feature measurements. Not enough is known at this point to argue for or against this position.

A few perceptually based models of higher level have also been proposed in the literature [94, 174, 15, 155]. These, however, tend to be restricted to specific domains such as texture or color perception. Since there are no universal strategies for the experimental validation of the predictions made by these models, it is difficult to reach definitive conclusions about their strengths and weaknesses. In any case, these models tend to emphasize a decomposition into properties like randomness, periodicity, scale, and orientation that are all easily extracted from a representation localized in space and frequency.

More concrete evidence for the benefits of multi-resolution representations is that provided by decades of experience in image compression, where frequency decompositions are universally accepted as a good pre-processing stage to compression [74, 128, 63, 29]. The observation that discontinuities in the low-frequency components of an image (*blocking artifacts*) are much more noticeable than similar discontinuities in their high frequency counterparts indicates that low frequency information is perceptually more relevant than that in the high-frequencies. The facts that PCA is well approximated by the DCT for many natural images and the DCT is better matched to human perception are indeed the fundamental reasons for the widespread use of the DCT in image compression.

Finally, there is a long history of machine vision problems where multi-resolution representations are known to lead to the best solutions [20, 105, 22, 95, 4, 187] and some striking recent advances in image synthesis, based on the statistical analysis of such representations, reinforce the idea that they are central to perceptually meaningful image modeling [64, 138, 135, 14, 151].

For all these reasons, multi-resolution spaces are natural candidates for CBIR from both the perceptual and Bayes error points of view. While we rely on the DCT, it should be pointed out that Bayesian retrieval is not tied to this particular transform. In fact, any other multi-resolution feature transformation could be employed, including wavelets [99, 100], Laplacian [20], or Gabor [137, 102] pyramids. Because most images are compressed using the DCT [128, 63], the DCT has the practical advantages of compatibility with a wide installed base of image processing hardware. This is the fundamental reason that motivated us to select it.

4.6 Experimental evaluation

In this section we present experimental results on the performance of the DCT features. Since the discussion on image representation will only be complete in the next chapter we postpone a detailed evaluation until then. Here, we simply want to dispel the common belief that the DCT coefficients are not a good feature set for texture recognition [96, 134, 163]. Figure 4.2 presents precision/recall curves, on the Brodatz database, for retrieval based on both the DCT and the MRSAR features. The DCT features were obtained by sliding an 8×8 window by increments of two pixels over each image to be processed. The implementation of MRSAR is that of [94]. Each feature transformation is combined with two similarity functions: the Mahalanobis distance, which is the standard in the texture literature, and the ML criteria, in a total of 4 image representations.

The figure confirms that, under MD, the performance of the DCT features is indeed terrible: precision is never better than 25%. However, a very different result is obtained when the similarity function is ML, in which case precision improves by up to 65% points! Hence, while the DCT is significantly worse than MRSAR under MD - a difference in precision that can be as high as 70% - it becomes competitive under ML - difference usually below 5%. Notice that, *when combined with ML, the DCT features even outperform the standard MRSAR/MD combination.*

Figure 4.3 provides an explanation for these observations. In the figure we present equi-

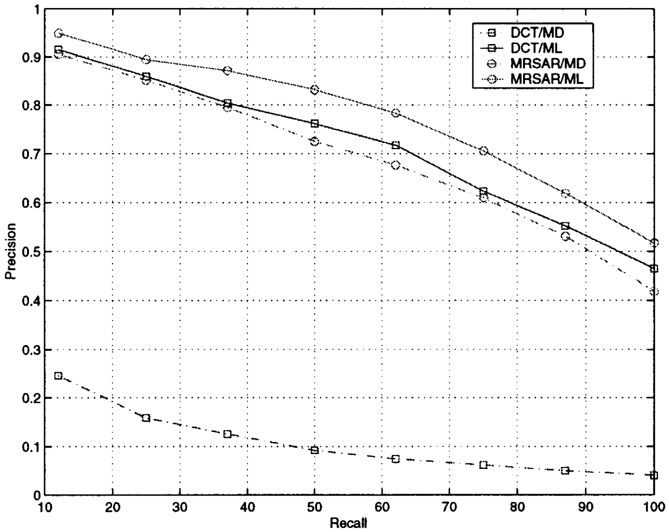


Figure 4.2: Precision/recall curves for Brodatz. In the legend, MRSAR means MRSAR features, DCT the DCT features, ML maximum likelihood, and MD the Mahalanobis distance.

probability contours for the best Gaussian fits to the features extracted from ten texture classes in the Brodatz database. In both plots, we show the joint density of the first two coefficients other than the pixel mean.

Since the DCT is an orthonormal transform, it preserves in $\mathcal{X}$ the shape of the densities in $\mathcal{Z}$. Hence, it is not surprising that there is a significant amount of overlap between the densities of different classes. On the other hand, following strategy S2 of Chapter 3, MRSAR tries to separate these densities as much as possible. Small overlap is a very important requirement under the MD since, as discussed in section 2.3.5, this metric does not consider the full query density, but only its mean. Consequently, because the DCT features have zero mean for all classes, retrieval is very error-prone when they are combined with MD. On the other hand, since ML can account for the entire query density, it has enough information to distinguish the different classes, even when the DCT features are used. It is therefore not surprising that the performance of the DCT improves so dramatically.

In summary, and contrary to prior beliefs, it is not the DCT coefficients that are a bad feature set for texture retrieval. Instead, the problem relies on the use of the Mahalanobis

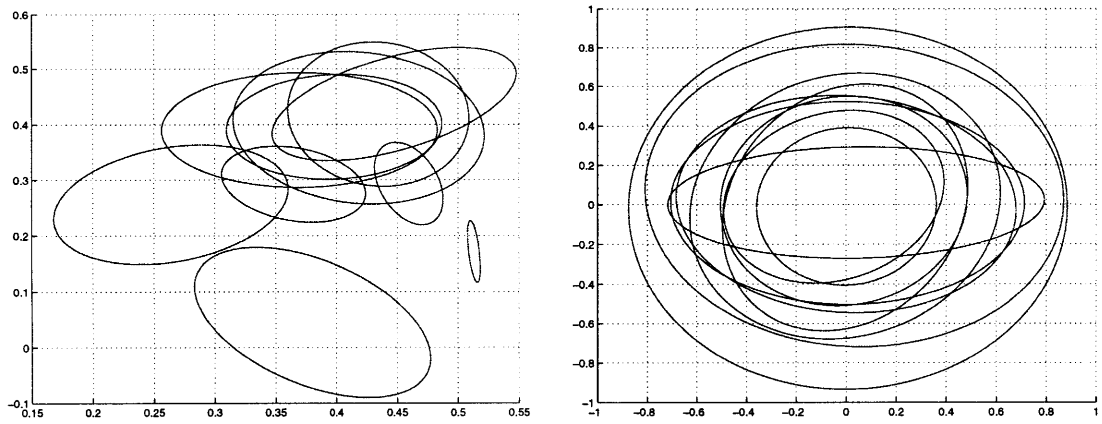


Figure 4.3: Gaussian fits (contours where probability drops to 65% of the maximum) to the features from ten texture classes from Brodatz. Only the first two components (other than the mean coefficient) of the feature space are shown. Left: MRSAR, right: DCT.

distance as a similarity criteria. The significance of this result is that, because the DCT features are generic, there is potential to design a unified representation for color and texture capable of doing well on generic databases. This is not possible with transformations that are highly specialized for texture, such as MRSAR.

Chapter 5

Embedded multi-resolution mixture models

Since, in practice, there is no way of rendering feature representation a trivial problem, it is important to always rely on the most expressive density models available. In this chapter, we review several models that have been proposed in the retrieval literature and show that they are particular cases of a parametric family of densities know as mixture models. A mixture model is in turn shown to define a collection of embedded probabilistic descriptions over subspaces of the original feature space.

When combined with a multi-resolution feature transformation, this embedded representation leads to an interesting extension of the color histogram that provides explicit control over the trade-off between spatial support and invariance. We present experimental evidence that the embedded multi-resolution mixture representation outperforms the standard methods for color- and texture-based retrieval, even on the specific domains for which these techniques were designed. This confirms that the new representation can account for both color and texture and should do well on generic image databases.

5.1 Spatially supported representations

The main challenge for a feature representation designed to account for both color and texture is to combine tractability on high dimensions with expressive power to model complicated densities. We have already established that the two most popular representations in current use cannot fulfill this goal: the Gaussian assumption is too simplistic, the histogram model does not scale to high dimensional spaces. There have been, however, some efforts to extend these representations into usable joint models of color and texture.

5.1.1 Extensions to the Gaussian model

The simplest among such solutions is to represent the class-conditional densities by finite collections of their *moments*

$$\gamma_i^p = E_{\mathbf{x}|Y}[\mathbf{x}^p|Y = i] = \int \mathbf{x}^p P_{\mathbf{X}|Y}(\mathbf{x}|i)d\mathbf{x},$$

where γ_i^p is the p^{th}-order moment of the density associated with image class i. This is an extension of the Gaussian model that can only account for second-order moments.

While, theoretically, any density can be characterized by a collection of moments [124], it may take a large number number of them for the characterization to be accurate. The difficulty of determining how many moments are enough in any given situation may be the reason why the approach has not been extensively pursued. While the use of more than two moments has been advocated by some authors [167, 101], the total number is usually kept small. It is therefore not clear that the resulting characterization will be expressive enough to model complex densities, especially in high dimensions.

One answer to this problem is to arbitrarily divide the image into regions and compute moments for each region [166]. If there are M regions, this is equivalent to the probabilistic model

$$P_{\mathbf{X}|Y}(\mathbf{x}|i) = \frac{1}{M}\sum_{r=1}^{M} \hat{P}_{\mathbf{X}|R,Y}(\mathbf{x}|r, i) \tag{5.1}$$

where $\hat{P}_{\mathbf{X}|R,Y}(\mathbf{x}|r, i)$ is the moment-based approximation to the density of the r^{th} region of the i^{th} image class. Since the total number of moments is proportional to the number of regions, the resulting representation is compact if and only if the segmentation is limited to a small number of large regions. This usually leads to arbitrary image partitions that are not tailored to the image statistics.

5.1.2 Extensions to the histogram model

It has been suggested that the solution to the limitations of color-based retrieval is to rely on histograms of spatially supported features [156, 125]. This fails to realize that the main limitation of color-based retrieval is the histogram itself. In result, feature spaces are constrained to small dimensions and a significant amount of information is lost. We have already shown that this is not a good idea.

A better extension to the histogram is to explicitly augment this model with spatial information. This is the rationale behind representations like *color coherence vectors* (CCVs) [126] and *color correlograms* [66]. CCVs divide the image pixels into two classes, coherent and non-coherent (where the coherency of a pixel is a function of the number of similar pixels that are spatially connected to it), and compute an histogram for each class. They provide a probabilistic model of the form

$$P_{\mathbf{X}|Y}(\mathbf{x}|i) = \sum_{c=0}^{1} P_{\mathbf{X}|\mathcal{C}(\mathbf{X}),Y}(\mathbf{x}|c,i) P_{\mathcal{C}(\mathbf{X})}(c), \tag{5.2}$$

where $\mathcal{C}(\mathbf{x}) = 1$ for coherent pixels, $\mathcal{C}(\mathbf{x}) = 0$ for non-coherent ones, and $P_{\mathbf{X}|\mathcal{C}(\mathbf{X}),Y}(\mathbf{x}|c,i)$ are the color histograms for each case.

Correlograms are spatial extensions of the histogram, registering the relative frequencies of occurrence of color-pairs on pixels separated by a pre-determined set of spatial distances. They are a simplification the co-occurrence matrices that have been widely studied in the texture literature during the seventies [62, 61, 178, 144]. Under the co-occurrence model a texture is characterized by a collection of matrices $\{\mathbf{M}_{\mathbf{d}_k}\}_{k=1}^{K}$. Each matrix is associated with a distance vector $\mathbf{d}_k = (d_k, \theta_k)$, of norm d_k and angle θ_k with the horizontal axis, and contains color co-occurrence probabilities for pixels separated by that distance. If $C_{l,m}$ is the color of pixel (l, m), then the element (i, j) of $M_{\mathbf{d}_k}$ contains the relative frequencies with which two pixels separated by $\mathbf{d}_k$ occur on the image, one with color i and the other with color j

$$[M_{\mathbf{d}_k}]_{i,j} = P((l,m),(p,q)|C_{l,m} = i, C_{p,q} = j, (l,m) = (p,q) + \mathbf{d}).$$

The main limitation of co-occurrence matrices is the large number of probabilities that have to be estimated, leading to significant computational complexity and poor estimates (there may not even be any observations for a significant number of the matrix elements). The color correlogram tries to avoid some of these difficulties by simply averaging co-occurrence matrices with respect to θ. In practice, this is usually not enough to make the implementation feasible and the correlogram is replaced by the *auto-correlogram* which only consider pairs of pixels of the same color. The auto-correlogram is the collection of vectors

$$[\mathbf{M}_{d_k}]_i = P((l,m),(p,q)|C_{l,m} = C_{p,q} = i, \rho((l,m),(p,q)) = d_k), \tag{5.3}$$

where ρ is usually the L^∞ norm.

For both CCVs and auto-correlograms, retrieval is based on straightforward extensions of the standard histogram similarity metrics. It has been shown experimentally that auto-correlograms achieve the best performance in this class of representations, significantly outperforming the histogram [66]. This is not surprising since the auto-correlogram accounts for both color and texture.

5.1.3 Vector quantization

Since both the exponential dependence on dimensionality and sparseness of the histogram are a consequence of the rigid rectangular partition of the feature space of (2.25), a final solution is to adapt this partition to the particular characteristics of the image data. One possible way to do this is to vector quantize [56, 91] the feature space. A *vector quantizer* (VQ) is a map

$$\mathcal{Q} : \mathcal{R}^n \to \mathcal{C},$$

where $\mathcal{C} = \{\mathbf{y}_1, \ldots, \mathbf{y}_C\}$ is a finite set of *reconstruction vectors*, or *codebook*, and $\mathbf{y}_i \in \mathcal{R}^n$. This map defines a partition of R^n into C regions $\{\mathcal{R}_1, \ldots, \mathcal{R}_C\}$, associating a reconstruction vector $\mathbf{y}_i$ with each region

$$\mathcal{Q}(\mathbf{x}) = \sum_{i=1}^{C} \mathbf{y}_i \, \chi_{\mathcal{R}_i}(\mathbf{x}), \tag{5.4}$$

where $\chi_{\mathcal{R}_i}(\mathbf{x})$ are set indicator functions (2.2).

For a random variable $\mathbf{x}$ characterized by a density $P_{\mathbf{X}}(\mathbf{x})$ and a distortion measure $d(\mathbf{x}, \mathcal{Q}(\mathbf{x}))$, the average distortion introduced by a VQ is

$$\mathcal{D} = \int d(\mathbf{x}, \mathcal{Q}(\mathbf{x})) \, P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}.$$

It can be shown [56, 91] that, when the goal is to minimize this distortion, the optimal partition for a fixed codebook must satisfy the *nearest-neighbor condition*

$$\mathcal{R}_i = \{\mathbf{x} : d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j), \forall j \neq i\},$$

while the optimal codebook for a given partition must satisfy the *generalized-centroid condition*

$$\mathcal{Q}(\mathbf{x}) = \min_{\mathbf{y}_i} \{E[d(\mathbf{x}, \mathbf{y}_i) | \mathbf{x} \in \mathcal{R}_i]\}.$$

In practice, the distortion measure is usually one of the quadratic distances discussed in Chapter 2. The mean squared error is a particularly popular choice leading to

$$\mathcal{R}_i = \{\mathbf{x} : ||\mathbf{x} - \mathbf{y}_i||^2 \leq ||\mathbf{x} - \mathbf{y}_j||^2, \forall j \neq i\}, \tag{5.5}$$

and

$$\mathbf{y}_i = \mathcal{Q}(\mathbf{x}) = \{E[\mathbf{x} | \mathbf{x} \in \mathcal{R}_i]\}. \tag{5.6}$$

Under this distance, given the codebook $\{\mathbf{y}_1, \ldots, \mathbf{y}_C\}$, the feature vectors extracted from each image are quantized by simply finding the reconstruction vectors that are closest to them under the Euclidean norm. Each feature vector is therefore replaced by a scalar (the index or label of the corresponding reconstruction vector) and, for retrieval, the entire image can be represented by the histogram of the labels. Standard histogram metrics, such as L^p norms, can then be used to evaluate image similarity [70, 183, 68, 118, 109, 193, 163, 139].

While vector quantization reduces the sparseness and dimensionality problems, there are a few significant problems associated with label histograms. The first is the assumption that it is possible to find a universal VQ that will be a good representation for all the images in the database. It is not clear that this is the case or, even when it is, if the resulting codebook will have a manageable size. Second, a learned VQ must typically be retrained whenever new image classes are added to the database. These problems can be solved by avoiding histogram similarity measures, and simply reverting back to the ML similarity criteria of (2.16), using VQ-based density estimates. In other words, instead of learning a universal VQ and evaluating similarity between label histograms (the quantization view), a VQ is learned for each image class and similarity evaluated through the ML criteria, using VQ-based density estimates (the probabilistic retrieval view) [136, 192, 190, 181]. For this, we need a probabilistic, or *generative*, interpretation for vector quantization. We address this issue in the next section, where we will consider a generic family of probability models that encompasses, as special cases, the majority of the representations discussed so far.

5.2 Mixture models

A mixture density [177, 143, 13] has the form[1]

$$P(\mathbf{x}) = \sum_{c=1}^{C} P(\mathbf{x}|\omega_c)P(\omega_c), \tag{5.7}$$

where C is a number of feature classes, $\{P(\mathbf{x}|\omega_c)\}_{c=1}^{C}$ a sequence of *feature class-conditional densities* or *mixture components*, and $\{P(\omega_c)\}_{c=1}^{C}$ a sequence of *feature class probabilities*. Mixture densities model processes with hidden structure: one among the C feature classes is first selected according to the $\{P(\omega_c)\}$, and the observed data is then drawn according to the respective feature class-conditional density. These densities can be any valid probability density functions, i.e. any set of non-negative functions integrating to one.

Because the complexity of the mixture model is proportional to the complexity of the

[1] In this section, we drop the dependence on the image class $Y = i$ which is always implicit and the subscript from all densities since the random variables are clear from the context. For example, we write $P(\mathbf{x}|\omega_c)$ instead of $P_{\mathbf{X}|\Omega}(\mathbf{x}|\omega_c)$. We also use the words "feature class" for the different feature sub-classes that may exist within each image class.

feature class-densities, if the latter are tractable in high dimensions, the former will also be. This is indeed the case for most of the feature class-conditional densities in common use and, in particular, the Gaussian. Furthermore, because it is, by definition, a multi-modal model, the mixture density can easily capture the details of complex densities.

In particular, Li and Barron have recently shown [90, 88] that if $\mathcal{C}$ is the space of all convex combinations[2] of a density $P_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)$ parameterized by θ and F_k a k-component mixture of $P_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)$ then, for any density F,

$$KL(F||F_k) \leq KL(F||F^*) + \frac{c_F^2\gamma}{k},$$

where c_F is a constant that depends on F, γ depends on the family $P_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)$, and $F^* = \inf_{G \in \mathcal{C}} KL(F||G)$. This bounds the difference between $KL(F||F_k)$ and $KL(F||F^*)$, which is a measure of the distance between F and $\mathcal{C}$. Obviously, if $F \in \mathcal{C}$ then $KL(F||F^*) = 0$. For Gaussian mixtures of covariance $\sigma\mathbf{I}$, $\gamma = O(n/\sigma^2)$, i.e. the bound is linear on the dimension of the space. Hence, by selecting k large enough, it is possible to make the bound arbitrarily tight as long as $\sigma > 0$. On the other hand, by making $\sigma \rightarrow 0$, it is always possible to make $KL(F||F^*) = 0$. This result therefore suggests that, by selecting σ small enough and then k large enough, it is always possible to approximate F arbitrarily well.

In practice, densities can usually be well approximated by mixtures with a small number of components. In this section, we show that most of the representations in current use for image retrieval are particular cases of the mixture model.

5.2.1 Some obvious relationships

By simply making $C = 1$, it is obvious from (5.7) that any parametric density is a particular case of the mixture model. Similarly, the representation of an image by a collection of global moments is nothing more than an approximation to a one-component mixture model. If the image is segmented and each region modeled by a different set of moments, we obtain the approximation of (5.1), where each feature class corresponds to an image region. Finally, a

[2] This includes both finite and continuous mixtures models.

CCV is a 2-component mixture model, where the features are pixel colors, feature classes are determined by the coherence of those colors, and feature class-conditional densities are modeled by histograms.

5.2.2 Relationship to non-parametric models

It is also clear from (5.7) that, given a sample of observations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, by making the number of feature classes equal to the number of observations $|\mathbf{X}|$, assuming each class to be equally likely, and feature class conditional densities to be replicas of the same kernel $\mathcal{K}_\Sigma(\mathbf{x})$ centered on the observations

$$P(\mathbf{x}) = \frac{1}{|\mathbf{X}|} \sum_{i=1}^{|\mathbf{X}|} \mathcal{K}_\Sigma(\mathbf{x} - \mathbf{x}_i) \tag{5.8}$$

we obtain what are usually called *Parzen* or *kernel* density estimates [39, 162, 48]. These models are traditionally referred to as non-parametric densities, even though they usually require the specification of a scale (or *bandwidth*) parameter Σ. One popular choice for the kernel $\mathcal{K}_\Sigma(\mathbf{x})$ is the Gaussian distribution, in which case Σ is a covariance matrix.

The kernel model can be seen as the limit case of the region-based moments approach, where a window is placed on top of each image pixel, a set of moments measured from the features extracted from that window, and the kernel defined by those moments. Of course, the mixture model supports any representation in between one single set of global moments and a different set of moments for each pixel.

5.2.3 Relationship to vector quantization

Several authors have pointed out the existence of relationships between mixture models and vector quantization [3, 192, 145, 136, 78, 13, 27]. However, this tends to be done by comparing the standard algorithms for learning mixture models (the *EM algorithm* [36, 143, 13]) and vector quantizers (the *generalized Lloyd* or *LBG* algorithm [56, 91]), and showing that the later is a special case of the former.

This is somewhat unsatisfying since the learning algorithm does not completely specify the probabilistic model. In fact, various algorithms have been proposed both for vector quantization [91, 82, 3, 197, 147] and mixture density estimation [36, 90, 143, 108]. The comparison of algorithms therefore leads to different views on the relationship between the underlying representations [3, 192, 136, 27]. Here, we seek an explicit relationship between the probabilistic models.

For this, we start by noticing that, associated with any mixture model, there is a *soft partition* of the feature space. In particular, given an observation $\mathbf{x}$, it is possible to assign that observation to each of the feature classes according to

$$\begin{aligned} P(\omega_i|\mathbf{x}) &= \frac{P(\mathbf{x}|\omega_i)P(\omega_i)}{\sum_{k=1}^{C} P(\mathbf{x}|\omega_k)P(\omega_k)} \\ &= \begin{cases} \frac{1}{1+\sum_{k\neq i}\frac{P(\mathbf{X}|\omega_k)P(\omega_k)}{P(\mathbf{X}|\omega_i)P(\omega_i)}}, & \text{if } P(\mathbf{x}|\omega_i)P(\omega_i) > 0 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{5.9}$$

The following theorem makes explicit the relationship between vector quantization and mixture models.

Theorem 5 *If* $\mathbf{x}$ *is a random vector distributed according to a Gaussian mixture*

$$P_\epsilon(\mathbf{x}) = \sum_c P(\omega_c)\mathcal{G}(\mathbf{x}, \mu_c, \mathbf{\Sigma}_c(\epsilon))$$

with covariances

$$\mathbf{\Sigma}_c(\epsilon) = \epsilon\mathbf{I},\ \forall c,$$

then

$$\lim_{\epsilon\to 0} P_\epsilon(\omega_i|\mathbf{x}) = \begin{cases} 1, & \text{if } ||\mathbf{x}-\mu_i|| \leq ||\mathbf{x}-\mu_k|| \forall k < i \\ 0, & \text{otherwise} \end{cases} \tag{5.10}$$

and

$$\lim_{\epsilon\to 0} P_\epsilon(\mathbf{x}) = \sum_{i=1}^{C} \delta(\mathbf{x}-\mu_i)P(\omega_i). \tag{5.11}$$

where $\delta(\mathbf{x})$ *is the Dirac delta function (2.7).*

Proof: Since a mixture model with C classes of which z have zero probability is the same as a model with $C - z$ classes of non-zero probability, we assume, without loss of generality, that all the classes have non-zero probability, i.e.

$$P(\omega_i) > 0, \ \forall i.$$

For Gaussian feature class-conditional densities, (5.9) then becomes

$$P(\omega_i|\mathbf{x}) = \frac{1}{1 + \sum_{k \neq i} \sqrt{\frac{|\mathbf{\Sigma}_i|}{|\mathbf{\Sigma}_k|}} \frac{e^{||\mathbf{x}-\mu_i||^2_{\mathbf{\Sigma}_i} - \log P(\omega_i)}}{e^{||\mathbf{x}-\mu_k||^2_{\mathbf{\Sigma}_k} - \log P(\omega_k)}}},$$

and for $\mathbf{\Sigma}_i = \epsilon \mathbf{I}, \forall i$,

$$P_\epsilon(\omega_i|\mathbf{x}) = \frac{1}{1 + \sum_{k \neq i} \frac{P(\omega_k)}{P(\omega_i)} e^{\frac{1}{\epsilon}(||\mathbf{x}-\mu_i||^2 - ||\mathbf{x}-\mu_k||^2)}}.$$

Hence

$$\lim_{\epsilon \to 0} P_\epsilon(\omega_i|\mathbf{x}) = \begin{cases} a, & \text{if } ||\mathbf{x} - \mu_i|| \leq ||\mathbf{x} - \mu_k|| \ \forall k \neq i \\ 0, & \text{otherwise.,} \end{cases} \tag{5.12}$$

where

$$a = \frac{P(\omega_i)}{P(\omega_i) + \sum_{\{k:||\mathbf{x}-\mu_k||=||\mathbf{x}-\mu_i||\}} P(\omega_k)}.$$

Since the set $\{\mathbf{x} : ||\mathbf{x}-\mu_k|| = ||\mathbf{x}-\mu_i||\}$ has measure zero, (5.12) is equivalent to (5.10) almost everywhere. Furthermore, because some arbitrary tie-breaking rule is always necessary to vector quantize the points that lie on the boundaries between different cells, the same rule can be applied to (5.12) and the two equations are equivalent. Equation (5.11) is a direct consequence of the fact that the Gaussian density converges to the delta function as its covariance tends to zero [123].□

Equations (5.10) and (5.11) are nothing more than a generative model for a VQ. While (5.10) is simply the nearest neighbor condition (5.5), (5.11) is the probabilistic version of (5.4) and (5.6): given a cell label, a vector quantizer draws a sample from the conditional density associated with that cell. Since this density is the delta function centered on the cell's centroid, this sampling operation is equivalent to the combination of (5.4) and (5.6). It is, therefore, clear that a VQ is a particular case of the Gaussian mixture model.

5.2.4 Relationship to histograms

Equations (5.10) and (5.11) also provide an interpretation of a VQ as an histogram since the vector $\mathbf{H} = [P(\omega_1) \dots P(\omega_C)]^T$ is estimated with normalized counts of the number of samples that land on each of the quantization cells. Because this is also the definition of histogramming, it is clear that histograms are a particular case of the mixture model. A special case of interest occurs when the reconstruction vectors μ_i are constrained to lie on a rectangular grid of size $h_1, \dots, h_n$. In this case, the quantization cells become rectangles

$$P(\omega_i|\mathbf{x}) = \begin{cases} 1, & \text{if } |\mathbf{x}_1 - \mu_{i,1}| \leq \frac{h_1}{2}, \dots, |\mathbf{x}_n - \mu_{i,n}| \leq \frac{h_n}{2} \\ 0, & \text{otherwise,} \end{cases}$$

and we obtain the standard histogram model of (2.25).

5.2.5 A unified view of feature representations

The relations between the various representations are depicted in Figure 5.1. Starting from the generic mixture model and a sample $\mathbf{X}$, if the number of classes C is set to the sample size and the classes assumed to be equally likely, we obtain a non-parametric model. If, on the other hand, $C = 1$ we have the parametric density defined by the particular kernel of choice. For C in between these two extrema, selecting a Gaussian kernel of zero covariance, leads to a vector quantizer and further restriction of the class centroids to lie on a rectangular grid leads to the standard histogram.

Independently of the number of classes C, the mixture model can always be approximated by storing a collection of moments instead of the complete class-conditional densities. In particular, for $C = 1$ we obtain the approximation by a collection of global moments and for C equal to a pre-defined number R of regions we obtain a region-based moment approximation. Finally, if there are only two classes (determined by the coherence of pixel colors) and each class-conditional density is represented by an histogram we obtain a CCV.

Since all these feature representations are particular cases of the mixture model, it is expected that they will lead to suboptimal performance when applied to the retrieval problem. We now analyze this issue in greater detail.

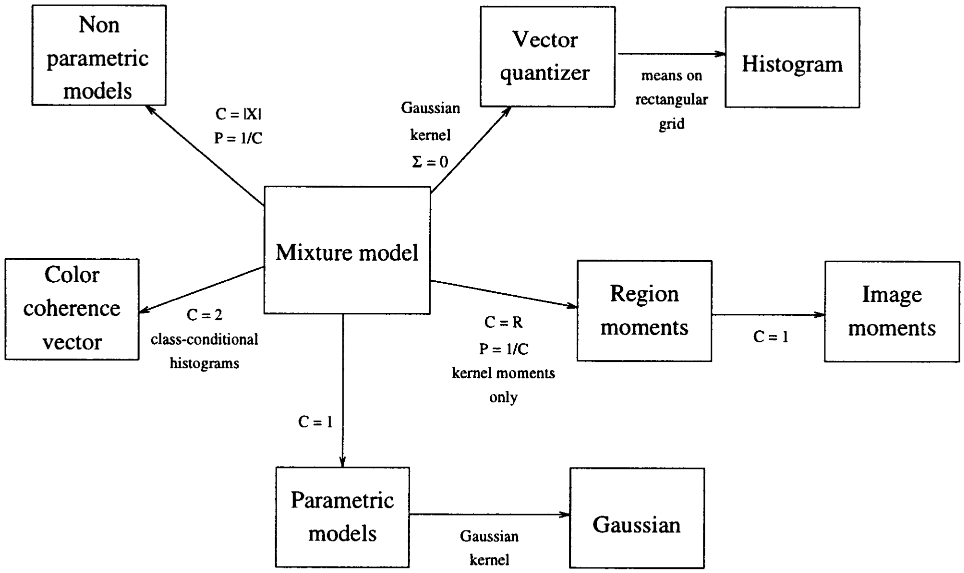


Figure 5.1: Relations between different feature representations. C is the number of feature classes and P the probability of each class.

We start by noticing that because non-parametric models have as many degrees of freedom as the number of observations, these models are not compact. Consequently, the evaluation of equation (5.8) is computationally expensive (complexity proportional to the number of training features). On the other hand, there is no guarantee that the density estimates will be better than those provided by the mixture model, and there is usually no easy way to set the bandwidth parameter [162]. It is, therefore not clear that relying on a non-parametric model will justify the increase in retrieval complexity.

Relying on moment approximations is usually also not a good idea. While global moments provide a very crude approximation to the underlying density; region-based moment descriptions tend to be fairly arbitrary when the segmentation is not dictated by the image itself, and automated segmentation is well known to be a very difficult problem. This criticism is also valid for CCVs, where the classification into coherent and non-coherent pixels is rather arbitrary.

While non-parametric models have too many degrees of freedom, parametric ones have too few. Most parametric densities are unimodal and, as seen in the previous chapter, do not have enough expressive power to capture the details of the densities associated with real images. Standard histograms overcome this limitation but, given their intractability in high dimensions, are ineffective for joint modeling of color and texture. On the other hand, by adapting the partition of the space to the characteristics of the data, vector quantization can outperform the standard histogram with lower complexity. While this enables estimation on high-dimensional feature spaces, the fact that VQ-based estimates rely on a hard partition of the space restricts their robustness to small image variations. In fact, as illustrated in Figure 5.2, slight feature perturbations may lead to drastic changes in quantization and, consequently, image similarity.

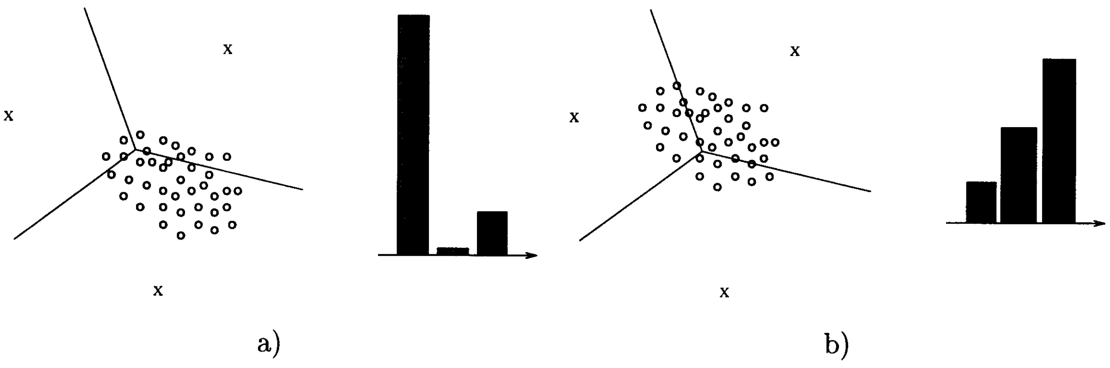


Figure 5.2: a) Partition of the feature space by a 3-cell VQ, a set of feature vectors, and the corresponding label histogram. b) For a universal VQ, small perturbations of the feature vectors can lead to entirely different label histograms.

Because mixture models rely on a soft partition of the feature space, they eliminate this problem. Furthermore, by allowing arbitrary covariances for each of the feature classes, Gaussian mixtures provide a much better approximation to the true density than the train of delta functions inherent to a VQ. The difficulty of accounting for more than the first-order moments of each cell is a well known problem for VQ-based density models [170, 3, 106, 145, 107, 26].

Despite the appealing properties of mixture modeling, very small attention has been devoted to their application to CBIR. To the best of our knowledge only two retrieval systems have used mixture density estimates. One is the *Blob-world* system [7] which relied on Gaussian mixtures for image segmentation. These mixtures were not used, however, for feature representation. Instead, image regions were characterized by a low-dimensional feature vector and retrieval based on the Mahalanobis distance. The other is the *Candid* system [80, 79] that does rely on Gaussian mixtures for feature representation but does not apply them to high dimensional spaces of spatially supported features nor relies on a probabilistic similarity function. In this situation, the mixture representation does not have any significant advantage over the histogram.

5.3 Embedded multi-resolution mixture models

The discussion in the previous section suggests that there is no strong justification for relying on any of the above feature representations instead of the mixture model. In this section, we connect feature representation with feature transformation and show that, also from this point of view, there are good reasons to rely on the Gaussian mixture. A property of particular interest is that a projection of a high-dimensional Gaussian mixture into a linear subspace is still a Gaussian mixture.

Lemma 1 *Let $\mathbf{X} \in R^n$ be a random vector distributed according to the Gaussian mixture density*

$$P_{\mathbf{X}|Y}(\mathbf{x}|i) = \sum_{c=1}^{C} \pi_c \mathcal{G}(\mathbf{x}, \mu_c, \mathbf{\Sigma}_c), \tag{5.13}$$

and consider the projection map $\mathbf{V} = \pi_k(\mathbf{X}) = \mathbf{\Gamma}_k \mathbf{X}$ defined in (4.3). Then

$$P_{\mathbf{V}|Y}(\mathbf{v}|i) = \sum_{c=1}^{C} \pi_c \mathcal{G}(\mathbf{\Gamma}_k \mathbf{x}, \mathbf{\Gamma}_k \mu_c, \mathbf{\Gamma}_k \mathbf{\Sigma}_c \mathbf{\Gamma}_k^T). \tag{5.14}$$

Proof: Consider a Gaussian random vector $\mathbf{x}$ with

$$P_{\mathbf{X}|Y}(\mathbf{x}|y = i) = \mathcal{G}(\mathbf{x}, \mu_\mathbf{x}, \mathbf{\Sigma}_\mathbf{x}),$$

and define $\mathbf{V} = \pi_k(\mathbf{X})$. Since π_k is a linear transformation, $\mathbf{V}$ is also Gaussian distributed. Therefore, $P_\mathbf{V}(\mathbf{v})$ is uniquely determined by its mean and covariance. Using the relationships (4.9) and (4.10)

$$\mu_\mathbf{v} = \mathbf{\Gamma}_k \mu_\mathbf{x}$$
$$\mathbf{\Sigma}_\mathbf{v} = \mathbf{\Gamma}_k \mathbf{\Sigma}_\mathbf{x} \mathbf{\Gamma}_k^T,$$

it follows that

$$P_{\mathbf{V}|Y}(\mathbf{v}|i) = \mathcal{G}(\mathbf{\Gamma}_k\mathbf{x}, \mathbf{\Gamma}_k\mu_{\mathbf{x}}, \mathbf{\Gamma}_k\mathbf{\Sigma}_{\mathbf{x}}\mathbf{\Gamma}_k^T).$$

Applying this result to each of the C components of (5.13) we obtain (5.14).□

This lemma implies that the Gaussian mixture of (5.13) not only defines a probability density on R^n, but also a family of Gaussian mixture densities $\{P_{\mathbf{V}|Y}(\pi_k(\mathbf{x})|i)\}_{k=1}^{n-1}$ on the subspaces $R^1, \ldots, R^{n-1}$. We denote this collection as the family of *embedded mixture models* associated with the original distribution.

When, as is the case of the DCT features, the underlying feature space results from a multi-resolution decomposition this leads to an interesting interpretation of the mixture density as a family of densities defined over multiple image scales, each adding higher resolution information to the characterization provided by those before it. Disregarding the dimensions associated with high-frequency basis functions is therefore equivalent to modeling densities of low-pass filtered images. In the extreme case where only the first, or DC, coefficient is considered the representation is equivalent to the histogram of a smoothed version of the original image. This is illustrated in Figure 5.3.

The *embedded multi-resolution mixture* (EMM) model (embedded mixtures on a multi-resolution feature space) can thus be seen as a generalization of the color histogram, where the additional dimensions capture the spatial dependencies that are crucial for fine image discrimination (as illustrated in Figure 3.3). One of the interesting consequences of this generalization is that it enables fine control over the invariance properties of the image representation.

5.3.1 Invariance

There are many ways to encode invariance into an image representation. While the features can themselves be made invariant to common image transformations by reducing their spatial support (e.g. color histograms), filtering out high-frequency information [156, 158, 103, 166], or application of arbitrary invariant transformations [104, 163, 66], invariance can also be achieved at the level of similarity function [161, 187, 142, 159] or feature representation. In the case of representations that are learned from data, the latter can be achieved by sim-

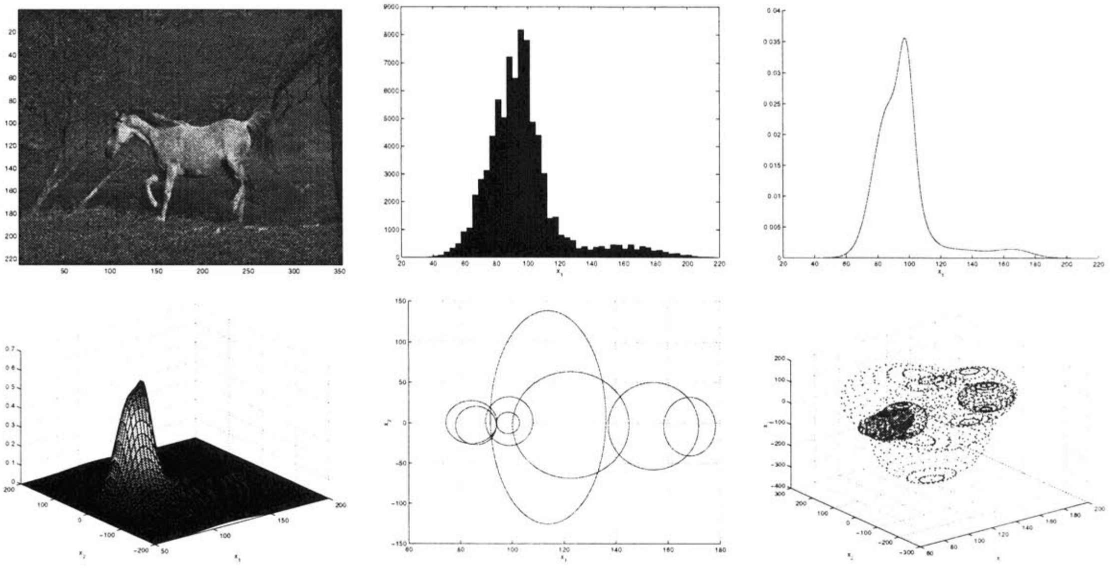


Figure 5.3: An image from the Corel database (top left), its histogram (top center), and projections of the corresponding 64-dimensional embedded mixture onto the DC subspace (top right), the subspace of the two lower frequency coefficients (density on the bottom left, contours where the likelihood drops to 60% on the bottom center), and the subspace of the three lower frequency coefficients (contours shown on bottom right).

ply including a large number of (real or artificial) examples covering all types of variation in the training set [148, 179, 115, 120, 166].

Explicitly modeling all the transformations in the similarity function usually implies a significant complexity increase in the evaluation of similarity and is not recommended in the context of CBIR. On the other hand, learning invariance does not affect retrieval complexity and is the optimal solution from the Bayes error point of view (since no information is discarded). However, the complexity of learning the individual models is usually combinatorial in the number of degrees of freedom that must be accounted for, and it may be impossible to rely on it uniquely. The best solution is, therefore, to combine learning with explicit encoding of invariance in the features.

The EMM representation provides automatic support for this combination. On one hand, by considering less or more of the subspaces, the features can be made more or less invariant to image transformations. In particular, since the histogram is approximately invariant to scaling, rotation, and translation, when only the DC subspace is considered the representation is invariant to all these transformations. As high-frequency coefficients are included, invariance is gradually sacrificed. On the other hand, invariance can always be improved by including the proper examples in the training sample used to learn the parameters of the model.

5.4 Experimental evaluation

In this section, we present results of experiments on the performance of ML retrieval with EMM models. We start by showing that this approach can outperform the standard methods for texture and color-based retrieval even in the specific domains for which these methods were designed, i.e. texture (Brodatz) and color (Columbia) databases. Improvements are shown in terms of both objective (precision/recall) and subjective evaluation. A detailed analysis of the invariance properties of the embedded mixture representation is then carried out both in terms of the trade-off between invariance and spatial support, and how invariance can be encoded in the learning process.

5.4.1 Embedded mixtures

We start by comparing EMM/ML with MRSAR/ML and HI on the Brodatz and Columbia databases. Once again, the DCT features were obtained with an 8×8 window sliding by increments of two pixels. Mixtures of 8 Gaussians were used for the Brodatz database and 16 for Columbia. Only the first 16 subspaces (DCT coefficients) were considered for retrieval. Because in the Columbia database objects are presented against a black background, we restricted Columbia queries to the central square of 1/2 the image dimensions. Diagonal covariances were used for all Gaussians, and all the mixture parameters were learned with the EM algorithm [36]. The implementations of the other techniques were the same as in the previous chapters.

Figure 5.4 presents precision/recall curves obtained on the two databases. It is clear that the EMM/ML combination achieves equivalent performance or actually outperforms the best of the two other approaches in each image domain (MRSAR for texture, HI for

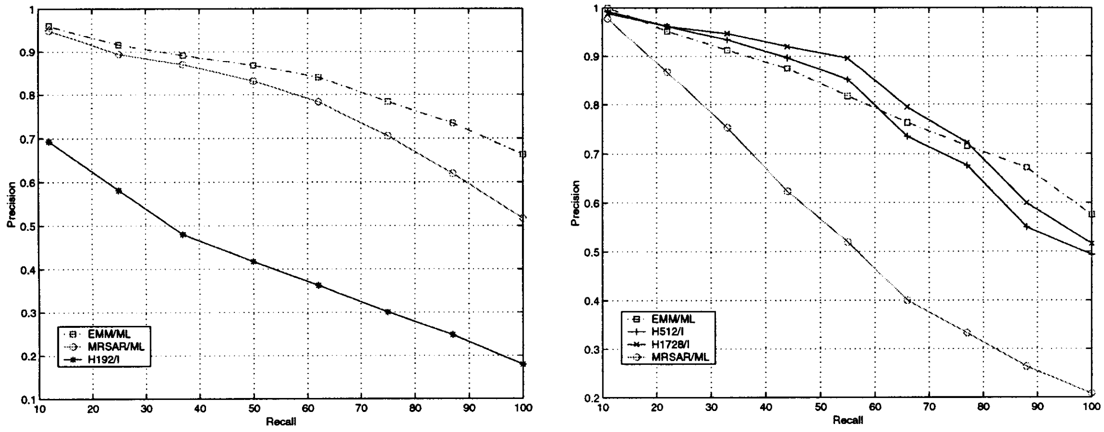


Figure 5.4: Precision/recall curves of EMM/ML, MRSAR/ML, and HI on Brodatz (left) and Columbia (right).

color). This is a significant achievement because EMM is a generic representation, not specifically tailored to any of these domains, and proves that EMM/ML can handle both color and texture. The new representation should therefore do well across a large spectrum of databases.

5.4.2 Perceptual relevance

The visual observation of all the retrieved images suggests that, also along the dimension of perceptual relevance, EMM/ML clearly beats the MRSAR and histogram-based approaches. In this subsection, we present retrieval examples that are representative of the three major advantages of EMM/ML: 1) when it makes errors, these errors tend to be perceptually less annoying than those originated by the other approaches, 2) when there are several visually similar classes in the database, images from these classes tend to be retrieved together, and 3) even when the performance in terms of precision/recall is worse than that of the other approaches, the results are frequently better from a perceptual point of view.

We start by presenting three retrieval examples from the Brodatz database. The three images in the top row of Figure 5.5 present the results of queries performed under the MRSAR/ML combination. The three images in the bottom row present the corresponding results for queries based on EMM/ML. The two images on the left illustrate how the errors

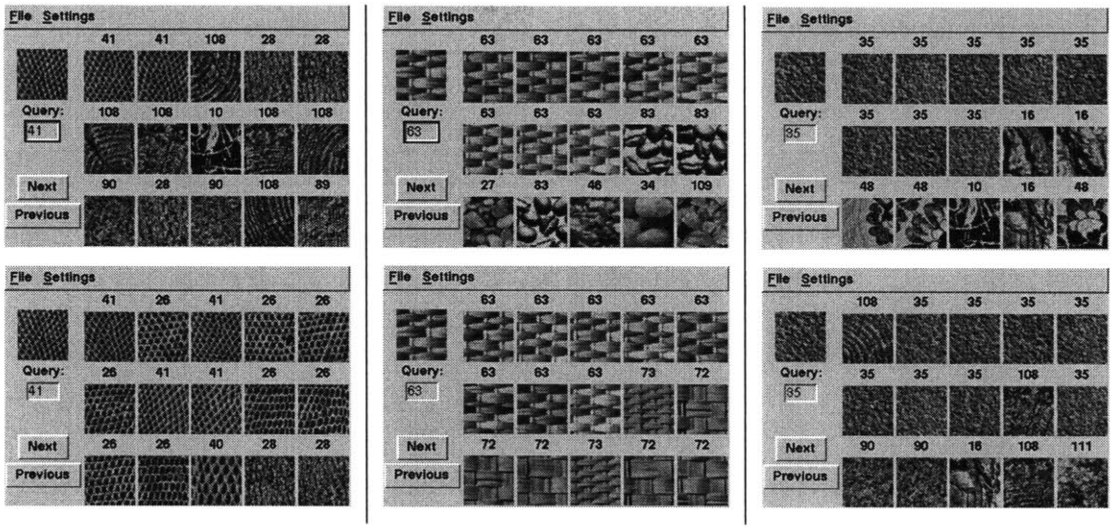


Figure 5.5: Three queries from Brodatz. MRSAR/ML results are shown at the top, EMM/ML results at the bottom. The number above each image indicates the class to which it belongs.

of EMM/ML are usually less annoying than those of MRSAR/ML. Notice that even though EMM/ML makes several errors, most of these would actually be hard to detect by a human if the image class were not indicated on top of each retrieved image. On the other hand, the errors of MRSAR/ML are very clear.

The two images on the center are an example of situations where both approaches perform perfectly in terms of precision/recall, yet the perceptual retrieval quality is very different. MRSAR/ML ranks all the images in the query class at the top, but produces nonsensical matches after that. On the other hand, EMM/ML retrieves images that are visually similar to the query after all the images in its class are exhausted. This observation is frequent and derives from the fact that the MRSAR features have no perceptual justification.

Finally, the two images on the right are an example of how, even when better in terms of precision/recall, the performance of MRSAR/MD can actually be worse from a perceptual viewpoint. Notice how, even though it gets the first image wrong, EMM/ML produces images that are visually similar to the query. On the other hand, MRSAR/ML has perfect precision/recall but produces poor matches after all the images in the class of the query are retrieved.

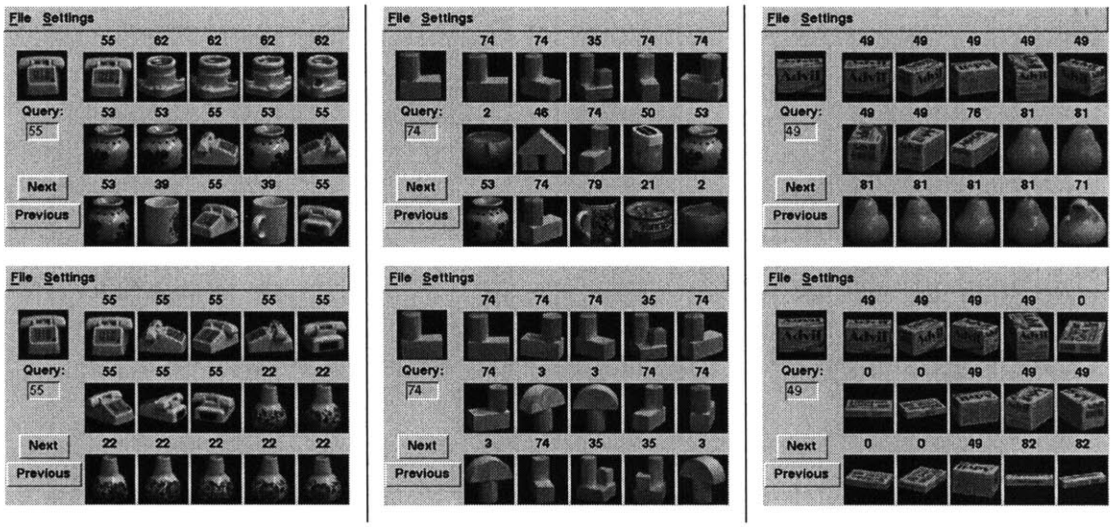


Figure 5.6: Three queries from Columbia. HI results are shown at the top, EMM/ML results at the bottom.

The Columbia database leads to similar observations with respect to the perceptual superiority of EMM/ML. Figure 5.6 presents examples comparing the performance of HI (top row) with EMM/ML (bottom one). The images on the left depict a situation in which EMM/ML is better under both the objective and the perceptual performance criteria. This example illustrates well how color similarity is not enough for good recognition. Because there are several objects with color distributions close to that of the query, variations due to the simple rotation of the object are enough to originate poor retrieval results. In particular, the histogram-based approach does not appear to lead to a perceptually consistent retrieval strategy. Notice how light bulb sockets, vases, and coffee mugs are all returned before the desired telephone images.

On the other hand, EMM/ML not only ranks all the telephones at the top, but also consistently ranks a vase as the most similar object to the telephone query. While it is difficult to say if a person would agree with this judgment (there are no more pictures of telephones in the database), its consistency is a positive sign. The main reason why EMM/ML does better is that, by relying on features with spatial support, it is able to capture the local appearance of the object surface. It will thus tend to match surfaces with the same shape, texture, and reflection properties. This is not possible with color histograms.

The pictures on the center exemplify how capturing local appearance can lead to perceptually pleasing retrievals by EMM/ML, even when the precision/recall performance is only mediocre. In this case, while HI retrieves several objects unrelated to the query, EMM/ML only returns objects that, like the query, are made of wood blocks. Finally, the pictures on the right illustrate how, even when it has higher precision/recall, HI frequently leads to perceptually poorer results than EMM/ML. In this case, images of a pear and a duck are retrieved by HI after the images in the right class ("Advil box"), even though there are several boxes with colors similar to those of the query in the database. On the other hand, EMM/ML only retrieves boxes, although not in the best possible order.

5.4.3 Invariance vs spatial support

As discussed in section 5.3.1, EMMs provide two ways to encode invariance into the retrieval operation: learning and low-pass filtering (by discarding high-frequency subspaces). We now carry out a quantitative analysis of these two mechanisms.

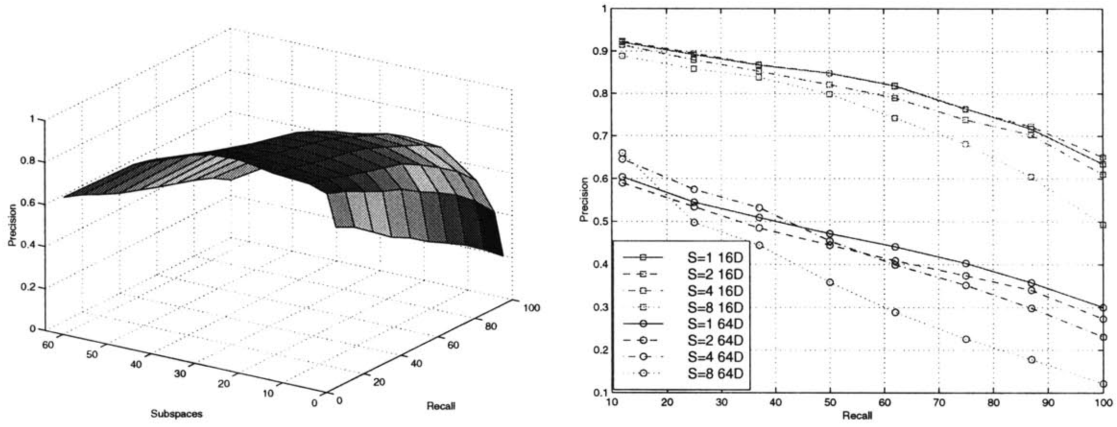


Figure 5.7: Analysis of invariance on Brodatz. Left: surface spanned by precision/recall as a function of the number of subspaces considered during retrieval. Right: precision/recall as a function of block spacing (S) during learning when 16 and 64 subspaces are considered.

Figure 5.7 presents 1) the surface spanned by precision/recall as a function of the number of subspaces considered during retrieval, and 2) the impact on precision/recall of the spatial distance between adjacent features vectors used for learning. The precision/recall surface clearly illustrates the trade-off between invariance and spatial support. When too few subspaces are considered, the spatial support is not enough to capture the correlations that characterize each texture class. Performance increases as the number of subspaces grows, but starts to degrade as high frequencies are included. At this point, because the representation is much more detailed, good recognition requires precise alignment between the query and the database feature vectors. Hence, the recognition becomes very sensitive to small spatial deformations between the textures.

The plot on the right confirms that increasing the number of examples in the training set also improves the retrieval accuracy. When the image blocks are non-overlapping, the representation is invariant only to translations by multiples of the block size. As the space between samples decreases, invariance happens for smaller displacements and retrieval accuracy increases. Notice that, while this is true with both small and large number of subspaces, the relative gain is much higher in the latter case. This was expected since invariance is a bigger problem when high frequencies are included.

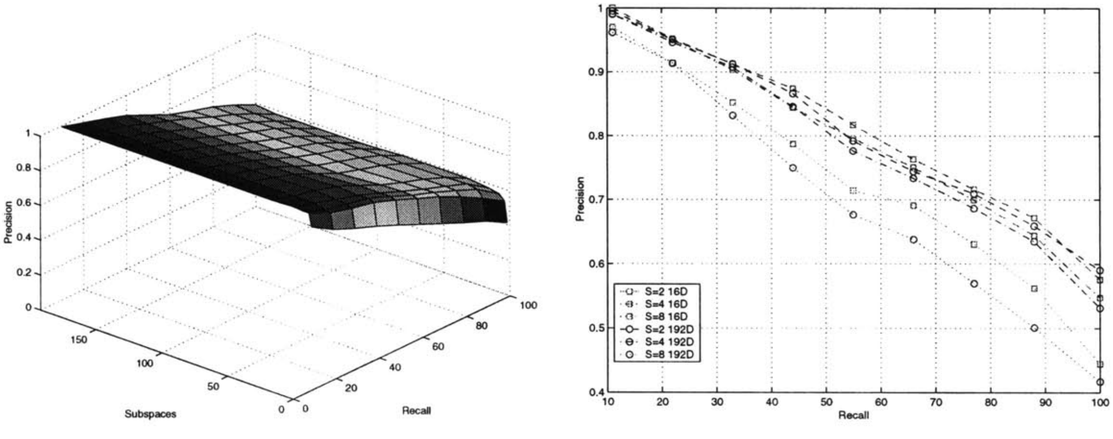


Figure 5.8: Analysis of invariance on Columbia. Left: precision/recall surface. Right: precision/recall as a function of block spacing (S) during learning when 16 and 192 subspaces are considered.

Figure 5.8 presents similar plots for Columbia. While the general behavior is the same, there are interesting differences in the details. In particular, precision/recall increases much faster with the number of subspaces and remains approximately constant after that. This indicates that the intrinsic dimensionality of the images in this database is significantly smaller than the textures in Brodatz and explains why approaches based on low-dimensional feature spaces can perform well for object recognition [172, 156, 103, 158]. This intrinsic low-dimensionality is a consequence of a much smaller high-frequency content and responsible for the flatness of the precision/recall surface: since there is no high-frequency energy, adding or deleting high-frequency subspaces does not have a significant impact on the retrieval accuracy.

On the other hand, as seen by the plot on the right, the spacing between training samples still has a significant impact on the retrieval performance, independently of the number of subspaces considered. We should note here that, while there is a significant amount of variation in scale and rotation on the Columbia database, the results above were obtained without explicit encoding of invariance to these transformations. This is encouraging since one would expect the inclusion of rotated and scaled copies of the feature vectors in the learning sample to further reduce the slope of the precision-recall curve. The main limitation of this approach is the inherent increase in the computational complexity of the learning and, for this reason, we have not conducted any experiments along these lines.

Overall, the results above lead to two main conclusions regarding invariance. First, there seems to be a relatively large set of subspaces for which the representation achieves a good trade-off between spatial support and invariance. In particular, any number between 16 and 32 subspaces per color channel seems to work well. This implies that a precise selection of the number of subspaces is, in practice, not crucial for good performance. Second, while the interval between consecutive samples in the training set clearly affects the performance, there seems to be no need to consider all possible image positions. In fact, while it is important to use a sampling grid where there is overlap between spatially neighboring samples, a spacing of half the block size already achieves performance equivalent to the best results.

5.5 Discussion

While we have shown that mixture models generalize most of the feature representations that have been proposed in the retrieval literature, we do not claim that they are the definitive answer to the problem of feature representation. The main limitation of the EMM image representation is that it can only model short-term image dependencies. In particular, it has no ability to model the dependencies that occur between DCT coefficients of spatially adjacent image neighborhoods.

One obvious extension to account for these dependencies would be to rely on *Markov Random Fields* (MRFs) [25]. These are models that define a probability measure over the entire image lattice by imposing a Markov condition: given the observation of its neighbors, each pixel is independent of the remainder of the image. The complexity of the MRF model depends on the size of the neighborhood underlying this Markov condition and, in practice, only very small neighborhoods are computationally feasible. This limits the ability of MRFs to capture long range structure and the success of these models has been, for the most part, limited to modeling random texture [33, 178, 37]. Since, in this case, correlation decreases quickly with distance, it is not clear that an MRF would provide vast improvements over EMM.

Recently, however, Zhu et al. [151] have shown that it is possible to design MRFs that can account for long-range and periodic structures. Given a multi-resolution decomposition of an image, they compute histograms on each of the frequency subbands and search for the distribution whose marginal densities matches those histograms. This search is formulated as a *maximum entropy* problem which leads to a MRF solution. While their results were impressive, the technique is extraordinarily slow and infeasible in the context of image databases. Nevertheless, it has prompted great interest on the question of modeling the statistics of multi-resolution representations [135, 138, 14].

While it has not yet been shown that it is possible to derive a true probabilistic model, or a suitable approximation, capable of accounting for long-range *spatial* dependencies between frequency coefficients and having tractable complexity, the question of whether such a model exists remains open for further debate.

Chapter 6

From local to global similarity

We have mentioned, in Chapter 2, that probabilistic retrieval establishes a common framework for local and global queries. In this chapter, we analyze the issue of local queries in greater detail. We start by discussing their importance and reviewing the most popular approaches for their implementation. We then show why probabilistic retrieval provides a natural solution to the problem and present experimental evidence of its robustness against incomplete queries.

The major limitation of the straightforward implementation of probabilistic retrieval is then shown to be the linear growth of retrieval complexity with the cardinality of the query. When the goal is to evaluate global similarity, this makes such an implementation much more expensive than standard methods, such as the histogram intersection or MRSAR techniques. We derive an alternative implementation which is competitive with these techniques from a complexity point of view, while achieving performance similar to that of the straightforward implementation.

At the core of this implementation is the evaluation of the KL divergence between two Gaussian mixtures. This is an interesting problem on its own, with applications that go well beyond the CBIR problem. We show that this divergence can be computed exactly when the mixture models are vector quantizers and introduce an asymptotic approximation for generic Gaussian mixtures. This approximation significantly reduces the computational complexity of the KL divergence without any significant impact on the resulting similarity judgments.

6.1 Local similarity

In addition to evaluating holistic similarity between images, a good retrieval architecture should also provide support for local queries, i.e. queries consisting of user-selected image regions. The ability to satisfy such queries is of paramount importance for two fundamental reasons. First, a retrieval architecture that supports local similarity will be much more tolerant to incomplete queries than an architecture that can only evaluate global similarity. In particular, it will be able to perform partial matches and therefore deal with events involving occlusion, object deformation, and changes of camera parameters. This is likely to improve retrieval accuracy even for global queries.

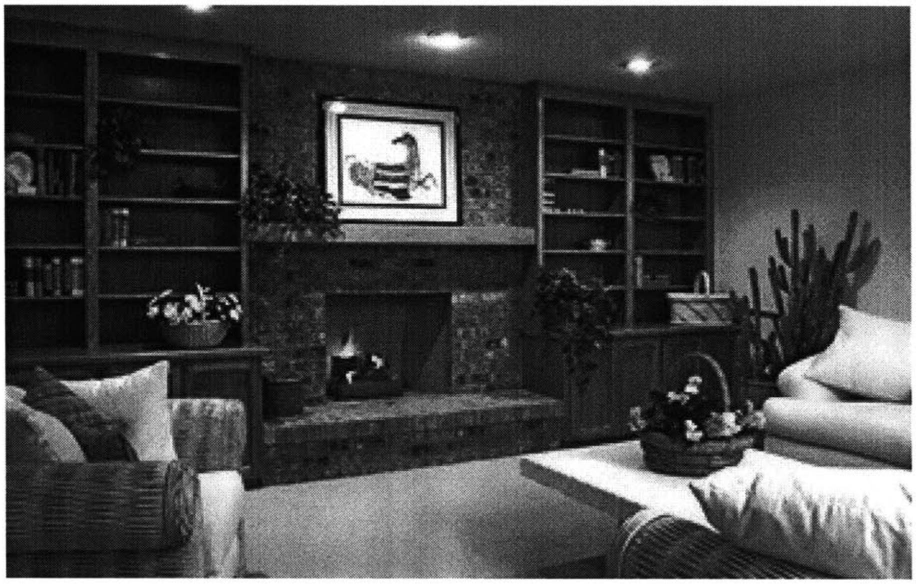


Figure 6.1: Example of a query image with multiple interpretations.

Second, local queries are much more revealing of the user's interests than global ones. Consider a retrieval system faced with the query image of Figure 6.1. Given the entire picture, the only possible inference is that the user may be looking for any combination of the objects in the scene (fireplace, bookshelves, painting on the wall, flower baskets, white table, sofas, carpet, rooms with light painted walls) and the query is too ambiguous.

By allowing the user to indicate the relevant regions of the image, the ambiguity can be significantly reduced.

6.1.1 Previous solutions

The standard solution for handling local queries is to rely on image segmentation and then perform retrieval on the individual segments, i.e. evaluate the similarity of each query region against all the regions extracted from the images in the database. This approach suffers from two fundamental problems: 1) segmentation is a difficult problem, and 2) there is a combinatorial explosion of the number of similarity evaluations to be performed.

Despite the difficulty of automatic image segmentation, several retrieval systems have relied on it for determining image regions [5, 7, 65, 164, 165]. While, theoretically, precise image segmentation enables shape-based retrieval, in practice it is not uncommon for a segmentation algorithm to break a single object into several regions or unify various objects into one region, making shape-based similarity close to hopeless. Hence, even when automated segmentation is used, shape representations tend to be very crude. Therefore, it is not clear that precise segmentation is an advantage for region-based queries. In fact, the use of sophisticated segmentation can be more harmful than beneficial: for example, in the context of "blob-world", Howe [65] reports significant improvements by replacing the sophisticated segmentation algorithm used by Belongie et al. [7] with a much simpler variation.

The only clear exceptions to this observation seem to be applications where it is possible to manually pre-segment all the imagery because 1) there is an economic incentive to do this, and 2) it is very clear what portions of each database image will be relevant to the queries posed to the retrieval system. An example of such application domain is that of medical imaging, in particular what concerns to lesion diagnostics [160]. On the contrary, for generic databases there is usually too much imagery to allow manual processing and it is rarely known what specific objects may be of interest to the users of such databases.

Since precise segmentations are difficult, several authors have adopted the simplifying view of relying on arbitrary image partitions to obtain local information [5, 110, 111, 141, 153, 166, 175]. While this solves the problem of segmentation complexity without noticeable degradation of performance (in fact it does not even seem clear at this point that segmentation works better than arbitrary image partitioning), it still does not address the second problem, i.e. the combinatorial explosion associated with matching all image segments.

In order to overcome this difficulty, several mechanisms have been proposed in the literature. The simplest among these is to make the individual regions large enough and their feature representation compact enough so that each image can still be represented by a simple feature vector (concatenation of the individual region features) of manageable dimensions [166, 175]. Such approaches are of limited use for local queries since 1) several objects or visual concepts may fall on a single image region, 2) feature representations are not expressive enough to finely characterize each region, and 3) it is hard to guarantee invariance to image transformations when dealing with regions of large spatial support.

An alternative view is not to worry with compactness and simply deal with the combinatorics of region-based retrieval at the level of traditional database indexing [110, 111, 141, 164]. Minka and Picard [110] propose clustering of the individual image regions as a database organization step that significantly reduces query time (since query regions are matched against cluster representatives instead of all the members). The use of clustering as an indexing tool has the major disadvantage that the entire database must be re-clustered (an expensive operation) when images are included in or deleted from the database.

An alternative to clustering, proposed by Ravela et al. [141] and Smith and Chang [164], is to rely on indexing mechanisms derived from those traditionally used with text databases. The idea is to consider all the dimensions of the feature space independent, create one dimensional indices (which can be searched quickly) for each of them, and then use standard database operations, such as joins, during retrieval. The main problem with these approaches is that, for the high-dimensional spaces required for meaningful image characterization, the indexing savings vanish as the database grows. The problem is, therefore, particularly acute for databases of image regions.

In summary, the downside of approaches based on indexing is that region databases complicate the indexing problem by orders of magnitude. Since, at this point, indexing is still an open question (even for the simpler case of non-region based representations) this can be a significant hurdle.

6.1.2 Probabilistic retrieval

One of the main attractives of probabilistic retrieval is that, conceptually, it makes local queries straightforward. Recall, from Chapter 2, that for a query composed by a collection of N vectors $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the Bayesian retrieval criteria is

$$g^*(\mathbf{x}) = \arg\max_i \sum_{j=1}^{N} \log P_{\mathbf{X}|Y}(\mathbf{x}_j|i) + \log P_Y(i).$$

Notice that there is, under this criteria, no constraint for the query set to have the same cardinality as the set used to estimate the class-conditional densities $P_{\mathbf{X}|Y}(\mathbf{x}_j|i)$. In fact, it is completely irrelevant if $\mathbf{x}$ consists of one query vector, a collection of vectors extracted from a region of a query image, or all the vectors that compose that query image. Hence, there is no difference between local and global queries.

The ability of Bayesian similarity to explain individually each of the vectors that compose the query is a major advantage over criteria based on measures of similarity between entire densities, such as L^p norms or the KL divergence, for two fundamental reasons. First, it enables local similarity without requiring explicit segmentation of the images in the database. The only segmentation information that is required are the image regions which make up the query and which are provided by the user himself. The indexing complexity is therefore not increased. Second, since probabilistic retrieval relies on a generative model (a probability density) that is compact independently of the number of elemental regions that compose each image, these can be made as small as desired, all the way down to the single pixel size. Our choice of local 8×8 neighborhoods is motivated by concerns that are not driven by the feasibility of the representation per se, but rather by the desire to achieve a good trade-off between invariance, the ability to model local image dependencies, and the ability to allow users to include regions of almost arbitrary size and shape in their queries.

6.1.3 Experimental evaluation

We are now ready to evaluate the accuracy of Bayesian retrieval with region-based queries. For this, we start by replicating the experiments of section 5.4.1 but now considering incomplete queries, i.e. queries consisting only of a subset of the query image. All parameters were set to the values that were previously used to evaluate global similarity and a series of experiments conducted for query sets of different cardinalities. From a total of 256 non-overlapping blocks, the number of vectors contained in the query varied from 1 (0.3% of the image) to 256 (100%)[1]. Blocks were selected starting from the center in an outward spiral fashion.

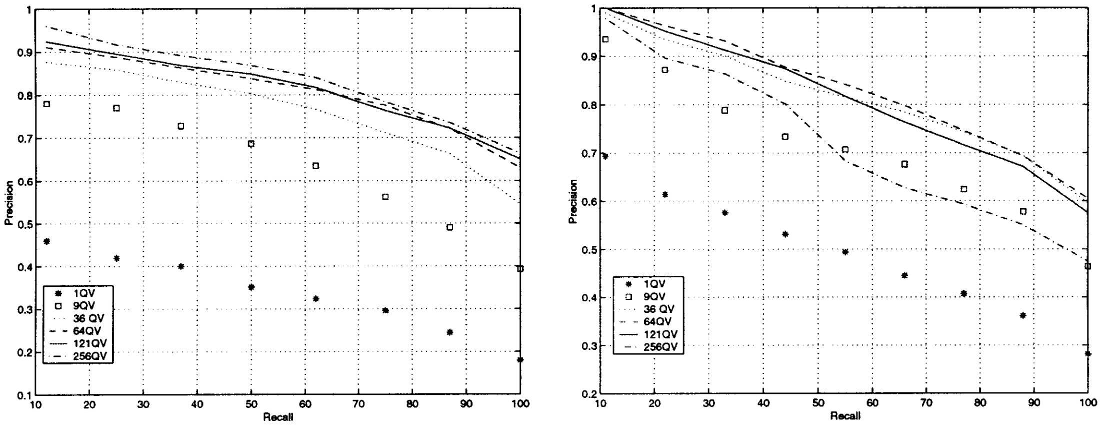


Figure 6.2: Precision/recall curves of EMM/ML on Brodatz (left) and Columbia (right). X QV means that only X feature vectors from the query image were actually included in query.

Figure 6.2 presents precision/recall curves for these experiments. The figure clearly shows that it only takes a small subset of the query feature vectors to achieve retrieval performance identical to the best possible. In both cases, 64 query vectors, 0.4% of the total number that could be extracted from the image and covering only 25% of its area, are enough. In fact, for Columbia, performance is significantly worse when all 256 vectors are considered than when only 64 are used. This is due to the fact that, in Columbia, all

[1] Notice that even 256 vectors are a very small percentage (1.5%) of the total number of blocks that could be extracted from the query image if overlapping blocks were allowed.

objects appear over a common black background that can cover a substantial amount of the image area. As Figure 6.3 illustrates, when there are large variations in scale among the different views of the object used as query, the consequent large differences in uncovered background can lead to retrieval errors. In particular, images of objects in a pose similar to that of the query are preferred to images of the query object in very different poses.

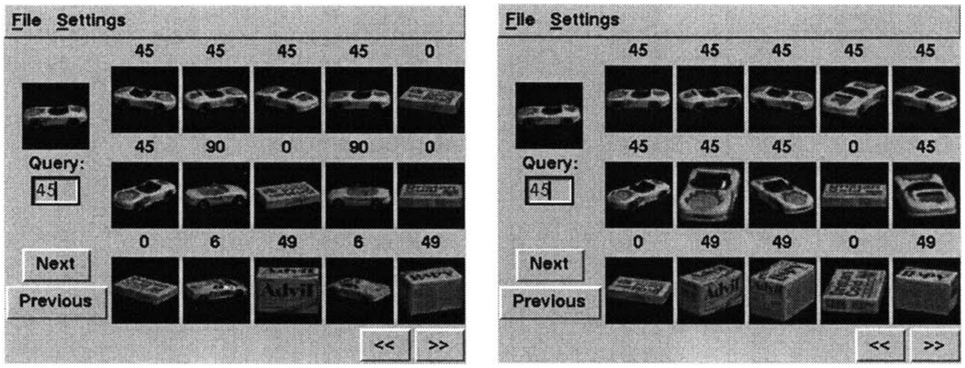


Figure 6.3: Global similarity (left) can lead to worse precision/recall than local similarity (right) on Columbia due to the large black background common to all objects.

Notice that these are two natural interpretations of similarity (prefer objects similar to the query and presented in the same pose vs. prefer the query object in different poses) and Bayesian retrieval seems to oscillate between the two. Under global similarity, the more generic interpretation (pictures of box-shaped objects in a particular orientation) is favored. When the attention of the retrieval system is focused explicitly on the query object (local query), this object becomes preferred independently of its pose. Obviously, precision/recall cannot account for these types of subtleties and the former interpretation is heavily penalized. In any case, these experiments show that, on databases like Brodatz and Columbia, Bayesian retrieval is very robust against missing data and can therefore handle local queries very easily.

A more challenging situation is that in which all images are composed by multiple visual stimulae, e.g. the mosaic databases presented in the appendix where each image is a mosaic of four Columbia or Brodatz images. The goal here is to, given a query image containing only one texture or object, to find all the images in the mosaic database that contain that query image. Figure 6.4 presents precision/recall curves for this case. Once again, retrieval is performed for query sets of various cardinalities. For comparison, we also show the performance based on global similarity, i.e. where one image of the mosaic database containing the texture or object of interest is used as query, and all feature vectors are considered in the retrieval operation.

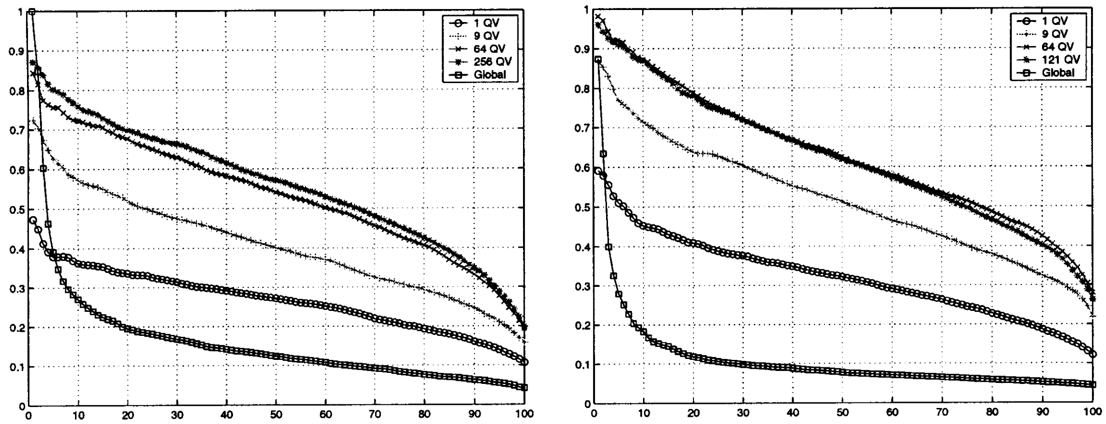


Figure 6.4: Precision/recall curves of EMM/ML on Brodatz mosaics (left) and Columbia mosaics (right). X QV means that only X feature vectors from the query image were actually included in query. For comparison, the curves obtained with global similarity are also shown.

The figure clearly shows that 1) retrieval based on local queries is significantly better than that based on global similarity, and 2) a small sample of the texture or object of interest (64 query vectors covering 25% of its area and containing 0.4% of the total number of vectors that could be extracted from it) is sufficient to achieve performance similar to the best. These results confirm the argument that Bayesian retrieval leads to effective region-based queries even for imagery composed by multiple visual stimulae.

The overwhelming superiority of local over global queries is explained by Figure 6.5, where we present the results of the two types of query for a particular object (yellow onion). When global similarity is employed, the retrieval system returns mosaics that have objects in common with the query with high probability. While this may be satisfactory in certain contexts, typically the user is interested in only one of the objects. There is however no way for the retrieval system to know this from the query alone. By selecting the region of interest, the user reduces the ambiguity of the query, enabling significantly higher retrieval

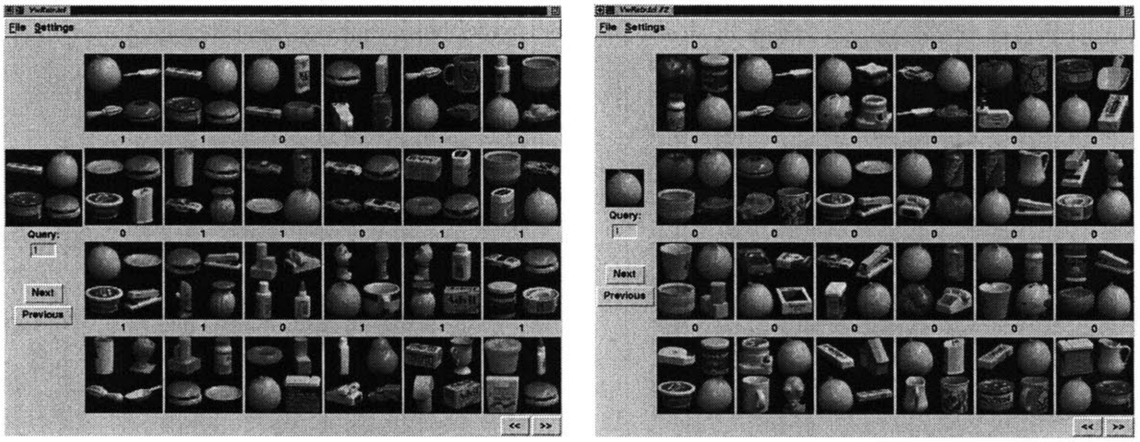


Figure 6.5: Examples of retrieval based on global (left) and local (right) similarity. In this case, the user is looking for images containing yellow onions. The number on top of each retrieved image is a flag indicating retrieval errors.

precision.

6.2 The complexity of global similarity

The results above show that, in addition to capturing both color and texture, the combination of EMM models with Bayesian retrieval is an elegant solution to the problem of local similarity. There is, however, one aspect in which this retrieval architecture is still not competitive with standard solutions like MRSAR/MD and HI: the computational cost of global similarity. We now investigate solutions to this problem.

6.2.1 Computational cost

The main limitation of (2.16) is that, because it evaluates the relevance of each query vector individually, its complexity is linear in the cardinality of the query. While this is not a significant problem for local queries since these consider only a small subset of the query image, it may become a major problem for the evaluation of global similarity, where all vectors must be taken into account. Ideally, one would like the complexity of global Bayesian similarity to be equivalent to that of standard approaches like MRSAR/MD and

HI.

Representation	Similarity function	Expression	Complexity
histogram	L^p norms	$\left(\sum_{r=1}^{R}\left\|\frac{q_r}{Q} - \frac{p_r^i}{P^i}\right\|^p\right)^{\frac{1}{p}}$	$O(R) = O(k^n)$
Gaussian	Mahalanobis	$(\hat{\mathbf{x}} - \mu_i)^T \mathbf{\Sigma}_i^{-1} (\hat{\mathbf{x}} - \mu_i)^T$	$O(n^2)$
Gaussian	ML or MDI	$\log \|\mathbf{\Sigma}_i\| + trace[\mathbf{\Sigma}_i^{-1}\hat{\mathbf{\Sigma}}_{\mathbf{x}}]$ $+(\hat{\mathbf{x}} - \mu_i)^T \mathbf{\Sigma}_i^{-1} (\hat{\mathbf{x}} - \mu_i)^T$	$O(n^2)$
EMM	ML	$\sum_{i=1}^{N} \log \sum_{c=1}^{C} \pi_c \mathcal{G}(\mathbf{x}_i, \mu_c, \mathbf{\Sigma}_c)$	$O(NCn^2)$

Table 6.1: The complexity of various retrieval solutions. See Chapters 2 and 5 for the meaning of all the symbols in the expressions of the third column. On the fourth column, n is the dimension of the feature space, k the number of cells per coordinate axis of the histogram, N the number of feature vectors on the Bayesian query, and C the number of classes of the EMM.

Table 6.1 presents a comparison of the computational complexity of the various approaches discussed so far. While the complexity of the histogram is exponential in the dimension n of the feature space, the complexity of the Gaussian model is only quadratic. This is substantially less than the linear complexity of EMM/ML on the product of the cardinality of the query with the number of classes in the EMM. In practice, a few tricks can be used to reduce this gap.

6.2.2 Practical ways to reduce complexity

It is well known from speech research that, because the Gaussian components act together to model the overall density, full covariance matrices are usually not required by a mixture model even when the features are not independent [145, 169, 106, 107]. In fact, a combination of Gaussians of diagonal covariance can model correlations between the elements of the feature vector. Since the DCT coefficients are already approximately uncorrelated [29, 74], this is particularly true in the case of EMMs.

The use of diagonal covariances has two important consequences: 1) it reduces complexity from $O(NCn^2)$ to $O(NCn)$, and 2) significantly reduces the number of parameters to be estimated (and consequently the sample sizes required for estimation) and inherent complexity of learning the database models [107]. We rely on diagonal covariances in our implementation and all the results presented in the thesis were obtained in this way. Notice that using full covariance matrices is significantly more important under the rigid single-Gaussian model, where the diagonal covariance approximation can lead to a significant loss in performance [107, 145].

The embedded nature of the representation also makes it suitable for the implementation of filtering strategies, similar to those proposed in [60, 24], to minimize the computational requirements of retrieval. These strategies start by finding the K_1 best matches considering only the first DCT coefficient. Next, among these matches, the K_2 best matches are found using the first two coefficients. The search can continue in this way until the best K_n matches are obtained at full resolution. The average complexity of the similarity evaluation will then be $O(NC(1 + 2K_1/S + 3K_2/S + \ldots + nK_{n-1}/S))$, where S is the database size. If $K_i << S$, this complexity will only be marginally larger than $O(NC)$. Since these type of strategies can also be used for histograms and the standard Gaussian model, we do not investigate this issue any further. Instead, our goal is to derive a global similarity function of complexity competitive with that of the standard retrieval approaches.

For this, it is useful to compare the cost of the different solutions for typical values of the parameters involved. For color histograming, the standard approach is to quantize the luminance axis into 8 bins and the two color components into 16 each [172]. This leads to a total of $R = 2,048$ cells. With respect to texture, the dimension of the feature space depends strongly on the particular feature transformation employed. We use the 15 dimensional vectors of the MRSAR transformation as a benchmark. In both cases, complexity is significantly smaller than that of the direct implementation of Bayesian retrieval with EMMs.

In addition to using diagonal covariances, this complexity can be reduced by considering only a small number of feature classes (between 8 and 16) and embedded subspaces (typically 16) in the EMMs. We saw in the previous chapter that this restriction does not hurt the retrieval performance in any way. Nevertheless, the complexity per feature vector of EMM/ML ($128 < Cn < 256$) is still equivalent to the total complexity of MR-SAR ($d^2 = 256$) and only a few orders of magnitude smaller than the total complexity of the histogram methods (2,048). This means that if $Cost_{hist/L^p}$, $Cost_{mrsar/md}$, and $Cost_{emm/ml}$ are, respectively, the total complexities for histogram/L^p norms, MRSAR/MD and EMM/ML, then

$$\frac{N}{2} Cost_{mrsar/md} < Cost_{emm/ml} < N Cost_{mrsar/md}$$

and

$$Cost_{emm/ml} \propto \frac{N}{8} Cost_{hist/L^p}.$$

Since the number of feature vectors N can be as large as the number of pixels in the query image, this is very problematic.

6.2.3 Asymptotic approximations for global similarity

In section 2.3.3, we saw that, as the cardinality of the query grows, Bayesian retrieval converges asymptotically to the MDI criteria, i.e. the minimization of the KL divergence between the densities of the query and retrieved image. This suggests an alternative to (2.16) for the evaluation of global similarity: start by estimating the density of the query and then evaluate the distance between that density and those in the database. If the density estimates are based on a compact feature representation, this procedure will have much smaller complexity than the direct application of (2.16). The main problem with this strategy is that there is no easy way to evaluate the KL divergence between mixtures. We next investigate when this is possible and devise approximations for when it is not.

Start by recalling (see section 2.3.3) that, when the cardinality of the query is large, the

Bayesian criteria (2.16) converges to

$$g(\mathbf{x}) = \arg\max_i \int P_{\mathbf{X}}(\mathbf{x}) \log P_{\mathbf{X}|Y}(\mathbf{x}|i) d\mathbf{x}. \tag{6.1}$$

To simplify the notation, we adopt the following conventions in the remainder of this chapter

$$P_{\mathbf{X}}(\mathbf{x}) = P(\mathbf{x}) = \sum_{j=1}^{C} P(\mathbf{x}|\omega_j) P(\omega_j) \tag{6.2}$$

and

$$P_{\mathbf{X}|Y}(\mathbf{x}|i) = P_i(\mathbf{x}) = \sum_{j=1}^{C_i} P_i(\mathbf{x}|\omega_j) P_i(\omega_j) \tag{6.3}$$

where the subscript i refers to the image class and the ω_j to the feature classes within each image class.

Given no constraints on the feature class-conditional densities $P(\mathbf{x}|\omega_j)$ and $P_i(\mathbf{x}|\omega_j)$, it is only possible to derive a generic expression for global similarity.

Lemma 2 *For a retrieval problem with the query and database densities of (6.2) and (6.3),*

$$\begin{aligned} \int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} &= \qquad (6.4) \\ &= \sum_{j,k} P(\omega_j) \left[\log P_i(\omega_k) + \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \log \frac{P_i(\mathbf{x}|\omega_k)}{P_i(\omega_k|\mathbf{x})} d\mathbf{x} \right] \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} \end{aligned}$$

where

$$\chi_k(\mathbf{x}) = \begin{cases} 1, & \textit{if } P_i(\omega_k|\mathbf{x}) \geq P_i(\omega_l|\mathbf{x}) \, \forall l \neq k \\ 0, & \textit{otherwise}, \end{cases} \tag{6.5}$$

$\chi_k = \{\mathbf{x} : \chi_k(\mathbf{x}) = 1\}$, *and*

$$P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) = \begin{cases} \frac{P(\mathbf{x}|\omega_j)}{\int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}}, & \textit{if } \mathbf{x} \in \chi_k \textit{ and } \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} > 0 \\ 0 & \textit{otherwise.} \end{cases}$$

Proof: From (6.2) and (6.3),

$$\begin{aligned}\int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} &= \sum_j P(\omega_j) \int P(\mathbf{x}|\omega_j) \log \sum_l P_i(\mathbf{x}|\omega_l) P_i(\omega_l) d\mathbf{x} \\ &= \sum_j P(\omega_j) \sum_k \int_{\chi_k} P(\mathbf{x}|\omega_j) \log \sum_l P_i(\mathbf{x}|\omega_l) P_i(\omega_l) d\mathbf{x}.\end{aligned}$$

Using Bayes rule

$$P_i(\omega_k|\mathbf{x}) = \frac{P_i(\mathbf{x}|\omega_k) P_i(\omega_k)}{\sum_l P_i(\mathbf{x}|\omega_l) P_i(\omega_l)}, \tag{6.6}$$

we have, $\forall k$ such that $P_i(\omega_k|\mathbf{x}) \neq 0$,

$$\sum_l P_i(\mathbf{x}|\omega_l) P_i(\omega_l) d\mathbf{x} = \frac{P_i(\mathbf{x}|\omega_k) P_i(\omega_k)}{P_i(\omega_k|\mathbf{x})}.$$

Since $\sum_k P_i(\omega_k|\mathbf{x}) = 1$, from the definition of χ_k we obtain $P_i(\omega_k|\mathbf{x}) > 0$, $\forall \mathbf{x} \in \chi_k$, and

$$\begin{aligned}&\int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \\ &= \sum_j P(\omega_j) \sum_k \int_{\chi_k} P(\mathbf{x}|\omega_j) \log \frac{P_i(\mathbf{x}|\omega_k) P_i(\omega_k)}{P_i(\omega_k|\mathbf{x})} d\mathbf{x} \\ &= \sum_j P(\omega_j) \sum_k \left[\log P_i(\omega_k) \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} + \int_{\chi_k} P(\mathbf{x}|\omega_j) \log \frac{P_i(\mathbf{x}|\omega_k)}{P_i(\omega_k|\mathbf{x})} d\mathbf{x} \right] \\ &= \sum_j P(\omega_j) \sum_k \left[\log P_i(\omega_k) + \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \log \frac{P_i(\mathbf{x}|\omega_k)}{P_i(\omega_k|\mathbf{x})} d\mathbf{x} \right] \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}. \, \square\end{aligned}$$

Equation (6.4) reveals that there are two fundamental components to global similarity. The first

$$\sum_{j,k} P(\omega_j) \log P_i(\omega_k) \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}$$

is a function of the feature class probabilities, the second

$$\sum_{j,k} P(\omega_j) \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \log \frac{P_i(\mathbf{x}|\omega_k)}{P_i(\omega_k|\mathbf{x})} d\mathbf{x} \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}$$

a function of the class-conditional densities. The overall similarity is strongly dependent on the partition $\{\chi_1, \ldots, \chi_{C_i}\}$ of the feature space determined by $P_i(\mathbf{x})$, the term

$$\int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}$$

weighting the contribution of each cell according to the fraction of the query probability that it contains. In particular, if $\mathcal{S}(\omega_j)$ is the support set of $P(\mathbf{x}|\omega_j)$, then

$$\begin{array}{ll} \int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x} = 0, & \text{if } \mathcal{S}(\omega_j) \cap \chi_k = \emptyset \\ \int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x} = 1, & \text{if } \mathcal{S}(\omega_j) \subset \chi_k \\ \int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x} \in (0,1), & \text{otherwise,} \end{array}$$

and $\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x}$ can be seen as a measure of overlap between $P(\mathbf{x}|\omega_j)$ and the cell χ_k determined by $P_i(\mathbf{x}|\omega_k)$.

Histograms

When all image classes share the same feature class-conditional densities and the feature space is divided into a collection of disjoint cells, the evaluation of global similarity is straightforward. This is the case of the standard histogram model and VQ label histograms for a fixed quantization of the feature space.

Lemma 3 *If all mixture densities define the same hard partition*

$$\chi_k(\mathbf{x}) = \begin{cases} 1, & \textit{if } P(\omega_l|\mathbf{x}) = P_i(\omega_l|\mathbf{x}) = \delta_{k,l} \, \forall i \\ 0, & \textit{otherwise,} \end{cases} \tag{6.7}$$

where $\delta_{k,l}$ is the Kronecker delta function (2.3), then

$$\int P(\mathbf{x}) \log P_i(\mathbf{x})d\mathbf{x} = \sum_j P(\omega_j) \log P_i(\omega_j) + \sum_j P(\omega_j) \int_{\chi_j} P(\mathbf{x}|\omega_j) \log P_i(\mathbf{x}|\omega_j)d\mathbf{x}. \tag{6.8}$$

Proof: Using the same argument as in the proof of Theorem 5, we assume without loss of generality that all the classes in all mixture models have non-zero probability, i.e.

$$P(\omega_l) > 0 \text{ and } P_i(\omega_l) > 0, \; \forall l, i.$$

It follows from (6.6) that $P(\omega_k|\mathbf{x}) = 1$ if and only if

$$\sum_{l \neq k} P_i(\mathbf{x}|\omega_l) P_i(\omega_l) = 0.$$

Since all the terms in the summation are non-negative, this implies

$$P_i(\mathbf{x}|\omega_l) = 0 \ \forall l \neq k.$$

I.e., for a hard partition such as (6.7), the support sets of $P(\mathbf{x}|\omega_k)$ and $P_i(\mathbf{x}|\omega_k)$ are contained in χ_k. Hence

$$\int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} = \delta_{k,j}$$

and, since $P_i(\omega_k|\mathbf{x}) = 1, \forall \mathbf{x} \in \chi_k$, (6.8) follows from Lemma 2. □

Because when all image classes share the same feature-class conditional densities, the second term of (6.8) does not depend on i, this lemma implies that

$$\begin{aligned} \arg\max_i \int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} &= \arg\max_i \sum_j P(\omega_j) \log P_i(\omega_j) \\ &= \arg\min_i \sum_j P(\omega_j) \log \frac{P(\omega_j)}{P_i(\omega_j)}. \end{aligned} \tag{6.9}$$

Hence, if the feature space is first vector quantized and all image classes represented by label histograms, Bayesian retrieval is equivalent to minimizing the KL divergence between those label histograms. This is an interesting result from the computational point of view, since the complexity of this operation is $O(C)$, where C is the number of VQ cells, as opposed to the $O(NCn^2)$ complexity inherent to the straightforward application of the Bayesian criteria.

Gaussian mixtures

A much more challenging case occurs when we lift the restrictions of a common hard partition and consider generic Gaussian mixtures. We now concentrate in this case, starting with a preliminary result.

Lemma 4 *For any probability density $P(\mathbf{x})$, $\mathbf{x} \in R^n$, $\alpha \in R^n$, $B \in R^{n \times n}$ and set χ, if*

$$\int_\chi P(\mathbf{x}) d\mathbf{x} = 1,$$

then

$$\int_\chi P(\mathbf{x})(\mathbf{x} - \alpha)^T B(\mathbf{x} - \alpha) d\mathbf{x} = trace[B\hat{\Sigma}_{\mathbf{x}}] + (\hat{\mu}_{\mathbf{x}} - \alpha)^T B(\hat{\mu}_{\mathbf{x}} - \alpha),$$

where

$$\hat{\mu}_{\mathbf{x}} = \int_\chi P(\mathbf{x})\mathbf{x}\, d\mathbf{x}$$

$$\hat{\Sigma}_{\mathbf{x}} = \int_\chi P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})^T d\mathbf{x}.$$

Proof:

$$\begin{aligned}
&\int_\chi P(\mathbf{x})(\mathbf{x} - \alpha)^T B(\mathbf{x} - \alpha) d\mathbf{x} = \\
&= \int_\chi P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}} + \hat{\mu}_{\mathbf{x}} - \alpha)^T B(\mathbf{x} - \hat{\mu}_{\mathbf{x}} + \hat{\mu}_{\mathbf{x}} - \alpha) d\mathbf{x} \\
&= \int_\chi P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})^T B(\mathbf{x} - \hat{\mu}_{\mathbf{x}}) d\mathbf{x} + 2\int_\chi P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})^T B(\hat{\mu}_{\mathbf{x}} - \alpha) d\mathbf{x} + (\hat{\mu}_{\mathbf{x}} - \alpha)^T B(\hat{\mu}_{\mathbf{x}} - \alpha) \\
&= trace\left[B \int_\chi P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})^T d\mathbf{x}\right] + 2(\hat{\mu}_{\mathbf{x}} - \hat{\mu}_{\mathbf{x}})^T B(\hat{\mu}_{\mathbf{x}} - \alpha) + (\hat{\mu}_{\mathbf{x}} - \alpha)^T B(\hat{\mu}_{\mathbf{x}} - \alpha) \\
&= trace[B\hat{\Sigma}_{\mathbf{x}}] + (\hat{\mu}_{\mathbf{x}} - \alpha)^T B(\hat{\mu}_{\mathbf{x}} - \alpha).\square
\end{aligned}$$

This lemma allows us to specialize (6.4) to Gaussian mixtures.

Lemma 5 *For a retrieval problem with the query densities of (6.2) and Gaussian mixtures for the database densities (6.3),*

$$P_i(\mathbf{x}) = \sum_{k=1}^{C_i} \mathcal{G}(\mathbf{x}, \mu_{i,k}, \mathbf{\Sigma}_{i,k}) P_i(\omega_k),$$

where $\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma})$ is as defined in (2.5),

$$\begin{aligned}
&\int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \\
&= \sum_{j,k} P(\omega_j) \log P_i(\omega_k) \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} +
\end{aligned}$$

$$
\begin{aligned}
+ & \sum_{j,k} P(\omega_j) \left[\log \mathcal{G}(\hat{\mu}_{q,j,k}, \mu_{i,k}, \mathbf{\Sigma}_{i,k}) - \frac{1}{2} trace[\mathbf{\Sigma}_{i,k}^{-1} \hat{\Sigma}_{q,j,k}] \right] \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} \\
- & \sum_{j,k} P(\omega_j) \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \log P_i(\omega_k|\mathbf{x}) d\mathbf{x} \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} \qquad (6.10)
\end{aligned}
$$

where

$$
\hat{\mu}_{q,j,k} = \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \mathbf{x} \, d\mathbf{x}, \qquad (6.11)
$$

$$
\hat{\Sigma}_{q,j,k} = \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1)(\mathbf{x} - \hat{\mu}_{q,j,k})(\mathbf{x} - \hat{\mu}_{q,j,k})^T d\mathbf{x}, \qquad (6.12)
$$

and χ_k *and* $P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1)$ *are as defined in Lemma 2.*

Proof: Since $P_i(\mathbf{x}|\omega_k) = \mathcal{G}(\mathbf{x}, \mu_{i,k}, \mathbf{\Sigma}_{i,k})$ and $\int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) d\mathbf{x} = 1$, simple application of the previous lemma results in

$$
\begin{aligned}
& \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \log P_i(\mathbf{x}|\omega_k) d\mathbf{x} = \\
= & \log \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}_{i,k}|}} \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) d\mathbf{x} \\
& - \frac{1}{2} \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1)(\mathbf{x} - \mu_{i,k})^T \mathbf{\Sigma}_{i,k}^{-1} (\mathbf{x} - \mu_{i,k})^T d\mathbf{x} \\
= & \log \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}_{i,k}|}} - \frac{1}{2} trace[\mathbf{\Sigma}_{i,k}^{-1} \hat{\Sigma}_{q,j,k}] - \frac{1}{2} (\hat{\mu}_{q,j,k} - \mu_{i,k})^T \mathbf{\Sigma}_{i,k}^{-1} (\hat{\mu}_{q,j,k} - \mu_{i,k}) \\
= & \log \mathcal{G}(\hat{\mu}_{q,j,k}, \mu_{i,k}, \mathbf{\Sigma}_{i,k}) - \frac{1}{2} trace[\mathbf{\Sigma}_{i,k}^{-1} \hat{\Sigma}_{q,j,k}].
\end{aligned}
$$

The lemma follows by simple algebraic manipulation of (6.4). □

It is interesting to analyze each of the terms in (6.10). Consider the query feature class w_j and the database feature class w_k. The first term in the equation is simply a measure of the similarity between the class probabilities $P(\omega_j)$ and $P_i(\omega_k)$ weighted by measure of overlap $\int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}$. This term is equivalent to that appearing in (6.9) but accounts for the fact that the partitions defined by the query and image class densities are now not aligned.

Comparing the term in square brackets with (2.23), it is straightforward to show that this term is equivalent to the similarity, under Bayesian retrieval, between the Gaussian $P_i(\mathbf{x}|\omega_k)$ and a Gaussian with parameters $\hat{\mu}_{q,j,k}$ and $\hat{\mathbf{\Sigma}}_{q,j,k}$. From (6.11) and (6.12), these are simply the mean and covariance of $\mathbf{x}$ according to $P(\mathbf{x}|\omega_j)$ given that $\mathbf{x} \in \chi_k$. Hence, the second term is simply a measure of the similarity between the feature class conditional densities inside the cell defined by $P_i(\mathbf{x}|\omega_k)$. Once again, this measure is weighted by the amount of overlap between the two densities.

Finally, the third term weights the different cells χ_k according to the ambiguity of their ownership. Recall that, $\forall \mathbf{x} \in \chi_k$, $P_i(\omega_k|\mathbf{x}) > P_i(\omega_l|\mathbf{x})$, $\forall l \neq k$. If $P_i(\omega_k|\mathbf{x}) = 1$, the cell is uniquely assigned to ω_k and this term will be zero. If, on the other hand, $P_i(\omega_k|\mathbf{x}) < 1$, then the cell will also be assigned to other classes and the overall likelihood will increase.

While providing insight on the different factors involved in global similarity, (6.10) is not very useful from a computational standpoint since the integrals that it involves do not have a closed-form expression. There is, however, one particular case where a closed-form solution is available: the case where all mixture models are vector quantizers.

Vector quantizers

Using Theorem 5, the VQ case can be analyzed by assuming Gaussian feature class-conditional densities and investigating what happens when all covariances tend to zero. This leads to the following result.

Lemma 6 *For a retrieval problem with Gaussian mixtures for the query (6.2) and database densities (6.3)*

$$P(\mathbf{x}) = \sum_{j=1}^{C} \mathcal{G}(\mathbf{x}, \mu_{q,j}, \epsilon\mathbf{\Sigma}_{q,j})P(\omega_j)$$

$$P_i(\mathbf{x}) = \sum_{k=1}^{C_i} \mathcal{G}(\mathbf{x}, \mu_{i,k}, \epsilon\mathbf{\Sigma}_{i,k})P_i(\omega_k)$$

where $\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma})$ *is defined in (2.5), when* $\epsilon \to 0$

$$\begin{aligned}\int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} &= \\ &= \sum_j P(\omega_j) \log P_i(\omega_{\alpha(j)}) \\ &+ \sum_j P(\omega_j) \lim_{\epsilon \to 0} \left[\log \mathcal{G}(\hat{\mu}_{q,j,\alpha(j)}, \mu_{i,\alpha(j)}, \epsilon\mathbf{\Sigma}_{i,\alpha(j)}) - \frac{1}{2\epsilon} trace[\mathbf{\Sigma}_{i,\alpha(j)}^{-1} \hat{\Sigma}_{q,j,\alpha(j)}] \right],\end{aligned}$$

where χ_k is as defined in Lemma 2, $\hat{\mu}_{q,j,\alpha(j)}$ and $\hat{\Sigma}_{q,j,\alpha(j)}$ as defined in Lemma 5, and

$$\alpha(j) = k \text{ such that } ||\mu_{q,j} - \mu_{i,k}||^2_{\mathbf{\Sigma}_{i,k}} < ||\mu_{q,j} - \mu_{i,l}||^2_{\mathbf{\Sigma}_{i,l}} \, \forall l \neq k.$$

Proof: When $\epsilon \to 0$,

$$\mathcal{G}(\mathbf{x}, \mu_{q,j}, \epsilon\mathbf{\Sigma}_{q,j}) \to \delta(\mathbf{x} - \mu_{q,j})$$

and since, from the definition of the delta function,

$$\int f(\mathbf{x})\delta(\mathbf{x} - \mu)d\mathbf{x} = f(\mu),$$

it follows that

$$\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x} \to \chi_k(\mu_{q,j}).$$

On the other hand, from Theorem 5 and the definition of χ_k, if $\epsilon \to 0$ then

$$P_i(\omega_k|\mathbf{x}) \to 1 \, \forall \mathbf{x} \in \chi_k,$$

and $\chi_k(\mu_{q,j}) = 1$ if and only if

$$(\mu_{q,j} - \mu_{i,k})\mathbf{\Sigma}_{i,k}^{-1}(\mu_{q,j} - \mu_{i,k}) < (\mu_{q,j} - \mu_{i,l})\mathbf{\Sigma}_{i,l}^{-1}(\mu_{q,j} - \mu_{i,l}), \, \forall l \neq k.$$

The lemma follows from the application of these results to (6.10). □

We are now ready to derive a closed-form expression for global similarity under Bayesian retrieval with VQ density estimates.

Theorem 6 *For a retrieval problem with VQ density estimates for the query (6.2) and database densities (6.3)*

$$P(\mathbf{x}) = \sum_{j=1}^{C} \delta(\mathbf{x}, \mu_{q,j})P(\omega_j)$$

$$P_i(\mathbf{x}) = \sum_{k=1}^{C_i} \delta(\mathbf{x} - \mu_{i,k})P_i(\omega_k),$$

when evaluating global similarity the Bayesian retrieval criteria reduces to

$$\begin{aligned}\arg\max_i \int P(\mathbf{x})\log P_i(\mathbf{x})d\mathbf{x} &= \\ &= \arg\min_i \lim_{\lambda\to\infty}\left\{\sum_j P(\omega_j)\log\frac{P(\omega_j)}{P_i(\omega_{\alpha(j)})} + \lambda\sum_j P(\omega_j)||\mu_{q,j}-\mu_{i,\alpha(j)}||^2\right\} \qquad (6.13)\end{aligned}$$

where

$$\alpha(j) = k \textit{ such that } ||\mu_{q,j}-\mu_{i,k}||^2 < ||\mu_{q,j}-\mu_{i,l}||^2, \forall l \neq k.$$

Proof: From (6.11) and (6.12), when $\epsilon \to 0$

$$\hat{\mu}_{q,j,\alpha(j)} \to \mu_{q,j}$$

$$\hat{\Sigma}_{q,j,\alpha(j)} \to \epsilon\mathbf{\Sigma}_{q,j}.$$

Using Lemma 6,

$$\begin{aligned}\arg\max_i \int P(\mathbf{x})\log P_i(\mathbf{x})d\mathbf{x} &= \\ = \arg\max_i &\left\{\sum_j P(\omega_j)\log P_i(\omega_{\alpha(j)}) + \sum_j P(\omega_j)\lim_{\epsilon\to 0}\log\mathcal{G}(\mu_{q,j},\mu_{i,\alpha(j)},\epsilon\mathbf{\Sigma}_{i,\alpha(j)})\right. \\ &\left. -\sum_j P(\omega_j)\frac{1}{2}trace[\mathbf{\Sigma}_{i,\alpha(j)}^{-1}\mathbf{\Sigma}_{q,j}]\right\}.\end{aligned}$$

Since, for a vector quantizer, $\mathbf{\Sigma}_{i,k} = \mathbf{\Sigma}_{q,j} = \mathbf{I}, \forall k, j$, the third term on the right-hand side of the above equation does not depend on i, and setting $\lambda = 1/2\epsilon$ leads to

$$\begin{aligned}\arg\max_i \int P(\mathbf{x})\log P_i(\mathbf{x})d\mathbf{x} &= \\ &= \arg\max_i\left\{\sum_j P(\omega_j)\left[\log P_i(\omega_{\alpha(j)}) - \lim_{\lambda\to\infty}\lambda||\mu_{q,j}-\mu_{i,\alpha(j)}||^2\right]\right\} \\ &= \arg\min_i \lim_{\lambda\to\infty}\left\{\sum_j P(\omega_j)\log\frac{P(\omega_j)}{P_i(\omega_{\alpha(j)})} + \lambda\sum_j P(\omega_j)||\mu_{q,j}-\mu_{i,\alpha(j)}||^2\right\}.\square\end{aligned}$$

The theorem states that, for VQ-based density estimates, Bayesian retrieval is equivalent to a constrained optimization problem [11]. Given a query VQ and a VQ associated with a database image class, one starts by vector quantizing the codewords of the former according to the latter, i.e. each codeword of the query VQ is assigned to the cell of the database VQ whose centroid is closest to it. The best database VQ is the one that minimizes a sum of two terms resulting from this procedure: a term that accounts for the average distortion of the quantization ($\sum_j P(\omega_j)||\mu_{q,j} - \mu_{i,\alpha(j)}||^2$) and the KL divergence between the feature-class probability distributions. λ is a Lagrange multiplier that weighs the contribution of the two terms.

By making $\lambda \to \infty$, all the emphasis is placed on the average quantization distortion. This leads to two distinct situations of practical interest. The first is when the two quantizers share the same codewords. In this case, the quantization distortion is null and the cost function becomes that of (6.9), i.e. the KL divergence between label histograms. Since equal quantizers with equal codewords define equal partitions of the feature space, this situation is equivalent to that of histogramming and the result is, therefore, not surprising.

If the quantizers have different codewords (and consequently define different partitions), the quantization distortion becomes predominant and the retrieval criteria becomes

$$\arg\max_i \int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \arg\min_i \sum_j P(\omega_j)||\mu_{q,j} - \mu_{i,\alpha(j)}||^2.$$

Computationally, this reduces the complexity of the retrieval operation from $O(NCn)$ to $O(C^2n)$. Since C, the number of feature classes, is fixed and orders of magnitude smaller than the cardinality of the query, N, the resulting savings are very significant. In fact, using the typical values of section 6.2.2,

$$4Cost_{mrsar/md} < Cost_{emm/ml} < 16Cost_{mrsar/md},$$

$$Cost_{hist/L^p} < Cost_{emm/ml} < 2Cost_{hist/L^p},$$

rendering the complexity of Bayesian retrieval with EMM similar to that of the standard approaches.

The asymptotic likelihood approximation

Vector quantization is a case of particular interest not only because it has a closed-form solution for global similarity, but also because the analysis performed for VQ provides insight on how to approximate (6.10) for generic Gaussian mixtures. In particular, Lemma 6 suggests the following approximation.

Definition 5 *Given a retrieval problem with Gaussian mixtures for the query (6.2) and database densities (6.3)*

$$P(\mathbf{x}) = \sum_{j=1}^{C} \mathcal{G}(\mathbf{x}, \mu_{q,j}, \mathbf{\Sigma}_{q,j})P(\omega_j)$$

$$P_i(\mathbf{x}) = \sum_{k=1}^{C_i} \mathcal{G}(\mathbf{x}, \mu_{i,k}, \mathbf{\Sigma}_{i,k})P_i(\omega_k),$$

the asymptotic likelihood approximation (ALA) is defined by

$$\begin{aligned} \int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} \approx & \\ \approx & \sum_j P(\omega_j) \log P_i(\omega_{\alpha(j)}) \\ + & \sum_j P(\omega_j) \left[\log \mathcal{G}(\mu_{q,j}, \mu_{i,\alpha(j)}, \mathbf{\Sigma}_{i,\alpha(j)}) - \frac{1}{2} trace[\mathbf{\Sigma}_{i,\alpha(j)}^{-1} \mathbf{\Sigma}_{q,j}] \right], \end{aligned}$$

where

$$\alpha(j) = k \text{ such that } ||\mu_{q,j} - \mu_{i,k}||^2_{\mathbf{\Sigma}_{i,k}} < ||\mu_{q,j} - \mu_{i,l}||^2_{\mathbf{\Sigma}_{i,l}}, \forall l \neq k.$$

A comparison of the ALA with the true likelihood of (6.10) reveals two assumptions underlying this approximation.

Assumption 3 *Each cell χ_k of the partition determined by $P_i(\mathbf{x})$ is assigned to one feature class with probability one, i.e.*

$$P_i(\omega_k|\mathbf{x}) = 1, \ \forall \mathbf{x} \in \chi_k.$$

Assumption 4 *The support set of each feature class-conditional density of the query mixture is entirely contained in a single cell χ_k of the partition determined by $P_i(\mathbf{x})$. I.e.,*

$$\forall j \; \exists k : \mathcal{S}(\omega_j) \subset \chi_k.$$

Under Assumption 3, the third term of (6.10) vanishes. Under Assumption 4, $\int_{\chi_k} P(\mathbf{x}|\omega_j) = \delta_{k,\alpha(j)}$, $\hat{\mu}_{q,j,\alpha(j)} = \mu_{q,j}$, and $\hat{\Sigma}_{q,j,\alpha(j)} = \mathbf{\Sigma}_{q,j}$. Taken together, these equalities lead to the ALA. While both assumptions are valid in the VQ case, the ALA does not necessarily imply a VQ model. In particular, all feature class-conditional densities are allowed to have non-zero covariances. However, Assumption 3 will only be reasonable if the feature class-conditional densities of $P_i(\mathbf{x})$ have reduced overlap. This implies that the distance between each pair of $\mu_{i,q}$ should be larger than the spread of the associated Gaussians. A 1-D illustration of this effect is provided by Figure 6.6, where we show two Gaussians class-conditional likelihoods and the posterior probability function $P_i(\omega_0|\mathbf{x})$ for class 0. As the separation between the Gaussians increases, the posterior probability changes more abruptly and the partition becomes harder.

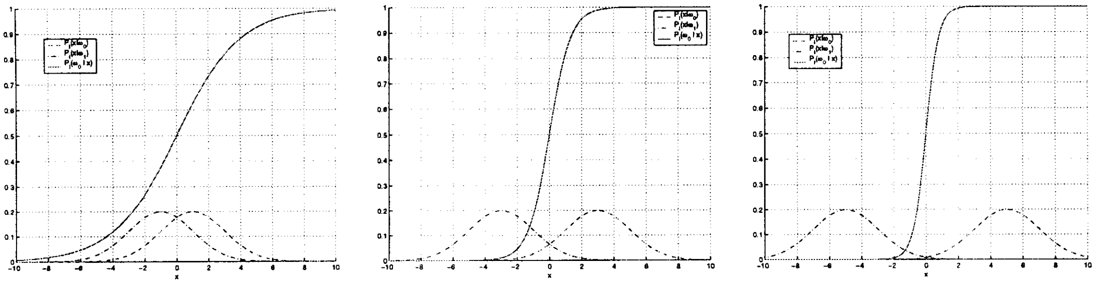


Figure 6.6: Impact of the separation between two Gaussian class conditional likelihoods on the partition of the feature space that they determine.

Assumption 4 never really holds for Gaussian mixtures, since Gaussians have infinite support. However, if Assumption 3 holds and, in addition, the spread of the Gaussians in

$P(\mathbf{x})$ is much smaller than the cells χ_k, then

$$\int_{\chi_{\alpha}(j)} P(\mathbf{x}|\omega_j)d\mathbf{x} \approx 1$$

with high probability.

In summary, the crucial assumption for the validity of the ALA is that the Gaussian feature class-conditional densities within each model have reduced overlap. The plausibility of this assumption grows with the dimension of the feature space, since high-dimensional spaces are more sparsely populated than low-dimensional ones. This is already visible in Figure 5.3, where it is clear that as the dimension of the space grows the Gaussians tend to have smaller overlap. To validate this point with more concrete evidence, we performed the following experiment

- a $10,000$ point sample was drawn from the mixture model of Figure 5.3;
- for each sample point $\mathbf{x}_i, i = 1, \ldots, 10,000$, we evaluated the maximum posterior class-assignment probability $max_k P(\omega_k|\mathbf{x}_i)$;
- the maximum posterior probabilities were histogramed.

The experiment was repeated for several mixture models obtained by restricting the EMM of Figure 5.3 to an increasing number of subspaces. Figure 6.7 presents the histograms of the maximum posterior probability obtained with 2, 4, 8, 16, 32, and 64 subspaces. It is clear that, in high-dimensional spaces, Assumption 3 is realistic.

6.2.4 Experimental evaluation

We are now ready to conclude the experimental evaluation of probabilistic retrieval with EMMs. Since the Brodatz and Columbia databases contain only specific types of visual concepts (textures and objects), are organized into relatively unambiguous classes, and each of their images consists of only one concept, these databases provide a controlled environment that enables important insights on the different retrieval solutions. However, because realistic image retrieval rarely occurs under such controlled circumstances, it is

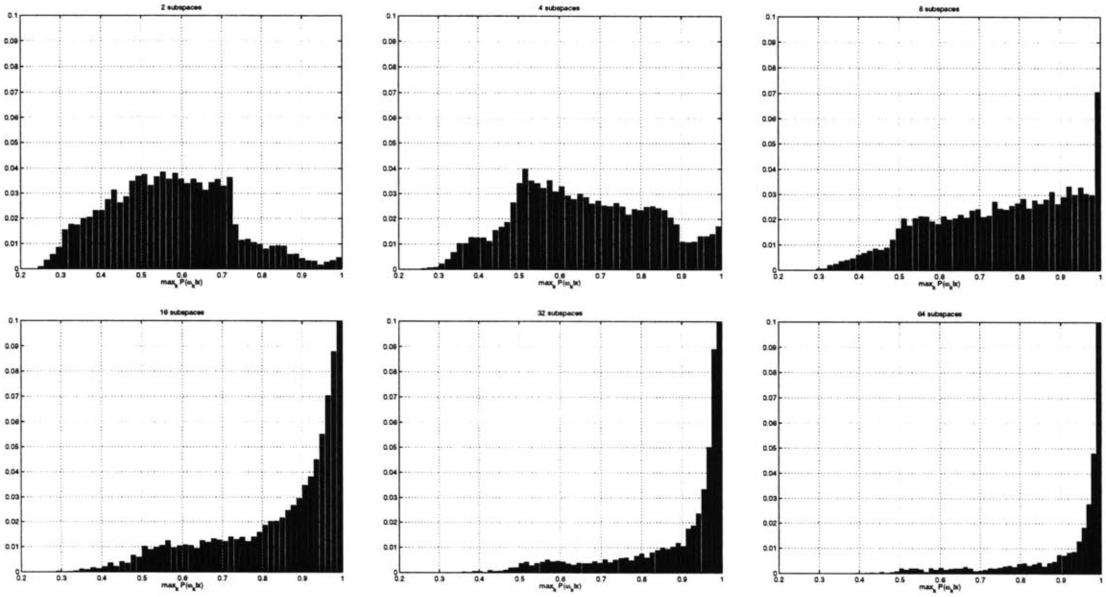


Figure 6.7: Maximum class posterior probability histograms for different numbers of subspaces of the EMM of Figure 5.3.

important to validate the results obtained so far with evaluation on a generic database. In particular, it is important to consider databases that require joint modeling of texture and color. In this section, we consider one such database (Corel) and compare the performance of probabilistic retrieval against that of the domain-specific approaches discussed so far (HI and MRSAR/MD) and the two other approaches that, to the best of our knowledge, represent the state of the art in the joint modeling of the two attributes: color autocorrelograms and linear combinations of color and texture.

In these experiments, we used mixtures of 8 Gaussians and a spacing of four pixels between consecutive training samples. The implementations of MRSAR and HI were as discussed above, in the latter case we used a histogram with 512 bins. For color autocorrelograms, we followed the implementation of Huang et al. [66]. One of the limitations of the autocorrelogram is that, because each of its entries is a probability conditioned on a different event, it cannot be combined with probabilistic retrieval criteria such as ML. We therefore relied on the variation of the L^1 norm proposed in [66] as a similarity criteria. In order to make a fair comparison, we restricted its region of support to be the 8×8 pixel window also used by the embedded mixtures, i.e. the set of distances used for computing the autocorrelogram was $D = 1, 2, 3$. Overall, the autocorrelogram contained 2,048 bins.

To combine linearly color and texture, we started by evaluating all the distances between query and database entries according to both HI and MRSAR/MD. For each query, we then normalized all distances by their mean and variance, clipped all values with magnitude larger than three standard deviations, and mapped the resulting interval into $[0, 1]$. This is a standard normalization to ensure that the distances relative to the two attributes are of the same order of magnitude [72, 44, 150]. An overall distance was then computed for each entry in the database, according to

$$d' = (1 - w)d_c + wd_t$$

where d_c and d_t are, respectively, the normalized distances according to HI and MRSAR/MD, and $w \in [0, 1]$ a pre-defined weight. These distances were then used to rank all the entries and measure precision/recall.

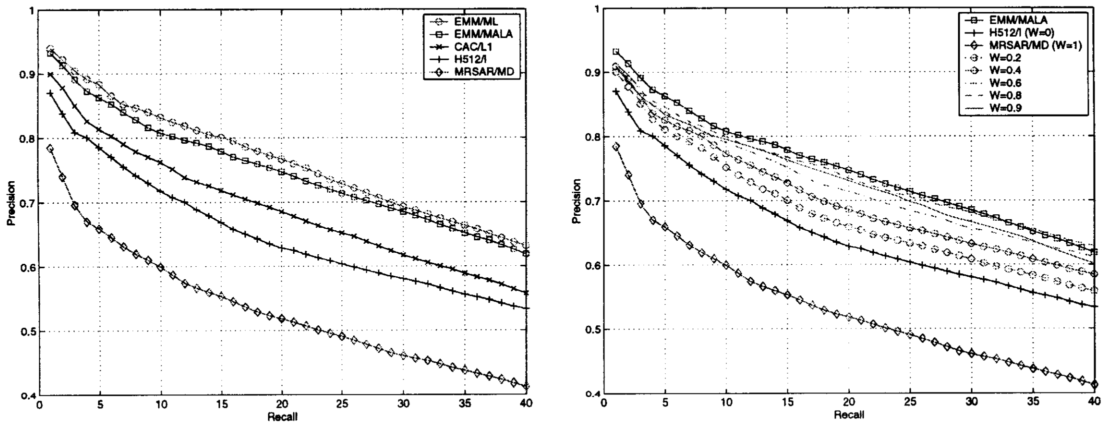


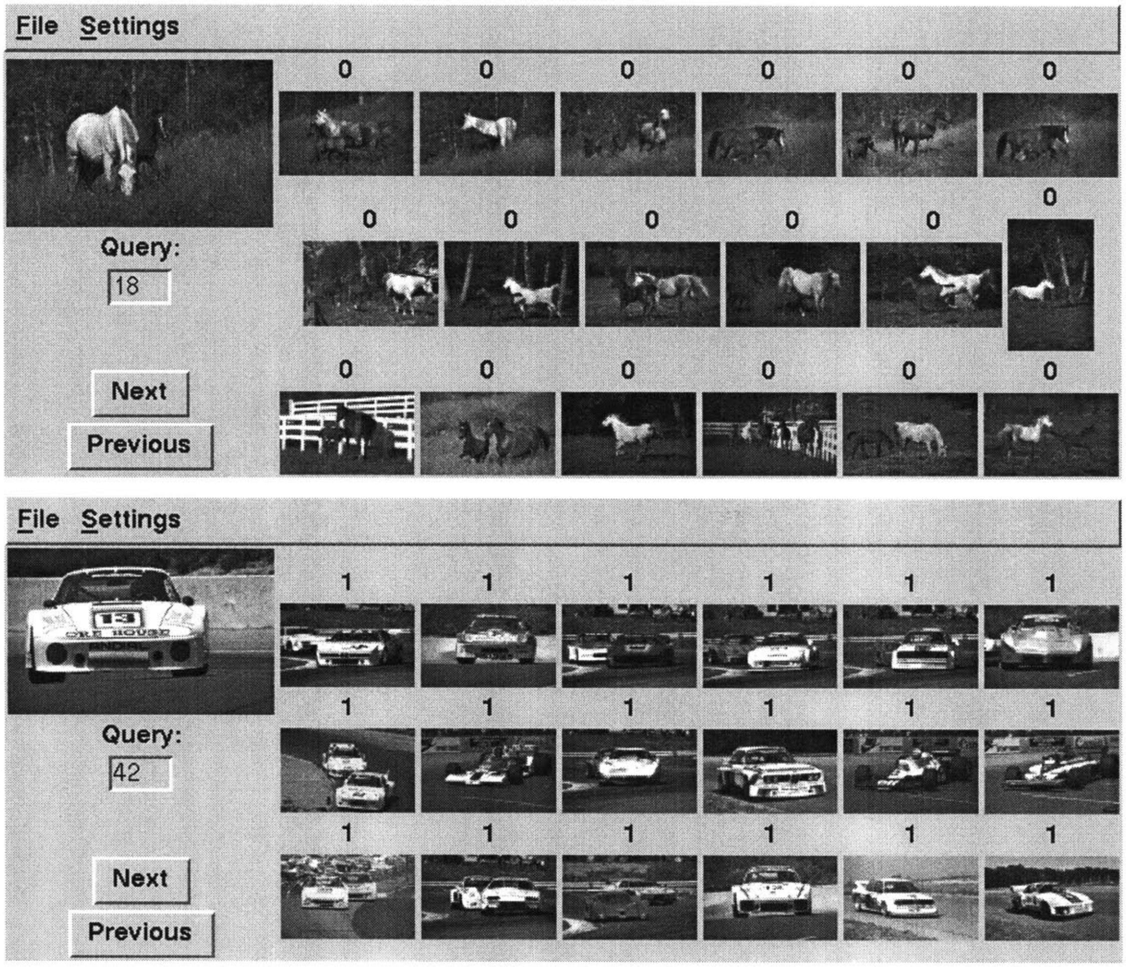
Figure 6.8: Left: precision/recall on Corel for MRSAR/MD, HI, color autocorrelogram (CAC), EMM/ML, and EMM/MALA. Right: comparison of precision /recall achieved with EM/MALA and linear weighting of MRSAR/MD and HI for different weights.

The left plot on Figure 6.8 presents the precision/recall curves for the different retrieval solutions. It is clear that the texture model alone performs very poorly, color histogramming does significantly better, and the autocorrelogram further improves performance by about 5%. However, all these approaches are significantly less effective than either EMM/ML or

EM/MALA (where we maximize the asymptotic likelihood approximation discussed in the previous section). Furthermore, there is no significant difference between the two EMM approaches. This confirms the argument that, for global queries, 1) ALA is a good approximation to the true likelihood, and 2) EMM/MALA is the best overall solution when one takes computational complexity into account.

Finally, the right plot on the figure compares the precision/recall curves of EMM/MALA with those obtained by linear weighting of the color and texture distances. Several curves are shown for values of $w \in [0, 1]$. It is clear that the performance of the latter approach is never better than that of EM/MALA. Given that, in a realistic retrieval scenario, the value of the optimal weight is not known, there are no intuitive ways to determine it, and the linear combination always requires an increase in complexity (distances have to be computed according to the two representations), we see no reason to prefer these types of solutions to probabilistic retrieval.

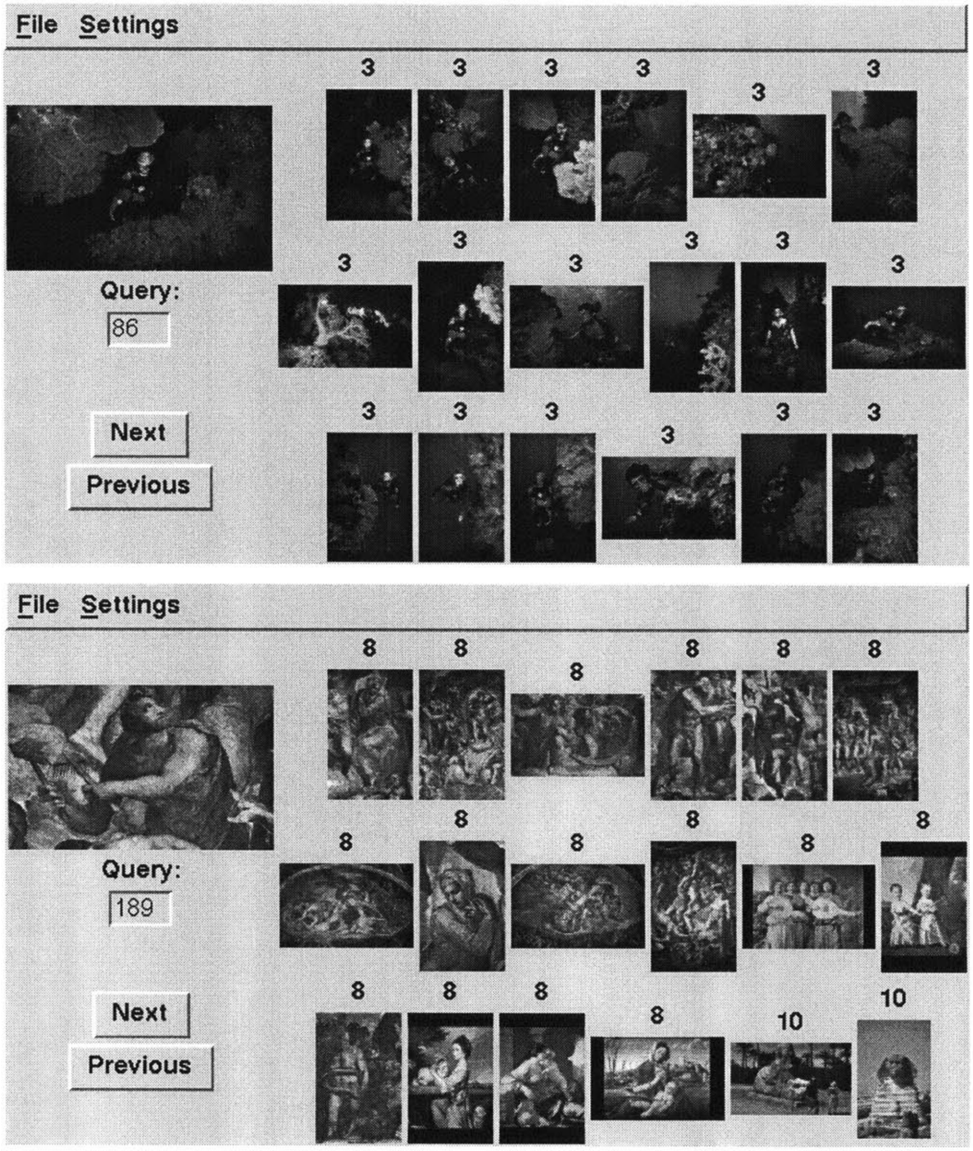
We conclude this chapter by giving some visual examples of the outcome of queries in the Corel database. Figures 6.9 to 6.11 present typical results for queries with horses, cars, diving scenes, gardens, and paintings. These pictures illustrate some of the nice properties of the probabilistic retrieval formulation: robustness to changes in object position and orientation, robustness against the presence of distracting objects in the background, good performance even when there are large chunks of missing data in the query (notice that, in the diving example, even though almost no sea is visible in the query, the retrieved images are all from the right class and most contain large patches of blue), and perceptually intuitive errors (in the painting example, two pictures of the sphinx - pyramids class - are returned after all the paintings of human figures).



Query: 18

Query: 42

Figure 6.9: Outcome of queries on Corel for horses and cars.



Query: 86

Query: 189

Figure 6.10: Outcome of queries on Corel for diving scenes and paintings.

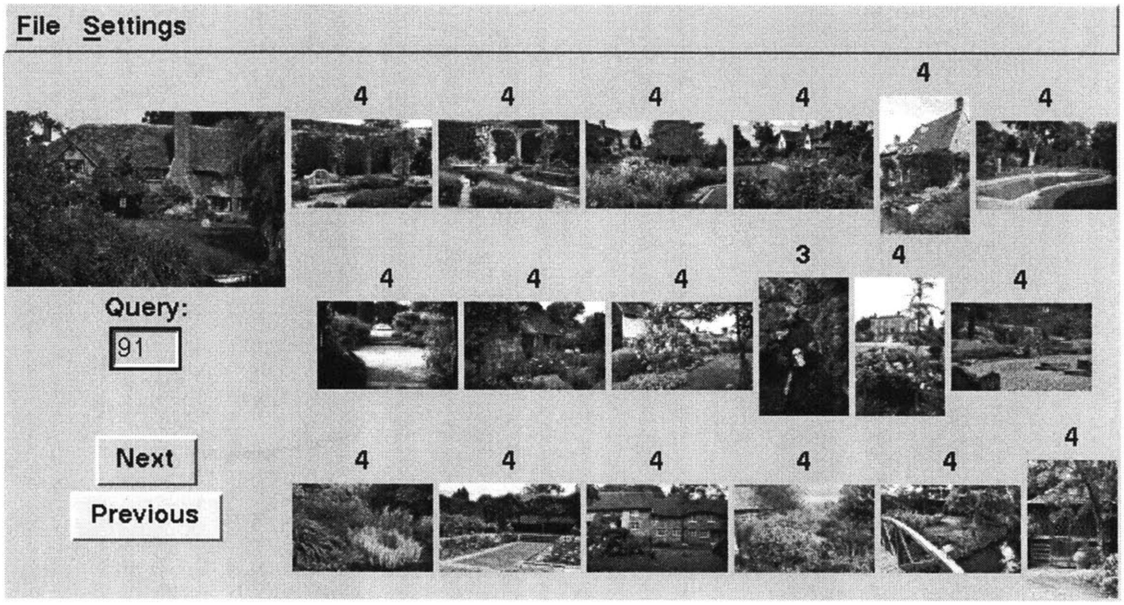


Figure 6.11: Outcome of queries on Corel for gardens.

Chapter 7

Short-term learning

There are various reasons to doubt that any retrieval system (no matter how sophisticated) will always be able to find the desired images in response to the first query from a user. This is due to 1) the difficulty of the image understanding problem and 2) the fact that users themselves are not always sure of what they want to retrieve before browsing through the database. In practice, it means that retrieval is always an interactive process, consisting of the following sequence of events: 1) user provides a query, 2) system retrieves best matches, 3) user selects a new query, 4) process is iterated.

The interactive nature of the retrieval problem can be both a blessing and a curse for retrieval systems. On one hand, feedback provided by the user can be exploited to guide the search. This is a major departure from traditional vision problems and makes it feasible to build effective systems without solving the complete artificial intelligence problem [132]. In this context, a retrieval system is nothing more than an interface between an intelligent high-level system (the user's brain) that can perform amazing feats in terms of visual interpretation but is limited in speed, and a low-level system (the computer) that has very limited visual abilities but can perform low-level operations very efficiently.

On the other hand, users tend to be frustrated if the system does not appear to know how to integrate their feedback. This means that the low-level retrieval system cannot be completely dumb and, at the very least, should be able to *integrate* the information provided by the user throughout the entire course of interaction. Otherwise, it will simply keep oscillating between the image classes that best satisfy the latest query and convergence to the right solution will be difficult.

Consequently, in addition to a powerful image representation, a good retrieval system must also incorporate inference mechanisms to facilitate the user-interaction. However, the two problems cannot be solved in isolation, as the careless selection of the representation will make inference more difficult and vice-versa. In the previous chapters, we have established that probabilistic retrieval is a powerful solution to the problem of evaluating image similarity. We now show that it is also the natural answer to the inference problem.

7.1 Prior work

The design of inference algorithms is particularly difficult for retrieval systems based on holistic image similarity because, in this case, two different tasks must be accomplished. First, the system must figure out what exactly is the set of visual image properties or concepts that the user is interested in. Finding a good match for these concepts is possible only after they are identified. As the example in Figure 6.1 demonstrates, the first step cannot be accomplished from the observation of a single image, and several iterations of the interaction between user and retrieval system must occur before the latter knows exactly what the former is looking for, assuming that this is ever clear. By avoiding this first learning step, systems relying on localized feedback need to concentrate only on the second problem, which has easier solution.

Given this observation, it is somewhat surprising to realize that, while various solutions have been presented to the inference problem (commonly referred to as *relevance feedback* in the retrieval literature) [12, 32, 110, 150, 175], most of them are intimately related with image representations that preclude local similarity. In fact, to the best of our knowledge, only the "Four eyes" system [110] combines learning with local queries, and even these are restricted to image patches of a sizeable dimension.

With respect to the inference mechanisms, both the "Four eyes" [110] and "PicHunter" [32] systems are Bayesian in spirit. "Four eyes" pre-segments all the images in the database, and groups all the resulting regions. Learning consists of finding the groupings that maximize the product of the number of examples provided by the user with a *prior grouping weight*. This can be seen as an approximation to Bayes rule. "PicHunter" defines a set of actions that a user may take and, given the images retrieved at a given point, tries to estimate the probabilities of the actions the user will take next. Upon observation of these actions, Bayes rule gives the probability of each image in the database being the desired target.

The main limitations of these two systems are due to the fact that the underlying image representations and similarity criteria are not conducive to learning per se. For example, because there is no easy way to define priors for region groupings, in [110] this is done through a greedy algorithm based on heuristics that are not always easy to justify or guaranteed to lead to an interesting solution. On the other hand, because user modeling is a difficult task, [32] relies on several simplifying assumptions and heuristics to estimate action probabilities. These estimates can only be obtained through an ad-hoc function of image similarity which is hard to believe valid for all or even most of the users the system will encounter. Indeed it is not even clear that such a function can be derived when the action set becomes more complicated than that supported by the simple interface of "PicHunter". For example, in the context of local queries, the action set would have to account for all possible segmentations of the query image, which are not even defined a priori.

All these problems are eliminated by the Bayesian formulation of the retrieval problem introduced in this thesis because it grounds all inferences directly on the image observations selected by the user. In this chapter, we show that, by combining a probabilistic criteria for image similarity with a generative model for image representation, there is no need for heuristic algorithms to learn priors or heuristic functions relating image similarity and the belief that a given image is the target. Under the new formulation, 1) the similarity function is, by definition, this belief and 2) prior learning follows naturally from belief propagation according to the laws of probability [127, 75, 85, 76]. Since all the necessary beliefs are an automatic outcome of the similarity evaluation and all previous interaction can be summarized by a small set of prior probabilities, this belief propagation is very simple, intuitive, and extremely efficient from the points of view of computation and storage.

7.2 Bayesian relevance feedback

Following Cox et al. [32], we identify two types of searches: *target search* and *open-ended browsing*. While in target search users seek to find an image from a specific image class, in open-ended browsing they only have a vague idea of what they are looking for. It is relatively easy to extend the Bayesian retrieval model so that it can account for both situations. Instead of a single query[1] $\mathbf{x}$, we consider a sequence of queries $\mathbf{x}_1^t = \{\mathbf{x}_i\}_{i=1}^t$ and, instead of a class indicator variable Y, we define a collection $Y_1^t = \{Y_i\}_{i=1}^t$, where t is the iteration number. The event $Y_t = i$ indicates that the i^{th} image class is the target for iteration t.

Applying Theorem 1 and denoting the sequences $\{\mathbf{X}_1, \ldots, \mathbf{X}_t\}$ and $\{i_1, \ldots, i_t\}$ by $\mathbf{X}_1^t$ and i_i^t, respectively, the decision function that minimizes the probability of error is

$$\begin{aligned} g^*(\mathbf{x}_1^t) &= \arg\max_{i_1^t} \log P_{Y_1^t|\mathbf{X}_1^t}(i_1^t|\mathbf{x}_1^t) \\ &= \arg\max_{i_1^t} \{\log P_{\mathbf{X}_1^t|Y_1^t}(\mathbf{x}_1^t|i_1^t) + \log P_{Y_1^t}(i_1^t)\}, \end{aligned} \tag{7.1}$$

where the maximum is taken over all the possible configurations of Y_1^t. This is a well known problem in many areas of engineering and statistics including dynamics systems [54], speech processing [140], statistical learning [97], information theory [52] and, more recently, machine vision [28, 122, 196], where Y is a variable that encodes the state of the world and $\mathbf{x}$ observations from a phenomena to be modeled.

Application of the chain rule of probability leads to

$$\begin{aligned} g^*(\mathbf{x}_1^t) = \; & \arg\max_{i_1^t} \{\log P_{\mathbf{X}_1|Y_1}(\mathbf{x}_1|i_1^t) + \log P_{Y_1}(i_1) \\ & + \sum_{k=2}^{t} [\log P_{\mathbf{X}_k|\mathbf{X}_1^{k-1},Y_1^t}(\mathbf{x}_k|\mathbf{x}_1^{k-1}, i_1^t) + \log P_{Y_k|Y_1^{k-1}}(i_k|i_1^{k-1})]\}. \end{aligned}$$

Since, in practice, it is difficult to estimate the conditional probabilities $P_{\mathbf{X}_k|\mathbf{X}_1^{k-1},Y_1^t}(\mathbf{x}_k|\mathbf{x}_1^{k-1}, i_1^t)$ and $P_{Y_k|Y_1^{k-1}}(i_k|i_1^{k-1})$ for large t (due to the combinatorial explosion of the number of possi-

[1] Notice that, as in previous chapters, each query $\mathbf{x}_i$ is a collection of N_i feature vectors that we now denote by $\mathbf{x}_i = \{\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,N_i}\}$.

bilities for the conditioning event), it is usually necessary to rely on simplifying assumptions. A common solution is to rely on the following conditional independence assumption for the observations.

Assumption 5 *Given the target image class for iteration k, the query for that iteration is independent of the queries and target image classes for all other iterations*

$$P_{\mathbf{X}_k|\mathbf{X}_1^{k-1},Y_1^t}(\mathbf{x}_k|\mathbf{x}_1^{k-1},i_1^t) = P_{\mathbf{X}_k|Y_k}(\mathbf{x}_k|i_k).$$

In the retrieval context, the assumption of conditional independence implies that the user provides the retrieval system with new information at each iteration. This is reasonable since users will tend to get frustrated if they feel that they have to repeat themselves, and will probably stop using the system when conditional independence does not hold.

Under Assumption 5, we obtain what is usually referred to as an *hidden Markov model* (HMM) [140] or a *Markov source* [52], and the probability of retrieval error is minimized when

$$g^*(\mathbf{x}_1^t) = \arg\max_{i_1^t}\{\log P_{Y_1}(i_1) + \sum_{k=1}^{t} \log P_{\mathbf{X}_k|Y_k}(\mathbf{x}_k|i_k) + \sum_{k=2}^{t} \log P_{Y_k|Y_1^{k-1}}(i_k|i_1^{k-1})\}.$$

This model is valid for both target search and open-ended browsing. When the transition probabilities $P_{Y_k|Y_1^{k-1}}(i_k|i_1^{k-1})$ are unconstrained, users are free to change their mind as frequently as they want and we have a model for browsing. If, on the other hand, switching between states is not allowed,

$$P_{Y_k|Y_1^{k-1}}(i_k|i_1^{k-1}) = P_{Y_k|Y_{k-1}}(i_k|i_{k-1}) = \delta_{i_{k-1},i_k},$$

where δ_{i_{k-1},i_k} is the Kronecker delta function defined by (2.3), we have a model for target search; i.e., the user decides on a target image class at the start of the interaction according to $P_{Y_1}(i_1)$ and holds on to that target class until it is found, or the search aborted. In this case, the retrieval model can be simplified into

$$g^*(\mathbf{x}_1^t) = \arg\max_i\{\log P_Y(i) + \sum_{k=1}^{t} \log P_{\mathbf{X}_k|Y}(\mathbf{x}_k|i)\}, \tag{7.2}$$

where $Y = Y_1$. In practice, estimating transition probabilities involves implementing an actual retrieval system, assembling a body of users and collecting extensive statistics on the patterns of interaction. This a "chicken and egg" problem since without the transition probabilities it is not possible to implement a system that supports browsing. For this reason, we restrict ourselves to the problem of target search, leaving the more general question of open-ended browsing open for subsequent discussion.

7.3 Target search

Using Assumption 5 the chain rule of probability and Bayes rule, (7.2) can also be written as

$$\begin{aligned} g^*(\mathbf{x}_1^t) &= \arg\max_i\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \sum_{k=1}^{t-1} \log P_{\mathbf{X}_k|Y}(\mathbf{x}_k|i) + \log P_Y(i)\} \\ &= \arg\max_i\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \sum_{k=1}^{t-1} \log P_{\mathbf{X}_k|\mathbf{X}_1^{k-1},Y}(\mathbf{x}_k|\mathbf{x}_1^{k-1}, i) + \log P_Y(i)\} \\ &= \arg\max_i\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \log P_{\mathbf{X}_1^{t-1}|Y}(\mathbf{x}_1^{t-1}|i) + \log P_Y(i)\} \\ &= \arg\max_i\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \log P_{Y|\mathbf{X}_1^{t-1}}(i|\mathbf{x}_1^{t-1})\}. \end{aligned} \tag{7.3}$$

By comparing (7.3) with (2.14), the term $P_{Y|\mathbf{X}_1^{t-1}}(i|\mathbf{x}_1^{t-1})$ can be seen simply as a prior belief on the ability of the i^{th} image class to explain the query. However, unlike the straightforward application of the Bayesian criteria, this is not a static prior determined by some arbitrarily selected prior density. Instead, it is learned from the previous interaction between user and retrieval system and summarizes all the information in this interaction that is relevant for the decisions to be made in the future.

Recalling from (7.1) that, in target search mode,

$$g^*(\mathbf{x}_1^t) = \arg\max_i\{\log P_{Y|\mathbf{X}_1^t}(i|\mathbf{x}_1^t)\} \tag{7.4}$$

and comparing with (7.3) reveals an intuitive mechanism to integrate information over time. Together, (7.3) and (7.4) state that the system's beliefs on the user's interests at time $t-1$ simply become the prior beliefs for iteration t. New data provided by the user at time t are then used to update these beliefs, generating the posteriors on which the retrieval decisions are based. These posteriors in turn become the priors for iteration $t+1$. In other words, prior beliefs are continuously updated from the observation of the interaction between user and retrieval system. This is illustrated in Figure 7.1.

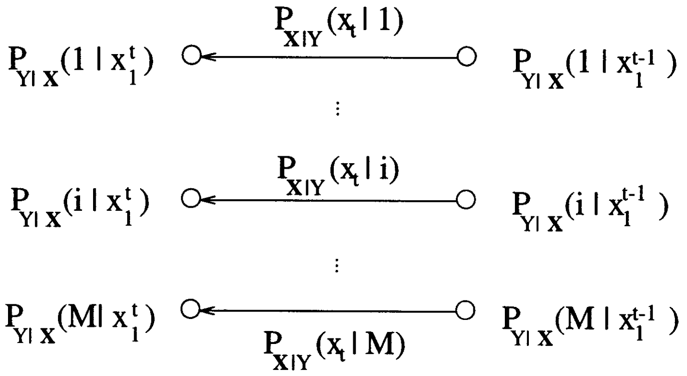


Figure 7.1: Belief propagation across iterations of the retrieval process.

From a computational standpoint, the procedure is very efficient since the bulk of the computation at each time step is due to the evaluation of the log-likelihood of the data in the corresponding query. Notice that this is exactly equation (2.16) and would have to be computed even in the absence of any learning. From the storage point of view, the efficiency is even higher since the entire interaction history is reduced to a number per image class. It is a remarkable fact that this number alone enables decisions that are optimal with respect to the entire interaction.

There is however one limitation associated with this belief propagation which is evident from (7.2): for large t, the contribution of the new data provided by the user is very small and the posterior probabilities tend to remain constant. This limitation can be avoided by replacing (7.3) with the more generic maximization problem [11],

$$g^*(\mathbf{x}) = \arg\max_i\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \lambda \log P_{Y|\mathbf{X}_1^{t-1}}(i|\mathbf{x}_1^{t-1})\},$$

where one looks for the image class that best explains the current query, under a constraint on how well it explains all the previous interaction. The scalar λ is a Lagrange multiplier that weighs the importance of the past. Defining $\alpha = 1/(1+\lambda) \in [0,1]$ this is equivalent to

$$g^*(\mathbf{x}) = \arg\max_i \{\alpha \log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + (1-\alpha) \log P_{Y|\mathbf{X}_1^{t-1}}(i|\mathbf{x}_1^{t-1})\}, \tag{7.5}$$

in which case the past is all that matters when $\alpha = 0$, while all the emphasis is on the current query when $\alpha = 1$, and we recover (7.3) when $\alpha = 0.5$. Rewriting this equation as

$$\begin{aligned} g^*(\mathbf{x}) &= \arg\max_i \{\alpha \log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \alpha(1-\alpha) \log P_{\mathbf{X}_{t-1}|Y}(\mathbf{x}_{t-1}|i) \\ &\quad +(1-\alpha)^2 \log P_{Y|\mathbf{X}_1^{t-2}}(i|\mathbf{x}_1^{t-2})\} \\ &= \arg\max_i \{\sum_{k=1}^{t} \alpha(1-\alpha)^{t-k} \log P_{\mathbf{X}_k|Y}(\mathbf{x}_k|i) + (1-\alpha)^t \log P_Y(i)\} \end{aligned}$$

$\alpha(1-\alpha)^{t-k}$ can be seen as a *decay factor* that penalizes older terms.

7.4 Negative feedback

While positive feedback is a powerful addition to retrieval systems, there are many situations in CBIR where it is not sufficient to guarantee satisfactory performance. In such situations, it is usually possible to attain significant improvements by combining it with negative feedback. One example is when various image classes in the database have overlapping densities. This is illustrated in Figure 7.2, where we depict an hypothetical search on a database with two major image classes that share a common attribute (large regions of blue sky), but are different in other aspects (images in class A also contain regions of white snow, while those in class B contain regions of grass). This could, for example, be a database of recreation sites where class A contains pictures of a ski resort, while class B contains pictures of the same resort but taken during the summer.

If the user starts with an image of class A (e.g. a picture of a snowy mountain), using regions of sky as positive examples is not likely to quickly lead to the images of class B. In fact, all other factors being equal, there is an equal likelihood that the retrieval system will return images from the two classes. This is illustrated in the top row. On the other

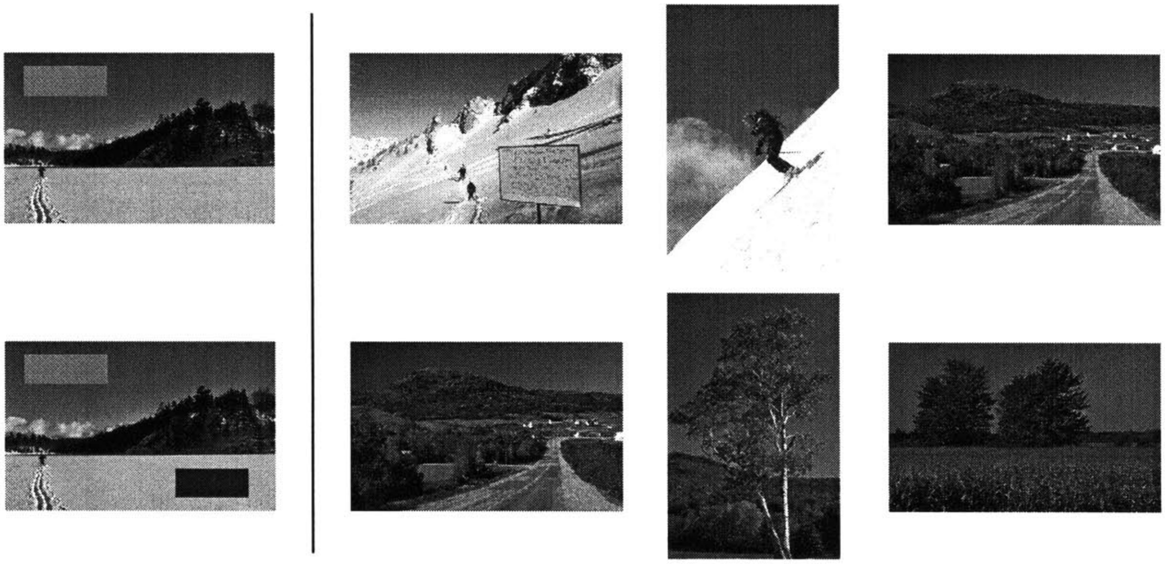


Figure 7.2: Two queries based on the same query image (shown on the left). The regions blocked by the light green (dark red) rectangle are positive (negative) examples for the search. Top: query for sky. Bottom: query for sky but not snow.

hand, if the user can explicitly indicate interest in regions of sky but not in regions of snow, the likelihood that only images from class B will be returned increases drastically. This is illustrated in the bottom row.

Another example of the importance of negative feedback are situations in which there does not appear to be a good positive example to select next. These happen when, in response to user feedback, the system returns a collection of images that have already been retrieved in previous iterations. Assuming the user has already given the system all the possible positive feedback, the only way to escape from such situations is to choose some regions that are not desirable and use them as negative feedback. In the example above, when users get stuck with a screen full of pictures of white mountains, they can simply select some regions of snow to escape the local minima. On the other hand, if only positive examples were allowed, what to do next would not be clear. This is illustrated in Figure 7.3.

In order to account for negative examples, we must penalize the classes under which these score well while favoring the classes that assign a high score to the positive examples.

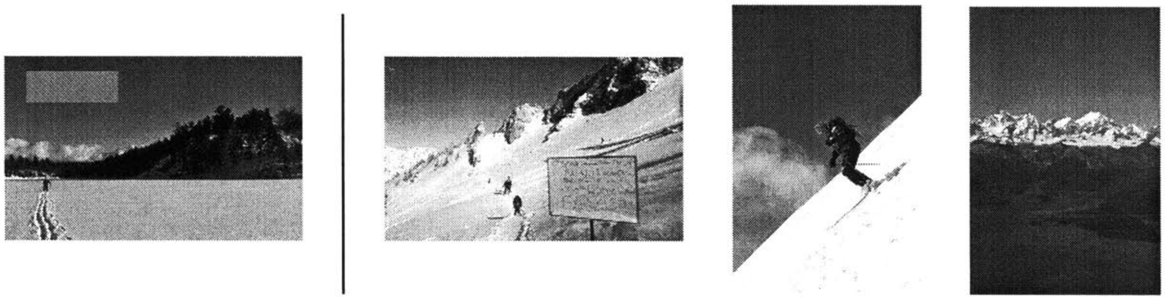


Figure 7.3: The user is looking for pictures taken in the summer and has already provided the retrieval system with various examples of sky. What to do next is not clear, unless negative examples are allowed.

Unlike positive examples, for which the likelihood is known, it is not straightforward to estimate the likelihood of a particular negative example given that the user is searching for a certain image class. We denote the use of the vector $\mathbf{z}$ as a negative example by $\bar{\mathbf{z}}$ and rely on the following assumption.

Assumption 6 *The likelihood that* $\mathbf{z}$ *will be used as a negative example, given that the target is class* i*, is equal to the likelihood it will be used as a positive example given that the target is any other class*

$$P_{\mathbf{Z}|Y}(\bar{\mathbf{z}}|Y = i) = P_{\mathbf{Z}|Y}(\mathbf{z}|Y \neq i). \tag{7.6}$$

This assumption captures the intuition that, when searching for class i, a good negative example is one that would be a good positive example if the user were looking for any class other than i. For example, if class i is the only one that does not contain regions of sky, using pieces of sky as negative examples will quickly eliminate the other images in the database. Thus, one would expect the user to provide sky as a negative example with high probability.

Denoting by $\bar{\mathbf{z}}_1^t = \{\bar{\mathbf{z}}_i\}_{i=1}^t$ the collection of negative queries, these can be accounted for by simply replacing (7.3) and (7.4) with

$$\begin{aligned}
g^*(\mathbf{x}_1^t) &= \arg\max_i\{\log P_{Y|\mathbf{X},\mathbf{Z}}(i|\mathbf{x}_1^t, \bar{\mathbf{z}}_1^t)\} \\
&= \arg\max_i\{\log P_{\mathbf{X}_t,\mathbf{Z}_t|Y}(\mathbf{x}_t, \bar{\mathbf{z}}_t|i) + \log P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})\} \\
&= \arg\max_i\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \log P_{\mathbf{Z}_t|Y}(\bar{\mathbf{z}}_t|i) + \log P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})\} \\
&= \arg\max_i\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \log P_{\mathbf{Z}_t|Y}(\mathbf{z}_t|Y \neq i) + \log P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})\} \\
&= \arg\max_i\left\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \log \frac{P_{Y|\mathbf{Z}_t}(Y \neq i|\mathbf{z}_t)}{P_Y(Y \neq i)} + \log P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})\right\} \\
&= \arg\max_i\left\{\log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \log \frac{1 - P_{Y|\mathbf{Z}_t}(i|\mathbf{z}_t)}{1 - P_Y(i)} + \log P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})\right\}
\end{aligned}$$

where we have also used Assumptions 5 and 6 and the fact that $P_{\mathbf{Z}_t}(\mathbf{z}_t)$ does not depend on i. Applying Bayes rule recursively then leads to the following natural generalization of (7.2)

$$g^*(\mathbf{x}_1^t) = \arg\max_i\left\{\log P_Y(i) + \sum_{k=1}^{t} \log P_{\mathbf{X}_k|Y}(\mathbf{x}_k|i) + \log \frac{1 - P_{Y|\mathbf{Z}_k}(i|\mathbf{z}_k)}{1 - P_Y(i)}\right\}.$$

In practice, however, this equation is not very useful since the terms $1 - P_{Y|\mathbf{Z}_k}(i|\mathbf{z}_k)$ and $1 - P_Y(i)$ tend to be close to one. To see this, suppose that we are dealing with a database of 10,000 image classes which are assumed to be equally likely a priori: $P_Y(i) = 1/10,000$. In this case, even if the observation of $\mathbf{z}_k$ increases, the probability of class i one-hundred-fold $P_{Y|\mathbf{Z}_k}(i|\mathbf{z}_k) = 1/100$, the ratio $(1 - P_{Y|\mathbf{Z}_k}(i|\mathbf{z}_k))/(1 - P_Y(i))$ is only 0.99. This means that negative examples have very small influence in the overall decision function.

An alternative solution is to choose the class i that maximizes the *posterior odds ratio* [55] between the hypotheses "class i is the target" and "class i is not the target"

$$\begin{aligned}
g^*(\mathbf{x}_1^t) &= \arg\max_i \log \frac{P_{Y|\mathbf{X}_1^t,\mathbf{Z}_1^t}(i|\mathbf{x}_1^t, \bar{\mathbf{z}}_1^t)}{P_{Y|\mathbf{X}_1^t,\mathbf{Z}_1^t}(Y \neq i|\mathbf{x}_1^t, \bar{\mathbf{z}}_1^t)} \\
&= \arg\max_i \log\left(\frac{P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i)}{P_{\mathbf{X}_t|Y}(\mathbf{x}_t|Y \neq i)} \frac{P_{\mathbf{Z}_t|Y}(\bar{\mathbf{z}}_t|i)}{P_{\mathbf{Z}_t|Y}(\bar{\mathbf{z}}_t|Y \neq i)} \frac{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})}{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}} P(Y \neq i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})}\right) \\
&= \arg\max_i \log\left(\frac{P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i)}{P_{\mathbf{X}_t|Y}(\mathbf{x}_t|Y \neq i)} \frac{P_{\mathbf{Z}_t|Y}(\mathbf{z}_t|Y \neq i)}{P_{\mathbf{Z}_t|Y}(\mathbf{z}_t|i)} \frac{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})}{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(Y \neq i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})}\right) \\
&= \arg\max_i \log\left(\frac{P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i)}{P_{\mathbf{Z}_t|Y}(\mathbf{z}_t|i)} \frac{P_{Y|\mathbf{Z}_t}(Y \neq i|\mathbf{z}_t)}{P_{Y|\mathbf{X}_t}(Y \neq i|\mathbf{x}_t)} \frac{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})}{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(Y \neq i|\mathbf{x}_1^{t-1}, \bar{\mathbf{z}}_1^{t-1})}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \arg\max_i \log \left(\frac{P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i)}{P_{\mathbf{Z}_t|Y}(\mathbf{z}_t|i)} \left[\frac{1 - P_{Y|\mathbf{Z}_t}(i|\mathbf{z}_t)}{1 - P_{Y|\mathbf{X}_t}(i|\mathbf{x}_t)} \right] \frac{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1},\bar{\mathbf{z}}_1^{t-1})}{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(Y \neq i|\mathbf{x}_1^{t-1},\bar{\mathbf{z}}_1^{t-1})} \right) \\
&\approx \arg\max_i \left\{ \log \frac{P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i)}{P_{\mathbf{Z}_t|Y}(\mathbf{z}_t|i)} + \log \frac{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1},\bar{\mathbf{z}}_1^{t-1})}{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(Y \neq i|\mathbf{x}_1^{t-1},\bar{\mathbf{z}}_1^{t-1})} \right\}
\end{aligned}
$$

where we have used the fact that $(1 - P_{Y|\mathbf{Z}_t}(i|\mathbf{z}_t))/(1 - P_{Y|\mathbf{X}_t}(i|\mathbf{x}_t)) \approx 1$. Including a decay factor to penalize ancient terms, we obtain

$$
g^*(\mathbf{x}_1^t) \approx \arg\max_i \left\{ \alpha \log \frac{P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i)}{P_{\mathbf{Z}_t|Y}(\mathbf{z}_t|i)} + (1-\alpha) \log \frac{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(i|\mathbf{x}_1^{t-1},\bar{\mathbf{z}}_1^{t-1})}{P_{Y|\mathbf{X}_1^{t-1},\mathbf{Z}_1^{t-1}}(Y \neq i|\mathbf{x}_1^{t-1},\bar{\mathbf{z}}_1^{t-1})} \right\}. \quad (7.7)
$$

This equation is similar to (7.5) but now the terms on the denominator penalize the image classes that explain well the negative examples. Overall, the decision function favors image classes that explain well the positive examples and poorly the negative ones.

There is however, under the posterior odds ratio, a tendency to over-emphasize the importance of negative examples. In particular, any class with zero probability of generating the negative examples will lead to an infinite ratio, even if it explains very poorly the positive examples. To avoid this problem, we proceed in two steps:

- start by solving (7.5), i.e. sort the classes according to how well they explain the positive examples.
- select the subset of the best N classes and solve (7.7) considering only the classes in this subset;

Overall, the inference algorithm has two free parameters: the decay factor α, and the size N of the window used in the second step. In the next section, we present experimental evidence that the learning performance is quite robust to variations in these parameters.

7.5 Experimental evaluation

We performed several experiments to evaluate the improvements in retrieval performance achievable with Bayesian inference. Because in an ordinary browsing scenario it is difficult to know the ground truth for the retrieval operation (at least without going through the tedious process of hand-labeling all images in the database), we relied on the mosaic databases (for which ground truth is available).

7.5.1 Experimental setup

The goal of the experiments was to determine if it is possible to reach a desired target image by starting from a weakly related one and providing feedback to the retrieval system. This simulates the interaction between a real user and the CBIR system and is an iterative process, where each iteration consists of 1) selecting a few examples, 2) using them as queries for retrieval, and 3) examining the top V retrieved images to find examples for the next iteration. V should be small since most users are not willing to go through lots of false positives to find the next query.

The most challenging problem in automated testing is to determine a good strategy for selecting the examples to be given to the system. The closer this strategy is to what a real user would do, the higher the practical significance of the results. However, even when there is clear ground truth for the retrieval (as is the case of the mosaic databases), it is not completely clear how to make the selection. While it is obvious that regions of texture or object classes that appear in the target should be used as positive feedback, it is much harder to determine automatically what are good negative examples. As shown in Figure 7.4, there are cases in which images from two different classes are visually similar. Selecting images from one of these classes as a negative example for the other will be a disservice to the learner.

While real users tend not to do this, it is hard to avoid such mistakes in an automated setting, unless one does some sort of pre-classification of the database. Because we wanted to avoid such pre-classification, we decided to stick with a simple selection procedure and live with these mistakes. At each step of the iteration, examples were selected in the following way: among the top V images returned by the retrieval system, the one with most sub-images from image classes also present in the target was selected to be the next query. One block from each sub-image in the query was then used as a positive (negative) example if the texture or object depicted in that sub-image was also (was not) represented in the

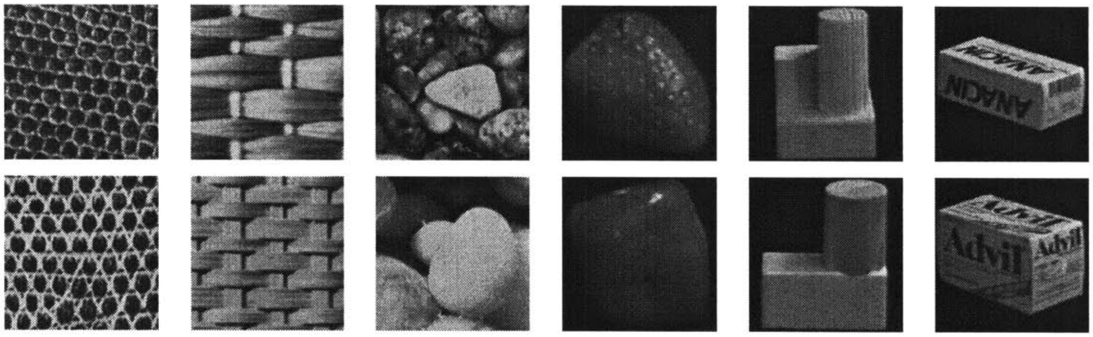


Figure 7.4: Examples of pairs of visually similar images that appear in different image classes.

target image.

This strategy is a worst-case scenario. First, the learner might be confused by conflicting negative examples. Second, as seen in Chapter 6, better retrieval performance can be achieved if more than one block from each region is included in the queries. However, using only one block reduces the computational complexity of each iteration, allowing us to 1) average results over several runs of the inference process and 2) experiment several values for the α and V parameters of the retrieval system. We selected several (α, V) pairs and performed 100 runs with random target images for each. In all cases, the initial query image was the first in the database containing one sub-image in common with the target.

The performance of the learning algorithm can be evaluated in various ways. We considered two metrics: the percentage of the runs that converged to the right target and the number of iterations required for convergence. Because, to prevent the learner from entering loops, any given image could only be used once as a query, the algorithm can diverge in two ways. Strong divergence occurs when, at a given time step, the images (among the top V) that can be used as queries do not contain any sub-image in common with the target. In such situation, a real user will tend to feel that the retrieval system is incoherent and abort the search. Weak divergence occurs when all the top V images have previously been used. This is a less troublesome situation because the user could simply look up more images (e.g. the next V) to get new examples. To make the presentation compact, in the next sections we only show results relative to strong divergence.

7.5.2 Positive feedback

Figure 7.5 presents plots of the convergence rate, mean number of iterations until convergence, and number of iterations until divergence as a function of the decay factor α and the number of matches V, for the two mosaic databases. In both cases, the inclusion of learning ($\alpha < 1$) typically increases the convergence rate. This increase can be very significant (as high as 15%), and larger gains occur when the convergence rate without learning is low. If the convergence rate is already high without learning, the inclusion of learning does not change it significantly. In general, a precise selection of α is not crucial for achieving a rate of convergence close to the best possible.

The rate of convergence is also affected by the number of matches V from which the user is allowed to select the next query. While, as expected, the larger this number the faster the convergence, a law of diminishing returns seems to be in effect: while the convergence rate increases quickly when V is small, it levels off for large values of V. This is an interesting result because there is usually a cost associated with a large V: users are not willing to go through lots of image screens in order to find a suitable next query. Another interesting result is that, under the assumption that users will only look at the first image screen returned by the retrieval system ($V \in [15, 20]$), the inclusion of learning leads to visible convergence improvements.

In terms of the number of iterations, when convergence occurs it is usually very fast (from 4 to 8 iterations). On the other hand, the number of iterations until divergence is usually well above 8. This indicates that, in practice, it would be easy to detect when the retrieval system is not likely to converge: if the number of iterations is above 8 to 10, then it is probably preferable to start a new query (from another initial image) than to insist on the current one.

We next present examples of the relevance feedback process in the Columbia mosaic database. Figure 7.6 depicts a search for a target image consisting of a plastic bottle, a white hanger, a blue plastic donut, and a coffee cup. The top picture depicts the first iteration of the retrieval process. The search starts with an image containing the blue donut and, since the retrieval precision is high for this object, it appears in all the 15 retrieved

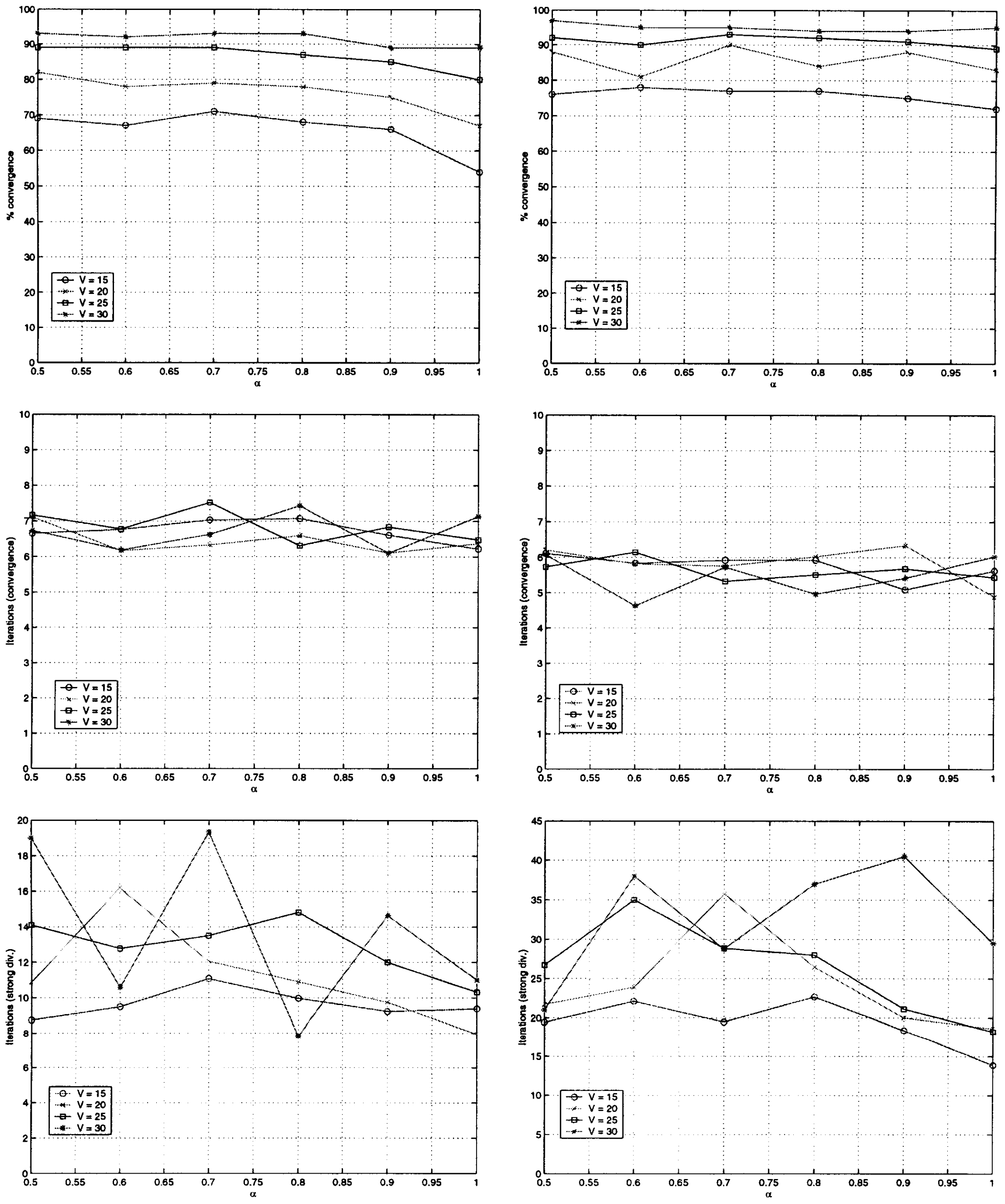
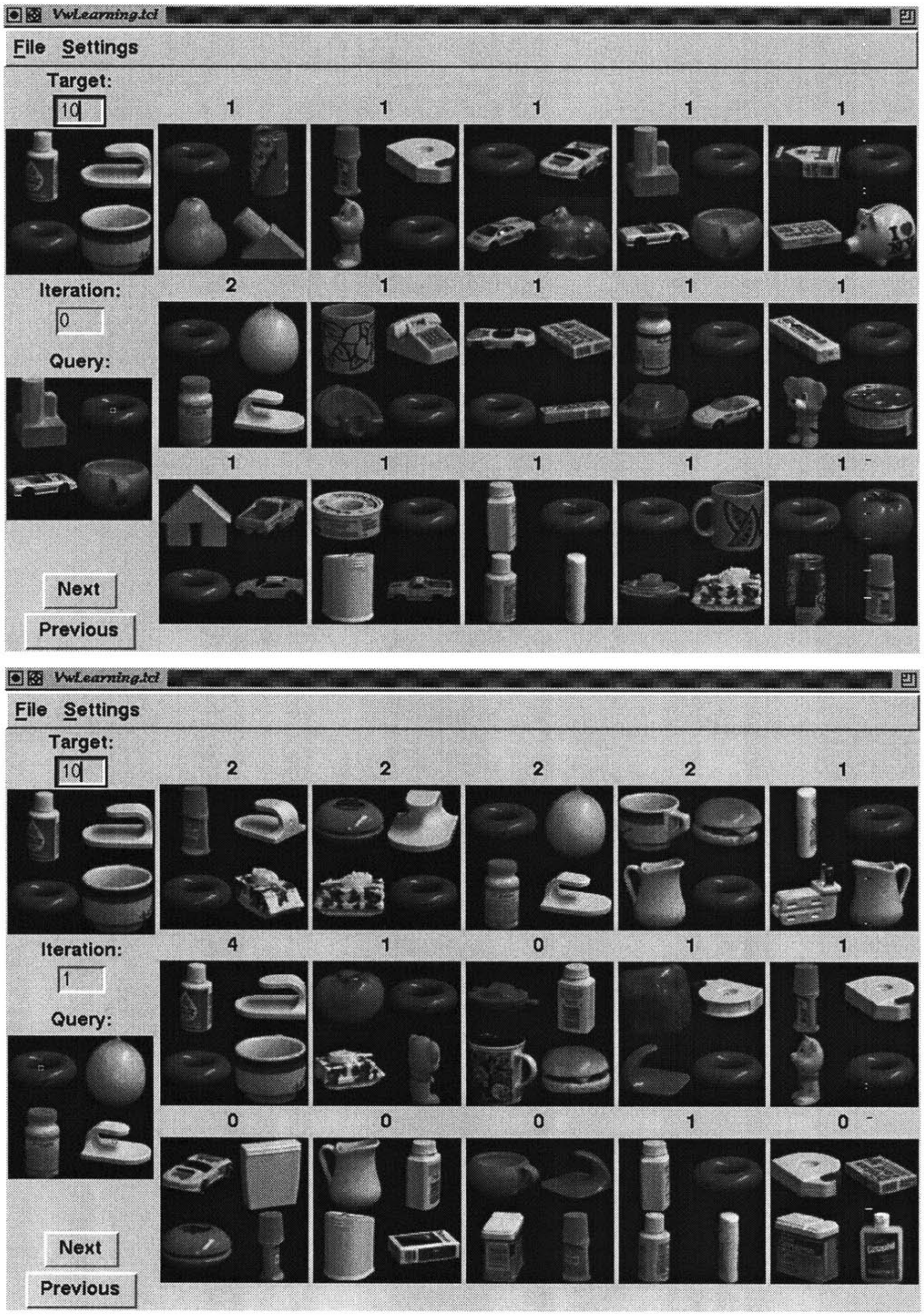


Figure 7.5: Plots, as a function of the learning factor α, of the convergence rate (top) average number of iterations for convergence (middle) and divergence (bottom). In all plots different curves correspond to different values for the number of images V examined at the end of each iteration. Left: Brodatz mosaic database. Right: Columbia mosaic database.



VwLearning.tcl

File Settings

Target: 10

Iteration: 0

Query:

VwLearning.tcl

File Settings

Target: 10

Iteration: 1

Query:

Figure 7.6: Two iterations of positive feedback. In both cases, the target image is shown at the top left and the query image immediately below. The query itself is based on a single feature vector from each of the sub-images (8 × 8 neighborhood indicated in the center of the sub-image) that are shared by the query and the target images. The number above each retrieved image indicates the number of objects that it shares with the target.

image slots. At this point the retrieval system has basically restricted the set of possible matches to the K images that contain the blue donut. Hence, for each of the 15 slots, the probability of the target image appearing in the slot is approximately $1/K$. Since $K << D$, where D is the database size, this is dramatically higher than the $1/D$ associated with random guessing. Furthermore, if there are M images containing both the blue donut and one of the three other objects of interest, the corresponding probability is M/K. This can be high and it is therefore not surprising that, due to chance alone, one of the other objects in the target will also appear among the retrieved images. This is indeed the case of the first image in the second row, which also contains the white hanger.

By the selecting this image as next query (bottom picture) the user increases the probability per slot from approximately $1/K$ to approximately $1/L$ where L is the total number of images containing the two query objects. Notice that, as more objects are included in the query, the total number of images containing those objects decreases substantially and the above probabilities increase drastically. In the example of the figure, because there are less than 15 images in the entire database containing both objects, one would expect the target to appear among the top 15 matches with very high probability. This is indeed the case, and the target shows up as the first image in the second row.

The example illustrates how, provided that precision is high for the individual query objects, retrieval can be very fast. In this sense, it corresponds to a best-case scenario since the system will typically have to deal with objects for which precision is not so high. It is in these situations that learning becomes most important.

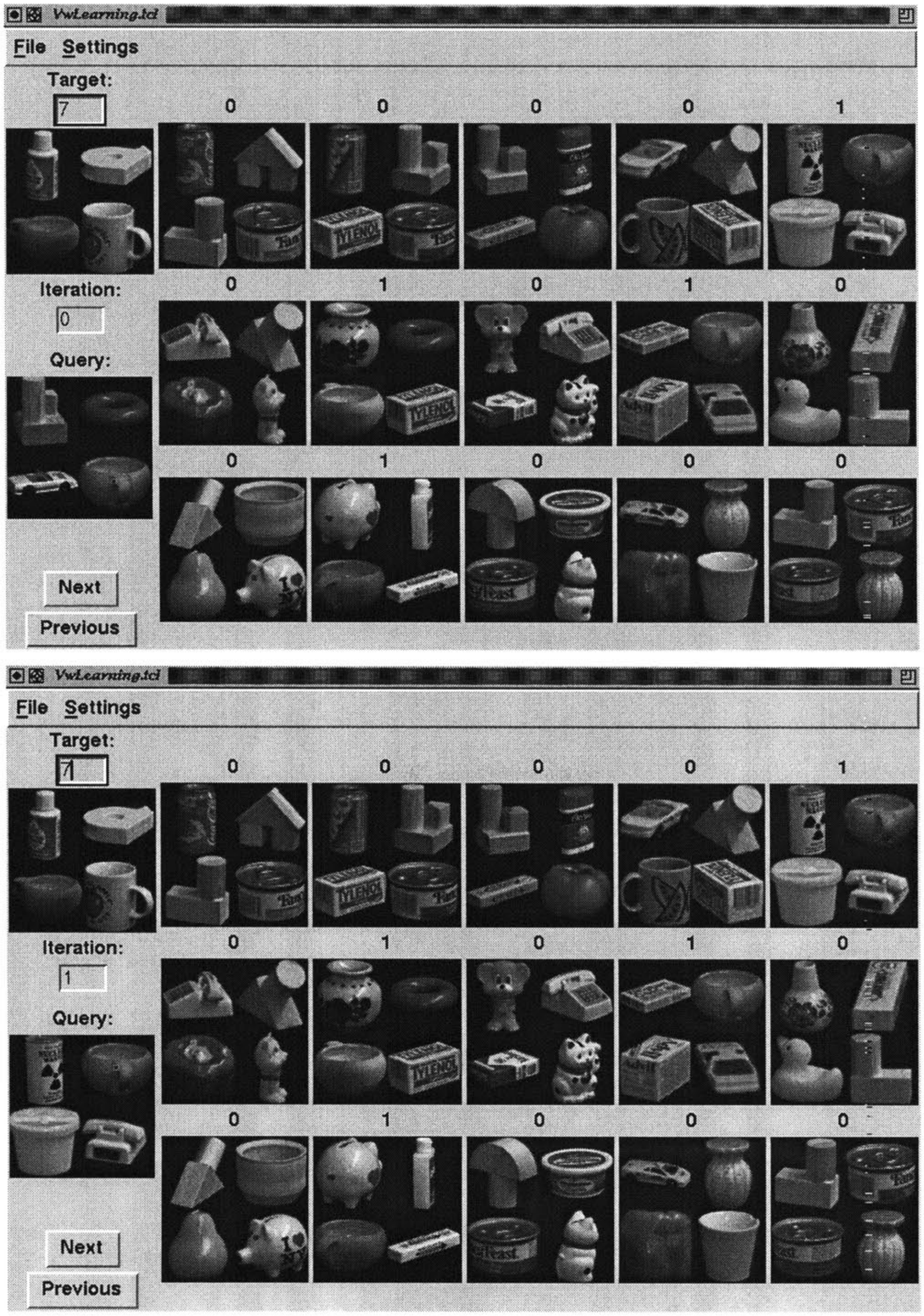
Figures 7.7 and 7.8 show one such example. The target consists of a plastic bottle, a container of adhesive tape, a clay cup, and a white mug, while the initial query is an image containing the clay cup. Since there are various objects made of wood in the Columbia database and these have surface properties visually similar to those of the clay cup, precision is now significantly smaller (top picture of Figure 7.7): only 4 of the 15 top matches are correct. This makes it difficult to zero in on the target for two fundamental reasons: first, it is not as likely as in the previous example that the other objects in the target will appear among the top matches. Second, when this happens, it is likely that the new target objects will not share an image with the object used in the query. Both of these points are illustrated by the example. First, the feedback process must be carried for three iterations before a target object other than that in the query appears among the top matches. Second, when this happens (top picture of Figure 7.8), the new object (tape container) is not part of an image that also contains the clay cup.

In this situation, the most sensible option is to base the new query on the newly found target object (tape container). However, in the absence of learning, it is unlikely that the resulting matches will contain any instances of the query object used on the previous iterations (clay cup) or the objects that are confounded with it. As illustrated by the bottom picture of Figure 7.8, the role of learning is to favor images containing these objects. In the example of the figure, 7 of the 15 images returned in response to a query based on the tape container include the clay cup or visually similar objects (in addition to the tape container itself). This enables new queries based on both target objects which, as seen in the previous example, have an increased chance of success. In this particular case, it turns out that one of the returned images is the target itself.

7.5.3 Negative feedback

Figure 7.9 presents plots of the convergence rate, mean number of iterations until convergence, and strong divergence rate for the two mosaic databases when both positive and negative feedback are used and the number, N, of top positive feedback matches considered in (7.7) is 50. Comparing with the plots of Figure 7.5, it is clear that the convergence rate is significantly improved by the inclusion of negative feedback. In particular, for most values of V the convergence rate is close to 100% and, in all cases, the rate of strong divergence is zero.

Since the convergence rate is high, learning is usually less relevant than it was in the experiments where only positive feedback is allowed. Notice, however, that the best convergence happens for values of α larger than 0.5. In fact, some degradation is noticeable on Columbia as we approach the 0.5 limit. This degradation is related to the fact that the number of iterations required for convergence is now larger than for positive feedback-only retrieval. This increase in the number of iterations is more significant for the harder retrieval scenarios (smaller V) and particularly noticeable for $V = 15$, where it reaches 100%



Target: 7

Iteration: 0

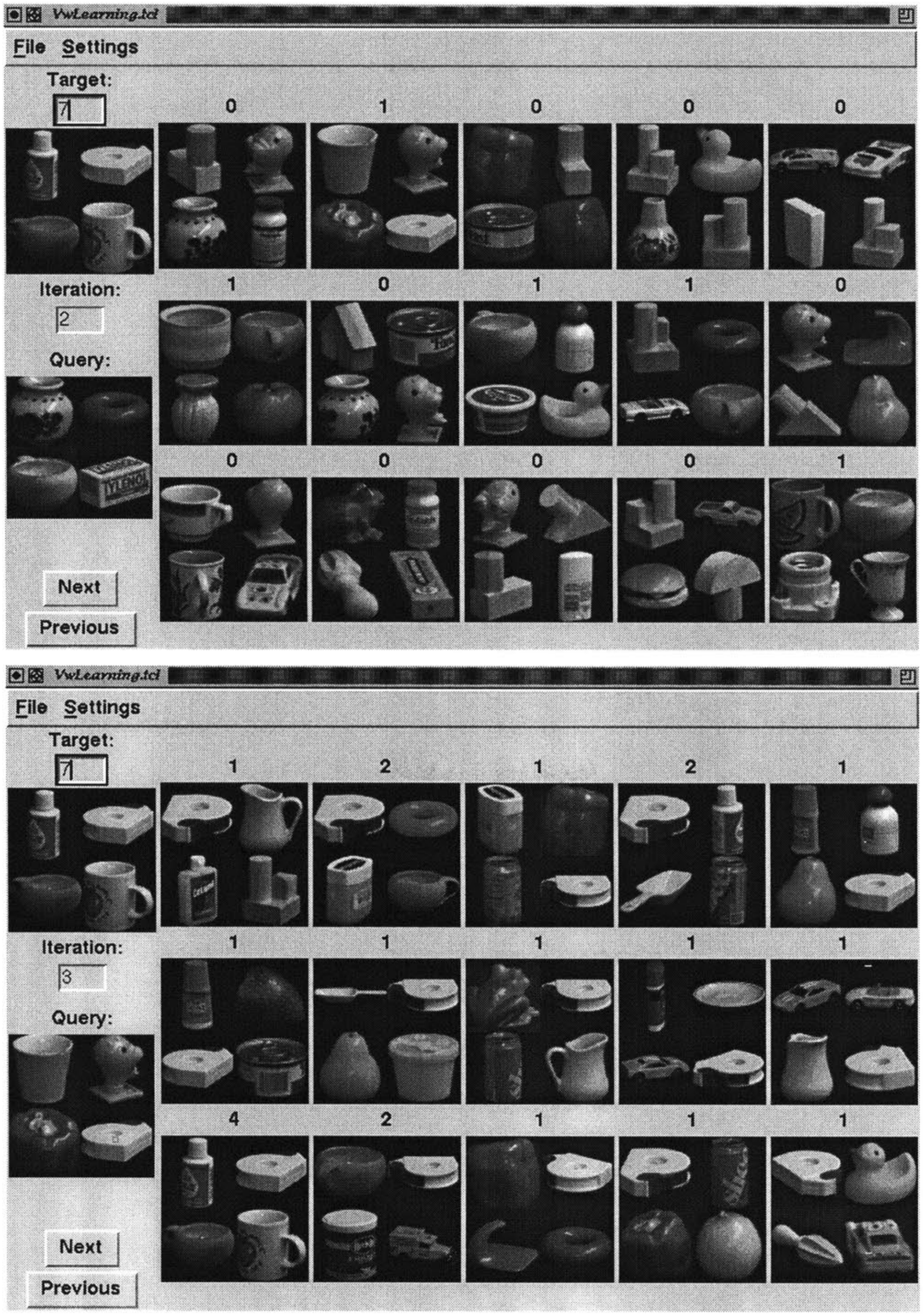
Query:

Target: 7

Iteration: 1

Query:

Figure 7.7: First two iterations of positive feedback for the example discussed in the text.



Target: 7

Iteration: 2

Query:

Target: 7

Iteration: 3

Query:

Figure 7.8: Last two iterations of positive feedback for the example discussed in the text.

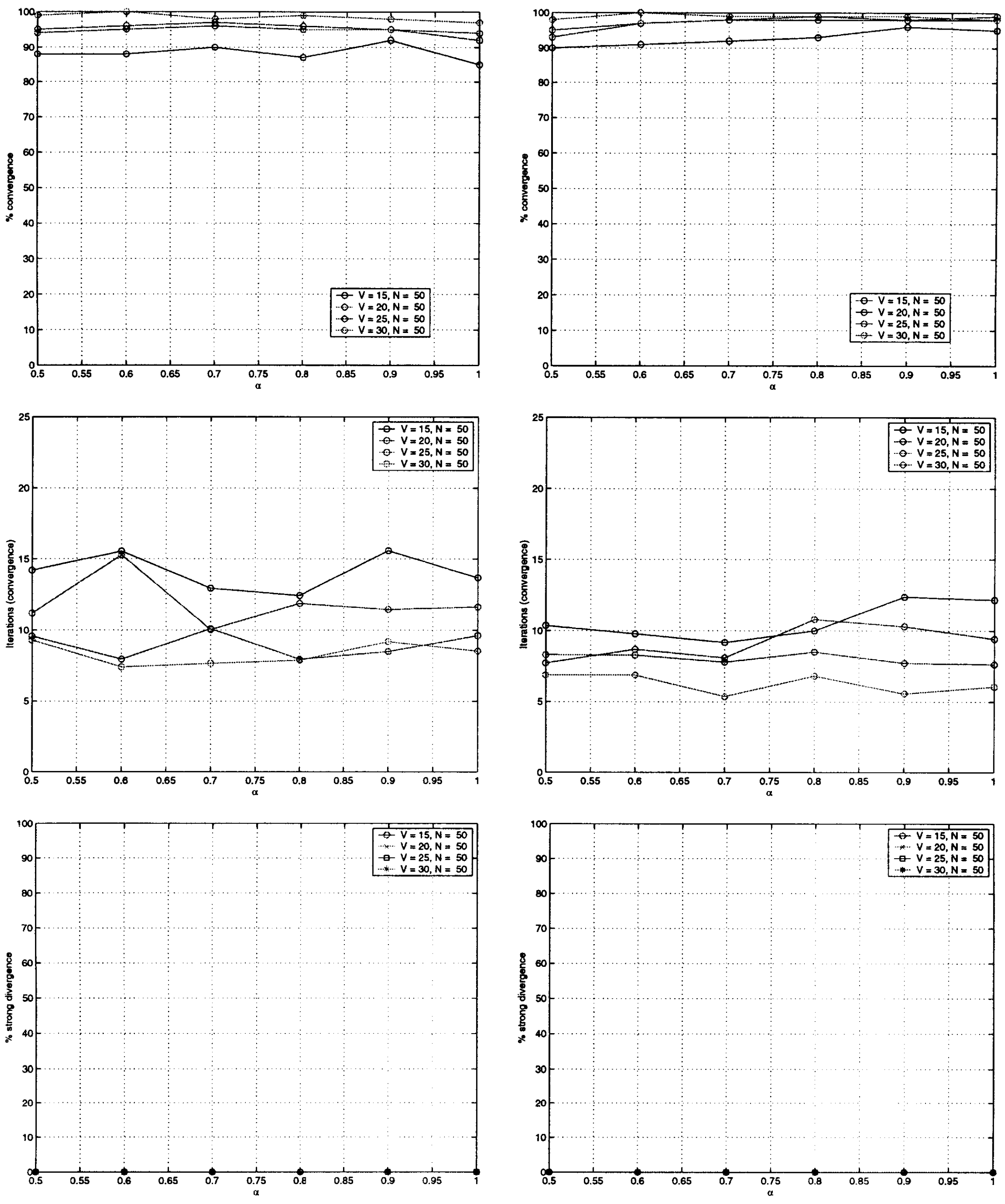


Figure 7.9: Plots, as a function of the learning factor α, of the convergence rate (top), average number of iterations for convergence (middle), and divergence rate (bottom). In all plots, different curves correspond to different values of the number of images V examined at the end of each iteration and N is number of top positive feedback matches that are considered when negative feedback is taken into account. Left: Brodatz mosaic database. Right: Columbia mosaic database.

on Brodatz and and 50% on Columbia.

The combination of all these observations allows the following conclusions. First, introducing negative feedback allows *exploration* of the database and this, in turn, leads to better convergence. In particular, retrieval is never stopped because the user runs out of examples to select next. Second, an increase in the number of iterations required for convergence is inherent to this exploration. Notice that, while in Figure 7.5 this number was approximately the same for all V, we now have significant differences. In particular, convergence takes longer for smaller V, i.e. when retrieval is most difficult. This is a sensible result, and suggests that the average number of iterations increases because retrieval takes much longer on a small set of difficult cases. Our personal experience of interaction with the retrieval system confirms this hypothesis.

The plots also expose a trade-off between the number of images that the user inspects to find the next query (V) and the number of iterations for convergence. Notice that, by increasing V from 15 (approximately one screen of images) to 30 (two screens), it is possible to increase the speed of convergence to the levels of Figure 7.5. It remains to be studied what would be more appealing to users: less iterations or a smaller number of images to inspect per iteration.

Figure 7.10 shows the impact of the parameter N in the retrieval performance. The figure depicts the convergence rate and speed for the Columbia mosaic database when $N = 100$ and $N = 150$. Notice that, while performance is not as good as when $N = 50$, it is still clearly superior to that achievable with positive feedback alone. This confirms that a very precise selection of N is not required to guarantee the improvements inherent to negative feedback. More drastic differences happen for the convergence speed, which can vary substantially with N. Here, however, learning plays a significant role and, when learning is in effect, the number of iterations necessary for convergence can also be reduced to the levels of Figure 7.5.

We finish by presenting some examples of how negative feedback can indeed improve the speed of convergence to the target image. Figure 7.11 depicts a search for a target image consisting of a rubber frog, a toy boat, a plastic jar, and a plastic bottle. The top picture depicts the first iteration of the retrieval process when only positive feedback is allowed.

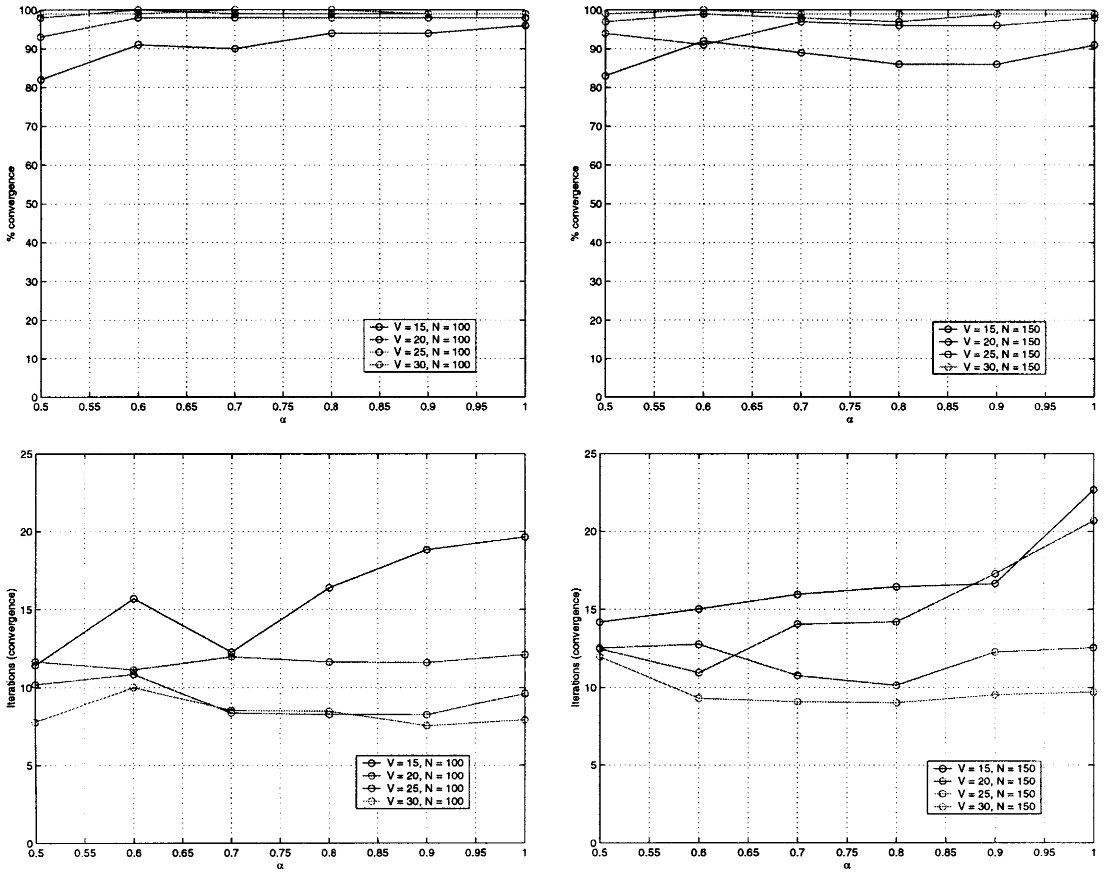


Figure 7.10: Plots, as a function of the learning factor α and number of images V examined at the end of each iteration, of the convergence rate (top) and average number of iterations for convergence (bottom) on the Columbia mosaic database. N is number of top positive feedback matches that are considered when negative feedback is taken into account. Left: $N = 100$. Right: $N = 150$.

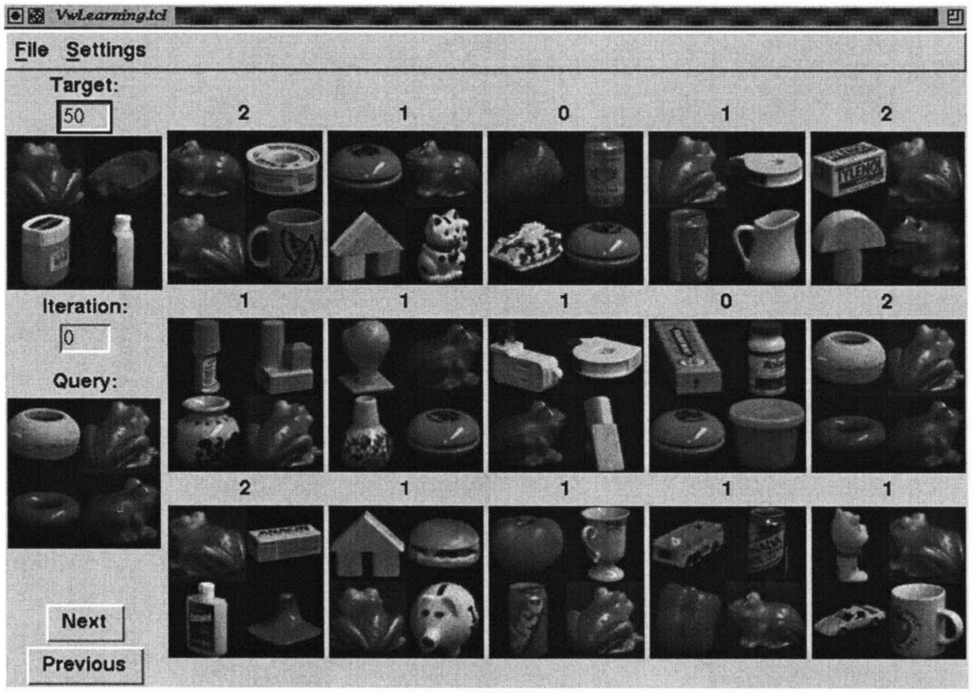
The bottom picture depicts the same iteration when both positive and negative feedback are used. The search starts with an image containing two views of the rubber frog, a plastic donut, and a clay object.

Observation of the top picture reveals that there are several images in the database where the rubber frog appears along with wooden objects, that have surface properties similar to those of clay. When negative feedback is allowed and the clay object used as a negative example, these images are penalized and the rank of the target image improves significantly. Notice that, while for positive feedback only (top image) seven slots are occupied by objects that have similar surface properties to those of the clay object, only two appear when negative examples are also allowed (bottom image). Consequently, the rank of the target image improves from higher than 15 to 6.

This example illustrates the importance of negative examples when dealing with overlapping images classes, as we had already suggested through Figure 7.2. For the mosaic databases, overlap means images sharing the same objects. In this particular query, convergence takes 6 iterations when the retrieval system is based on positive feedback alone and 1 iteration when negative examples are also allowed.

The final example (Figures 7.12 and 7.13) illustrates the importance of negative examples when it is not clear what positive examples to choose next. The target image is now composed of a blue hanger, a stapler, a clay object, and a plastic jar. The search starts with the query image that was also used to initiate the previous example (two rubber frogs, clay object, and plastic donut).

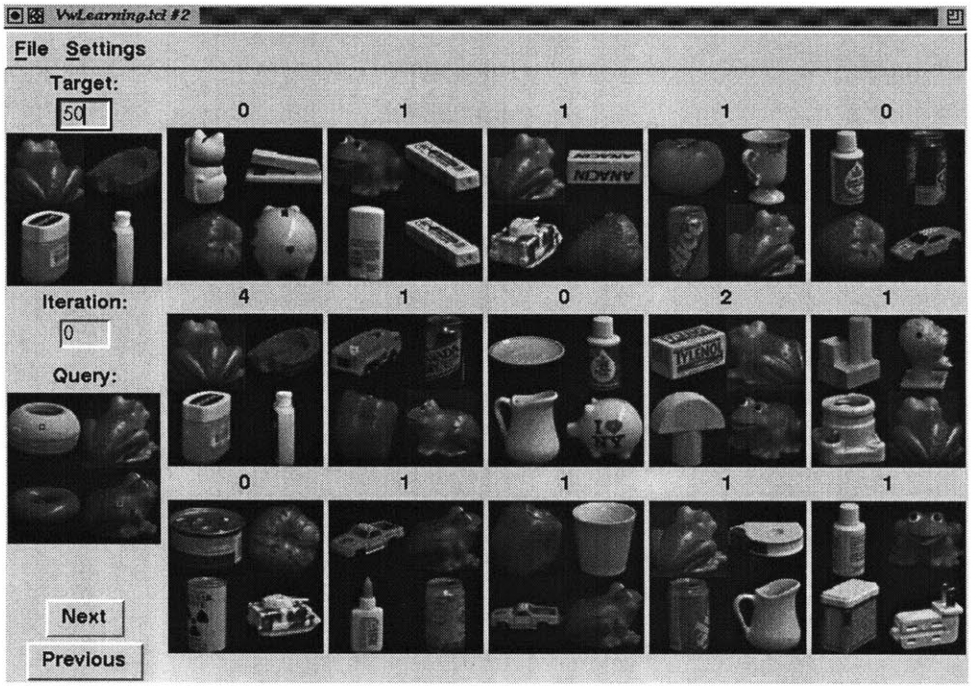
Figure 7.12 presents the first two iterations for the situation in which only positive feedback is allowed. The clay object is selected in the first iteration, and 15 images containing clay objects are returned. While this is an impressive result in terms of precision/recall, it is not very useful from the point of view of getting to the target image. In fact, it is not clear that any of the returned images is closer to the target than the query image itself. Since only positive examples are allowed, the only alternative is to choose another image containing the clay object (preferably under a different view than the previously used). This is exactly what happens, and an image containing two different views of the clay object is selected for the next query. However, since the new examples are not all that different from



Target: 50

Iteration: 0

Query:

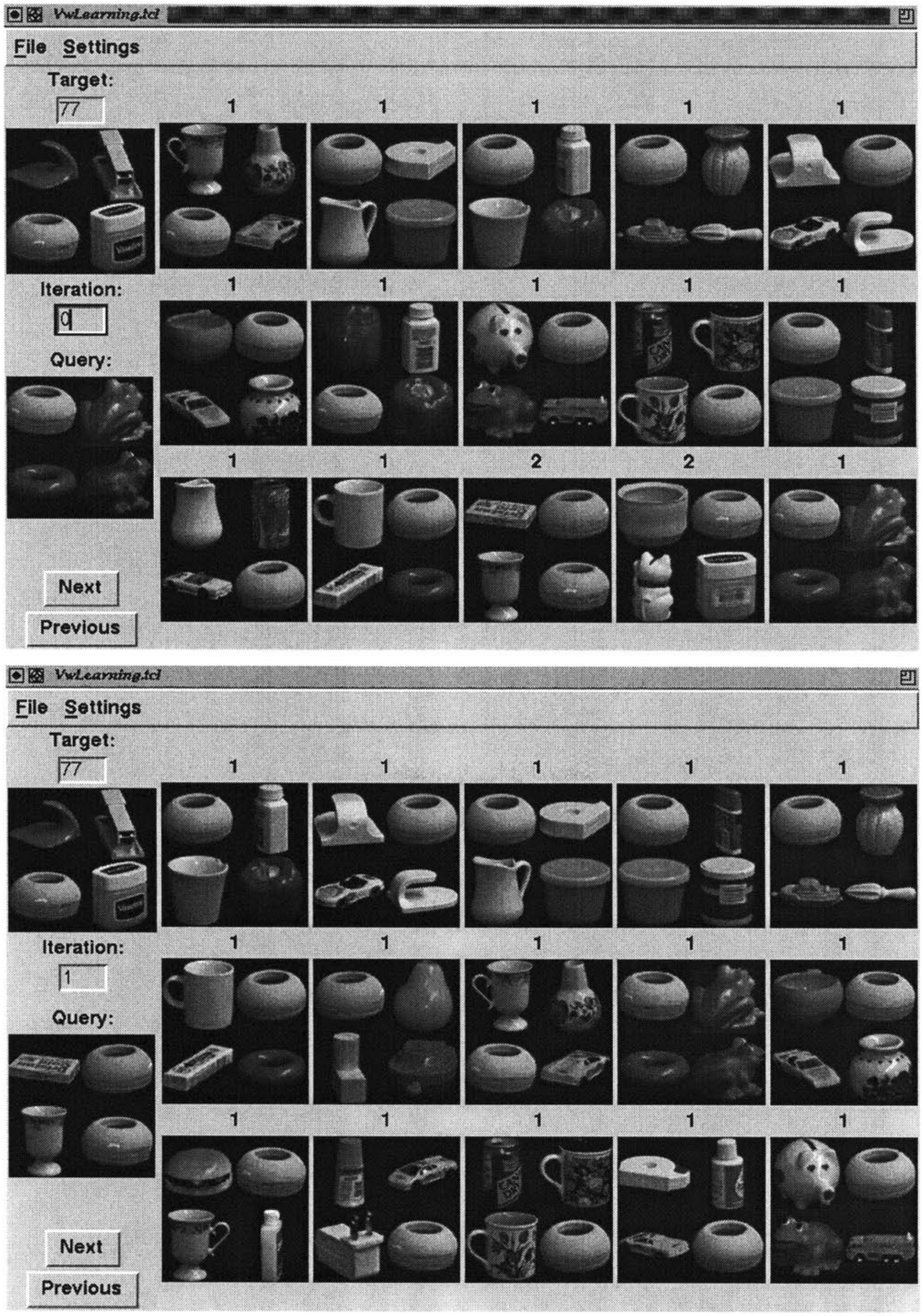


Target: 50

Iteration: 0

Query:

Figure 7.11: First iteration of relevance feedback for the same query image when only positive (top) and both positive and negative feedback (bottom) are allowed. The query itself is based on a single feature vector from each of the sub-images (8 × 8 neighborhood indicated in the center of the sub-image). Positive examples are extracted from the sub-images that are shared by the query and the target images, negative examples are extracted from the remaining sub-images.



Target: 77

Iteration: 0

Query:

Target: 77

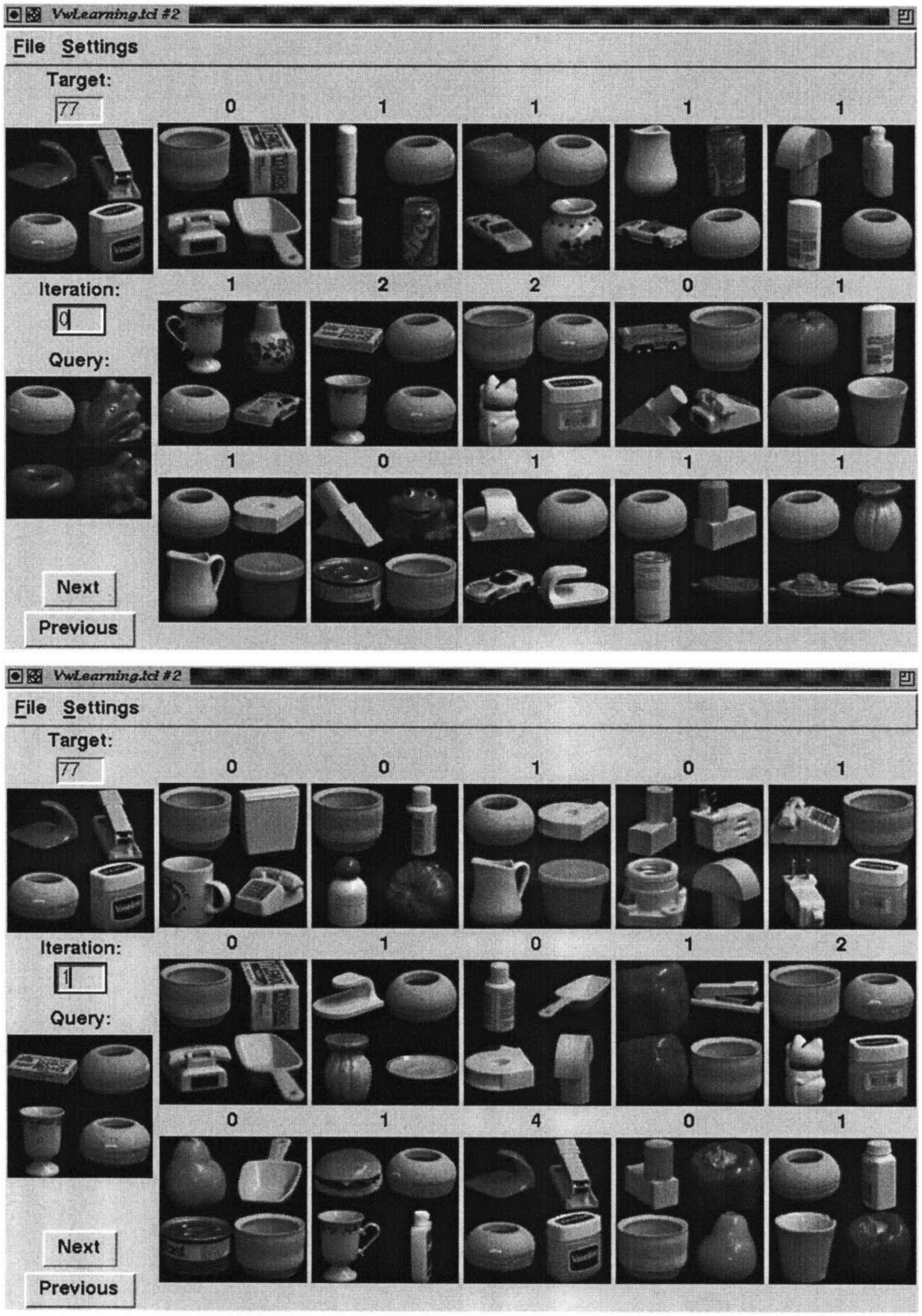
Iteration: 1

Query:

Figure 7.12: Two iterations of positive feedback.

the ones used in the first iteration, the retrieved images are approximately the same. In fact, out of the 15 images returned in the second iteration only 4 had not been already retrieved in the first. This makes it even more difficult to decide on which image to use as next query. It appears that the retrieval system is not doing much progress and, in a real retrieval scenario, the user would tend to get frustrated. In fact, proceeding in this way takes 8 iterations to get to the target.

Figure 7.13 presents results for the same query when negative feedback is allowed. In this case, in addition to the clay objects, several ceramic objects are also returned in response to the first query. Since there are no ceramic objects in the target, such objects are a good selection to use as negative examples in the next iteration. This is what happens and, despite the fact that the query images used in the two iterations are the same as in Figure 7.12, the number of images retrieved in both iterations is now only 3. Since, as before, the positive examples constrain these images to contain clay objects, it is not surprising that the target is reached in the second iteration. This example confirms what had already been pointed out in Figure 7.3: negative examples allow users to escape situations in which, because there are no good positive examples to use next, it is difficult to make progress through positive reinforcement.



Target: 77

Iteration: 0

Target: 77

Iteration: 1

Figure 7.13: Two iterations of the query of Figure 7.12 but now allowing both positive and negative feedback.

Chapter 8

Long-term learning

We have seen in the previous chapter that, in order to enable rapid convergence to the desired target image, good retrieval systems must know how to *integrate* information provided by the user over the entire retrieval session. In this chapter, we argue that this ability to learn from user interaction must also occur at longer time scales. In particular, retrieval systems should be able to develop internal representations of concepts that are likely to be of interest to their users.

Some of these representations may be hard-coded into the retrieval system from the start, i.e. it may contain modules specialized on the recognition of certain concepts that are required for semantic image understanding. Examples include detection of faces and people [148, 170, 45], face and gender recognition [112, 113], semantic scene classification [173, 182, 191, 186, 47], or even image classification according to form or function (e.g. graphics vs. photographs [6]). Since the design of such modules is usually difficult and, when feasible, requires large amounts of expert knowledge and training, they are likely to have economic justification only for visual concepts that are known to be of interest to most users.

This leaves out the majority of the concepts that are of interest to each individual user. For example, while it is unlikely that there will ever be sufficient economic pull to build a detector for the bluefinch shown in Figure 8.1, images of this bird may be among the top choices for a particular bird lover. Furthermore, users do not always require full-fledged

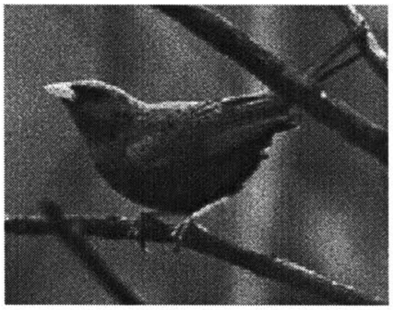


Figure 8.1: A bluefinch.

semantic recognition capabilities. E.g. while designing a generic dog recognizer is a daunting task, detecting the particular dog of an individual user may be a much simpler problem. In fact, the retrieval examples from the Columbia database show that it is relatively easy to find complicated objects in a relatively large database without requiring high-level knowledge of what an object is.

The really difficult task is *generalization*, e.g. the ability to classify the object of Figure 8.1 as a bird even though you may never have seen it before. While desirable, it is not clear that generalization will be required at all times and by all users. Instead, in most situations, users will probably be satisfied with the ability to train the retrieval system on the specific objects that are of interest to them. And, since users interested in particular visual concepts will tend to search for them quite often, there will be plenty of examples to learn from, by simply monitoring user actions. Hence, the retrieval system can build internal concept representations and become progressively more apt at recognizing them as time progresses. We refer to such mechanisms as *long-term learning* or *learning between retrieval sessions*, i.e. learning that does not have to occur on-line, or even in the presence of the user.

In addition to integrating information over time (learning), a good retrieval system should also be able to integrate information from diverse sources. Besides enabling more sophisticated queries (e.g. the ability to fuse face and speech recognition would allow queries for "the video clip where the president talks about the budget"), this ability to integrate information from diverse sources can significantly simplify the retrieval process. For example, any text annotations that may be available with the database can be used to constrain the visual search to the classes that are semantically relevant to the query.

In this chapter, we show that Bayesian retrieval can be easily extended to multiple content modalities and design a Bayesian long-term learning mechanism that complements the inference procedures discussed in Chapter 7. In particular, we show that the Bayesian approach scales well with the number of modalities to be integrated, has intuitive interpretation, and leads to extremely simple algorithms. Experimental evaluation on the Corel database demonstrates that it is possible to learn various types of visual concepts with surprising accuracy.

8.1 Probabilistic model

We start by extending the Bayesian retrieval model to multimodal content sources. First, we should notice that, conceptually, the problem does not change. The only difference is that, instead of a single feature space $\mathcal{X}$, we now have as many features spaces as the number of content modalities or *attributes* under consideration. Denoting the individual feature spaces by $\mathcal{X}_i$, we can combine them into a *meta feature space* $\mathcal{M}$ by simply concatenating the individual feature vectors. I.e. if $\mathbf{x}_i$ is a feature vector in $\mathcal{X}_i$, then

$$\mathbf{m} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\},$$

where m is the total number of attributes, is a feature vector in $\mathcal{M}$. If we define a query $\mathbf{m}_i$ as a collection of N_i feature vectors $\mathbf{m}_i = \{\mathbf{m}_{i,j}\}_{j=1}^{N_i}$ and a retrieval session as a sequence of queries $\mathbf{m}_1^t = \{\mathbf{m}_i\}_{i=1}^t$, all the results that were previously derived for $\mathcal{X}$ are also valid for $\mathcal{M}$. The only difference is that we now talk about *content classes* instead of image classes.

While conceptually the two problems are identical, in practice there is a substantive difference: the dimension of $\mathcal{M}$ can be significantly higher than that of $\mathcal{X}_i$. While this makes the estimation of joint densities infeasible in $\mathcal{M}$, it is usually true that the different modalities are associated with processes that can be considered independent. For example, it is reasonable to assume that a set of features used to characterize speech is independent of the visual features extracted from images of the speaker. This leads to the following independence assumption.

Assumption 7 *Given the target content class, the different content modalities are independent*

$$P_{\mathbf{M}|Y}(\mathbf{m}|i) = \prod_{k=1}^{m} P_{\mathbf{M}_k|Y}(\mathbf{m}_k|i).$$

For retrieval, the user instantiates a subset of the m modalities. The particular process of instantiation depends on the nature of the content associated with each attribute. While text can be instantiated by the simple specification of a few keywords, pictorial attributes are usually instantiated by example.

8.2 Incomplete queries

Of course, not all attributes need to be instantiated in all queries. Borrowing the terminology from the Bayesian network literature [127, 75, 76], we denote, for a given query, an index set $\mathbf{o}$ of *observed* attributes and an index set $\mathbf{h}$ of *hidden* attributes. E.g. if $m = 3$, attributes 1 and 2 are instantiated and attribute 3 is left unspecified then $\mathbf{o} = \{1, 2\}$ and $\mathbf{h} = \{3\}$. The likelihood of a query $\mathbf{q}$ is then given by

$$P_{\mathbf{M}|Y}(\mathbf{q}|i) = \sum_{\mathbf{q_h}} P_{\mathbf{M}|Y}(\mathbf{q_o}, \mathbf{q_h}|i), \tag{8.1}$$

where the summation is over all possible configurations of the hidden attributes[1]. Using Assumption 7 and the fact that $\sum_{\mathbf{x}} P_{\mathbf{X}|Y}(\mathbf{x}|i) = 1$,

$$\begin{aligned} P_{\mathbf{M}|Y}(\mathbf{q}|i) &= \sum_{\mathbf{q_h}} P_{\mathbf{M_o}|Y}(\mathbf{q_o}|i) P_{\mathbf{M_h}|Y}(\mathbf{q_h}|i) \\ &= P_{\mathbf{M_o}|Y}(\mathbf{q_o}|i) \sum_{\mathbf{q_h}} \prod_{k \in \mathbf{h}} P_{\mathbf{M}_k|Y}(\mathbf{q}_k|i) \\ &= P_{\mathbf{M_o}|Y}(\mathbf{q_o}|i) \prod_{k \in \mathbf{h}} \sum_{\mathbf{q}_k} P_{\mathbf{M}_k|Y}(\mathbf{q}_k|i) \\ &= P_{\mathbf{M_o}|Y}(\mathbf{q_o}|i), \end{aligned} \tag{8.2}$$

[1] The formulation is also valid in the case of continuous variables with summation replaced by integration.

i.e. the likelihood of the query is simply the likelihood of the instantiated attributes. In addition to being intuitively correct, this result is also of considerable practical significance. It means that the complexity of retrieval grows with the number of attributes specified by the user and not with the number of attributes known to the system, which can therefore be made arbitrarily large.

8.3 Combining different content modalities

While the Bayesian framework can integrate all types of content modalities, in this thesis we restrict our attention to the integration of visual attributes and text annotations. In particular, we consider retrieval sessions $\mathbf{m}_1^t = \{\mathbf{t}_1^t, \mathbf{x}_1^t\}$, composed of text ($\mathbf{t}_1^t$) and visual attributes ($\mathbf{x}_1^t$). Combining (7.7) with Assumption 7, assuming only positive examples, and disregarding the decay factor[2], we obtain

$$g^*(\mathbf{m}_1^t) = \arg\max_i \left\{ \log P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i) + \log P_{\mathbf{T}_t|Y}(\mathbf{t}_t|i) + \log P_{Y|\mathbf{M}_1^{t-1}}(i|\mathbf{m}_1^{t-1}) \right\}. \tag{8.3}$$

This equation has several equally intuitive interpretations. The standard one is to consider $\log P_{Y|\mathbf{M}_1^{t-1}}(i|\mathbf{m}_1^{t-1})$ as a prior belief, for iteration t, that gives more weight to those classes that have performed well in the past and $P_{\mathbf{X}_t|Y}(\mathbf{x}_t|i)$ and $P_{\mathbf{T}_t}(\mathbf{t}_t|i)$ as the likelihoods of the current observations. Two alternative interpretations are however possible.

The first, and vision centric, is that the optimal class is the one which would best satisfy the visual query alone but with a prior consisting of the combination of the second and third terms. By instantiating text attributes, the user establishes a *context* for the evaluation of visual similarity that changes the system's prior beliefs about which class is most likely to satisfy the visual query. Or, in other words, the text attributes provide a means to *constrain* the visual search. The second, and text centric, is to consider the second term the likelihood function, with the combination of the first and the third forming the prior. In this interpretation, the visual attributes constrain what would be predominantly a text-based

[2] In order to simplify the presentation, in this chapter we ignore negative examples and decay factors. However, all results are extensible to the generic case and the extension is straightforward: simply modify the denominator of (7.7) in the same way as the numerator is changed and add α and $(1-\alpha)$ where appropriate.

search.

Independently of the interpretation, the significance of (8.3) is that it illustrates one of the most attractive properties of the Bayesian retrieval formulation: because all models speak the same *language* (the language of probabilities) it is relatively easy to combine the outputs of specialized modules into a *global* inference. Because, due to Assumption 7, we are relying on a simple model for the dependencies between the attributes, the integration is fairly simple (simply adding log-likelihoods). If more complex models were available, it would still be possible even though probably through a more complicated expression. Notice that the only fundamental requirement for the integration to be possible is that the different attributes can be modeled probabilistically. In the previous chapters, we have seen how this can be done for pictures. We next concentrate on the issue of representing text.

8.4 Text representation

The standard representation for text retrieval is the *vector space model* [154, 50] where each document is represented as a collection of *indexing terms*. An indexing term can be a word, a group of words, or a word stem[3] among others. The space of indexing terms can be seen as the observation space $\mathcal{Z}$ for text, and associated with each index term and each document there is a binary feature indicating if the term appears, or not, in the document. That is, the document is represented by a binary vector $\mathbf{t} = \{t_1, \dots, t_L\}$, where $t_j = 1$ ($t_j = 0$) if the index term i appears (does not appear) in the document, and L is the total number of indexing terms known to the retrieval system. A query is simply a collection of index terms and is therefore represented in the same way.

The simplest possible retrieval model is to, in response to a query, extract from the database all the documents whose feature vectors match that of the query. This, however, does not take into account the fact that some terms are more relevant to the characterization of a document than others. A better approach is therefore to associate a weight with each component of the feature vector. Even though such weights are not always justified

[3] A word stem [154] is a descriptor for a collection of words with similar semantics, e.g. different verb tenses, variations on a word by the introduction of suffixes, etc.

probabilistically [51], usually they are constrained to the interval $[0, 1]$ and therefore have a probabilistic interpretation.

Truly probabilistic representations are, however, popular in the text retrieval literature. Among these the most commonly used is the so-called *naive Bayes* model [87] which assumes that the L binary features are independent, and models each feature as a Bernoulli random variable

$$P_{T_j|Y}(t_j|i) = \begin{cases} 1 - p_{i,j}, & \text{if } t_j = 0 \\ p_{i,j}, & \text{if } t_j = 1. \end{cases} \tag{8.4}$$

The probabilities $p_{i,j}$ can be seen as the weight vector for documents from class i.

Despite its simplicity, and after decades of research on more sophisticated alternatives, the naive Bayes model has proven difficult to beat [87]. In terms of the discussion above, it is equivalent to considering each possible index term as a different content attribute and relying on Assumption 7. Hence, the naive Bayes model fits naturally into the overall Bayesian retrieval formulation developed in this thesis and, because it is simple, we adopt it[4]. Combining Assumption 7 with (8.2) and (8.4), and taking logs we obtain

$$\log P_{\mathbf{T}|Y}(\mathbf{t}|i) = \sum_j \delta_{t_j,1} \log p_{i,j} + \sum_j \delta_{t_j,0} \log(1 - p_{i,j}),$$

where δ is the Kronecker delta function (2.3). When there are only positive examples

$$\log P_{\mathbf{T}|Y}(\mathbf{t}|i) = \sum_j \delta_{t_j,1} \log p_{i,j}. \tag{8.5}$$

8.4.1 Parameter estimation

There are several ways to estimate the parameters $p_{i,j}$. When an actual text document is available (as is usually the case with text retrieval), the estimates can be derived from frequency counts, i.e. $p_{i,j}$ is a function of the number of times that term j appears in document (or document class) i. While there are image retrieval scenarios in which a text document is associated with each image (e.g. web pages), this does not always hold.

[4] Notice that this does not mean that other more sophisticated text models could not be used in the Bayesian retrieval formulation.

Furthermore, it has not yet been established that, when a free text document is available, visual retrieval will add any improvements to text-based retrieval. For these reasons, we concentrate on the situations in which a detailed textual description of the image content is not available. In such cases, the straightforward solution to the annotation problem is to use manual labeling, relying on the fact that many databases already include some form of coarse image classification. For example, an animal database may be labeled for cats, dogs, horses, and so forth. In this case, it suffices to associate the term "cats" with t_1, the term "dogs" with t_2, etc and make $p_{i,1} = 1$ for pictures with the cats label and $p_{i,1} = 0$ otherwise, $p_{i,2} = 1$ for pictures with the dogs label and $p_{i,2} = 0$ otherwise, and so forth. In response to a query instantiating the "cats" attribute, (8.5) will return 0 for the images containing cats and $-\infty$ for those that do not. In terms of (8.3) (and associated discussion in section 8.3), this is a *hard constraint*: the specification of the textual attributes eliminates from further consideration all the images that do not comply with them.

Hard constraints are usually not desirable, both because there may be annotation errors and because annotations are inherently subjective. For example, while the annotator may place leopards outside the cats class, a given user may use the term "cats" when searching for leopards. A better solution is to rely on *soft constraints* where the $p_{i,j}$ are not restricted to be binary. In this case, the "cats" label could be assigned to leopard images, even though the probability associated with the assignment would be small. In this context, $p_{i,j}$ should be thought of as the answer to the question "what is the likelihood that users will instantiate attribute t_j given that they are interested in images from class i?". In practice, it is usually 1) too time consuming to define all the $p_{i,j}$ manually, and 2) not always clear how to decide on the probability assignments. A better alternative is to rely on learning.

8.4.2 Long term learning

Unlike the learning algorithms discussed in section 7.2, here we are talking about *long-term learning* or *learning across retrieval sessions*. The basic idea is to let users attach a label to each of the regions that are provided as queries during the course of the normal interaction with the retrieval system. For example, if in order to find a picture of a snowy mountain a user selects a region of sky, the user has the option of labeling that region with the word "sky" establishing the "sky" attribute. Learning then consists of estimating the probabilities $p_{i,j}$ from the example regions. For this we rely on the following assumption.

Assumption 8 *When, during retrieval, a user instantiates a text attribute t_j, the user is looking for images that contain regions similar to those previously provided as examples for that attribute.*

The assumption simply means that we expect users to be consistent, providing a basis for the estimation of the $p_{i,j}$. It states that the instantiation of attribute t_j is equivalent to complementing the visual component of the query with the examples that were previously stored for attribute t_j. Mathematically, the query $\{\mathbf{x}, t_i = 1\}$, where $\mathbf{x}$ is a collection of visual feature vectors, is equivalent to the query $\{\mathbf{x}, \mathbf{e}_j\}$ where $\mathbf{e}_j$ is the *example set* containing the example regions for attribute t_j, $\mathbf{e}_j = \{\mathbf{e}_{j,1}, \ldots \mathbf{e}_{j,K}\}$. Since, from Assumption 7 and (8.5),

$$\begin{aligned} \log P_{\mathbf{X},T_j|Y}(\mathbf{x}, 1|i) &= \log P_{\mathbf{X}|Y}(\mathbf{x}|i) + \log P_{T_j|Y}(1|i) \\ &= \log P_{\mathbf{X}|Y}(\mathbf{x}|i) + \log p_{i,j}, \end{aligned}$$

this means that

$$\log P_{\mathbf{X}|Y}(\mathbf{x}|i) + \log p_{i,j} = \log P_{\mathbf{X}|Y}(\mathbf{x}|i) + \log P_{\mathbf{X}|Y}(\mathbf{e}_{j,1}, \ldots \mathbf{e}_{j,K}|i)$$

or

$$p_{i,j} = P_{\mathbf{X}|Y}(\mathbf{e}_{j,1}, \ldots \mathbf{e}_{j,K}|i).$$

From Assumption 2,

$$\begin{aligned} \log p_{i,j} &= \sum_k \log P_{\mathbf{X}|Y}(\mathbf{e}_{j,k}|i) \\ &= \sum_{k,l} \log P_{\mathbf{X}|Y}(\mathbf{e}_{j,k,l}|i), \end{aligned} \tag{8.6}$$

where $\mathbf{e}_{j,k,l}$ is the l^{th} feature vector from example region k for text attribute j.

This expression is all that has to be computed in the learning stage. Notice that, because only the running sum of $\log P_{\mathbf{X}|Y}(\mathbf{e}_{j,k}|i)$ must be saved from session to session, there is no need to keep the examples themselves. Instead, it suffices to store one number per image class/attribute pair. Notice also that, since all the $\log P_{\mathbf{X}|Y}(\mathbf{e}_{j,k}|i)$ terms have to be computed for the queries in which the examples $\mathbf{e}_{j,k}$ are defined, there is no computational cost associated with the learning procedure itself; i.e., long-term learning is highly efficient in terms of both computation and memory.

Grounding the annotation model directly in visual examples also guarantees that the beliefs of (8.6) are of the same order of magnitude as those of (5.7), making the application of (8.3) straightforward. If different representations were used for annotations and visual attributes, one would have to define weighting factors to compensate for the different scales of the corresponding beliefs. Determining such weights is usually not a simple task.

There is, however, one problem with the example-based learning solution. While the complete set of examples of a given concept may be very diverse, individual image class models may not be able to account for all this diversity. In the case of "sky" discussed above, while there may be examples of sunsets, sunrises, and skies shot on cloudy, rainy or sunny days in the "sky" example set, particular image classes will probably not encompass all this variation. For example, as illustrated in Figure 8.2, images in the class "pictures of New York at sunset" will only explain well a fraction of the sunset examples. It follows that, while this class should receive a high rank with respect to "skyness," there is no guarantee that this will happen since it assigns low probability to a significant number of sky examples.

The fact is that most image classes will only overlap partially with broad concept classes like "sky". The problem can however be solved by requiring the image classes to explain well only a subset of the concept examples. One solution is to rank the examples according to their probability and apply (8.6) only to the top ones,

$$\log p_{i,j} = \sum_{r=1}^{R} \log P_{\mathbf{X}|Y}(\mathbf{e}_{j,k}^{(r)}|i), \tag{8.7}$$

where $\mathbf{e}_{j,k}^{(r)}$ is the example region of rank r and R a small number (10 in our implementation).

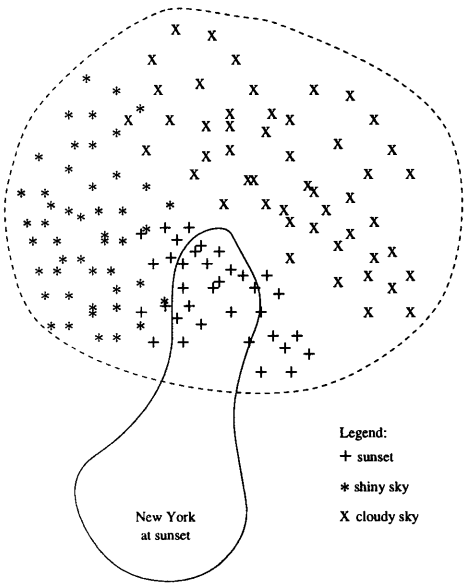


Figure 8.2: An image class may not encompass all the examples from a given attribute, even when the attribute is present. The solid line represents the density of the class "pictures of New York at sunset", "+" are examples of sunsets, "x" of shiny sky, "*" of cloudy skies, and the dashed line the overall density for "sky".

8.5 Experimental evaluation

The performance of a long-term learning algorithm will not be the same for all concepts that may need to be learned. In fact, the learnability of a concept is a function of two main properties: *visual diversity* and *distinctiveness* on the basis of local visual appearance. Diversity is responsible for misses, i.e. instances of the concept that cannot be detected because the learner has never seen anything like them. Distinctiveness is responsible for false positives, i.e. instances of other concepts that are confused with the desired one. Since the two properties are functions of the particular image representation, it is important to test the performance of the learner with concepts from various points in the diversity/distinctiveness space.

We relied on the Corel database to evaluate long-term learning and identified five such concepts: a flag, tigers, sky, snow, and vegetation. The flag is representative of computer graphics objects, such as logos, that tend to be presented with small variation of visual appearance and therefore are at the bottom of the diversity scale. Tigers (like most animals) are next: while no two tigers are exactly alike, they exhibit significant uniformity in visual appearance. However, they are usually subject to much stronger imaging transformations than logos (e.g. partial occlusion, lighting, perspective). Snow and sky are representative of the next level in visual diversity. Even though relatively simple concepts, their visual appearance varies a lot with factors like imaging conditions (e.g. shiny vs. cloudy day) or the time of the day (e.g. sky at noon vs. sky at sunset). Finally, vegetation encompasses a large amount of diversity.

In terms of distinctiveness, logos rank at the top (at least for Corel where most images contain scenes from the real world), followed by tigers (few things look like a tiger), vegetation, sky and snow. Snow is clearly the less distinctive concept, on the basis of local visual appearance, since large patches of smooth white surfaces are common in many scenes (e.g. clouds, white walls or other objects like tables, paper, etc.). The distribution of the five concepts on the diversity/distinctiveness space is shown in Figure 8.3.

In order to train the retrieval system, we annotated all the images in the database according to the presence or not of each of the five concepts. We then randomly selected a

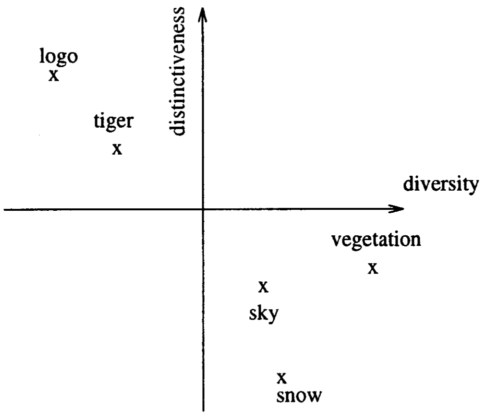


Figure 8.3: Distribution of the concepts in terms of diversity and distinctiveness.

number of example images for each concept and manually segmented the regions where the concepts appeared. These regions were used as examples for learning. Concept probabilities were estimated for each image outside the training set using (8.7) and, for each concept, the images were ranked according to these probabilities. Figure 8.4 presents the resulting precision/recall curves for the five concepts. Retrieval accuracy seems to be directly related to concept distinctiveness: a single training example is sufficient for perfect recognition of the logo and with 20 examples the systems does very well on tigers, reasonably well on vegetation and sky, and poorly on snow. These are surprisingly good results, particularly if one takes into account the reduced number of training examples and the fact that the degradation in performance is natural for difficult concepts.

Performance can usually be improved by including more examples in the training set, as this reduces the concept diversity problem. This is illustrated in Figure 8.5, where we show the evolution of precision/recall as a function of the number of training examples for sky and tigers. In both cases, there is a clear improvement over the situation in which only one example is used. This result is particularly significant, since the one-example scenario is equivalent to the standard query-by-example paradigm. Under this paradigm, a user would try to retrieve images containing a given concept by providing the retrieval system with one example of that concept. As the figures clearly demonstrate, one example is usually not enough, and long-term learning does improve performance by a substantial amount. In the particular case of sky, it is clear that performance can be made substantially better than

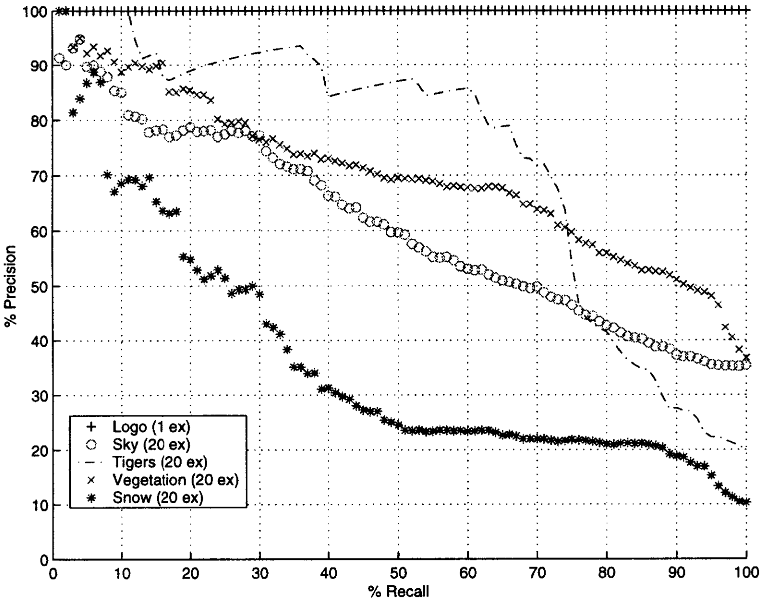


Figure 8.4: Performance of long-term learning. Precision/recall curves for the five concepts described in the text.

that of Figure 8.4 by taking more examples into account.

On the other hand, Figure 8.6 shows that more examples make a difference only when the performance is limited by a poor representation of the concept diversity, not distinctiveness. For snow, where the latter is the real bottleneck, providing more examples does not seem to make a difference.

Figures 8.7 to 8.11 show the top 36 matches for the five concepts. These figures illustrate well how the long term learning mechanism is robust with respect to concept diversity, either in terms of different camera viewpoints, shading, occlusions, etc (e.g. tiger images) and variations in visual appearance of the concept itself (e.g. sky or vegetation images). In general, the errors are intuitive: for the logo retrieval is perfect; for sky and vegetation errors correspond to images that could have been labeled either way (e.g. images of a tiger or an owl with some trees on the background were labeled as not containing vegetation); and for snow errors tend to be images containing large smooth white surfaces (walls, flower, clouds, car hoods, etc.). The less intuitive errors happen for tigers, where the texture of some paintings is confused with a tiger, but are few.

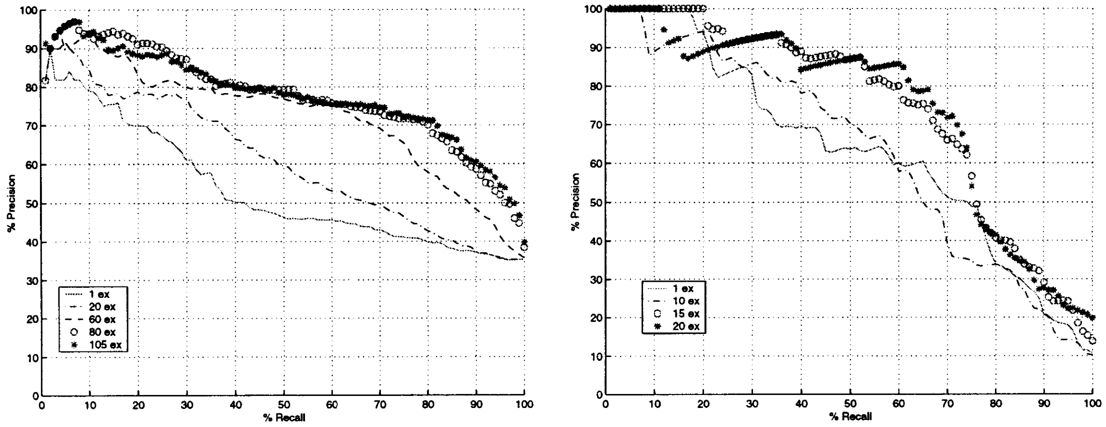


Figure 8.5: Evolution of precision/recall as a function of the training set size for sky (left), and tigers (right).

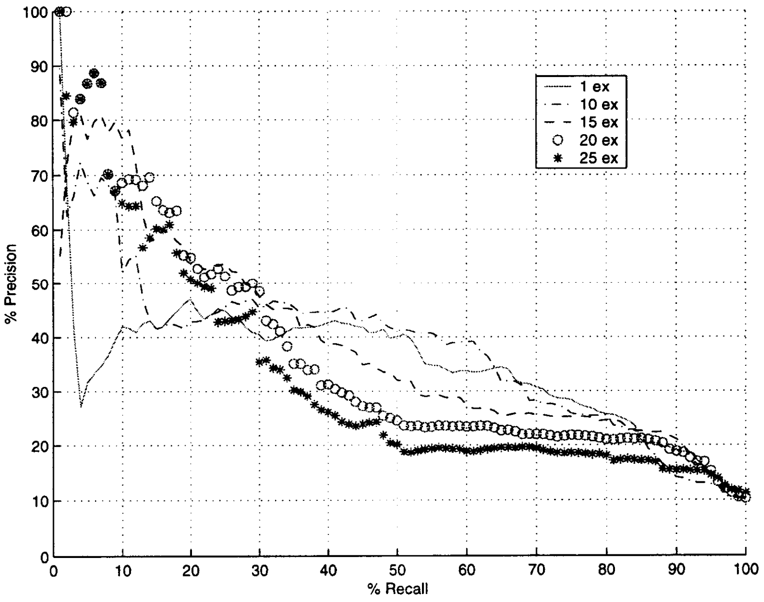


Figure 8.6: Evolution of precision/recall for snow.

The snow example actually illustrates one advantage of systems that learn by example: since users provide all the examples, they can develop an understanding of which concepts are easier to learn. In this case, because the training for snow consisted of smooth white image patches and all the errors contain such patches, it is not clear how the system could be trained to improve its ability to detect snow. Hence, a user could quickly realize that snow is a difficult concept for the system. This is indeed the case since distinguishing a patch of snow from a white wall requires high-level scene understanding abilities that the system does not possess.

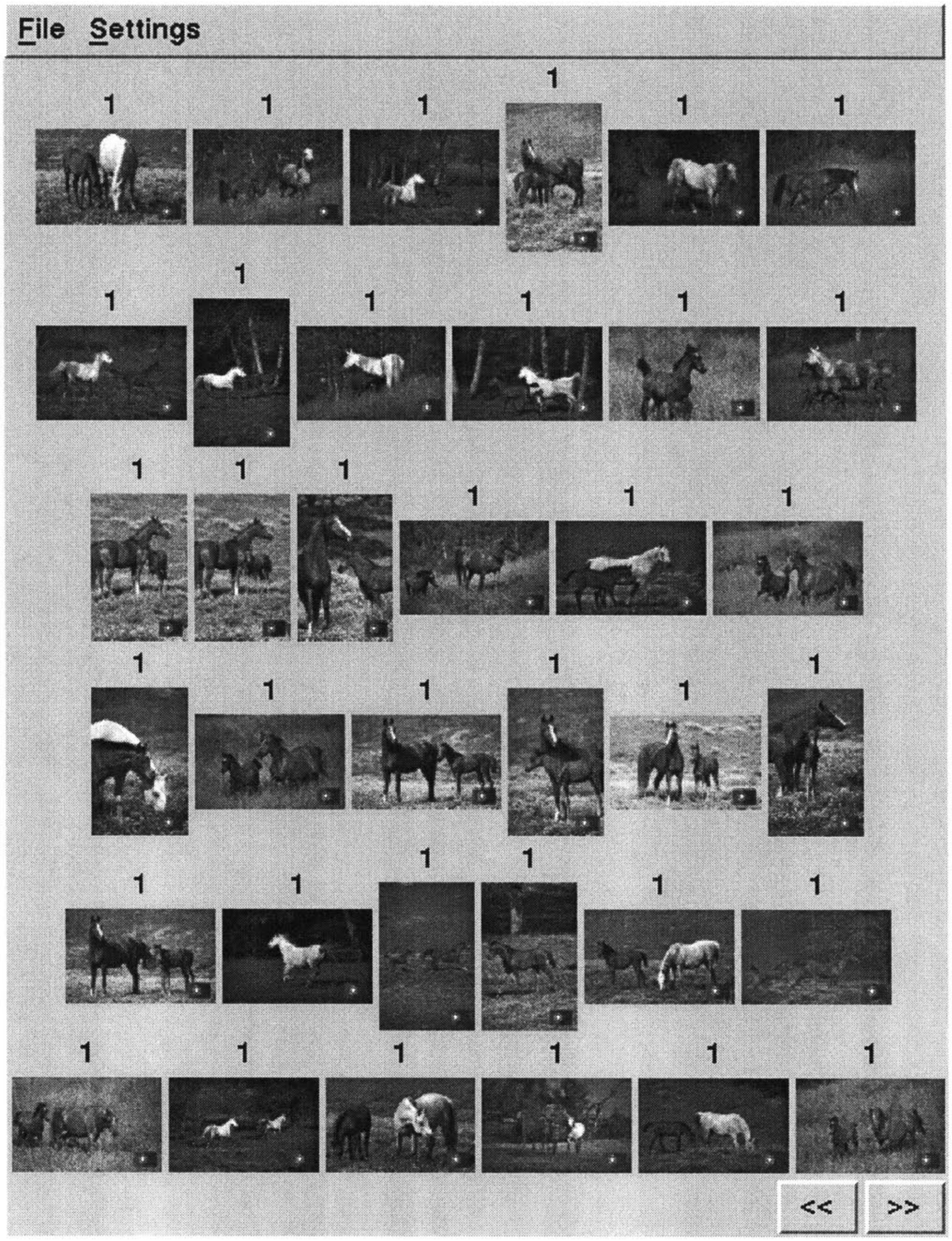


Figure 8.7: Top 36 matches for the flag concept. The number shown on top of each image indicates if the image was annotated as containing the concept (1) or not (0).

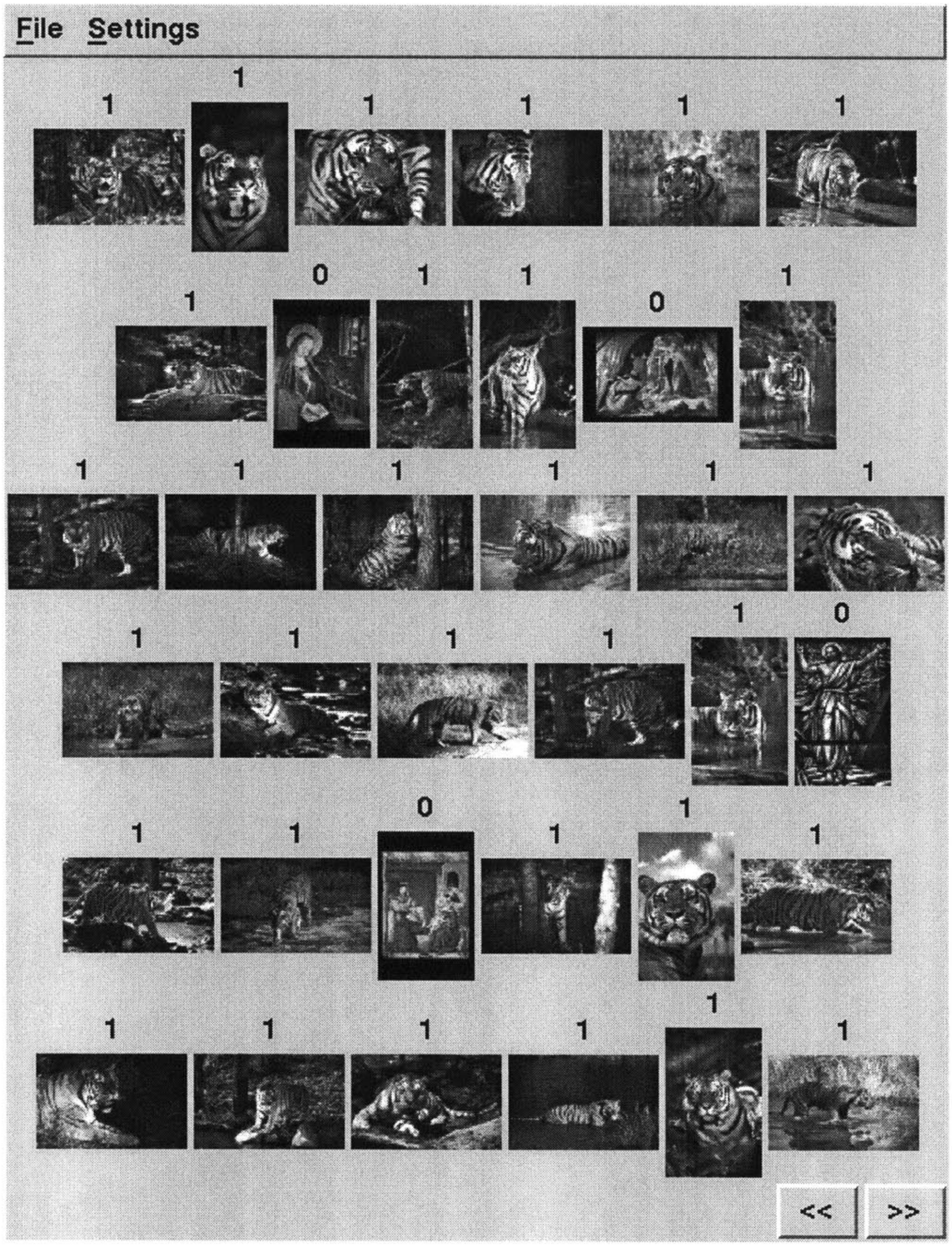


Figure 8.8: Top 36 matches for the tiger concept.

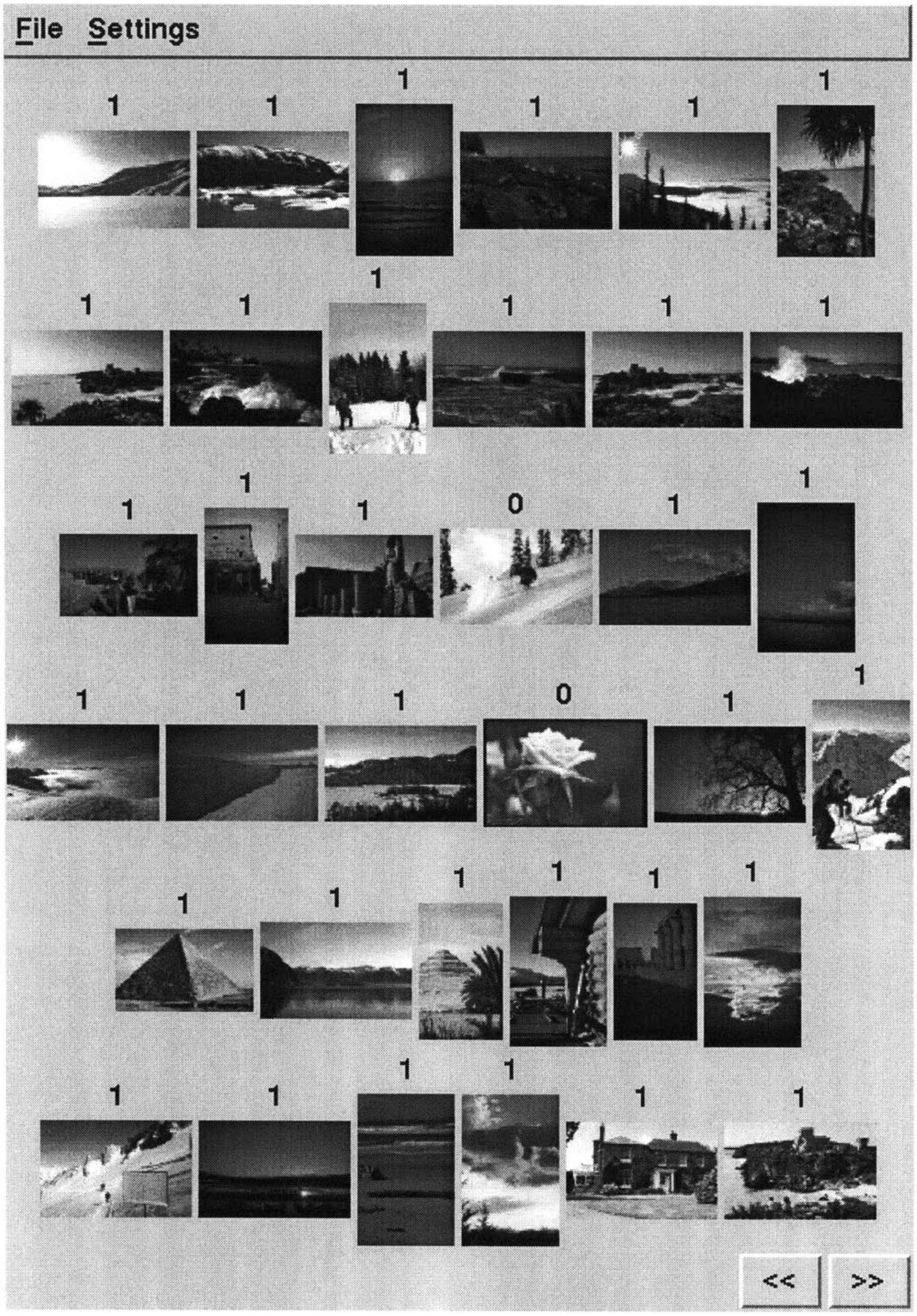


Figure 8.9: Top 36 matches for the sky concept.

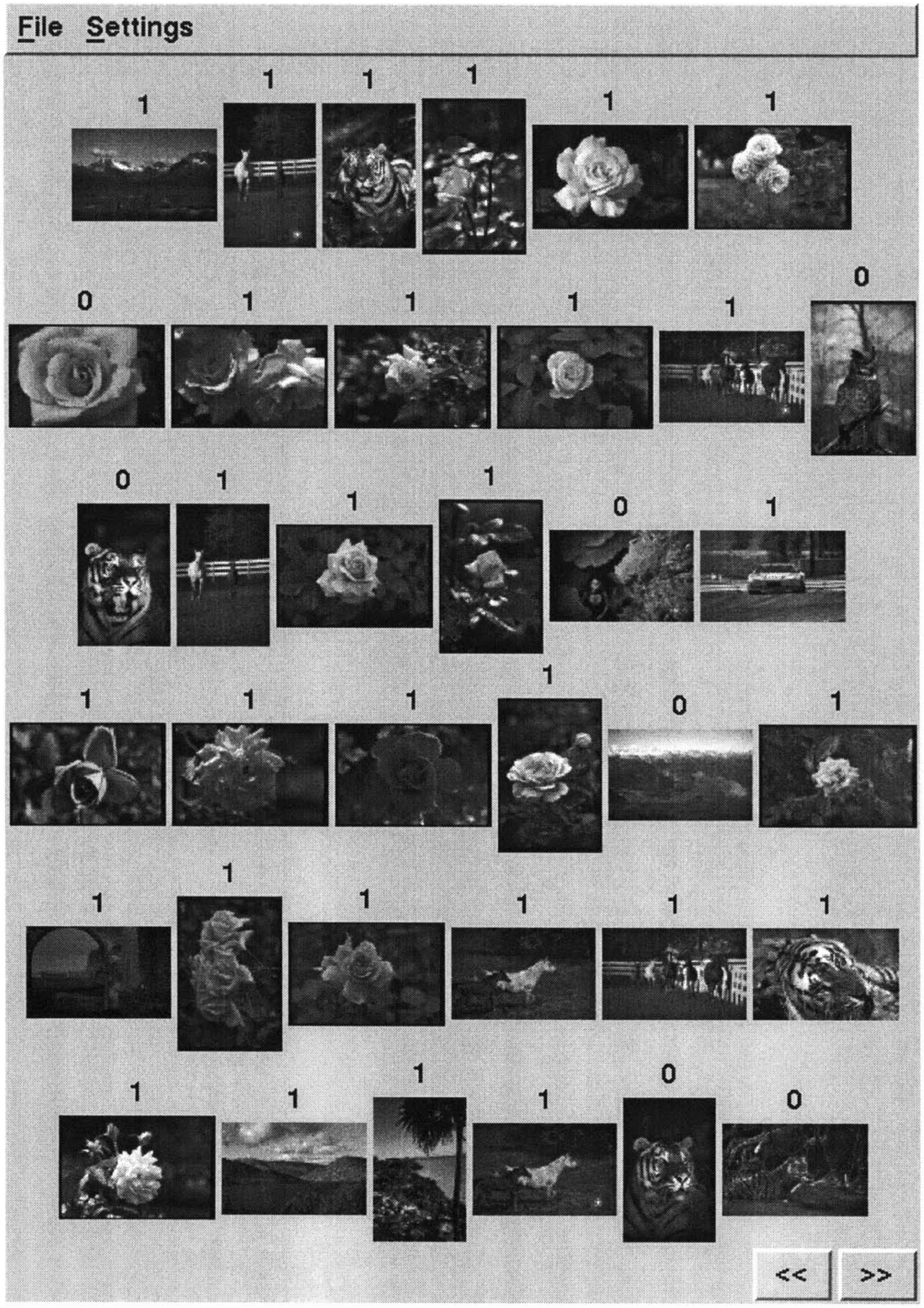


Figure 8.10: Top 36 matches for the vegetation concept.

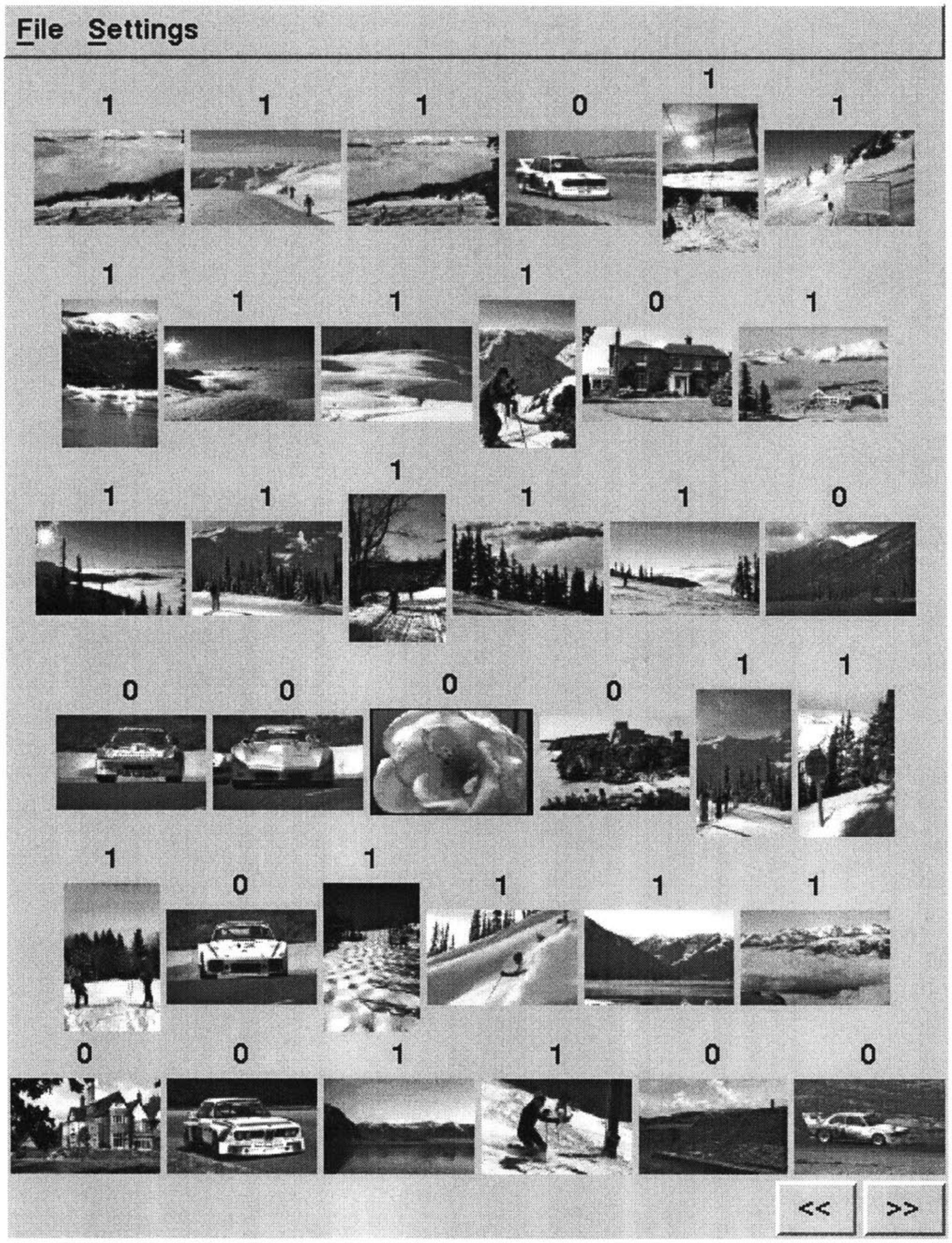


Figure 8.11: Top 36 matches for the snow concept.

Chapter 9

The RaBI retrieval system

In the previous chapters, we introduced a decision-theoretic formulation for the retrieval problem and shown that it has interesting properties as a unified solution to the visual recognition and learning problems. In this chapter, we discuss the implementation of a CBIR system based on this decision-theoretic formulation. The "Retrieval as Bayesian Inference" (RaBI) system is an image search engine designed to operate under the query by example search paradigm. It implements most of the retrieval functionalities discussed so far, including support for multimodal queries (visual and text), short- and long-term learning, local queries, and both positive and negative examples. We next discuss the implementation of each of these functionalities in some detail.

9.1 System overview

Figure 9.1 presents a generic block diagram of RaBI. The system consists of three software modules: a Java client, a SQL server, and an image server. The Java client implements the user interface and communicates with the image server through a simple custom protocol. It is a light-weight applet that can run on any platform either in stand-alone mode or from an Internet browser. Once connected to the image server, the client allows the user to select a database and perform text-based searches, visual searches, or a combination of both.

The SQL server implements two functionalities. First, it allows one level of indirection

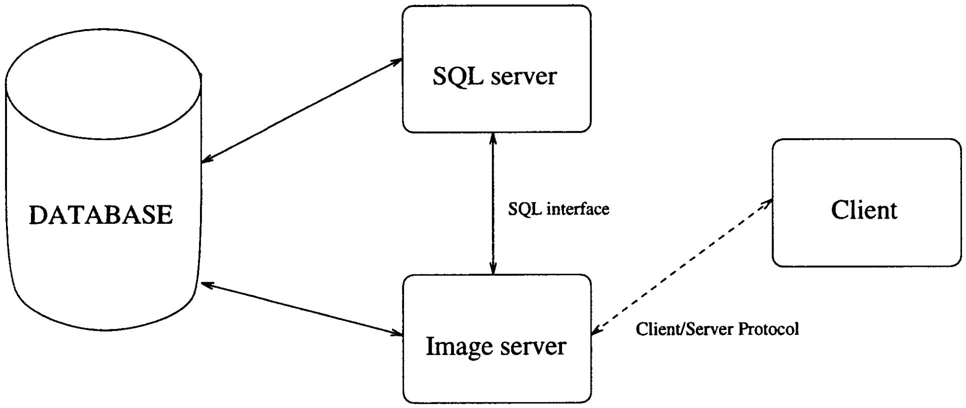


Figure 9.1: Block diagram of the RaBI retrieval system.

with respect to the physical location of all the files on the database. This simplifies the process of relocating all the data if that becomes necessary. Second, it supports simple text-based queries that can be performed very efficiently using standard SQL indexing mechanisms. The image server accepts queries from the client, performs the searches, and transmits back the search results. Some of the text-based searches are dispatched to the SQL server. We next provide a more detailed description of each of the modules.

9.1.1 Client

The client consists of two major modules: a graphical user interface (GUI) and the client/server protocol (CSP) interface. The GUI is presented in Figure 9.2. The larger image displayed in the top left corner is the *query canvas*. By dragging the mouse over a given region of the image, the user can select that region as a positive (by pressing the left button) or negative (right button) query example. Boundaries of the image blocks in the positive examples are displayed in green, while those in negative examples are shown in red.

Three text boxes are shown below the query canvas. The one on the left displays the text keywords known to the system, the one in the middle shows which vision modules are available, and the one on the left displays the visual concepts on which the system has been trained. By selecting the "Show me" or "Not interested in" buttons the user can select any of the entries on the text boxes as positive or negative text examples, respectively. The

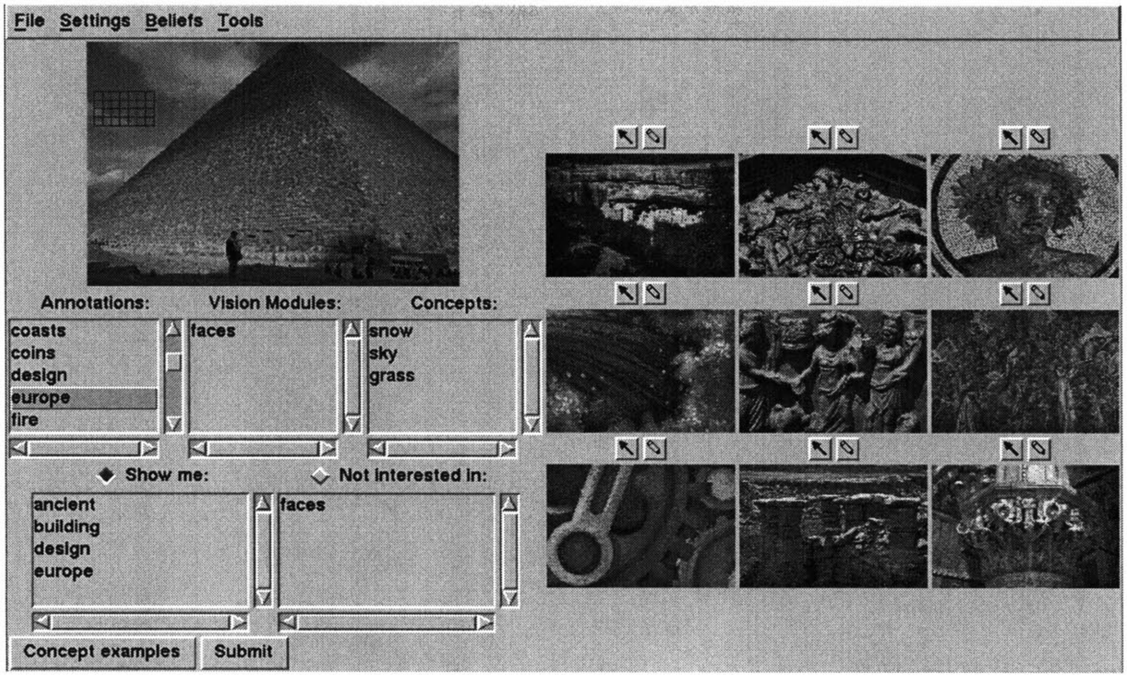


Figure 9.2: The GUI of the RaBI system.

selected keywords then appear in the two boxes in the bottom-left.

When the user presses the "Submit" button, the query (consisting of all the keywords, the lists of coordinates of the selected image blocks, and the identifier of the query image) is passed to the CSP interface that encodes all the information and dispatches it to the image server. The server replies with a list of URLs that point to the images that best satisfy the query. This response is decoded by the CSP interface and the top $M \times N$ images are displayed on the right half of the GUI, as illustrated by Figure 9.2. The variables M and N can be defined by the user. Two buttons are displayed above each image. By pressing the button containing an arrow, the user can make the image appear in the query canvas and use it as a basis for subsequent queries. By pressing the other button (pencil icon), the user can change the text annotations associated with the image (e.g. give less weight to a particular keyword).

Whenever visual examples are selected in the query canvas, it is possible to associate a concept keyword with them. For this, the user simply needs to press the "Concept examples" button. A dialog box then presents a list of the known concepts and the user has the option of either selecting one of these concepts or defining a new one. The concept examples are then sent to the image server and passed on to the concept learning algorithm.

Finally, the menus on the top-left corner of the interface allow a series of routine operations, such as clearing the short-term learning memory (when the user is starting a new query), opening and closing databases, and setting the values of the various parameters of the retrieval algorithm (e.g. the number of subspaces to be considered and the decay factor α for learning).

The CSP is a simple communications protocol consisting of 1) a control sequence that indicates what type of operation the user is carrying out (e.g. query, opening new database, providing concept examples, etc.), and 2) a collection of data fields that depends on the type of operation.

9.1.2 SQL server

The SQL database contains pointers to all the image information and implements a preliminary image classification according to database and image class. It associates three information components to each image: a database name, an image class, and two strings indicating the image URL and the physical image location on disk.

The database classification establishes a very coarse image grouping. Current databases are the ones discussed in the previous chapters: Corel, Columbia, Brodatz, Columbia mosaics, and Brodatz mosaics. Each database defines a set of keywords that allow finer image grouping. These appear in the "Annotations" box of the GUI of Figure 9.2.

Image classes allow the user to restrict the query to a subset of the database, and can be defined in several ways. For some databases, images are naturally grouped into obvious classes. For example, Brodatz (Columbia) contains various views of each texture (object), while Corel images are organized into thematic directories, each containing 100 images. In these cases, textures, objects, and themes constitute natural classes for image grouping. If a text model is available, classes can also be derived automatically, by associating each image to the keyword that explains it with highest probability. Finally, classes can always be defined manually.

Despite the support for database and image classes, there is no obligation on the part of the user to use this information at any point of the retrieval operation. Instead, the classification should be seen as a filtering mechanism to quickly eliminate many images that are a priori known to be irrelevant for the search, and direct the attention of the retrieval system to the images that matter. Since changing the database in the middle of a query may lead to inconsistencies in learning, this mechanism can only be used at the beginning of a visual retrieval session. If it is, this initial database/class query is based on standard SQL indexing structures and, therefore, very efficient.

9.1.3 Image server

The image server has two main operation modes. In the database selection mode, it 1) provides the client with a list of available databases and image classes, 2) receives a text-based query relative to these, and 3) dispatches this query to the SQL server. The SQL server replies with the list of internal file paths and URLs associated with the images to be considered in the subsequent visual queries. The image server then assembles all the data-structures required for these queries and enters the visual query mode.

Seven different data-structures are assembled. The first is a collection of matrices containing the mixture parameters that are required to evaluate the likelihood of the visual query features according to (2.14). Two others are a *text probability matrix*, and a *concept probability matrix*. In these matrices, each row corresponds to a textual or visual concept and each column corresponds to one image, the entry i, j containing the parameter $\log p_{i,j}$ required by (8.5). Textual concepts are keywords derived either from the text-model discussed in section 8.4 or from vision sensors. For example, a face detector can be applied to each image in the database, assigning the "face" keyword to each image with a certain probability. Visual concepts are learned over time as discussed in section 8.4.2.

The remaining data-structures are a *visual likelihood buffer*, a *concept likelihood buffer*, a *text likelihood buffer*, and a *short-term memory buffer*. The first three hold information relative to the current iteration of the retrieval process, while the fourth is reset only when the user starts a new query. Their role is illustrated by Figure 9.3, which provides a graphical depiction of the inner workings of the image server.

In response to a visual query, the image server uses the mixture parameters associated with each image in the database and (2.14) to evaluate the log-likelihood of the visual features. The resulting log-likelihoods are stored in the visual likelihood buffer. The textual component of the query is then analyzed. The rows of the text and concept likelihood matrices associated with the keywords and concepts selected in the query are added and stored in the text and concept likelihood buffers, respectively. The log-likelihoods associated with the current query are then evaluated by adding the three buffers. The following step is to combine these log-likelihoods with the log-posterior probabilities given the past interaction, that are stored in the short-term memory buffer. This is done according to (7.5), i.e. by weighting the entries in the two buffers according to the memory decay factor α. The result is a set of log posterior probabilities that are passed to an *image ranking* module.

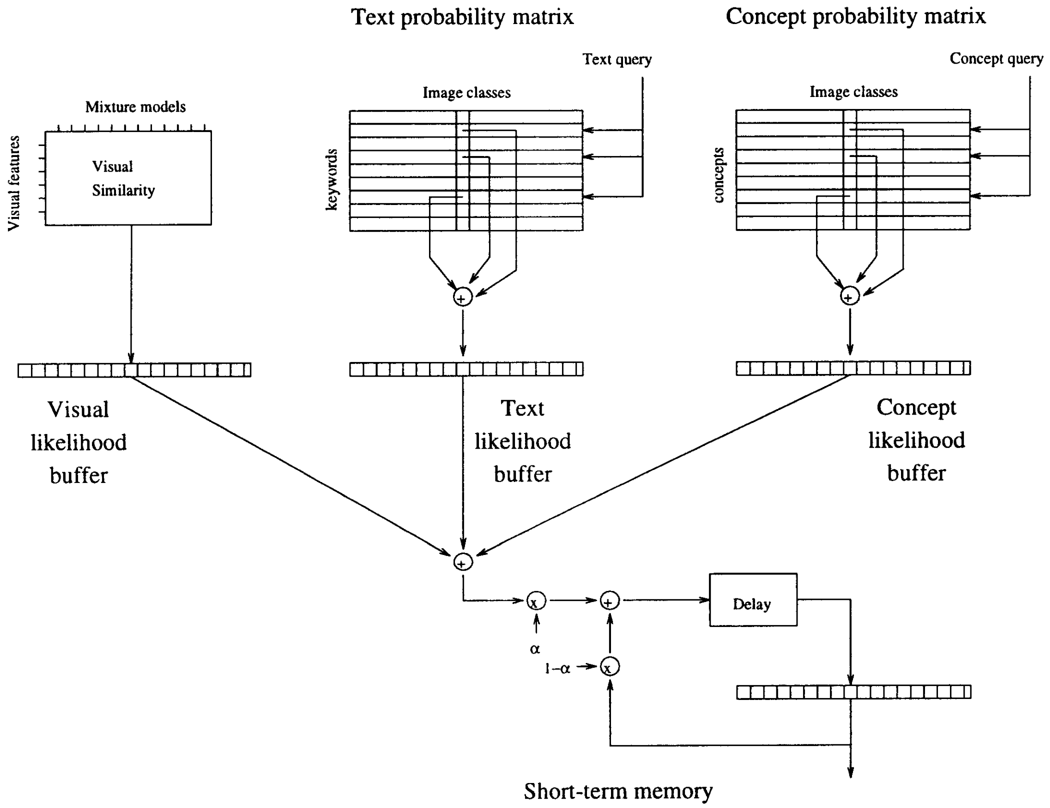


Figure 9.3: Block diagram of the image server.

As described, the process only accounts for positive examples. Extension to negative examples is straightforward, simply requiring the duplication of the four buffers and application of (7.7) instead of (7.5). As discussed in section 7.4, the main difference occurs in the image ranking module. When only positive examples are considered, this module simply orders the images by decreasing log-posterior probabilities. When negative examples are present, the top N matches according to the positive examples are then re-ordered by decreasing log posterior odds-ratio. The parameter N can be modified by the user, the default being $N = 50$.

Given the list of top matches, the image server requests the corresponding URLs from the SQL database and transmits them to the client using the CSP. The client simply displays the images and waits further instructions from the user, upon which the entire query process is repeated.

9.2 Implementation details

We conclude this chapter by discussing some implementation details of the RaBI system. This discussion is intended mostly for those interested in replicating portions or the entirety of the work described in the thesis. Although many of the details have already been mentioned in the previous chapters, we believe that it is useful to present a cohesive summary of the most important points. We concentrate on the three components of the retrieval architecture: feature transformation, feature representation, and evaluation of image similarity.

Before proceeding to describe the details of each component we note that, even though the theoretical formulation developed in the thesis is valid for any grouping of the database images into image classes, in RaBI each image is currently considered as a class by itself for the purposes of visual retrieval. In the future, we plan to investigate the impact of more sophisticated hierarchical database organization strategies in the retrieval efficiency. A preliminary step in this direction has, indeed, already been presented in [188].

9.2.1 Feature transformation

Given a database image, feature transformation currently consists of the following steps.

- Extract all 8×8 image blocks separated by d pixels in each dimension. d can be defined by the user, the default value being $d = 4$.

- Compute the DCT for each block according to (4.11).

- Vectorize the 2-D DCT array into a row vector. While this can be done in several ways, RaBI relies on the coefficient scanning mechanism defined by the MPEG [63] standard.

- If there are various color channels, interleave the corresponding vectors. For example, if there are three color channels, the first three coefficients of the interleaved vector contain the first coefficient from each of the three color channels.

- RaBI currently uses the YBR color-space defined by MPEG, but this selection has not been subject to detailed scrutiny. Better results may be possible with a perceptual color space, e.g. HSV [163].

9.2.2 Feature representation

In RaBI, feature representation is based on Gaussian mixtures with a fixed number C of classes. The default value is $C = 8$, but can be modified when the database is initialized. Several methods have been proposed in the literature to automatically determine the number of mixture components in each model. These include *minimum description length* [146], *Akaike's information criteria* [81] and the *Bayesian information criteria* [23] among others. Because these methods increase the complexity of density estimation by a significant amount, we decided to use a fixed number of mixture components. Automatic determination of the number of components may, however, be included in future versions of RaBI.

All Gaussian mixture parameters are estimated using the EM algorithm [36, 143, 13]. The implementation is fairly standard, the only details worth mentioning are the following.

- All Gaussians have diagonal covariances.

- In order to avoid singularities, a variance limiting constraint is applied. This constraint places a minimum value of 0.1 on the variance along each dimension of the feature space.

- Initialization is performed with a vector quantizer designed by the LBG algorithm, using a variation of the cell splitting method described in [58].
- In order to split a cell, 1) we compute the (diagonal) covariance matrix of all the points that land inside it, and 2) replicate its centroid, adding to the replica a perturbation in the direction of largest variance. The perturbation is the square root of this largest variance.
- Given the codebook obtained with the LBG algorithm, we compute covariances and mixing probabilities for each of the quantization cells. These, together with the centroids contained in the codebook, provide an initial estimate for the mixture parameters.
- This initial estimate is refined with EM. Since significant changes usually only occur in the first iterations, we limit these to eight.

9.2.3 Evaluation of image similarity

The evaluation of image similarity consists of the following steps.

- Given the list of query block coordinates, extract these blocks from the query image.
- Compute the DCT, vectorize, and interleave the color channels of each block as described in section 9.2.1.
- Evaluate the likelihood under each database mixture model according to (2.16).
- To prevent numerical underflow, the likelihood of each feature vector is replaced by a fixed threshold T before taking the logarithm if its value is smaller than T. The default value is $T = 10^{-300}$.

9.2.4 Vision modules

Currently, only a face detection module is implemented in RaBI. This face detector consists of a set of libraries that were provided by Henry Rowley at CMU. It is described in [148]. Since vision sensors can be applied to the images in the database off-line, RaBI is highly extensible. In fact, the system does not even need to know the implementation details of the sensors, since all that is required is the table of probabilities generated by these. We therefore expect to incorporate other vision sensors in RaBI in the future.

Chapter 10

Conclusions

10.1 Contributions of the thesis

This thesis introduced a new decision-theoretical formulation for the visual information retrieval problem. This formulation was shown to lead to 1) new insights on the retrieval problem, and 2) new guidelines for the design of practical systems. In particular, we have shown that the decision-theoretic formulation has the following appealing properties.

- Provides a unified solution to the problems of visual recognition and learning, that is optimal in the sense of minimizing the probability of retrieval error.
- Establishes a universal probabilistic language for the retrieval problem which enables the design of systems that can seamlessly integrate information from multiple content modalities.
- Establishes objective guidelines for the design of the three main components of a visual recognition architecture: feature transformation, feature representation, and similarity function.
- Unifies a significant number of recognition approaches that have been previously proposed both in terms of image representation and similarity function.
- Enables the design of systems that learn across multiple temporal scales.

These theoretical properties were shown to be of important practical consequence. In particular, we have presented new solutions to the following challenging problems.

- Joint modeling of image color and texture.
- Precise characterization of the trade-off between feature transformation and feature representation, and guidelines for the design of each of these modules.
- Unified support for local and global queries without requiring image segmentation.
- Integration of textual and visual queries.
- Decision-theoretically optimal design of short-term learning (relevance feedback) algorithms that allow fast convergence to the desired images.
- Decision-theoretically optimal design of long-term learning (concept learning) algorithms that, over time, allow personalization of the retrieval system to the preferences of the user.

These solutions were combined into a new visual recognition architecture that was experimentally shown to 1) perform well on object, texture, and generic image databases, 2) provide a good trade-off between retrieval accuracy, invariance, and complexity, 3) lead to perceptually relevant judgments of similarity, and 4) support learning through belief propagation algorithms that, although optimal in a decision-theoretic sense, are extremely simple, intuitive, and easy to implement. This recognition architecture is the basis of the RaBI image retrieval system that was designed according to the theoretical principles laid out by the thesis.

10.2 Directions for future work

Obviously, there are several interesting questions in retrieval that we could not solve, or even address, in the course of the thesis research. Some of these questions were ignored simply because of temporal constraints. Two good examples are 1) how to extend the models now used for static images to other content types, such as video or audio, and 2) how to create indexing structures compatible with Bayesian retrieval?

We would like to emphasize that one of the added, and not thoroughly discussed in the thesis, advantages of Bayesian retrieval is exactly the fact that it provides a unified solution to these questions. On one hand, probabilistic representations from the class of mixture models are among the best known for modeling speech (where hidden Markov models are predominant [140]) and there is good reason to believe that they will be equally successful for video [69]. On the other, we have already shown that probabilistic representations are amenable to the design of hierarchical descriptors that exploit the structure of the database to efficiently build indexing structures [188]. In fact, because they maintain a complete description of the conditional density of each image class at each step of the hierarchy, we have strong reason to believe that they will outperform many of the standard indexing techniques that keep only a representative vector. Hence, while indexing and extensions to other data types remain topics for future work, we believe that they will not pose major problems to Bayesian retrieval.

A more challenging question is how to incorporate spatial relationships in Bayesian retrieval. Ideally, one would like to allow not only local queries, but also queries of the type "a region similar to x *above and to the left* of a region similar to y". Theoretically, there is no fundamental difference between these and the local queries currently supported, one simply has to rely on a more sophisticated model, capable of capturing spatial dependencies *between* regions, e.g. a Markov random field. In practice, however, it usually turns out that this is more complex than predicted by the theory since inference is much more difficult, sometimes even intractable, in such models. In spite of this difficulty and the fact that, so far, we do not have a completely satisfying solution to the problem, we are convinced that Bayesian retrieval is the best framework in which to address it. The key question is how to develop models that achieve a good trade-off between the expressive power required to account for spatial relationships and complexity. Most alternative solutions that we are aware of (e.g. [93, 164]), tend to be based on heuristics that are not always easy to justify, rely on assumptions that are usually not made explicit, and lead to representations that cannot be easily extended to deal with other components of the retrieval problem.

Establishing a language to deal with spatial relationships would bring us one step closer to the holy grail of image retrieval: the automatic extraction of semantic content descriptors. This is, without question, the main challenge for the next generation of retrieval systems.

While the retrieval by *visual similarity* presented in this thesis is sufficient in some domains, a substantial number of applications require instead *semantic* retrieval, e.g. support for queries such as "pictures of a child pointing to a bird on the sky", or the "the scene where the murder takes place".

In [186, 191], we have shown that it is possible to extend the Bayesian retrieval framework introduced in this thesis to the extraction of semantic-level descriptors, and introduced a semantic classifier based on attributes such as action, type of set (man-made vs natural), presence of close-ups (commonly associated with dialog), and crowds (scenes containing a large number of people). While this classifier demonstrates the feasibility of extracting semantic information from images and video, its practical value is somewhat limited by the fact that it requires expert knowledge about the content domain where the characterization takes place. This is an expected limitation for semantic characterization since some form of regularization will always be required to disambiguate between the multiple interpretations that a given scene may have. Better understanding of the semantic content characterization problem can only be attained though a substantial amount of research in questions such as 1) which semantic attributes can and cannot be modeled and detected and 2) how generic can semantic classifiers be?

In the absence of semantic classifiers, or as a complement to these, it is imperative that retrieval systems can learn from user-interaction and become better matched to the interests of their users as time progresses. We believe we have presented convincing evidence supporting the claim that Bayesian retrieval is a natural solution to the learning problem. However, we relied on several assumptions that may not always hold. It remains to be seen how much is lost by relying on these assumptions, what would be the complexity of learning algorithms that did not rely on them, and what other forms of learning could be implemented.

Appendix A

Experimental setup

The ultimate performance test for a CBIR system is the degree to which it is found useful by its users. This is however not an easy property to test without conducting field tests with real users, and performing experiments with human subjects is usually a complex task. A significant number of subjects must be assembled for the results to be statistically significant, the experiments must be carefully designed to ensure that they do not bias the subjects to the desired responses, and only simple and relatively fast tests can be conducted if one expects to engage a large number of subjects. Furthermore, it is not possible to repeat an experiment when something goes wrong, or to modify the system parameters and try again.

The next best alternative is to define an objective criteria for performance evaluation that does not require human intervention. Because it is so much simpler than field testing, this has been the evaluation method of choice among the retrieval community. It must, however, be performed carefully if one is to avoid oversimplified scenarios that have small resemblance with reality. In this context, the two main free variables for the design of an automated testing strategy are the choice of databases and performance criteria. In this appendix, we discuss the reasons that motivated the experiments described in the thesis.

A.1 Databases

While CBIR systems are ultimately designed to deal with generic images, generic databases are not always the most suited to allow the understanding of the strengths and weaknesses of different retrieval approaches. This is due to various reasons. First, classifying a collection of generic images is usually a subjective task to which different people will give different answers. This ambiguity makes it difficult to establish the ground truth that is required for automated testing. Second, on generic databases it is difficult to determine exactly what are the properties of the recognition architecture that are responsible for particular successes and failures. E.g. one image representation may characterize better color, while other may characterize better texture, and a third may characterize better object shape, all leading to similar overall results. While combining the strengths of the three methods would lead to significantly better performance, it may be difficult to determine what those strengths are since it is difficult to tell what properties are most important for each image. Third, tests that require ground-truth for particular visual concepts (e.g. the presence of a given object in an image) can only be performed upon manual annotation of the entire database. This is particularly difficult in cases where it is not even clear what the objects of interest may be before testing begins, e.g. the evaluation of learning algorithms. Finally, because there is only a short history of evaluation with generic databases, it is difficult to compare results with previously proposed retrieval solutions.

For all these reasons, while it is imperative to, whenever possible, evaluate performance on generic databases, it is also useful to consider databases that 1) stress specific aspects of the retrieval problem, 2) have unambiguous classification ground truth, and 3) have a long usage history in the retrieval literature. This observation motivated us to, in addition to the generic Corel database of stock photography, also consider in many chapters of the thesis two specialized databases: the Brodatz texture database, and the Columbia object database. For all databases, we have also tried to identify the approaches that are known to give best retrieval results for the image properties captured by the database, implement those approaches, and compare results to those obtained with the recognition architecture now proposed. This is an unfair test in the sense that these approaches tend to be specific to the database domain (e.g. texture), while the architecture now proposed is generic. It was, nevertheless, necessary since one of the important goals of the thesis was to know how close a generic architecture can get to the performance of the specialized approaches in their domains of expertise. We next provide a more detailed description of each of the databases.

A.1.1 Standard databases

The Brodatz database is a set of images from 112 fairly homogeneous textures. Each of these 112 images was broken into 9 128 × 128 patches for a total of 1008 database entries [134]. This database was further divided into two subgroups: while the first image in each texture class was stored in a *query database* (112 total images), the remaining 896 images formed the *retrieval database*. Among the various approaches proposed in the texture recognition literature, independent tests conducted by different laboratories [134, 94, 96] have shown that the combination of 1) the coefficients of a least squares fit of the multi-resolution SAR (MRSAR) texture model [104] and 2) the Mahalanobis distance (MD) achieves the best performance on Brodatz. The performance of the MRSAR/MD combination is, therefore, usually considered a good benchmark for the evaluation of state of the art texture recognition algorithms.

The Columbia database is a set of images of 100 objects each shot in 72 different views obtained by rotating the object in $3D$ in steps of 5^o. Once again, the database was split into two subgroups, one containing the even and the other the odd images from each object. For computational simplicity, the retrieval database was further sub-sampled by four (9 views of each object separated by 40^o) and only the first image of each object was kept in the query database (100 total images). All images were converted from the original RGB to the YBR color space (as defined in the JPEG standard [128]). The Columbia database is similar in many aspects to the color database used by Swain and Ballard when they introduced histogram-based recognition [172], but significantly larger. While there as not been such an extensive evaluation of color-based retrieval techniques as in the case of texture, it is safe to say that the combination of color histograms with the histogram intersection (HI) metric [172] has so far become the de-facto standard in the area. We therefore rely on HI as a benchmark for the evaluation of color-based retrieval.

The Corel database is a set of 20,000 images from 200 generic image classes (100 images in each class). While the groupings are of a semantic nature and there are several classes which are impossible to recover by the analysis of low-level properties such as color and texture (e.g. classes containing too much variety of visual stimuli like "China" or "Egypt," broad concepts like "nature scenes" that overlap with more specific ones like "North American wild flowers," concepts that require higher-level content understanding like the "spirit of Buddha," etc.), there are also various classes characterized by an amount of visual uniformity that makes the task feasible. For our experiments we selected 16 among these ("Arabian horses," "auto racing," "coasts," "divers and diving," "English country gardens," "fireworks," "glaciers and mountains," "Mayan and Aztec ruins," "oil paintings," "owls," "land of the pyramids," "roses," "ski scenes," "religious stained glass," "tigers") leading to a total of $1,600$ images.

In order to create the query database, we randomly selected 20% of the images in each class, leaving the remaining 80% in the retrieval database. All images were converted from the original RGB to the YBR color space. Note that, even though the classes are somewhat visually uniform, there is plenty of variation within them and retrieval is significantly harder than in the case of the two previous databases. In particular, it does not suffice to use color or texture attributes alone, but representations that can account for both color and texture. Thus, in addition to MRSAR/MD and histogram intersection , we considered two such approaches: color correlograms, and linear weighting of texture and color. These are discussed in Chapter 6.

A.1.2 Artificial databases

Automated performance evaluation is particularly difficult for local queries, since these involve image segmentation and it is infeasible to manually segment all the images in the database to establish ground truth. The problem is even worse when evaluating learning algorithms because, in this case, the objects or concepts to retrieve may themselves change during learning. A feasible alternative is to construct artificial databases where the ground truth is always known. In this thesis, we pursue this alternative exactly for the evaluation of local queries and short-term learning. In particular, all experiments performed in these areas were based on two artificial databases constructed from Brodatz and Columbia.

In each case, an artificial database was created from the retrieval databases described above, by randomly selecting 4 images at a time and making a 2×2 mosaic out of them. Figure A.1 shows two examples of these mosaics. We call these image sets the *mosaic* databases. They are representative of databases whose images do not consist of a single object or visual concept but are instead a composition of different visual stimulae.

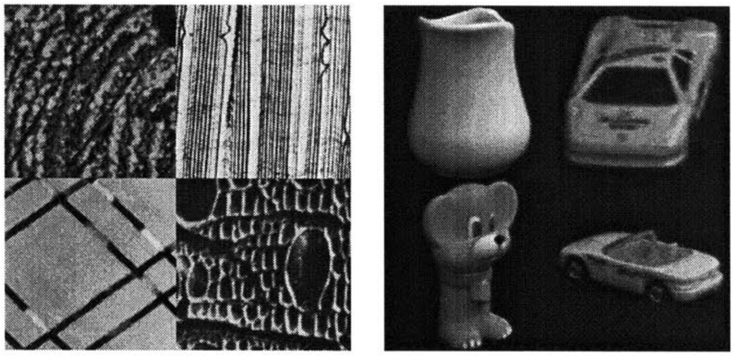


Figure A.1: Example mosaic images derived from the Brodatz (left) and Columbia databases (right).

A.2 Evaluation criteria

In addition to a database for which the classification ground truth is unambiguous, the objective evaluation of a retrieval system also requires a criteria for performance evaluation. The most commonly used criteria in the visual recognition literature is classification accuracy. This is obtained by performing several queries and measuring the fraction of these in which the top match belongs to the same class as the query. Despite the long history of its use, classification accuracy only provides a limited view of the abilities of a retrieval system. If there is an image in the database which is a close replica of the query (e.g. two pictures of the same scene taken a few minutes apart), any sensible retrieval approach will return that image as the best match. This, however, does not mean that the retrieval results are good. In fact, if the retrieval system is poorly designed and the remaining images of the same class are not near exact replicas of the query, these images may receive a very low rank in the list of returned images.

This issue has been discussed at length in the text retrieval literature, where it has long been agreed upon that a good evaluation criteria should consider more than the best match alone. The standard performance metric in text retrieval is *precision/recall* (PR), and consists of a mix of two different criteria. The basic idea is that, since users are likely to look only at the top matches, only a portion of database entries should actually be returned in response to a query[1]. *Precision* is the fraction of the returned images that are relevant to the query, and *recall* the fraction of the total number of relevant images that are returned. If $\mathbf{T}$ is the set of returned images and $\mathbf{R}$ the set of images that are relevant to the query, then

$$\mathit{precision} = \frac{|\mathbf{R} \cap \mathbf{T}|}{|\mathbf{T}|} \tag{A.1}$$

$$\mathit{recall} = \frac{|\mathbf{R} \cap \mathbf{T}|}{|\mathbf{R}|} \tag{A.2}$$

where $|\mathbf{A}|$ is the cardinality of the set $\mathbf{A}$. Since there is no optimal value for the cardinality of the set of retrieved images, results are usually presented in the form of a PR curve. Several levels of recall $\{l_1, \ldots, l_m\}$ are established and $\mathbf{T}_i$ is the smallest set of returned images that satisfy recall level l_i. Precision is then measured for each $\mathbf{T}_i$ originating a PR curve. Usually, the curve is averaged over several queries.

PR is a much more complete performance criteria than classification error, since it also provides information about the images that were not returned as the best match. For example, low precision at high recall indicates that the system has difficulty in capturing the *diversity* of the images in the class of the query. Since generalization is one of the most difficult problems in visual recognition (where a simple change of the imaging parameters, e.g. 3-D rotation, can lead to substantial changes of visual appearance), PR is a much better performance criteria than classification error for this domain. This has indeed become the prevalent view in the image retrieval community, where PR is the main tool for performance evaluation.

[1] Otherwise, users may feel overwhelmed and assume that the system is not sure about the results of the search.

Bibliography

[1] N. Ahmed, T. Nataranjan, and K. Rao. Discrete Cosine Transform. *IEEE Trans. on Computers*, pages 90–93, January 1974.

[2] P. Aigrain, H. Zhang, and D. Petkovic. Content-based Representation and Retrieval of Visual Media: A State-of-the-Art Review. *Multimedia Tools and Applications*, Vol. 3:179–202, 1996.

[3] E. Alpaydin. Soft Vector Quantization and the EM Algorithm. *Neural Networks*, 11:467–477, 1998.

[4] P. Anandan, J. Bergen, K. Hanna, and R. Hingorani. Hierarchical Model-Based Motion Estimation. In M. Sezan and R. Lagendijk, editors, *Motion Analysis and Image Sequence Processing*, chapter 1. Kluwer Academic Press, 1993.

[5] J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic. Automatic and Semi-Automatic Methods for Image Annotation and Retrieval in QBIC. In *Storage and Retrieval for Image and Video Databases*, pages 24–35, SPIE, 1995, San Jose, California.

[6] V. Athitsos, M. Swain, and C. Frankel. Distinguishing Photographs and Graphics on the World Wide Web. In *Workshop in Content-based Access to Image and Video Libraries*, pages 10–17, 1997, San Juan, Puerto Rico.

[7] 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 *International Conference on Computer Vision*, pages 675–682, Bombay, India, 1998.

[8] J. Bergen and E. Adelson. Early Vision and Texture Perception. *Nature*, 333(6171):363–364, 1988.

[9] J. Bergen and M. Landy. Computational Modeling of Visual Texture Segregation. In M. Landy and J. Movshon, editors, *Computational Models of Visual Processing.* MIT Press, 1991.

[10] J. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag New York, 1997.

[11] D. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[12] B. Bhanu, J. Peng, and S. Qing. Learning Feature Relevance and Similarity Metrics in Image Databases. In *Workshop in Content-based Access to Image and Video Libraries*, pages 14–18, 1998, Santa Barbara, California.

[13] C. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[14] J. De Bonet. Multiresolution Sampling Procedure for Analysis and Synthesis of Textured Images. In *Proc. ACM SIGGRAPH*, 1997.

[15] J. De Bonet and P. Viola. Structure Driven Image Database Retrieval. In *Neural Information Processing Systems*, volume 10, Denver, Colorado, 1997.

[16] J. De Bonet, P. Viola, and J. Fisher III. Flexible Histograms: A Multiresolution Target Discrimination Model. In E. G. Zelnio, editor, *Proceedings of SPIE*, volume 3370-12, 1998.

[17] J. Boreczky and L. Rowe. Comparison of Video Shot Boundary Detection Techniques. In *Proc. SPIE Conf. on Visual Communication and Image Processing*, 1996.

[18] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Chapman & Hall, 1993.

[19] R. Brunelli and T. Poggio. Face recognition: features versus templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, October 1993.

[20] P. Burt and E. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. on Communications*, Vol. 31:532–540, 1983.

[21] T. Caelli. A Brief Overview of Texture Processing in Machine Vision. In T. Papathomas, editor, *Early Vision and Beyond*, chapter 8. MIT Press, 1996.

[22] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, Vol. 8:679–698, 1986.

[23] B. Carlin and T. Louis. *Bayes and Empirical Bayes Methods for Data Analysis.* Chapman Hall, 1996.

[24] S. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong. A Fully Automated Content-based Video Search Engine Supporting Spatiotemporal Queries. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8(5):602–615, September 1998.

[25] R. Chellapa and A. Jain. *Markov Random Fields: Theory and Application.* Academic Press, 1993.

[26] F. Chen, L. Wilcox, and D. Bloomberg. A Comparison of Discrete and Continuous Hidden Markov Models for Phrase Spotting in Text Images. In *Proc. third International Conference on Document Analysis and Recognition*, volume 1, pages 398–402, 1995.

[27] V. Cherkassky and F. Mulier. *Learning from Data.* John Willey& Sons, NY, 1998.

[28] R. Cipolla and Editors A. Pentland. *Computer Vision and Human-Computer Interaction.* Cambridge University Press, 1999.

[29] R. Clarke. *Transform Coding of Images.* Academic Press, 1985.

[30] D. Comaniciu, P. Meer, K. Xu, and D. Tyler. Retrieval Performance Improvement through Low Rank Corrections. In *Workshop in Content-based Access to Image and Video Libraries*, pages 50–54, 1999, Fort Collins, Colorado.

[31] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley, 1991.

[32] I. Cox, M. Miller, S. Omohundro, and P. Yianilos. PicHunter: Bayesian Relevance Feedback for Image Retrieval. In *Int. Conf. on Pattern Recognition*, Vienna, Austria, 1996.

[33] G. Cross and A. Jain. Markov Random Field Texture Models. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, vol. PAMI-5, January 1983.

[34] J. Daugman. Complete Discrete 2-D GAbor Transforms by Neural Networks for Image Analysis and Compression. *IEEE Trans. on Acoust. Speech and Signal Processing*, 36(7):1169–1179, 1978.

[35] J. Daugman. Entropy Reduction and Decorrelation in Visual Coding by Oriented Neural Receptive Fields. *IEEE Trans. on Biomedical Engineering*, 36(1):107–114, January 1989.

[36] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from Incomplete Data via the EM Algorithm. *J. of the Royal Statistical Society*, B-39, 1977.

[37] H. Derin and H. Elliott. Modeling and Segmentation of Noisy and Textured Images using Gibbs Random Fields. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, vol. PAMI-9, January 1987.

[38] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, 1996.

[39] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[40] D. Dunn and W. Higgins. Optimal Gabor Filters for Texture Segmentation. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 7(4), July 1995.

[41] Y. Ephraim, A. Denbo, and L. Rabiner. A Minimum Discrimination Information Approach for Hidden Markov Modeling. *IEEE Trans. on Information Theory*, 35(5):1001–1013, September 1989.

[42] Y. Ephraim, H. Lev-Ari, and R. Gray. Asymptotic Minimum Discrimination Information Measure for Asymptotically Weakly Stationary Processes. *IEEE Trans. on Information Theory*, 34(5):1033–1040, September 1988.

[43] J. Bach et al. The Virage Image Search Engine: An open framework for image management. In *SPIE Storage and Retrieval for Image and Video Databases*, 1996, San Jose, California.

[44] M. Ortega et al. Supporting Ranked Boolean Similarity Queries in MARS. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 10(6):905–925, December 1998.

[45] M. Fleck, D. Forsyth, and C. Bregler. Finding Naked People. In *European Conference on Computer Vision*, pages 593–602, 1996.

[46] I. Fogel and D. Sagi. Gabor Filters as Texture Discriminators. *Biol. Cybern.*, 61:103–113, 1989.

[47] D. Forsyth and M. Fleck. Body Plans. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 678–683, San Juan, Puerto Rico, 1997.

[48] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[49] B. Funt and G. Finlayson. Color Constant Color Indexing. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 17(5):522–529, May 1995.

[50] Salton G., Wong A., and Yang C. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18:613–620, 1975.

[51] Salton G. and Buckley C. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24:513–523, 1988.

[52] R. G. Gallager. *Information Theory and Reliable Communication.* McGraw Hill, 1965.

[53] W. Gardner and B. Rao. Theoretical Analysis of the High-Rate Vector Quantization of LPC Parameters. *IEEE Trans. Speech and Audio Processing*, 3(5):367–376, September 1995.

[54] A. Gelb, editor. *Applied Optimal Estimation.* MIT Press, 1974.

[55] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis.* Chapman Hall, 1995.

[56] A. Gersho and R. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Press, 1992.

[57] M. Gorkani and R. Picard. Texture Orientation for Sorting Photos "at a Glance". In *Proc. Int. Conf. Pat. Rec.*, pages 459–464, 1994, Jerusalem, Israel.

[58] R. Gray. Vector Quantization. *IEEE Acoustics, Speech, and Signal Processing Magazine*, Vol. 1, April 1984.

[59] R. Gray A. Gray, G. Rebolledo, and J. Shore. Rate-Distortion Speech Coding with a Minimum Discrimination Information Distortion Measure. *IEEE Trans. on Information Theory*, vol. IT-27:708–721, November 1981.

[60] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient Color Histogram Indexing for Quadratic Form Distance Functions. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[61] R. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, Vol. 67, May 1979.

[62] R. Haralick and L. Shapiro. *Computer and Robot Vision.* Addison Wesley Longman, Inc., 1992.

[63] B. Haskell, A. Puri, and A. Netravali. *Digital Video: An Introduction to MPEG-2.* Chapman and Hall, 1997.

[64] D. Heeger and J. Bergen. Pyramid-based Texture Analysis/Synthesis. In *Proc. ACM SIGGRAPH*, 1995.

[65] N. Howe. Percentile Blobs for Image Similarity. In *Workshop in Content-based Access to Image and Video Libraries*, pages 78–83, 1998, Santa Barbara, California.

[66] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image Understanding Using Color Correlograms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 762–768, San Juan, Puerto Rico, 1997.

[67] D. Hubel and T. Wiesel. Brain Mechanisms of Vision. *Scientific American*, September 1979.

[68] F. Idris and S. Panchanathan. Storage and Retrieval of Compressed Sequences. *IEEE Transactions on Consumer Electronics*, 41(3):937–941, August 1995.

[69] G. Iyengar and A. Lippman. Models for automatic classification of video sequences. In *SPIE Storage and Retrieval for Image and Video Databases*, San Jose, 1998.

[70] G. Iyengar and A. Lippman. Clustering Images Using Relative Entropy for Efficient Retrieval. In *International workshop on Very Low Bitrate Video Coding*, Urbana, Illinois, 1998.

[71] A. Jain. A Sinusoidal Family of Unitary Transforms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1(4):356–365, October 1979.

[72] A. Jain and A. Vailaya. Image Retrieval using Color and Shape. *Pattern Recognition Journal*, 29:1233–1244, August 1996.

[73] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[74] N. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Prentice Hall, 1984.

[75] F. Jensen. *An Introduction to Bayesian Networks.* Springer-Verlag, 1996.

[76] M. Jordan. *Learning in Graphical Models.* MIT Press, 1999.

[77] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1988.

[78] M. Kearns, Y. Mansour, and A. Ng. An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering. In M. Jordan, editor, *Learning in Graphical Models*, pages 495–520. MIT Press, 1999.

[79] P. Kelly, M. Cannon, and J. Barros. Efficiency Issues Related to Probability Density Function Comparison. In *SPIE Storage and Retrieval for Image and Video Databases*, 1996, San Jose, California.

[80] P. Kelly, M. Cannon, and D. Hush. Query by Example: the CANDID approach. In *SPIE Storage and Retrieval for Image and Video Databases*, 1995, San Jose, California.

[81] G. Kitagawa, Hirotsugu Akaike, Emanuel Parzen (Editor), and Kunio Tanabe (Editor). *Selected Papers of Hirotugu Akaike.* Springer-Verlag New York, 1998.

[82] T. Kohonen. *Self Organizing Maps.* Springer Verlag, Berli, 1995.

[83] S. Kullback. *Information Theory and Statistics.* Dover Publications, New York, 1968.

[84] M. Kupperman. Probabilities of Hypothesis and Information-Statistics in Sampling from Exponential-Class Populations. *Annals of Mathematical Statistics*, 29:571–574, 1958.

[85] S. Lauritzen. *Graphical Models.* Oxford University Press, Inc, 1996.

[86] H. Lev-Ari, S. Parker, and T. Kailath. Multidimensional Maximum-Entropy Covariance Extension. *IEEE Trans. on Information Theory*, 35(3):497–508, May 1988.

[87] D. Lewis. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *European Conference on Machine Learning*, 1998.

[88] J. Li and A. Barron. Mixture Density Estimation. In *Neural Information Processing Systems*, Denver, Colorado, 1999.

[89] J. Li, N. Chadda, and R. Gray. Asymptotic Performance of Vector Quantizers with a Perceptual Distortion Measure. *IEEE Trans. on Information Theory*, 45(4):1082–1091, May 1999.

[90] Q. Li. *Estimation of Mixture Models.* PhD thesis, Yale University, 1999.

[91] Y. Linde, A. Buzo, and R. Gray. An Algorithm for Vector Quantizer Design. *IEEE Trans. on Communications*, Vol. 28, January 1980.

[92] T. Linder and R. Zamir. High-Resolution Source Coding for Non-Difference Distortion Measures: the Rate-Distortion Function. *IEEE Trans. on Information Theory*, 45(2):533–547, March 1999.

[93] P. Lipson, E. Grimson, and P. Sinha. Configuration Based Scene Classification and Image Indexing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1007–1013, San Juan, Puerto Rico, 1997.

[94] F. Liu and R. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 18(3):722–733, July 1996.

[95] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. DARPA Image Understanding Workshop*, 1981.

[96] W. Ma and H. Zhang. Benchmarking of Image Features for Content-based Retrieval. In 32^{nd} *Asilomar Conference on Signals, Systems, and Computers*, 1998, California.

[97] I. MacDonald and W. Zucchini. *Hidden Markov and Other Models for Discrete-Valued Time Series.* Chapman and Hall, 1997.

[98] J. Malik and P. Perona. Preattentive Texture Discrimination with Early Vision Mechanisms. *Journal of the OPtical Society of America*, 7(5):923–932, May 1990.

[99] S. Mallat. Multifrequency Channel Decompositions of Images and Wavelet Models. *IEEE Trans. on Acoust. Speech and Signal Processing*, Vol. 37, December 1989.

[100] S. Mallat. A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11:674–693, July 1989.

[101] M. Mandal, T. Aboulnasr, and S. Panchanathan. Image Indexing using Moments and Wavelets. *IEEE Trans. on Consumer Electronics*, 42(3):557–565, August 1996.

[102] B. Manjunath and W. Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

[103] R. Manmatha, S. Ravela, and Y. Chitti. On Computing Local and Global Similarity in Images. In *SPIE Conference on Human Vision and Electronic Imaging III*, 1998, San Jose, California.

[104] J. Mao and A. Jain. Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models. *Pattern Recognition*, 25(2):173–188, 1992.

[105] D. Marr and E. Hildreth. Theory of Edge Detection. *Proc. R. Soc. Lond.*, vol. 207:187–207, 1980.

[106] T. Matsui and S. Furui. Comparison of Text-Independent Speaker Recognition Methods Using VQ-Distortion and Discrete/Continuous HMM's. *IEEE Trans. Speech and Audio Processing*, Vol. 2(3):456–459, July 1994.

[107] T. Matsuoka and K. Shikano. Robust HMM Phoneme Modeling for Different Speaking Styles. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages I.265–I.268, 1991.

[108] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* John Willey & sons, NY, 1997.

[109] G. McLean. Vector Quantization for Texture Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23(3):637–649, May/June 1993.

[110] T. Minka and R. Picard. Interactive learning using a "society of models". *Pattern Recognition*, 30:565–582, 1997.

[111] B. Moghaddam, H. Bierman, and D. Margaritis. Defining Image Content with Multiple Regions-of-Interest. In *Workshop in Content-based Access to Image and Video Libraries*, pages 89–93, 1999, Fort Collins, Colorado.

[112] B. Moghaddam, W. Wahid, and A. Pentland. Beyond Eigenfaces: Probabilistic Matching for Face Recognition. In 3^{rd} *IEEE Int'l Conference on Automatic Face & Gesture Recognition*, Nara, Japan, 1998.

[113] B. Moghaddam and M. Yang. Gender Classification with Support Vector Machines. In 4^{th} *IEEE Int'l Conference on Automatic Face & Gesture Recognition*, 2000.

[114] J. Munkres. *Analysis on Manifolds.* Addison-Wesley, California, 1990.

[115] H. Murase and S. Nayar. Visual Learning and Recognition of 3-D Objects from Appearence. *International Journal of Computer Vision*, 14:5–24, 1995.

[116] N. Negroponte. *Being Digital.* Alfred A. Knopf, Inc, 1995.

[117] W. R. Neuman. *The Future of the Mass Audience.* NY: Cambridge University Press, 1991.

[118] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Pektovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content using color, texture, and shape. In *Storage and Retrieval for Image and Video Databases*, pages 173–181, SPIE, Feb. 1993, San Jose, California.

[119] N. Nill. A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment. *IEEE Trans. on Communications*, 33:551–557, June 1985.

[120] P. Niyogi, F. Girosi, and T. Poggio. Incorporating Prior Information in Machine Learning by Creating Virtual Examples. *Proceedings of the IEEE*, 86(11):2196–2209, November 1998.

[121] V. Ogle and M. Stonebraker. Chabot: Retrieval from a Relational Database of Images. *IEEE Computer*, Vol. 28(9):40–48, September 1995.

[122] N. Oliver, B. Rosario, and A. Pentland. A Bayesian Computer Vision System for Modeling Human Interactions. In *Workshop on Interpretation of Visual Motion, Santa Barbara, CA*, CVPR 1998.

[123] A. Papoulis. *The Fourier Integral and its Applications.* McGraw-Hill, 1962.

[124] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 1991.

[125] G. Pass and R. Zabih. Comparing Images Using Joint Histograms. *ACM Journal of Multimedia Systems*, Vol. 7(3):234–240, May 1999.

[126] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Intl. Multimedia Conference*, pages 65–73. ACM, Nov. 1996.

[127] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[128] W. Pennebaker and J. Mitchell. *JPEG: Still Image Data Compression Standard.* Van Nostrand Reinhold, 1993.

[129] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based Manipulation of Image Databases. *International Journal of Computer Vision*, Vol. 18(3):233–254, June 1996.

[130] A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically Based Shape Modeling and Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, July 1991.

[131] R. Picard. A Society of Models for Video and Image Libraries. *IBM Systems Journal*, 35(3/4):292–312, 1996.

[132] R. Picard. Digital Libraries: Meeting Place for High-Level and Low-Level Vision. In *Proc. Asian Conf. on Computer Vision*, December 1995, Singapore, USA.

[133] R. Picard. Light-years from Lena: Video and Image Libraries of the Future. In *Proc. Int. Conf. Image Processing*, October 1995, Washington DC, USA.

[134] R. Picard, T. Kabir, and F. Liu. Real-time Recognition with the entire Brodatz Texture Database. In *Proc. IEEE Conf. on Computer Vision*, New York, 1993.

[135] A. Popat. *Conjoint Probabilistic Subband Modeling.* PhD thesis, Massachussetts Institute of Technology, 1997.

[136] K. Popat and R. Picard. Cluster-based Probability Model and its Application to Image and Texture Processing. *IEEE Trans. on Image Processing*, 6(2):268–284, 1997.

[137] M. Porat and Y. Zeevi. Localized Texture Processing in Vision: Analysis and Synthesis in the Gaborian Space. *IEEE Trans. on Biomedical Engineering*, 36(1):115–129, January 1989.

[138] J. Portilla and E. Simoncelli. Texture Modeling and Synthesis using Joint Statistics of Complex Wavelet Coefficients. In *IEEE Workshop on Statistical and Computational Theories of Vision*, Fort Collins, Colorado, 1999.

[139] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical Evaluation of Dissimilarity Measures for Color and Texture. In *International Conference on Computer Vision*, pages 1165–1173, 1999, Korfu, Greece.

[140] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[141] S. Ravela and R. Manmatha. Retrieving Images by Appearence. In *International Conference on Computer Vision*, 1998, Bombay, India.

[142] S. Ravela, R. Manmatha, and E. Riseman. Image Retrieval Using Scale-Space Matching. In *European Conference on Computer Vision*, pages 273–282, 1996, Cambridge, UK.

[143] R. Redner and H. Walker. Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*, 26(2):195–239, April 1984.

[144] T. Reed and J. Hans du Buf. A Review of Recent Texture Segmentation and Feature Extraction Techniques. *Computer Vision, Graphics, and Image Processing*, Vol. 57, May 1993.

[145] D. Reynolds and R. Rose. Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE Trans. Speech and Audio Processing*, Vol. 3(1):72–83, January 1995.

[146] J. Rissanen. *Stochastic Complexity in Statistical Inquiry Theory.* World Scientific Pub. Co, Singapore, 1989.

[147] K. Rose, E. Gurewitz, and G. Fox. Vector Quantization by Determinisc Annealing. *IEEE Trans. on Information Theory*, Vol. 38, July 1992.

[148] H. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

[149] Y. Rui, T. Huang, and S.-F. Chang. Image Retrieval: Current Techniques, Promising Directions and Open Issues. *Journal of Visual Communication and Image Representation*, 10:39–62, March 1999.

[150] Yong Rui, T. 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*, 8(5):644–655, September 1998.

[151] Y. Wu S. Zhu and D. Mumford. FRAME: Filters, Random field And Maximum Entropy: Towards a Unified Theory for Texture Modeling. *International Journal of Computer Vision*, 27(2), March/April 1998.

[152] D. Sagi. The Psychophysics of Texture Segmentation. In T. Papathomas, editor, *Early Vision and Beyond*, chapter 7. MIT Press, 1996.

[153] H. Sakamoto, H. Suzuki, and A. Uemori. Flexible Montage Retrieval for Image Data. In *SPIE Storage and Retrieval for Image and Video Databases*, 1994, San Jose, California.

[154] G. Salton and J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

[155] S. Santini and R. Jain. Similarity Measures. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 21(9):871–883, September 1999.

[156] B. Schiele. *Object Recognition Using Multidimensional Receptive Field Histograms.* PhD thesis, I. N. P. Grenoble, 1997.

[157] B. Schiele and J. Crowley. Object Recognition Using Multidimensional Receptive Field Histograms. In *Proc. 4th European Conference on Computer Vision*, 1996, , Cambridge, UK.

[158] C. Schmid and R. Mohr. Local Greyvalue Invariants for Image Retrieval. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

[159] S. Sclaroff. Deformable Prototypes for Encoding Shape Categories in Image Databases. *Pattern Recognition*, 30(4), April 1997.

[160] C. Shyu, C. Brodley, A. Kak, A. Kosaka, A Aisen, and L. Broderick. Local versus Global Features for Content-Based Image Retrieval. In *Workshop in Content-based Access to Image and Video Libraries*, pages 30–34, 1998, Santa Barbara, California.

[161] P. Simard, Y. Le Cun, and J. Denker. Efficient Pattern Recognition Using a New Transformation Distance. In *Proc. Neural Information Proc. Systems*, Denver, USA, 1994.

[162] J. Simonoff. *Smoothing Methods in Statistics.* Springer-Verlag, 1996.

[163] J. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis.* PhD thesis, Columbia University, 1997.

[164] J. Smith and S. Chang. VisualSEEk: a fully automated content-based image query system. In *ACM Multimedia*, pages 87–98, 1996, Boston.

[165] J. Smith and C. Li. Decoding Image Semantics Using Composite Region Templates. In *Workshop in Content-based Access to Image and Video Libraries*, pages 9–13, 1998, Santa Barbara, California.

[166] M. Stricker and A. Dimai. Color Indexing with Weak Spatial Constraints. In *SPIE Storage and Retrieval for Image and Video Databases*, volume 2670, pages 29–40, 1996, San Jose, California.

[167] M. Stricker and M. Orengo. Similarity of Color Images. In *SPIE Storage and Retrieval for Image and Video Databases*, 1995, San Jose, California.

[168] M. Stricker and M. Swain. The Capacity of Color Histogram Indexing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 704–708, 1994.

[169] Y. Stylianou, O. Cappé, and E. Moulines. Continuous Probabilistic Transform for Voice Conversion. *IEEE Trans. Speech and Audio Processing*, Vol. 6(2):131–141, March 1998.

[170] K. Sung and T. Poggio. Example Based Learning for View-Based Human Face Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):39–51, January 1998.

[171] A. Sutter, J. Beck, and N. Graham. Contrast and Spatial Variables in Texture Segregation: testing a simple spatial-frequency channels model. *Perceptual Psychophysics*, 46:312–332, 1989.

[172] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, Vol. 7(1):11–32, 1991.

[173] Martin Szummer and Rosalind Picard. Indoor-Outdoor Image Classification. In *Workshop in Content-based Access to Image and Video Databases*, 1998, Bombay, India.

[174] H. Tamura, S. Mori, and T. Yamawaki. Texture Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 6:460–473, 1978.

[175] L. Taycher, M. Cascia, and S. Sclaroff. Image Digestion and Relevance Feedback in the Image Rover WWW Search Engine. In *Visual 1997*, San Diego, California.

[176] B. Thai and G. Healey. Spatial Filter Selection for Illumination-Invariant Color Texture Discrimination. In *IEEE Computer Society Conference on Computer Vision and Pattern recognition*, Fort Collins, Colorado, 1999.

[177] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley, 1985.

[178] M. Tuceryan and A. Jain. Texture Analysis. In C. Chen, L. Pau, and P. Wang, editors, *The Handbook of Pattern Recognition and Computer Vision*, chapter 11. World Scientific Publishing Co., 1992.

[179] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3, 1991.

[180] S. Ullman. *High-level vision :object recognition and visual cognition.* MIT Press, 1996.

[181] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. A Bayesian Framework for Semantic Classification of Outdoor Vacation Images. In *SPIE Conference on Electronic Imaging*, 1999, San Jose, California.

[182] A. Vailaya, A. Jain, and H. Zhang. On Image Classification: City vs. Landscape. *Pattern Recognition*, 31:1921–1936, December 1998.

[183] K. Valkealahti and E. Oja. Reduced Multidimensional Co-Occurrence Histograms in Texture Classification. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 20(1):90–94, January 1998.

[184] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.

[185] N. Vasconcelos. Library-based Image Coding using Vector Quantization of the Prediction Space. Master's thesis, Massachusetts Institute of Technology, 1993.

[186] N. Vasconcelos and A. Lippman. Statistical Models of Video Structure for Content Analysis and Characterization. *IEEE Trans. on Image Processing*, 9(1):3–19, January 2000.

[187] N. Vasconcelos and A. Lippman. Multiresolution Tangent Distance for Affine Invariant Classification. In *Neural Information Processing Systems*, Denver, Colorado, 1997.

[188] N. Vasconcelos and A. Lippman. Learning Mixture Hierarchies. In *Neural Information Processing Systems*, Denver, Colorado, 1998.

[189] N. Vasconcelos and A. Lippman. A Probabilistic Architecture for Content-based Image Retrieval. In *Proc. IEEE Computer Vision and Pattern Recognition Conf.*, Hilton Head, North Carolina, 2000.

[190] N. Vasconcelos and A. Lippman. Content-based Pre-Indexed Video. In *Proc. Int. Conf. Image Processing*, Santa Barbara, California, 1997.

[191] N. Vasconcelos and A. Lippman. A Bayesian Framework for Semantic Content Characterization. In *Proc. IEEE Computer Vision and Pattern Recognition Conf.*, Santa Barbara, California, 1998.

[192] N. Vasconcelos and A. Lippman. Library-based Coding: a Representation for Efficient Video Compression and Retrieval. In *Proc. Data Compression Conference*, Snowbird, Utah, 1997.

[193] X. Wan and C. Kuo. A New Approach to Image Retrieval with Hierarchical Color Clustering. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):628–643, September 1998.

[194] J. Wang and E. Adelson. Representing Moving Images with Layers. *IEEE Trans. on Image Processing*, Vol. 3, September 1994.

[195] J. Wang, G. Wiederhold, O. Firschein, and A. Wei. Content-based Image Indexing and Searching using Daubechies' Wavelets. *International Journal of Digital Libraries*, 1:311–328, 1997.

[196] A. Wilson and A. Bobick. Parametric Hidden Markov Models for Gesture Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21(9), 1999.

[197] E. Yair, K. Zeger, and A. Gersho. Competitive Learning and Soft Competition for Vector Quantizer Design. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 40(2), February 1992.

Acknowledgments

I would like to say thanks to some people which, in a way or another, made this work possible.

First, thanks to Andy Lippman, my advisor, for the good ideas and inspiration, the freedom to pursue all the ideas that interested me (and take as many courses), and an unique perspective on research and life. Few people have influenced my thinking as much as he has. I am also particularly grateful to the thesis readers, Robert Gray, Aaron Bobick, and Murat Kunt, for the detailed comments, suggestions, and encouragement. I can safely say that this document is ten times better than it would have been without their help.

Equally influential in the research that lead to the thesis were various people at MIT. Roz Picard, who first got me interested in the retrieval problem and pointed out the importance of combining recognition with learning; Michael Jordan, who introduced me to the beauty of Bayesian inference; Ted Adelson, who pointed out the importance of building models for "stuff" (which is what this thesis is mostly about); and the many others that taught most of what I know today through their courses. Outside MIT, I would like to express my gratitude to John Kender, who has been a great source of encouragement, and Frederic Dufaux, who, in addition to providing interesting professional insights, has long ago become a true friend.

One of the annoying facts of graduate school is that, sooner or later, one has to graduate. In addition to Andy, I would like to thank Mike Bove, Chris Schmandt, and Walter Bender for creating (and maintaining) the *Garden*, where I had so much fun throughout all these years. This would, of course, not have been possible without the contact with the Garden students, many of whom were my office-mates at one point or another: Giri Iyengar, Ed

Chalom, Ken Kung, Vadim Gerasimov, Shawn Becker, Henry Holtzman, Bill Butera, and Kris Popat, among others. Needless to say, nothing would have worked without the help of people like Deborah Widener, Polly Guggenheim, Kimberly Schneider, Ruth Rothstein, Gillian Galloway, and Linda Peterson.

Life is not all about work, and I was lucky enough to establish friendships with a sizeable contingent of great people. When I first arrived to the U.S., Luiz Quaresma, Rita and Paulo Guedes, and Julia and Miguel Silveira were instrumental in making the transition to the new culture very smooth. All of them are now back in Portugal, but Rita and Paulo have returned almost every year, and their visits are always great fun. With Miguel Silveira, I started the *Portuguese Students Association at MIT* which, from a group of five or six, rapidly grew into more than fifty people from several universities in the Boston area. This allowed me to establish good friendships with several people: José Camões, José Tavares, Jorge Gonçalves, José Monteiro, Miguel and Inês Castro, José Duarte, Francisco Veloso, Inês Sousa, and Hermínio Rico, among others.

One of the hardest parts of studying in a foreign country is that one tends to loose contact with old acquaintances. But even in this respect I was very lucky since two of my closest friends came to the U.S. with me. The first, Paulo Ferreira, went on to the University of Illinois where he graduated in 1996 only to come to MIT as a post-doc. I have to thank Paulo in many respects: he was the one who first challenged me to come to the U.S.; the long telephonic chats that we had while he was in Illinois were always a source of enjoyment; and when he came to Boston, with his wife Anna and daughter Verónica, we had numerous moments of great fun. I truly admire his entrepreneur spirit and the belief that no obstacles are large enough to be unbeatable. The second, Nuno Pontes, never came to Boston (other than to visit) but we still talked regularly on the phone (usually for several hours). His great sense of humor is always a delight. Finally, I was able to keep contact with most of my good friends in Portugal, mainly by getting together with them when I go back: José Cabral, Pedro and Isabel Cardoso (and my lovely god-daughter Joana), João and Dulce Mota, Fernando and Teresa Azevedo, Luís and Paula Mota.

One of the good things that happened while I was abroad was that, by getting married, I added a collection of great people to my family. My mother-in-law Hermínia who I really think of as a second mother, my brothers- and sisters-in-law Teresinha and Paulo, and Teresinha (the second) and André that I am very fond of. I am grateful for being part of their family, and for how they take great care of us when we go back to Portugal.

The only aspect of graduate life that I really hated was having to be separated from most of my family. Of course we try to make the best of it: I religiously call my mother and father every week and that makes the situation more tolerable; my mother comes to Boston as frequently as she can, and I always go back at least once a year. That trip, and the "royal treatment" that I get at home, are always the highlight of the year. Even so, there were times when it was particularly tough to be separated from them: the passing away of my grandfather Joaquim and grandmother Carmelinda, and the birth of my daughter Francisca. I will not thank my parents for anything in particular because I really owe them almost all of what I am today. Equally sad is to be separated from my brothers Pedro and Ricardo, and sister Verinha, which have always been my best friends. I truly miss the passionate soccer discussions with Pedro, the more philosophical debates with Ricardo, and giving Verinha a hard-time (specially now that she has a boyfriend, Mike). I have greatly enjoyed seeing Pedro get married with Rita, and Ricardo and Vera graduate and start their professional carriers. I am very proud of them all.

Finally, a very special word to "my girls": Manuela and Francisca. I keep trying to figure out which one of them is the best thing that happened in my life, but the answer is not easy. Manuela is the real strength behind my ability to stay abroad for so long. Thanks for the love, the care, the company, the long hours of study together, the great ideas, and the enormous amount of fun. It is hard to imagine that someone could ever be luckier than I was when I chose you to be my wife. Francisca is a ray of light continuously shining in our home, that has brought us immense amounts of joy and happiness (well, I admit that on a Saturday morning at 7:00AM it does not always sound so good). Nothing compares to the way in which she receives me at home everyday with her arms wide open and shouting "Papi, Papi". I truly feel blessed.