Appears in European Conference in Computer Vision, Dublin, Ireland, 2000.

Learning over multiple temporal scales in image databases

Nuno Vasconcelos and Andrew Lippman

MIT Media Laboratory,
20 Ames St. E15-354, Cambridge, USA
{nuno,lip}@media.mit.edu
http://www.media.mit.edu/~nuno

Abstract. The ability to learn from user interaction is an important asset for content-based image retrieval (CBIR) systems. Over short times scales, it enables the integration of information from successive queries assuring faster convergence to the desired target images. Over long time scales (retrieval sessions) it allows the retrieval system to tailor itself to the preferences of particular users. We address the issue of learning by formulating retrieval as a problem of Bayesian inference. The new formulation is shown to have various advantages over previous approaches: it leads to the minimization of the probability of retrieval error, enables region-based queries without prior image segmentation, and suggests elegant procedures for combining multiple user specifications. As a consequence of all this, it enables the design of short and long-term learning mechanisms that are simple, intuitive, and extremely efficient in terms of computational and storage requirements. We introduce two such algorithms and present experimental evidence illustrating the clear advantages of learning for CBIR.

1 Introduction

Due to the large amounts of imagery that can now be accessed and managed via computers, the problem of CBIR has recently attracted significant interest from the vision community. As an application domain, CBIR poses new challenges for machine vision: since very few assumptions about the scenes to be analyzed are allowable, the only valid representations are those of a generic nature (and typically of low-level) and image understanding becomes even more complex than when stricter assumptions hold. Furthermore, large quantities of imagery must be processed, both off-line for database indexing and on-line for similarity evaluation, limiting the amount of processing per image that can be devoted to each stage.

On the other hand, CBIR systems have access to feedback from human users that can be exploited to simplify the task of finding the desired images. This is a major departure from most previous vision applications and makes it feasible to build effective systems without having to solve the complete AI problem. In fact, 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 retrieval system cannot be completely dumb, but must know how to *integrate* all the information provided to it 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 above, 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 low-level system from the start, e.g. the system may contain a specialized face-recognition module and therefore know how to recognize faces. Hard-coded modules are justifiable only for visual concepts that are likely to be of interest to most users. Most components should instead be learned over time, as different users will need to rely on retrieval systems that are suited for their tastes and personalities. While for some users, e.g. bird lovers, it maybe important to know how to recognize parrots, others could not care less about them. Fortunately, users interested in particular visual concepts will tend to search for those concepts quite often and there will be plenty of examples to learn from. 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.

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 paper we address the issue of learning in image databases by formulating image retrieval as a problem of Bayesian inference. This new formulation is shown to have various interesting properties. First, it provides the optimal solution to a meaningful and objective criteria for performance evaluation, the *minimization of the retrieval error*. Second, the complexity of a given query is a function only of the number of attributes specified in that query and not of the total number of attributes known by the system, which can therefore be virtually unlimited. Third, when combined with generative probabilistic representations for visual information it enables region-based queries without prior image segmentation. Fourth, information from multiple user-specifications can be naturally integrated through belief propagation according to the laws of probability. This not only allows the retrieval operation to take into account multiple content modalities (e.g. text, audio, video, etc) but is shown to lead to optimal integration algorithms that are extremely simple, intuitive, and easy to implement. In result, it becomes relatively easy to design solutions for both short and long-term learning. We introduce two such mechanisms, and present experimental evidence that illustrates the clear benefits of learning for CBIR.

2 Prior work

Even though the learning ability of a retrieval system is determined to a significant extent by its image representation, the overwhelming majority of the work in CBIR has been devoted to the design of the latter without much consideration about its impact on the former. In fact, a small subset of CBIR papers addresses the learning issue altogether [1, 2, 4, 6, 7] and even these are usually devoted to the issue of short-term learning (also known as *relevance feedback*).

Two of the most interesting proposals for learning in CBIR, the "Four eyes" [4] and "PicHunter" [2] 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*. "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 target.

Because, in both of these systems, the underlying image representations and similarity criteria are not conducive to learning per se, they lead to solutions that are not completely satisfying. For example, because there is no easy way to define priors for region groupings, in [4] 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, [2] 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".

All these problems are eliminated by our formulation, where all inferences are drawn directly from the observation of the image regions selected by the user. 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, that belief and 2) prior learning follows naturally from belief propagation according to the laws of probability [5]. Since all the necessary beliefs are an automatic outcome of the similarity evaluation and all previous interaction can be summarized in 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.

3 Retrieval as Bayesian inference

The retrieval problem is naturally formulated as one of statistical classification. Given a representation space $\mathcal{F}$ for the entries in the database, the design of a retrieval system consists of finding a map

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

from $\mathcal{F}$ to the set M of classes identified as useful for the retrieval operation.

In our work, we set as goal of content-based retrieval to *minimize the probability of retrieval error*, i.e. the probability $P(g(\mathbf{X}) \neq y)$ that if the user provides the retrieval system with a query $\mathbf{X}$ drawn from class y the system will return images from a class $g(\mathbf{X})$ different than y. Once the problem is formulated in this way, it is well known that the optimal map is the Bayes classifier [3]

$$\begin{aligned} g^*(\mathbf{X}) &= \arg\max_i P(y = i|\mathbf{X}) &(1)\\ &= \arg\max_i \{P(\mathbf{X}|y = i)P(y = i)\}, &(2) \end{aligned}$$

where $P(\mathbf{X}|y = i)$ is the likelihood function for the i^{th} class and $P(y = i)$ the prior probability for this class. In the absence of prior information about which class is most suited for the query, an uninformative prior can be used and the optimal decision is the maximum likelihood (ML) criteria

$$g^*(\mathbf{X}) = \arg\max_i P(\mathbf{X}|y = i). \quad (3)$$

3.1 Probabilistic model

To define a probabilistic model for the observed data, we assume that each observation $\mathbf{X}$ is composed by A attributes $\mathbf{X} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(A)}\}$ which, although marginally dependent, are independent given the knowledge of which class generated the query, i.e.

$$P(\mathbf{X}|y = i) = \prod_k P(\mathbf{X}^{(k)}|y = i). \quad (4)$$

Each attribute is simply a unit of information that contributes to the characterization of the content source. Possible examples include image features, audio samples, or text annotations. For a given retrieval operation, the user instantiates a subset of the A attributes. While text can be instantiated by the simple specification of a few keywords, pictorial attributes are usually instantiated by example.

Borrowing the terminology from the Bayesian network literature, we define, for a given query, a set of *observed attributes* $\mathbf{O} = \{\mathbf{X}^{(k)}|\mathbf{X}^{(k)} = \mathbf{Q}^{(k)}\}$ and a set of *hidden attributes* $\mathbf{H} = \{\mathbf{X}^{(k)}|\mathbf{X}^{(k)}$ is not instantiated by the user$\}$, where $\mathbf{Q}$ is the query provided by the user. The likelihood of this query is then given by

$$P(\mathbf{Q}|y=i) = \sum_{\mathbf{H}} P(\mathbf{O}, \mathbf{H}|y=i), \tag{5}$$

where the summation is over all possible configurations of the hidden attributes[1]. Using (4) and the fact that $\sum_{\mathbf{X}} P(\mathbf{X}|y=i) = 1$,

$$\begin{aligned} P(\mathbf{Q}|y=i) &= P(\mathbf{O}|y=i) \sum_{\mathbf{H}} \prod_{k|\mathbf{X}^{(k)} \in \mathbf{H}} P(\mathbf{X}^{(k)}|y=i) \\ &= P(\mathbf{O}|y=i) \prod_{k|\mathbf{X}^{(k)} \in \mathbf{H}} \sum_{\mathbf{X}^{(k)}} P(\mathbf{X}^{(k)}|y=i) \\ &= P(\mathbf{O}|y=i), \end{aligned} \tag{6}$$

i.e. the likelihood of the query is simply the likelihood of the instantiated attributes. In addition to intuitively correct, this result also has considerable practical significance. It means that retrieval complexity 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 arbitrarily large.

In domains, such as image databases, where it is difficult to replicate human judgments of similarity it is impossible to assure that the first response to a query will always include the intended database entries. It is therefore important to design retrieval systems that can take into account user feedback and tune their performance to best satisfy user demands.

4 Bayesian relevance feedback

We start by supposing that, instead of a single query $\mathbf{X}$, we have a sequence of t queries $\mathbf{X}_1^t = \{\mathbf{X}_1, \ldots, \mathbf{X}_t\}$, where t is a time stamp. From (2), by simple application of Bayes rule, the optimal map becomes

$$\begin{aligned} g^*(\mathbf{X}_1^t) &= \arg\max_i P(y=i|\mathbf{X}_1, \ldots, \mathbf{X}_t) \\ &= \arg\max_i \{P(\mathbf{X}_t|y=i, \mathbf{X}_1, \ldots, \mathbf{X}_{t-1}) P(y=i|\mathbf{X}_1, \ldots, \mathbf{X}_{t-1})\} \\ &= \arg\max_i \{P(\mathbf{X}_t|y=i) P(y=i|\mathbf{X}_1, \ldots, \mathbf{X}_{t-1})\}. \end{aligned} \tag{7}$$

Comparing (7) with (2) it is clear that the term $P(y=i|\mathbf{X}_1, \ldots, \mathbf{X}_{t-1})$ is simply a prior belief on the ability of the i^{th} image class to explain the query. However,

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

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.

Equation (7) is, therefore, a simple but intuitive mechanism to integrate information over time. It states that the system's beliefs about 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 is then used to update these beliefs, which in turn become the priors for iteration $t+1$. I.e. prior beliefs are continuously updated from the observation of the interaction between user and retrieval system.

We call this type of behavior *short-term learning* or *in-session learning*. Starting from a given dataset (for example an image) and a few iterations of user feedback, the retrieval system tries to learn what classes in the database best satisfy the desires of the user. From a computational standpoint the procedure is very efficient since the only quantity that has to be computed at each time step is the likelihood of the data in the corresponding query. Notice that this is exactly what appears in (3) and would have to be computed even in the absence of any learning. In terms of memory requirements, the efficiency is even higher since the entire interaction history is reduced to a number per image class. It is an interesting fact that this number alone enables decisions that are optimal with respect to the entire interaction.

By taking logarithms and solving for the recursion, (7) can also be written as

$$g^*(\mathbf{X}_1^t) = \arg\max_i \left\{ \sum_{k=0}^{t-1} \log P(\mathbf{X}_{t-k}|y=i) + \log P(y=i) \right\}. \tag{8}$$

This exposes a limitation of the belief propagation mechanism: for large t the contribution, to the right-hand side of the equation, of the new data provided by the user is very small, and the posterior probabilities tend to remain constant. This can be avoided by penalizing older terms with a *decay factor* α_{t-k}

$$g^*(\mathbf{X}_1^t) = \arg\max_i \left\{ \sum_{k=0}^{t-1} \alpha_{t-k} \log P(\mathbf{X}_{t-k}|y=i) + \alpha_0 \log P(y=i) \right\},$$

where α_t is a monotonically decreasing sequence. In particular, if $\alpha_{t-k} = \alpha(1-\alpha)^k, \alpha \in (0,1]$ we have

$$g^*(\mathbf{X}_1^t) = \arg\max_i \{\alpha \log P(\mathbf{X}_t|y=i) + (1-\alpha) \log P(y=i|\mathbf{X}_1, \ldots, \mathbf{X}_{t-1})\}. \tag{9}$$

5 Combining different content modalities

So far we have not discussed in any detail what types of data can be modeled by the the attributes $\mathbf{X}_t^{(k)}$ of equation (4). Because there is no constraint for these attributes to be of the same type, the Bayesian framework can naturally integrate many different modalities. In this work we restrict our attention to the integration of visual attributes with text annotations.

Assuming a query $\mathbf{X}_1^t = \{\mathbf{T}_1^t, \mathbf{V}_1^t\}$, composed of both text ($\mathbf{T}_1^t$) and visual attributes ($\mathbf{V}_1^t$), and using (9), and (4)

$$g^*(\mathbf{X}_1^t) = \arg\max_i \{\alpha \log P(\mathbf{V}_t|y=i) + \alpha \log P(\mathbf{T}_t|y=i) + (1-\alpha)\log P(y=i|\mathbf{X}_1^{t-1})\} \tag{10}$$

Disregarding the decay factor α, the comparison of this equation with (2) reveals an alternative interpretation for Bayesian integration: 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 in the equation. I.e. 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. Similarly, the second term in the equation can be considered 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 search. Both interpretations illustrate the power of the Bayesian framework to take into account any available contextual information and naturally integrate information from different sources. We next concentrate on the issue of finding good representations for text and visual attributes.

6 Visual representations

We have recently introduced an image representation based on embedded multiresolution mixture models that has several nice properties for the retrieval problem. Because the representation has been presented in detail elsewhere [8], here we provide only a high-level description.

Images are characterized as groups of visual concepts (e.g. a picture of a snowy mountain under blue sky, is a grouping of the concepts "mountain", "snow" and "sky"). Each image class in the database defines a probability density over the universe of visual concepts and each concept defines a probability density over the space of image measurements (e.g. the space of image colors). Each image in the database is seen as a sample of independent and identically distributed feature vectors drawn from the density of one of the image classes

$$P(\mathbf{V}_t|y=i) = \sum_{j,k} P(\mathbf{v}_{t,j}|c(\mathbf{v}_{t,j})=k, y=i)P(c(\mathbf{v}_{t,j})=k|y=i), \tag{11}$$

where $\mathbf{v}_{t,j}$ are the feature vectors in $\mathbf{V}_t$ and $c(\mathbf{v}_{t,j}) = k$ indicates that $\mathbf{v}_{t,j}$ is a sample of concept k. The density associated with each concept can be either a Gaussian or a mixture of Gaussians, leading to a mixture of Gaussians for the overall density

$$P(\mathbf{V}_t|y=i) = \sum_j \sum_{c=1}^{C} \pi_c \frac{1}{\sqrt{(2\pi)^n|\mathbf{\Sigma}_c|}} e^{-\frac{1}{2}(\mathbf{v}_{t,j}-\mu_c)^T \mathbf{\Sigma}_c^{-1}(\mathbf{v}_{t,j}-\mu_c)}. \tag{12}$$

This model is 1) able to approximate arbitrary densities and 2) computationally tractable on high dimensions (complexity only quadratic in the dimension of the feature space), avoiding the limitations of the Gaussian and histogram models traditionally used for image retrieval.

When combined with a multi-resolution feature space, it defines a family of embedded densities across image resolutions that has been shown to provide precise control over the trade-off between retrieval accuracy, invariance, and complexity. We have shown that relying on the coefficients of the 8×8 discrete cosine transform (DCT) as features leads to 1) good performance across a large range of imagery (including texture, object, and generic image databases) and 2) perceptually more relevant similarity judgments than those achieved with previous approaches (including histograms, correlograms, several texture retrieval approaches and even weighted combinations of texture and color-representations) [8].

Finally, because the features $\mathbf{v}_{t,j}$ in (12) can be any subset of a given query image, the retrieval criteria is valid for both region-based and image-based queries. I.e., the combination of the probabilistic retrieval criteria and a generative model for feature representation enables region-based queries without requiring image segmentation.

7 Text representation

Given a set of text attributes $\mathbf{X} = \{X^{(1)}, \dots, X^{(T)}\}$ known to the retrieval system, the instantiation of a particular attribute by the user is modeled as a Bernoulli random variable. Defining

$$P(X^{(j)} = 1|y = i) = p_{i,j} \tag{13}$$

and assuming that different attributes are independently distributed, this leads to

$$\log P(\mathbf{T}_t|y = i) = \sum_j I_{\delta(X^{(j)}=1)} \log p_{i,j} \tag{14}$$

where $I_{x=k} = 1$ if $x = k$ and zero otherwise, and we have used (6).

7.1 Parameter estimation

There are several ways to estimate the parameters $p_{i,j}$. The most straightforward is to use manual labeling, relying on the fact that many databases already include some form of textual annotations. 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 $X^{(1)}$, the term “dogs” with $X^{(2)}$, 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, (14) will return 0 for the images containing cats and $-\infty$ for those that do not. In terms of (10) (and associated discussion in section 5) 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 probability that the user will instantiate attribute $X^{(j)}$ given that he/she is interested in images from class i?". In practice, it is usually too time consuming to define all the $p_{i,j}$ manually and not clear how to decide on the probability assignments. A better alternative is to rely on learning.

7.2 Long term learning

Unlike the learning algorithms discussed in section 4, here we are talking about *long-term learning* or *learning across retrieval sessions*. The basic idea is to let the user attach a label to each of the regions that are provided as queries during the course of the normal interaction with the retrieval system. E.g., if in order to find a picture of a snowy mountain the user selects a region of sky, he/she has the option of labeling that region with the word "sky" establishing the "sky" attribute.

Given K example regions $\{e_{j,1}, \ldots e_{j,K}\}$ of a given attribute $X^{(j)}$, whenever, in a subsequent retrieval session, the user instantiates that attribute, its probability is simply the probability of the associated examples. I.e. (14) becomes

$$\log P(\mathbf{T}_t|y=i) = \sum_j I_{\delta(X^{(j)})=1} \log P(e_{j,1}, \ldots e_{j,K}|y=i). \qquad (15)$$

The assumption here is that when the user instantiates an attribute, he/she is looking for images that contain patterns similar to the examples previously provided. Since, assuming independence between examples,

$$\log P(e_{j,1}, \ldots e_{j,K}|y=i) = \sum_k \log P(e_{j,k}|y=i) \qquad (16)$$

only the running sum of $\log P(e_{j,k}|y=i)$ must be saved from session to session, there is no need to keep the examples themselves. Hence, the complexity is proportional to the number of classes in the database times the number of known attributes and, therefore, manageable.

Grounding the annotation model directly in visual examples also guarantees that the beliefs of (15) are of the same scale as those of (12), making the application of (10) 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 solution of (15). 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, images of "New York at sunset" will only explain well the sunset examples. Thus, 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 examples.

The fact is that most image classes will only overlap partially with broad concept classes like sky. The problem can be solved by requiring the image classes to explain well only a subset of the examples. One solution is to rank the examples according to their probability and apply (15) only to the top ones,

$$\log P(\mathbf{T}_t|y=i) = \sum_j I_{\delta(X^{(j)})=1} \sum_{r=1}^{R} \log P(e_{j,k}^{(r)}|y=i), \tag{17}$$

where $e_{j,k}^{(r)}$ is the example region of rank r and R a small number (10 in our implementation).

8 Experimental evaluation

We performed experiments to evaluate the effectiveness of both short and long term learning. Because short term learning involves the selection, at each iteration, of the image regions to provide as next query it involves the segmentation of the query image. While this is not a problem for human users, it is difficult to simulate in an automated set up. To avoid this difficulty we relied on a pair of database for which segmentation ground truth is available.

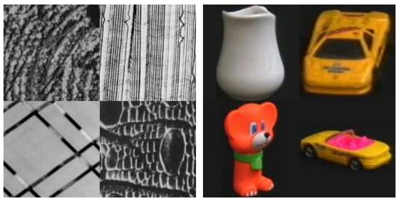

Fig. 1. Example mosaics from Brodatz (left) and Columbia (right).

These databases were created from the well know Brodatz (texture) and Columbia (object) databases, by randomly selecting 4 images at a time and making a 2×2 mosaic out of them. Each of the *mosaic databases* contains 2,000 images, two of which are shown in Figure 1. Since the individual images are of quite distinct classes (texture vs objects), testing on both Brodatz and Columbia assures us that the results here presented should hold for databases of generic imagery. All experiments were based on the DCT of a 8×8 window sliding by two-pixel increments. Mixtures of 8 (16) Gaussians were used for Brodatz (Columbia). Only the first 16 DCT coefficients were used for retrieval.

The goal of the short-term learning 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 is an iterative process where each iteration consists of selecting image regions, using them as queries for retrieval and examining the top V retrieved images. From these, the one with most sub-images in common with the target is selected to be the next query. One 8×8 image neighborhood from each sub-image in the query was then used as an example if the texture or object depicted in that sub-image was also present in the target. Performance was averaged over 100 runs with randomly selected target images.

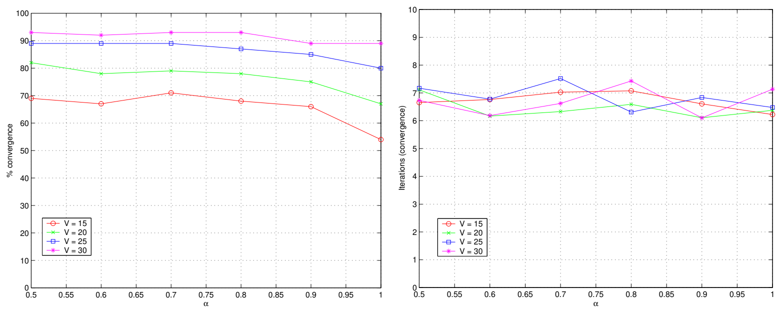

Fig. 2. Plots of the convergence rate (left), and average number of iterations for convergence (right), for the Brodatz mosaic database.

Figure 2 presents plots of the convergence rate and mean number of iterations until convergence as a function of the decay factor α and the number of matches V, for the Brodatz mosaic database (the results for Columbia are similar). In both cases the inclusion of learning ($\alpha < 1$) always increases the rate of convergence. This increase can be very significant (as high as 15%) when V is small. Since users are typically not willing to go through various screens of images in order to pick the next query, these results show that learning leads to visible convergence improvements. In general, a precise selection of α is not crucial to achieve good convergence rates. In terms of the number of iterations, when convergence occurs it is usually very fast (from 4 to 8 iterations).

Figure 3 illustrates the challenges faced by learning. It depicts a search for an image containing a plastic bottle, a container of adhesive tape, a clay cup, and a white mug. The initial query is the clay cup. Since there are various objects made of wood in Columbia and these have surface properties similar to those of clay, precision is low for this particular query: only 4 of the 15 top matches are correct (top left picture). This makes the subset of the database that is considered to satisfy the query relatively large and the likelihood that other objects in the target will appear among the top matches is low. Consequently the feedback process must be carried for three iterations before a target object, other than that in the query, appears among the top matches. When this happens, the new object does not appear in the same image as the query object.

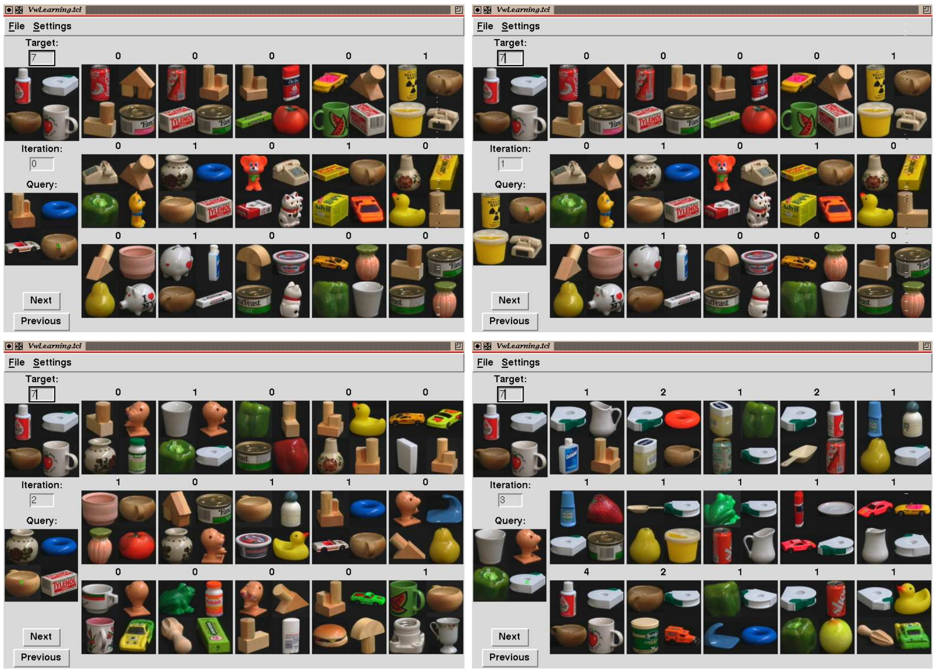

Fig. 3. Four iterations of relevance feedback (shown in raster-scan order). For each iteration, the target image is shown at the top left and the query image immediately below. Shown above each retrieved image is the number of target objects it contains.

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 right picture, the role of learning is to favor images containing these objects. In particular, 7 of the 15 images returned in response to a query based on the tape container include the clay cup or similar objects (in addition to the tape container itself). This enables new queries based on both target objects that considerably narrow down the number of candidates and, therefore, have a significantly higher chance of success. In this particular example it turns out that one of the returned images is the target itself but, even when this is not so, convergence takes only a few iterations.

8.1 Long term learning

The performance of a long-term learning algorithm will not be the same for all concepts 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 image representation, it is important to evaluate the learning of concepts from various points in the diversity/distinctiveness space.

We relied on a subset of the Corel database ($1,700$ images from 17 classes) to evaluate long-term learning and identified 5 such concepts: a logo, tigers, sky, snow and vegetation. Since common variations on a given logo tend to be restricted to geometric transformations, logos 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 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 since large smooth white surfaces are common in many scenes (e.g. clouds, white walls, objects like tables or paper).

To train the retrieval system, we annotated all the images in the database according to the presence or not of each of the 5 concepts. We then randomly selected a number of example images for each concept and manually segmented the regions where concepts appeared. These regions were used as examples for the learner. Concept probabilities were estimated for each image outside the training set using (17) and, for each concept, the images were ranked according to these probabilities. Figure 4 a) presents the resulting precision/recall (PR) curves for the 5 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 very good results, considering 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 4 b) and c) where we show the evolution of PR as a function of the number of training examples for sky and tigers. In both cases, there is a clear improvement over the one-example scenario. This is particularly significant, since this scenario is equivalent to the standard query-by-example (where users retrieve images of a concept by providing the system with one concept example). 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 in Figure 4 a) by considering more examples. On the other hand, figure 4 d) shows that more examples make a difference only when performance is limited by a poor representation of concept diversity, not distinctiveness. For snow, where the latter is the real bottleneck, more examples do not make a difference.

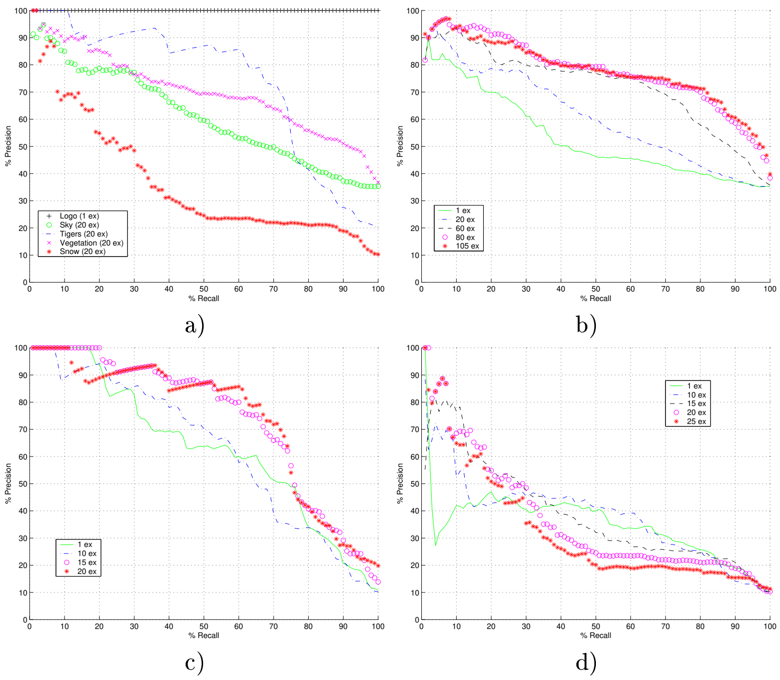

Fig. 4. Long-term learning. a) PR curves for the 5 concepts. b), c), and d) PR as a function of the training set size for sky [b)], tigers [c)], and snow [d)].

Figure 5 shows the top 25 matches for the tiger and sky concepts. It illustrates well how the new long term learning mechanism is robust with respect to concept diversity, either in terms of different camera viewpoints, shading, occlusions, etc (tiger) and variations in visual appearance of the concept itself (sky).

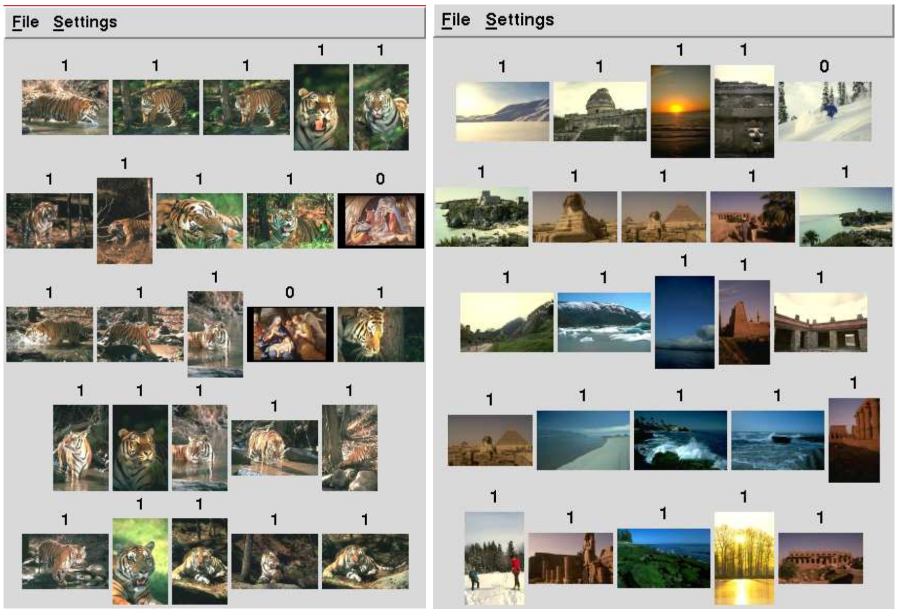

Fig. 5. Top 25 matches for the tiger (left) and sky (right) concepts.

References

1. 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.
2. 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.
3. L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, 1996.
4. T. Minka and R. Picard. Interactive learning using a "society of models". *Pattern Recognition*, 30:565–582, 1997.
5. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.
6. 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.
7. 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.
8. 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.