The Bayesian Image Retrieval System, *PicHunter*: Theory, Implementation and Psychophysical Experiments

Ingemar J. Cox[2], Matt L. Miller[3], Thomas P. Minka[3],
Thomas V. Papathomas[4], and Peter N. Yianilos[2]

Abstract— **This paper presents the theory, design principles, implementation, and performance results of *PicHunter*, a prototype content-based image retrieval (CBIR) system that has been developed over the past three years. In addition, this document presents the rationale, design, and results of psychophysical experiments that were conducted to address some key issues that arose during *PicHunter*'s development. The *PicHunter* project makes four primary contributions to research on content-based image retrieval. First, *PicHunter* represents a simple instance of a general Bayesian framework we describe for using relevance feedback to direct a search. With an explicit model of what users would do, given what target image they want, *PicHunter* uses Bayes's rule to predict what is the target they want, given their actions. This is done via a probability distribution over possible image targets, rather than by refining a query. Second, an entropy-minimizing display algorithm is described that attempts to maximize the information obtained from a user at each iteration of the search. Third, *PicHunter* makes use of *hidden annotation* rather than a possibly inaccurate/inconsistent annotation structure that the user must learn and make queries in. Finally, *PicHunter* introduces two experimental paradigms to quantitatively evaluate the performance of the system, and psychophysical experiments are presented that support the theoretical claims.**

Keywords— **Image Search, Content-Based Retrieval, Relevance Feedback, Digital Libraries, Bayesian Search.**

EDICS number: IP 4.2 (Image Search and Sorting)

I. Introduction

Searching for digital information, especially images, music, and video, is quickly gaining in importance for business and entertainment. Content-based image retrieval (CBIR) is receiving widespread research interest [1], [4], [2], [3], [7], [8], [9], [10], [11], [12], [13], [14], [6], [15], [16], [17], [18], [19], [20]. It is motivated by the fast growth of image databases which, in turn, require efficient search schemes. A search typically consists of a query followed by repeated relevance feedback, where the user comments on the items which were retrieved. The user's query provides a description of the desired image or class of images. This description can take many forms: it can be a set of keywords in the case of annotated image databases, or a sketch of the desired image [21], or an example image, or a set of values that represent quantitative pictorial features such as overall brightness, percentages of pixels of specific colors, etc. Unfortunately, users often have difficulty specifying such descriptions, in addition to the difficulties that computer programs have in understanding them. Moreover, even if a user provides a good initial query, the problem still remains of how to navigate through the database. After the query is made, the user may provide additional information, such as which retrieved images meet their goal, or which retrieved images come closest to meeting their goal. This "relevance feedback" stage differs from the query by being more interactive and having simpler interactions.

To date, there has been a distinct research emphasis on the query phase and therefore finding better representations of images. So much emphasis is placed on image modeling that relevance feedback is crude or nonexistent, essentially requiring the user to modify their query [7], [11], [17]. Under this paradigm, retrieval ability is entirely based on the quality of the features extracted from images and the ability of the user to provide a good query. Relevance feedback can be richer than this. In particular, the information the user provides need not be expressible in the query language, but may entail modifying feature weights [22] or constructing new features on the fly [23].

PicHunter takes this idea further with a Bayesian approach, representing its uncertainty about the user's goal by a probability distribution over possible goals. This Bayesian approach to the problem was pioneered by Cox et al. [3]. With an explicit model of a user's actions, assuming a desired goal, *PicHunter* uses Bayes's rule to predict the goal image, given their actions. So the retrieval problem is inverted into the problem of predicting users. Section IV describes how to obtain this predictive model.

An impediment to research on CBIR is the lack of a quantitative measure for comparing the performance of search algorithms. Typically, statistics are provided on the search length, e.g., the number of images that were visited before an image was found that was satisfactorily "similar" to a desired target image. The use of quotes around the word "similar" is deliberate; it is obvious that the search length depends on the content structure of the database and on how strict the criteria are for accepting an image as similar. In this context, searches can be classified in three broad categories:

[1]The authors' names are presented in alphabetical order. Portions of this paper were presented at various conferences, starting in 1996. The published record from these presentations is contained in references [1], [2], [3], [4], [5], [6].

[2]NEC Research Institute, 4 Independence Way, Princeton, NJ 08540.

[3]MIT Media Lab, 20 Ames St, Cambridge, MA 02139

[4]Lab. of Vision Research & Dept. of Biomedical Engineering, Rutgers University, Piscataway, N.J., 08854

Target-specific search or, simply, target search Users are required to find a specific target image in the database; search termination is not possible with any other image, no matter how similar it is to the singular image sought. This type of search is valuable for testing purposes (see section V) and occurs, for example, when checking if a particular logo has been previously registered, or when searching for a specific historical photograph to accompany a document, or when looking for a specific painting whose artist and title escapes the searcher's memory.

Category search Users search for images that belong to a prototypical category, e.g., "dogs", "skyscrapers", "kitchens", or "scenes of basketball games"; in some sense, when a user is asked to find an image that is adequately similar to a target image, the user embarks on a category search.

Open-ended search (browsing) Users search through a specialized database with a rather broad, nonspecific goal in mind. In a typical application, a user may start a search for a wallpaper geometric pattern with pastel colors, but the goal may change several times during the search, as the user navigates through the database and is exposed to various options.

The Bayesian approach described above can be adapted to accommodate all three search strategies. We focused on the target search paradigm for the reasons explained in section V.

Another advantage of having a predictive model is that we can simulate it in order to estimate how effective a particular kind of interaction will be, and thereby design an optimal interaction scheme. In section VII, we describe a novel display algorithm based on minimum entropy. This approach is evaluated by both simulation and psychophysical experiments.

Searching for images in large databases can be greatly facilitated by the use of semantic information. However, the current state of computer vision does not allow semantic information to be easily and automatically extracted.[1] Thus, in many applications, image databases also include textual annotation. Annotated text can describe some of the semantic content of each image. However, text-based search of annotated image databases has proved problematic for several reasons, including the user's unfamiliarity with specialized vocabulary and its restriction to a single language. Section VI examines this problem in more detail.

This paper presents an overview of *PicHunter*, a prototype image retrieval system that uses an adaptive Bayesian scheme, first introduced in 1996 [3], and continuously updated with improved features up to the present [1], [2], [4], [5], [6]. We present a conceptually coherent and highly expressive framework for the image retrieval problem, and report on validation of this framework using a simple system and careful experimental methods. Section II describes the theoretical basis for *PicHunter* and derives the necessary Bayesian update formulae. In order to implement the theoretical framework, it is necessary to decide upon a user interface and a model of the user. These are described in Sections III and IV. The user model is supported by psychophysical experiments that are also reported in Section IV. In order to evaluate the effectiveness of relevance feedback and a variety of other implementation issues, we introduce two experimental paradigms that are described in Section V. We also provide experimental results that evaluate the performance of *PicHunter* with and without relevance feedback. Next, in Section VI we describe how annotation can be hidden from the user yet still provide valuable semantic information to expedite the search process. Usually, the set of retrieved items that is displayed to a user is the closest set of current matches. However, such a scheme is not optimal from a search perspective. In Section VII we describe a strategy for display which attempts to maximize the information obtained from the user at each iteration of the search. Theoretical and psychophysical studies demonstrate the utility of the information maximization approach. Finally, Section VIII describes possible extensions to the *PicHunter* model, Section IX details future avenues of research, and Section X concludes with a discussion of the contributions *PicHunter* makes to CBIR research together with a discussion of broader issues.

II. Bayesian Formulation

During each iteration $t = 1, 2, \ldots$ of a *PicHunter* session, the program displays a set D_t of N_D images from its database, and the user takes an action A_t in response, which the program observes. For convenience the *history* of the session through iteration t is denoted H_t and consists of $\{D_1, A_1, D_2, A_2, \ldots, D_t, A_t\}$.

The database images are denoted $T_1, \ldots, T_n$, and *PicHunter* takes a probabilistic approach regarding each of them as a putative target.[2] After iteration t *PicHunter*'s estimate of the probability that database image T_i is the user's target T, given the session history, is then written $P(T = T_i|H_t)$. The system's estimate prior to starting the session is denoted $P(T = T_i)$. After iteration t the program must select the next set D_{t+1} of images to display. The canonical strategy for doing so selects the most likely images, but other possibilities are explored later in this paper. So long as it is deterministic, the particular approach taken is not relevant to our immediate objective of giving a Bayesian prescription for the computation of $P(T = T_i|H_t)$. From Bayes' rule we have:

$$\begin{aligned} P(T = T_i|H_t) &= \frac{P(H_t|T = T_i)P(T = T_i)}{P(H_t)} \\ &= \frac{P(H_t|T = T_i)P(T = T_i)}{\sum_{j=1}^{n} P(H_t|T = T_j)P(T = T_j)} \end{aligned}$$

That is, the *a posteriori* probability that image T_i is the target, given the observed history, may be computed by

[1] Color has proven to be an image feature with some capability of retrieving images from common semantic categories [24], [25], [26], [27], [28], [19], [29].

[2] This amounts to the implicit assumption that the target is in the database, and this is indeed the case in all of our experiments. Formulations without this assumption are possible but are beyond the scope of this paper.

evaluating $P(H_t|T = T_i)$, which is the history's likelihood given that the target is, in fact, T_i. Here $P(T = T_i)$ represents the *a priori* probability. The canonical choice of $P(T = T_i)$ assigns probability $1/n$ to each image, but one might use other starting functions that digest the results of earlier sessions.[3]

The *PicHunter* system performs the computation of $P(T = T_i|H_t)$ incrementally from $P(T = T_i|H_{t-1})$ according to:

$$\begin{aligned} P(T = T_i|H_t) &= P(T = T_i|D_t, A_t, H_{t-1}) \\ &= \frac{P(D_t, A_t|T = T_i, H_{t-1})P(D_t, T = T_i|H_{t-1})}{\sum_{j=1}^{n} P(D_t, A_t|T = T_j, H_{t-1})P(T = T_j|H_{t-1})} \\ &= \frac{P(A_t|T = T_i, D_t, H_{t-1})P(T = T_i|H_{t-1})}{\sum_{j=1}^{n} P(A_t|T = T_j, D_t, H_{t-1})P(T = T_j|H_{t-1})} \end{aligned}$$

where we may write $P(A_t|T = T_i, D_t, H_{t-1})$ instead of $P(D_t, A_t|T = T_i, H_{t-1})$ because D_t is a deterministic function of H_{t-1}.

The heart of our Bayesian approach is the term $P(A_t|T = T_i, D_t, H_{t-1})$, which we refer to as the *user model* because its goal is to predict what the user will do given the entire history D_t, H_{t-1} and the assumption that T_i is his/her target. The user model together with the prior give rise inductively to a probability distribution on the entire event space $\mathcal{T} \times \mathcal{H}^t$, where $\mathcal{T}$ denotes the database of images and $\mathcal{H}^t$ denotes the set of all possible history sequences $D_1, A_1, \ldots, D_t, A_t$. The particular user model used in our experimental instantiation of the *PicHunter* paradigm is described in section IV. Note that the user model's prediction is conditioned on image T_i and on all images that have been displayed thus far. This means that the model is free to examine the image in raw form (i.e. as pixels), or rely on any additional information that might be attached. In practice the model does not examine pixels directly but relies instead on an attached feature vector or other hidden attributes as described later in this paper.

Letting N_D denote the number of images in each iteration, our implementation assumes a space of $2^{N_D} + N_D + 1$ possible actions corresponding to the user's selection of a subset of the displayed images, or his/her indication that one of the N_D images is the target, or an "abort" signal respectively. But much more expressive action sets are possible within our framework (section IX-C).

A contribution of our work is then the conceptual reduction of the image search problem to the three tasks: 1) designing a space of user actions, 2) constructing a user model, and 3) selecting an image display strategy.

Our implementation makes the additional simplifying assumption that the user model has the form $P(A_t|T = T_i, D_t)$, i.e. that the user's action is time-invariant. Note, however, that as a consequence of our Bayesian formulation, even this simple time-invariant model leads *PicHunter* to update its probability estimate in a way that embodies all the user's actions from the very beginning of the search.

Beyond the time-invariant user models of our experiments are models that fully exploit our Bayesian formulation and adapt their predictions based on the entire history. To preserve the possibility of incremental computation we introduce the notion of user models with *state* and write the *PicHunter* update equation as:

$$P(T = T_i|H_t) = \frac{P(A_t|T = T_i, D_t, S_{t-1})P(T = T_i|H_{t-1})}{\sum_{j=1}^{n} P(A_t|T_j, D_t, S_{t-1})P(T_j|H_{t-1})} \quad (1)$$

where the model starts in some initial state S_0 and updates its state S_{t-1} to produce S_t after observing action A_t. Notice that we have said nothing of the structure of the state variable. But for efficiency's sake it makes sense to design it as a finite and succinct digest of the history H_t.

Equation 1 is, however, a fully general way to express *PicHunter* update since it spans the entire spectrum from time-invariant models where the state is trivial and constant, through models that carry forward a finite amount of state, to the original form $P(A_t|T = T_i, D_t, H_{t-1})$ where the state S_t is just H_t and grows without bound.

Finding effective models with state is an intriguing opportunity for future work within the *PicHunter* framework. We imagine that state might be used to carry forward estimates of feature relevancy, user type (e.g. expert vs. beginner), general model type (e.g. color vs. texture), and others.

III. User Interface

PicHunter uses a simple user interface designed to search for target images with minimum training. The rationale is that CBIR systems should ultimately be used as image-search tools by the general user on the World Wide Web, hence their usage should be effortless and self-explanatory. The user provides relevance feedback on each iteration of the search. The interface and user model (described in section IV) are based on *relative similarity judgments* among images, i.e. "these images are more similar to the target than the others." If all images seem dissimilar to the target, the user can select none. Many systems instead use *categorical feedback*, where the user only selects the images that are in the same category as the target [23], [16]. However, this burdens the user to decide on a useful categorization of images in a possibly unfamiliar database, and is more suited to category search (section I) than target search.

The user interface is shown in Figure 1. It consists of a small number N_D of images; in this particular implementation $N_D = 9$. The initial display is determined by the display-update algorithm. The target is always present in the display to avoid possible interference from memory problems. Of course, the target could be in the form of a traditional printed photograph, in which case the CBIR system is unaware of what the target is. The user selects zero or more images that are similar to the desired target image by clicking on them with the mouse. If users wish to

[3]The starting function must not assign probability zero to any image; otherwise the system's *a posteriori* estimate of its probability will always remain zero.

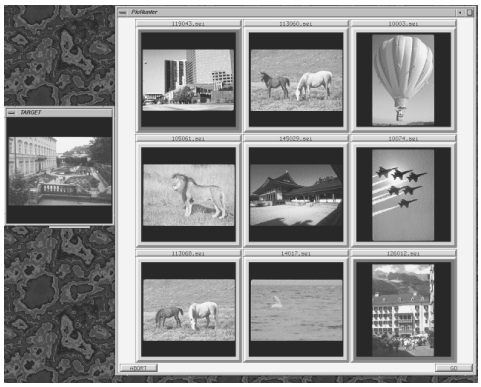


Fig. 1. *PicHunter*'s user interface.

change their selection, they can unselect images by clicking on them again; the mouse clicks function as toggles in selecting/unselecting images. As mentioned above, users can select no images if they think that all images are dissimilar to the desired target image. After users are satisfied with their selection, they hit the "GO" button to trigger the next iteration. The program then interprets their selection based on the user model, and subsequently the display-update algorithm (Section VII) decides which N_D images will be shown in the next iteration. The process is repeated until the desired image is found. When this is achieved, the user clicks the mouse button on the image identifier that is found directly above the image.

IV. User Model: Assessing Image Similarity

As explained in the previous section, the key term in the Bayesian approach is the term $P(A_t|T = T_i, D_t, U)$, where U stands for the specific user conducting the search. Assume that $D_t = \{X_1, X_2, \ldots, X_{N_D}\}$. The task of the user model is to compute $P(A_t|T = T_i, D_t, U) = P(A_t|T = T_i, X_1, X_2, \ldots, X_{N_D}, U)$, in order to update the probability that each image T_i in the database might be the target image T. The first approximation we make is that all users respond in the same way, so that the dependence on U can be dropped. This approximation is not entirely supported by our human experiments, but we believe that more complex models should be motivated by the failure of a simple one. Kurita and Kato (1993) [30] reported work in taking account of individual differences.

The second approximation is that the user's judgment of image similarity can be captured by a small number of statistical pictorial features, in addition to some semantic labels, chosen in advance. That is, it is a function of some distance measure $d(\mathbf{f}(Y), \mathbf{f}(Z))$ between the feature values $\mathbf{f}(Y) = \{f_1(Y), f_2(Y), \ldots, f_F(Y)\}$ and $\mathbf{f}(Z) = \{f_1(Z), f_2(Z), \ldots, f_F(Z)\}$ of images Y and Z.

Psychophysical experiments helped us choose the distance measure as well as the form of the probability function. Different models are compared in terms of the probability they assign to the experimental outcomes; models which assign higher probability are preferred.

When $N_D = 2$ and the user must pick an image (A_t is either 1 or 2), the probability function that we found to perform best was sigmoidal in distance (in what follows, we drop the iteration subscript, t, for simplicity):

$$P_{sigmoid}(A = 1|X_1, X_2, T) = \frac{1}{1 + \exp((d(X_1, T) - d(X_2, T))/\sigma)} \tag{2}$$

where σ is a parameter of the model chosen to maximize the probability of the data using a one-dimensional search.

When $N_D > 2$ and the user must pick $A = 1, \ldots, N_D$, a convenient generalization is the softmin:

$$p_{softmin}(A = a|X_1, \ldots, X_{N_D}, T) = \frac{\exp(-d(X_a, T)/\sigma)}{\sum_{i=1}^{N_D} \exp(-d(X_i, T)/\sigma)} \tag{3}$$

Note that transitive ordering of the images is not required by this model.

When the user can pick any number of images, including zero, no complete model has been found. One approach is to assume that the user selects each image independently according to $p_{softmin}$. Another approach is to assume that the user first decides the number k of images to select and then chooses one of the $\binom{N_D}{k}$ possible selections of k images, according to a softmin. Both approaches achieved similar probabilities for the data once their weights were tuned. This paper reports on the latter approach. Unfortunately, both give a constant probability to selecting zero images, independent of the target and the choices, which is at odds with our experimental results and limits the accuracy of our simulations.

Two possible schemes for combining multiple distance measures were considered. The first scheme multiplied the softmin probabilities for each distance measure. The second scheme simply added the distance measures before computing the softmin. In both cases, each distance measure was multiplied by an adaptive scaling factor w_i, since distance measures are generally not on the same scale. These scaling factors were set to maximizing the probability of the training data, using gradient ascent. The second model achieved a higher maximum probability, so it was chosen for the *PicHunter* experiments. The resulting formula is:

$$d(\mathbf{f}(Y), \mathbf{f}(Z)) = \sum_{i=1}^{F} w_i d_i(f_i(Y), f_i(Z)) \tag{4}$$

The individual distance d_i was the simple L1 distance between feature $f_i(Y)$ and $f_i(Z)$.

A. Pictorial Features

This subsection deals with the pictorial features that the model uses for predicting human judgment of image similarity. It must be emphasized that we used rudimentary pictorial features, because our objective was not to test features as such, but only to use them as a tool to test the Bayesian approach and the entropy display-updating scheme. Hidden semantic features are covered in section VI.

The original pictorial version of *PicHunter* [3] worked with 18 global image features that are derived for each picture in the database. These features are: the percentages of pixels that are of one of eleven colors (red, green, blue, black, grey, white, orange, yellow, purple, brown pink), mean color saturation of entire image, the median intensity of the image, image width, image height, a measure of global contrast, and two measures of the number of "edgels", computed at two different thresholds. Thus the dominant influence is that of chromatic content, in the form of the 11-bin color histogram. These features are admittedly not as sophisticated as those used in other CBIR systems, but they merely provided a starting point for experimenting with the initial system.

The current version of *PicHunter* incorporates some rudimentary information on the spatial distribution of colors, in addition to a conventional color histogram. The current version's pictorial features have the following three components: 1) HSV-HIST, a 64-element-long histogram of the HSV (Hue, Saturation, Value) values of the image's pixels. These values are obtained after conversion to HSV color space and quantization into $4 \times 4 \times 4 = 64$ color bins. 2) HSV-CORR, a 256-element long HSV color autocorrelogram at distances 1, 3, 5 and 7 pixels [24]. The pixel values are subjected to the same preprocessing as HSV-HIST. The first 64 bins are the number of times each pixel of a given color had neighbors of the same color at distance 1. The next 64 bins are for distance 3, etc. 3) RGB-CCV, a 128-element long color-coherence vector of the RGB image after quantization into $4 \times 4 \times 4 = 64$ color bins. This vector is the concatenation of two 64-bin histograms: one for coherent pixels and one for incoherent pixels. A coherent pixel is defined as one belonging to a large connected region with pixels of the same color [25].

B. Relative – Versus Absolute-Distance Criteria

Relative-distance criterion: In this scheme, the set $Q = \{X_{q1}, X_{q2}, \ldots, X_{qC}\}$ of selected images in the display D_t, as well as the set $N = \{X_{n1}, X_{n2}, \ldots, X_{nL}\}$ of non-selected images, play a role in approximating the *user-model* term $P(A_t|T_i, D_t)$ by a function S [3], [4]. The distance difference $d(T_i, X_{qk}) - d(T_i, X_{nm})$ is computed for every pair $\{X_{qk}, X_{nm}\}$ of one selected and one non-selected image. This difference determines, of course, whether T_i is closer to X_{qk} or to X_{nm}; the difference is first transformed through a sigmoid function (Equation 2 or 3), and is then applied toward computing the function S. Thus, each pair $\{X_{qk}, X_{nm}\}$ increases the probabilities of images, T_c, that are closer to X_{qk}, and decreases the probabilities of images that are closer to X_{nm} in feature space.

Absolute-distance criterion: In this scheme, only one image X_q in the display D_t can be selected by the user in each iteration. The selection of X_q either increases or decreases the probability of an image T_i, depending on whether $d(T_i, X_q)$ is small or large, respectively. In our implementation of the absolute-distance criterion, this updating of the probability $P(T = T_i)$ takes the form

$$P(T = T_i) \leftarrow P(T = T_i)G(d(T_i, X_q))$$

where $G()$ is a monotonically decreasing function of its argument. One way to think about the updating of probabilities is to visualize the selected image X_q as defining an "enhancement region" in the F-dimensional feature space, centered at $\mathbf{f}(X_q)$. The probability of each image T_i in this region is enhanced, and the magnitude of the enhancement decreases as the distance from $\mathbf{f}(X_q)$ increases. After obtaining a new value $P(T = T_i)$ for each image by multiplying it by $G()$, each value is divided by the grand total $\sum_{i=1}^{n} P(T = T_i)$, such that the ultimate values at the end of each iteration sum up to 1. This post-normalization has the effect of enhancing or depressing the probabilities of images whose feature vectors are near or far, respectively, from the selected image $\mathbf{f}(X_q)$ in feature space, independently of the magnitude of $G()$; the only requirement is that $G()$ be monotonically decreasing. The series of iterations can be visualized as a series of enhancement regions that progress toward the target from one iteration to the next, getting progressively narrower as they converge to a small region that contains the target. In this scheme, the $(N_D - 1)$ non-selected images do not influence at all the distribution of probabilities in the database. Thus, this scheme can also be referred to as a "query-by-example" search, because only one image can be selected in each iteration, providing an example for converging to the target.

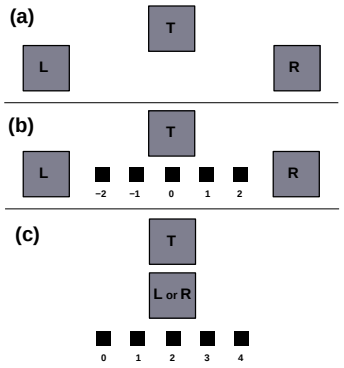


Fig. 2. The three types of displays used in the experiments. (a) The "2AFC" configuration, (b) The "relative-similarity" configuration. (c) The "absolute-similarity" configuration.

C. Experiments: Judgment of Image Similarity by Humans

This subsection deals with experiments that were designed to collect data on how humans judge image similarity, for use in developing a user model with some knowledge of human performance. In this experiment we used the three display configurations shown schematically in Figure 2. The task of the user was always the same for a given configuration, but differed across configurations.

Figure 2a shows the two-alternative forced-choice configuration, which we shall refer to simply as the "2AFC" configuration. Three images are presented on the screen: the target image on top, and two test images on the bottom. We will refer to the target, left test, and right test

images in this and similar triplet configurations as T, L, and R, respectively; collectively, the set will be referred to as the LTR triplet. The user must select the test image that he/she thinks is more similar to the target image.

The second type of display, referred to as the "relative-similarity" configuration, is shown in Figure 2b. There are now five buttons between the bottom two images. The user clicks on one of the five buttons, depending on how he/she judges the relative similarities of the two test images are with respect to the target image, using the 5-point scale. If he/she thinks that one of them is clearly more similar to the target image, he/she clicks on the corresponding extreme button (left-most or right-most). If the two test images seem to be equally similar or equally dissimilar to the target image, then the user clicks on the middle button. If one of the test images is somewhat more similar to the target image, then he/she clicks on the button immediately to the left or to the right of the center, as appropriate.

The third type of display, referred to as the "absolute-similarity" configuration, involves two images, one on top of the other, and five buttons at the bottom of the screen, as shown in Figure 2c. These buttons are used by the user to denote the degree of similarity of the two images, on a 5-point scale. The extreme left button indicates the least degree of similarity (0), and the extreme right one is used to show the maximum degree of similarity (4). If the two images have intermediate degrees of similarity, the user clicks on one of the intermediate three buttons, as appropriate.

The stimuli for this experiment consisted of a set of 150 LTR triplets, in all of which the L, T, and R images were randomly selected from a database of 4522 images. The user was presented with a sequence of trials, i.e., a sequence of randomly selected LTR triplets, and was asked to indicate his/her choices based on image similarity. Each triplet was shown in all three configurations of Figure 2, and these three displays were randomly scattered among the 600 trials (150 of type 2a, 150 of type 2b, and 300 of type 2c, i.e., 150 for LT and 150 for RT pairings). Five users took part in this experiment. They were exposed to LTR triplets for about 20 minutes before the beginning of a session, so as to accustom themselves to the variety of images in the database and the range of similarities and dissimilarities. They were told that the images they were exposed to represented a good sample of all the images in the database. This exposure would allow them to calibrate their scales of similarity [31] to produce choices that are well distributed across the entire range, and this was indeed the case with most of the users. The results from these experiments indicated that 2AFC choices correlated very well with both the relative-similarity and the differences between the absolute-similarity judgments of the same LTR triplets. The data supported the idea of using some form of distance metric, and were used for adjusting the weights of the distance function for the pictorial features of the user model (see Eq. 4).

V. Experimental Paradigm – Target Testing

The paradigm of *target testing* requires the user to find a specific target image in the database. When a user signifies that he/she has found the target, there are two possibilities: 1) If this is indeed the target, the search is terminated. 2) If the user mistakenly thinks that she/he found the target, then an appropriate message informs them of their mistake and instructs them to continue the search (the "ABORT" button is there for frustrated users who lose interest in finding the target after a lengthy search). This section presents more details on the implementation of the target testing paradigm that was used in the vast majority of our experiments. General remarks are made in subsection V-A, and specific details on the databases are presented in subsection V-B. Subsection V-C discusses two major memory schemes, and experimental results are given in the last two subsections.

A. Rationale

The main problem with evaluating the performance of CBIR systems that terminate a search when the user finds an image which is "adequately similar" to a target image is that the similarity criteria can vary from user to user. This is reflected in the data we obtained in two different search-termination strategies: one in which users terminate the search when a "similar" image is encountered, and another employing target testing. The standard deviation across users is much higher in the former case (section V-D), underlying the wide variability in judging image similarity. Thus it is very difficult to evaluate a CBIR system's performance under a category search, or a very-similar-to-target search termination scheme.

The main reason for deciding to employ target testing in *PicHunter* was precisely our belief that the use of more objective criteria of performance than category search results in more reliable statistical measures. The *performance measure* that has been used throughout our experiments is *the average number V of images required to converge to the desired specific target.* Typically, we obtained this average across 6-8 users, with each user's score averaged across searches of 10-17 randomly selected target images. This performance measure is extremely useful in two ways: 1) It provides a yardstick for comparing different *PicHunter* versions and evaluating new algorithmic ideas; 2) it is also a first step in the direction of establishing a benchmark for useful comparisons between CBIR systems, when coupled with a baseline search scheme, as explained in section V-E.

B. The Databases

The pictorial database was assembled using images from 44 Corel compact disks (CD), each containing 100 images with a common theme such as horses, flower gardens, eagles, pictures of Eskimo everyday life, scenes from ancient Egyptian monuments, etc. [32]. To these 4400 images we added 122 images from a non-thematic Corel CD for a total of 4522 images. This database was used in all versions of *PicHunter* where the user model was based exclusively

on pictorial features. In addition, we created a database of 1500 annotated images, which was a proper subset of the 4522-image set, from 15 thematic CDs. This semantic database is described in more detail in section VI-A.

C. Schemes with and without Memory

PicHunter differs from most CBIR systems along another dimension: how the user's relevance feedback is treated from the very beginning of a search. Whereas most systems tend to concentrate on the user's action only in the previous iteration, *PicHunter*'s Bayesian formulation empowers it with "long-term" memory: all the user's actions during a target search are taken into consideration. Nevertheless, the benefit of such memory has not been demonstrated experimentally. It is conceivable that performance gains from the inclusion of memory may depend on other conditions. Investigating such dependencies was the purpose of the experiments presented in section V-D.

D. Experiments on Features, Distance, and Memory Schemes

All the experiments reported in this paper were conducted with the color images displayed on 1280×1024-pixel monitor screens, measuring 38 cm by 29 cm, viewed from a distance of about 70 cm. The programs ran on Silicon Graphics Indigo2 workstations. Individual images were either in "portrait" or in "landscape" format, and were referred to by their unique identification number. They were padded with dark pixels either horizontally or vertically to form square icons that measured 7.25×7.25 cm. All users tested perfect for color vision, scoring 15/15 on standard Ishihara test plates. All users were also tested for acuity, and found to have normal or corrected-to-normal visual acuity.

This set of experiments [5], [6] was designed to study the role of the following components: 1) memory during the search process; 2) relative-distance versus absolute-distance judgment of image similarity (section IV-B); 3) semantic information (section VI). Toward this goal, we tested six versions of *PicHunter*, which we code with trigraphs XYZ for mnemonic reasons. The letters in the trigraphs XYZ refer to components 1-3 above, in that order. Thus, the first letter X refers to memory: M or N denote that the algorithm did or did not use memory, respectively, in the search process. **M** refers to the standard Bayesian system of section II. **N** refers to a system that bases its actions on the user's relevance feedback for only the last display. The second letter Y, referring to distance, can be either **R** or **A** to denote whether the model used relative or absolute distances, respectively. Finally, the last letter Z is devoted to semantic features, and it can have three possible values: **P**, or **S**, or **B** denote, respectively, that only pictorial features, or only semantic features, or both, are used in the user model for predicting judgments of image similarity. The pictorial features in these experiments were the 18 features described in section IV-A. All the experiments of this section were run with algorithms that used the most-probable display-updating scheme of section VII-A. Our previous experience indicates that some XYZ combinations are of little practical value, thus we concentrated on the following six versions:

1. MRB: uses memory, relative distance, both semantic and pictorial features.
2. MAB: same as MRB, but with absolute distance.
3. NRB: same as MRB, but doesn't use memory.
4. NAB: same as MAB, but doesn't use memory.
5. MRS: same as MRB, but uses only semantic features.
6. MRP: same as MRB, but uses only pictorial features.

Six first-time *PicHunter* users, naive as to the experimental purposes, participated in this study. They ran the experiment in a 6-users×6-versions Latin-square design [33]. Each user went through 15 target searches, terminating the search under the target testing paradigm; all searches terminated successfully. The results of these experiments are shown in Table I. The first row has the average number V of 9-image displays visited before convergence to the target; smaller values of V denote better performances. The second row displays the standard error SE, and the third row shows the ratio SE/V, as a measure of the variability of V across users. Two experienced users also ran the experiments under the same conditions. Their averages are shown below the data for the naive users.

Version	MRB	MAB	NRB	NAB	MRS	MRP
No. displays, V	25.4	35.8	45.5	33.2	15.6	35.1
Standard Error, SE	2.35	2.37	2.48	2.44	1.76	2.11
Variability, SE/V	.093	.066	.055	.073	.113	.060
V, 2 exper. users	13.1	31.6	28.4	22.2	8.8	18.9

TABLE I

THE RESULTS OF THE EXPERIMENT WHICH WAS DESIGNED TO TEST THE ROLES OF MEMORY, DISTANCE METRIC, AND SEMANTIC FEATURES IN *PicHunter*. THE EXPECTED VALUE OF V UNDER RANDOM SEARCH IS $(1{,}500/2)/9 = 83.3$. IN THIS, AS WELL AS IN TABLES II, IV, AND V, SMALLER VALUES OF V SIGNIFY BETTER PERFORMANCES.

The following main trends can be observed in the data: 1) When one compares the results of the MRB and the MRS schemes, performance with the semantics-only features (MRS) is substantially better than with the semantics-plus-pictorial features (MRB). This is just the opposite of the expected behavior; namely, if the pictorial features were well chosen, their inclusion should improve, rather than worsen, performance (even if semantic features dominate in judgments of similarity, the addition of pictorial features should at worst keep performance the same). One obvious conclusion is that the 18 features of *PicHunter*'s original version need to be refined, and this is precisely what was done in the most recent version (see section IV-A). 2) The clear advantage of the MRS version over all others underscores the role played by semantic features in the search process. This fact is also corroborated by the experimental data of sections VI-B and VII-E. 3) Pair-wise comparison of versions MRB to NRB and MAB to NAB show that the effect of memory depends on the distance criterion. The former comparison indicates

that *memory improves the relative*-distance version, while the latter comparison shows that *memory slightly worsens the absolute*-distance version. This apparent paradox can be explained if one visualizes the search in the absolute-distance version as an enhancement region that moves toward the target across the iterations. Since probabilities are updated by multiplying factors cumulatively in long-memory versions, this memory adds a delay by introducing "inertia", due to the effect of all the previous iterations. By contrast, this accumulation is helpful in the the relative-distance versions, in which the target is approached as the feature space is successively partitioned in each iteration [5]. 4) Other than the optimal scheme MRS, the next best one is the MRB scheme, which incorporates memory, a relative-distance measure, and both kinds of features; all other schemes perform somewhat worse than the two best schemes. 5) As expected, the experienced users were substantially more efficient than the inexperienced ones.

E. Target and Baseline Testing as a Benchmark for Comparing CBIR Systems

As argued earlier, there is a great need for a benchmark for comparing CBIR systems. Such a benchmark can also be used for assessing the value of incorporating a new approach for a specific system, by comparing the new version's performance against that of the original version. Ideally, one hopes for an automated comparison, but this is not feasible at the present. Hence, our efforts must be focused on producing a benchmark, based on efficient experiments with as few human users as possible. The benchmark must yield a robust estimate of performance that is representative of performances of the population as a whole. In this section we describe such a scheme based on the target testing paradigm. Our experimental results tend to confirm our intuition, and in this sense are not surprising. But such confirmation is valuable in guiding the development of complex systems that interact with humans.

To be able to compare performances with systems that search for a similar-category image, rather than a unique image target, we need to establish a performance baseline against which to compare other versions. Such a baseline is provided by a similar-target search, with a random display update, since it is reasonable to determine what the performance would be in the complete absence of any relevance feedback from the user. This motivated the present set of experiments, that were conducted with six first-time *PicHunter* users, who were naive as to the purposes of the experiment [6]. These users were the same as those who participated in the experiments of section V-D. We have just introduced a new option, namely whether searches are terminated under target testing (T), or under "category" search (C), in which an image similar to the target is found. Thus MRB/T and MRS/T denote the same target-specific versions of *PicHunter* that were referred to as MRB and MRS, respectively, in section V-D. Similarly, MRB/C is the MRB version that terminates searches when a similar image is found. In addition to MRB/T, MRS/T and MRB/C, the fourth scheme that we experimented with was RAND/C. RAND indicates that displays are updated at random, independently of the user's feedback, with the only restriction of not displaying images repeatedly, if they were already displayed in previous iterations.

The first three rows in Table II below are the results with searches by these four schemes for the six naive users, each searching for the same 15 target images. In the XYZ/C searches, users were instructed to terminate the search when they encountered an image which looked similar to the target image. The entries of the Table follow the same convention as that of Table I. Namely, the first row shows the mean number V of 9-image displays required to converge to the target, averaged across the means of 6 users, where each user's performance was averaged across the 15 targets. The Table also includes the standard error SE, as well as the ratio SE/V, which is a measure of the relative variability of V across users. The last row has the averages V of the same two experienced users who also ran the experiments of section V-D. The entries for columns MRB/T and MRS/T are duplicated from Table I.

Version	MRB/T	MRS/T	MRB/S	RAND/C*
No. displays, V	25.4	15.6	12.2	19.7
Standard Error, SE	2.35	1.76	2.13	6.39
Variability, SE/V	0.093	0.113	0.175	0.324
V, 2 exper. users	13.1	8.8	8.9	20.1

TABLE II

THE RESULTS OF THE EXPERIMENT WITH TARGET SEARCH AND CATEGORY SEARCH. THE EXPECTED VALUE OF V UNDER RANDOM SEARCH IS 83.3. THE ASTERISK ON RAND/C IS MEANT TO INDICATE THAT THIS IS NOT A VERSION OF THE *PicHunter* CBIR SYSTEM.

The following observations can be drawn from the data of Table II. 1) RAND/C converged rather fast to a picture that the average user judged to be similar to the target, establishing a high baseline standard. This makes it necessary to revisit results given in other reports where similar images are retrieved, but no baseline is established. 2) Despite this high standard, performance with the corresponding *PicHunter* scheme MRB/C is substantially better. 3) Variability in the baseline scheme RAND/C is markedly higher by a factor of 1.85 than that in MRB/C, which in turn is higher than that of the MRB/T scheme by a factor of 1.88. Since low variability allows efficient tests with few users, target search offers a valuable testing paradigm for getting representative performance data. 4) One must remark on the solid performance of the semantics-only target-search MRS/T version, which is comparable to the category-search MRB/C version, and better than the baseline. 5) Again, as expected, the performance of the experienced users was considerably better that of the naive ones, with the notable, but expected, exception of the random category search.

VI. Hidden Annotation

Systems that retrieve images based on their *content* must in some way codify these images so that judgments and

inferences may be made in a systematic fashion. The ultimate encoding would somehow capture an image's semantic content in a way that corresponds well to human interpretation. By contrast, the simplest encoding consists of the image's raw pixel values. Intermediate between these two extremes is a spectrum of possibilities, with most work in the area focusing on *low-level features*, i.e. straightforward functions of the raw pixel values (see [34], [21], [7], [35], [11], [12], [36], [17] and many others [37], [38], [39], [19]). Some such features, such as color, begin to capture an image's semantics, but at best they represent a dim reflection of the image's true meaning. The ultimate success of content-based image retrieval systems will likely depend on the discovery of effective and practical approaches at a much higher level. In this section we report conceptual and experimental progress towards this objective.

Any attempt to codify image semantics inevitably leads to design of a language with which to express them. If a human operator is required to formulate a query using this language, and interpret a database image's description in terms of the language, two serious problems arise. First, the language must not only be effective in theory, but must also serve as a natural tool with which a human can express a query. Second, inaccurate or inconsistent expression of each database image in terms of the language can lead to confusion on the part of the user, and ultimately undermine the effectiveness of, and confidence in, the system. The need for accurate and consistent expression can also limit the language's design.

For these reasons we are led to study *hidden languages* for semantic encoding, and in particular hidden boolean attributes affixed to each database image.

A. Annotation Implementation

In an effort to characterize how CBIR performance is enhanced by the introduction of semantic cues, we created an annotated database of 1,500 images from 15 thematic CDs of 100 images each. A set of approximately 138 keywords was identified by one of the authors who had extensive exposure to our experimental database of 1,500 images taken from the Corel database [32]. The objective was to obtain a set of keywords that covered a broad spectrum of semantic attributes. Each image was then visually examined and all relevant keywords identified. An additional set of *category* keywords were then assigned automatically. For example, the "lion" attribute causes the category attribute "animal" to be present. Altogether there are 147 attributes. These supplement the pictorial features used by the basic *PicHunter* version, and described in [2]. The 147 semantic attributes are regarded as a boolean vector, and normalized Hamming distance combines their influence to form, in effect, an additional *PicHunter* feature. Table III shows representative semantic labels and suggests the level of semantic resolution. It must be emphasized that these semantic features are hidden: users are not required to learn a vocabulary of linguistic terms before using the system, or even use a particular language.

sky	cloud	ground
tree	one subject	aircraft
horse	two subjects	person
water	many subjects	lion
snow	sand	animal
rodent	arch	church
bicycle	field	shoe
Japan	Africa	woods
art	painting	umbrella
city	boat	night
interior	wall	autumn
mountain	close up	green grass
eagle	child	house
fish	pillar	texture

TABLE III
REPRESENTATIVE SEMANTIC LABELS IN THE ANNOTATED DATABASE

B. Experiments: Hidden Annotation and Learning

These experiments were designed to compare performances between the original pictorial-feature version of *PicHunter* [3] with a version that incorporated semantic features in addition to the image features. Furthermore, we examined whether user performances improved after they were explicitly taught which particular features were considered important by the algorithm's user model in both versions [1]. For notational purposes, we refer to the pictorial version as "P" and to the pictorial-*plus*-semantic version as "B" (B stands for *both*). The experiments involved eight first-time *PicHunter* users who were not aware of the purposes of the study. All sessions involved searches of a target image among the 1,500 images in the database. There were a total of 17 target images that were selected randomly. Users were required to locate all 17 targets in one session for each *PicHunter* version. Both the P and the B versions were implemented with displays of nine images.

The experiments consisted of two major phases, each using the same 17 target images. In the first phase, the pre-explanation phase, users were told to use their own similarity criteria. The order of exposure was balanced: four users went through sequence (P,B), and the others through (B,P). The eight users were then divided in two groups of four, to balance within-group average performances and standard deviations for the two groups. This grouping was done on the basis of their performances in the first phase, and it was constrained by requiring that each group have two members that went through the (P,B) sequence, and the other two through the (B,P) sequence. In the second phase, users were first given explicit instructions for judging image similarity, according to the user model. For the P model, we briefly explained to them the 18 features and their relative weights, and instructed them to ignore the images' semantic contents. For the B model, users were told to base similarity not only on image characteristics, but also on image semantics; they were shown the 42 words of Table III, to get an idea of how the B version was designed. This explanation was very brief, lasting at most 8 minutes

for each of the two versions. Explanations were given separately for each version, and users started the 17-target search with that particular version of *PicHunter*. This was followed by explanations for the other version, and ended with a 17-target search with that other version. The order of versions, (P,B) or (B,P), was balanced in this second phase, as well.

The results are given in Table IV, the entries of which are the mean number V of 9-image displays that were required for users to locate the target, averaged across the 8 users and the 17 targets. It is obvious from the entries of Table IV that both semantic features and training, in the form of explanations, improve users' performance. Specifically, the data indicate that: 1) Without prior instruction, users took on average about 1/3 fewer displays to converge to the target with the B version than with the P version, underlying the importance of semantics. 2) After users were instructed on the similarity criteria, performance improved for both versions, as expected. Users took over 25% more displays prior to instruction, when their performance is pooled over both versions of *PicHunter*. 3) In the P version the explanations reduced the search time to 77.8% of its original level; in the B version the search time was reduced to 81.2% of its original level. A 2×2 within-groups analysis of variance (ANOVA) was performed over both the version type and the instruction presence to look for an interaction between the two effects listed above. No such interaction was found ($F = 1.770; df = 1, 7; p = .225$). This shows that the instruction helped users equally with both versions.

	Before explanations	After explanations
Pictorial features only	17.1	13.2
Pictorial AND semantic	11.7	9.5

TABLE IV

THE EFFECT OF SEMANTICS AND EXPLANATIONS ON PERFORMANCE. THE EXPECTED VALUE OF ENTRIES UNDER RANDOM SEARCH IS 83.3.

Also, the issue of feature relevancy must be addressed. In observing the 8 users' strategies, we observed that test images were sometimes selected because of similarity with the target in terms of, say, color ("it has as much blue as the target"), and other times because of similarity in, say, overall brightness. To the extent that a user relies on a small number of features during a session, it may be possible to learn which are being used, and in so doing improve performance. This is in principle possible using user models with state as described in section II.

Because the attributes are hidden in our approach, we are free to consider attribute schemes in future work that might not work well in a traditional non-hidden approach. We might, for example, entertain a scheme that employs 10,000 attributes, far more than a human operator could reasonably be expected to deal with. Moreover, some of these attributes might correspond to complex semantic concepts that are not easily explained, or to overlapping concepts that do not fit well into the kind of hierarchies that humans frequently prefer. They might even include entirely artificial attributes that arise from a machine learning algorithm. Because the attributes are hidden, it may be that the system performs well despite considerable *error* in the assignment of attributes. For this reason we are free to consider attributes even if their proper identification seems very difficult.

We remark that there are errors and inconsistencies even in attributes assigned by humans. Here, the fact that the attribute values are hidden can result in more robust performance in the presence of error. We also observe that in some settings, such as the emerging area of Internet Web publication, authors are implicitly annotating their images by their choice of text to accompany them. Exploiting this textual proximity represents an immediate and interesting direction for future work and this general direction is explored in [27], [40]. Semantically annotated images are also appearing in structured environments such as medical image databases, news organization archives – and the trend seems to extend to generic electronic collections. In addition to using these annotations in a hidden fashion, mature image search systems may be hybrids that include an explicit query mechanism that corresponds to the space of available annotations. Even in query-based systems, learning may play a role as illustrated by related work in the field of textual information retrieval [41].

It is not clear how *high* in the semantic sense our approach of hidden attributes might reach. It is certainly conceivable that a large portion of an image's semantic content might be captured by a sufficiently large and rich collection of attributes – entirely obviating the need to produce a single succinct and coherent expression of an image's meaning.

VII. Display Updating Model

Once the user-model module of *PicHunter* updates the probability distribution across the entire database, the next task is to select the N_D images to be shown in the next display. We have experimented with several schemes in this area, but we report on the two that produced the best results: the most-probable scheme, and the most-informative scheme.

A. Most-Probable Display Updating Scheme

This is an obviously reasonable strategy: For the next display, choose the N_D images that possess the highest probabilities of being the target; possible ties are broken with random selections. This is the scheme that was used in all but the most recent version of *PicHunter*. It performed quite well, achieving search lengths that were about ten times better than random target-testing searches for purely picture-based features [2], [3]. Typically, this updating scheme produces displays whose images belong to a common theme, such as aircraft or horses, even with the

purely pictorial feature user model, somehow exhibiting an ability to extract semantic content. However, this greedy strategy suffers from an over-learning disadvantage that is closely related to its desired ability to group similarly looking images. The problem is that, in a search of, say, an image of a jungle scene, *PicHunter* occasionally "gets stuck" by showing display after display of, say, lion pictures as a result of the user having selected a lion picture in an earlier display. This problem is addressed by the information-based scheme, described below.

B. Most-Informative Display Updating Scheme

Another approach is to attempt to minimize the total amount of iterations required in the search. The result is a scheme which tries to elicit as much information from the user as possible, while at the same time exploiting this information to end the search quickly.

At any time during the search, all of the knowledge *PicHunter* has about the target is concisely summarized by the distribution $P(T = T_i)$ over the database $\{T_1, T_2, \ldots, T_n\}$. The idea is to estimate the number of iterations left in the search, based on the distribution $P(T = T_i)$. Call this estimate $C[P(T)]$. Then the display scheme chooses the display which minimizes the expected number of future iterations, which is

$$C(X_1, \ldots, X_{N_D}) = P(\text{target not found}) \sum_a C[P(T|A = a)]P(A = a|X_1, \ldots, X_{N_D})$$

where

$$P(A = a|X_1, \ldots, X_{N_D}) = \sum_{i=1}^{n} P(A = a|X_1, \ldots, X_{N_D}, T = T_i)P(T = T_i)$$

and

$$P(\text{target not found}) = 1 - P(T = X_1) - \ldots - P(T = X_{N_D})$$

and $p(T|A = a)$ is the distribution over targets after user response a.

Information theory suggests entropy as an estimate of the number of questions one needs to ask to resolve the ambiguity specified by $P(T = T_i)$:

$$C[P(T)] \approx -\alpha \sum_{i=1}^{n} P(T = T_i) \log P(T = T_i) \qquad (5)$$

for some positive constant α which is irrelevant for the purpose of minimization. This offers an alternative interpretation of minimizing future cost: maximizing immediate information gain.

To illustrate this scheme, consider an ideal case when $N_D = 2$:

$$P_{ideal}(A = 1|X_1, X_2, T) = \begin{cases} 1 & \text{if } d(X_1, T) < d(X_2, T) \\ 0.5 & \text{if } d(X_1, T) = d(X_2, T) \\ 0 & \text{if } d(X_1, T) > d(X_2, T) \end{cases}$$

If $A = 1$, all elements farther from T than X_1 will get zero probability. The remaining elements will have uniform probability (assuming no ties). The most-informative display updating scheme will therefore choose X_1 and X_2 so that the expected number of remaining elements is minimum. This minimum is achieved when the decision boundary $d(X_1, T) = d(X_2, T)$ exactly divides the set of targets in half. So in this idealized situation the most-informative display updating scheme behaves like the *vantage-point tree* algorithm of Yianilos [42], which is a kind of binary search on an arbitrary metric space.

Now consider the generalization

$$P_{sigmoid}(A = 1|X_1, X_2, T) = \frac{1}{1 + \exp((d(X_1, T) - d(X_2, T))/\sigma)}$$

When $\sigma \to 0$, this is the same as P_{ideal}. When $0 < \sigma < \infty$, there is a smooth transition from probability 1 to probability 0 as T varies. When $\sigma \to \infty$, outcomes are completely random. This formula can be interpreted as P_{ideal} after corrupting the distance measurements with Gaussian noise. The parameter σ can therefore be interpreted as the degree of precision in the distance measurements.

Unfortunately, finding $X_1, \ldots, X_{N_D}$ to minimize $C(X_1, \ldots, X_{N_D})$ is a non-trivial task. An incremental approach in N_D does not seem possible, since an optimal display for $N_D - 1$ can be far from an optimal display for N_D. The problem is at least as hard as vector quantization, which we know can only be solved approximately by local search algorithms. Local search does not seem feasible here, since evaluating C is quite costly and there can be many local minima. One needs an optimization scheme which can give decent results with a small number of evaluations. Inspired by Yianilos's vantage-point tree algorithm, we chose a Monte Carlo approach: sample several random displays $X_1, \ldots, X_{N_D}$ from the distribution $P(T = T_i)$ and choose the one which minimizes C. Though crude, it still achieves considerable gains over the most-probable display update strategy.

C. Related Work

The general idea of maximizing the expected information from a query has also been pursued in the machine learning literature under the name "Active Learning" or "Learning with Queries" [43]. Active learning techniques have been shown to outperform simple probability ranking for document classification [44]. We know of no application of active learning techniques to database retrieval.

Comparison searching with errors has also been studied in the theoretical computer science literature. The algorithm of Rivest et al. [45] assumes that the number of errors has a known bound. Nevertheless, their algorithm is similar to the one presented here, in the sense that it minimizes at each step an information-theoretic bound on the number of future comparisons. The algorithm of Pelc [46] allows errors to occur at random but requires them to be independent of the comparison and the target and furthermore does not guarantee that the target is found. So

while both of these algorithms run in provably logarithmic time, they also operate under more restrictive conditions than *PicHunter*.

D. Simulation Results

This section evaluates these two display update schemes (most-probable and most-informative) by comparing them to other plausible methods for choosing $X_1, \ldots, X_{N_D}$:
Sampling Sample $X_1, \ldots, X_{N_D}$ from the distribution $P(T = T_i)$. This is a special case of the Most Informative scheme where only one Monte Carlo sample is drawn.
Query by Example Let $X_1, \ldots, X_{N_D}$ be the N_D closest items to the winner of the last comparison. This is a favorite approach in systems without relevance feedback [7]. It does not exploit memory or a stochastic user model.
The idea is to simulate a user's responses by sampling from the stochastic user model. The database is synthetic, consisting of points uniformly-distributed inside the unit square. This allows databases of varying sizes to be easily drawn. The simulated users used the Euclidean distance measure.

D.1 Deterministic case

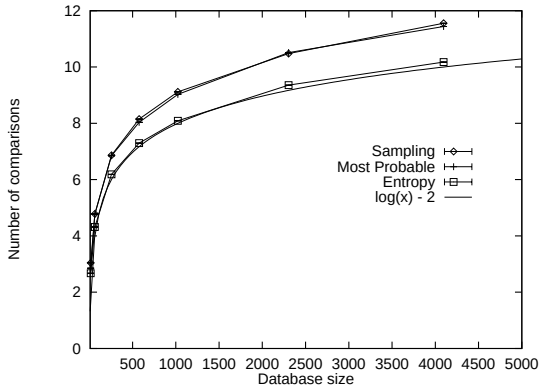


Fig. 3. The number of iterations needed to find a target, for varying database sizes and search strategies. User actions were generated according to P_{ideal}.

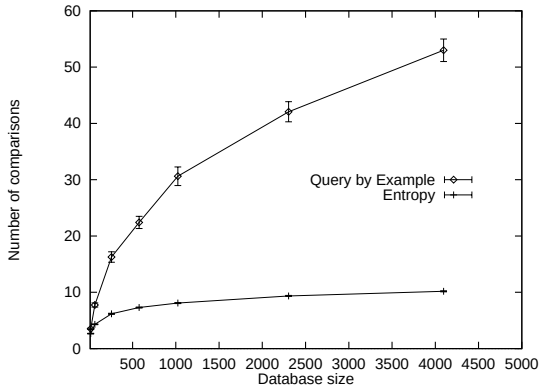


Fig. 4. Same as Figure 3 but including the Query-by-Example method.

Figure 3 plots the empirical average search time for finding a randomly selected target as a function of database size, using the Most Probable, Sampling, and Most Informative (entropy) schemes. The number of choices N_D was two. User actions were generated by the P_{ideal} model. In all experiments, the average is over 1000 searches, each with a different target, and the database was resampled 10 times. Performance of these three schemes is comparable, scaling like $\log_2 n$. In particular, the Most Informative scheme is virtually optimal, with deviations only due to a limited number of Monte Carlo samples. The Query-by-Example scheme is quite different, as shown in Figure 4; note the change in vertical scale. The Query-by-Example method is not exploiting comparison information very well; its time scales as $n^{0.5}$. Increasing N_D or the dimensionality will reduce the difference between the four schemes.

D.2 Nondeterministic case

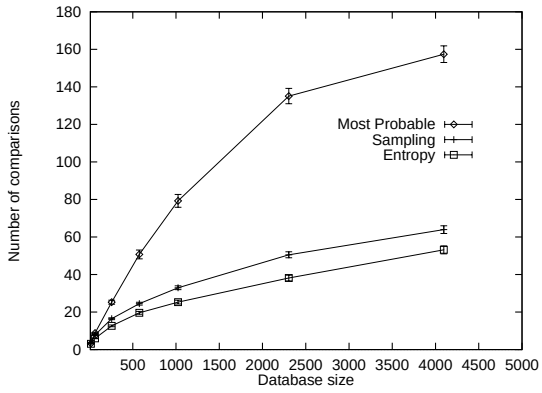


Fig. 5. Here user actions were generated according to $p_{sigmoid}$ with $\sigma = 0.1$.

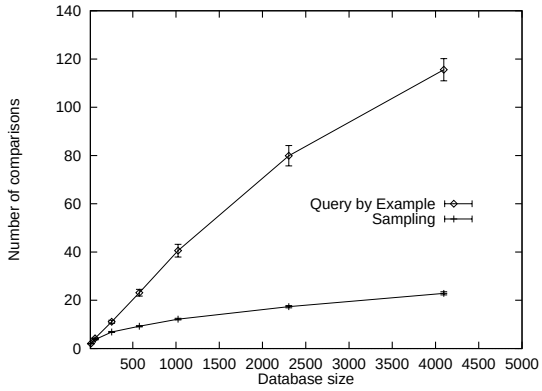


Fig. 6. Same as Figure 5 but including the Query-by-Example method. n square.

Figure 5 shows what happens when user actions are generated by the $p_{sigmoid}$ model, with $\sigma = 0.1$. Increasing the database size causes the unit square to be sampled more and more finely, while the distance uncertainty threshold σ remains the same. Thus it is much harder to isolate a particular target in a large database than in a small one, as would be true in a real situation. Again, the Sampling and Most Informative schemes are similar in search time, which scales like a square root. However, the fragility of the

Most Probable scheme is evident here. Figure 6 also reveals a large discrepancy in the Query-by-Example scheme. An explanation for this is that the Most Probable and Query-by-Example schemes tend to choose elements which are close together in feature space—exactly when comparisons are most unreliable. Entropy-minimization, by contrast, automatically chooses displays for which comparisons are reliable. The Most Probable scheme also does not properly exploit broad and nonuniform distributions, or distributions which are multi-modal. Furthermore, a multi-modal distribution causes this scheme to switch to different parts of the database between iterations, which is disconcerting to a real user.

E. Experiments on Updating Schemes

The most recent experiments on *PicHunter*, reported here for the first time, addressed two issues: 1) Compare performances with the two most promising display updating schemes. 2) A secondary issue was to evaluate the new pictorial features introduced in section IV-A. Towards this end, we tested seven versions of *PicHunter*, coded with a digraph notation XY that is analogous to that used in section V-D. Some of these versions were the same as those tested previously (section V-D); in these cases we label the scheme with the trigraph notation used in section V-D next to the new digraph notation. The first letter X of the digraph XY represents the display-updating mode: **E** stands for the entropy-based "most-informative" updating. **R** stands for a **r**elative-distance-based, most-probable scheme that uses memory. **A** is similar to **R**, but uses an **a**bsolute distance criterion without memory ("query-by-example"). The second letter Y of the digraph XY denotes the features used by the model for similarity judgments: **P** for pictorial only, **S** for semantic only, and **B** for both, with **P'** denoting the new pictorial features, and **B'**=S+P' denoting the combination of the semantic features and the new pictorial features (the semantic features remained the same). The 7 versions are the following: EB', EP', and ES, which are entropy-based schemes with S+P', P', and S, respectively; RB' and AB', which are the same as the versions denoted by MRB' and NAB' in the trigraph notation, but using the combination of the new pictorial features and the semantic features; finally, RS and RP, which are identical to versions MRS and MRP of section V-D. All 7 versions were run with the same set of 15 target images, which was different from the set of 15 images of the experiments of section V-D. 7 users, who were naive as to the purposes of the experiment and had never used *PicHunter* before, participated in the 7×7 Latin-square design [33]. The results are shown in Table V, which uses the same notation as that of Tables I and II. The same two experienced users who participated in all the previous experiments also ran a subset of the experiments.

The user model in the new version of *PicHunter* (the results of which are shown in the first 5 columns) differs from the old one (last two columns) in two major ways, besides the pictorial features: 1) The sigmoid slope *sigma* and the feature weights w_i are different, since they are based on more training data, and optimized in a better way than before. This affects the performance of individual metrics as well as combinations of metrics. 2) The user model in the old version was an approximate softmin while the new version uses an exact softmin.

PicHunter Version	EB'	EP'	ES	RB' MRB'	AB' NAB'	RS MRS	RP MRP
No. displays, V	11.3	25.8	16.0	12.0	20.4	11.8	29.6
Standard Error, SE	1.16	3.40	1.74	1.17	2.52	.755	1.70
Variability, SE/V	.103	.132	.109	.098	.124	.064	.057
V, 2 exper. users	6.80	10.2	8.30	8.65	11.5		

TABLE V

THE RESULTS OF THE EXPERIMENTS THAT TESTED ENTROPY-BASED DISPLAY UPDATING SCHEMES TO TRADITIONAL SCHEMES, AS WELL AS THE EFFECTIVENESS OF THE NEW PICTORIAL FEATURES. THE EXPECTED VALUE OF V UNDER RANDOM SEARCH IS 83.3.

One can make the following observations on the data of Table V. 1) A comparison of the entropy-based schemes reveals that the combination of both semantic and pictorial features (EB') results in better performances than using either semantic (ES) or pictorial (EP') features alone, as expected. This expected behavior is unlike the surprising pattern of results of the experiments in section V-D. One possibility for the difference is that the new set P' of pictorial features is better than the original ones P, hence they improve performance when they combine with the semantic features. 2) The best entropy-based scheme (EB') is at least as good as the best most-probable scheme (MRB'), and both are much better than the QBE search (NAB'). The superiority of the entropy-based scheme is even more evident in the results of the experienced users. It is interesting to note that such a display strategy produces a qualitatively different feel to the overall system. At the beginning of the search, the displayed set of images shows a large variety which is in contrast to traditional display algorithms that attempt to display a set of very similar images. 3) Conditions RS and RP were used in order to compare the old version to the new one, where both were tested with the common new set of 15 target images. The data indicate that the combination of both S and P' features (RB') does not seem to yield an improvement over the semantics-only version (RS), which performs remarkably well. Parenthetically, one piece of useful data that would enable a complete comparison is performance of the most-probable scheme with the new pictorial features alone, i.e., the RP' scheme.

At this point, it is useful to reflect on the improvements of the present schemes as compared with earlier versions. In the original implementation, about half of the searches by first-time users were labeled "unsuccessful" in that users gave up after an excessive number of iterations. The average number of images visited *in the successful searches only* was 300 [3] which was 13.3% of the expected number under random search for the 4522-image database. This number must be at least doubled if we want to include the effect of the unsuccessful searches. By contrast, our

users had only successful searches by definition, because they were required to continue searching until the target was found. This requirement necessitated some excessively long searches, which may be statistical outliers, yet their lengths inflate the mean value. Despite this, the improved schemes converged after visiting, on average, 100.8 images, which is still 13.4% of the expected number under random search for the 1500-image database. Experienced users do a lot better, averaging 8.2% of the expected length of random searches. Consistent users in *PicHunter* evaluations, in addition to the authors, report that present versions of *PicHunter* perform remarkably better than earlier versions in locating targets efficiently. It must be emphasized that these figures are for target testing, which is the most demanding of the search types.

VIII. Extensions

All *PicHunter* versions to date have been using the target search paradigm. However, when a user operates *PicHunter* to search for images that are similar to a prototype image, say, a North-Pole scene, the system quickly produces displays with similar images; in a lax sense, under these conditions, this type of search can be considered as a category search. More formally, however, *PicHunter* can become a *category-search* engine if the Bayesian scheme is modified to treat sets of images rather than individual images. The challenge for the system would be to discern the commonality of the features that specify a certain category that the user has in mind.

The main characteristic of *open-ended browsing* is that users change their goals during the search either gradually or quite abruptly, as a result of having encountered something interesting that they had not even considered at the beginning of the search. Accommodating these changes necessitates a modification of the probability distribution updating scheme. For the gradual changes one may assign weights to the probability updating factors that are strongest for the most recent iteration steps, and decay exponentially for distant past steps. For the abrupt changes, one option is to enable the user to indicate such switches, and then assign small weights to iterations prior to the abrupt change.

Although *PicHunter* was developed specifically for searching image databases, its underlying design and architecture make it suitable for other types of databases that contain digital data, such as audio passages or video-sequence databases.

IX. Ideas for Improvement

A. More Representative Databases

The main problem of the initial database, described in section V-B, is that its images are clustered into thematic categories of 100 elements each. This results in a clustered distribution in feature space, which may not be representative of distributions in larger databases. *PicHunter*'s problem of occasionally "getting stuck", i.e. producing displays of a certain category in step after step (section VII-A), may in fact turn out to be an advantage in databases that have a wider, non-clustered, distribution in feature space. A representative image database is needed by the CBIR community as a means towards establishing a benchmark for algorithm assessment.

B. More Relevant Image Features

PicHunter's performance improved when the new pictorial features were incorporated in the user model. The main advantage of the new features of the color autocorrelogram and the color-coherence vector is that they embody some measure of the spatial extent of each color, rather than a conventional color histogram's mere first-order statistics. Along the same lines, the user model can benefit by adding more information on the spatial properties of images, such as location, size, shape, and color of dominant objects in the image. The inclusion of spatial and figural features is especially important for the minority of color-blind people. Another feature can be the first few low-frequency Fourier components of the image's spectrum, or other measures of the distribution of spatial frequencies [17]. The need is evident for more psychophysical studies that investigate what criteria are used by humans in judging image similarity [47], [48]. Ultimately, some shape information [9] or object-based scene description [49] must be employed in CBIR systems.

C. More Complex User Feedback

PicHunter was deliberately designed with a very simple user interface, to concentrate on more fundamental issues in CBIR research. The items below remove this simplicity constraint by suggesting more complex ways of accepting users' feedback. Obviously, the user model needs to be adjusted accordingly to accommodate the additional feedback. Naturally, the introduction of new feedback modes has to be evaluated vis-a-vis the conflicting requirement for a simple user interface; appropriate experiments can decide whether there are any significant gains by the proposed idea to make it worth pursuing.

Specify which feature(s) are relevant in a selected image. Post-experimental interviews with the users reveal that some of them followed a common strategy in selecting similar images in a display. They selected one image because it looked similar to the target in terms of, say, overall color, and another image for its similarity in, say, overall contrast. This suggests the possibility of allowing users to specify which feature(s) make a selected image desirable, and can be extended to cover semantic features as well.

Strength of selected image. Independently of specifying feature relevance, the user could also indicate the degree, or strength, of similarity between a selected image and the pursued target. This can be done by providing either a slide bar or a series of buttons below each image in the display.

Portions of selected image. Yet another independent form of more complex user feedback is to indicate the portion(s) of the image that is (are) similar to the target. The

interface can still maintain simplicity by allowing the user to circumscribe relevant portions using the mouse.

D. More Complex Displays

The first three items below discuss how best to start the iteration process by using as informative an initial display as possible (the first item deals with expanding the current version by just providing more images in the initial display, the next two deal with initial queries). The last item provides the user with information on why images were selected to be included in the current display.

Initial display. It would be helpful to give the user a head start by using a more complex initial display, keeping displays in the rest of the iterations as simple as described so far. For the particular database that we worked with, one idea that we experimented with was to take advantage of the fact that the database contained clusters in feature space. Thus we included in the first display a large number of images (50 or so), each being at the center of a cluster. This seemed to speed up search time but as yet we have no comprehensive data from such informal experiments.

Initial query template. PicHunter can be modified to add a feature that is common in many "query-by-example" CBIR systems that use a "sketch" to specify a template in order to start a search with a better-than-random initial display. The user can be given the option to select desirable values for the pictorial features by using, say, "slide bars". These bars can be used to specify mean brightness, luminance contrast, color content, etc. This will enable the user to start the search with a good guess in the first iteration.

Initial query. Just as textual search engines do with words and phrases, CBIR systems may use Boolean expressions on semantics. The analogy is the following: with a database browser, one specifies logical expressions of words when searching for a paper in the literature; by analogy, one can use self-explanatory icons (such as for tree, house, animal, town, aircraft, person, crowd, lake, etc.), and build an interface for forming Boolean expressions that characterize the target image. This will enable users to start with an initial display that is very close to the desired target.

Which features caused an image to be displayed. The previous subsection dealt with allowing users to provide more complex feedback to the system. Reciprocally, users can benefit by knowing *PicHunter*'s current "beliefs", as this will give them an idea of how their choices affect the system. A simple way is to provide an indicator, next to each displayed image, on the system's relative strength of belief. A more complex display could indicate which feature(s) caused each image to be selected in the current display.

E. Improved User Model

One area in which the scheme can be improved is in handling the special case in which the user does not select any image in the current display before hitting the "GO" button to continue the search. This is an essential special case because users frequently find themselves forced to proceed to the next iteration without selecting any image. Currently, the program keeps the probability vector unchanged and then enters the display-update routine, in essence ignoring the user's action. However, some, perhaps most, users make this selection precisely to indicate that they want to avoid the types of displayed images. Experiments are needed to explore modifications to the algorithm for dealing with this special case.

X. Conclusions – Discussion

PicHunter's new approach is its formulation on a Bayesian framework, which tries to predict the user's actions for refining its answers to converge to a desired target image. The central data structure is a *vector of posterior probability distribution* across the entire database, i.e., each image has an entry in the vector that represents the probability of its being the target. This distribution is updated based on the user action after each iterative display. This action is "interpreted" by the *user model*, which is the second major component of the system, together with the probability vector. This is an action-predictor model that uses rudimentary knowledge of humans' judgments of image similarity, based on empirically derived pictorial and semantic features. The user model was refined on the basis of data obtained from our similarity judgment experiments (section IV-C). The third major component, the *display-updating scheme*, is concerned with how to select the images for the next iteration's display. We presented two major alternatives, a most-probable and a most-informative scheme, which exhibited considerably improved performances over alternative schemes. Overall, the system performs quite well for a wide spectrum of users tested on a wide variety of target images. The improvement over earlier versions, as verified by the reported experiments and attested by consistent users of the system, is very promising.

In comparing algorithms based on their performances under the target testing scheme, we make the implicit assumption that systems which are optimized under this target testing condition will also perform well in category searches and open-ended browsing. We reported on experiments that support this assumption when the target testing version is used for a form of category searching (section V-E). Performance under open-ended browsing is much more difficult to quantify because of the vague nature of the task at hand. The main requirement in open-ended browsing is that the system display images that are similar to those selected by the user, and avoid displaying images that are similar to the non-selected images, resulting in appropriate changes to the display updating scheme. At the same time, because the goal changes during the search, the user must be allowed to reset the memory when he/she makes such a goal change, so that earlier choices no longer affect the display updating decisions.

It would be highly desirable to rank-order the various criteria used by humans for judging image similarity according to their importance. Weights can the be assigned to such criteria according to the role they play in predicting

judgment of similarity by humans. Relevant research has been carried out on the application of multi-dimensional scaling (MDS) methods for finding principal attributes to characterize texture perception [47]. Much image processing research has also been conducted for utilizing texture as a pictorial feature in CBIR systems [50], [14], [51]. Rogowitz et al. (1998) [48] applied MDS analysis to humans' judgments of similarity using natural images; this task is quite complex, mainly due to the presence of semantics. An interesting experiment along these lines is to let humans play the role of *PicHunter*, to see what criteria they use, and to compare their performance with that of *PicHunter*.

The computation performed by *PicHunter* with each user interaction, and its main memory space requirements scale linearly with the number of images in the database assuming the user model requires constant time. Execution time is dominated by the user model[4], and space by the storage of feature vectors.[5] As such our approach might be expected to handle perhaps millions of images in today's technological environment, but not hundreds of millions. We remark that approximating its Bayesian update with a sublinear number of user model executions and the feature vectors in secondary storage, represents an interesting area for future theoretical and systems work.

While we have demonstrated search times that are much shorter than brute force, they are clearly not short enough to satisfy many users. It is possible that our pure relevance feedback approach might lead to a fully acceptable system, but it is also possible that a hybrid approach will prove best. That is, one that involves some explicit querying, but uses relevance feedback to further shorten the search.

Our experiments indicate that humans attend to the semantic content of images in judging similarity. Highly specialized databases, such as medical image databases in large medical centers, have started to get semantically annotated, and the trend appears to carry to images in generic electronic libraries. Thus, it seems that searching for an image will have much in common with searching for text documents in library databases.

In all our experiments, experienced users performed at a level that was considerably better than users with little experience, as expected. For example, they completed the average search by visiting only 65.4% and 53.2% , respectively, of the images visited by first-time users for the experiments reported in sections V-D and VII-E. It must be noted, however, that even first-time users improve their scores substantially, after we explained to them the algorithm's user model (section VI-B). This training was very brief, lasting less than 8 minutes, after which their (already good) pre-training performance improved by reducing the search length by about 20%. This substantial improvement after minimal training of non-expert users is a desirable feature for a search engine, enabling the development of a short on-line training session for first-time users.

Most published papers provide data on the search length in terms of how many iterations are needed before users find an image that is similar to a desired target. This, however, may not be a reliable measure, because even a random search can produce relatively short search lengths, as shown in the experiments of section V-E (column RAND/C in Table II). In fact, this latter search length could be used as a baseline against which to measure the performance of an algorithm under test. Even better, we believe that data under the target search paradigm offer an objective measure of performance. In addition, this measure exhibits small standard deviations across users' scores, when each user's score is averaged over an adequately large number of searches with different targets, whereas the corresponding random-search baseline measure exhibits much higher variability [6], [5]. Thus, target testing requires experiments with fewer users to establish the same degree of confidence in the statistics.

The experiments in this paper were designed with *PicHunter* in mind. Nevertheless, their results and findings are useful and potentially applicable to any CBIR system and, more generally, to any system that involves judgment of image similarity by humans.

[4]Any machine learning technique capable of producing a predictive model may be used to implement the required user model, so it is hard to say anything general about its computational burden.

[5]If an entropic display update is used, its computational burden is significant as well.

Acknowledgements

We thank Y. Hara and K. Hirata of NEC Central Laboratories for directing our attention to CBIR system research and for providing us with their system and image database. We also thank Joumana Ghosn, Kevin Lang, and Steve Omohundro, who have contributed significantly in *PicHunter*'s development. We are grateful to Bob Krovetz, Talal Shamoon, and Harold Stone for valuable discussions, and to our anonymous reviewers for their useful suggestions. We thank Ebony Brooks, Tiffany Conway, and Sejal Shah for administering experiments, and Akos Feher for providing technical support.

References

[1] I. J. Cox, J. Ghosn, M. L. Miller, T. V. Papathomas, and P. N. Yianilos, "Hidden annotation in content based image retrieval," in *Proc IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997, pp. 76–81.

[2] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos, "Pichunter: Bayesian relevance feedback for image retrieval," in *Int. Conf. on Pattern Recognition*, 1996, vol. 3, pp. 362–369.

[3] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos, "Target testing and the `PicHunter` Bayesian multimedia retrieval system," in *Proc. of the 3rd Forum on Research and Technology Advances in Digital Libraries, ADL'96*, 1996, pp. 66–75.

[4] I. J. Cox, M. L. Miller, T. P. Minka, and P. N. Yianilos, "An optimized interaction strategy for bayesian relevance feedback," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1998, pp. 553–558.

[5] T. V. Papathomas, T. E. Conway, I. J. Cox, J. Ghosn, M. L. Miller, T. P. Minka, and P. N. Yianilos, "Psychophysical studies of the performance of an image database retrieval system," in *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology, Conference on Human Vision and Electronic Imaging III*, 1998, pp. 591–602.

[6] T. V. Papathomas, T. E. Conway, I. J. Cox, M. L. Miller, T. P. Minka, and P. N. Yianilos, "Psychophysical evaluation for the performance of content-based image retrieval systems," *Investigative Ophthalmology and Visual Science*, vol. 39, no. 4, pp. S1096, 1998.

[7] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[8] D. Forsyth, J. Malik, and R. Wilensky, "Searching for digital pictures," *Scientific American*, pp. 88–93, 1997.

[9] B. Gunsel and A. M. Tekalp, "Shape similarity matching for query-by-example," *Pattern Recognition*, vol. 31, no. 7, pp. 931–944, 1998.

[10] B. Gunsel, A. M. Tekalp, and P. J. L. van Beek, "Content-based access to video objects: Temporal segmentation, visual summarization, and feature extraction," *Signal Processing*, vol. 66, no. 2, pp. 261–280, 1998.

[11] K. Hirata and T. Kato, "Query by visual example; content based image retrieval," in *Advances in Database Technology– EDBT '92*, A. Pirotte, C. Delobel, and G. Gottlob, Eds., Berlin, 1992, Springer-Verlag.

[12] T. Kato, T. Kurita, H. Shimogaki, T. Mizutori, and K. Fujimura, "Cognitive view mechanism for multimedia database system," in *IMS '91 Proceedings. First International Workshop on Interoperability in Multidatabase Systems*, 1991, pp. 179–186.

[13] W. Y. Ma and B. S. Manjunath, "A texture thesaurus for browsing large aerial photographs," *Journal of the American Society for Information Science*, 1997.

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

[15] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233–254, 1996.

[16] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in *Proc. of IEEE Int. Conf. on Image Processing*, Santa Barbara, CA, October 1997.

[17] H. S. Stone and C.-S. Li, "Image matching by means of intensity and texture matching in the Fourier domain," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1996, pp. 337–349.

[18] Q. Tain and H. J. Zhang, "Digital video analysis and recognition for content-based access," *ACM Computing Surveys*, vol. 27, no. 4, 1995.

[19] G. Yihong, Z. Hongjiang, and C. H. Chuan, "An image database system with fast image indexing capability based on color histograms," in *Proceedings of 1994 IEEE Region 10's Ninth Annual International Conference. Theme: Frontiers of Computer Technology*, 1994, vol. 1, pp. 407–11.

[20] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu, "Video parsing, retrieval and browsing: An integrated and content-based solution," in *Proc. of ACM Multimedia'95*, Nov 1995.

[21] A. Del Bimbo, P. Pala, and S. Santini, "Visual image retrieval by elastic deformation of object sketches," in *Proceedings IEEE Symposium on Visual Languages*, 1994, pp. 216–223.

[22] Y. Rui, T. S. Huang, and S. Mehrotra, "Relevance feedback techniques in interactive content-based image retrieval," in *Proc. of IS&T and SPIE Storage and Retrieval of Image and Video Databases VI*, San Jose, CA, January 1998.

[23] T. P. Minka and R. W. Picard, "Interactive learning using a "society of models"," *Pattern Recognition*, vol. 30, no. 4, 1997.

[24] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *IEEE Computer Vision and Pattern Recognition Conference*, San Juan, Puerto Rico, June 1997, pp. 762–768.

[25] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Fourth ACM Conference on Multimedia*, Boston, Massachusetts, November 1996, pp. 65–73.

[26] E. Saber, A. M. Tekalp, R. Eschbach, and K. Knox, "Annotation of natural scenes using adaptive color segmentation," *IS&T/SPIE Electronic Imaging*, Feb. 1995, San Jose, CA.

[27] J. R. Smith and S.-F. Chang, "Visually searching the web for content," *IEEE Multimedia*, vol. 4, no. 3, pp. 12–20, 1997.

[28] M. Stricker and A. Dimai, "Color indexing with weak spatial constraints," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1996, pp. 29–40.

[29] H. J. Zhang, Y. Gong, C. Y. Low, and S. W. Smoliar, "Image retrieval based on color features: An evaluation study," in *Proc. SPIE Conf. on Digital Storage and Archival*, Oct 1995.

[30] T. Kurita and T. Kato, "Learning of personal visual impressions for image database systems," in *Proceedings of the Second International Conference on Document Analysis and Recognition*, 1993, pp. 547–552.

[31] M. Eisenberg and C. Barry, "Order effects: A preliminary study of the possible influence of presentation order on user judgements of document relevance," *Proc. of the American Society for Information Science*, vol. 23, 1986.

[32] "Corel stock photo library," Corel Corp., Ontario, Canada, 1990.

[33] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters: An Introduction to Design, Analysis, and Model Building*, J. Wiley and Sons, New York, N.Y., 1978.

[34] J. Barros, J. French, W. Martin, P. Kelly, and J. M. White, "Indexing multispectral images for content-based retrieval," in *Proceedings of the 23rd AIPR Workshop on Image and Information Systems, Washington DC, Oct.*, 1994.

[35] M. Hirakawa and E. Jungert, "An image database system facilitating icon-driven spatial information definition and retrieval," in *Proceedings 1991 IEEE Workshop on Visual Languages*, 1991, pp. 192–198.

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

[37] P. M. Kelly and T. M. Cannon, "Candid: Comparison algorithm for navigating digital image databases," in *Proceedings Seventh International Working Conference on Scientific and Statistical Database Management*, 1994, pp. 252–258.

[38] M. Stricker and M. Swain, "The capacity of color histogram indexing," in *Proceedings 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 704–708.

[39] M. J. Swain and D. H. Ballard, "Indexing via color histograms," in *Proceedings Third International Conference on Computer Vision*, 1990, pp. 390–393.

[40] C. Frankel, M. J. Swain, and V. Athitsos, "Webseer: An image search engine for the world wide web," Tech. Rep. TR-96-14, University of Chicago Department of Computer Science, July 1996.

[41] D. Haines and W. B. Croft, "Relevance feedback and inference networks," in *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993, pp. 2–11.

[42] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1993.

[43] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," in *Advances in Neural Information Processing Systems*, Cambridge, MA, 1993, MIT Press.

[44] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. of ACM-SIGIR Conf. on R&D in Information Retrieval*, W. Bruce Croft and C. J. van Rijsbergen, Eds., Dublin, Ireland, July 1994, Springer-Verlag.

[45] R. L. Rivest, A. R. Meyer, D. J. Kleitman, K. Winklmann, and J. Spencer, "Coping with errors in binary search procedures," *Journal of Computer and System Sciences*, vol. 20, pp. 396–404, 1980.

[46] A. Pelc, "Searching with known error probability," *Theoretical Computer Science*, vol. 63, pp. 185–202, 1989.

[47] R. A. Rao and G. L. Lohse, "Towards a texture naming system: Identifying relevant dimensions of texture," *Vision Research*, vol. 36, pp. 1649–1669, 1996.

[48] B. Rogowitz, T. Frese, J. R. Smith, C. A. Bouman, and E. Kalin, "Perceptual image similarity experiments," in *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology, Conference on Human Vision and Electronic Imaging III*, 1998, pp. 576–590.

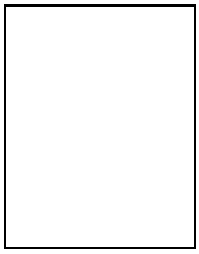
[49] J. Ponce, A. Zisserman, and M. Hebert, *Object representation in computer vision - II*, Number 1144 in LNCS. Springer, 1996.

[50] A. Kankanhalli and H. J. Zhang, "Using texture for image retrieval," in *Proc. of ICARCV'94*, 1994.

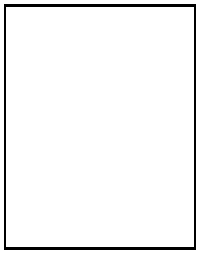
[51] M. Beatty and B. S. Manjunath, "Dimensionality reduction using multi-dimensional scaling for content-based retrieval," in *IEEE International Conference on Image Processing*, 1997.

[52] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," in *IEEE International Conference on Image Processing*, 1997.

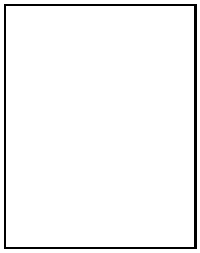
[53] W. Y. Ma and B. S. Manjunath, "Texture features and learning similarity," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1996, pp. 425–430.
[54] V. Ogle and M. Stonebraker, "Chabot: Retrieval from a relational database of images," *IEEE Computer*, vol. 28, no. 9, pp. 40–48, 1995.
[55] R. Rickman and J. Stonham, "Content-based image retrieval using color tuple histograms," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1996, pp. 2–7.
[56] J. R. Smith and S.-F. Chang, "Searching for images and videos on the world-wide web," Columbia University CU/CTR Technical Report 459-96-25, to appear in IEEE Multimedia Magazine, 1997.
[57] Marc Davis, "Media Streams: An iconic visual language for video representation," in *Readings in Human-Computer Interaction: Toward the Year 2000*, Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton, and Saul Greenberg, Eds., pp. 854–866. Morgan Kaufmann Publishers, Inc., San Francisco, 2nd edition, 1995.
[58] Yuri Boykov, Olga Veksler, and Ramin Zabih, "Disparity component matching for visual correspondence," in *IEEE Computer Vision and Pattern Recognition Conference*, San Juan, Puerto Rico, June 1997.
[59] G. Pass and R. Zabih, "Histogram refinement for content based image retrieval," in *Proc. IEEE Workshop on Applications of Computer Vision*, 1996, pp. 96–102.
[60] R. W. Picard and F. Liu, "A new Wold ordering for image similarity," in *Proc. IEEE Conf. on Acoustics, Speech, and Signal Proc.*, Adelaide, Australia, April 1994, pp. V–129–V–132.
[61] Donald E. Knuth, *The Art of Computer Programming*, vol. 3, Addison-Wesley, Reading, Massachusetts, 2nd edition, 1973.
[62] Stephen M. Omohundro, "Five balltree construction algorithms," Tech. Rep. TR-89-063, International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, CA 94704, December 1989.
[63] K. V. S. Murthy, *On Growing Better Decision Trees from Data*, Ph.D. thesis, Johns Hopkins University, 1995.
[64] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 2nd edition, 1992.
[65] V. V. Federov, *Theory of Optimal Experiments*, Academic Press, New York, 1972.
[66] Frederick Jelinek, John D. Lafferty, and Robert L. Mercer, "Basic methods of probabilistic context free grammars," in *Speech Recognition and Understanding*, Pietro Laface and Renato De Mori, Eds., Berlin, 1992, vol. F75 of *NATO Advanced Sciences Institutes Series*, pp. 345–360, Springer Verlag.
[67] Taylor L. Booth and Richard A. Thompson, "Applying probability measures to abstract languages," *IEEE Transactions on Computers*, vol. 22, pp. 442–450, 1973.
[68] Noel Cressie, *Statistics for Spatial Data*, Wiley, New York, 1993.
[69] Eric Saund, "A multiple cause mixture model for unsupervised learning," *Neural Computation*, vol. 7, no. 1, January 1995.
[70] J. Smith and S.-F. Chang, "Tools and techniques for color image retrieval," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1996, pp. 426–437.
[71] R. W. Picard and T. P. Minka, "Vision texture for annotation," *Journal of Multimedia Systems*, vol. 3, pp. 3–14, 1995.
[72] Howard R. Turtle and W. Bruce Croft, "A comparison of text retrieval models," *The Computer Journal*, vol. 35, no. 3, pp. 279–290, 1992.
[73] A. R. Smith, "Color gamut transform pairs," *ACM Computer Graphics (SIGGRAPH)*, vol. 12, no. 3, pp. 12–19, 1978.
[74] Donna Harman, "Relevance feedback revisited," in *15th Ann Int'l SIGIR '92/Denmark-6/92*, 1992.
[75] IJsbrand Jan Aalbersberg, "Incremental relevance feedback," in *SIGIR '92/Denmark-6/92*, 1992.
[76] Masaomi Oda, "Context dependency effect in the formation of image concepts and its application," in *Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics. Decision Aiding for Complex Systems*, 1991, vol. 3, pp. 1673–8.
[77] V. V. V. N. Gudivada and Raghavan, "Content-based image retrieval systems," *IEEE Computer*, vol. 28, no. 9, pp. 18–22, September 1995.
[78] Tat-Seng Chua, Hung-Keng Pung, Guo-Jun Lu, and Hee-Sen Jong, "A concept-based image retrieval system," in *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences*, 1994.
[79] C. C. Chang and S. Y. Lee, "Retrieval of similar pictures on pictorial databases," *Pattern Recognition*, vol. 24, no. 7, pp. 675–680, 1991.
[80] B. J. Oommen and C. Fothergill, "Fast learning automaton-based image examination and retrieval," *The Computer Journal*, vol. 36, no. 6, pp. 542–553, 1993.
[81] C. H. C. Leung, J. Hibler, and N. Mwara, "Content-based retrieval in multimedia databases," *Computer Graphics*, vol. 28, no. 1, 1994.
[82] G. Healey and D. Slater, "Global color constancy: Recognition of objects by use of illumination-invariant properties of color distributions," *J. Opt. Soc. Am.*, vol. 11, no. 11, 1994.



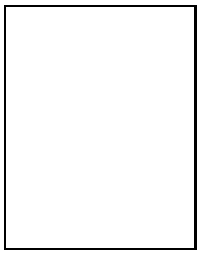
Ingemar J. Cox (SM) received his B.Sc. from University College London and Ph.D. from Oxford University. He was a member of the Technical Staff at AT&T Bell Labs at Murray Hill from 1984 until 1989 where his research interests were focused on mobile robots. In 1989 he joined NEC Research Institute in Princeton, NJ as a senior research scientist in the computer science division. At NEC, his research shifted to problems in computer vision and he was responsible for creating the computer vision group at NECI. He has worked on problems to do with stereo and motion correspondence and multimedia issues of image database retrieval and watermarking. He is a senior member of the IEEE and on the editorial board of the Int. Journal of Autonomous Robots. He is the co-editor of two books, 'Autonomous Robots Vehicles' and 'Partitioning Data Sets: With Applications to Psychology, Computer Vision and Target Tracking'.



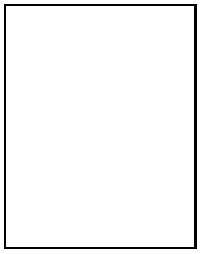
Matt L. Miller Matthew Miller began working in computer graphics at AT&T Bell Labs in 1979. After receiving a B.A. in cognitive science from The University of Rochester in 1986, he became lead programmer at NPS, a start-up developing color desktop publishing software. In 1987, he moved to Hollywood and divided his time between programming (for a living) and working on film crews (for fun). Between 1990 and 1993, he delivered graduate-level lecture courses in color graphics at Aarhus University in Denmark, and Charles University and Czech Technical University in Prague. From 1993 to 1997, he divided his time between running Baltic Images, a company he founded in Lithuania, and consulting for NEC Institute in Princeton, NJ. In 1997 he sold Baltic Images and returned full-time to Princeton to join Signafy Inc.



Thomas P. Minka Thomas P. Minka is a research assistant at the MIT Media Lab, pursuing a Ph.D. in Electrical Engineering and Computer Science. He strives for a formal Bayesian approach to statistical learning and pattern recognition problems. His interests in database retrieval include statistical models of content, relevance feedback, learning from multiple users, and adaptive visualization. He received the M.Eng. in Electrical Engineering and Computer Science from MIT in 1996.



Thomas V. Papathomas was born in Kastoria, Macedonia, Greece. He received his BS, MS, and P.h.D from Columbia University. He worked at Bell Laboratories from 1978 to 1989. Since 1989 he has been at Rutgers University, as Professor of Biomedical Engineering and as Associate Director of the Laboratory of Vision Research. He is also associated with the NEC Research Center at Princeton as a consultant. His research interests are in human and machine vision, and image processing. He is the editor-in-chief of Early Vision and Beyond, a volume of interdisciplinary research in vision, MIT Press, 1995, and member of the Editorial Board of the International Journal of Imaging Systems and Technology.



Peter N. Yianilos (SM'86) received B.S. and M.S. degrees from Emory University in 1978, and his Ph.D. in computer science from Princeton University in 1997. In 1979 he founded Proximity Technology Inc. which merged in 1988 to become Franklin Electronic Publishers, where he served as chief scientist and then president until 1991. At Franklin his compression techniques, search algorithms, data structures, and product concepts formed the basis for the first hand-held electronic books, ranging from spellers and dictionaries to Bibles and encyclopedias. Since 1991 he has been a senior research scientist at NEC Research Institute. Within electronic publishing his research interests include digital libraries, digital books, internet distributed storage systems, and information retrieval. Other research interests include Machine Learning and Stochastic Modeling, Pattern Recognition, Nearest Neighbor Search, and Data Compression,