Psychophysical studies of the performance of an image database retrieval system

Thomas V. Papathomas[a,b], Tiffany E. Conway[a], Ingemar J. Cox[c], Joumana Ghosn[d], Matt L. Miller[c], Thomas P. Minka[c], Peter N. Yianilos[c]

[a]Laboratory of Vision Research & [b]Department of Biomedical Engineering, Rutgers University, Psychology Building, Piscataway, New Jersey, 08854
[c]NEC Research Institute, 4 Independence Way, Princeton, NJ 08540
[d]Dept. Informatique et Recherche Operationnelle, Universite de Montreal, Montreal, Qc H3C-3J7, Montreal, Canada
[1]

ABSTRACT

We describe psychophysical experiments conducted to study `PicHunter`, a content-based image retrieval (CBIR) system. Experiment 1 studies the importance of using (a) semantic information, (b) memory of earlier input and (c) relative, rather than absolute, judgements of image similarity. The target testing paradigm is used in which a user must search for an image identical to a target. We find that the best performance comes from a version of `PicHunter` that uses only semantic cues, with memory and relative similarity judgements. Second best is use of both pictorial and semantic cues, with memory and relative similarity judgements.

Most reports of CBIR systems provide only qualitative measures of performance based on how similar retrieved images are to a target. Experiment 2 puts `PicHunter` into this context with a more rigorous test. We first establish a baseline for our database by measuring the time required to find an image that is similar to a target when the images are presented in random order. Although `PicHunter's` performance is measurably better than this, the test is weak because even random presentation of images yields reasonably short search times. This casts doubt on the strength of results given in other reports where no baseline is established.

Keywords: content-based image retrieval, image similarity, image annotation, image semantics

1. INTRODUCTION

Content-based image retrieval (CBIR) systems are essential in searching for specific images, or for a member of a desired category of images, in digital image databases. Traditional techniques based on textual-query search have limitations. For example, image databases may not be annotated. Furthermore, textual-based techniques based on *explicit* annotation require users to know the system's native language, or else the designers need to provide a translational front-end to achieve language independence. An additional problem, from the user's point of view, even for speakers of the CBIR system's native language, is how to translate the desired image into a textual query that will capture its semantic content in the system designers' spirit, so as to produce a near-optimal search. Finally, one has to deal with the problem of annotation errors, whether this is done manually or automatically. CBIR systems offer the hope of avoiding many of these problems.[1-11] They can be used in conjunction with annotation in a complimentary mode, such as in Cox et al.,[10] as discussed in more detail in Section 2.4 below.

CBIR systems usually start by displaying to the user an initial small subset of images, and expect the user to provide feedback by indicating which image(s) in the set look similar to the desired target image. The

Further author information -
T.V.P. (correspondence): Email: papathom@zeus.rutgers.edu; WWW: http://zeus.rutgers.edu/~papathom/thomas.html; Telephone: 732-445-6533; Fax: 732-445-6715 T.E.C.: Email: teconway@eden.rutgers.edu
I.J.C.: Email: ingemar@research.nj.nec.com; WWW: http://www.neci.nj.nec.com/homepages/ingemar
J.G.: Email: ghosn@IRO.UMontreal.CA M.L.M.: Email: mlm@research.nj.nec.com
T.M.: Email: tpminka@media.mit.edu; WWW: http://vismod.www.media.mit.edu/~tpminka/
P.N.Y.: Email: pny@research.nj.nec.com

algorithm then attempts to interpret the user's choices and to produce a new subset that contains images which are more similar to the target than those in the previous subset. This process is repeated, resulting in an iterative search that eventually converges to a subset containing one or more images that satisfy the user. More complex interfaces are employed in other CBIR systems.11 Some of them allow the user to provide a sketch of the target image through a special graphics front-end module, thereby providing some facility to incorporate semantics implicitly by virtue of location, shape, color, and other image attributes. Others refine the user's query in an adaptive learning paradigm.[12]

For CBIR algorithms that work purely with pictorial features, the main problem is to select features that are important in judging similarity between images by humans. After all, most such algorithms rely on the users' feedback to converge to a desired image. A set of features must have the property that similarity between any pair of images, as judged by humans, correlate well with some distance metric between the images that is a function of the differences in the features. The choice of features that optimizes such a correlation is a very complex problem, principally because: 1) It is hard for purely pictorial image features, no matter how sophisticated they are, to embody semantics; we are still a long time away from automated extraction of semantics from images. Nevertheless, semantics play a very important role in comparing images, as our earlier[10] and current experiments have shown. 2) Assuming that an optimal set of image features exists, identifying those features that are important in judging image similarity by humans requires years of systematic psychophysical research; what little work has been done along these lines has concentrated on simple subproblems, such as the identification of relevant dimensions for texture.[13, 3]) Judgement of image similarity varies among users, or even within the same user (depending perhaps on mindset, expectations, recent exposure, etc.).

This document describes the rationale, design, and results of psychophysical experiments that were conducted to address some key design and testing issues for `PicHunter`, a content-based image retrieval (CBIR) system.1,10 Although the experiments were designed with `PicHunter` in mind, their results can be applied to any CBIR system and, more generally, to any system that involves judgement of image similarity by humans. The rest of the paper is organized as follows: Section 2 is devoted to a brief description of `PicHunter`, to set the stage for the issues that were addressed in the experiments. Section 3 describes some preliminary experiments that were designed to address some of these issues. The main experimental design and results are covered in Section 4, and a general discussion of broader issues is in Section 5.

2. `PicHunter`: A BAYESIAN RELEVANCE-FEEDBACK CBIR SYSTEM

The `PicHunter` CBIR system has the following main properties: 1) It possesses an extremely simple user interface. 2) It is designed to perform optimally in searches that terminate only when a specific target image is obtained. 3) It employs a Bayesian scheme with long-term memory that uses the entire history of user responses during the search, rather than just the user feedback provided in the last iteration. 4) It incorporates a user model for interpreting the user's feedback. 5) Its design is flexible to allow the development of different versions, some of which use purely pictorial features, or semantics, or a combination of the two. The subsections below expand briefly on these properties. The interested reader is referred to the papers by Cox et al.[1,10] for more details.

2.1 User interface. `PicHunter` possesses a very simple interface that can be easily explained to perspective users, even if they are not computer literate. The search consists of a series of displays, D_0, D_1, D_2, D_3, ... , each of which contains N images. Various values of N have been tried in different implementations of `PicHunter`, but the experiments reported here used N=9, arranged in a 3x3 array, as shown in Figure 1. The series of displays converges to one that ultimately includes the desired target image. The system starts the search by displaying a "seed display", D_0, the images of which are randomly selected from the database I. The user's task is to select, from among the N images, the image(s) that is (are) similar to the target. The user may select zero or more images form the set of 9. After making a selection, the user signals to the system to execute the first iteration. The algorithm finds the N images that are most likely to be the target image, based on the user feedback as interpreted according to the user model. These N images constitute the next display, D_1, which is shown to the user, completing one iteration. In an iterative manner, the user selects the image(s) that is (are) similar to the target, if any, and submits his/her feedback to the system, which uses the model to select the images for the next display D_2. This process is repeated until the desired target shows up in one of the displays.

2.2 Bayesian scheme for using history of user feedback. Another important characteristic of `PicHunter` is its ability to utilize the entire history of user responses in determining the next display in the iteration sequence. We will present a simplified view of how this is implemented, starting from the very first display D_0. Given a target image, $I_t \in I$, there is no a priori reason for any one image to be favored over another, i.e., all images are equally likely to be the target image. Thus the algorithm assigns to each image I_i in the database the same probability $P(I_i) = 1/B$, where $B = |I|$ is the total number of images in the database I, for i=1, 2, ... , B. For each subsequent iteration k, employing display D_k, k=1, 2, 3, ... (see subsection 2.1), the probability that each image I_i is the target image is updated by multiplying its current value by the probability that the user, U, would give his/her actual response to D_k, if the target were I_i.

$$P(I_i) \leftarrow P(I_i)\, P(a_k \mid I_i, D_k, U) \qquad (1)$$

where a_k denotes the user's action in the kth iteration. Of course, when the current display D_k does not contain the target, its images are all assigned zero probability, and the probabilities of the remaining images are post-normalized to sum up to unity. Thus, the probability that an image I_i is the target image is the product of appropriate terms as shown in Eq. (1), and reflects the influence of all the user feedback from the very beginning of the search. Eq. (1) is derived from Bayes's rule, under the additional assumptions that: 1) the user's action a_k in the kth iteration depends only on I_t and D_k, i.e., it is independent of previous iterations. 2) The probability that I_i is the target is independent of the sequence of displays. More details on how Eq. (1) follows from Bayes' rule under these assumptions are provided in Cox et al.[1]

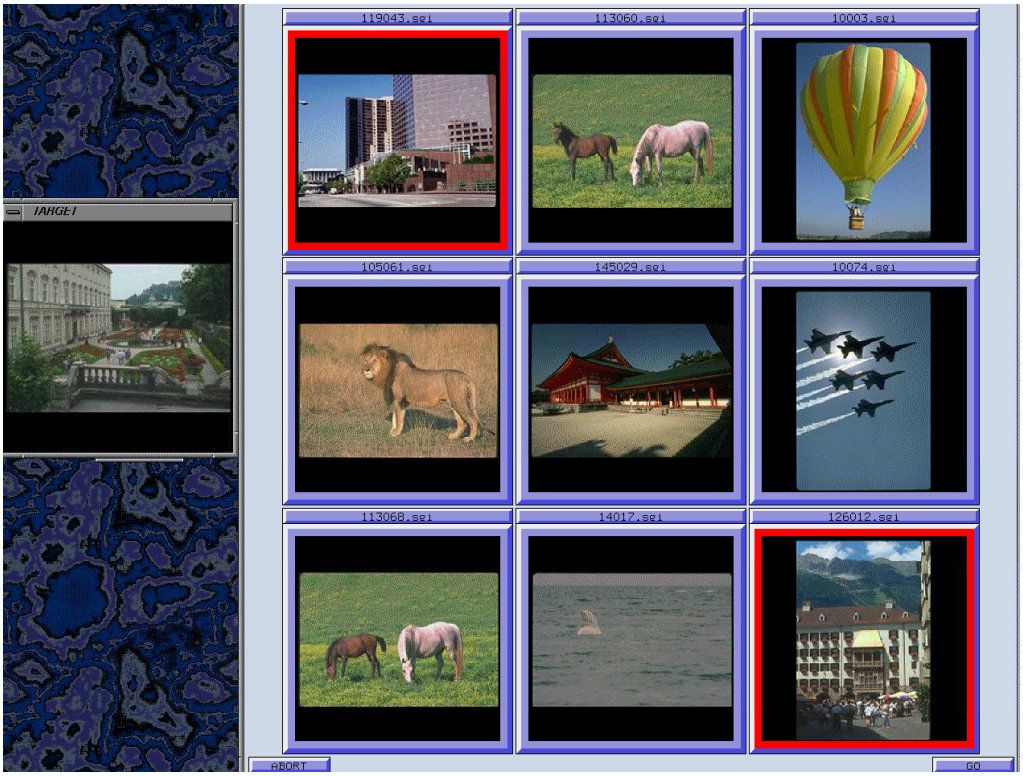

Figure 1. A typical display of the `PicHunter` system. The target image is shown at all times at the top-left corner of the display. In any display of this type, the user's task is to select the image(s), if any, that is (are) similar to the target image.

2.3 User model. The algorithm needs to interpret the user feedback according to the user model, and update the probability that each image in the database is the target image according to Eq. (1). Toward this end, we assume the existence of a function S that approximates the probability expressed in the right-hand side of Eq. (1)[1]:

$$P(a_k \mid I_i, D_k, U) \approx C\, S(a_k, I_i, D_k, U) \approx C\, S(a_k, I_i, D_k) \quad (2)$$

for an arbitrary constant C. In dropping U from Eq. (2) we have made the additional assumption that there are no differences among users. This is a simplifying assumption that can be revised in later versions; some work in taking account of individual differences was reported by Kurita and Kato.[14] The purely pictorial version of `PicHunter` works with 18 features that are derived for each image in the database. Some of these are the percentages of pixels that are of color blue, ditto for black, grey, white, red, etc. (for a total of 11 colors), the median intensity of the image, some measure of image contrast, etc. Parenthetically, it must be noted that color has proven to be a remarkably powerful image feature with some capability of retrieving images from common semantic categories.

The main task of the algorithm is to update the probability that each image $I_i \in I$ is the target image, based on the user's feedback a_k to the set of images in the kth iteration's display $D_k = \{I_{k1}, I_{k2}, I_{k3}, \ldots I_{k9}\}$, i.e., to compute $S(a_k, I_i, D_k)$ of Eq. (2). This is accomplished by establishing some distance metric $\mathbf{d}(I_i, I_m) \geq 0$ for any pair of images I_i and I_m, based on the images' vectors of features; if we let $\mathbf{f}(I_z) = \{f_1(I_z), f_2(I_z), f_3(I_z), \ldots, f_{18}(I_z)\}$ be the 18-dimensional feature vector for image I_z, then $d(I_i, I_m)$ is some function of $| f_n(I_i) - f_n(I_m)|$, n=1, 2, ... , 18, that satisfies the properties of a distance metric. Various weights are assigned to the features, according to their importance in judging image similarity. We then use this distance function to interpret the user's feedback, under the assumption that image similarity is inversely correlated with this distance. There are two main schemes of using the distances to update the probabilities:

1) *Absolute distance* criterion: In this scheme, only one image $I_{ks} \in D_k$ can be selected by the user in each iteration, and the user is instructed to choose the one that is most similar to the target. Subsequently, the probabilities of images $I_i \in I$ are enhanced or suppressed, depending on whether $d(I_i, I_{ks})$ is small or large, respectively. One way to do this is to let

$$P(I_i) \leftarrow P(I_i)\, G(\, d(I_i\, , I_{ks})\,) \quad (3)$$

for the image I_{ks} that was selected by the user, where G() is a monotonically decreasing function of its argument. Notice that, in this scheme, images $I_{kn} \in D_k$ that are not selected to be similar to the target do not play any role in the updating of the probabilities. The fact that they were not selected is ignored by the program; this is not the case in the second updating scheme.

Thus, the selected image I_{ks} defines an *"enhancement region"* in the 18-dimensional feature space. The probability of each image in this region is enhanced; the enhancement is largest at the center, $\mathbf{f}(I_{ks})$, and decreases as the distance from $\mathbf{f}(I_{ks})$ increases. The probabilities of images whose feature vectors are far from $\mathbf{f}(I_{ks})$ are suppressed in the probability post-normalization phase. Figure 2A shows schematically the enhancement region for a 2-dimensional feature space. This scheme can be visualized as a series of enhancement regions that get closer to the target over the iterations; they also get progressively better tuned as they converge ultimately to a small region that contains the target.

2) *Relative distance* criterion: In this scheme, the set of selected images, $\{J_{ks}\} \in D_k$ (D_k is the kth iteration's display), as well as the set of non-selected images, $\{J_{kn}\} \in D_k$, play a role in how the images of the next display are selected. To update the probability that each image $I_i \in I$ is the target image, the distance difference $d(I_i\, , I_{ks}) - d(I_i\, , I_{kn})$ is computed for every pair $\{I_{ks} \in J_{ks}, I_{kn} \in J_{kn}\}$ of one selected and one non-selected image. This difference determines, of course, whether I_i is closer to I_{ks} or to I_{kn}, and helps determine the multiplying factor, through a sigmoid function, that will update the probability that I_i is the target image.[1,15] This is best illustrated in the 2-dimensional feature space of Fig. 2B, using the simplest case of one selected and one non-selected image. The combined effect of these influences partitions the feature space via a border line. Images on one side of this line are enhanced, images on the other side are suppressed. Thus, this scheme can be visualized as a series of partitions that keep carving away the feature space to converge to a region that contains the target.

2.4 Pictorial and semantic features. The original `PicHunter`[1] was based solely on a set of 18 pictorial features. Although it is known that image annotation would improve retrieval accuracy, the state of the art

in automatic image recognition is not presently adequate to provide robust semantic labeling. One solution to this problem, adopted by Cox et al.[10], is to use *implicit annotation*, i.e., to provide semantic labeling of images but to hide the annotation from the user. This pictorial/semantic version of `PicHunter`[1,10,15] combines the hidden annotation with the simple set of 18 global image features and applies relevance feedback to refine the search. Because the annotation is hidden from the user, the annotation may be in any language, or even in symbolic form. The cost of annotation can therefore be significantly reduced by use of non-specialized workers. The use of relevance feedback compensates for inaccuracies and ambiguities in the annotation.

The semantic version was created as follows: Based on extensive experience with the pictorial database, one of us generated a list of about 125 keywords that were present in many images (such as "tree", "bicycle", "fish", "cloud", "night", etc.). Subsequently, 1,500 images were visually inspected, one by one, and the keywords corresponding to items that were present in a particular image were stored in a file associated with that image. Thus, each image was now characterized by a list of keywords, in addition to the values of the 18 image features that were stored earlier. Additional category semantic labels were automatically created by a simple OR Boolean operation, and appended to the list of keywords (for example, the "squirrel" keyword activated the "rodent" category label), resulting in a total of 134 semantic labels. In the actual implementation, there is a 134-element vector containing ones or zeroes for semantic features' presence or absence, respectively. This 134-dimensional vector was used as the 19th `PicHunter` feature, and the normalized Hamming distance between the semantic vectors was used along with the other features' distances, when computing the distance between two images.

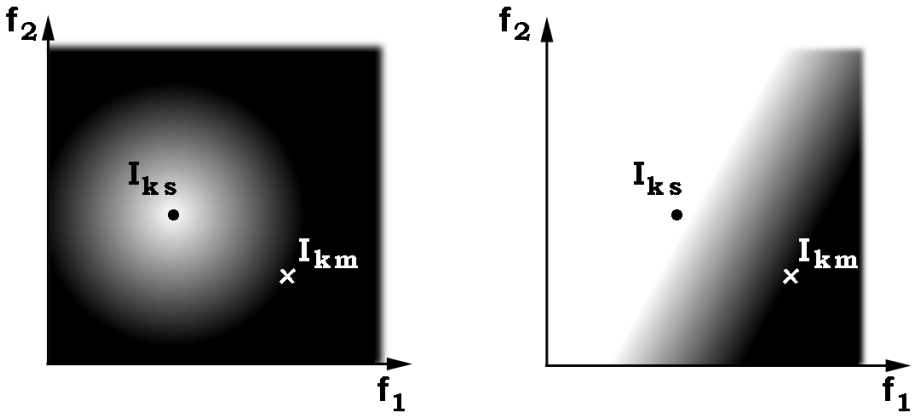

Figure 2. Illustrating the two schemes of updating probabilities based on users' feedback for a two-dimensional space, defined by features f_1 and f_2. A 2-image display is assumed (N=2), with the user having selected only one image (I_{ks}) and not the other (I_{kn}). **A.** Absolute distance criterion: The selection of image Iks in the kth iteration produces an enhancement region around the selected image's feature vector. **B.** Relative distance criterion: The space is dissected into an enhancement and a suppression region, because the user selected image I_{ks} and did not select image I_{kn} in the kth iteration.

3. PRELIMINARY EXPERIMENTS

In each of three experiments, we used one or more of the following three stimulus configurations: 1) A two-alternative forced-choice configuration, the "2AFC" configuration. Three images are presented on the screen: the target image and two test images. Throughout this section, we will refer to the target, left test, and right test images as T, L, and R, respectively; collectively, the set will be referred to as the LTR triplet. The observer had to select the test image that he/she thought was more similar to the target. 2) A "relative-similarity" configuration, with three images presented on the screen as a LTR triplet, but there are now five buttons between the bottom two test images. The observer chose one of the buttons, depending on how he/she judged the relative similarities of the two test images with respect to the target, using a 5-point scale. If one of them seemed clearly more similar to the target, he/she chose the corresponding extreme button

(left-most or right-most). If one of them was somewhat more similar to the target, he/she pressed the button immediately to the left or to the right of the center, as appropriate. If the two test images seem to be equally similar (or dissimilar) to the target, then the user chose the middle button. 3) An "absolute-similarity" configuration, involving two images and five buttons, used 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), with the intermediate three buttons indicating intermediate degrees of similarity.

3.1 Experiment A. Initially, this experiment was designed to test one of the assumptions that `PicHunter` is based on, i.e., that users' actions are largely independent of previous actions. There is some evidence that the order of presentation plays a role for textual document search.[16] The main idea, then, was to present the user with the same LTR triplet, but in different time sequences, and examine how the user's choices correlate in the two conditions. The results indicate a very good correlation among the users' responses.

3.2. Experiment B. This experiment was designed to investigate whether the judgement of image similarity obeys some form of a distance metric. We wanted to examine whether the 2AFC selection in a LTR triplet (stimulus configuration 1) can be predicted by the judgement of relative similarity of L and R with respect to T (configuration 2), and by some measure of difference between the judgements of absolute similarity of L to T and R to T (configuration 3). An additional objective was to study how consistent responses are for different sessions of the same user subjected to the same stimuli (within-user consistency), and for sessions of different users subjected to the same stimuli (across-user consistency). Accordingly, 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. 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 stimulus configurations, and these three displays were randomly scattered among the 600 trials (150 of stimulus configuration type 1, 150 of type 2, and 300 of type 3, i.e., 150 for LT and 150 for RT pairings).

In addition to re-testing the correlation between relative and 2AFC choices, as in Experiment A, we also investigated the relationship between absolute similarities and 2AFC choices. In particular, we studied whether the two independent judgements of absolute similarity between L and T, on one hand, and R and T, on the other, could be used to predict the user's selection in the 2AFC paradigm, through the use of a distance metric. Five users took part in this experiment. Some of the users ran the same experiment 3 times to allow us to examine intra- and inter-user differences. The first part is identical to the first experiment. The results from this experiment show the expected correlation and they are quite similar to those of experiment A. The second part deals with the issue whether in a 2AFC trial with LTR, the selection of L or R is governed by some type of distance metric. If this were the case, then L or R is judged to be more similar to T if $d(L,T)<d(R,T)$ or if $d(L,T)>d(R,T)$, respectively. The results across observers conform quite closely with this prediction.

3.3. Experiment C. This was designed to test whether the user can learn `PicHunter`'s user model based on immediate feedback. The user was shown a series of randomly selected LTR triplets in the configuration of stimulus 1, and was asked to indicate whether the L or R image was more similar to T, in a 2AFC paradigm. For each LTR triplet, the user-model algorithm determined which test image was closer to the target, and used this information to provide audible feedback to the user. If the user's response coincided with the algorithm's choice, a pleasant sound was provided as positive reinforcement; if the choices differed, an unpleasant sound provided negative feedback. The user was instructed to pay attention to the feedback and try to learn the criteria the computer model uses. Each session lasted for about an hour. Eight users participated in this experiment. Results were very disappointing in all cases. Performances had erratic behavior and exhibited no pattern of improvement. In fact, the patterns of different users were not correlated at all as to when some learning was taking place, even though all users ran the same sequence of trials. This experiment showed that there is no short-term learning, and its negative outcome discouraged us from investing the time to study long-term learning effects.

4. MAIN EXPERIMENTS

One property that differentiates CBIR systems from each other is the type of search goal, and the

convergence criterion that stops the search. PicHunter is characterized as an identical-target-search system, in that it has been designed to terminate its search when the *identical* target image is located; this makes it necessary that a target be selected from among the | I | images in the database I. (The way this single-target search was implemented was to have an interface module pick up a target at random, and then call PicHunter to start the search procedure.) Most CBIR systems terminate the search when an image that is *"adequately similar"* to a target is found. The use of quotation marks is intended to highlight the subjective nature of the convergence criterion, and to contrast it to the objective nature of PicHunter's goal. The second type of search is very similar to a "category search",[1] in which the goal is to find an image that belongs to a certain category of images, such as "pictures of automobiles". A third type of search is of the "open-ended" type,[1] where the user navigates through the database's images, and browses with a vague, usually pictorial, goal in mind, expecting to find images that either fit the goal or help shape a clearer goal as the search progresses. A typical example involves a homeowner browsing through a database of wallpaper designs, with the original vague goal of "some geometric design with earth colors."

One expects that performance results with runs of category-search systems have a large variability across observers as compared to identical-target-search systems. This is expected because the criterion "adequately similar" allows for a wide latitude of interpretations by different observers, whereas no such latitude is permitted when the specific target has to be found. This difference is indeed obtained in practice, as presented in subsection 4.4. Along the same lines, the variability in performance with open-ended search systems is expected to be even greater than that of category searches.

The two main experiments were conducted simultaneously. In fact, there is some overlap between the two, in that they share a common session. Nevertheless, we feel that it is best to discuss some parts of them separately, to highlight the different issues addressed in each. This is done in subsections 4.1 and 4.2, which present the two main experiments' motivation and rationale, and describe the various versions of PicHunter that were employed in each. The common experimental design is covered in subsection 4.3, and the performance results of both are supplied in subsection 4.4. Before discussing the experiments separately in subsections 4.1 and 4.2, we provide some information that applies to both.

Both experiments employed the same database of 1,500 images described in section 2.4. As described in section 2.3, the number of images the user could select depended on whether the user model employed absolute or relative distances. At most one image could be selected in the former case. After they made their selection, they pressed a "GO" button, and PicHunter reassigned probabilities to all the remaining images in the database. The user always had the option of selecting no images, which had no effect on the probability distribution, other than eliminating the images which were displayed. The next display consisted of the nine images that had the highest probability of being the target. User performance was measured by the number of iterations needed to converge to the target image.

Six naive (as to the purposes of the experiments) first-time users and two non-naive experienced users participated in these experiments. The non-naive users were two of the authors (TEC and TVP). They went through the various PicHunter versions in a random sequence, without knowing which version they were running in a given session. The results of the non-naive users are not included in the data analyses. Users appeared to have normal color vision, because they had perfect scores when tested with a series of 15 Ishihara test plates. All users also had normal or corrected-to-normal visual acuity. In both experiments, 15 different target images were used, and users searched for each of them using the same PicHunter version within each session.

4.1 Specifics on Experiment 1. This experiment's objective was to: (i) study the importance of using the memory of earlier user actions, (ii) compare models that used relative and absolute distance criteria for interpreting the user's feedback, and (iii) study the role of pictorial and semantic information for image retrieval.

4.1.1 Motivation - Rationale. To achieve the objectives above, the experiment was designed to compare search performances with several versions of PicHunter.The system's flexible design allowed the implementation of versions that differed along the desired strategic dimensions: i) Use of long-term memory of user preferences, or no memory; ii) use of absolute or relative distance criteria in judging image similarity; and iii) use of pictorial, or semantic cues, or both. The importance of each attribute was assessed based on the users' performances.

4.1.2 Methods - `PicHunter` versions. Experiment 1 involved various versions of `PicHunter`, which differed from each other along the three dimensions presented above. Two of the dimensions are binary (memory vs. no memory, relative vs. absolute distance), whereas the third has 3 options (purely pictorial features, purely semantic ones, or both). These choices define a total of 2x2x3 = 12 combinations. However, results of previous experiments indicated that some of these combinations are of no practical value. In particular, versions employing long-term memory performed better, and those with relative-distance criteria were found to be better than with absolute-distance criteria, in general. This previous experience, together with practical considerations about the length of the experiments, led to the six options shown in Table 1. All six versions were run in the identical-target-search mode, in which users were required to find the identical target image, or else the program would not allow them to move on to the next search. The only way of "skipping" a search and moving on to the next target was to press the "ABORT" button, but users were instructed not to exercise this option in Experiment 1; this option was reserved for some versions of Experiment 2, as discussed in subsection 4.2.

4.2 Specifics on Experiment 2. Experiment 2 had two subgoals: First, to establish a baseline performance for the database employed, so that `PicHunter`'s performance could be judged against it; second, to see if `PicHunter`'s solid performance as a identical-target-search system would extend if it was run in some form of category search.

4.2.1 Motivation - Rationale. Most image retrieval systems only provide qualitative measures of performance based on a somewhat vague judgment of how similar a set of retrieved images is to the target image. In contrast, an implicit assumption of the identical-target testing paradigm is that systems which are optimized using this measure will also perform well in the more general context of finding similar, rather than identical, images. Experiment 2 compared performances in the two termination conditions, namely: **2a)** when the actual target is found, as in Experiment 1; and **2b)** when an image is found that the user regards as very similar to the target.

`PicHunter` **Version⇒**	**1 MRB**	**2 MAB**	**3 NRB**	**4 NAB**	**5 MRS**	**6 MRP**
Memory	Yes	Yes	No	No	Yes	Yes
Distance Criterion	Relative	Absolute	Relative	Absolute	Relative	Relative
Pictorial, Semantic, or Both	Both	Both	Both	Both	Semantic	Pictorial

Table 1. The six versions of `PicHunter` used in Experiment 1. The 3-letter code below each version number is meant as a mnemonic for that version: The first letter is either **M** (memory) or **N** (no memory); the second is either **A** (absolute distance) or **R** (relative distance); the third is **S** (semantic), or **P** (pictorial), or **B** (both).

4.2.2 Methods - `PicHunter` versions. Experiment 2 shared version 1 of Experiment 1 (see Table 1). In addition, two more versions were tried, as shown in Table 2.

Version 7 was identical to version 1, i.e., it used the very same algorithm, but had a relaxed stopping criterion. The only difference was that users were instructed to terminate the search when they found an image, among the 9 images in the current iteration, that was "very similar" to the target. They were explained to do that by hitting the "ABORT" button. Users were deliberately not provided with any criteria for judging when two images are "very similar", because we did not want to bias them in any direction. Furthermore, one of the design principles of `PicHunter` is to have as simple a user interface as possible, thus requiring as few explanations to the user as possible.

Version 8 had the same stopping criterion as did version 7. As in that version, they were explained to terminate a search when a very similar image was found by hitting the "ABORT" button. However, this version ran under no systematic image search scheme. Unbeknown to the user, the program simply ignored all feedback provided by him/her in each iteration. Independently of the user's response in the kth iteration, it merely picked up 9 images at random (from among those that were not displayed yet) and displayed them in the (k+1)th iteration. Thus, this is not a bona-fide variation of `PicHunter`; this is why the "8" in Table 2 was enclosed in quotation marks.

PicHunter **Version ⇒**	**1 - identical MRB**	**7 - very similar MRB**	**"8" - very similar random search**
Memory	Yes	Yes	Not a PicHunter version
Distance Criterion	Relative	Relative	
Pictorial, Semantic, or Both	Both	Both	
Stopping Criterion	2a. Identical	2b. Very similar	2b. Very similar

Table 2. The two versions of PicHunter (1 and 7) used in Experiment 2, together with a random search ("8"). The stopping criterion is mentioned next to the version number. See the caption of Table 1 for the three-letter codes below each version number.

4.3 Experimental design. All 8 users ran all 8 versions, and were instructed to use the same criteria for judging image similarity throughout the sessions, as much as that was possible. The 6 naive users ran experiment 1 in a Latin-square design of 6 users × 6 versions (1 through 6; see Table 1); all 8 users also ran the two versions of Experiments 2 (7 and 8; see Table 2), with half of them running the sequence in one order (7,8) and the rest in the reverse order (8,7), to balance conditions.

The experiments were performed on a Silicon Graphics Indigo2 workstation, driving a 1280×1024-pixel color monitor, 38 cm by 29 cm, viewed from a distance of 70 cm. Individual images were 7.25×7.25 cm, padded with dark pixels, if necessary, to form square icons. The images were obtained from a set of CDs by Corel Inc.[17], where each CD contained 100 images of the same theme, such as "horses", "airplanes", "scenes from ancient Egyptian monuments", etc.

4.4 Experimental Results. The program recorded the number of iterations, or 9-image displays, that it took the user to complete each search successfully. For each observer running a particular version, we computed the performance as the average number of iterations across the 15 targets. These averages are given in Table 3 for the naive users, which also displays the average performance across observers within each version, together with the standard deviation and standard error. Figure 3 shows the data in a graph format. Table 4 shows the data for the two experienced users of PicHunter for comparison; their data were not included in the statistical analyses. It is obvious from a quick comparison of Tables 3 and 4 that knowledge of the system, particularly of its user's model, is helpful in shortening search time.

VERSION ⇒ **USER ⇓**	**1 MRB**	**2 MAB**	**3 NRB**	**4 NAB**	**5 MRS**	**6 MRP**	**7* MRB**	**8* random**
	Experiment 1							
	Exp 2						Exper. 2	
AR	24.9	29.2	44.2	26.6	10.9	32.1	6.7	8.0
BDS	30.7	33.0	36.5	29.1	14.5	35.5	21.2	17.4
SS	26.7	33.5	53.3	31.8	14.0	34.5	12.7	23.9
ABL	15.9	35.0	51.4	34.7	13.0	27.5	11.1	20.8
LF	31.5	37.9	43.1	43.9	22.7	42.3	7.7	10.5
MM	22.6	46.1	44.7	33.1	18.7	38.9	13.8	50.2
A = Average across users	25.4	35.8	45.5	33.2	15.6	35.1	12.2	19.7
SD = Standard deviation	5.75	5.79	6.08	5.99	4.30	5.17	5.20	15.7
SE = Standard error	2.35	2.37	2.48	2.44	1.76	2.11	2.13	6.39
Relative variability = SE/A	0.093	0.066	0.055	0.073	0.113	0.060	0.175	0.324

Table 3. The results of Experiments 1 and 2 for each of the 6 naive users, averaged across the 15 target images. Statistics for each version, averaged across observers, are given in the four bottom rows. The last row indicates the relative variability across users. As shown by the shadowed boxes, versions 1-6 served the purposes of Experiment 1, while versions 1, 7, and 8 served the purposes of Experiment 2. See the caption of Table 1 for the three-letter codes below each version number. In versions 1-6 users had to find the identical target, whereas they looked for an image that was very similar to the target in versions 7 and 8, indicated by asterisks.

VERSION ⇒ / USER ⇓	1 MRB	2 MAB	3 NRB	4 NAB	5 MRS	6 MRP	7* MRB	8* random
TEC	14.2	32.7	30.2	23.1	7.9	21.3	8.5	26.1
TVP	11.9	30.4	26.6	21.3	9.7	16.5	9.3	14.1

Table 4. The results of two experienced users of `PicHunter`, using the same notation as in Table 3.

For experiment 1, in which the identical-target search was employed exclusively, the following trends that are evident from a first glance at the data were also verified by an analysis of variance: 1) The use of long-term memory improves performance significantly when relative distance is employed (compare versions 1 and 3). 2) There is no improvement for the absolute-distance versions (2 and 4). A possible explanation of this strange behavior is provided in the Discussion section. 3) Remarkably, it was found that exclusive use of semantic cues (version 5) was best with 15.6 iterations, underscoring the need for annotation in image databases. 4) Among the rest of the options, performance with both pictorial and semantic cues, incorporating relative distance and memory (version 1) was best with 25.4 iterations. 5) All other options were significantly worse, averaging 33.2 or more displays, depending on the combination of features employed.

For experiment 2, which employed "very-similar"-target searches in versions 7 and 8, and identical-target search in version 1, the following are noteworthy in the results: 1) The random, similar-target search baseline performance (version 8) was found to be surprisingly good. It was even better than version 1's average; the latter, however, was an identical-target-search task. 2) `PicHunter` version 7's performance was significantly better than that of version 1, as expected. It was also significantly better than the baseline, but the improvement was not appreciably large. (See the next section for a possible explanation of items 1 and 2 above.) 3) The relative variabilities of the "very-similar"-target searches (versions 7 and 8) are significantly larger than those of the identical-target searches (versions 1-6), highlighting large inter-observer differences of similarity judgements that affected search terminations.

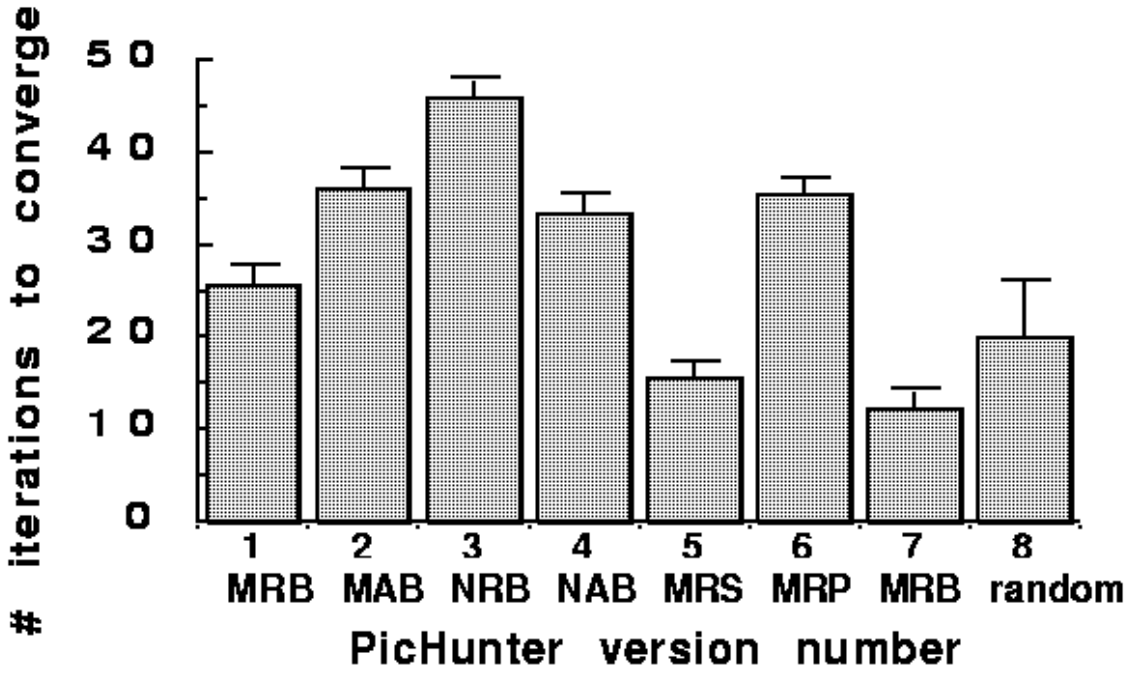

Figure 3. This graph plots the average number of iterations it took to converge to a desired image for the various `PicHunter` versions, with the bars showing standard errors. See the caption of Table 1 for the three-letter code of each version.

5. DISCUSSION - CONCLUSIONS

In general, the data form trends that are more or less predictable. However, there are some unexpected outcomes that, at first glance, seem to be counter-intuitive. We first discuss the results of the experiments to get a better understanding of the data by attempting to explain the unexpected outcomes. We begin by visiting what seems to be a paradox, namely, that long-term memory improves the relative-distance version but not the absolute-distance version (compare the difference between columns 1 and 3 to the difference between 2 and 4). It is as if the former search gets accelerated, while the latter one is delayed by comparison. If one visualizes the search in the absolute-distance versions as an enhancement region that moves toward the target over the iterations, it is as though memory adds delay by adding inertia in this motion. Indeed, since probabilities are updated by multiplying together factors in each iteration for long-memory versions, this enhancement region needs more iterations to move in a desired direction, due to the "inertia" present by the effect of all the previous iterations. Thus, even though the use of memory helps the search, this delay hinders an overall improvement. By contrast, this accumulation is helpful in relative-distance versions in which the target is approached in successively smaller partitions of the feature space. Absolute distance is taken for granted in virtually all retrieval systems besides `PicHunter`. These results show that it should be reconsidered. The results also show that these systems are justified in not using memory, at least in the way that `PicHunter` does.

The surprisingly good performance of the baseline random-search condition with the relaxed stopping criterion may be partly due to the nature of the image database, which contains thematic clusters of similar images. This clustering may explain why `PicHunter`'s performance under the same relaxed stopping criterion was not much superior to the baseline performance, since the latter did not leave much room for improvement. It remains to perform the same test with a more representative selection of images in a new database. Other open questions are how to change the single-target-search strategy of `PicHunter` into a category-search scheme, and how to modify the algorithm so as to learn a particular user's criteria for image similarity, i.e., to produce a system that is user-adaptable.

We believe that this is the first time that a baseline measurement has been performed for similar-target-search algorithms, yet it is clear that use of such baselines is essential for comparing performances of CBIR systems. In particular, our experiments demonstrate that even random search may appear to perform surprisingly well when users are given the ill-defined task of judging when images are similar rather than identical to a query.

It is extremely difficult to establish which are the most important features used by humans for judging image similarity, and to rank-order them according to their importance. Ideally, one would like to assign weights to such criteria, according to their relative contribution to the judgement of similarity.[6,11] Very little work has been done along this direction, primarily because of the enormous complexity of the problem. Multi-dimensional scaling (MDS) methods could be used toward this end, but the choice of an appropriate set of images is not clear. The closest relevant work involved the application of MDS for finding principal attributes to characterize textures[13], but the generalization to arbitrary images is not straightforward, being partly complicated by the presence of semantics. Along the same lines, it seems that CBIR systems can benefit by the use of some information on the spatial properties of images, such as location, size, shape, and color of dominant objects (or items, or "structures") in the image, distribution of spatial frequencies[18], distribution of orientations, etc., in addition to the important global color properties.

Overall, one of the main observations is that humans pay a lot of attention to semantic content in judging image similarity. Annotated images are being used regularly in specialized applications, such as medical image databases of patients in large medical institutions or archive images for news releases in news organizations, but the trend seems to extend to generic electronic images. Thus, it seems that searching for an image will have much in common with searching for text documents in library databases. In this context, future versions of `PicHunter` or other image retrieval systems may use Boolean expressions on semantics: Just as specifying such expressions when searching for a paper in the literature using a database browser, one can have self-explanatory icons (such as for animal, house, town, cloud, person, crowd, lake, mountain, etc.), and build an interface for forming Boolean expressions in a workspace for the target image, so as to start with as good an initial display as possible.

The use of icons to help initiate the search is one possible modification to the user interface. Another idea is build a front-end for PicHunter that enables the user to use a "sketch", i.e. to specify a good template so as to begin a search. For example, there can be "slide bars" to select appropriate values for each image feature, so as to start with as good of a guess in the first iteration as possible. In the extreme case, each image in each display can be accompanied by its own “signature histogram” of the 18 features, and the user could adjust the target histogram so as to “attract” similar images from the database for subsequent iterations. If spatial properties are also employed (see two paragraphs above), the use of a sketchpad on which the user can provide approximate values for the location, size, shape and color of the prominent objects will help in selecting the original N images in the initial display D_0.

Finally, a third possible modification to the user interface is to allow users to specify which feature(s) are relevant in a selected image during each iteration. In observing the users' search strategies, we noticed that a lot of them selected one of the images in the display because it was similar to the target in terms of one attribute, such as color, and another image because it was similar in another attribute, such as overall brightness. It would be informative to specify which of the features the user regards as relevant in selecting a particular image. Of course, this creates a conflict with the requirement of simplicity in the user interface, but the idea may be worth pursuing, perhaps toward a "sophisticated version" of PicHunter.

Going back to the issue of how the system starts the search by displaying a “seed display”, D_0, the standard PicHunter versions presently pick up 9 images that are randomly located in the database. However, one of the problems is that none of the 9 images may be similar to the target. Of course, the user may opt not to select any of them in the first iteration, thus forcing the system to present another set of 9 randomly selected images, until one is found that can be selected as similar to the target. An alternative strategy, which has been with tried with PicHunter, is to start the search with D_0 containing a very large number (say, 50) of judiciously selected seed images so as to give the user a wide choice on which to base the initial selection, and then to revert back to 9-image displays for each subsequent iteration. This strategy may be especially suited with databases in which images form clusters of similar images; in this case the obvious choice is to display in D_0 the images that are at the center of each cluster in the feature space.

6. ACKNOWLEDGEMENTS

The authors thank Bob Krovetz for valuable discussions on content-based retrieval systems for textual databases, Steve Omohundro for useful contributions, and Akos Feher for technical support. We appreciate the efforts of Ebony Brooks and Sejal Shah in administering experiments.

7. REFERENCES

1. I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos, "Target testing and the PicHunter Bayesian multimedia retrieval system," *Proceedings of the Forum on Research and Technology Advances in Digital Libraries*, Washington, D.C., May 13-15, pp. 66-75, 1996.
2. J. Barros, J. French, W. Martin, P. Kelly, and J. White, "Indexing multispectral images for content-based retrieval," *Proceedings of the 23rd AIPR Workshop on Image and Information Systems*, Washington, D.C., Oct. 1994.
3. A. D. Bimbo, P. Pala, and S. Santini, "Visual image retrieval by elastic deformation of object sketches," *Proceedings of IEEE Symposium on Visual Languages*, pp. 216-223, 1994.
4. G. Yihong, Z. Hongjiang, and C. Chuan, "An image database system with fast image indexing capability based on color histograms," *Proceedings of IEEE Region 10 9th Annual International Conference*, 1, pp. 407-411, 1994.
5. M. Hirakawa and E. Jungert, "An image database system facilitating icon-driven spatial information definition and retrieval," *Proceedings of IEEE Workshop on Visual Languages*, pp. 192-198, 1991.
6. K. Hirata and T. Kato, "Query by visual example: content based image retrieval," in Pirotte, A., Delobel, C., and Gottlob, G. (eds.*), Advances in Database Technology - EDBT ‘92*, Springer-Verlag, Berlin, 1992.
7. T. Kato, T. Kurita, H. Shimogaki, T. Mizutori, and K. Fujimura, "Cognitive view mechanism for multimedia database system," *IMS ‘91 Proceedings of First International Workshop on Interoperability in Multidatabase Systems*, pp. 179-186, 1991.
8. P. Kelly and T. Cannon, "Candid: Comparison algorithm for navigating digital image databases,"

Proceedings of 7th International Working Conference on Scientific and Statistical Database Management, pp. 252-258, 1994.
9. A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *SPIE Storage and Retrieval Image and Video Databases II*, 2185, 1994.
10. I. J. Cox, J. Ghosn, M. L. Miller, T. V. Papathomas and P. N. Yianilos, "Hidden annotation in content based image retrieval," *IEEE Proceedings of Workshop on Content Based Access of Image and Video Libraries*, pp. 76-81, 1997.
11. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petrovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system," *IEEE Computer*, 28, pp. 23-32, 1995.
12. T. P. Minka and R. W. Picard, "Interactive learning using a 'society of models'," *Technical Report 349*, MIT Media Lab, 1995.
13. A. Ravishankar Rao & G. L. Lohse, "Towards a texture naming system: Identifying relevant dimensions of texture," *Vision Research*, 36, pp. 1649-1669, 1996.
14. T. Kurita and T. Kato, "Learning of personal visual impressions for image database systems," *Proceedings of the Second International Conference on Document Analysis and Recognition*, pp. 547-552, 1993.
15. T. P. Minka, M. L. Miller, I. J. Cox, P. N. Yianilos, and S. M. Omohundro "Toward optimal search of image databases," manuscript in preparation, 1998.
16. M. Eisenberg & C. Barry, "Order effects: A preliminary study of the possible influence of presentation order on user judgements of document relevance," *Proceedings of the American Society for Information Science*, 23, pp. 80-86, 1986.
17. Corel stock photo library, Corel Corp., Ontario, Canada.
18. H. S. Stone and C.-S. Li. "Image matching by means of intensity and texture matching in the Fourier domain," *Proceedings of the SPIE Conference in Image and Video Databases*, 1996.