# Q & A Models for Interactive Search

Donald Geman * Roland Moquet †

December 2000

*Department of Mathematics and Statistics, University of Massachusetts, Amherst, MA 01003. Email:geman@math.umass.edu. Supported in part by the IMEDIA Research Group at INRIA, ONR under contract N00014-97-1-0249 and ARO under MURI grant DAAH04-96-1-0445.

†IMEDIA Research Group, INRIA-Rocquencourt, 78153 Le Chesnay, France. Email:roland.moquet@inria.fr

**Abstract:** We study interactive protocols to assist a person in finding a particular object in a large database, focusing on image retrieval: A person has a particular image "in mind" and responds to a sequence of machine-generated queries designed to show the target image as quickly as possible. For example, at each step the user declares which of two displayed images is "closest" to his target. The central limiting factor is the "semantic gap": The images are generally indexed by "low-level" intensity-based features rather than "high-level" semantic content. As a result, the answers are inevitably subjective and the interaction is inherently stochastic.

We explore statistical models for the "questions" and for the "answers" in the Bayesian formulation of comparison search introduced in (Cox, Miller, Minka, Papathomas & Yianilos 2000). Each new question (pair of displayed images) is chosen to minimize the expected conditional entropy of the current posterior distribution over targets given the previous answers. We introduce answer models based on independent random metrics whose distribution may depend on both the question and the target; different metrics correspond to different weightings of individual features. The modeling challenge is to account for psychovisual factors and sources of variability in human decision making. Perceptual limits and "inconsistent answers" are addressed by randomizing the choice of metric and final answers; and a logarithmic transformation accounts for the special importance of very close matches.

The resulting algorithms are demonstrated in a simplified scenario: Searching for a polygon characterized by color, size, shape and elongation. Performance is measured by the expected number of queries necessary to locate the target polygon. Data collected from human users and from simulations is analyzed in terms of two fundamental factors which reduce the flow of information: "Desynchronization" between the underlying answer model and observed human behavior and residual uncertainty in the answers given the target. Also, performance is compared with theoretical bounds and previous models. Finally, we discuss extensions to a practical search engine for large, real databases.

**Keywords:** man-machine interaction, comparison search, image retrieval, Bayesian model, decision tree, random metric

# 1 Introduction

Let $\mathcal{Y}$ denote a set of objects, such as pictures, or segments of text or sound, and suppose the user of a search engine has a subset of these "in mind." Although $\mathcal{Y}$ is

often quite large, a few objects can be presented to the user in an interactive format which allows him to make choices among them. For instance, the "retrieval system" might display several sentences or images on a computer screen and the user might click on those which are "relevant" or "closest" to his target object(s). This process of alternating between presentation and feedback continues until the user terminates the search, for instance when a target object is presented, or when an object is presented which is deemed "close enough." Or the user may simply give up. Thus minimizing the search time - number of iterations - is important.

The problem is reasonably well-understood for written documents, such as books in libraries or on web sites, and automatic indexing of text is feasible. The user might supply a list of key words and the system display the matching documents, either by searching through the original documents or through pre-processed indices (summary statistics); see (Salton 1968). Searching can be made efficient by exploiting the information residing in the statistical distribution of words and other verbal constructs.

We concentrate on images, and certain types of queries, although much of our analysis extends to other objects and protocols. Recently, the number of images stored numerically, and the number of people searching for particular ones in large databases, have grown significantly. There are many applications for interactive systems, from simple web-based "browsing" (e.g., of large and varied public databases) to more dedicated searches involving specialized material (medical records, art catalogues, criminal records, historical photographs, industrial parts, etc.). There is also a large and growing literature on image retrieval; see the recent survey (Smeulders, Worring, Santini, Gupta & Jain 2000); in fact, most of the papers are from the late nineties.

Many interactive scenarios involve rather complex "queries" submitted by the user to the system, at least in order to initiate the search; for example, the user might provide a sketch, an image, or constraints of a numerical or semantic nature. The user then plays an active role and is often assumed to have knowledge about the manner in which images are represented and processed. Such systems are difficult to analyze in mathematical terms, say with statistical or information-theoretic tools. Here, on the contrary, we consider only simple queries submitted by the system to a relatively passive user.

Decisions about what to display usually involve a "distance" $d(y, y')$ between

two images $y$ and $y'$ which is based on standard metrics adapted to the individual "features" (see below). Ideally, $d(y, y')$ is "small" when $y$ and $y'$ look alike to human beings. A variety of query types and selection criteria are possible. For example, the system might display $k$ images at each step and the user might select the one which is, *in his opinion*, closest to the specific image, or general type, he is seeking. Or the user might select a subset of those displayed. The system then displays another set of images, hopefully more homogeneous and closer to the target(s), for instance the $k$ nearest neighbors of the one selected by the user under the metric $d$. The interaction continues until a target image is displayed (and presumably recognized). Most image retrieval algorithms employ some variation of such *relevance feedback*, introduced in (Minka & Picard 1997); example systems are Photobook (Pentland, Picard & Sclaroff 1996), Pictoseek (Gevers & Smeulders 2000), Surfimage (Meilhac & Nastar 1999), PicHunter (Cox et al. 2000) and QBIC (Flickner, Sawhney, Niblack, Ashley, Huang, Dom, Gorkani, Hafner, Lee, Petkovic, Steele & Yanker 1995).

In contrast to text, images are usually not indexed by their "high-level" symbolic content since automatic indexing of this nature is currently an unsolved problem in computer vision. In particular, natural global descriptions, such as *"a large river behind a small cottage,"* are virtually impossible to match automatically to images in a large database. Instead, the images are represented by "low-level" intensity-based features, often given as a histogram, such as color (Swain & Ballard 1991), (Vertan & Boujemaa 2000), photometric or geometric invariants (Schmid & Mohr 1997), (Tuytelaars & van Gool 1999), space-frequency filters (Manjunath & Ma 1996), texture (Kankanhalli & Zhang 1994), or combinations thereof as in (Gupta & Jain 1997), (Jain & Vailaya 1996) and (Pala & Santini 1999). The user's choices are assumed to be driven by these attributes however inconclusive or unfamiliar (or even meaningless) they might be to many people. This discrepancy between numerical indexing and symbolic content is sometimes referred to as the "semantic gap." *Consequently, the answers are inevitably subjective and the interaction is inherently stochastic.* Indeed, this is the most distinctive aspect of the image retrieval problem and the motivation for a statistical approach.

## 1.1 Bayesian Framework

We adopt the Bayesian formulation in (Cox et al. 2000) based on stochastic comparison search. The user has exactly one image $Y$ in mind, and $Y \in \mathcal{Y}$. (This is called "target search"; two other prominent scenarios are "category search" and "browsing.") Several images from $\mathcal{Y}$ are displayed at each step and the user selects the one which he deems closest to $Y$. This process continues until $Y$ is among the displayed images; it is assumed the user recognizes it and the search is over. Performance is measured by the expected number of queries until $Y$ is displayed. By concentrating on the interactive process and specializing to target search and comparison tests, the authors in (Cox et al. 2000) are able to develop ties with Bayesian inference and information theory.

The simplest case, and the only one we consider, is a choice between two images $y$ and $y'$ from $\mathcal{Y}$. In (Cox et al. 2000), a distance $d$ is fixed. Were the user's choices based on $d$, the response would be $y$ if $d(y, Y) < d(y', Y)$ and $y'$ otherwise. Instead, to account for subjectivity, the authors assume a "blurring" of this response and the actual one is modeled as a random variable whose probability distribution depends on $d(y, Y) - d(y', Y)$. As in CART (Breiman, Friedman, Olshen & Stone 1984) and other tree-structured algorithms, each new query is chosen to minimize the expected conditional entropy of the distribution over targets given the previous responses. This expected entropy depends in turn on the posterior distribution - a barometer of search efficiency. These quantities are easily estimated with Monte Carlo sampling. As we shall show, when the posterior distribution becomes peaked the display items are necessarily those with relatively high mass.

## 1.2 Modeling Human Behavior

The real interactive process may be more fundamentally random than a blurred response to a fixed and known metric. We consider more general "answer models" because typical users do not have a specific metric in mind (let alone know what a metric is) and certainly not a universal one. Also, some attributes are typically weighted more heavily than others depending on the interaction between what the user has in mind and what he sees. We therefore explore behavior models based on a *sequence of independent random metrics* whose distribution may depend $y, y'$ and

$Y$. The different metrics correspond to different weightings of individual features.

Successful modeling must account for psychovisual factors and sources of variability in human decision making, and the individual models we present are but examples of how one might address these issues. We concentrate on three inter-connected factors: i) perceptual limits due to imprecise measurements of color, size and other metric attributes; ii) "inconsistent" answers due to fatigue, boredom, etc; and iii) the exaggerated importance of very close matches. The first two factors are accommodated by the randomization of the choice of the metric and by random vote-switching in order to insure that the posterior mass on the actual target remains positive. To better understand the third factor, imagine a simplified world with only four shades of grey - black, dark grey, light grey, white - with equal "spacings" among. If a person has a light grey disk in mind and is shown a dark grey triangle and a light grey square, he will likely base his answer on brightness not shape and choose the square; basically all metric-based models accommodate such behavior, assuming the attributes being matched are among those in the image representation. However, suppose the two images presented are a black triangle and a white square; the choice is then less evident even though, on a linear scale, the relative gap in brightness is the same. Put differently, there is often a premium on being "very close." This argues for adopting a logarithmic scale for similarity measurements.

We also attempt to isolate and quantify the main factors which reduce the flow of information provided by the user's answers, settling on "randomness," the amount of uncertainty in the answers knowing the target, and "synchronization," coherence of the assumed answer model with actual human behavior. The ideal state is low randomness and high synchronization, for instance when *both* query selection *and* user responses are based on the same fixed metric $d$. But this never happens, at least not in our experiments with people. Instead, our data suggest that more random models - anticipating more variability in the answers - lead to better synchronization and better performance in human searches.

## 1.3 Experiments

To facilitate data collection and quantitative analysis we specialize to images containing one simple and clearly characterized shape and texture. Specifically, we consider

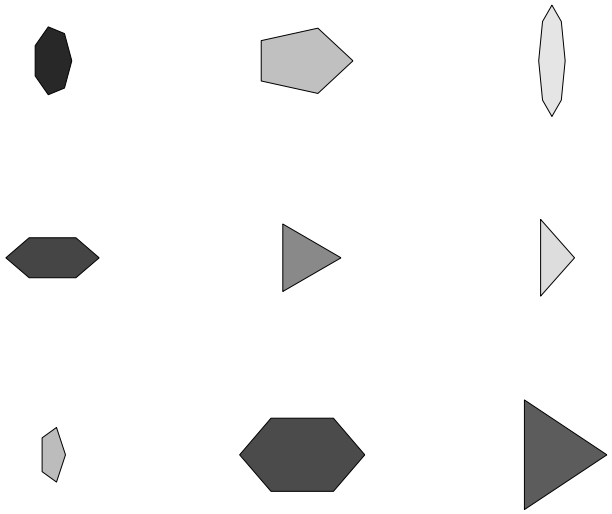

Figure 1: A sample of polygon images.

a database of polygons characterized by size, shape, color and elongation. A sample of these is shown in Figure 1 and the user interface for searching is shown in Figure 2.

Two types of experiments are reported:

**Synthetic Searches:** The responses to queries are machine-generated - according to the same model which drives query selection and evaluation of the posterior distribution. Data collected from synthetic experiments yield the search time distribution as well as the optimal performance of any given model and degree of randomness.

**Real Searches:** The feedback is provided by people, and hence there is always a degree of desynchronization between the assumed answer model and the responses of any given population of users. Indeed, the degree of desynchronization strongly affects performance; the role of randomness is less straightforward.

For instance, with a database of 200 polygons, the theoretically optimal mean search time is 5.8 (iterations). For the models we explore, the means in synthetic experiments ranges from 6.5 to 8.5, and can be ranked in accordance with residual entropy, and the means in human searches range from approximately 8.5 to 11.5 and

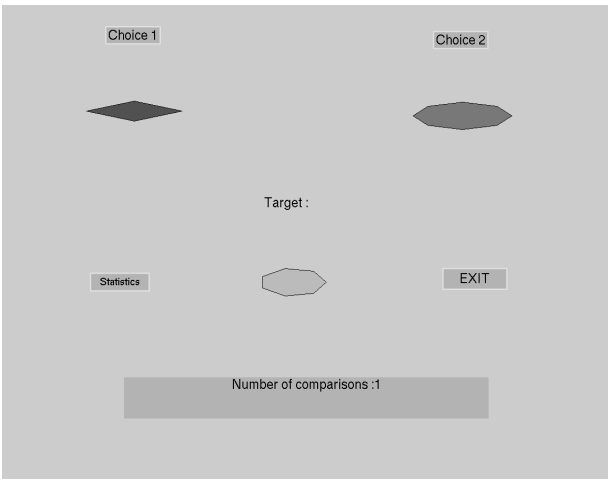


Figure 2: The user interface for experiments with polygons

can be ranked in accordance with desynchronization, but *not* with residual entropy. Of course the *relative* means among models are more instructive than the actual values. And surely all the models presented here would undergo significant alterations in any specific application; see the concluding discussion.

## 1.4 Organization of the Paper

Question and answer models are defined in §2. We specialize to comparison search in §3 and to query selection by successive entropy reduction in §4. Then, in §5, we derive a lower bound on the mean search time in the special case in which the answers, and hence the search time, are deterministic given the target. Features and random metrics are introduced in §6 and some computational issues addressed in §7, including Monte Carlo estimation of the posterior distribution and the entropy-minimizing display. In §8 various answer models are introduced, including the "IID" case, several "dedicated" models in which the distribution of the chosen metric depends on both the query and the target, and a logarithmic transformation motivated by psychovisual observations. Two useful model properties, desynchronization and randomness, are defined in §9 and then plotted against the mean search time in §10, in which experiments with polygons are presented, both synthetic ones and those based on a small subpopulation of users. Finally, in §11, we discuss our findings as well as extensions

to real image databases and the prospects for further statistical analysis.

# 2 Statistical Formulation

Each user has a single target during any given interactive session. That target is a random variable $Y$ with marginal (or "prior") distribution $p_0(y), y \in \mathcal{Y}$; we can interpret $p_0(y)$ as the fraction of potential users with target $Y = y$. This distribution will be change as information is collected from queries. In all our experiments we take $p_0$ to be uniform; however, we make no such assumption in the general development, anticipating cases in which a non-uniform prior is appropriate. This results, for example, when the representation of objects includes linguistic annotation; prior to the search, keywords supplied by the user can be matched to the elements of $\mathcal{Y}$ and a prior distribution constructed based on an appropriate linguistic metric.

We also assume that that user's response to a system query is not a deterministic function of the query and the target; instead, it is a random variable whose probability distribution will generally depend on both. In fact, our model for query selection is based on the joint probability distribution of the target and a sequence of query responses. After briefly considering a rather general framework for question and answer models, we specialize (§3) to the case of comparison queries generated by stepwise uncertainty reduction and eventually (§6) to answers based on auxiliary random metrics.

## 2.1 Question Model

Let $Q$ denote a set of possible system queries designed to solicit information about the user's target. We assume all the queries are of the same general type. The elements of $q \in Q$ might be subsets of $\mathcal{Y}$ a fixed size and the user might be asked to declare which of these are "relevant," "not relevant" or "closest" to his target $Y$. Let $A$ denote the set of possible answers to the type of questions in $Q$.

Since the interactive process is sequential and adaptive, we define a *question model* to be a family of functions

$$\Pi = \{\pi_t : A^{t-1} \to Q, \;\; t = 1, 2, 3, ...\}$$

where $\pi_1 \in Q$ represents the first question asked and $\pi_{t+1} = \pi_{t+1}(x_1, ..., x_t)$ is the question asked at step $t+1$ if $x_1, ..., x_t \in A$ are the responses to the $t$ previous questions and the target $Y$ is not yet identified. Thus a question model is simply a protocol for deciding what question to ask the user at any given step depending on the previous questions and the corresponding answers.

## 2.2 Answer Model

Let $X_q$ denote the answer to question $q$. Due to user subjectivity and variation, we regard $X_q$ as a random variable whose distribution given $Y$ remains nonsingular. An *answer model* refers to a joint conditional distribution $\mathcal{A} = \mathcal{L}(X_q, q \in Q|Y)$. (An implicit assumption is time-invariance: As in (Cox et al. 2000) the answer model is the same from one session to another and from one user to another.) The only law of importance is the one for sequence of answers determined by the protocol $(\pi_1, \pi_2, ...) \in \mathcal{Q}$. The corresponding sequence is denoted $\mathbf{X}_\pi = (X_{\pi_1}, X_{\pi_2}, ...)$ and defined recursively: $X_{\pi_2} = X_q$ where $q = \pi_2(X_{\pi_1})$; $X_{\pi_3} = X_q$ where $q = \pi_3(X_{\pi_1}, X_{\pi_2})$; and so forth.

In §4 we shall consider a particular question model based on an *assumed* answer model $\mathcal{A}$. In other words, the recipe for generating queries is determined by a particular statistical model for how the user responds. Of course in reality there is a discrepancy between the statistical properties of the assumed responses and those of the actual responses.

The basic assumption we make about the answer model is *conditional independence:* Given $Y$, the random variables $\{X_q, q \in Q\}$ are independent. This appears to be a reasonable assumption about human behavior (and is made is (Cox et al. 2000) as well) although we have not attempted to test it. Thus for any sequence of queries $q_1, ..., q_t$ and corresponding answers $x_1, ..., x_t \in A$:

$$P(X_{q_1} = x_1, ..., X_{q_t} = x_t|Y = y_k) = \prod_{s=1}^{t} P(X_{q_s} = x_s|Y = y_k).$$

In particular, the same factorization applies to $(X_{\pi_1}, X_{\pi_2}, ..., X_{\pi_t})$.

# 3 Comparison Search

From here on we specialize to the case of *comparison search*. Each query corresponds to a distinct pair $y_i, y_j \in \mathcal{Y}$; thus $Q = \{(i,j), 1 \leq i < j \leq n\}$. We refer to $y_i$ as the "left image" and $y_j$ as the "right image." The set of possible values assumed by $X_{ij}$ is $A = \{L, R, l, r\}$:

$$X_{ij} = \begin{cases} L & \text{if } Y \text{ is the left image} \\ R & \text{if } Y \text{ is the right image} \\ l & \text{if } Y \text{ is neither, but closer to the left image} \\ r & \text{if } Y \text{ is neither, but closer to the right image} \end{cases}$$

Since the search terminates with the appearance of the target among the two displayed images we have $P(X_{ij} = L|Y = y_i) = P(X_{ij} = R|Y = y_j) = 1$ and we need only consider as answer sequences finite strings from $\{l, r\}$:

$$\mathbf{x} \in \{l,r\}^* = \bigcup_{t \geq 1} \{l,r\}^t.$$

Given $\mathbf{x}$ of length $t \geq 1$, the pair displayed at time $t+1$ is denoted $\pi_{t+1}(\mathbf{x}) = (L_{t+1}(\mathbf{x}), R_{t+1}(\mathbf{x}))$. (The first pair is $\pi_1 = (L_1, R_1)$.) Therefore, since the answers are conditionally independent, we can represent an answer model as follows:

$$\mathcal{A} = \{p_{ijk}\}, \quad p_{ijk} = P(X_{ij} = l|Y = y_k),$$

where $i, j, k$ runs over all triples from $\{1, 2, ..., n\}$ with $i < j$ and $k \neq i, j$.

It is useful to visualize the question model as a quaternary tree, as illustrated in Figure 3. The four branches emanating from each vertex (i.e., internal node) correspond, from left to right, to the four answers $\{L, l, r, R\}$. Two of these symbols, $L$ and $R$, are reserved for terminal nodes, and hence only two of the four branches are developed at each vertex. Notice that each vertex at depth $t$ corresponds to a string $\mathbf{x}$ of length $t$. We will write $\mathcal{Y}(\mathbf{x})$ for the set of "active hypotheses" at $\mathbf{x}$, meaning the elements of $\mathcal{Y}$ with positive mass under the posterior distribution over $\mathcal{Y}$ given the query history at $\mathbf{x}$. More specifically,

$$\mathcal{Y}(\mathbf{x}) = \{y \in \mathcal{Y} : p_t(y|\mathbf{x}) > 0\}$$

where

$$p_t(y|\mathbf{x}) = P(Y = y|B_t(\mathbf{x}))$$

is the *posterior distribution* after $t$ queries, $\mathbf{x} = (x_1, ..., x_t)$ and $B_t(\mathbf{x}) = \{X_{\pi_1} = x_1, ..., X_{\pi_t} = x_t\}$.

We make several assumptions about the question protocol $\Pi$. In order to insure that the search terminates, every image in $\mathcal{Y}(\mathbf{x})$ must be eventually displayed at one of the terminal descendants of $\mathbf{x}$. Assume also that only images from $\mathcal{Y}(\mathbf{x})$ can be displayed at vertex $\mathbf{x}$; in particular, no image is then displayed twice. In principle this might limit the capacity to obtain "good splits" of the database but in practice it a harmless assumption and guarantees termination in at most $\frac{n}{2}$ steps (assuming $n$ is even).

The *search time* $T$ is well-defined once $\Pi$ is specified and depends on *both* the target $Y$ *and* the answer path corresponding to the query sequence. Specifically, $T = T(Y, \mathbf{X}_\pi)$ where, for a string $\mathbf{x}$ of length $\frac{n}{2}$,

$$T(y, \mathbf{x}) = \min\{t \geq 1 : y = L_t(x_1, ..., x_{t-1}) \text{ or } y = R_t(x_1, ..., x_{t-1})\}.$$

We will write $E_{\mathcal{A}}T$ for the mean of $T$ relative to an answer model $\mathcal{A}$. Efficiency of the search process is then measured by $ET$ and other distributional properties.

# 4 Stepwise Uncertainty Reduction

Setpwise uncertainty reduction is the standard recipe for building decision trees in machine learning and statistics (Breiman et al. 1984) and is the method we will use for constructing $\Pi$. Define

$$\pi_1 = \arg\min_{q \in Q} H(Y|X_q)$$

and, for $t \geq 1$ and $\mathbf{x} = (x_1, x_2, ..., x_t)$,

$$\pi_{t+1}(\mathbf{x}) = \arg\min_{q \in Q} H(Y|B_t(\mathbf{x}), X_q). \tag{1}$$

Here, $H(U|B, V)$ denotes the conditional Shannon entropy (Cover & Thomas 1991) of $U$ given $V$ under the measure $P(.|B)$, i.e., the expectation under $P(.|B)$ with respect

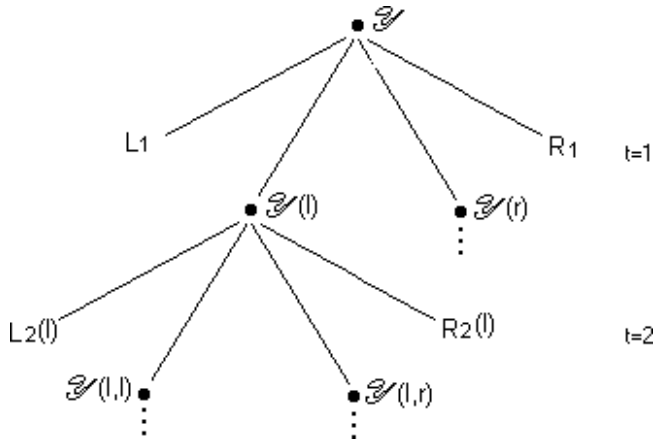


Figure 3: The query protocol represented as a quaternary tree.

to $p_V$ of $H(U|V = v)$.) In §7 we will indicate how to compute or estimate these quantities based on a sequence of random metrics.

One difference with CART and most other work on decision trees is that we do not precompute and store the entire tree (see Figure 3). Instead, only a portion of the tree is computed - the branch containing the queries indexed by the answers of the user. Consequently, this computation is performed *on-line*, i.e., during the interactive session. Another difference is that the evaluation of (1) is model-based not data-driven, although, in order to limit delays in presenting a query, the entropy-minimizing query in (1) is only estimated from a sample under the posterior; see §7.

Solving (1) for the best query $q$ at step $t+1$ is *not* equivalent to maximizing the information $H(X_q|B_t(\mathbf{x}))$ in $X_q$ given the past. Instead, one maximizes the *mutual information* $I(X_q, Y|B_t(\mathbf{x})) = H(X_q|B_t(\mathbf{x})) - H(X_q|B_t(\mathbf{x}), Y)$:

$$\begin{aligned} H(Y|B_t(\mathbf{x}), X_q) &= H(X_q, Y|B_t(\mathbf{x})) - H(X_q|B_t(\mathbf{x})) \\ &= H(Y|B_t(\mathbf{x})) - I(X_q, Y|B_t(\mathbf{x})) \end{aligned}$$

Thus minimizing uncertainty involves balancing large values of $H(X_q|B_t(\mathbf{x}))$, the information content of the next answer given the answer history, and small values of $H(X_q|B_t(\mathbf{x}), Y)$, the amount of uncertainty in the next answer given both $Y$ and the answer history. Due to conditional independence, and since we can restrict the minimization in (1) to values $q \neq \pi_1, ..., \pi_t$,

$$H(X_q|B_t(\mathbf{x}), Y) = \sum_k P(Y = y_k|B_t(\mathbf{x}))H(X_q|Y = y_k)$$

Other criteria for query selection are possible. One could, for example, sample two images from the posterior distribution $p_t(y|\mathbf{x}), y \in \mathcal{Y}$, or choose the two images with the largest posterior masses. Experiments are performed in (Cox et al. 2000) to assess the relative merits of different selection criteria in the context of one particular answer model (see §8); two conclusions are that entropy reduction is the most efficient, especially as $n$ grows large, and that choosing the most probable images under the posterior tends to give image pairs which are too similar to each other.

In fact, stepwise entropy reduction will automatically favor queries with relatively large posterior masses since termination of the search coincides with the presentation

of $Y$. Fix $q = (i, j)$ and define $Z_q = l$ if $X_q \in \{L, l\}$ and $Z_q = r$ if $X_q \in \{R, r\}$. Let $B_{tij}(\mathbf{x}) = B_t(\mathbf{x}) \cap \{Y \neq y_i, Y \neq y_j\}$. As usual, $\mathbf{x} = (x_1, ..., x_t) \in \{l, r\}^t$.

**PROPOSITION:** $H(Y|B_t(\mathbf{x}), X_{ij}) = P(Y \neq y_i, y_j|B_t(\mathbf{x}))H(Y|B_{tij}, Z_{ij})$.

**Proof:** Since $\{X_{ij} = L\} = \{Y = y_i\}$ and $\{X_{ij} = R\} = \{Y = y_j\}$, we have $H(Y|B_t(\mathbf{x}), X_{ij} = L) = H(Y|B_t(\mathbf{x}), X_{ij} = R) = 0$. Consequently,

$$\begin{aligned}
H(Y|B_t(\mathbf{x}), X_{ij}) &= \sum_{a=L,R,l,r} P(X_{ij} = a|B_t(\mathbf{x}))H(Y|B_t(\mathbf{x}), X_{ij} = a) \\
&= \sum_{a=l,r} P(X_{ij} = a|B_t(\mathbf{x}))H(Y|B_t(\mathbf{x}), X_{ij} = a) \\
&= \sum_{a=l,r} P(Z_{ij} = a, Y \neq y_i, y_j|B_t(\mathbf{x}))H(Y|B_t(\mathbf{x}), Z_{ij} = a, Y \neq y_i, y_j) \\
&= \sum_{a=l,r} P(Y \neq y_i, y_j|B_t(\mathbf{x}))P(Z_{ij} = a|B_{tij}(\mathbf{x}))H(Y|B_{tij}(\mathbf{x}), Z_{ij} = a) \\
&= P(Y \neq y_i, y_j|B_t(\mathbf{x}))H(Y|B_{tij}(\mathbf{x}), Z_{ij})\Box.
\end{aligned}$$

Of course $P(Y \neq y_i, y_j|B_t(\mathbf{x}))$ is just the posterior mass on $\mathcal{Y} \setminus \{y_i, y_j\}$ after $t$ questions. Consequently, images $y$ with "significant" mass under the posterior are often chosen as query candidates due to the multiplication by the factor $P(Y \neq y_i, y_j|B_t(\mathbf{x}))$.

# 5 The Deterministic Case

During an interactive session, exactly one path of the tree in Figure 3 is traversed from the root to a terminal node labeled by a display image. In general that path is not determined by $Y$ due to residual uncertainty in the answers; in particular, even with the same target and the same display protocol, a different path might be traversed during another session.

In this section we consider an ideal scenario in which the answers are determined by the target. Thus, $P(B_t(\mathbf{x})|Y = y) \in \{0, 1\}$, $H(X_q|B_t(\mathbf{x}), Y) = 0$, and query selection at step $t + 1$ then reduces to maximizing $H(X_q|B_t(\mathbf{x}))$ over all $q \in Q$. Notice also that $\mathcal{Y}(\mathbf{x}) \cap \mathcal{Y}(\mathbf{x}') = \emptyset$ for any two distinct vertices $\mathbf{x}$ and $\mathbf{x}'$ at the same depth.

The deterministic case corresponds to source coding for the elements of $\mathcal{Y}$ under the distribution $p_0$. Two symbols, $L, R$, from the quaternary alphabet $A = \{L, R, l, r\}$ are reserved for terminal nodes. Were $A$ *binary*, entropy reduction would amount to minimizing $|P(X_q = l|B_t(\mathbf{x})) - \frac{1}{2}|$ over available queries and the setup would be "constrained twenty questions": There is a prior distribution on "hypotheses" $y \in \mathcal{Y}$ and a distinguished set of questions of the form "*Is* $Y \in C$?" for selected subsets $C \subset \mathcal{Y}$; the problem of finding the *optimal* mean search time is well-known to be NP-complete. Were *all* such subset questions available, the optimal mean search time is provided by Huffman coding and lies in the interval $[H(p_0), H(p_0) + 1)$; in fact, the same bounds are known to hold for top-down code construction with successive entropy reduction. Comparison search is more efficient due to the larger alphabet, and one would expect mean search times of order between $H_4(p_0) = \frac{1}{2}H(p_0)$ and $H_2(p_0) = H(p_0)$ for efficient codes. Indeed, the optimal mean search time with comparison search and a uniform prior is asymptotically $\log n - 2$ (see below). (Throughout this paper log means $\log_2$.)

It is not difficult to determine the ideal protocol $\Pi$ and corresponding mean search time. Suppose $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$ where $p_0(y_1) \geq p_0(y_2) \cdots \geq p_0(y_n)$. Since $T = T(Y)$, have

$$ET = \sum_{y \in \mathcal{Y}} p_0(y)T(y)$$

where $T : \mathcal{Y} \longrightarrow \{1, 2, ...\}$ satisfies the constraint

$$\sharp\{y \in \mathcal{Y} : T(y) = t\} \leq 2^t, \;\; t = 1, 2, ...$$

(This representation of $ET$ does not exist in the general case.) The function $T$ (equivalently, the protocol $\Pi$) which minimizes $ET$ is clearly $T(y_1) = T(y_2) = 1, T(y_3) = \cdots = T(y_6) = 2$ and, in general, $T(y_i) = t$ for $i \in G_t = \{2^t - 1, ..., 2^{t+1} - 2\}$ for $t = 1, 2, ..., \phi(n) - 1$ and $G_{\phi(n)} = \{2^{\phi(n)} - 1, ..., n\}$. Here $\phi(n) = [\log(n + 1)]$, the greatest integer less than or equal to $n + 1$. Thus,

$$ET \geq \sum_{i=1}^{n} p_0(y_i)[\log(i + 1)].$$

In order to interpret this bound in coding terms, let $0 < \eta < 1$ satisfy

$$\sum_{i=1}^{n} \eta^{[log(i+1)]} = 1$$

and put

$$p^*(y_i) = \eta^{[log(i+1)]}, \;\; i = 1, ..., n. \tag{2}$$

Then

$$\begin{aligned} D(p_0 \| p^*) &= \sum_{i=1}^{n} p_0(y_i) \log \frac{p_0(y_i)}{p^*(y_i)} \\ &= -\log \eta \sum_{i=1}^{n} p_0(y_i)[\log(i+1)] + \sum_{i=1}^{n} p_0(y_i) \log p_0(y_i) \end{aligned}$$

Thus,

$$Min_{\Pi} ET = (-\log \eta)^{-1} \left[ D(p_0 \| p^*) + H(p_0) \right]$$

Since $\eta \to \frac{1}{4}$ as $n \to \infty$, the limiting bound is $\frac{1}{2}(D(p_0 \| p^*) + H(p_0))$, which is at least $\frac{1}{2} H(p_0)$ with equality if and only if $p_0 = p^*$. In other words, only for $p_0 = p^*$ does reserving $L, R$ for terminal nodes still yield an optimal code with a quaternary alphabet.

Finally, in the uniform case,

$$\begin{aligned} Min_{\Pi} ET &= \sum_{t=1}^{\phi(n)} P(T \geq t) \\ &= \sum_{t=1}^{\phi(n)} \sum_{i=2^t-1}^{n} \frac{1}{n} \end{aligned}$$

which sums to the expression in (3) below. Summarizing the above arguments:

**THEOREM:** *In the deterministic case, the mean search time is minimized by*

$$\begin{aligned} \mu^* &= \sum_{i=1}^{n} p_0(y_i)[\log(i+1)] \\ &= (-\log \eta)^{-1} \left[ D(p_0 \| p^*) + H(p_0) \right] \end{aligned}$$

*which is achieved by displaying the images with indices in* $\{2^t - 1, ..., 2^{t+1} - 2\}$ *at depth* $t$ *where* $p^*$ *is defined in (2). In the uniform case,*

$$\mu^* = \phi(n) - \frac{2}{n} \left[ 2^{\phi(n)} - \phi(n) - 1 \right] \tag{3}$$

$$= \log n - 2 + O(\frac{\log n}{n}), \;\; n \to \infty. \tag{4}$$

**Note:** It should be emphasized that these bounds could only be realized by "ideal answers" in the sense that the optimal display items at vertex $\mathbf{x}$ are indeed available; put differently, there must be enough queries to generate all possible subset questions. In particular, the first query must put half of $G_2$ on the left side and half on the right side, the queries at depth two must divide $G_3$ in exactly the right way, and so forth. Needless to say, this cannot be achieved in practice when splits (answers) are based on measuring target-to-query distances in terms of image attributes. (The case of metric-based splitting in general spaces is treated in (Yianilos 1993).) It should also be noted that this query strategy is *global* in nature and does not correspond to ideal behavior for stepwise uncertainty reduction, namely equal division of mass.

As an example, with $n = 200$, perfect splitting and a uniform prior, the distribution of $T$ is $(.01, .02, .04, .08, .16, .32, .37)$ and (3) yields $ET = 5.8$.

# 6 Features and Metrics

In order to construct answer models we need a quantitative way to compare objects. This will involve a linear combination of standard metrics over individual "features" of the objects in $\mathcal{Y}$. *From now we only consider answer variables which depend functionally on a metric.* Hopefully, the corresponding family of answer models captures how people respond to queries. We need not assume, as in (Cox et al. 2000), that there is one distinguished metric. On the contrary, we will express the answer variables in terms of a sequence of random metrics, and consequently the answer model in terms of probability distributions on a space of metrics.

## 6.1 Features

We suppose that each object $y$ is represented by a "feature vector" or "index" $f(y)$ (and that $f(y) \neq f(y')$ for $y \neq y'$ so that the elements of the database remain distinct). The objects in $\mathcal{Y}$ are automatically pre-processed and all the indices are stored. In the case of images, the features are computed from the raw intensity data and are typically grouped into broad classes which represent certain local or global characteristics of $y$. Some common examples are color histograms, Fourier or wavelet coefficients, texture attributes and edge statistics (e.g., edge orientation histograms).

The index of $y$ is then typically of the form $f(y) = (f_1(y), ..., f_M(y))$ where each feature $f_m(y)$ is a real vector say of dimension $s_m$.

## 6.2 Metrics

The distance between two images is based on the two feature vectors. Let $d^{(m)}$ be an appropriate metric on $\mathbf{R}^{s_m} \times \mathbf{R}^{s_m}$ for feature $f_m$, $m = 1, ..., M$. These metrics are fixed throughout. For simplicity we will write $d^{(m)}(y, y')$ instead of $d^{(m)}(f_m(y), f_m(y'))$. For each sequence $(\lambda_1, ..., \lambda_M)$ of positive coefficients, define a metric on images by

$$d(y, y') = \sum_{m=1}^{M} \lambda_m d^{(m)}(y, y'), \tag{5}$$

and let $\mathcal{D}$ denote the space of all such metrics generated by coefficients in $[0, 1]^M$. Distributions on $\mathcal{D}$ are then distributions on the $M$-dimensional unit cube.

For each $1 \leq i < j \leq n$, $d \in \mathcal{D}$ and $y \in \mathcal{Y}$, define

$$x_{ij}(d, y) = \begin{cases} l & \text{if } d(y_i, y) < d(y_j, y) \\ r & \text{otherwise} \end{cases}$$

Let $\{D_{ij}\}$ be a family of random variables with values in $\mathcal{D}$, let $Z_{ij} = x_{ij}(D_{ij}, Y)$ and define

$$X_{ij} = \begin{cases} L & \text{if } Y = y_i \\ R & \text{if } Y = y_j \\ Z_{ij} & \text{if } Y \neq y_i, y_j \end{cases}$$

Thus $D_{ij}$ and $Y$ together determine the answer to the question $(i, j)$.

In order to satisfy our conditional independence assumption we will suppose that $\{D_{ij}\}$ are conditionally independent given $Y$. Consequently the answer model is determined by the law $p_0$ and the marginal conditional distributions

$$\nu_{ijk}(F) = P(D_{ij} \in F | Y = y_k), \quad F \subset \mathcal{D}, \; i < j, k \neq i, j.$$

These distributions determine the answer model $\{p_{ijk}\}$:

$$\begin{aligned} p_{ijk} &= P(X_{ij} = l | Y = y_k) \\ &= P(x_{ij}(D_{ij}, y_k) = l | Y = y_k) \\ &= \nu_{ijk}(\mathcal{D}_{ijk}) \end{aligned}$$

where $\mathcal{D}_{ijk} = \{d \in \mathcal{D} : d(y_i, y_k) < d(y_j, y_k)\}$. (Notice, however, that for $k = i$, $P(X_{ij} = l|Y = y_k) = 0$ whereas $\nu_{ijk}(\mathcal{D}_{ijk}) = 1$, and similarly for $k = j$.)

# 7 Computational Issues

Computing the display protocol $\Pi$ is relatively straightforward. The two key quantities are the posterior distribution $p_t(y|\mathbf{x}) = P(Y = y|B_t(\mathbf{x}))$ and the conditional entropy $H(Y|B_t(\mathbf{x}), X_q)$. These are expressed in terms of $p_0$ and $\{\nu_{ijk}\}$ (equivalently, $\{p_{ijk}\}$) in the following two subsections.

## 7.1 Computing the Posterior

Fix $\mathbf{x} = (x_1, ..., x_t) \in \{l, r\}^t$ and $B_t(\mathbf{x}) = \{X_{\pi_1} = x_i, ..., X_{\pi_t} = x_t\}$. Let $\mathcal{W}_t(\mathbf{x}) \subset \mathcal{Y}$ be the set of images displayed prior to reaching $\mathbf{x}$, namely those images with indices $L_1, R_1, L_2(x_1), R_2(x_1), ..., L_t(x_1, ..., x_{t_1}), R_t(x_1, ..., x_{t_1})$. Notice that $B_t(\mathbf{x}) \subset \{Y \in \mathcal{W}_t(\mathbf{x})\}^c$.

Then $p_t(y|\mathbf{x}) = 0$ for $y \in \mathcal{W}_t(\mathbf{x})$ and for $y \in \mathcal{W}_t(\mathbf{x})^c$,

$$p_t(y|\mathbf{x}) = \frac{P(B_t(\mathbf{x})|Y = y)p_0(y)}{\sum_y P(B_t(\mathbf{x})|Y = y)p_0(y)}$$

with

$$\begin{aligned} P(B_t(\mathbf{x})|Y = y_k) &= \prod_{s=1}^{t} \left[ I_{\{x_s = l\}} p_{i_s j_s k} + I_{\{x_s = r\}} (1 - p_{i_s j_s k}) \right] \\ &= \prod_{s=1}^{t} \left[ I_{\{x_s = l\}} \nu_{i_s j_s k}(\mathcal{D}_{i_s j_s k}) + I_{\{x_s = r\}} \nu_{i_s j_s k}(\mathcal{D}^c_{i_s j_s k}) \right] \end{aligned}$$

## 7.2 Minimizing Target Entropy

Recall from §4:

$$H(Y|B_t(\mathbf{x}), X_{ij}) = \sum_{a=l,r} P(X_{ij} = a|B_t(\mathbf{x}))H(Y|B_t(\mathbf{x}), X_{ij} = a).$$

For $a \in \{l, r\}$,

$$P(X_{ij} = a|B_t(\mathbf{x})) = \sum_{k \neq i,j} \nu_{ijk}(\mathcal{D}^a_{ijk}) p_t(y|\mathbf{x})$$

where $A^l = A$ and $A^r = A^c$.

For the entropy term, we need $P(Y = y_k | B_t(\mathbf{x}), X_{ij} = a)$, which is zero for $k = i$ or $k = j$ and, otherwise, reasoning as above:

$$P(Y = y_k | B_t(\mathbf{x}), X_{ij} = a) = \frac{\nu_{ijk}(\mathcal{D}^a_{ijk}) p_t(y_k|\mathbf{x})}{\sum_{k \neq i,j} \nu_{ijk}(\mathcal{D}^a_{ijk}) p_t(y_k|\mathbf{x})}. \tag{6}$$

Once the next query $\pi_{t+1}$ is selected, the posterior is updated from (6):

$$p_{t+1}(y|\mathbf{x}, x_{t+1}) = P(Y = y | B_t(\mathbf{x}), X_{\pi_{t+1}} = x_{t+1}).$$

### 7.3 Monte Carlo Estimation

We approximate the posterior distribution and the solution of (1) by Monte Carlo sampling. In order to update the posterior (see §7.2), for each $y_k$, we randomly sample a fixed number of metrics in $\mathcal{D}$ (under the measure $\nu$ determined by the new query $i, j$) and count the number of these which satisfy the inequality appearing in the definition of $\mathcal{D}_{ijk}$; this gives a reasonable approximation to $\nu_{ijk}(\mathcal{D}_{ijk})$. The posterior distribution is then easily obtained by normalization.

In principle, finding the entropy-minimizing query in (1) would involve updating the posterior for each possible pair $(i, j)$ and then computing the corresponding conditional entropy. This would result in an unacceptable delay in presenting the next pair of images to the user. Instead, we sample $\eta$ pairs from the current posterior distribution and choose the pair which yields the smallest conditional entropy. In the experiments we report, $\eta = 10$, which is evidently quite small; choosing $\eta = 100$ gave slightly more efficient searches, whereas the difference between $\eta = 100$ and $\eta = 1000$ was not detectable.

## 8 Some Metric-Based Answer Models

In this section we specify a variety of system models $\mathcal{A}$ based on fixed and random metrics. It is assumed throughout that a candidate query $q = (i, j)$ at step $t + 1$ is distinct from the previous ones $\pi_1, ..., \pi_t$.

## 8.1 One Fixed Metric

This is a baseline case - deterministic answers based on one fixed metric $d^* \in \mathcal{D}$. Thus $\nu_{ijk} \equiv \delta_{d^*}$ (a point mass), $Z_{ij} = x_{ij}(d^*, Y)$ and of course $\nu_{ijk}(\mathcal{D}_{ijk}) \in \{0, 1\}$ depending on whether or not $d^*(y_i, y_k) < d^*(y_j, y_k)$. Since $Y$ determines $T$, minimizing (1) is equivalent to maximizing $H(X_{ij}|B_t(\mathbf{x}))$ and the optimal mean search time is $\log n - 2$ plus smaller order terms in the uniform case. This strategy was executed in (Tisserand & Moquet 1998) on artificial databases (points in $M$-space with Euclidean distance) for various sizes $n$, dimensions $M$, and priors $p_0$. The only meaningful parameter is $n$; in particular, the dependence on $M$ is minimal. Also, simulations yielded mean search times virtually on the curve $\log n - 2$ for $n = 100 - 4000$. Needless to say, this is a very poor model for human behavior.

## 8.2 The Model of Cox et al

There is one fixed metric $d^*$ but the "hard decision" in the deterministic case is replaced by a "soft decision" based on a sigmoid function:

$$p_{ijk} = \left[1 + \exp\left(\frac{d^*(y_i, y_k) - d^*(y_j, y_k)}{\sigma}\right)\right]^{-1}$$

where $\sigma$ is a "blur" (or "smoothing") parameter. The case $\sigma = 0$ corresponds to one fixed metric and the case $\sigma = \infty$ yields random answers. The posterior is

$$p_t(y|\mathbf{x}) \propto p_0(y) \prod_{s=1}^{t} \left[I_{\{x_s=l\}} p_{i_s j_s k} + I_{\{x_s=r\}}(1 - p_{i_s j_s k})\right] .$$

## 8.3 The IID Case

The conditional distribution of $D_{ij}$ given $Y = y_k$ is independent of $ijk$ and hence the metrics $\{D_{ij}\}$ are independent and identically distributed. (Of course $p_{ijk}$ still depends on $ijk$.) The only case we consider is uniformly distributed coefficients $(\lambda_1, ..., \lambda_M)$, so that $\nu_{ijk}(\mathcal{D}^a_{ijk}) \propto vol(\mathcal{D}^a_{ijk})$.

## 8.4 Dedicated Metrics

Whereas the answer probabilities $p_{ijk}$ evidently depend on the display pair $i, j$ and the target index $k$, there are various ways to make the distributions $\nu_{ijk}$ actually depend

on $ijk$. One can, for example, weight the attributes depending on the observed differences $|DIFF_{ijk}^{(m)}|$, where

$$DIFF_{ijk}^{(m)} = d^{(m)}(y_i, y_k) - d^{(m)}(y_j, y_k),\ m = 1, ..., M,$$

making $m$ more influential than $m'$ when $|DIFF^{(m)}| > |DIFF^{(m')}|$.

One way is to choose one of the "pure" metrics $d^{(m)}, m = 1, ..., M$, with probability proportional to $|DIFF_{ijk}^{(m)}|$; thus $\nu_{ijk}$ is concentrated on the subset $\{d^{(1)}, ..., d^{(M)}\}$. We shall refer to the corresponding answer model as "choose one feature." It leads to a closed-form expression for the answer probabilities and eliminates the need for Monte Carlo estimation. Specifically, let $F_{ij} \in \{1, ..., M\}$ be the chosen metric; then, for $k \neq i, j$,

$$\begin{aligned} p_{ijk} &= \sum_{m=1}^{M} P(X_{ij} = l | Y = y_k, F_{ij} = m) P(F_{ij} = m | Y = y_k) \\ &= \sum_{m=1}^{M} I_{\{DIFF_{ijk}^{(m)} < 0\}} \frac{|DIFF_{ijk}^{(m)}|}{\sum_{m=1}^{M} |DIFF_{ijk}^{(m)}|} \end{aligned}$$

Another way to incorporate the observed features distances into $\nu_{ijk}$ is to choose the coefficients $\lambda_1, ..., \lambda_M$ independently, with $\lambda_m$ uniform on $[0, |DIFF_{ijk}^{(m)}|]$, referred to in the experiments as "matching on a linear scale."

## 8.5 Matching on a Logarithmic Scale

People appear to be particularly influenced by *very close matches* - cases in which either $d^{(m)}(y_i, Y) \approx 0$ or $d^{(m)}(y_j, Y) \approx 0$ (but not both) for one or more attributes $m$. Similarly, there is difficulty in assessing the relative importance of two features $m$ and $m'$ when $DIFF_{ijk}^{(m)} \approx DIFF_{ijk}^{(m')}$. To see this, consider the situation in Figure 4. There are two attributes, size and brightness, say $m = 1$ and $m = 2$ respectively. In the righthand case, the display $y_j$ matches the target $Y = y_k$ exactly. Thus, $d^{(1)}(y_j, y_k) = d^{(2)}(y_j, y_k) = 0$. Clearly the user chooses $y_j$. In the lefthand case, neither feature gives a very close match and the choice is ambiguous to many users. However, the differences $DIFF_{ijk}^{(1)}$ and $DIFF_{ijk}^{(2)}$ are exactly the same in both cases when size and brightness are measured on a linear scale, i.e., with Manhattan distance.

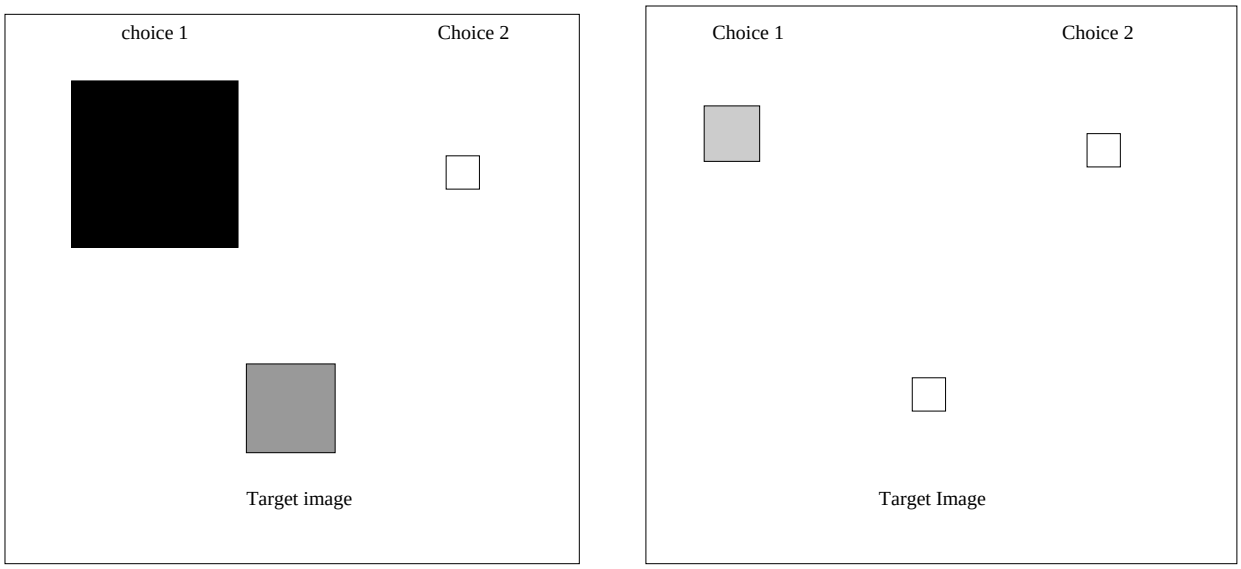


Figure 4: The target is obviously perceived as "closer" to Choice 2 than to Choice 1 in the righthand case, whereas for many people there is no clear choice in the lefthand case; however, the two cases would be identical were choices based solely on relative distances in size and intensity. This argues for measuring "closeness" on a logarithmic scale; see text.

One way to accommodate this is to choose the coefficients $\lambda_1, ..., \lambda_M$ independently with $\lambda_m$ uniform on the interval $0 \leq \lambda \leq \Psi(MIN_{ijk}^{(m)})$ for some positive, decreasing function $\Psi$, where

$$MIN_{ijk}^{(m)} = \min\{d^{(m)}(y_i, y_k), d^{(m)}(y_j, y_k)\}.$$

Another way, somewhat more transparent, is to introduce a logarithmic transformation of the basic metrics $d^{(m)}$:

$$\delta^{(m)}(y, y') = \alpha_m \log\left(1 + \beta_m d^{(m)}(y, y')\right), \ m = 1, ..., M$$

where $\alpha_m, \beta_m$ are positive parameters, the range of values assumed by $\beta_m d^{(m)}$ is large compared with one, and the $\{\alpha_m\}$ can be used to scale and weight the attributes. "Matching on a logarithmic scale" refers to this metric in the context of the IID model.

Let $\Delta_{ijk}^{(m)}$ be the corresponding difference $\delta^{(m)}(y_i, y_k) - \delta^{(m)}(y_j, y_k), \ m = 1, ..., M$. We can express it in terms of the original factors $DIFF$ and $MIN$ as follows. Suppose $\Delta_{ijk}^{(m)} > 0$. Then

$$\begin{aligned} \Delta_{ijk}^{(m)} &= \alpha_m \log\left[\frac{1 + \beta_m d^{(m)}(y_i, y_k)}{1 + \beta_m d^{(m)}(y_j, y_k)}\right] \\ &= \alpha_m \log\left[1 + \frac{|DIFF_{ijk}^{(m)}|}{\beta_m^{-1} + MIN_{ijk}^{(m)}}\right]. \end{aligned}$$

The ratio $|DIFF|/(Const. + MIN)$ favors close matches at a given value of $DIFF$.

**Example:** Suppose $\alpha_m = \beta_m = 1$ and that the brightness scale in Figure 4 is $\{1, 2, ..., 64\}$, with $y_i, y_j, y_k$ assuming the values $19, 64, 49$ respectively in the lefthand case and $49, 64, 64$ respectively in the righthand case. In both cases the discrepancy in brightness is the same, namely $DIFF_{ijk} = 15$, on the original scale. However, $\Delta_{ijk} \approx 1$ in the lefthand case and $\Delta_{ijk} = 4$ in the righthand case, which therefore strongly favors the likelihood of choosing $y_j$ for any of the models above and coheres with our perception of a less ambiguous situation.

## 8.6 Random Vote Switching

We can interpret the Cox model as one fixed metric together with display- and target-dependent flip noise. In other words, $p_{ijk} = P(\xi_{ij} X_{ij} = l | Y = y_k)$ where the $\xi_{ij} \in$

$\{-1,+1\}$ are conditionally independent with

$$P(\xi_{ij} = -1|Y = y_k) = \left[1 + \exp\left(\frac{|d^*(y_i, y_k) - d^*(y_j, y_k)|}{\sigma}\right)\right]^{-1}.$$

Consequently,

$$p_t(y|\mathbf{x}) > 0, \;\; y \in \mathcal{W}_t(\mathbf{x})^c \tag{7}$$

where $W_t(\mathbf{x})$ is the set of previously displayed images along the path to $\mathbf{x}$. In particular, $p_t(Y|\mathbf{x}) > 0$, where the posterior distribution is computed relative to the system model $\mathcal{A}$. This property acts as a safeguard against modeling errors because it prevents the user from making choices which are deemed "impossible" by the system for the user's target, hence reducing to zero the mass on that image.

This advantage is not shared by the models in §§8.3-8.5. Consider the IID model and let $\Lambda_1, ..., \Lambda_M$ be i.i.d uniform on $[0, 1]$. Then

$$p_{ijk} = \nu_{ijk}(\mathcal{D}_{ijk}) = P\left(\sum_{m=1}^{M} \Lambda_m DIFF_{ijk}^{(m)} < 0\right). \tag{8}$$

A similar expression holds if the coefficients are chosen independently on $[0, |DIFF_{ijk}^{(m)}|]$ (§8.4) with $DIFF_{ijk}^{(m)}$ in the sum in (8) replaced by $DIFF_{ijk}^{(m)}|DIFF_{ijk}^{(m)}|$. It follows that $p_{ijk} = 1$ if $DIFF_{ijk}^{(m)} < 0$ for each $m = 1, ..., M$. However, if $DIFF_{ijk}^{(m)} \approx 0$ for each $m = 1, ..., M$ the choice could be extremely ambiguous to the user, who might indeed then select $y_j$. Similarly, the sampling of $\Lambda_1, ..., \Lambda_M$ is irrelevant if $DIFF_{ijk}^{(m)} = 0$ for all but one of the coefficients, and similar remarks apply to the other models in §8.4 and §8.5.

One remedy is random vote switching: When $k \neq i, j$, the answers $\{l, r\}$ are reversed with a small probability $\epsilon$. Put differently, we "smooth" the answers by the transformation

$$p_{ijk} \longrightarrow (1 - \epsilon)p_{ijk} + \epsilon(1 - p_{ijk}).$$

In this way, (7) holds for all the models in §§8.3-8.5.

# 9 Randomness and Desynchronization

In order to capture the residual uncertainty in the answers once the target is specified, we define the *randomness* of $\mathcal{A}$ to be the average over queries $q \in Q$ of the conditional entropy of $X_q$ given $Y$. Specifically,

$$\begin{aligned} rand(\mathcal{A}) &= \binom{n}{2}^{-1} \sum_{i<j} H(X_{ij}|Y) \\ &= \binom{n}{2}^{-1} \sum_{i<j;k\neq i,j} H(p_{ijk})p_0(y_k) \end{aligned}$$

When $p_0(y) \equiv \frac{1}{n}$, this reduces to $\frac{1}{c_n}\sum_{ijk} H(p_{ijk})$ where $c_n = \frac{n(n-1)(n-2)}{2}$ and $H(p) = -p\log p - (1-p)\log(1-p)$. Small values of $rand(\mathcal{A})$ correspond to high determinism and the deterministic case is $rand(\mathcal{A}) = 0$. It is therefore reasonable to assume that small values of $rand(\mathcal{A})$ would contribute to choosing good queries *provided the user behaves according to* $\mathcal{A}$. Of course in reality - during interactive sessions with people - the answers are *not* generated according to the model $\mathcal{A}$ employed by the system for computing the posterior distribution and the corresponding expected information gain from a new query.

The answer model used by the system then represents the *predicted* behavior of the user and will be referred to as the "system model." The actual answer statistics are denoted by $\mathcal{U} = \{p^*_{ijk}\}$ and referred to as the "user model." Again, we are assuming that the choices made by and among individuals are sufficiently coherent to allow such an invariant representation. In particular, we are assuming that the likelihood that a person answers "left" given a particular target and display does not change from session to session or from user to user, so that $p^*_{ijk}$ represents the fraction of users with target $Y = y_k$ who choose "left" when presented with $y_i, y_j$ for $k \neq i, j$. The model $\mathcal{U}$ is unknown and difficult to estimate due to the large number (order $n^3$) of parameters and other factors. In our synthetic searches $\mathcal{U} = \mathcal{A}$ (although one could envision computer-generated responses according to a model $\mathcal{U} \neq \mathcal{A}$). In real searches, $\mathcal{U}$ is our model of generic human behavior relative to some subpopulation.

Finally, the *desynchronization* between $\mathcal{A}$ and $\mathcal{U}$ is defined to be

$$desyn(\mathcal{A},\mathcal{U}) = \frac{1}{c_n} \sum_{i<j;k\neq i,j} |p_{ijk} - p^*_{ijk}|$$

As with $rand(\mathcal{A})$, the normalization renders this quantity independent of the size of the database. "Full synchronization" means $desyn(\mathcal{A},\mathcal{U}) = 0$. In synthetic experiments we can estimate $E_{\mathcal{A}}T$ and even $P_{\mathcal{A}}(T = t)$ with high precision, whereas in real searches we can only estimate $desyn(\mathcal{A},\mathcal{U})$, $E_{\mathcal{U}}T$ and $P_{\mathcal{U}}(T = t)$ with low precision.

Finally, in regard to vote-switching: *For "small" $\epsilon$, the effect is to increase $rand(\mathcal{A})$ and decrease $desyn(\mathcal{A},\mathcal{U})$; however, in terms of the mean search time with people, the net effect is favorable.* In our experiments $\epsilon = 0.05$.

# 10 Experiments

We do not report experiments with real image databases, but rather with complex scenes replaced by single geometric objects - polygons. This provides a controlled setting for evaluating various models and parameters. Polygons are characterized by $M = 4$ scalar features: size, number of vertices (3 to 9), brightness ($[1, 2, ..., 255]$, and a measure of "flatness." The metrics $d^{(m)}, m = 1, 2, 3, 4$, measure absolute difference on a range of values normalized to $[0, 1]$. A sample of randomly generated polygons was given in Figure 1. A database $\mathcal{Y}$ was constructed by randomly choosing the normalized features. By way of a user interface, a randomly chosen polygon $Y$ is displayed - the bottom one in Figure 2. Thus, the user has the target $Y$ "in mind" in the very literal sense of having it persistently displayed, which is not altogether realistic. The two polygons labeled "Choice 1" and Choice 2" are $y_i$ and $y_j$; the user answers by clicking on one of them and the search ends when either "Choice 1" or "Choice 2" is $Y$.

## 10.1 Synthetic Searches

In simulations, the choices are based on very precise information, such as the exact values of $d^{(m)}(y_i, y_j)$ and $d^{(m)}(y_i, y_j)$ for each attribute, which is clearly not available to people.

In Figure 5 we display the estimated distribution of the search time $T$ for synthetic runs with $n = 200$ and the models from §8. These histograms were obtained by generating several thousand samples from $T$ for each model. The corresponding mean search times (see Figure 6) are $8.5, 8.0, 7.3, 7.2, 6.5$. This ranking coheres with that of

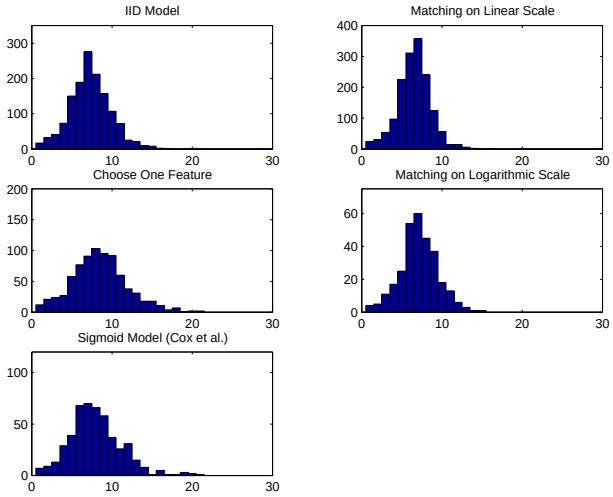


Figure 5: Histograms of synthetic search times for several models

the five values of $rand(\mathcal{A})$, which are plotted against $E_{\mathcal{A}}T$ in Figure 6. (Estimates of $rand(\mathcal{A})$ are based on 1000 triples $ijk$). In particular, the most deterministic model is "matching on a linear scale" in which the distribution $\nu_{ijk}$ actually depends on $ijk$, and this model achieves a mean search time surprisingly close to the theoretical limit of $ET = 5.8$ obtained by substituting $n = 200$ into (3).

## 10.2 Real Searches

### 10.2.1 Our User Subpopulation

We collected data from a variety of people, some with modest exposure to mathematics and computers and some with a great deal of both. The results varied enormously. Eventually we decided to limit our study to persons familiar with concepts such as "polygon" and "grey level". Each person was informed that the "system" would interpret the answers as providing information about the four polygon attributes described above, but told nothing about the various models. For each model, each person made fifty searches, each time with a new target. Since completing a search (i.e., having the target displayed) can take up to a few minutes, only a rather limited amount of data was collected - approximately 300 searches per model. This accounts for the roughness of the histograms in Figure 7. From user to user, the results were consistent in terms of the *relative* performance among models but less so in terms of

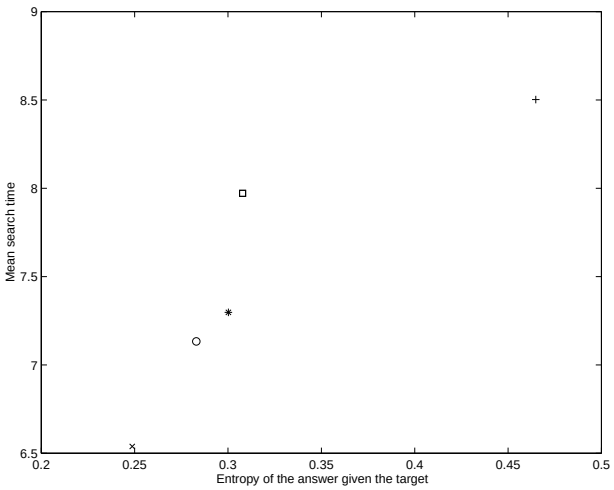


Figure 6: Dependence of performance on information content with synthetic searches. The five symbols correspond to answer models as follows: "Cox et al" ($\Box$); "IID" ($*$); "choose one feature" ($+$); "matching on linear scale" ($\times$); "matching on logarithmic scale" ($\circ$).

*absolute* performance, i.e., mean search times for a given model.

### 10.2.2 Parameter Estimation

All parameters were estimated by maximum likelihood estimation. The data was obtained by extracting the individual choices from the actual searches over sessions and models. Thus the data consist of a series of answers $x_1, x_2, ..., x_N \in \{l, r\}$ to a corresponding series of questions $(i_s, j_s, k_s), s = 1, ..., N$, where $(i_s, j_s)$ indicates the two polygons displayed in question number $s$ and $k_s$ is the index of the target at that time. Now given a model $\mathcal{S}(\theta) = \{p_{ijk}(\theta)\}$ depending on a parameter $\theta$, and assuming independent answers from trial to trial, the likelihood is

$$L(\theta; x_1, ..., x_N) = \prod_{s=1}^{N} \left[ I_{\{x_s = l\}} p_{i_s j_s k_s}(\theta) + I_{\{x_s = r\}} (1 - p_{i_s j_s k_s}(\theta)) \right] .$$

Two key parameters are $\theta = \sigma$ in the Cox model and $\theta = (\beta_1, ..., \beta_4)$ in the logarithmic model, which controls the importance of close matches. We also estimated the relative overall importance of the four attributes for our subpopulation of users. This was done by assuming that the coefficients $\lambda_m$ in the IID model were chosen uniformly

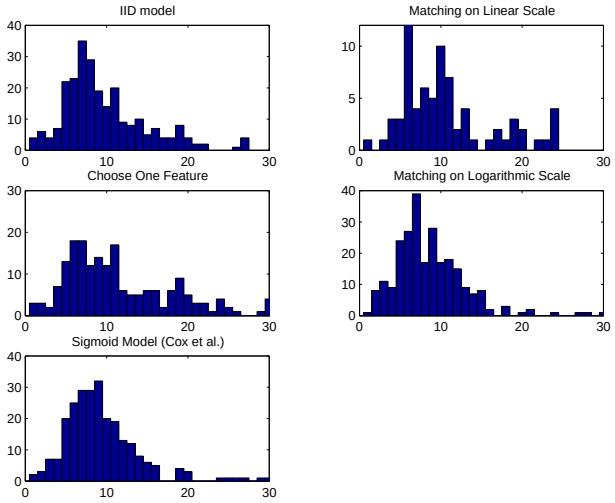


Figure 7: Histograms of human search times for several models

in the intervals $[0, w_m]$, $m = 1, 2, 3, 4$ with $w_4 = 1$. The estimated values for shape, brightness, size and elongation are, respectively, $\hat{\theta} = (1.83 : 1.50 : 1.22 : 1.00)$. These a priori weightings were then used in the experiments with the IID model.

### 10.2.3 Search Times

In Figure 7 we display the histograms of search times for the same five models as before and the effect of desynchronization on performance is illustrated in Figure 8 by plotting $E_{\mathcal{U}}T$ (the estimated mean search times for our user subpopulation) vs. $desyn(\mathcal{A},\mathcal{U})$ for the five models. The monotonic behavior in Figure 8 is present for individual users as well.

Relative model efficiency is very different than in the synthetic case. In particular, matching on a linear scale performs relatively worse in real searches than in the synthetic case, probably due to poor synchronization resulting from high determinism. The reverse is true for the Cox model (and good results are reported in (Cox et al. 2000) with real image databases). Naturally the dependence on $\sigma$ is strong. In Figure 9 we display $rand(\mathcal{A})$ and $desyn(\mathcal{A},\mathcal{U})$ for the Cox model as functions of $\sigma$. Of course $rand(\mathcal{A})$ increases with $\sigma$. The optimal degree of synchronization is comparable to using the logarithmic metric. (The poor performance of the Cox model in earlier experiments in (Geman & Moquet 2000) might be due to the choice of $\sigma$.)

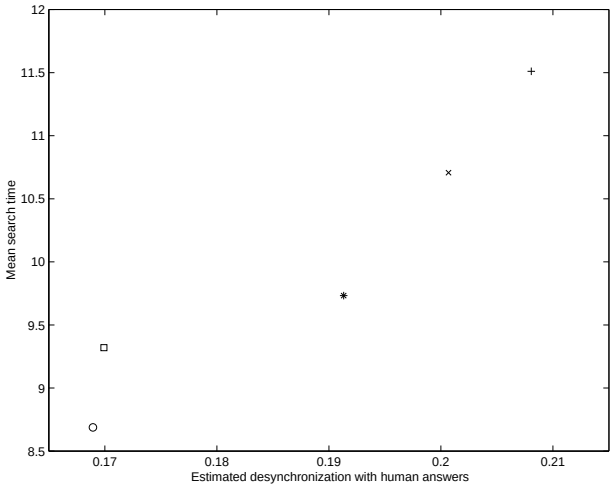


Figure 8: Dependence of performance on desynchronization with human searches. See Figure 6 for the correspondence between symbols and models.

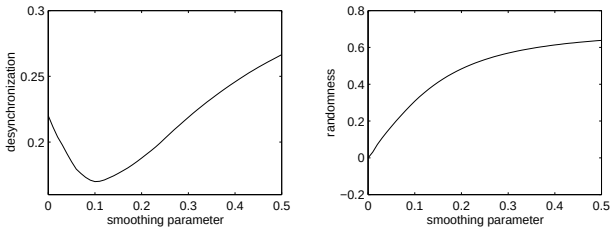


Figure 9: Dependence of desynchronization (left) and randomness (right) on the smoothing parameter in the Cox model

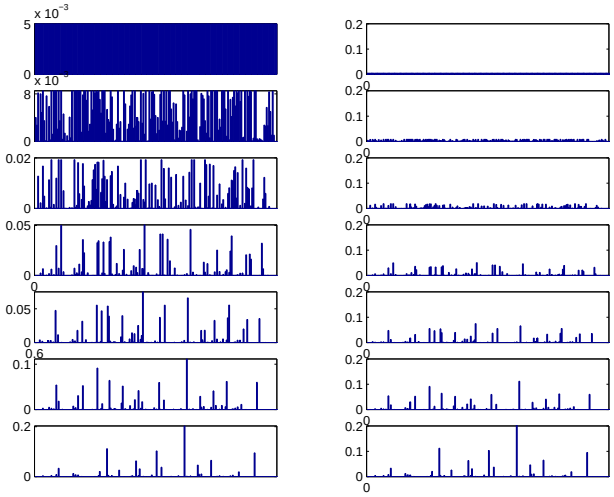

Figure 10: Evolution of the posterior distribution over six queries for a database of size 200. Results are given for one particular target, model and interactive session; however, the rate of peaking is qualitatively similar from one experiment to another. The vertical scale varies on the left and is constant on the right. The mass of one of the polygons (the actual target, which was displayed at step seven) has grown from $\frac{1}{200} = 0.005$ at the outset of the search (top) to approximately 0.2 just prior to the last query (bottom).

Finally, it is informative to observe the rate at which the posterior distribution $p_t(y|\mathbf{x})$ "peaks" as a function of the iteration number $t$. An example from one session and one model is shown in Figure 10. In Figure 11 we contrast the peaking rate for query selection by entropy minimization and random sampling from the posterior; in this case, $n = 100$ and the posterior is displayed after $t = 0, 1, 2, 3, 4$ queries. Clearly entropy minimization is more efficient, albeit more computationally intensive.

# 11 Discussion

The ultimate aim of any system model $\mathcal{A}$ is to maximize the flow of information from the user to the system at each iteration. Since all the models have approximately the same computational load, we regard the mean search time as a reasonable indicator of efficiency. We have concentrated on two factors which reduce this flow, namely

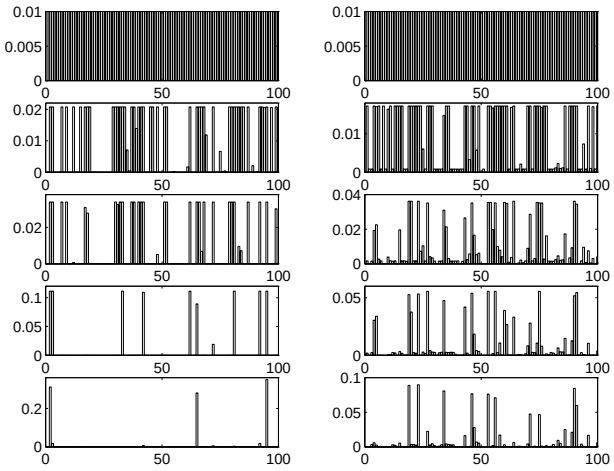

Figure 11: Evolution of the posterior distribution when queries are selected by entropy minimization (left) and random sampling (right).

$rand(\mathcal{A})$ and $desyn(\mathcal{A},\mathcal{U})$. Small values of $rand(\mathcal{A})$ often lead to fast peaking of the posterior, but also occasionally to very long search times due to the user providing answers deemed highly unlikely by the system model $\mathcal{A}$ given the actual target. Indeed, the efficiency of all the models is reduced by perceptual limits; for example, it is not clear that people can discern fine differences between the distance of the target to the left and right images, *especially when these distances are relatively large.*

It is hardly surprising that synchronization correlates highly with performance in real searches. And it seems reasonable to conjecture that the best result among all models $\mathcal{A}$ would be achieved when $desyn(\mathcal{A},\mathcal{U}) \approx 0$, i.e., with $\mathcal{A} \approx \mathcal{U}$. Also, the values of $rand(\mathcal{A})$ and $desyn(\mathcal{A},\mathcal{U})$ for the Cox model suggest the dominant role of the latter. In the case of poor synchronization, the effect of $rand(\mathcal{A})$ on performance is not clear. But in synthetic searches, $rand(\mathcal{A})$ is also a good predictor of performance.

The role of synchronization is not symmetric with respect to the system and the user. For any given user model $\mathcal{U}$, the best performance of the system is achieved with $\mathcal{A} = \mathcal{U}$. However, the converse is not true: Given that $\mathcal{A}$ is fixed, the user may fare better with a model $\mathcal{U} \neq \mathcal{A}$. One example is provided by choosing $\mathcal{A}$ to be the Cox model with $\sigma = 0.1$. In synthetic searches $E_{\mathcal{A}}T = 8.0$, whereas $E_{\mathcal{U}}T = 6.4$ for the Cox model with $\sigma = 0$ (deterministic answers based on the metric $d^*$). In this case, the high determinism outweighs non-trivial desynchronization.

The natural mathematical questions are open, and perhaps difficult to resolve. The situation is the same in other domains in which successive entropy reduction is the basis for constructing tree-structured algorithms, such as in pattern classification. It could be useful, for instance, to estimate properties of the distribution of $T$ in the synthetic case, even for $n \to \infty$ and in a purely Euclidean setting. Or to estimate $ET$ when one model drives the questions and another model drives the answers. It might also be useful to bound the difference between the conditional entropy $H(Y|B_t(\mathbf{x}))$ computed under the two models $\mathcal{A}$ and $\mathcal{U}$ starting from an estimate of $desyn(\mathcal{A},\mathcal{U})$. An intermediate step might be to bound $\|p_t^{\mathcal{A}}(.|\mathbf{x}) - p_t^{\mathcal{U}}(.|\mathbf{x})\|$, the divergence of the model posterior from the true posterior, and then use the inequality

$$|H(P) - H(Q)| \leq -\|P - Q\| \log \frac{\|P - Q\|}{n}$$

where $P, Q$ are probability measures on $\{1, 2, ..., n\}$. The efficiency of query selection should depend on the rate of separation between the two posteriors as a function of search length $t$.

Experiments with real and heterogeneous image databases are preliminary. The ones we have done - within the IMEDIA research group at INRIA-Rocquencourt - involve standard features such as color and edge statistics with individual dimensions on the order of 100. Elementary comparison search needs to be modified in order to accommodate clustering of the images in $\mathcal{Y}$ into distinct groups. When the user is presented with two images, both very different from his target, his answer is virtually random. In searching for a forest scene, how does he choose between a calm sea and a brick wall? Yet for certain metrics in $\mathcal{D}$, the sea or wall might be much closer to the forest image and hence the feedback is misleading. Simply allowing the user to choose neither image (and updating the posterior by eliminating the two displayed images and renormalizing) significantly reduces the mean search time.

Moreover, there appears to be an initial, inefficient, search for the right "cluster" - the one containing the target - and then a rather efficient, within-cluster search somewhat similar to that with polygons. One way to reduce the first stage is to increase the number of images presented at each iteration. For instance, displaying four images, and allowing the user to select any subset, reduces the average length of the search. A more direct extension of the framework here is to display $k$ images at each iteration and ask the user to declare which, if any, is the target and otherwise

which is closest to the target; hence there would be $2k$ possible responses and ideally one would expect search times on the order of $\log_k n$, although efficient query selection would evidently be a problem.

Finally, from one perspective we have put a magnifying glass over one aspect of a large and diverse subject. Indeed, our analysis might appear highly "elaborate" to practitioners since that adjective was applied in (Smeulders et al. 2000) to the less rigorous Bayesian analysis in (Cox et al. 2000). Our motivation is that interactive search is an important and inherently stochastic process, and yet much of the work in image retrieval (as opposed to the more sophisticated state of text retrieval) may not be amenable to a satisfying statistical analysis, due mainly to the complexity of the interaction between the system and user. As the field "shakes out," and certain interactive protocols are shown to handle limiting factors such as large databases, impatient users and the "semantic gap," it should become clearer how statistical modeling and reasoning can contribute to performance.

# Acknowledgements

The work began during an internship at the University of Massachusetts by Matthieu Tisserand and the second author, supported by Ecole Polytechnique. Matthieu made numerous contributions. So has the IMEDIA Research group at INRIA-Rocquencourt where experiments are underway on their state-of-the-art image retrieval platform. Finally, special thanks to A. Kolydenko, G. Garibotti, J. Fernandez and others for cheerfully accepting the eventually boring task of generating data.

# References

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984), *Classification And Regression Trees*, Wadsworth, Statistics/probability series.

Cover, T. & Thomas, J. (1991), *Elements of Information Theory*, John Wiley.

Cox, I., Miller, M., Minka, T., Papathomas, T. & Yianilos, P. (2000), 'The bayesian image retrieval system, pichunter: theory, implementation and psychological experiments', *IEEE Trans. Image Processing* **9**, 20–37.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. & Yanker, P. (1995), 'Query by image and video content: the qbic system', *IEEE Computer* **28**, 23–32.

Geman, D. & Moquet, R. (2000), A stochastic feedback model for image retrieval, *in* 'Proc. RFIA 2000', Paris.

Gevers, T. & Smeulders, A. (2000), 'Pictoseek: Combining color and shape invariant features for image retrieval', *IEEE Trans. Image Processing* **9**, 102–119.

Gupta, A. & Jain, R. (1997), 'Visual information retrieval', *Comm. ACM* **40**, 71–79.

Jain, A. & Vailaya, A. (1996), 'Image retrieval using color and shape', *Pattern Recognition* **29**, 1233–1244.

Kankanhalli, A. & Zhang, H. J. (1994), Using texture for image retrieval, *in* 'Proc. ICARCV '94'.

Manjunath, P. & Ma, W. (1996), 'Texture features for browsing and retrieval of image data', *IEEE Trans. PAMI* **18**, 837–842.

Meilhac, C. & Nastar, C. (1999), Relevance feedback and category search in image databases, *in* 'Proc. Inter. Conf. Multimedia Computing and Systems', pp. 512–517.

Minka, T. & Picard, R. (1997), 'Interactive learning using a society of models', *Pattern Recognition* **30**.

Pala, P. & Santini, S. (1999), 'Image retrieval by shape and texture', *Pattern Recognition* **32**, 517–527.

Pentland, A., Picard, R. & Sclaroff, S. (1996), 'Photobook: Content-based manipulation of image databases', *Int'l J. Computer Vision* **18**, 233–254.

Salton, G. (1968), *Automatic Information Organization and Retrieval*, McGraw-Hill, New York.

Schmid, C. & Mohr, R. (1997), 'Local grayvalue invariants for image retrieval', *IEEE Trans. PAMI* **19**, 530–535.

Smeulders, A., Worring, M., Santini, S., Gupta, A. & Jain, R. (2000), 'Content-based image retrieval at the end of the early years', *IEEE Trans. PAMI* **22**, 1348–1375.

Swain, M. & Ballard, B. (1991), 'Color indexing', *Int'l J. Computer Vision* **7**, 11–32.

Tisserand, M. & Moquet, R. (1998), A flexible algorithm for image retrieval, Technical report, Ecole Polytechnique, Palaiseau, France.

Tuytelaars, T. & van Gool, L. (1999), Content-based image retrieval based on local affinely invariant regions, *in* 'Proc. Visual '99: Information and Information Systems', pp. 493–500.

Vertan, C. & Boujemaa, N. (2000), Upgrading color distributions for image retrieval: Can we do better?, *in* 'Advances in Visual Information Systems. Proc. 4'th Int'l Conf., VISUAL 2000', Springer.

Yianilos, P. N. (1993), Data structures and algorithms for nearest- neighbor search in general metric spaces, *in* 'Proc. Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)'.