

Adaptive Relevance Feedback for Large-scale Image Retrieval

Nicolae Suditu · François Fleuret

Received: date / Accepted: date

Abstract Content-based image retrieval aims at substituting traditional indexing based on manual annotation by using automatically-extracted visual indexing features. Novel techniques are needed however to efficiently deal with the semantic gap (i.e. the partial match between the low-level features and the visual content). Here, we investigate a query-free retrieval approach first proposed by Ferencat and Geman. This approach relies solely on an iterative relevance feedback mechanism that drives a heuristic sampling of the collection, and aims to take explicitly into account the semantic gap.

Our contributions are related to three complementary aspects. First, we formalize a large-scale approach based on a hierarchical tree-like organization of the images computed off-line. Second, we propose a versatile modulation of the exploration/exploitation trade-off based on the consistency of the system internal states between successive iterations. Third, we elaborate a long-term optimization of the similarity metric based on the user searching session logs accumulated off-line. We implemented a web-application that integrates all our contributions, and distribute it under the AGPL Version 3 free software license.

We organized user-based evaluation campaigns using ImageNet dataset, and show empirically that our contributions significantly improve the retrieval performance of the original framework, that they are complementary to each other, and that their overall integration is consistently beneficial.

Keywords Content-based image retrieval · Query-free · Large-scale · Interactive relevance feedback · Adaptive exploration/exploitation trade-off · Log-based similarity learning · Multi-modal indexing features · User-based evaluation

1 Introduction

The trend in recent years shows that image retrieval needs are evolving beyond the capabilities of traditional indexing based on manual annotation, and that the most desirable characteristic of any image retrieval system is to deal with automatically-extracted visual indexing features, while providing an intuitive and simple interaction with users.

The expansion of the World Wide Web, accompanied by inexpensive recording capabilities, mass storage and sharing tools, facilitate public access to collections of unprecedented size. Facebook, Flickr, and less-known on-line repositories such as Image Shack, host billions of images and keep growing at fast rates. We adhere to the assumption that the collections are not only large but also inherently un-structured (i.e. lacking a reliable semantic or thematic indexing) and are continuously out-dated (i.e. images are frequently being added, replaced or removed).

Research has begun to tackle these challenges via automatic tagging based on annotation propagation [24, 20, 29]. However, formulating a query might not be the most efficient way of searching for images since the visual content is often difficult to describe in terms of keywords. Relevance feedback is envisioned by many researchers as the most appropriate alternative to cope properly with the challenges in image retrieval, and multimedia retrieval in general [22, 30].

We investigate an innovative query-free retrieval approach first proposed by Ferencat and Geman [13, 14]. Starting from an heuristic sampling of the collection, this approach does not require any explicit query, neither keywords nor image-examples. It relies solely on an iterative relevance feedback mechanism driven by the user's subjective judgments of image similarities. At each iteration, the system displays a small set of images and the user chooses the image that best matches in her opinion what she is searching for. The system updates an internal state based on automatically-extracted indexing features, and displays a new set of images accordingly. The system iteratively converges towards what the user is searching for, and displays more and more relevant images.

Our contributions are related to three complementary aspects of the iterative relevance feedback mechanism. First, we formalize a large-scale approach based on a hierarchical tree-like organization of the images computed off-line. Second, we propose a versatile modulation of the exploration/exploitation trade-off based on the consistency of the system internal states between successive iterations. Third, we elaborate a long-term optimization of the similarity metric based on user searching session logs accumulated off-line. We rounded up our research by integrating all our contributions together into one comprehensive retrieval system.

Experimental validation was carried out by implementing a web-application which includes all our contributions. This software is distributed to the public under the AGPL Version 3 free software license. We carried out plenty of user-based evaluation campaigns with two collections from the ImageNet

dataset [10], a large collection of 1,000,000 images and a small collection of 60,000 images, for which we acquired the provided pre-computed SIFT features (Scale Invariant Feature Transform) [21]. We systematically analyzed all our contributions, and got evidence that each of them significantly improves the retrieval performance of the original framework. Moreover, empirical evidence shows that they are complementary, and their integration is consistently beneficial.

In § 2 we motivate our research by an overview of the relevant state-of-the-art, and describe in § 3 the retrieval framework that is central to our work. Next, we elaborate our contributions: the large-scale HEAT framework in §4, the exploration/exploitation trade-off in §5, and the log-based similarity metric in §6. Our experiments and user-based evaluations are in §7.

2 Related work

While relevance feedback is a very efficient solution for content-based image retrieval, there are still many open questions in both the perceptual, cognitive and the algorithmic, technical aspects. Regarding the cognitive aspect, novel similarity measures or rankings are needed to better capture the human perception of image similarities. Regarding the technical aspect, novel algorithms are needed to compute or efficiently approximate such similarity measures at a large-scale [22, 30, 9]. Moreover, the relevance feedback is usually seen as a post-retrieval mechanism for refining the retrieved results of an initial query formulated explicitly via *query-by-example*[25] or *query-by-sketching*[15]. There are early works like MARS[5] and MindReader[19] that develop mechanisms for rich feedback information (e.g. ranking many images, tuning many parameters).

Our research is related to the innovative idea of searching images without any explicit query, which was pioneered by Cox et al. [8]. The core of their work is a Bayesian framework for iterative relevance feedback. Ferecatu and Geman [13, 14] extended the framework and provided theoretically sound interpretations. Moreover, they conducted user evaluations that demonstrate the retrieval capabilities of such an approach. One can have a broader context of the query-free retrieval by studying the perception-based image retrieval system developed by Chang et al. [6]. In essence, the system models the user retrieval needs as feature grouping of k -CNF/DNF Boolean form, and requires an iterative rich relevance feedback consisting of positive and negative labeled images. The ostensive models proposed by Campbell and Rijsbergen [4] and more recently by Urban and Jose [28] share the assumption that the user information need is dynamic and developing, and therefore they include a forgetting process. The retrieval approach of Ferecatu and Geman [13, 14] replaces a few iterations of rich feedback by multiple iterations of a minimalist feedback (i.e. one positive example) which is intuitive, simple, efficient and robust to noisy relevance feedback.

Ω	image collection, where the images are identified by their indexes $\{1, 2, \dots\}$
$S \subset \Omega$	set of images that the user is searching
$D_t \subset \Omega$	set of images shown to the user at iteration t
$x_t^* \in D_t$	image chosen by the user at iteration t
$\{D_t, x_t^*\}$	relevance feedback event at iteration t
$p_t(k)$	probability of relevance of image k at iteration t
\mathcal{N}	nodes of the partitioning tree
$C(N)$	children nodes of node N
$\Omega(N)$	set of images associated with node N
m_t	target mass for building the display set in the original system
m_t^{zoom}	target mass in the mass-zoom approach
z_t	change of the target mass at iteration t
α	weighting vector learned off-line from the user logs
$d_\alpha(k, h)$	weighted Euclidean distance between k and h
C	cost function defined on the probabilities of relevance of the images indicated by the users

Table 1 Notation

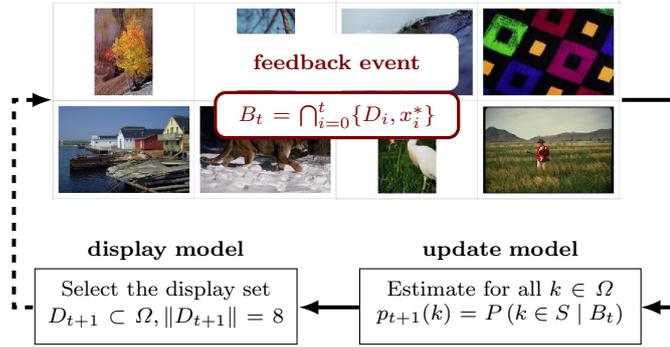


Fig. 1 Relevance feedback loop. At iteration t the system displays D_t . The next iteration $t+1$ is triggered by the relevance feedback event $\{D_t, x_t^*\}$. The system will update $p_{t+1}(k)$ for all $k \in \Omega$, and will then select the new display set D_{t+1} .

Motivated by the potential of this query-free retrieval approach, we elaborated a large-scale HEAT framework in [26], and proposed an adaptive exploration/exploitation trade-off approach in [27]. Here we will extend these precursors, and will elaborate a long-term optimization of the similarity metric based on the user searching session logs accumulated off-line. These three contributions are complementary to each other and result into one comprehensive retrieval system.

Our large-scale HEAT framework [26] uses a hierarchical tree-like organization of the image collection. Although hierarchical trees have been extensively used for zoom-able user interfaces as PhotoMesa [2] and many other browsing solutions [17], to the best of our knowledge there is no system that uses such a concept in order to scale up relevance feedback mechanisms. The idea of a dynamically adaptive and traceable cut within a hierarchical tree-like organization has been used in the field of information visualization and visual data mining, where it is referred to as a tree map [23], and other equivalent terms like fish-eye [1] or tree view [3].

Although there are plenty of sophisticated similarity metrics [11, 9], it has been recognized that it is too ambitious to expect a single automatically-derived metric to model reasonably well the user perception of image similarity. The similarity metrics should go beyond the low-level automatically-derived indexing information and should model explicitly the user perception. Extensive research has been done in order to derive similarity metrics based fully on user input such as for example relative similarity of pairs of images [7]. Unfortunately, collecting such user input is as prohibitive as the traditional manual annotation of the images, and is not suitable for large-scale collections. An interesting alternative is to attempt to tune an existing automatically-generated similarity metric by learning from the user feedback. The use of relevance feedback for learning the correlation between low-level indexing features and high-level semantics has been attempted in Han et al. [16] and Hoi et al. [18]. Our retrieval framework is particularly feasible for such machine-learning approaches that require user logs information (i.e. image labeling) since the relevance feedback is the core mechanism of searching.

3 Query-free retrieval framework

This section presents the retrieval framework proposed in [14]. Given a collection of images $\Omega = \{1, 2, \dots\}$, the objective of the retrieval process is to identify the small subset $S \subset \Omega$ of images that the user is looking for. Notation is shown in Table 1.

The retrieval framework embodies an iterative relevance feedback mechanism that has two components. First, there is a Bayesian framework that models the probabilities of relevance of the images in the collection as conditional probabilities, given the relevance feedback events. Second, there is a strategy for selecting what images to show next given the estimates of the probabilities of relevance of all the images in the collection.

For an intuitive illustration of the system behavior, a synthetic collection it comes in handy, where each image has two indexing features in $[0, 1]$. These features are interpreted as coordinates in the 2D Cartesian space, and are used in a dual manner. On the one hand, the features define the image visual content as a single point positioned accordingly. On the other hand, the features define the position of the image itself in the landscape of the entire collection. Additionally, we can display the probabilities of relevances over the full collec-

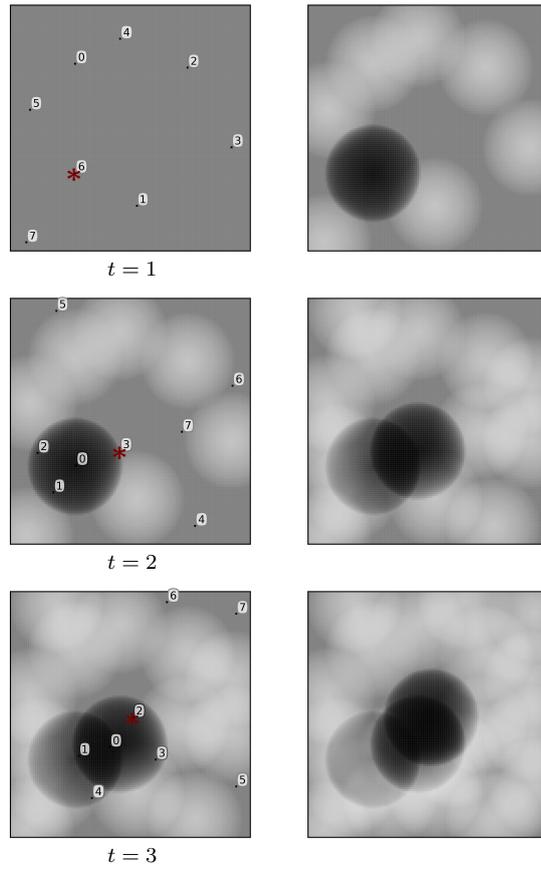


Fig. 2 The posterior probabilities $p_t(k)$ for all $k \in \Omega$ are updated iteratively. Here, the relevance feedback events are given by a user who is searching for images with points close to the center. One can see how the distribution of probabilities evolves towards matching the user retrieval objective.

tion as a single picture by associating to every point a gray level corresponding to the said probability.

3.1 Posterior probabilities of relevance

Relevance feedback events are accumulated iteratively as shown in Figure 1. At iteration t , after the system displays a set of images $D_t \subset \Omega$, $\|D_t\| = 8$, the user chooses one single image $x_t^* \in D_t$ that she considers to be the closest to S (i.e. the set of images that she is looking for), and this event is denoted as $\{D_t, x_t^*\}$. The cumulative event up to iteration t can be expressed as:

$$B_t = \cap_{i=0}^t \{D_i, x_i^*\} \quad \forall t \geq 0. \quad (1)$$

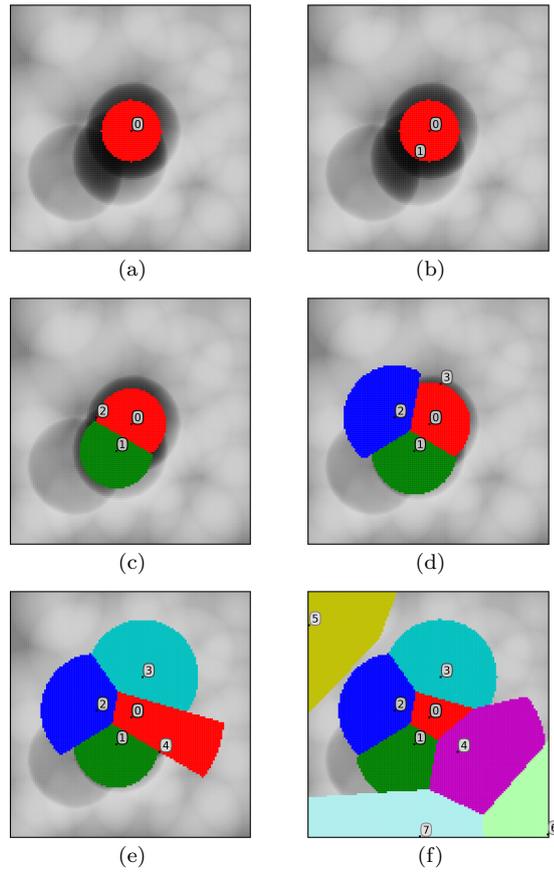


Fig. 3 The set of displayed images is generated via the Voronoi tessellation algorithm. To illustrate its intermediate steps, the images already selected are marked in black and their current Voronoi cells are indicated by colors. (a): The first image $x^{(0)}$ is selected, and the first Voronoi cell $\mathcal{C}^{(0)}$ is grown. (b): The second image $x^{(1)}$ is selected. (c): The Voronoi cells $\mathcal{C}^{(0)}$ and $\mathcal{C}^{(1)}$ are grown in parallel. $\mathcal{C}^{(0)}$ is shrunken by detaching the images closer to $x^{(1)}$, and then re-grown by including other images that are still closer to $x^{(0)}$. (d-f): The algorithm proceeds in the same manner until the display set is complete.

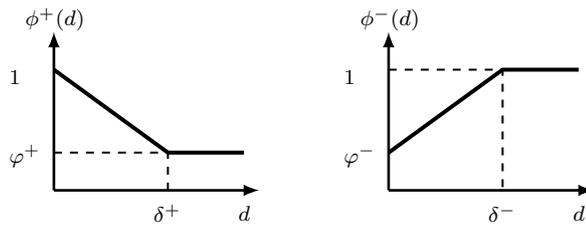


Fig. 4 Calibration functions and their parameters. δ^+ , δ^- are the thresholds that normalize the distances, and φ^+ and φ^- are the attenuations that compensate for the semantic gap as explained in [13].

The conditional probabilities $p_{t+1}(k) = P(k \in S | B_t)$ are estimated after each relevance feedback event. Initially, when there is no relevance feedback yet, the probabilities $p_0(k)$ are initialized with 0.5 for all $k \in \Omega$. Subsequently, the conditional probabilities are estimated via an image similarity model defined over the metric space of the indexing features. Before we return to this issue in §3.2, we further elaborate the Bayesian modeling.

Assuming that the events $\{D_t, x_t^*\}$ are conditionally independent from each other given the retrieval objectives, and using Bayes theorem, $p_t(k)$ can be expressed recursively:

$$p_{t+1}(k) = \frac{p_t(k) \cdot P_t^+(k)}{p_t(k) \cdot P_t^+(k) + (1 - p_t(k)) \cdot P_t^-(k)}, \quad (2)$$

where

$$P_t^+(k) = P(\{D_t, x_t^*\} | k \in S), \quad (3)$$

$$P_t^-(k) = P(\{D_t, x_t^*\} | k \notin S). \quad (4)$$

One may observe that the probabilities in Equations (3-4) should model as much as possible the user similarity judgments, and the better the model, the more reliable the relevance feedback. We shall return to this issue in §3.2.

Figure 2 shows how the probabilities of relevance are gradually updated on successive iterations. We can see how the system is calibrated in such a way that images closer to the indicated image get higher probabilities and images closer to the other displayed images get lower probabilities. The images far from any of the displayed images keep their probabilities unchanged.

3.2 Similarity metric

The probabilities $P_t^+(k)$ and $P_t^-(k)$ in Equations (3-4) are modeled based on a similarity metric defined over the image feature space as in [14], which puts higher probability on the images similar to the chosen ones and accounts for an effect of “saturation” that ignores the increase in the image dissimilarities beyond a certain threshold:

$$P_t^+(k) = \frac{\phi^+(d(k, x_t^*))}{\sum_{x \in D_t} \phi^+(d(k, x))}, \quad (5)$$

$$P_t^-(k) = \frac{\phi^-(d(k, x_t^*))}{\sum_{x \in D_t} \phi^-(d(k, x))}. \quad (6)$$

The distance d between the images is the L^2 norm between the image feature vectors as in Equation (7).

$$d(k, h) = \sqrt{\sum_{f=1}^F (k_f - h_f)^2}, \quad (7)$$

```

Function ComputeDisplaySet( $\mathbf{p}, Q, m$ )
for  $q = 1, \dots, Q$  do
   $\mathcal{C}^{(1)}, \dots, \mathcal{C}^{(q-1)} \leftarrow \mathbf{ComputeCells}(\mathbf{p}, x^{(1)}, \dots, x^{(q-1)}, m)$ 
   $x^{(q)} \leftarrow \operatorname{argmax}_{k \in \Omega \setminus \cup_{i=1}^{q-1} \mathcal{C}^{(i)}} p(k)$ 
end for
return  $x^{(1)}, \dots, x^{(Q)}$ 

Function ComputeCells( $\mathbf{p}, x^{(1)}, \dots, x^{(i)}, m$ )
return  $\mathcal{C}^{(1)}, \dots, \mathcal{C}^{(i)}$ 
  s.t.  $\forall q \sum_{k \in \mathcal{C}^{(q)}} p(k) = m$ 
  and  $\forall q, r \neq q, \forall k \in \mathcal{C}^{(q)} \quad \|k - x^{(q)}\| \leq \|k - x^{(r)}\|$ 

```

Table 2 Procedures to compute a meaningful display set D_t .

where F is the dimensionality of the image feature space. As we explain in §7, our experiments use bags-of-words based on SIFT, but any other indexing feature vectors will do.

ϕ^+ and ϕ^- are calibration functions designed to capture the user perception of image similarities and error-prone decision-making behavior. We consider calibration functions of parametric form as shown in Figure 4 reasoning that δ^+ , δ^- are thresholds beyond which the L^2 norm fails to resemble the user perception, and φ^+ and φ^- are attenuations that compensate for the partial mismatch between the distances and the user perception of similarities (i.e. the semantic gap).

3.3 Selection of the displayed images

Ideally, each next display set D_{t+1} should maximize the flow of information from the user to the system, and therefore should minimize the uncertainty about S given the relevance feedback history B_t and the new evidence x_{t+1}^* that would be provided on D_{t+1} itself. This optimization is intractable since it implies looping over all possible subsets in Ω .

$$D_{t+1} = \operatorname{argmin}_{D \in \Omega} H(S | B_t, \{D, x^*\}). \quad (8)$$

In our system, the display set D_t is generated via a Voronoi tessellation algorithm proposed by Fang and Geman [12] that approximates the ideal intractable case. The algorithm selects the images in D_t by growing Voronoi cells based on the image similarity distances and their current probabilities of relevance. The optimum probability mass of each Voronoi cell would be an exact fraction m_t of the total probability mass:

$$m_t = \frac{1}{\|D_t\|} \cdot \sum_{k \in \Omega} p_t(k). \quad (9)$$

Table 2 formalizes the procedures to select the set D_t to be displayed next. Given a target mass m , the procedure **ComputeDisplaySet** picks each image successively, each time selecting the one with the highest p_t which does not belong to the neighborhoods of mass m centered on the images already selected. In the function **ComputeCells**, the neighborhoods are grown in parallel by including images one by one, as ordered by their similarity distances, until the probability mass of each neighborhood reaches the target mass m .

The first display set D_0 is generated by running the algorithm with the initial probabilities of relevance, $p_0(k) = 0.5$ for all $k \in \Omega$. The algorithm still grows the Voronoi cells but chooses the images randomly between the equally probable candidates.

Figure 3 shows the intermediate steps of the Voronoi tessellation algorithm. One can see how the Voronoi cells are grown, and how the images to be displayed are selected. Intuitively, the cells including regions with higher probabilities are smaller than the cells including regions with lower probabilities. In this way, the system concentrates on regions with high probabilities while still sampling the entire collection.

4 Large-scale HEAT framework

The original approach requires a computational effort that is tightly related to the size of the collection. The probabilities of relevance are computed for all the images in the collection. Although the computational load of the probability model is very light in itself, it requires access to the similarity distances from all the images in the collection to each of the displayed images, and this implies either storage capacity of $\mathcal{O}(\|\Omega\|^2)$ complexity off-line or computational effort of $\mathcal{O}(\|\Omega\|)$ on-the-fly. Additionally, the Voronoi tessellation algorithm involves sorting operations of $\mathcal{O}(\|\Omega\| \cdot \log \|\Omega\|)$ complexity over the entire collection.

While maintaining all the core operations unchanged, our approach manages to compute the probabilities of relevance of only a small set of representative images. The probabilities of relevance of all the other images in the collection are approximated from these ones. This is achieved by organizing the image collection as a pre-computed hierarchical tree based on the image similarity distances and by updating during the retrieval process a partitioning of the image collection according to the estimated probabilities.

4.1 Tree and trace

The image collection Ω is organized in a hierarchical tree \mathcal{N} as sketched in the left side of Figure 5. Formally, each node $N \in \mathcal{N}$ has a set of children denoted as $C(N) \subset \mathcal{N}$, and is associated with a set of images $\Omega(N) \subset \Omega$. Each leaf node is associated with one single image. Thus if N is a leaf node, then $C(N) = \emptyset$ and $\|\Omega(N)\| = 1$. These sets of images are hierarchically

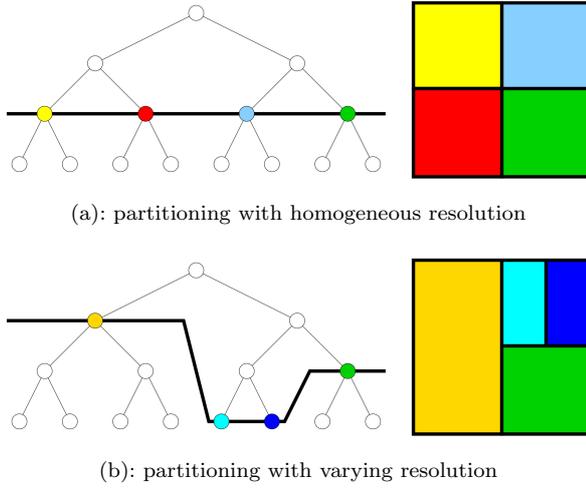


Fig. 5 Relation between the hierarchical tree and the trace adaptive partitioning. The graph depicted on the left stands for the tree \mathcal{N} , and the square on the right stands for the full image collection Ω . Intuitively, each node $N \in \mathcal{N}$ is associated with a subset of images $\Omega(N)$. The thick black lines running through the trees show two different traces \mathcal{T} . The colored rectangles show the resulting partitions of the collection, as each rectangle stands for the $\Omega(N)$ associated to the node N of same color. The trace in (a) stays at the same depth, resulting in a homogeneous partitioning. The trace in (b) goes shallower in one part of the collection and deeper in the other part, resulting in a partitioning with varying resolution.

disjunctive and naturally respect the properties:

$$\forall M, M' \in \mathcal{C}(N), M \neq M' \Rightarrow \Omega(M) \cap \Omega(M') = \emptyset, \quad (10)$$

$$\cup_{M \in \mathcal{C}(N)} \Omega(M) = \Omega(N). \quad (11)$$

Additionally, each node $N \in \mathcal{N}$ has a representative image k_N^* that is the closest image to the center of $\Omega(N)$ in the image feature space.

A trace $\mathcal{T} \subset \mathcal{N}$ is any set of nodes which is a partition of the image collection:

$$\forall A, B \in \mathcal{T}, A \neq B \Rightarrow \Omega(A) \cap \Omega(B) = \emptyset, \quad (12)$$

$$\cup_{A \in \mathcal{T}} \Omega(A) = \Omega. \quad (13)$$

These properties guarantee that any image in the collection is associated to one and only one node in any trace. Therefore, if $N \in \mathcal{T}$ is a node included in the trace, it can be used without ambiguity to represent all its associated images $\Omega(N)$ as illustrated in Figure 5.

4.2 Approximation of p_t

The computational effort is controlled by estimating the probabilities of relevance only for the representative images of the nodes that are part of the

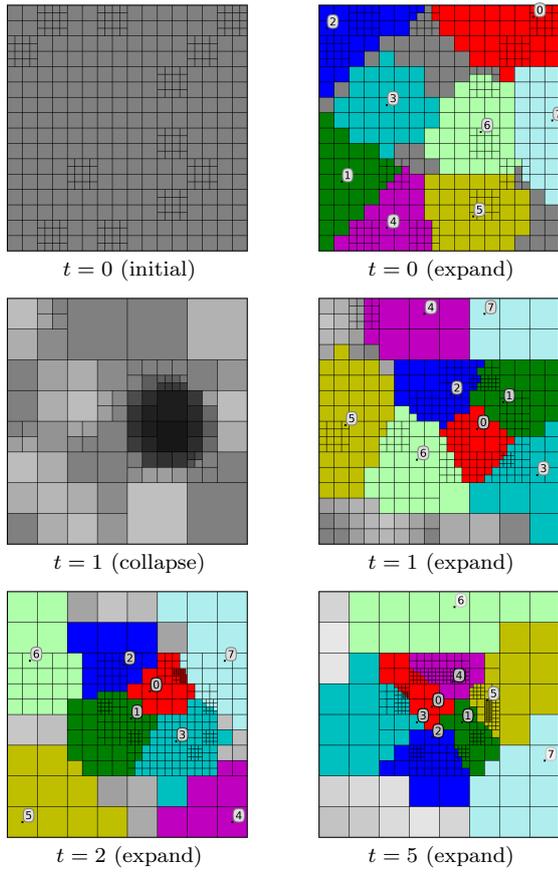


Fig. 6 Evolution of the trace for the synthetic collection, when searching for images with points close to the center. At iteration 0, the trace is initialized randomly. At each iteration, the trace is collapsed and expanded, the probabilities of relevance are updated, and then the display set to be shown next is selected. After 5 iterations, the trace concentrates mostly in the intended region.

current trace. From this bounded set of probabilities, we both infer a sound approximation of the Voronoi tessellation algorithm previously described in § 3.3 and optimize the resolution of the trace as presented next in § 4.3.

For any node $N \in \mathcal{T}$, the probabilities of relevance of all the individual images in $\Omega(N)$ are approximated by the probability of relevance of its representative image k_N^* .

At each iteration t , the conditional probabilities $p_t(k_N^*)$ are computed from scratch based on the full history of relevance feedback events B_{t-1} as shown in § 3.1. They are not approximated in any way.

Furthermore, the prerequisites of the Voronoi tessellation algorithm described in § 3.3 are reconsidered as follows. The probability mass of a node N

is approximated as:

$$q(N) = \sum_{k \in \Omega(N)} p_t(k) \approx p_t(k_N^*) \cdot \|\Omega(N)\|. \quad (14)$$

The probability mass of the entire collection is approximated as:

$$q^{all} = \sum_{k \in \Omega} p_t(k) \approx \sum_{N \in \mathcal{T}} q(N). \quad (15)$$

The optimum probability mass of the Voronoi cells is approximated as:

$$q^{opt} = \frac{1}{\|D_t\|} \cdot q^{all} \approx \frac{1}{\|D_t\|} \cdot \sum_{N \in \mathcal{T}} q(N). \quad (16)$$

When a node N is expanded, its probability mass $q(N)$ is substituted by the probability masses of its children, and this results in a finer approximation:

$$q(N) = \sum_{M \in C(N)} q(M) \approx \sum_{M \in C(N)} p_t(k_M^*) \cdot \|\Omega(M)\|. \quad (17)$$

When the nodes in $C(N)$ are collapsed, the sum of their probability masses is substituted by the probability mass of their parent, and this results in a coarser approximation:

$$\sum_{M \in C(N)} q(M) = q(N) \approx p_t(k_N^*) \cdot \|\Omega(N)\|. \quad (18)$$

Based on these approximations, the Voronoi tessellation algorithm is now performed at the granularity level of the trace instead of the individual images. Therefore, the centers of the Voronoi tessellation are selected among the nodes in the current trace, and the displayed images are their corresponding representative images.

4.3 Trace refinement

The aim of the trace refinement is to optimize the approximation of the probabilities of relevance of the individual images while keeping the trace size bounded. Intuitively, this is achieved when the variances of the probabilities within each node in the trace are small, or in other words when the probability of each image in the collection is approximated as well as possible by the probability of its corresponding representative image. The trace refinement consists of a collapsing operation followed immediately by an expansion operation.

Starting from the current trace, the collapsing operation book-keeps the sets of children that are completely included in the trace, and thus may be replaced by their parents. Recursively, one at a time, the set of children that minimizes the mean-variance cost function

$$\operatorname{argmin}_{\forall N, C(N) \subset \mathcal{T}} \mu(N) \cdot (\sigma^2(N) + \epsilon \cdot \|\Omega(N)\|), \quad (19)$$

where $\mu(N)$ is the mean and $\sigma(N)$ is the standard deviation of the image probability distribution in the node N , is collapsed into its corresponding parent. The probability of relevance of the representative image $p_t(k_N^*)$ is computed from scratch as mentioned in § 4.2, and is then used for computing the subsequent mean-variance values. The recursive routine for collapsing nodes exits when the size of the trace reaches the minimum bound.

The probability mean and variance of each node are estimated based on its children:

$$\mu(N) = \frac{\sum_{M \in \mathcal{C}(N)} p_t(k_M^*) \cdot \|\Omega(M)\|}{\sum_{M \in \mathcal{C}(N)} \|\Omega(M)\|}, \quad (20)$$

$$\sigma^2(N) = \frac{\sum_{M \in \mathcal{C}(N)} p_t^2(k_M^*) \cdot \|\Omega(M)\|}{\sum_{M \in \mathcal{C}(N)} \|\Omega(M)\|} - \mu^2(N). \quad (21)$$

In Equation (19), ϵ introduces an infinitesimal preference toward collapsing the nodes with smaller cardinality when nodes with different cardinality have comparable mean-variance values. Thus, ϵ is not a sensitive parameter and was set to 10^{-6} , a value related to the size of the collection.

As soon as the collapsing operation exits, the expansion operation replaces all the nodes in the trace with their children and computes the probabilities of relevance of their representative images. This expansion operation could be seen as a sampling of the parent nodes that will be used in the subsequent trace refinement, at the next iteration, in order to identify the new nodes that should be further expanded or can be safely collapsed.

4.4 Algorithm integration

The skeleton of our proposed approach is as follows. At iteration $t + 1$, the algorithms proceed with the trace from the previous iteration \mathcal{T}_t :

1. Update the probabilities of relevance $p_{t+1}(k_N^*)$ for $\forall N \in \mathcal{T}_t$ based on the previously computed $p_t(k_N^*)$ and according to the newly received relevance feedback event $\{D_t, x_t^*\}$.
2. Perform the trace refinement. The trace \mathcal{T}_t is altered via the collapsing and expanding operations resulting in the new trace \mathcal{T}_{t+1} .
3. Update the probabilities of relevance $p_{t+1}(k_N^*)$ for $\forall N \in \mathcal{T}_{t+1}$ according to the full history of relevance feedback events $B_t = \cap_{i=0}^t \{D_i, x_i^*\}$.
4. Select the set of images D_{t+1} by performing the Voronoi tessellation algorithm on the current trace \mathcal{T}_{t+1} .
5. Display D_{t+1} . Wait for the relevance feedback event $\{D_{t+1}, x_{t+1}^*\}$ to occur, and then proceed with the next iteration.

For an intuitive illustration of the system behavior, we set up the HEAT system for the synthetic collection, and we once more take the case of searching for images with points close to the center. Figure 6 shows how the trace evolves at each iteration and how the image collection is sampled at different resolutions in different regions.

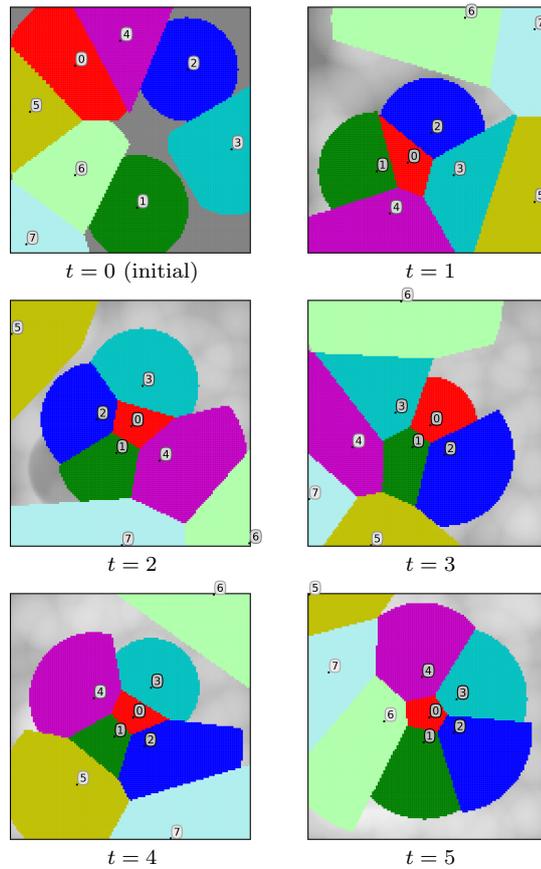


Fig. 7 Evolution of the display set for the original framework with the synthetic collection, when searching for images with points close to the center. After 5 iterations, the displayed images concentrate slightly on the intended region. Again, the selected images are marked in black and their corresponding Voronoi cells are indicated by colors.

5 Exploration/exploitation trade-off

As argued by Ferecatu and Geman [13, 14], the original approach is well suited for image category search and that is, in other words, the first retrieval regime of exploring the image collection. They explicitly suggest that other retrieval techniques should be employed to retrieve specific images among these identified categories and that is, in other words, the second retrieval regime of exploiting the image collection.

A useful insight is given by analyzing the evolution of the retrieval system for the synthetic collection, when searching for images near the center. Figure 7 shows the evolution of the displayed images, and Figure 8 shows the distribution of the probabilities of relevance.

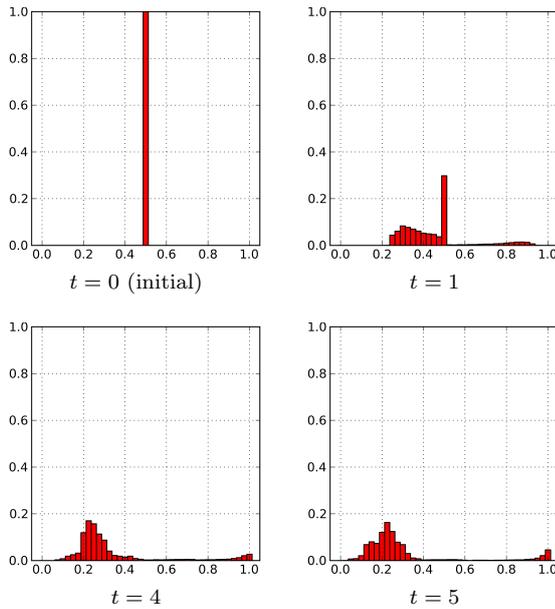


Fig. 8 Evolution of the distribution of probabilities of relevance for the searching session illustrated in Figure 7. The plots have the probability bins on axis X, and the percentage of images in the collection on axis Y. Initially, all images have the same probability, $p_0(k) = 0.5 \forall k \in \Omega$. After the very first iterations, the distribution evolves relatively slow.

As shown in Figure 8, the distribution of the probabilities evolves quite rapidly in the first iterations. These early iterations correspond to the first retrieval regime when the system is in the process of understanding broadly the categories of interest to the user. Later, after the system has achieved a good understanding of the user interest, the distribution of the probabilities evolves quite slowly from one iteration to the next. These later iterations correspond to the second retrieval regime when the system is meant to refine the search and to converge to specific images.

As shown in Figure 7, the sets of displayed images include an image that is closer and closer, with each iteration, to the user interest. After 3 iterations, the system succeeds to display an image that is clearly in the intended region. Still after 5 iterations, the displayed images concentrate only slightly on the intended region.

The system succeeds efficiently to display an image in the intended region, but has a hard time to display more and more images in the intended region. The “sampling” algorithm insists on covering the entire collection even after the distribution of probabilities becomes rather stable. One can say that the original system has a big inertia to maintain an exploration regime and goes very slowly into an exploitation regime.

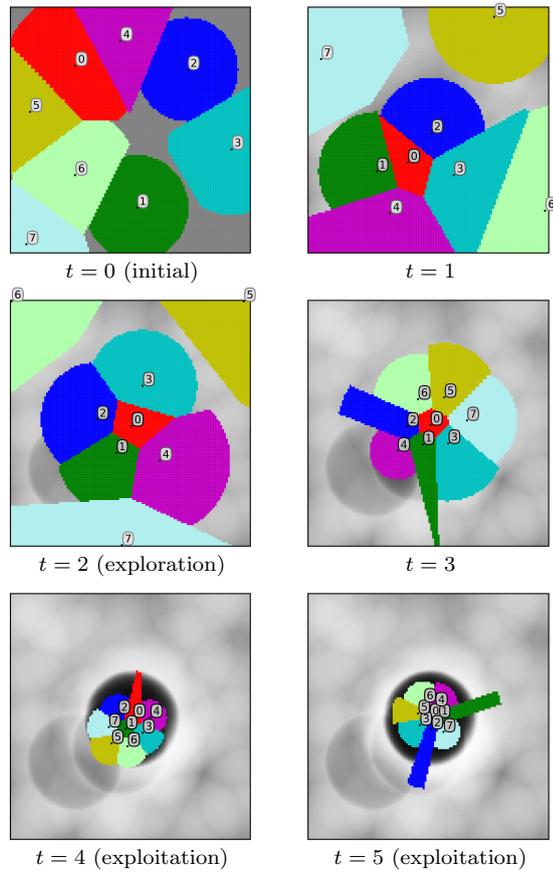


Fig. 9 Evolution of the display set for the mass-zoom system with the synthetic collection, when searching for images with points close to the center. After 5 iterations, the displayed images concentrate mostly on the intended region. The displayed images provide the freedom to escape the exploitation if necessary. The system continuously estimates the exploration/exploitation trade-off that suits the user.

5.1 Mass-zoom extension

Intuitively, the system should be aware of the degree of alignment of the distribution of probabilities with the user intent. When the distribution of probabilities is in line with the user intent, the system should concentrate the “sampling” in regions with high probability.

First, we present an adaptive strategy to handle the trade-off between exploration and exploitation, by modulating the concentration of the display set on promising images. Second, we present a heuristic that dynamically infers, at each iteration, from the user actions a consistency score that achieve a seamless trade-off that suits the user intent.

Our mass-zoom algorithm handles the trade-off between exploration and exploitation by modulating how much the display set should be concentrated on the images assessed as the most relevant. This is achieved by estimating at every iteration the target mass m_t for the displayed image neighborhoods. While this value was a constant fraction of the total mass in the baseline in Equation (9) from § 3.3, we propose to link it to an estimate of the confidence of our current estimate of the image relevance. Making the value of this target mass smaller makes the neighborhoods around the images of the display set smaller, which leads to a more compact display set, concentrated in the regions of high probability.

Our approach increases the concentration of the display set if the choice of the user is consistent with our current estimate, and decreases it otherwise. We propose the following update scheme:

$$m_t^{zoom} = z_t \cdot m_t, \quad (22)$$

where $z_t \in \left(\frac{1}{m_t}, 1\right]$ accounts for the consistency between our estimates of the p_t and the user choice.

5.2 Heuristics based on a consistency score

Immediately after the relevance feedback event $\{D_t, x_t^*\}$, right at the beginning of the next iteration $t + 1$, the consistency score aims to estimate the alignment of the system and the user intent, which is defined in Equation (23) as an increasing function of the probability, under our model, that an image picked at random in the display set would have a lower probability than the image chosen by the user

$$c_{t+1} \nearrow \widehat{P}_{x \sim \mathcal{U}(D_t)} [p_t(x_t^*) \geq p_t(x)]. \quad (23)$$

In the first iteration, the user intent is totally unknown and the consistency score c_0 is initialized to 1. Subsequently, the consistency score is estimated based on the probability of relevance of the chosen image $p_t(x_t^*)$ versus the probabilities of relevance of the other displayed images, namely $p_t(x)$, for all $x \in D_t$.

The consistency score is estimated based on the cumulative distribution function for the Gaussian distribution. The proposed heuristic gives a consistency score in the interval $[0.5, 2.0]$:

$$c_{t+1} = 0.5 + 1.5 \cdot \left(\frac{1}{2} + \operatorname{erf} \left(\frac{p_t(x_t^*) - \hat{\mu}}{\hat{\sigma} \cdot \sqrt{2}} \right) \right), \quad (24)$$

where $\hat{\mu}$ is the average of the probabilities in $\|D_t\|$

$$\hat{\mu} = \frac{1}{\|D_t\|} \cdot \sum_{x \in D_t} p(x), \quad (25)$$

and $\hat{\sigma}$ is the standard deviation of the probabilities in $\|D_t\|$

$$\hat{\sigma}^2 = \frac{1}{\|D_t\|} \cdot \sum_{x \in D_t} (p(x) - \hat{\mu})^2. \quad (26)$$

This is motivated by the intuition that if the $p_t(x_t^*)$ is already among the highest probabilities it means that the system has a distribution of the probabilities that is in line with the user intent, and thus the system is consistent with the user intent. If $p_t(x_t^*)$ is relatively low, the system is less consistent with it.

The zoom value that impacts the exploration/exploitation trade-off of the selection of the displayed images is derived from the consistency scores as follows:

$$z_t = \prod_{i=0}^t \frac{1}{c_i}. \quad (27)$$

5.3 Capabilities of the mass-zoom system

For an intuitive illustration, we set up the mass-zoom system for the synthetic collection described in §3, and we once more take the case of searching for images with points close to the center. In Figure 9, we show the evolution of the displayed images for intermediate iterations during one such searching session.

After efficiently identifying the intended region, the mass-zoom system is able to display more and more images in the intended region. The “sampling” algorithm concentrates in the intended region after the distribution of probabilities becomes rather stable. Although the “sampling” algorithm does not cover the entire collection anymore, the system continuously estimates the exploration/exploitation trade-off that suits the user.

Note that while the synthetic collection is very handy for intuitive illustrations, it should not be mistaken for a real image collection, which typically involves high-dimensional image indexing feature spaces. Besides the misalignment between the image feature space and the user subjective perception of image similarities, the distribution of the image similarity distances impacts the Voronoi tessellation algorithm as well as the distribution of the probabilities of relevance. We argue that the exploration/exploitation trade-off has even higher impact in the case of real image collections.

6 Log-based similarity learning

The similarity metric used by the system as explained in 3.2 must be aligned reasonably well with the user similarity judgments, and the better the alignment, the more reliable the relevance feedback. In the original approach, the parameters $\delta^{+/-}$ and $\varphi^{+/-}$ in Figure 4 are learned via a statistical technique

that requires an image labeling task. During the labeling session, the user input is collected in the same manner as a searching session, with the key difference that the visual target S is communicated to the user. The user is supposed to indicate on the display sets D the image that is closest to the target in her opinion as during a searching session. In this way, the labeling session collects data triplets of the form (S_i, D_i, x_i^*) that can be used to formulate a maximum likelihood technique.

We propose to derive a more optimal similarity measure between the images by exploiting the user feedback histories that are acquired naturally during the searching sessions (i.e. user logs information). During the on-the-fly sessions, the user feedback histories are stored as user logs in a database. Then, as the user logs are accumulated in the database, they can be used off-line to improve the similarity metric. This approach has the major advantage that it does not require any extra image labeling that would imply additional effort and resources. As the retrieval system is used, the user logs will gradually cover the entire collection, and the log-based similarity metric will systematically improve.

We observe that the user feedback can be seen as a weak partially reliable image labeling, and we formulate a technique that tunes off-line the image similarity metric in order to model explicitly the user similarity judgments. Internally, we define a weighted Euclidean metric over the image feature space which we then optimize in order to maximize the probabilities of relevance of the images chosen by the users. Then, we use this optimized log-based metric instead of the original L^2 norm on-the-fly to run the retrieval process.

Our retrieval system stores all the searching sessions that are performed by the users during the evaluation campaigns in so-called user logs. Each user log corresponds to a searching session, and contains all the data necessary to recall the context of the evaluation campaign and to reproduce that corresponding session. Besides the information about the user, the system configuration and the target task, each user log contains the history of relevance feedback events $\{D_t, x_t^*\}$, $t = 1, 2, \dots, T$ where $T \leq 20$ and the user label (i.e. successfully terminated or failed).

Our challenge is that the searching session are labeled only globally as successfully terminated or failed. This tells us if the last display set contains target images or not, but unfortunately it does not explicitly tell which images.

6.1 Weighted Euclidean distance

The original framework employs a similarity metric that is the Euclidean metric over visual-based feature vectors based on SIFT (Scale Invariant Feature Transform) [21]. Our approach re-defines the similarity metric as a weighted Euclidean distance over the image feature space as in Equation (28). Thus, we introduce a weighting vector α for which we will elaborate an optimization

scheme based on the user logs.

$$d_{\alpha}(k, h) = \sqrt{\sum_{f=1}^F \alpha_f \cdot (k_f - h_f)^2}, \quad (28)$$

where F is the dimensionality of the image feature space.

6.2 Log-based weights learning

Our alternative is to adapt the weighting vector in the sense of making the probabilistic model able to better predict the images indicated by the user. We consider that, in the searching sessions that were successfully terminated, all the history of relevance feedback events was for good and helped the user to get to the final display set that satisfied her. With this assumption, all the history of relevance feedback events are regarded as equally important, and it makes sense to adapt the weighting vector in order to maximize the probabilities of all the images chosen in all relevance feedback events.

With these considerations, we define the cost function as the total sum-log of the probabilities of relevance of the chosen images at the time of their display:

$$C = \sum_{u=1}^U \sum_{t=0}^T \log p_t(x_t^*), \quad (29)$$

where U stands for all the user logs (i.e. searching session histories) and T stands for the number of iterations of each searching session.

Next, we should choose an optimization algorithm in order to learn the optimal weighting parameter α that maximizes the cost function in Equation (29).

$$\alpha_{optim} = \arg \max_{\alpha} C. \quad (30)$$

We propose to optimize the weighting vector based on the full-batch gradient descent method combined with a simple line search. If the amount of user logs becomes large, the cost function could be optimized using approximations as for example the stochastic gradient descent method.

Initially, at iteration $n = 0$ the weighting vector α^0 is set to $\mathbf{1}$. In the subsequent iterations, the gradient descent algorithm is performed according to Equation (31), which can be expanded for each weighting coefficient α_i as in Equation (32).

$$\alpha^{n+1} = \alpha^n + \gamma_n \cdot \nabla C(\alpha^n), \quad n > 0, \quad (31)$$

$$\begin{aligned} \alpha_i^{n+1} &= \alpha_i^n + \gamma_n \cdot \nabla C(\alpha_i^n) \\ &= \alpha_i^n + \gamma_n \cdot \left. \frac{\partial C}{\partial \alpha_i} \right|_{\alpha_i = \alpha_i^n}. \end{aligned} \quad (32)$$

The partial derivatives can be elaborated starting from the top derivative in Equation (33).

$$\frac{\partial C}{\partial \alpha_i} = \sum_{u=1}^U \sum_{t=0}^T \frac{1}{p_t(x_t^*)} \cdot \frac{\partial p_t(x_t^*)}{\partial \alpha_i}. \quad (33)$$

We observe that the similarity metric learning has to deal with two inter-dependent parts: the $\phi^{+/-}$ parameters that align the user perception to the similarity feature vectors and the α coefficients that align the similarity feature vectors to the user perception. Our proposed solution is optimizing the α coefficients, and not the $\phi^{+/-}$ parameters. However, we analyze the influence of the $\phi^{+/-}$ parameters in §7.1. We observe that the optimal $\phi^{+/-}$ parameters do not differ significantly from their initial values. Our interpretation is that the α 's optimization compensates for the $\phi^{+/-}$ parameters.

Our approach has two major advantages. On the one hand, it exploits the user feedback that is acquired naturally during the searching sessions, and does not require any log acquisition campaigns. Even if the similarity models would be changed, the user logs can still be used in the same manner. On the other hand, it is generic and can leverage very large amounts of user logs. As the retrieval system is used, the user logs will gradually cover the entire collection, and the log-based similarity metric will systematically improve.

7 Experimental results

Our contributions touch different components of the retrieval system, and their integration into a comprehensive system is straight-forward. For illustration of the retrieval behavior, we once more take the case of searching for images with points close to the center. Figure 10 shows how the comprehensive system evolves at each iteration and how the image collection is sampled at different resolutions in different regions in combination with the mass-zoom extension.

The retrieval system is developed as a web-application¹. Besides the advantage of permanent availability for evaluations, this implementation encourages the adherence to a realistic system architecture. The application software is distributed under the AGPL Version 3 free software license that enforces the sharing of future extensions.

Since we are not aware of any other iterative relevance feedback mechanism that can be applied to a collection of 1M images, our quantitative analysis focuses on comparing systematically our contributions to the baseline non-scalable version of the approach. The experiments were organized with two collections obtained from the ImageNet database [10]. We obtained a *large collection* of about 1,054,000 images by considering all the images provided with valid url. Then, we obtained the *small collection* of 60,000 images by sampling uniformly the large collection (i.e 5% of the large collection). The indexing features are simply the histogram-like vectors (i.e. referred to also as

¹ The web-application software is available at <http://www.idiap.ch/software/imr/>

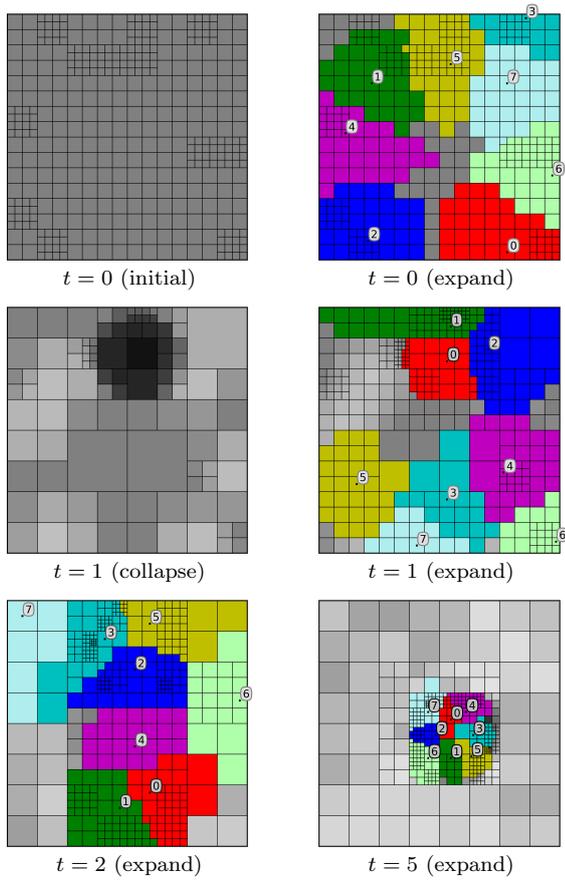


Fig. 10 Evolution of the comprehensive system for the synthetic collection, when searching for images with points close to the center. At iteration 0, the trace is initialized randomly. At each iteration, the zoom factor is estimated, the trace is collapsed and expanded, the probabilities of relevance are updated, and then the images to be shown next are selected. After 5 iterations, the trace concentrates mostly on the intended region.

bags of visual words) of dimension 1000, as they are provided by ImageNet. For further reference, this means about 5GB of data downloading and storage.

Our experiments are presented in two parts. First, we analyze the robustness of the optimization scheme and we obtain the optimal weighed Euclidean distance in §7.1. Second, we organize user-based evaluations and analyze the system behavior in §7.3.

7.1 Log-based weights analysis

In our optimization scheme, we make use of the user logs from some of our previous experiments in [27]. Namely, we considered all successfully terminated

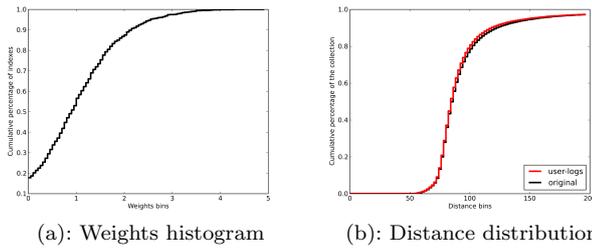


Fig. 11 Weights analysis. (a): Histogram of the feature weights in the optimal weighting vector α , which was obtained after running the gradient descent algorithm. About 15% of the image features are zeroed, and the maximum weight is no larger than 5. (b): Cumulative distribution of the similarity distances in the collection, for both the original distances and the optimized distances. We can see that the distributions remain rather alike, which means that the weighting vector is normalized properly by the optimization scheme.

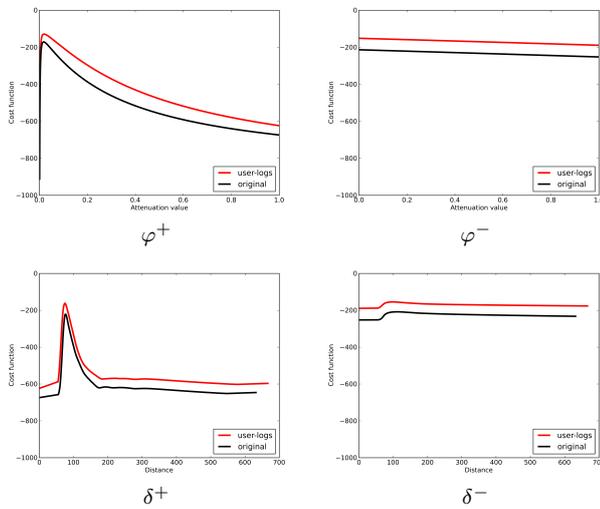


Fig. 12 Influence of the calibration parameters on the cost function, for both the original and the optimized distances. To identify the parameters, one should recall the calibration functions in Figure 4. Here each plot corresponds to a parameter and shows how the cost function depends on that parameter while keeping the others un-changed. We can see that the cost functions corresponding to the optimized distances give similar peaks as the ones corresponding to the original distances, which means that the weighting vector is normalized properly by the optimization scheme.

searching sessions that were performed for the L^2 type of distances. There are in total 142 searching sessions, that results in a cumulative set of 1050 relevance feedback events. With these data, we performed the full-batch gradient descent algorithm with a simple line search. The algorithm converged after 8,000 iterations, but we let it run up to 10,000 iterations.

The calibration parameters are set as in our previous works [26, 27]. $\delta^{+/-}$ are adjusted to saturate only after including on average 10% of the images in the collection, see Figure 11(b). $\varphi^{+/-}$ are given the same optimum values as derived in [13], namely 0.06 and 0.29.

Figure 11(a) shows the histogram of the weights in the final optimal weighting vector α , which were obtained after performing the gradient descent algorithm. Here we recall that the original distances are equivalent to the uniform weighting vector $\mathbf{1}$. We can see that the optimal weighting vector remains bounded although no upper constraints have been enforced.

Figure 11(b) shows the cumulative distributions of the image similarity distances in the collection, for both the original and the optimized distances. We can see that the distributions remain rather alike, which means that the weighting vector is normalized properly by the optimization scheme. About 10% of the distances in the collection are smaller than 75, and about 10% of them are larger than 125. The majority of the distances are in the range 75-125, and this is the “spherical” effect of the Euclidean distance on the high-dimensional feature vectors.

Figure 12 shows the influence of the calibration parameters on the cost function, for both the original and the optimized distances. Each plot corresponds to a calibration parameter and shows how the cost function depends on that parameter while keeping the others un-changed. We can see that cost functions corresponding to the optimized distances give very much the same peaks as the ones corresponding to the original distances, which means that the weighting vector is normalized properly by the optimization scheme, without imposing any ad-hoc hard constraints. The only parameter that may differ from our initial settings is φ^- , as the cost function is maximized when φ^- collapses to 0. Here we observe that in fact $\varphi^- = 0$ is not a critical setting in our setup since there are not many small distances in between the images in the collection, as we explain in Figure 11(b).

7.2 Evaluation scenario

Evaluations have been conducted with 20 users using the interface shown in Figure 13. In order to ensure a reliable diversity, there were 6 semantic categories described in words and accompanied by the corresponding images in Figure 14. In order to ensure comparable difficulty, these categories were chosen to be relevant for about 1% of our collections based on the evidence given by the ImageNet categories ground-truth.

- portraits/close-ups of dogs, wolves
- electronic devices as laptop, mobile phone
- big boats as ferryboats, cargoes
- baskets/plates with fruits, vegetables
- furniture items as tables, chairs
- entrances/windows of shops, shopping centers

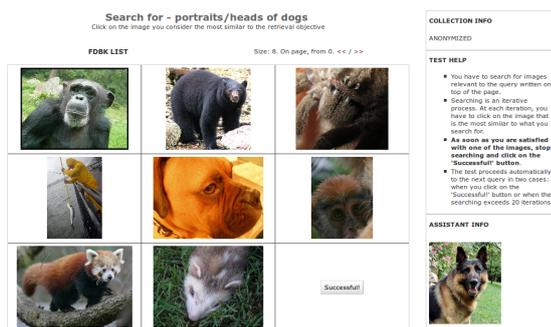


Fig. 13 Web interface of the system used for user tests. The searching sessions were presented in a random fashion. The users were only told to end the sessions when they were satisfied by one of the displayed images.

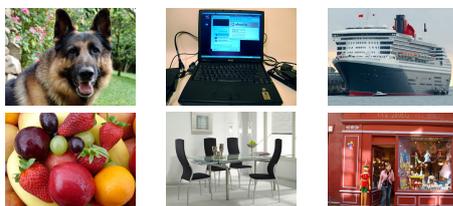


Fig. 14 The users were asked to search for semantic categories described in words and accompanied by image examples as shown here. There were 6 semantic categories, each being relevant for about 1% of our collections.

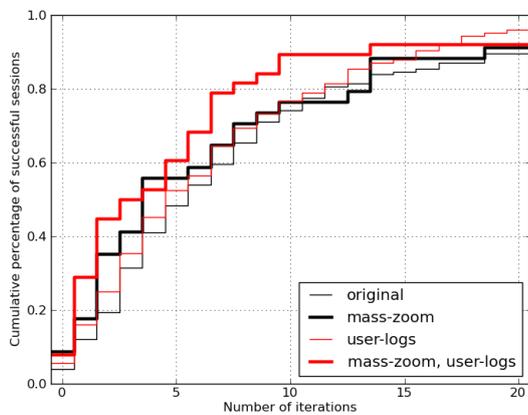


Fig. 15 Retrieval performance of the mass-zoom system in combination with the log-based similarity metric for the 60K image collection. Each of our contributions taken individually improves the retrieval performance of the original system. Furthermore, they complement each other and their combination significantly improves the overall performance.

In order to ensure a minimal variance in the scenario, each user was assigned to perform one searching session for each combination of the system configurations and the semantic targets included in the evaluation. Thus, there were $6 \times 20 = 120$ searching sessions for each system configuration.

In order to avoid any bias, the searching sessions were presented in a random fashion. The semantic categories and the system configurations were randomized all together in one single user test. The evaluation interface is shown in Figure 13. The users were not aware of the system configurations. The users were only told to end the searching sessions when they were satisfied by one of the displayed images.

7.3 Results analysis

We aim to evaluate the overall performance of the comprehensive system that integrates all our contributions. First, we systematically evaluate two partial combinations of our contributions, in order to get evidence that each contribution complements each other, and that their combination performs consistently. Then, we analyze the total integration.

7.3.1 Mass-zoom system with log-based similarity metric

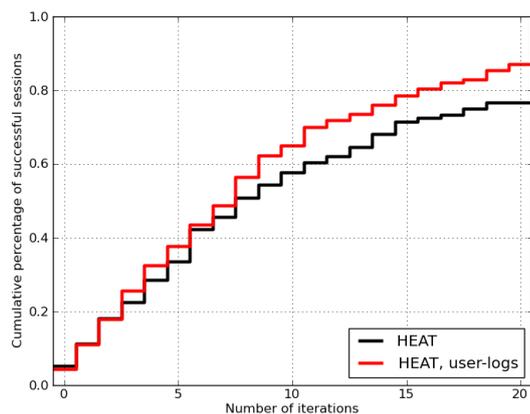
The retrieval performance of the mass-zoom system in combination with the log-based similarity metric is shown in Figure 15. Each contribution taken individually improves the retrieval performance of the original system. Furthermore, the two contributions complement each other and their combination significantly improves the overall performance. The optimized system provides 80% rate of success in less than 8 iterations, while the baseline system reaches the same rate only after 13 iterations, which is 5 iterations more. Experiments were conducted only with the small collection, since the baseline system cannot cope with the large collection.

7.3.2 HEAT framework with log-based similarity metric

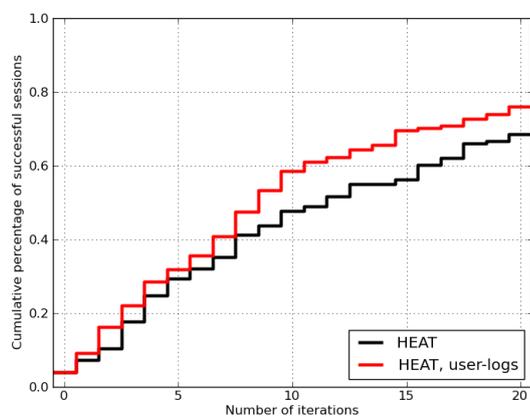
Figures 16(a-b) show the performance of the HEAT system in combination with the log-based similarity metric. For both collections, we can see that the integration of the HEAT system with the optimized metric is beneficial. In fact, the HEAT system benefits even more than the original system since the log-based similarity metric improves not only the on-the-fly models but also the quality of the pre-computed hierarchical organization of the collection. The optimized system performs consistently better, and saturates about 10% higher.

7.3.3 HEAT framework with mass-zoom extension

The retrieval performance of the HEAT system in combination with the mass-zoom extension is shown in Figure 17 among other system combinations. We



(a): 60K image collection



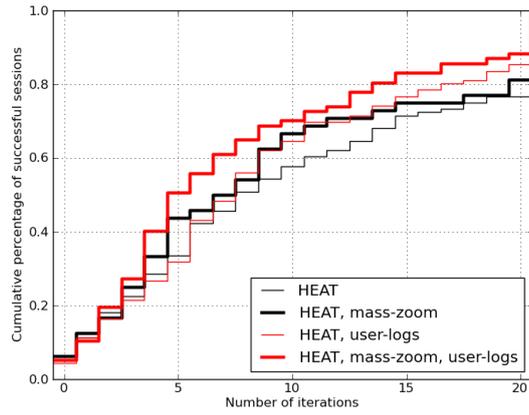
(b): 1M image collection

Fig. 16 Retrieval performance of the HEAT framework in combination with the log-based similarity metric for both small and large collections. The integration of the HEAT system with the log-based similarity metric is beneficial. The HEAT system benefits even more than the original system since the log-based similarity metric improves not only the on-the-fly models but also the quality of the pre-computed hierarchical organization of the collection.

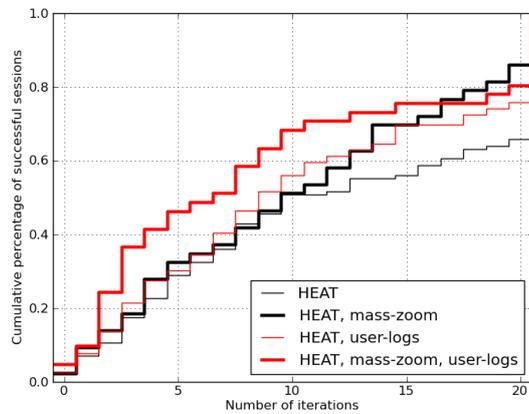
can see that their integration improves their individual performances for both small and large collections, which means that they complement each other.

7.3.4 Total integration

The retrieval performance of the comprehensive system that integrates all our contributions is shown in Figure 17. Both individual integrations of the mass-zoom extension and the log-based similarity metric improve the retrieval



(a): 60K image collection



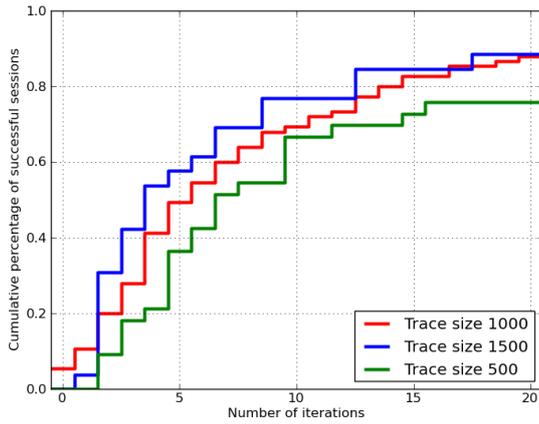
(b): 1M image collection

Fig. 17 Retrieval performance of the comprehensive system that integrates all our contributions. Taken individually, the mass-zoom extension and the log-based similarity metric improve the retrieval performance of the HEAT system. The contributions complement each other and their combination significantly improves the overall retrieval performance. Furthermore, the behavior of the system is stable and consistent for both small and large collections.

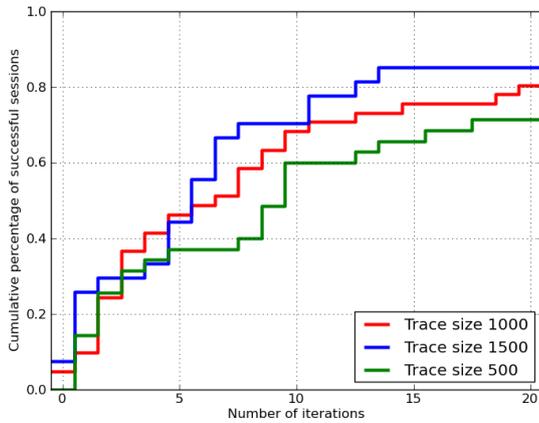
performance of the HEAT system. Furthermore, they complement each other and their combination significantly improves the overall retrieval performance.

The retrieval performance of the overall system is further analyzed in Figure 18. Here the performance is evaluated for three different trace sizes. We can see how the retrieval performance depends on the trace size. The bigger the trace, the better the performance, but the difference between 1,000 vs. 1,500 is smaller than the difference between 500 vs. 1,000.

Overall, about 50% of the sessions are successfully terminated in less than 5 iterations, and 80% in less than 15 iterations. The system performance remains



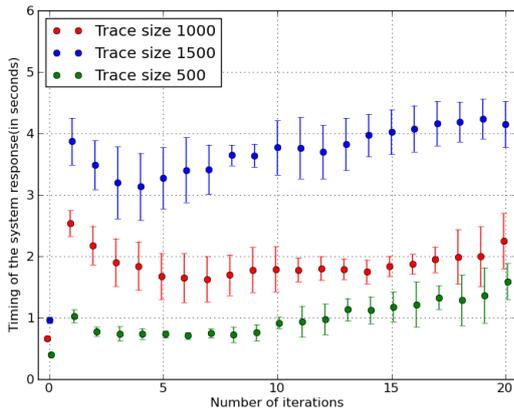
(a): 60K image collection



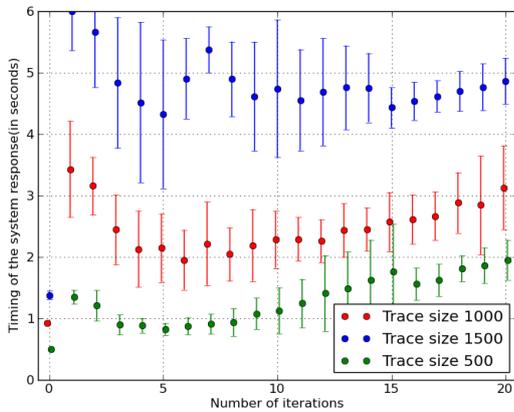
(b): 1M image collection

Fig. 18 Retrieval performance of the comprehensive system for three different trace sizes. The average performances for the small collection are shown in (a): We can see how the performance depends on the trace size. The bigger the trace, the better the performance, but the difference between 1,000 vs. 1,500 is smaller than the difference between 500 vs. 1,000. The average performances for the large collection are shown in (b): The performances remain consistent.

very reasonable when thinking of the theoretical bounds. In the ideal case, if the collection would be arranged as a tree with 8 branches at each node, the *perfectly-structured* search will need $\log_8 \|\Omega\| \approx 7$ iterations at maximum. In the worst case, if the collection would be totally unstructured, the *uniformly-random* search will need $\|\Omega\|/(\|D\| \cdot (L+1)) \approx 13$ iterations in average. Here, $\|\Omega\| \approx 1,000,000$, $\|D\| = 8$, $L \approx 1\%$ of $\|\Omega\|$ are the sizes of the image collection, the display set, and the semantic category.



(a): 60K image collection



(b): 1M image collection

Fig. 19 Timing of the comprehensive system responses (in seconds) as the users experienced them during the evaluations. Here we compare the computational cost (mean and standard deviation) for three different trace sizes. The timings for the small collection are shown in (a): The computational effort of the comprehensive system depends on the trace size, and has roughly $\mathcal{O}(\|\mathcal{T}\| \cdot \log \|\mathcal{T}\|)$ complexity. In the first iterations, the computation is higher due to the intensive collapse/expansion operations. In the later iterations, the trace is relatively more stable, and the computation increases slowly with the number of iterations due to the calculation from scratch of the probabilities of relevance. The timings for the large collection are shown in (b): The timings remain comparable with the ones for the small collection. The computational effort is decoupled from the collection size and it depends mainly on the trace size.

Figure 19 gives an insight on the computational effort of the comprehensive system, by showing the system response timing in seconds as the users experienced it.

8 Conclusion

We have presented a query-free retrieval approach that relies on iterative relevance feedback. Our contributions extend and reshape the retrieval mechanism in three complementary aspects, namely the large-scale HEAT framework, the mass-zoom extension and the log-based image similarity metric.

We systematically evaluated different combinations of our contributions in the same manner as each individual contribution, and we got evidence that the contributions complement each other. We also evaluated the comprehensive retrieval system, and showed that the overall integration of our contributions is consistently beneficial.

We foresee that our contributions, along with our free software web-application, will motivate further investigations and facilitate further experiments. We hope that our research brings the iterative relevance feedback mechanism one step closer to commercial applications.

Acknowledgment

Nicolae Suditu was supported by the Hasler Foundation through the EMMA project. François Fleuret was supported in part by the European Community's Seventh Framework Programme FP7 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement No 247022 - MASH. The authors would like to express their thanks to Prof.Dr. Donald Geman and Dr. Marin Ferecatu for their constructive feedback and fruitful discussions that served as inspiration.

References

1. Abello, J., Kobourov, S.G., Yusuf, R.: Visualizing large graphs with compound-fisheye views and treemaps. In: Graph Drawing, *Lecture Notes in Computer Science*, vol. 3383, pp. 431–441. Springer (2005)
2. Bederson, B.B.: PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps. In: Proceedings of the 14th ACM symposium on User interface software and technology, pp. 71–80 (2001)
3. Buchsbaum, A.L., Westbrook, J.R.: Maintaining hierarchical graph views. In: Proceedings of the 11th ACM-SIAM symposium on Discrete algorithms, pp. 566–575 (2000)
4. Campbell, I., van Rijsbergen, K.: The ostensive model of developing information needs. In: Proceedings of the International Conference on Conceptions of Library and Information Science: Integration in Perspective (CoLIS), pp. 251–268 (1996)
5. Carson, C., Thomas, M., Belongie, S., Hellerstein, J.M., Malik, J.: BlobWorld: A system for region-based image indexing and retrieval. In: Proceedings of the 3th International Conference on Visual Information Systems, vol. 1614, pp. 509–517 (1999)

6. Chang, E., Cheng, K.T., Lai, W.C., Wu, C.T., Chang, C., Wu, Y.L.: PBIR: Perception-based image retrieval – A system that can quickly capture subjective image query concepts. In: Proceedings of the 9th ACM International Conference on Multimedia, pp. 611–614 (2001)
7. Chechik, G., Sharma, V., Shalit, U., Bengio, S.: Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research* **11**, 1109–1135 (2010)
8. Cox, I.J., Miller, M.L., Minka, T.P., Papathomas, T.V., Yianilos, P.N.: The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing* **9**(1), 20–37 (2000)
9. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys* **40**(2), 1–60 (2008)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255 (2009)
11. Emre Celebi, M., Alp Aslandogan, Y.: Human perception-driven, similarity-based access to image databases. *Proceedings of the Artificial Intelligence Research Society Conference* pp. 245–251 (2005)
12. Fang, Y., Geman, D.: Experiments in mental face retrieval. In: *Proceedings of the 5th International Conference on Audio and Video-based Biometric Person Authentication*, pp. 637–646 (2005)
13. Ferecatu, M., Geman, D.: Interactive search for image categories by mental matching. In: *Proceedings of the 11th IEEE International Conference on Computer Vision*, pp. 1–8 (2007)
14. Ferecatu, M., Geman, D.: A statistical framework for image category search from a mental picture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(6), 1087–1101 (2009)
15. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. *Computer* **28**(9), 23–32 (1995)
16. Han, J., Ngan, K., Li, M.L., Zhang, H.J.: A memory learning framework for effective image retrieval. *IEEE Transactions on Image Processing* **14**, 511–524 (2005)
17. Heesch, D.: A survey of browsing models for content based image retrieval. *Journal of Multimedia Tools and Applications* **40**(2), 261–284 (2008)
18. Hoi, C.H., Lyu, M.R., Jin, R.: A unified log-based relevance feedback scheme for image retrieval. *IEEE Transactions on Knowledge and Data Engineering* **18**, 509–524 (2006)
19. Ishikawa, Y., Subramanya, R., Faloutsos, C.: MindReader: Querying databases through multiple examples. In: *Proceedings of 24rd International Conference on Very Large Data Bases*, pp. 218–227 (1998)
20. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(6), 985–1002 (2008)

21. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**(2), 91–110 (2004)
22. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Video Technology* **8**(5), 644–655 (1998)
23. Shneiderman, B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics* **11**(1), 92–99 (1992)
24. Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(12), 1349–1380 (2000)
25. Smith, J.R., Chang, S.F.: VisualSEEk: a fully automated content-based image query system. In: *Proceedings of the 4th ACM international conference on Multimedia*, pp. 87–98 (1996)
26. Suditu, N., Fleuret, F.: HEAT: Iterative relevance feedback with one million images. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2118–2125 (2011)
27. Suditu, N., Fleuret, F.: Iterative relevance feedback with adaptive exploration/exploitation trade-off. In: *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pp. 1323–1331 (2012)
28. Urban, J., Jose, J., Van Rijsbergen, C.J.: An adaptive technique for content-based image retrieval. *Multimedia Tools and Applications Processing (MTAP)* **31**, 1–28 (2006)
29. Weston, J., Bengio, S., Usunier, N.: Large scale image annotation: learning to rank with joint word-image embeddings. *Journal of Machine Learning* **81**(1), 21–35 (2010)
30. Zhou, X.S., Huang, T.S.: Relevance feedback for image retrieval: A comprehensive review. *Journal of Multimedia Systems* **8**(6), 536–544 (2003)