HEAT: Iterative Relevance Feedback with One Million Images

Nicolae Suditu and François Fleuret
Idiap Research Institute and École Polytechnique
Fédérale de Lausanne (EPFL), Switzerland

Abstract

It has been shown repeatedly that iterative relevance feedback is a very efficient solution for content-based image retrieval. However, no existing system scales gracefully to hundreds of thousands or millions of images.

We present a new approach dubbed Hierarchical and Expandable Adaptive Trace (HEAT) to tackle this problem. Our approach modulates on-the-fly the resolution of the interactive search in different parts of the image collection, by relying on a hierarchical organization of the images computed off-line. Internally, the strategy is to maintain an accurate approximation of the probabilities of relevance of the individual images while fixing an upper bound on the required computation.

Our system is compared on the ImageNet database to the state-of-the-art approach it extends, by conducting user evaluations on a sub-collection of 33,000 images. Its scalability is then demonstrated by conducting similar evaluations on 1,000,000 images.

1. Introduction

It has become evident in recent years that image retrieval systems must evolve beyond the capabilities of the straightforward text-based surrogates. In particular, they should be able to deal with automatically extracted features while providing an intuitive and simple interaction with the users.

Research started to tackle this challenge via automatic tagging based on annotation propagation [17, 13, 18]. 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. Large scale relevance feedback remains highly desired [15, 19, 5].

We propose an extension of an innovative retrieval approach proposed by Ferecatu and Geman [9, 10] which has the major advantage of being query-free. Starting from an heuristic sampling of the collection, it does not require any explicit query, and it relies solely on an iterative relevance feedback mechanism. At each iteration, the system displays a small set of images and the user chooses the image that best matches what she is looking for. The system updates an internal state and displays a new set of images accordingly. After a few iterations, the sets of displayed images are gradually concentrated on images that satisfy the user.

At the core of this approach, there are two components. First, there is the model to compute the probability for an image to be relevant to the user given what images have been shown to her until now and what she has chosen. Second, there is the strategy to select what images to show her given the estimates of the probabilities of relevance of all the images in the collection.

In the original approach, these two components require a computational effort that grows quasi-linearly with the size of the collection. Since these two components are involved in the on-line interaction with the user, the original approach can not be practically recommended for collections much larger than about 60,000 images.

The novel approach we propose computes a hierarchical organization of the images off-line. At each iteration of the on-line retrieval process, it selects a "trace" in this hierarchy that corresponds to a partition with a fine resolution in the parts that are rich in relevant images and a coarse resolution in the parts that are clearly discarded by the model.

Experiments show that the required size of the trace for maintaining the same retrieval performance is very modest when compared to the total number of images in the collection. Moreover, one can control explicitly the trade-off between the computational effort and the retrieval performance by bounding the cardinality of the trace.

This paper is structured as follows. In § 2 we present existing techniques related to the problem at hand, and summarize in § 3 the notation and the essence of the technique we are extending. In § 4 we elaborate our approach, and present in § 5 our experimental results. We conclude in § 6.

2. Related work

Whereas relevance feedback is a very efficient solution for content-based image retrieval, no existing system scales gracefully to hundreds of thousands or millions of images [15, 19, 5, 6]. Moreover, the relevance feedback is traditionally seen as a post-retrieval mechanism for refining the

Table 1. Notation

Ω	complete set of images, where the images are identified by their indexes $\{1, 2, \dots k, \dots\}$
$S \subset \Omega$	set of images that the user is looking for
$D_t \subset \Omega$	set of images shown to the user at iteration t
$x_t^* \in D_t$	image chosen by the user at iteration t
$\mathcal{N}$	complete set of nodes of the hierarchical tree
$C(N)$	children nodes of node N
$\Omega(N)$	set of images associated to node N

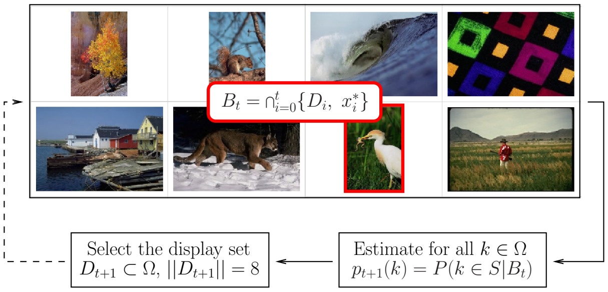

$B_t = \cap_{i=0}^{t}\{D_i,\ x_i^*\}$
Select the display set $D_{t+1} \subset \Omega$, $||D_{t+1}|| = 8$
Estimate for all $k \in \Omega$ $p_{t+1}(k) = P(k \in S|B_t)$

Figure 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 then it will select the new display set D_{t+1}.

retrieved results of an initial query formulated explicitly.

Our research is related to the innovative idea of searching images without any explicit query, which was pioneered by Cox et al. [4]. The core of their work is a Bayesian framework for iterative relevance feedback. Ferecatu and Geman [9, 10] extended the framework and provided theoretically sound interpretations. Moreover, they conducted user evaluations that demonstrate the retrieval capabilities of such an approach.

The contribution of this paper is an extension of the relevance feedback mechanism that uses the Bayesian framework on top of 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 [12], to the best of our knowledge there is no system that uses such a concept in order to scale up relevance feedback mechanisms.

The closest research we found related to the idea of a dynamically adaptive and traceable cut within a hierarchical tree-like organization is in the field of information visualization and visual data mining, where it is referred to as a tree map [16] and other equivalent terms like fish-eye [1] or tree view [3].

Apparently, it is well accepted by the research community that the advantages of such hierarchically structured organizations break down in the face of the high-dimensional image feature spaces that are typically seen in content-based retrieval. However, in comparison with other relevance feedback mechanisms, the work of Ferecatu and Geman [9, 10] has the specificity of dealing explicitly with the miss-alignment between the image feature space and the user subjective perception of image similarities. This fact encouraged us to look again into this research direction.

3. Relevance feedback framework

This section presents briefly the Bayesian framework of the retrieval process proposed in [10].

Given a collection of images $\Omega = \{1, 2, \dots k, \dots\}$, the objective of the retrieval process is to identify the small subset $S \subset \Omega$ containing all the images that the user is looking for. The backbone of the retrieval process is the Bayesian framework in which the probabilities of relevance of all the images in the collection are estimated as conditional probabilities depending on the relevance feedback events.

3.1. Posterior probabilities of relevance

Relevance feedback events are accumulated iteratively as shown in Figure 1. After the system displays a small 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 most similar to S, and this event is denoted as $\{D_t,\ x_t^*\}$. The cumulative event up to iteration t can be expressed as:

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

A sensible technique to select what images to display next in D_{t+1} is to use an estimate of the marginal conditional probabilities of relevance $p_{t+1}(k) = P(k \in S \mid B_t)$. The displayed images should at the same time concentrate on the relevant images and maintain some exploratory sampling among the non relevant images. To simplify the analysis of our work, we use for that matter the exact same model as in [9, 10], 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.

3.2. Selection of the displayed images

At each iteration t, the set of displayed images D_t is generated via a Voronoi tessellation algorithm proposed by Fang and Geman [8]. Instead of simply selecting the images with the highest probabilities of relevance, this algorithm samples the image collection with the purpose of maximizing the efficiency of the relevance feedback events.

The algorithm selects the images $x \in D_t$ by growing subsequent 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 of the total probability mass:

$$\frac{1}{||D_t||} \cdot \sum_{k \in \Omega} p_t(k) \qquad (2)$$

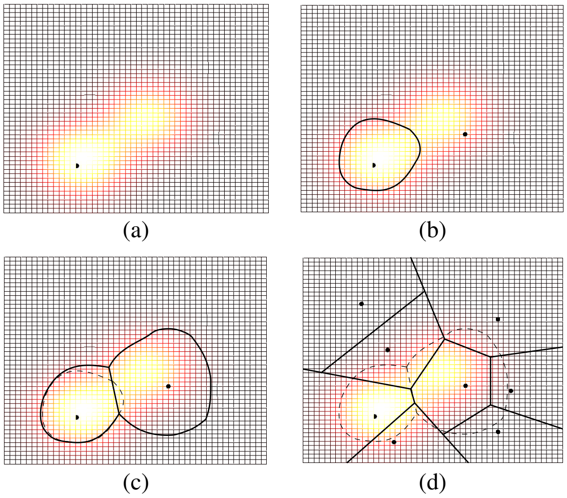


Figure 2. The set of displayed images is generated via the Voronoi tessellation algorithm. Here, the algorithm is illustrated using an abstract representation where each image is a point in the 2D Cartesian space and the similarity distances between images are simply the Euclidean distances between their corresponding points. (a): The first image x_0 is selected; (b): The first Voronoi cell $\mathcal{C}_0$ is grown and 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 $\mathcal{C}_1$, and then re-grown by including other images. (d): The set of displayed images is complete.

The first selected image is the image with the highest probability in the entire collection Ω:

$$x_0 = \underset{k \in \Omega}{\operatorname{argmax}} \; p_t(k) \tag{3}$$

and the Voronoi cell $\mathcal{C}_0$ is grown by including images one by one, as ordered by their similarity distances to x_0 in increasing order, until its probability mass reaches the optimum.

The second image is selected among the images from outside the first Voronoi cell:

$$x_1 = \underset{k \in \Omega \setminus \mathcal{C}_0}{\operatorname{argmax}} \; p_t(k) \tag{4}$$

and the Voronoi cell $\mathcal{C}_1$ is grown by including images in a similar manner. The algorithm loop continues until the set of images D_t is complete as illustrated in Figure 2.

4. Scalable system

The original approach requires a computational effort that is tightly related to the size of the collection. On one hand, 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 $\mathcal{O}(n^2)$ complexity off-line or computational effort of $\mathcal{O}(n)$ on-the-fly. On the other hand, the Voronoi tessellation algorithm involves sorting operations of $\mathcal{O}(n \log n)$ complexity over the entire collection.

While maintaining all the core operations basically the same, 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 Figure 3. Formally, $C(N)$ denotes the set of children of the node $N \in \mathcal{N}$, $C(N) \subset \mathcal{N}$. Each node N represents a set of images that is denoted by $\Omega(N) \subset \Omega$. Each leaf node represents one single image. If N is a leaf node, then $C(N) = \emptyset$ and $||\Omega(N)|| = 1$. These sets of images are hierarchically disjunctive and they naturally respect the property:

$$\cup_{M \in C(N)} \Omega(M) = \Omega(N) \tag{5}$$

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 that stands for a complete and disjunctive partitioning of the image collection:

$$\forall A, B \in \mathcal{T}, \; A \neq B, \; \Omega(A) \cap \Omega(B) = \emptyset \tag{6}$$

$$\cup_{A \in \mathcal{T}} \Omega(A) = \Omega \tag{7}$$

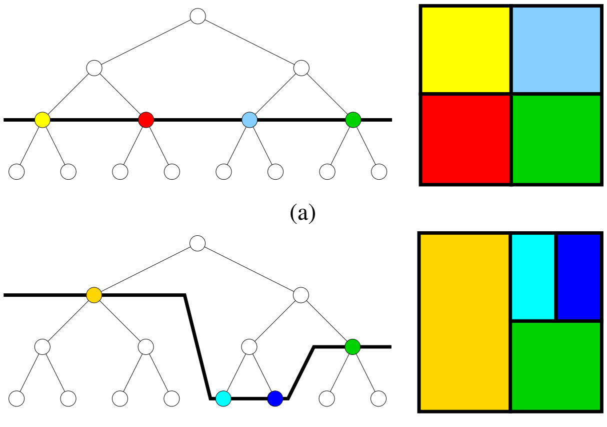
This definition guarantees that any image in the collection is represented by one and only one node in any trace. If $N \in \mathcal{T}$, then it represents all its associated images $\Omega(N)$ as explained in Figure 3.

4.2. Approximation of p_t

The computational effort is controlled in our approach by estimating the probabilities of relevance only for the representative images of the nodes that are part of the current trace. From this bounded set of probabilities, we both infer a sound approximation of the Voronoi tessellation algorithm described in § 3.2 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 indicated in § 3.1. They are not approximated in any way, and thus they are as if the node N would have been part of the trace since the beginning of the retrieval process.


(a)

(b)

Figure 3. Relation between the hierarchical tree and the 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.

Furthermore, the prerequisites of the Voronoi tessellation algorithm described in § 3.2 are reconsidered as follows. The probability mass of a node N is approximated as:

$$q(N) = \sum_{k \in \Omega(N)} p_t(k) \approx p_t(k_N^*) \cdot ||\Omega(N)|| \qquad (8)$$

The probability mass of the entire collection is approximated as:

$$q^{all} = \sum_{k \in \Omega} p_t(k) \approx \sum_{N \in \mathcal{T}} q(N) \qquad (9)$$

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

$$q^{opt} = \frac{1}{||D_t||} \cdot q^{all} \approx \frac{1}{||D_t||} \cdot \sum_{N \in \mathcal{T}} q(N) \qquad (10)$$

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

$$q(N) = \sum_{M \in C(N)} q(M) \approx \sum_{M \in C(N)} p_t(k_M^*) \cdot ||\Omega(M)|| \qquad (11)$$

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 into a coarser approximation:

$$\sum_{M \in C(N)} q(M) = q(N) \approx p_t(k_N^*) \cdot ||\Omega(N)|| \qquad (12)$$

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 under the constraint of preserving a bounded size of the trace. Intuitively, this is achieved when the variances of the probabilities within each node in the trace are as small as possible, 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 they may be replaced by their parent. Recursively, one at a time, the set of children that minimizes the mean-variance cost function:

$$\operatorname*{argmin}_{\forall N,\ C(N) \subset \mathcal{T}} \ \mu(N) \cdot (\sigma^2(N) + \epsilon \cdot ||\Omega(N)||) \qquad (13)$$

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 then it is 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.

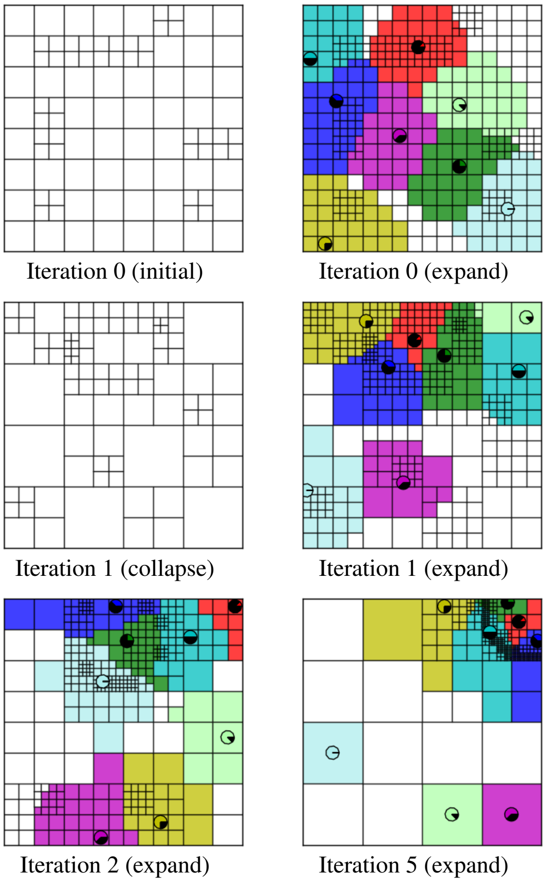
The probability mean and variance of each node are estimated based on its children:

$$\mu(N) = \frac{\sum_{M \in C(N)} p_t(k_M^*) \cdot ||\Omega(M)||}{\sum_{M \in C(N)} ||\Omega(M)||} \qquad (14)$$

$$\sigma^2(N) = \frac{\sum_{M \in C(N)} p_t^2(k_M^*) \cdot ||\Omega(M)||}{\sum_{M \in C(N)} ||\Omega(M)||} - \mu^2(N) \qquad (15)$$

In the Equation (13), 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 collection size.

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.


Iteration 0 (initial)

Iteration 0 (expand)

Iteration 1 (collapse)

Iteration 1 (expand)

Iteration 2 (expand)

Iteration 5 (expand)

Figure 4. Evolution of the trace for the synthetic collection, when searching for a point located in the upper right corner. At iteration 0, the trace is initialized randomly. At each iteration, the current trace is collapsed and expanded, the probabilities of relevance are updated, and then the new images to be shown are selected. After 5 iterations, the trace concentrates mostly in the intended region.

4.4. Algorithm integration

The skeleton of our proposed approach is as follows:

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, a synthetic image collection comes in handy, where the images have as content one single point at a certain location, and the indexing features are the corresponding 2D Cartesian coordinates. Figure 4 shows how the trace evolves at each iteration and how the image collection is sampled at different resolutions in different regions.

5. Experimental results

The retrieval systems are developed as a web-application (http://imr.idiap.ch/). Besides the advantage of permanent availability for evaluations, this implementation encourages the adherence to a realistic system architecture. The application software has been published under the GPL v3.0 open-source license at the time of publication of this paper.

The aim of our experiments was to evaluate our system in comparison with the original system in terms of both the retrieval performance and the computational effort. Regarding the retrieval performance, we looked for evidence that our extension is capable of providing a retrieval performance comparable to the original one. Regarding the computational effort, we looked for evidence that our extension is capable of scaling up beyond two orders of magnitude.

The experiments were organized with two collections obtained from the ImageNet database [7] that has the convenience of being structured in 1000 semantic categories, where each category has 500–2500 images. Considering the subset of 1,200,000 images provided with pre-computed SIFT features (Scale Invariant Feature Transform) [14], we obtained a *large collection* including about 1,054,000 images, namely all the images with valid url at that date. Then, we sampled a *small collection* of 33,000 images (i.e. 3% of the large collection) with the guarantee of being similarly and proportionally populated as the large collection.

5.1. System setup

The image similarity distances are defined simply as the Euclidean distance between the histogram-like feature vectors (i.e. bags of visual words) of dimension 1000, as they are provided by the ImageNet database.

The relevance feedback framework is calibrated as described in [9], and the parameters of the probability model are adjusted to saturate only after including on average 10% of the images in the collection.

The hierarchical tree is generated by applying a divisive top-down k-means algorithm. The tree is initialized with the root node as being the single node and representing all the images in the collection. Recursively, the images of each node are split in 8 k-means clusters. These resulting clusters are used to define new nodes, one level deeper in the tree.

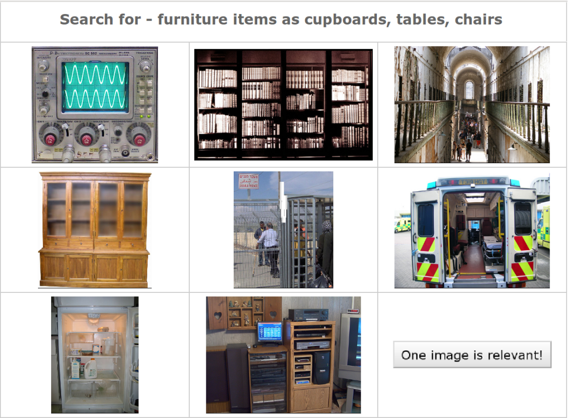

Search for - furniture items as cupboards, tables, chairs

Figure 5. Web interface of the retrieval system. The users were asked to search for semantic categories described only in words.

Naturally, the former node becomes a parent node with the newly defined nodes as its children.

Considering the size of the collection, we employ an approximation of k-means that is studied in terms of clustering feasibility and computational complexity in [11]. The clustering of sets of more than 50,000 images is done in two phases. In the first phase, k-means is initialized randomly and then performed – until convergence – on a random sample of 50,000 images in order to obtain an estimate of the centroids. In the second phase, k-means is initialized with the estimated centroids and then performed – only 2 iterations – on the full set of images.

5.2. Evaluation scenario

The evaluation was conducted with 20 users not familiar with the system, and it consisted of running user tests with three systems: our proposed system, the original system and a system displaying images randomly without replacement. The random system discards totally the relevance feedback and provides the lowest base-line performance.

In order to ensure a sufficiently reliable diversity, there were 6 semantic categories described only in words:

- domestic dogs in close-up portrait
- electronic devices as TV, radio, mobile
- big boats as ferryboats, cargoes
- exotic fruits in close up portrait
- furniture items as cupboards, tables, chairs
- public buildings as shops, malls

In order to ensure comparable difficulty, these categories were chosen to be relevant for 1–2% of the image collections based on the evidence given by the cardinality and the associated keywords of the ImageNet categories.

In order to avoid any bias, the searching sessions were presented in a random fashion. The semantic categories, the systems and the collections were randomized all together in one single user test. The users were not aware of which combination was active at a certain time. In fact, they were not introduced to anything beyond the evaluation interface in Figure 5. The interpretation of the semantic categories in the sense of visual content was left to the user. The users were only told to end the searching sessions when they were satisfied by at least one of the displayed images.

5.3. Performance impact

The experiments with the small collection show that our system preserves with fidelity the retrieval capabilities of the original system. Moreover, both systems outperform by far the random display of images. In 80% of the cases, both systems succeed to display a relevant image after 8 iterations, while the random one requires more than 16 iterations. The average performances are shown in Figure 6.

The experiments with the large collection show that our system provides a sustainable performance where the original system proposed by Ferecatu and Geman [9, 10] would cease to function within any reasonable timeframe.

The random system shows similar performance for both collections. Since both collections have a similar semantic diversity based on the ground truth given by the ImageNet, this is exactly what one would expect. Considering the randomized organization of the evaluations, the agreement of the two random baselines gives evidence that the users were consistent among the searching sessions and the performance curves are reliable.

Our evaluation scenario was meant to compare the capability of the systems to converge to semantic categories of a relatively small size. The users were told precisely to end the searching sessions the first time they were satisfied by one of the displayed images. Further evaluations should be conducted in more demanding scenarios.

For the experiments with the small collection, the trace was limited to collapse at minimum 500 nodes, and this means that each expansion included about 3000–4000 nodes. This variation in the number of nodes comes from the fact that the hierarchical tree is unbalanced. For the experiments with the large collection, the trace was limited to collapse at minimum 1000 nodes, and this means that each expansion included about 6000–8000 nodes. We observed that in order to maintain a similar retrieval performance the size of the trace should be slightly increased. It may be due to the larger tree that more nodes are inefficiently used just for maintaining the continuity of the trace. Further evaluations should certainly address this issue.

5.4. Computational impact

The system responses were timed during the user experiments. Although our implementation can be further optimized, these timings give a tangible evaluation of the computational effort of the systems as shown in Figure 7.

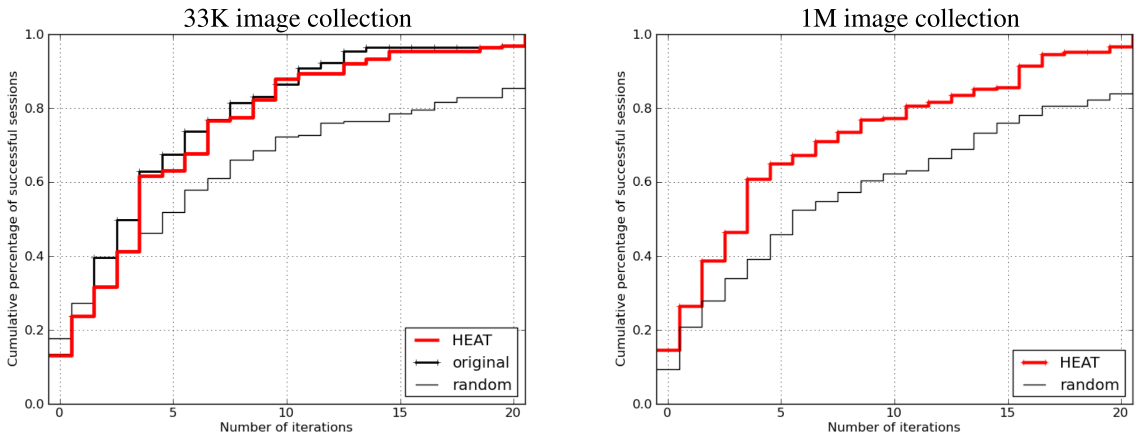


Figure 6. Cumulative percentage of successful sessions per number of iterations. The average performances for the small collection are shown on the left: Our system performs as well as the original system proposed by Ferecatu and Geman [9, 10]. Both systems outperform by far the random display of images. In 80% of the cases, both systems succeed to display a relevant image after 8 iterations, while the random one requires more than 16 iterations to achieve the same performance. The average performances for the large collection are shown on the right: Our system shows a sustainable performance against the random baseline.

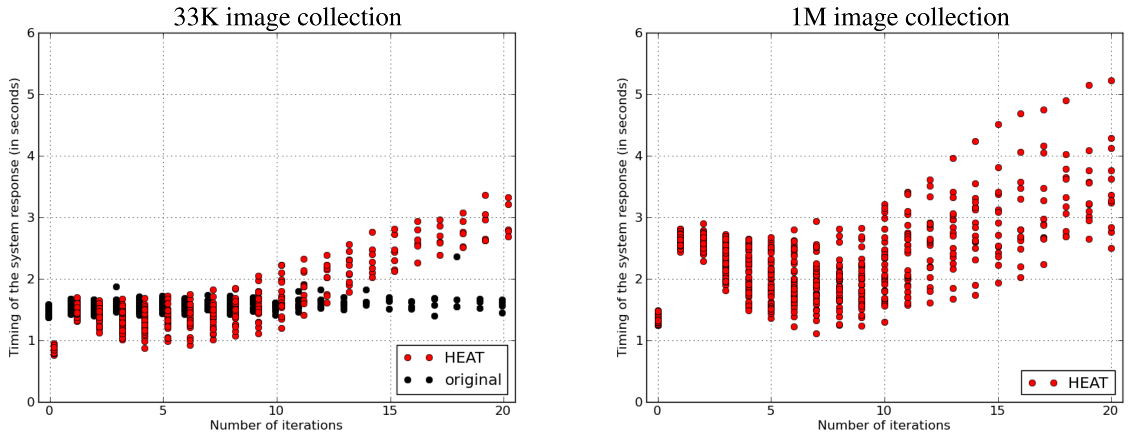


Figure 7. Timing of the system responses (in seconds) as the users experienced it during the evaluations. The timings for the small collection are shown on the left: The computational effort of the original system is constant over the iterations. The computational effort of our system stays in the same range, although it increases slowly with the number of iterations due to the computation from scratch of the probabilities of relevance. The timings for the large collection are shown on the right: The timings remain comparable with the ones for the small collection. The computational effort of our system is decoupled from the collection size and it depends mainly on the trace size.

The computational effort of the original system is rather constant over the iterations. It has to update the conditional probabilities and to perform the Voronoi tessellation based on a constant number of images, namely the size of the collection. For the small collection, the system responds in about 1.5 seconds. For the large collection, forgetting the required storage capacity of $\mathcal{O}(n^2)$ complexity, the system would totally fail to respond in any reasonable time.

The computational effort of our system is slightly variable over the iterations. Although it has to update the conditional probabilities and to perform the Voronoi tessellation only based on the representative images and the cardinality of the nodes in the current trace, the system has to access the image feature vectors and to compute the similarity distances on-the-fly. Moreover, the computation from scratch of the conditional probabilities is linearly dependent on the number of iterations. While the original system updates recursively the probabilities only based on the last relevance feedback event, our system updates most of the probabilities from scratch based on the full history of relevance feedback events. One can observe that the nodes in the trace are constantly replaced by the refinement operation.

For a complete view of the computational complexity, the pre-processing required for organizing and indexing the image collections should be taken into account as well. As mentioned already in § 5.1, the image feature vectors are provided by the ImageNet database, and thus their computation is not taken into account here.

The pre-processing in the original system consists of computing the similarity distances between every two images in the collection, and thus it has $\mathcal{O}(n^2)$ complexity. For the small collection, the required capacity for storing the similarity distances in binary files, one file per image, is nearly 4GB. For the large collection, the required storage capacity would be unacceptably large.

The pre-processing in our system consists of building the hierarchical tree based on k-means clustering. The computational complexity of the divisive top-down k-means clustering does not have a closed form but it is studied in [11]. The storage of the image feature vectors has $\mathcal{O}(n)$ complexity, and the storage of the hierarchical tree is truly negligible. The required capacity is only 100MB for the small collection and about 3GB for the large collection.

6. Conclusion

We have presented a retrieval approach that promises an interactive access to image collections of unprecedented size. The experiments show that this iterative relevance feedback mechanism can handle a collection of 1,000,000 images, which is already one order of magnitude larger than most of the state-of-the-art iterative approaches.

Using an adaptive partitioning of the image collection, we provide the means for controlling the trade-off between the retrieval performance and the computational effort. This may be a crucial characteristic for real-world applications.

We foresee no barrier in scaling up the approach up to 10,000,000 images or more. The key observation is that the trace refinement is suitable for parallel and distributed computing architectures. The trace could be divided into parts, and each part could be processed separately.

Acknowledgments

Nicolae Suditu was supported by the Hasler Foundation through the EMMA (Enhanced Medical Multimedia data Access) 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.

References

[1] J. Abello, S. G. Kobourov, and R. Yusufov. Visualizing large graphs with compound-fisheye views and treemaps. In *Graph Drawing*, volume 3383 of *Lecture Notes in Computer Science*, pages 431–441. 2005. 2

[2] B. B. Bederson. PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the 14th ACM symposium on User interface software and technology*, pages 71–80, 2001. 2

[3] A. L. Buchsbaum and J. R. Westbrook. Maintaining hierarchical graph views. In *Proceedings of the 11th ACM-SIAM symposium on Discrete algorithms*, pages 566–575, 2000. 2

[4] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000. 2

[5] M. Crucianu, M. Ferecatu, and N. Boujemaa. Relevance feedback for image retrieval: a short survey. In *State of the Art in Audiovisual Content-Based Retrieval, Information Universal Access and Interaction including Datamodels and Languages*, 2004. 1

[6] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, April 2008. 1

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[8] Y. Fang and D. Geman. Experiments in mental face retrieval. In *Proceedings of the 5th International Conference on Audio and Video-based Biometric Person Authentication*, pages 637–646, July 2005. 2

[9] M. Ferecatu and D. Geman. Interactive search for image categories by mental matching. In *Proceedings of the IEEE 11th International Conference on Computer Vision*, pages 1–8, October 2007. 1, 2, 5, 6, 7

[10] M. Ferecatu and D. Geman. A statistical framework for image category search from a mental picture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1087–1101, June 2009. 1, 2, 6, 7

[11] A. Goswami, R. Jin, and G. Agrawal. Fast and Exact Out-of-Core K-Means Clustering. In *IEEE International Conference on Data Mining*, pages 83–90, November 2004. 6, 8

[12] D. Heesch. A survey of browsing models for content based image retrieval. *Journal of Multimedia Tools and Applications*, 40(2):261–284, 2008. 2

[13] J. Li and J. Z. Wang. Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008. 1

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. 5

[15] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Video Technology*, 8(5):644–655, 1998. 1

[16] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, January 1992. 2

[17] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000. 1

[18] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Journal of Machine Learning*, 81(1):21–35, 2010. 1

[19] X. S. Zhou and T. S. Huang. Relevance feedback for image retrieval: A comprehensive review. *Journal of Multimedia Systems*, 8(6):536–544, 2003. 1