Similarity Query Postprocessing by Ranking

Petra Budikova, Michal Batko, and Pavel Zezula

Faculty of Informatics, Masaryk University
Brno, Czech Republic

Abstract. Current multimedia search technology is, especially in commercial applications, heavily based on text annotations. However, there are many applications such as image hosting web sites (e.g. Flickr or Picasa) where the text metadata are of poor quality in general. Searching such collections only by text gives usually rather unsatisfactory results. On the other hand, multimedia retrieval systems based purely on content can retrieve visually similar results but lag behind with the ability to grasp the semantics expressed by text annotations. In this paper, we propose various ranking techniques that can be transparently applied on any content-based retrieval system in order to improve the search results quality and user satisfaction. We demonstrate the usefulness of the approach on two large real-life datasets indexed by the MUFIN system. The improvement of the ranked results was evaluated by real users using an online survey.

Keywords: ranking, content-based retrieval, metric space

1 Introduction

With the rapid growth of volume and diversity of the digital data, the need of efficient storage and retrieval methods is indisputable. The traditional databases are not suitable for many new complex data types, such as multimedia, DNA sequences, time series, etc. Therefore, new methods of data management have been intensively researched in recent years.

Multimedia retrieval systems usually use one of two general approaches. The first one applies existing text-search mechanisms to retrieve the data based on the descriptive annotations. Recently, this approach was enhanced using result ranking with respect to content-based similarity [4]. Of course, the quality of results depends on the quality of text metadata, which is often not very high (especially in large general-purpose collections, such as web image galleries).

The second approach retrieves data by content. Data objects are indexed and searched using features extracted from the data that describe their important characteristics. The query-by-example paradigm is usually used for searching, which enables a natural definition of a complex object query, e.g. an image. The so-called content-based searching has been developing rapidly in recent years and has already grown to the web dimension. However, it suffers from the well-known *semantic gap* problem, i.e. the discrepancy between the similarity as

computed using the descriptors and human understanding of similarity [14]. Existing solutions try to bridge the gap using semantics-learning mechanisms [8] or iterative query refinement using relevance feedback [18].

While the text-based searching can be very successful in some applications, its obvious drawback is that it cannot be used on data where the text metadata is of low quality or not available. Here, the content-based approach is the only possibility. To overcome the semantic gap problem, the search engine can be trained to recognize semantic categories. However, there are a number of scenarios when the semantics-learning cannot be employed, e.g. in case of large datasets with many ambiguous semantic categories, where the computer learning is infeasible. Therefore, we need a general solution for fast content-based searching in data with no (or poor) semantic information as a fall-back option for situations where more precise approaches cannot be used.

Even though the concept of similarity is subjective and context-dependent, the search engine usually employs a general measure of similarity to provide fast retrieval. As a result, the retrieved objects are similar to the query in some ways but may not be the most relevant according to the user. To demonstrate this, imagine a user who searches for images of red apple and, based on visual similarity, the system provides tomatoes and red balls in addition to red apples.

In this paper, we propose to overcome this problem by combining several views on the relative importance of target objects. This method has already been proved to be very successful in the text-based searching but has not yet been used in a large-scale content-based retrieval. To provide efficient searching, we first retrieve a candidate set of objects using a general similarity measure. In the next step, other measures are applied on the initial result to adjust the ranking of the objects so that the most relevant results are displayed to the user. This solution has a number of advantages:

- **Generality**: Since the existing technologies for content-based searching are typically based on the metric model, this approach enables to search efficiently in a wide scope of data domains, ranging from multimedia to DNA sequences. Even more general (non-metric) measures can be used in the ranking phase where only a small number of objects needs to be processed.
- **Query-by-example search**: In many data domains, it is often difficult to describe the required objects by text or other attributes – "an image is worth a thousand words". The content-based search enables an example object to be used to define the query.
- **Multi-modal searching**: The ranking concept enables to combine more similarity measures. In particular, it can easily employ similarity measures that are computationally expensive as well as to use information that is not rich enough to provide a full-fledged result on its own.
- **Flexibility**: There are several sources of information that can be exploited in the search process. The search engine has the knowledge of the data properties, can compute distances between pairs of objects, and use statistical information about the collection. In addition, the system can interact with the user or other systems to adjust or re-evaluate the ranking procedure.

The rest of the paper is organized as follows. In Section 2, we discuss the most relevant related work. We briefly introduce the content searching based on metric space similarity in Section 3. Next, we formalize our concept of two-phase searching in Section 4 and propose a basic classification of ranking methods. User-satisfaction experiments with several ranking functions are described and evaluated in Section 5.

2 Related Work

The concept of query result post-processing and ranking has been employed in a number of search applications and strategies, both in text-based and similarity-based retrieval. Most of the research has been done in image and video searching, which is attractive for many users. In the text-based approaches, ranking is often used to prioritize objects from the result that have similar visual content as the query object. In content-based strategies, various post-processing methods try to bridge the semantic gap and identify the most relevant objects.

The text-based search in images has been provided by many web search engines for years. Recently, some of the major search engines (Google[1], Bing[2]) launched a new type of searching based on visual similarity of images. Both solutions exploit visual ranking of search results acquired by text retrieval. The Google approach [4] employs local image descriptors to measure the visual similarity of images. The famous *PageRank* algorithm idea has been adapted to *VisualRank*, which is used to propagate the similarity relationships in the result. Since the complete evaluation of the ranking algorithm is expensive, the results of visual search are precomputed for the more popular queries.

The Microsoft solution [16] is based on a similar concept, this time using both local and global visual image descriptors to rank objects retrieved by the initial text search. To obtain more precise results, the descriptors receive weights that express their importance for that particular image set. Again, the image set is modeled by a visual-similarity graph and the similarity information is propagated to identify the most important nodes.

An interesting extension to these methods has been proposed in [9]. The authors argue that the results of the visual ranking are often not satisfactory, which is caused by the fact that the initial text-based search result is not good enough to allow detecting important patterns for ranking. To overcome this, they propose to combine results from multiple web search engines and provide the *CrowdRanking* algorithm which identifies important visual features and ranks the results.

The text information associated with multimedia objects is often in the form of *tags*, i.e. keywords provided by users. Tagging is popular especially in social media repositories such as Flickr[3] and can be exploited in search processes. The authors of [6] investigate ways of differentiating between content-related and

[1] http://images.google.com/
[2] http://www.bing.com/images
[3] http://www.flickr.com

content-unrelated tags by the means of WordNet relationships between semantic concepts. Another study [7] explores the ways of determining ranking of tags according to their relevance.

One of the weaknesses of text-based search is the ambiguity of search terms. An active ranking strategy was proposed in [15] where a small set of images that represent different concepts is chosen from the result of the text-based search and displayed to the user for evaluation of relevance. The user input is used to disambiguate the search.

In case of results obtained by content-based searching, the post-processing methods try to filter out less interesting objects from the result, usually by means of result clustering. Two quality aspects that are mostly addressed are the presence of too many near-duplicates in the result set and the occurrence of objects that are not relevant from the user's point of view. For the first problem, a definition of a new *distinct nearest neighbors* query is provided in [13]. Such query only returns distinct objects, i.e. objects with mutual distances greater than some predefined constant denoted as the separation distance. To eliminate the less relevant objects, several methods have been proposed that try to analyze the relationships between the objects in the result set and identify the ones that are most important in some sense, e.g. they are most similar to the rest of the result. In [12], four methods of result ranking using clusters are presented, e.g. by penalizing objects in clusters other than the query object's cluster. The authors of [5] propose to use dynamic clustering, where the distance function for clustering is chosen with respect to the importance of individual features for the given query.

3 Content-based Searching using Metric Spaces

In our approach, the similarity is modeled by using a generic metric space abstraction. The image visual descriptors that are usually used in the field of image retrieval, e.g. the global descriptors defined in the MPEG-7 [10], in fact satisfy properties of the metric spaces and thus can be used in metric-based indexing engines. In order to work with very large collections of data, we employ the scalable indexing infrastructure of the Multi-Feature Indexing Network [11]. It is a versatile and highly modular similarity framework built on top of MESSIF library [2] which provides indexing layers as well as user and programming interfaces. Since the system works with any metric data, it allows us to work with a wide variety of data types including images.

3.1 Metric space approach

The metric space [17] is considered to be the most general data model for similarity search which can still be indexed and searched efficiently. The model treats its data as unstructured objects together with a function which measures proximity of object pairs. Formally, the *metric space* $\mathcal{M}$ is a pair $\mathcal{M} = (\mathcal{D}, d)$, where $\mathcal{D}$ is the *domain* of objects and d is a total *distance function* $d : \mathcal{D} \times \mathcal{D} \longrightarrow \mathbb{R}$ satisfying

the following postulates for all objects $x, y, z \in \mathcal{D}$: the reflexivity $d(x, x) = 0$, the strict positiveness $d(x, y) > 0$, the symmetry $d(x, y) = d(y, x)$, and the triangle inequality $d(x, y) \leq d(x, z) + d(z, y)$.

The semantics of this concept is: The smaller the distance between two objects, the more similar they are. The metric space is typically searched by queries which follow the query-by-example paradigm. A query is formed by an *object* $q \in \mathcal{D}$ and some *constraint* on the data to be retrieved from the indexed dataset $X \subseteq \mathcal{D}$. There are two basic types of these queries: (1) the *range query* $R(q, r)$, which retrieves all objects $o \in X$ within the range r from q, and (2) the *nearest neighbors query* $kNN(q, k)$, which returns the k objects from X with the smallest distances to q.

3.2 Multi-Feature Indexing Network

The Multi-Feature Indexing Network (MUFIN) [11] is a general purpose search engine that exploits results of more than ten years of research of the metric-based techniques. It allows to plug-in different metric indexing techniques and bind them together by various programming interfaces into a coherent system. A strong emphasis is put on scalability, so the system can process data collections of really big volumes. Moreover, the system provides easy-to-use user interfaces that can be used to offer the searching capabilities to any user. The system is also able to gather various statistics and thus can be easily adjusted to perform user satisfaction surveys and testing new searching paradigms.

In this paper, we use two instances of the MUFIN system for evaluating the results of content-based similarity queries. The capabilities of MUFIN allowed us to retrieve the results very fast even for large collections of data and its interfaces made it possible to plug-in the proposed ranking algorithms seamlessly into the web user interfaces.

4 Ranking

Ranking is often considered an integral part of the search process – search engines retrieve ranked results. However, we argue that it is more convenient to treat searching as a two-phase process, distinguishing between the initial search phase, which retrieves suitable candidates, and the ranking phase. The crucial difference between these phases is that in the first phase, the whole dataset is searched while only a small subset is processed in the second phase.

Let us now formalize the two search phases as functions over the data space $\mathcal{M} = (\mathcal{D}, d)$. The initial search $F_{initial}$ may be performed by any standard metric search query operation returning a set of k objects relevant to the given query object q, e.g. the k-nearest neighbor search (kNN) or the range search. In our framework, we choose the kNN search as the most convenient, since users do not need any prior knowledge about the distances in the dataset.

$$F_{initial}(q) = R \subseteq \mathcal{D}, |R| = k$$

In the ranking phase, a function $F_{rank} : \mathcal{D} \mapsto \mathbb{N}$ is applied on the result of the initial search $F_{initial}$ to establish a new rank of each object. In fact, the ranking function depends on the *context* in which it is evaluated and its computation may contain additional context-derived parameters. To increase the readability we relax the strictness of the function definition by including the context parameters in $RANK_{type}$ function as needed. We will discuss the possible context parameters later.

$$F_{rank}(o) = RANK_{type}(o, context) = i,$$
$$i \text{ is the rank of the object } o \in F_{initial} \text{ in the given context}$$

The ranking function F_{rank} must satisfy the following *unambiguity condition*:

$$\forall o_1, o_2 \in F_{initial}(q) : (F_{rank}(o_1) = F_{rank}(o_2)) \Rightarrow (o_1 = o_2).$$

Even though the user is interested in the first k objects with k typically ranging from 10 to 100, the initial search should provide significantly more objects in order to allow the ranking to show interesting new data. Note that the larger the initial result set is, the higher the chances of having more relevant objects are. On the other hand, if the initial result is too large, the post-processing step might be too costly. Therefore, the choice of the initial result size k' needs to balance the following three factors: the costs of the initial search for k' best objects, the cost of ranking the k' objects, and the probability that there are at least k relevant objects in the initial result of size k'.

In the following sections, we present several different types of ranking functions that are orthogonal to the content-based similarity. Thus, the visual similarity of the image is supplemented by its semantic content expressed by textual annotation. We split the ranking functions into two categories – functions that can automatically rank the initial results based on the retrieved data and user-defined ranking where users actively participate in the process of defining the ranking function.

4.1 Automatic ranking

As automatic we denote ranking methods that compute the result ranking using only the query context information, i.e. the query definition and the statistical properties of the initial result R. When the initial set is retrieved by a visual content, a successful ranking needs to exploit additional information available for data objects that was not used in the initial content-based search, e.g. keywords, location, searching object popularity, number of purchases of the object, etc. A more sophisticated ranking can try to identify and exploit some patterns in the properties of objects in the initial result, e.g. the most important keywords, or visual features in case of images. Finally, the ranking phase may also include another type of content-based similarity search. Naturally, several ranking functions can be combined to provide the final order of objects.

In the following we focus on text-based automatic ranking in collections with annotations of various quality, which is common in many web applications such as photo galleries.

Keyword ranking Inversely to the search model applied by the common web search engines that combine text-based retrieval and visual ranking, we propose to rank the content-based search result with respect to keywords of the query image. We measure the similarity between two sets of keywords by the Jaccard coefficient (see [17] for a formal definition of the Jaccard similarity).

$$RANK_{queryObjectKeywords}(o, R, q) = i \in \mathbb{N}, i = |X| - 1, X \subseteq R, \forall x \in X : \\ (d_{Jaccard}(q.keywords, x.keywords) \\ < d_{Jaccard}(q.keywords, o.keywords))$$

This ranking method is intended for data with rich and reliable annotations. In order to broaden the ranking range, we apply stemming and use WordNet to retrieve the keywords from semantic relationships as suggested in [6]. Using the WordNet, we also remove all words that are not nouns, verbs or adjectives.

Word cloud ranking For data with sparse and erroneous text metadata, the keyword ranking is not applicable. In this case, we propose to exploit the keywords of all objects in the initial result. The keywords are first cleaned and broadened by WordNet as anticipated above. Then we compute the frequencies of the keywords from all the objects in the initial result. We call the resulting set of keywords with their frequencies the *word cloud*. Finally, the ranking employs the most n frequent words from the cloud (denoted as $R.wordCloud.top(n)$) as the query object words in the text-similarity evaluation. Please note that the object keywords $o.keywords$ in the following definition are the keywords of the respective object cleaned by the WordNet as described above.

$$RANK_{wordCloud}(o, R, q, n) = i \in \mathbb{N}, i = |X| - 1, X \subseteq R, \forall x \in X : \\ (d_{Jaccard}(R.wordCloud.top(n), x.keywords) \\ < d_{Jaccard}(R.wordCloud.top(n), o.keywords))$$

Combined visual and text ranking In the previous methods, we have only used the textual (keyword) information for the ranking, ignoring the initial ranking of the visual (content-based) search. However, since the initial result is retrieved using the kNN query which provides the ranking of its own (the metric distance to the query object q), it may also be useful to add it into the final ranking. Therefore, we enrich the $RANK_{queryObjectKeywords}$ method by summing with the distance of the respective object from the visual space. Note that since the Jaccard measure gives values between zero and one, we need the visual distance to be normalized so that both of the two summed distances influence the ranking accordingly. Thus, we multiply the visual distance by a normalization factor f (e.g. the maximal distance in the dataset).

$$RANK_{queryObjKwAndVisual}(o, R, q) = i \in \mathbb{N}, i = |X| - 1, X \subseteq R, \forall x \in X : \\ (d_{Jaccard}(q.keywords, x.keywords) + f \cdot d(q, x) \\ < d_{Jaccard}(q.keywords, o.keywords) + f \cdot d(q, o))$$

Adjusting the factor f can also be used to strengthen or diminish the impact of the visual descriptors on the ranking. Moreover, the $RANK_{wordCloud}$ can be

modified in a similar fashion resulting in the $RANK_{wordCloudAndVisual}$ function that combines the results of the word cloud ranking with the visual distances.

Adaptive keyword/cloud ranking For datasets where some objects are poorly annotated or there is no annotation at all but some objects have a good metadata, it can be beneficial to adaptively choose a ranking method. Therefore, we propose the following heuristic that combines the previous raking methods. Given the query object's keywords and the word cloud of the initial result, we prepare the set of adaptive keywords A as follows. First, all the cleaned keywords of the query object are inserted. If there are less than c of these, the most frequent cloud words are added. However, the cloud words must exhibit some minimal frequency t to be considered relevant. Note that the WordNet cleaning and enrichment as defined above is used. The final ranking is computed as a combination of the text ranking defined by the described keyword set and the initial visual ranking.

$$
\begin{aligned}
&RANK_{adaptive}(o, R, q, c, t) = i \in \mathbb{N}, i = |X| - 1, X \subseteq R, \\
&\quad A = q.keywords \cup R.wordCloud.top(c - |q.keywords|, t), \forall x \in X : \\
&\qquad (d_{Jaccard}(A, x.keywords) + d(q, x) \\
&\qquad < d_{Jaccard}(A, o.keywords) + d(q, o))
\end{aligned}
$$

4.2 User-defined ranking

As we have discussed in the introduction, the understanding of similarity is subjective and varies in different conditions. Therefore, it is not always possible to obtain the optimal result automatically and the user needs to cooperate with the system. In this case, the system displays the results of the initial search and requires additional user input for the ranking phase. A new query object, a measure of the relevance of the initial result, or a specification of relevant values for associated object metadata are a few examples of possible user input for the ranking phase.

While the user-defined ranking functions can be very powerful, they need attention, knowledge, and time from the user. Therefore, these are only intended as advanced options for more experienced users. In the following subsections, we define two ranking functions for advanced searching in image data with text annotations.

Relevance feedback ranking In some search systems, users can provide a feedback on the relevance of results and ask for a refined result. To provide this, the system uses the relevance information to modify the query object or the similarity measure (see [18] for more details). This may be repeated in several iterations which finally produce a better result but may take a considerable amount of time, as a new query needs to be evaluated in each iteration. Therefore, we propose to implement the relevance feedback as the ranking function. Users choose relevant objects from the initial result and the ranking function defines

the final rank as a function on the content-based similarity to each of the objects marked as relevant.

$$RANK_{relevanceFeedback}(o, R, d_{agg}, [q_1, \ldots, q_n]) =$$
$$i \in \mathbb{N}, i = |X| - 1, X \subseteq R, \forall x \in X :$$
$$d_{agg}(d(q_1, x), \ldots, d(q_n, x)) < d_{agg}(d(q_1, o), \ldots, d(q_n, o))$$

Any monotonic function can be used as the aggregation function d_{agg}, for instance SUM, MIN or MAX.

User-defined keyword ranking Keywords may provide a strong ranking tool but automatic approaches may not always guess the optimal set of words. This method allows users to define the relevant keywords themselves.

$$RANK_{selectedKeywords}(o, R, keywordSet) =$$
$$i \in \mathbb{N}, i = |X| - 1, X \subseteq R, \forall x \in X :$$
$$(d_{Jaccard}(keywordSet, x.keywords)$$
$$< d_{Jaccard}(keywordSet, o.keywords))$$

One way of using this type of ranking is to let the users type any keywords they consider relevant. However, there is a high possibility that their choice will not match the keywords used in the images' metadata. Therefore, we allow the users to choose from the list of keywords contained in the initial result.

5 Experiments

To evaluate the quality of all the ranking functions defined in Section 4, we have organized several user-satisfaction surveys. We have provided a simple web interface where the participants were shown the initial and the ranked result sets and then they had to mark the relevant objects. In the two cases of user-defined ranking, the participants were first asked to choose the relevant objects/words from the initial result and then they evaluated the new ranking. About 40 users of different age, sex and computer skills participated in the experiments.

For the experiments, we used two different datasets. Dataset 1, which comes from a commercial microstock site, contains high-quality images with rich and systematic annotations. This dataset contains 8.3 million images and the content-based similarity is defined as a combination of *color layout*, *scalable color*, *region shape* and *edge histogram* MPEG-7 descriptors [10]. Each image is annotated by about 25 keywords on average. Dataset 2 contains images from the Flickr web site and exhibits worse quality of images and sparse and erroneous keywords. This dataset is formed by 100 million images each of which is represented by five MPEG-7 descriptors (see [3] for more information). The effectiveness of the visual search in Dataset 2 using the MUFIN system was published in [1]. The results indicate that even though the effectiveness is satisfactory, there is still space for improvement.

In each set of experiments, we used 50 randomly chosen query objects. For an easy visualization of several result sets on a screen, we only used a result set

with 10 objects. In the initial nearest neighbor search we always retrieved 200 objects, which were conveyed to the ranking function.

We express the user-perceived quality of each result as the ratio of the number r of objects marked as relevant to the number t of all displayed objects from the result. We denote this measure as the *result quality* throughout this section. Note that this measure is not the same as the well known *precision* metric, since in our case, there is no ground truth to compare with. In fact, the understanding of similarity is highly subjective and therefore each user may have his/her own ground truth. Consequently, 100 % quality may not be reachable using this measure if there are less than t relevant objects in the dataset. Unfortunately, there is no feasible way of determining the individual ground truth sets.

5.1 User-defined ranking

In this section, we summarize the results of the user-satisfaction with the two previously defined user-interactive ranking functions. Apart from evaluating their performance, we also used the experiments to find out about the usefulness of the ranking in principle – we asked the users for their opinion whether they want to try ranking for each result. About 50 % of results over Dataset 2 (the worse one) and 72 % of results over Dataset 1 were considered worth trying; the rest of the result sets was either perceived as already too good (17 % for Dataset 1) or too bad (33 % for Dataset 2). In case of Dataset 2, we remark that the low quality of results as perceived by users is caused by the low quality of some of the randomly picked query images rather than bad performance of the initial searching.

Relevance-feedback ranking We ran a set of experiments on each of the two datasets to test the $RANK_{relevanceFeedback}$. For the Dataset 1 (which is smaller), we also evaluated a *multi-object query* in order to compare the ranking results with the precise evaluation. The multi-object query $mkNN(q_1, q_2, \cdots, q_n, k)$ retrieves k objects that are most similar to a set of given query objects, i.e. objects that have the k lowest sums of distances to each query object $d(q_1, o) + d(q_2, o) + \cdots + d(q_n, o)$. Note that a precise answer for user supplied feedback objects can be retrieved by this query.

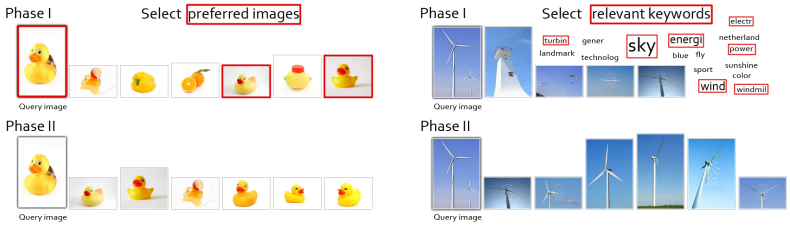

Fig. 1. User-defined ranking: relevance-feedback ranking (left), user-defined text ranking (right)

In the experiments, users were asked to choose any number of relevant objects from the displayed top-10 images from the initial result. Figure 1 shows an initial result set (Phase I) with the user-selected preferences marked by red border and the final result set after the ranking was applied (Phase II). The actual aggregation function used in the ranking process was SUM.

In the following table, we compare the quality of results obtained by initial searching, ranking, and multi-object query evaluation.

	Dataset 1 result quality	Dataset 2 result quality
$R_{initial}$	39.5 %	35.3 %
$RANK_{relevanceFeedback}$	59.2 %	48.0 %
Multi-object query	60.2 %	—

We can observe that the number of relevant objects in the initial result (i.e. ranked by the content-based similarity to the single query object) is increased significantly in both the experiments. Moreover, the ranking produces results of nearly the same quality as the full evaluation of the respective multi-object query, which finds the images most similar to all the query objects selected by the user precisely from the whole dataset. This confirms our assumption that there are enough good objects in the initial result.

Text ranking The ranking based on users' choice of keywords was evaluated only for the Dataset 1. Participants of the experiment were shown the initial result and a set of keywords, which comprised all keywords of the query object combined with the 50 most frequent keywords from the word cloud – we have not shown all the keywords to keep the list accessible. Different font sizes were used for the display of the keywords to emphasize the most frequent ones, as depicted in Figure 1 (right). Users were asked to choose any number of relevant keywords and evaluated the ranked result. The following table summarizes their satisfaction.

	Result quality
Initial result	34.1 %
Ranked result	48.6 %

The results show that the keyword-based ranking increase the user satisfaction by 15 %. On average, the users selected 3-4 words per search and the collected data also indicate that the more keywords were issued, the higher the satisfaction with the result was. About 90 % of all keywords selected by users belonged to the query object keywords. This confirms our assessment of high quality of the annotations in Dataset 1.

Consistency of user's preferences We can make some observations on the behavior of users during the search process, which may be useful for further

improvements of the automatic ranking methods. Although the users were not given any advice on how to decide the relevance, their evaluation of (ir)relevance of given object in a particular result was very consistent – on average, more than 80 % of users agreed on a relevance of a given object. In the phase of selecting words or images for ranking, the same object was chosen as preferred by 60-70 % of users on average. This implies that some kind of a general truth exists that is favorable in most situations. It is therefore realistic and reasonable to develop automatic methods that try to find the relevant objects by the analysis of human preferences.

Selection of initial result size We have also focused on the changes of effectiveness when the size of the initial set k' is increased. In particular, Figure 2 shows the percentage of the relevant objects in the results set as specified by the users when the k' was varied from 10 to 200 on Dataset 1. The same user-defined ranking and $k = 10$ final results as in the previous experiment were used.

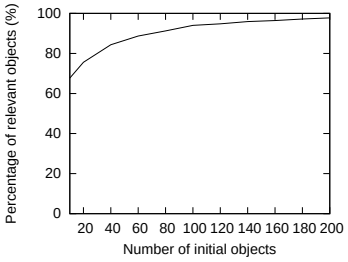

Fig. 2. Influence of the initial result set size on the number of relevant objects

As expected, we can observe that the effectiveness of the ranking increases with the size of the initial set. The improvement is increasing quickly as first, but as the size of the initial set contains more objects fewer relevant objects appear in the set reaching nearly 97% of all relevant objects at the size 200.

5.2 Automatic ranking

Another set of experiments was designed to test the performance of the proposed automatic ranking methods over the two datasets with different characteristics. In this case, participants of the experiments were shown several sets of results on one page and asked to mark the relevant ones. Figure 3 shows a part of one such screen.

Some of the automatic methods are further specified by parameters. In particular, the $RANK_{wordCloud}$ and $RANK_{wordCloudAndVisual}$ functions may work with a variable number of most frequent words. In the experiments, we tested two values of the parameter to understand its influence on the quality of results. The values 5 and 10 were chosen using our experience from the user-defined ranking. The following table comprises the obtained statistics.

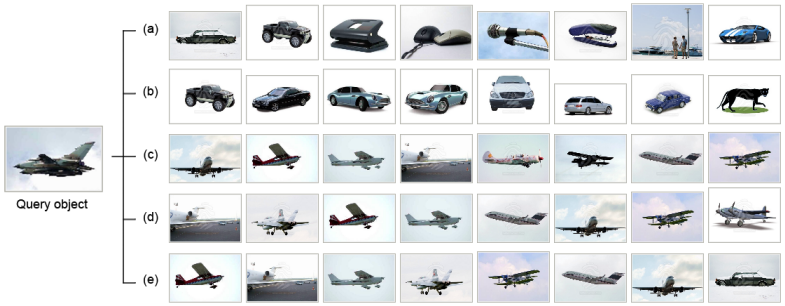

Fig. 3. Automatic ranking: a) $R_{initial}$, b) $RANK_{wordCloud}(o, R, q, 5)$, c) $RANK_{wordCloud}(o, R, q, 10)$, d) $RANK_{queryObjectKeywords}(o, R, q)$, e) $RANK_{queryObjKwAndVisual}(o, R, q)$.

	Dataset 1 result quality	Dataset 2 result quality
$R_{initial}$	36.2 %	23.5 %
$RANK_{wordCloud}(o, R, q, 5)$	33.2 %	25.4 %
$RANK_{wordCloudAndVisual}(o, R, q, 5)$	41.3 %	32.5 %
$RANK_{wordCloud}(o, R, q, 10)$	35.1 %	24.9 %
$RANK_{wordCloudAndVisual}(o, R, q, 10)$	42.0 %	33.7 %
$RANK_{queryObjectKeywords}(o, R, q)$	55.4 %	41.1 %
$RANK_{queryObjKwAndVisual}(o, R, q)$	56.8 %	43.0 %
$RANK_{adaptive}(o, R, q, 10, 10)$	56.8 %	45.4 %

Clearly, the best results for Dataset 1 are achieved when the keywords of the query object are taken into consideration. This observation conforms to the conclusion we derived from the user-defined ranking experiments. The adaptive ranking technique used the same keywords as the $RANK_{queryObjKwAndVisual}$ most of the time. As for Dataset 2, the query object keywords cleaned and enriched by the WordNet allow 10 % improvement of result quality. However, the best results were obtained by the adaptive ranking which capitalized on the cloud information combined with query object keywords.

Let us recall here that the quality of results is upper-bounded by the quality of the data and in many cases it is not possible to obtain 100 % quality. However, for any number of relevant objects in the dataset, a good retrieval system should rank them on the top positions. Therefore it is reasonable to compare search results with respect to the rank of relevant objects. A possible metric used for this purpose is the *Spearman footrule* [17], which requires the ground truth. As we do not have this, we proposed a different measure of rank quality, which we call *sparseness*. This metric is defined as the average number of irrelevant objects between two adjacent relevant ones. In an optimal search, the sparseness of a result is 0. The following table shows this measure evaluated for the initial result and the best of our ranking methods.

	Dataset 1 result sparseness	Dataset 2 result sparseness
$R_{initial}$	1.43	1.29
$RANK_{adaptive}(o, R, q, 10, 10, 10)$	0.63	0.67

5.3 Processing time

As one of our objectives is effective and efficient processing of large datasets, we also need to discuss the relationships between obtained quality and computation costs. The initial searching exploits a scalable and efficient metric search infrastructure (see Section 3 for more details) which provides retrieval with nearly constant costs. The average response time of the initial search in this implementation is 500 ms. The ranking phase costs depend on the number of processed objects. The average time needed for post-processing of a dataset with 200 objects is about 30 ms. Let us recall that the post-processing provides results of a quality comparable to the results of the multi-object query, which guarantees precise results (see Section 5.1). However, the costs of a precise evaluation of the multi-object query is much higher, ranging from seconds to tens of seconds.

6 Conclusion

In this paper, we have focused on improving the quality of results retrieved by content-based search engines via ranking. In our scenario, first an initial result set is retrieved using a standard search engine. Then, a ranking function is applied on the results to push the more relevant objects to the top of the rank list using an orthogonal similarity measure. This approach has several benefits – it can be applied to any search engine, there are no restrictions on the ranking function, and it allows to combine orthogonal views on the returned objects without computationally expensive combination techniques.

In particular, we have retrieved images by visual content and then ranked the result using text annotations. We have compared 7 different automatic ranking methods that worked on the images' keywords and 2 user-defined ranking methods where the feedback from the user was gathered. Since the similarity of images is subjective, we have measured the quality improvements by several user surveys with about 40 users of different age, sex and computer skills. Our experiments show that the ranking improved the satisfaction of users significantly – it has nearly doubled the quality of the results. We have also shown that even though the query result set still contains some irrelevant objects, the most relevant ones were pushed to the top.

The performance of the ranking methods depends heavily on the relevance of data objects in the initial result set. In the experiments, we have verified our assumption that there is a significant amount of relevant objects in the result of a general content-based search that are scattered among other objects and thus do not appear on the first result page. When several hundred top-ranking objects are submitted to the ranking method, the final result is comparable to the result of a much more expensive query processing over the whole dataset.

Acknowledgments

This work has been supported by the national research projects GACR 201/08/P507, GACR 201/09/0683 and by Brno PhD Talent Financial Aid. The hardware infrastructure was provided by the METACentrum under the research intent MSM6383917201.

References

1. Batko, M., Kohoutkova, P., Novak, D.: CoPhIR image collection under the microscope. Proceedings of SISAP 2009 pp. 47–54 (2009)
2. Batko, M., Novak, D., Zezula, P.: MESSIF: Metric similarity search implementation framework. In: First International DELOS Conference, Revised Selected Papers. LNCS, vol. 4877, pp. 1–10. Springer (2007)
3. Bolettieri, P., Esuli, A., Falchi, F., Lucchese, C., Perego, R., Piccioli, T., Rabitti, F.: CoPhIR: a test collection for content-based image retrieval. CoRR abs/0905.4627 (2009)
4. Jing, Y., Baluja, S.: VisualRank: Applying PageRank to large-scale image search. IEEE Trans. on Pattern Analysis and Machine Intell. 30(11), 1877–1890 (2008)
5. van Leuken, R.H., Garcia, L., Olivares, X., van Zwol, R.: Visual diversification of image search results. In: Proc. of the 18th international conference on World wide web. pp. 341–350. ACM, New York, NY, USA (2009)
6. Liu, D., Hua, X.S., Wang, M., Zhang, H.J.: Retagging social images based on visual and semantic consistency. In: Proceedings of WWW '10. pp. 1149–1150. ACM (2010)
7. Liu, D., Hua, X.S., Yang, L., Wang, M., Zhang, H.J.: Tag ranking. In: Proceedings of WWW '09. pp. 351–360. ACM (2009)
8. Liu, Y., Zhang, D., Lu, G., Ma, W.Y.: A survey of content-based image retrieval with high-level semantics. Pattern Recognition 40(1), 262–282 (2007)
9. Liu, Y., Mei, T., Hua, X.S.: Crowdreranking: exploring multiple search engines for visual search reranking. In: Proceedings of SIGIR '09. pp. 500–507. ACM (2009)
10. MPEG-7: Multimedia content description interfaces. Part 3: Visual. ISO/IEC 15938-3:2002 (2002)
11. Novak, D., Batko, M., Zezula, P.: Generic similarity search engine demonstrated by an image retrieval application. In: Proceedings of SIGIR '09. p. 840 (2009)
12. Park, G., Baek, Y., Lee, H.K.: Web image retrieval using majority-based ranking approach. Multimedia Tools and Applications 31(2), 195–219 (2006)
13. Skopal, T., Dohnal, V., Batko, M., Zezula, P.: Distinct nearest neighbors queries for similarity search in very large multimedia databases. In: Proceedings of WIDM '09. pp. 11–14. ACM, New York, NY, USA (2009)
14. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. on Pattern Analysis and Machine Intell. 22(12), 1349–1380 (2000)
15. Tian, X., Tao, D., Hua, X.S., Wu, X.: Active reranking for web image search. Trans. Img. Proc. 19(3), 805–820 (2010)
16. Wang, L., Yang, L., Tian, X.: Query aware visual similarity propagation for image search reranking. In: Proceedings of MM '09. pp. 725–728. ACM, New York, NY, USA (2009)
17. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search: The Metric Space Approach, Advances in Database Systems, vol. 32. Springer-Verlag (2006)
18. Zhou, X.S., Huang, T.S.: Relevance feedback in image retrieval: A comprehensive review. Multimedia Systems 8(6), 536–544 (2003)