Dynamic Two-Stage Image Retrieval from Large Multimodal Databases

Avi Arampatzis, Konstantinos Zagoris, and Savvas A. Chatzichristofis

Department of Electrical and Computer Engineering,
Democritus University of Thrace, Xanthi 67100, Greece
{avi,kzagoris,schatzic}@ee.duth.gr

Abstract. Content-based image retrieval (CBIR) with global features is notoriously noisy, especially for image queries with low percentages of relevant images in a collection. Moreover, CBIR typically ranks the whole collection, which is inefficient for large databases. We experiment with a method for image retrieval from multimodal databases, which improves both the effectiveness and efficiency of traditional CBIR by exploring secondary modalities. We perform retrieval in a two-stage fashion: first rank by a secondary modality, and then perform CBIR only on the top-K items. Thus, effectiveness is improved by performing CBIR on a 'better' subset. Using a relatively 'cheap' first stage, efficiency is also improved via the fewer CBIR operations performed. Our main novelty is that K is dynamic, i.e. estimated per query to optimize a predefined effectiveness measure. We show that such dynamic two-stage setups can be significantly more effective and robust than similar setups with static thresholds previously proposed.

1 Introduction

In content-based image retrieval (CBIR), images are represented by global or local features. Global features are capable of generalizing an entire image with a single vector, describing color, texture, or shape. Local features are computed at multiple points on an image and are capable of recognizing objects.

CBIR with global features is notoriously noisy for image queries of low *generality*, i.e. the fraction of relevant images in a collection. In contrast to text retrieval where documents matching no query keyword are not retrieved, CBIR methods typically rank the whole collection via some distance measure. For example, a query image of a red tomato on white background would retrieve a red pie-chart on white paper. If the query image happens to have a low generality, early rank positions may be dominated by spurious results such as the pie-chart, which may even be ranked before tomato images on non-white backgrounds. Figures 2a-b demonstrate this particular problem.

Local-feature approaches provide a slightly better retrieval effectiveness than global features [1]. They represent images with multiple points in a feature space in contrast to single-point global feature representations. While local approaches provide more robust information, they are more expensive computationally due to the high dimensionality of their feature spaces and usually need nearest neighbors approximation to perform points-matching [18]. High-dimensional indexing still remains a challenging problem in the database field. Thus, global features are more popular in CBIR systems as they

P. Clough et al. (Eds.): ECIR 2011, LNCS 6611, pp. 326–337, 2011.

are easier to handle and still provide basic retrieval mechanisms. In any case, CBIR with either local or global features does not scale up well to large databases efficiency-wise. In small databases, a simple sequential scan may be acceptable, however, scaling up to millions or billion images efficient indexing algorithms are imperative [15].

Nowadays, information collections are not only large, but they may also be *multimodal*. Take as an example Wikipedia, where a single topic may be covered in several languages and include non-textual media such as image, sound, and video. Moreover, non-textual media may be annotated in several languages in a variety of metadata fields such as object caption, description, comment, and filename. In an image retrieval system where users are assumed to target visual similarity, all modalities beyond image can be considered as secondary; nevertheless, they can still provide useful information for improving image retrieval.

In this paper, we experiment with a method for image retrieval from large multimodal databases, which targets to improve both the effectiveness and efficiency of traditional CBIR by exploring information from secondary modalities. In the setup considered, an information need is expressed by a query in the primary modality (i.e. an image example) accompanied by a query in a secondary modality (e.g. text). The core idea for improving effectiveness is to raise query generality before performing CBIR, by reducing collection size via filtering methods. In this respect, we perform retrieval in a two-stage fashion: first use the secondary modality to rank the collection and then perform CBIR only on the top-K items. Using a 'cheaper' secondary modality, this improves also efficiency by cutting down on costly CBIR operations.

Best results re-ranking by visual content has been seen before, but mostly in different setups than the one we consider or for different purposes, e.g. result clustering [4] or diversity [12]. Others used external information, e.g. an external set of diversified images [18] (also, they did not use image queries), web images to depict a topic [17], or training data [5]. All these approaches, as well as [16], employed a static predefined K for all queries, except [18] who re-ranked the top-30% of retrieved items. They all used global features for images. Effectiveness results have been mixed; it worked for some, it did not for others, while some did not provide a comparative evaluation or system-study. Later, we will review the aforementioned literature in more detail.

In view of the related literature, our main contributions are the following. Firstly, our threshold is calculated *dynamically* per query to optimize a predefined effectiveness measure, without using external information or training data; this is also our biggest novelty. We show that the choice between static or dynamic thresholding can make the difference between failure and success of two-stage setups. Secondly, we provide an extensive evaluation in relation to thresholding types and levels, showing that dynamic thresholding is not only more effective but also more robust than static. Thirdly, we investigate the influence of different effectiveness levels of the second visual stage on the whole two-stage procedure. Fourthly, we provide a comprehensive review of related literature and discuss the conditions under which such setups can be applied effectively. In summary, with a simpler two-stage setup than most previously proposed in the literature, we achieve significant improvements over retrieval with text-only, several image-only, and two-stage with static thresholding setups.

The rest of the paper is organized as follows. In Section 2 we discuss the assumptions, hypotheses, and requirements behind two-stage image retrieval from multimodal databases. In Section 3 we perform an experiment on a standardized multimodal snapshot of Wikipedia. In Section 4 we review related work. Conclusions and directions for further research are summarized in Section 5.

2 Two-Stage Image Retrieval from Multimodal Databases

Mutlimodal databases consist of multiple descriptions or media for each retrievable item; in the setup we consider these are image and annotations. On the one hand, textual descriptions are key to retrieve relevant results for a query but at the same time provide little information about the image content [12]. On the other hand, the visual content of images contains large amounts of information which can hardly be described by words.

Traditionally, the method that has been followed in order to deal effectively with multimodal databases is to search the modalities separately and fuse their results, e.g. with a linear combination of the retrieval scores of all modalities per item. While fusion has been proved robust, it has a few issues: a) appropriate weighing of modalities is not a trivial problem and may require training data, b) total search time is the sum of the times taken for searching the participating modalities, and most importantly, c) it is not a theoretically sound method if results are assessed by visual similarity only; the influence of textual scores may worsen the visual quality of end-results. The latter issue points to that there is a *primary modality*, i.e. the one targeted and assessed by users.

An approach that may tackle the issues of fusion would be to search in a two-stage fashion: first rank with a secondary modality, draw a rank-threshold, and then re-rank only the top items with the primary modality. The assumption on which such a two-stage setup is based on is the existence of a primary modality, and the success would largely depend on the *relative effectiveness* of the two modalities involved. For example, if text retrieval always performs better than CBIR (irrespective of query generality), then CBIR is redundant. If it is the other way around, only CBIR will be sufficient. Thus, the hypothesis is that CBIR can do better than text retrieval in small sets or sets of high query generality.

In order to reduce collection size raising query generality, a ranking can be thresholded at an arbitrary rank or item score. This improves the efficiency by cutting down on costly CBIR operations, but it may not improve too much the result quality: a too tight threshold would produce similar results to a text-only search making CBIR redundant, while a too loose threshold would produce results haunted by the red-tomato/red-pie-chart effect mentioned in the Introduction. Three factors determine what the right threshold is: 1) the number of relevant items in the collection, 2) the quality of the ranking, and 3) the measure that the threshold targets to optimize [20]. The first two factors are query-dependent, thus thresholds should be selected *dynamically* per query, not statically as most previously proposed methods in the literature (reviewed in Section 4).

The approach of [18], who re-rank the top-30% retrieved items which can be considered dynamic, does not take into account the three aforementioned factors. While the number of retrieved results might be argued correlated to the number of relevant items (thus, seemingly taking into account the first factor), this correlation can be very weak at times, e.g. consider a high frequency query word (almost a stop-word) which

would retrieve large parts of the collection. Further, such percentage thresholding seems remotely-connected to factors (2) and (3). Consequently, we will resort to the approach of [2] which, based on the distribution of item scores, is capable of estimating (1), as well as mapping scores to probabilities of relevance. Having the latter, (2) can be determined, and any measure defined in (3) can be optimized in a straightforward way. More on the method can be found in the last-cited study.

Targeting to enhance query generality, the most appropriate measure to optimize would be precision. However, since the *smoothed* precision estimated by the method of [2] monotonically declines with rank, it makes sense to set a precision threshold. The choice of precision threshold is dependent on the effectiveness of the CBIR stage: it can be seen as guaranteeing the minimum generality required by the CBIR method at hand for achieving good effectiveness. Not knowing the relation between CBIR effectiveness and minimum required generality, we will try a series of thresholds on precision, as well as, to optimize other cost-gain measures. Thus, while it may seem that we exchange the initial problem of where to set a static threshold with where to threshold precision or which measure to optimize, it will turn out that the latter problem is less sensitive to its available options, as we will see.

A possible drawback of the two-stage setup considered is that relevant images with empty or very noise secondary modalities would be completely missed, since they will not be retrieved by the first stage. If there are any improvements compared to single-stage text-only or image-only setups, these will first show up on early precision since only the top results are re-ranked; mean average precision or other measures may improve as a side effect. In any case, there are efficiency benefits from searching the most expensive modality only on a subset of the collection.

The requirement of such a two-stage CBIR at the user-side is that information needs are expressed by visual as well as textual descriptions. The community is already experimenting with such setups, e.g. the ImageCLEF 2010 Wikipedia Retrieval task was performed on a multimodal collection with topics made of textual and image queries at the same time [19]. Furthermore, multimodal or holistic query interfaces are showing up in experimental search engines allowing concurrent multimedia queries [21]. As a last resort, automatic image annotation methods [14,7] may be employed for generating queries for secondary modalities in traditional image retrieval systems.

3 An Experiment on Wikipedia

In this section, we report on experiments performed on a standardized multimodal snapshot of Wikipedia. It is worth noting that the collection is one of the largest benchmark image databases for today's standards. It is also highly heterogeneous, containing color natural images, graphics, grayscale images, etc., in a variety of sizes.

3.1 Datasets, Systems, and Methods

The ImageCLEF 2010 Wikipedia test collection has image as its primary medium, consisting of 237,434 items, associated with noisy and incomplete user-supplied textual annotations and the Wikipedia articles containing the images. Associated annotations exist in any combination of English, German, French, or any other unidentified

(non-marked) language. There are 70 test topics, each one consisting of a textual and a visual part: three title fields (one per language—English, German, French), and one or more example images. The topics are assessed by visual similarity to the image examples. More details on the dataset can be found in [19].

For text indexing and retrieval, we employ the Lemur Toolkit V4.11 and Indri V2.11 with the tf.idf retrieval model.[1] We use the default settings that come with these versions of the system except that we enable Krovetz stemming. We index only the English annotations, and use only the English query of the topics.

We index the images with two descriptors that capture global image features: the Joint Composite Descriptor (JCD) and the Spatial Color Distribution (SpCD). The JCD is developed for color natural images and combines color and texture information [8]. In several benchmarking databases, JCD has been found more effective than MPEG-7 descriptors [8]. The SpCD combines color and its spatial distribution; it is considered more suitable for colored graphics since they consist of a relatively small number of colors and less texture regions than color natural images. It is recently introduced in [9] and found to perform better than JCD in a heterogeneous image database [10].

We evaluate on the top-1000 results with mean average precision (MAP), precision at 10 and 20, and bpref [6].

3.2 Thresholding and Re-ranking

We investigate two types of thresholding: static and dynamic. In static thresholding, the same fixed pre-selected rank threshold K is applied to all topics. We experiment with levels of K at 25, 50, 100, 250, 500, and 1000. The results that are not re-ranked by image are retained as they are ranked by text, also in dynamic thresholding.

For dynamic thresholding, we use the Score-Distributional Threshold Optimization (SDTO) as described in [2] and with the code provided by its authors. For tf.idf scores, we used the *technically truncated* model of a normal-exponential mixture. The method normalizes retrieval scores to probabilities of relevance (prels), enabling the optimization of K for any user-defined effectiveness measure. Per query, we search for the optimal K in [0,2500], where 0 or 1 results to no re-ranking. Thus, for estimation with the SDTO we truncate at the score corresponding to rank 2500 but use no truncation at high scores as tf.idf has no theoretical maximum. If there are 25 text results or less, we always re-rank by image; these are too few scores to apply the SDTO reliably. In this category fall the topics 1, 10, 23, and 46, with only 18, 16, 2, and 18 text results respectively. The biggest strength of the SDTO is that it does not require training data; more details on the method can be found in the last-mentioned study.

We experiment with the SDTO by thresholding on prel as well as on precision. Thresholding on fixed prels happens to optimize *linear utility measures* [13], with corresponding rank thresholds:

- $\max K$: $P(\mathrm{rel}|D_K) > \theta$, where D_K is the Kth ranked document. For the prel threshold θ, we try six values. Two of them are:
  - $\theta = 0.5000$: It corresponds to 1 loss per relevant non-retrieved and 1 loss per non-relevant retrieved, i.e. the Error Rate, and it is precision-recall balanced.

[1] `http://www.lemurproject.org`

- $\theta = 0.3333$: It corresponds to 2 gain per relevant retrieved and 1 loss per non-relevant retrieved, i.e. the T9U measure used in the TREC 2000 Filtering Track [20], and it is recall-oriented.

These prel thresholds may optimize other measures as well; for example, 0.5000 optimizes also the utility measure of 1 gain per relevant retrieved and 1 loss per non-relevant retrieved. Thus, irrespective of which measure prel thresholds optimize, we arbitrarily enrich the experimental set of levels with four more thresholds: 0.9900, 0.9500, 0.8000, and 0.1000.

Furthermore, having normalized scores to prels, we can estimate precision in any top-K set by simply adding the prels and dividing by K. The estimated precision can be seen as the generality in the sub-ranking. According to the hypothesis that the effectiveness of CBIR is positively correlated to query generality, we experiment with the following thresholding:

– $\max K$: Prec@$K > g$, where for g is the minimum generality required by the CBIR at hand for good effectiveness. Having no clue on usable g values, we arbitrarily try levels of g at 0.9900, 0.9500, 0.8000, 0.5000, 0.3333, and 0.1000.

3.3 Setting the Baseline

In initial experiments, we investigated the effectiveness of each of the stages individually, trying to tune them for best results.

In the textual stage, we employ the tf.idf model since it has been found to work well with the SDTO [3]. The SDTO method fits a binary mixture of probability distributions on the score distribution (SD). A previous study suggested that while long queries tend to lead to smoother SDs and improved fits, threshold predictions are better for short queries of high quality keywords [3]. To be on the safe side, in initial experiments we tried to increase query length by enabling pseudo relevance feedback of the top-10 documents, but all our combinations of the parameter values for the number of feedback terms and initial query weight led to significant decreases in the effectiveness of text retrieval. We attribute this to the noisy nature of the annotations. Consequently, we do not run any two-stage experiments with pseudo relevance feedback at the first textual stage.

In the visual stage, first we tried the JCD alone, as the collection seems to contain more color natural images than graphics, and used only the first example image; this represents a simple but practically realistic setup. Then, incorporating all example images, the natural combination is to assign to each collection image the maximum similarity seen from its comparisons to all example images; this can be interpreted as looking for images similar to *any* of the example images. Last, assuming that the SpCD descriptor captures orthogonal information to JCD, we added its contribution. We did not normalize the similarity values prior to combining them, as these descriptors produce comparable similarity distributions [10]. Table 1 presents the results; the index i runs over example images.

The image-only runs perform far below the text-only run. This puts in perspective the quality of the currently effective global CBIR descriptors: their effectiveness in image retrieval is much worse than the effectiveness of the traditional tf.idf text retrieval model even on sparse and noisy annotations. Since the image-only runs would have provided

Table 1. Effectiveness of different CBIR setups against tf.idf text-only retrieval

item scoring by	MAP	P@10	P@20	bpref
JCD_1	.0058	.0486	.0479	.0352
$\max_i$ JCD_i	.0072	.0614	.0614	.0387
$\max_i$ JCD_i + $\max_i$ $SpCD_i$	.0112	.0871	.0886	.0415
tf.idf (text-only)	.1293	.3614	.3314	.1806

very weak baselines, we choose as a much stronger baseline for statistical significance testing the text-only run. This makes sense also from an efficiency point of view: if using a secondary text modality for image retrieval is more effective than current CBIR methods, then there is no reason at all for using computationally costly CBIR methods.

Comparing the image-only runs to each other, we see that using more information—either from more example images or more descriptors—improves effectiveness. In order to investigate the impact of the effectiveness level of the second stage on the whole two-stage procedure, we will present two-stage results for both the best and the worst CBIR methods.

3.4 Experimental Results

Table 2 presents two-stage image retrieval results against text- and image-only retrieval. It is easy to see that the dynamic thresholding methods improve retrieval effectiveness in most of the experiments. Especially, dynamical thresholding using θ shows improvements for all values we tried. The greatest improvement ($+28\%$) is observed in P@10 for $\theta = 0.8$. The table contains lots of numbers; while there may be consistent increases or decreases in some places, in the rest of this section we focus and summarize only the statistically significant differences.

Irrespective of measure and CBIR method, the best thresholds are roughly at: 25 or 50 for K, 0.95 for g, and 0.8 for θ. The weakest thresholding method is the static K: there are very few improvements only in P@20 at tight cutoffs, but they are accompanied by a reduced MAP and bpref. Actually, static thresholds hurt MAP and/or bpref almost anywhere. Effectiveness degrades also in early precision for $K = 1000$. Dynamic thresholding is much more robust. Comparing the two CBIR methods at the second stage, the stronger method helps the dynamic methods considerably while static thresholding does not seem to receive much improvement.

Concerning the dynamic thresholding methods, the probability thresholds θ correspond to tighter *effective* rank thresholds than these of the precision thresholds g, for g and θ taking values in the range $[0.1000, 0.9900]$. As a proxy for the effective K we use the median threshold $\widetilde{K}$ across all topics. This is expected since precision declines slower than prel. Nevertheless, the fact that a wide range of prel thresholds results to a tight range of $\widetilde{K}$, reveals a sharp decline in prel below some score per query. This makes the end-effectiveness less sensitive to prel thresholds in comparison to precision thresholds, thus more robust against possibly unsuitable user-selected values. Furthermore, if we compare the dynamic methods at similar $\widetilde{K}$, e.g. $g = 0.9900$ to $\theta = 0.9500$ ($\widetilde{K} \approx 50$) and $g = 0.8000$ to $\theta = 0.5000$ ($\widetilde{K} \approx 93$), we see that prel thresholds perform slightly better. Figure 1 depicts the evaluation measures against $\widetilde{K}$ for all methods and the stronger CBIR; Figure 2 presents the top image results for a query.

Table 2. Two-stage image retrieval results. The best results per measure and thresholding type are in boldface. Significance-tested with a bootstrap test, one-tailed, at significance levels 0.05 ($^{\triangle\triangledown}$), 0.01 ($^{\blacktriangle\triangledown}$), and 0.001 ($^{\blacktriangle\blacktriangledown}$), against the text-only baseline.

threshold		$\widetilde{K}$	JCD_1				$\max_i \mathrm{JCD}_i + \max_i \mathrm{SpCD}_i$			
			MAP	P@10	P@20	bpref	MAP	P@10	P@20	bpref
text-only		—	.1293	.3614	.3314	.1806	.1293	.3614	.3314	.1806
K	25	25	**.1162**$^{\triangledown}$	**.3957**$^{-}$	.3457$^{\triangle}$	.1641$^{\blacktriangledown}$	**.1168**$^{-}$	.3943^{-}	.3436$^{\triangle}$	.1659$^{\triangledown}$
	50	50	.1144$^{\triangledown}$	.3829^{-}	**.3579**$^{\triangle}$	.1608$^{\triangledown}$	.1154^{-}	**.3986**$^{-}$	.3557^{-}	.1648$^{\triangledown}$
	100	100	.1138^{-}	.3786^{-}	.3471^{-}	.1609$^{\triangledown}$	.1133^{-}	.3900^{-}	.3486^{-}	.1623^{-}
	250	250	.1081$^{\triangledown}$	.3414^{-}	.3164^{-}	**.1644**$^{-}$	.1092$^{\triangledown}$	.3771^{-}	**.3564**$^{-}$	**.1664**$^{-}$
	500	500	.0968$^{\blacktriangledown}$	.3200^{-}	.3007^{-}	.1575$^{\triangledown}$	.0999$^{\blacktriangledown}$	.3557^{-}	.3250^{-}	.1590$^{\triangledown}$
	1000	1000	.0865$^{\blacktriangledown}$	.2871$^{\triangledown}$	.2729$^{\triangledown}$	.1493$^{\triangledown}$	.0909$^{\blacktriangledown}$	.3329^{-}	.3064^{-}	.1511$^{\triangledown}$
g	.9900	49	**.1364**$^{-}$	**.4214**$^{\triangle}$	.3550^{-}	.1902$^{\triangle}$	.1385$^{\triangle}$	.4371$^{\triangle}$	.3743$^{\triangle}$	.1921$^{\triangle}$
	.9500	68	.1352^{-}	.4171$^{\triangle}$	**.3586**$^{-}$	**.1912**$^{\triangle}$	**.1386**$^{\triangle}$	**.4500**$^{\triangle}$	.3836$^{\triangle}$	**.1932**$^{\triangle}$
	.8000	95	.1318^{-}	.4000^{-}	.3536^{-}	.1892^{-}	.1365^{-}	.4443$^{\triangle}$	**.3871**$^{\triangle}$	.1924^{-}
	.5000	151	.1196^{-}	.3814^{-}	.3393^{-}	.1808^{-}	.1226^{-}	.4043^{-}	.3550^{-}	.1813^{-}
	.3333	237	.1085$^{\blacktriangledown}$	.3500^{-}	.3000^{-}	.1707^{-}	.1121$^{\triangledown}$	.3857^{-}	.3364^{-}	.1734^{-}
	.1000	711	.0864$^{\blacktriangledown}$	.2871$^{\triangledown}$	.2621$^{\blacktriangledown}$	.1461$^{\blacktriangledown}$	.0909$^{\blacktriangledown}$	.3357^{-}	.2964^{-}	.1487$^{\blacktriangledown}$
θ	.9900	42	.1342^{-}	.4043^{-}	.3414^{-}	.1865^{-}	.1375$^{\triangle}$	.4371$^{\triangle}$	.3700$^{\triangle}$	.1897$^{\triangle}$
	.9500	51	.1371^{-}	.4214$^{\triangle}$	.3586^{-}	.1903$^{\triangle}$	.1417$^{\triangle}$	.4500$^{\triangle}$	.3864$^{\triangle}$	.1924$^{\triangle}$
	.8000	81	**.1384**$^{\triangle}$	**.4229**$^{\triangle}$	.3614^{-}	.1921$^{\triangle}$	**.1427**$^{\triangle}$	**.4629**$^{\blacktriangle}$	.3871$^{\triangle}$	**.1961**$^{\triangle}$
	.5000	91	.1367^{-}	.4057^{-}	.3571^{-}	.1919$^{\triangle}$	.1397$^{\triangle}$	.4400$^{\triangle}$	.3829$^{\triangle}$	.1937$^{\triangle}$
	.3333	109	.1375^{-}	.4129^{-}	**.3636**$^{-}$	**.1933**$^{\triangle}$	.1404$^{\triangle}$	.4500$^{\triangle}$	**.3907**$^{\triangle}$	.1949$^{\triangle}$
	.1000	130	.1314^{-}	.4100^{-}	.3629^{-}	.1866^{-}	.1370^{-}	.4371$^{\triangle}$	.3843$^{\triangle}$	.1922$^{\triangle}$
image-only		—	.0058$^{\blacktriangledown}$	.0486$^{\blacktriangledown}$	.0479$^{\blacktriangledown}$	.0352$^{\blacktriangledown}$	.0112$^{\blacktriangledown}$	.0871$^{\blacktriangledown}$	.0886$^{\blacktriangledown}$	.0415$^{\blacktriangledown}$

In summary, static thresholding improves initial precision at the cost of MAP and bpref, while dynamic thresholding on precision or prel does not have this drawback. The choice of a static or precision threshold influences greatly the effectiveness, and unsuitable choices (e.g. too loose) may lead to a degraded performance. Prel thresholds are much more robust in this respect. As expected, better CBIR at the second stage leads to overall improvements, nevertheless, the thresholding type seems more important: While the two CBIR methods we employ vary greatly in performance (the best has almost double the effectiveness of the other), static thresholding is not influenced much by this choice; we attribute this to its lack of respect for the number of relevant items and for the ranking quality. Dynamic methods benefit more from improved CBIR. Overall, prel thresholds perform best, for a wide range of values.

4 Related Work

Image re-ranking can be performed using textual, e.g. [11], or visual descriptions. Next, we will focus only on visual re-ranking. Subset re-ranking by visual content has been seen before, but mostly in different setups than the one we consider or for different purposes, e.g. result clustering or diversity. It is worth mentioning that all the previously proposed methods we review below used global image features to re-rank images.

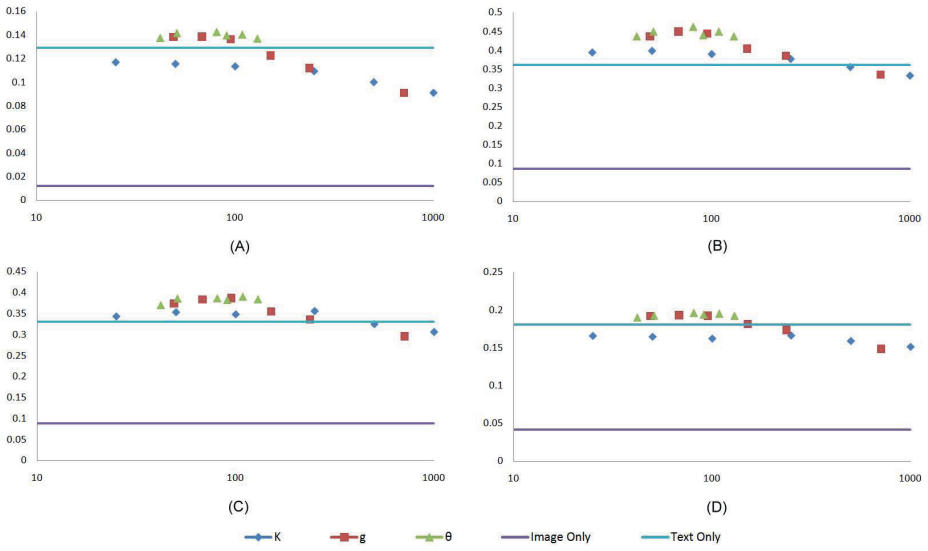

Fig. 1. Effectiveness, for the strongest CBIR stage: (A) MAP, (B) P@10, (C) P@20, (D) bpref

For example, [4] proposed an image retrieval system using keyword-based retrieval of images via their annotations, followed by clustering of the top-150 results returned by Google Images according to their visual similarity. Using the clusters, retrieved images were arranged in such a way that visually similar images are positioned close to each other. Although the method may have had a similar effect to ours, it was not evaluated against text-only or image-only baselines, and the impact of different values of K was not investigated. In [12], the authors retrieved the top-50 results by text and then clustered the images in order to obtain a diverse ranking based on cluster representatives. The clusters were evaluated against manually-clustered results, and it was found that the proposed clustering methods tend to reproduce manual clustering in the majority of cases. The approach we have taken does not target to increasing diversity.

Another similar approach was proposed in [18], where the authors state that Web image retrieval by text queries is often noisy and employ image processing techniques in order to re-rank retrieved images. The re-ranking technique was based on the visual similarity between image search results and on their dissimilarity to an external contrastive class of diversified images. The basic idea is that an image will be relevant to the query, if it is visually similar to other query results and dissimilar to the external class. To determine the visual coherence of a class, they took the top 30% of retrieved images and computed the average number of neighbors to the external class. The effects of the re-ranking were analyzed via a user-study with 22 participants. Visual re-ranking seemed to be preferred over the plain keyword-based approach by a large majority of the users. Note that they did not use an image query but only a text one; in this respect, the setup we have considered differs in that image queries are central, and we do not require external information.

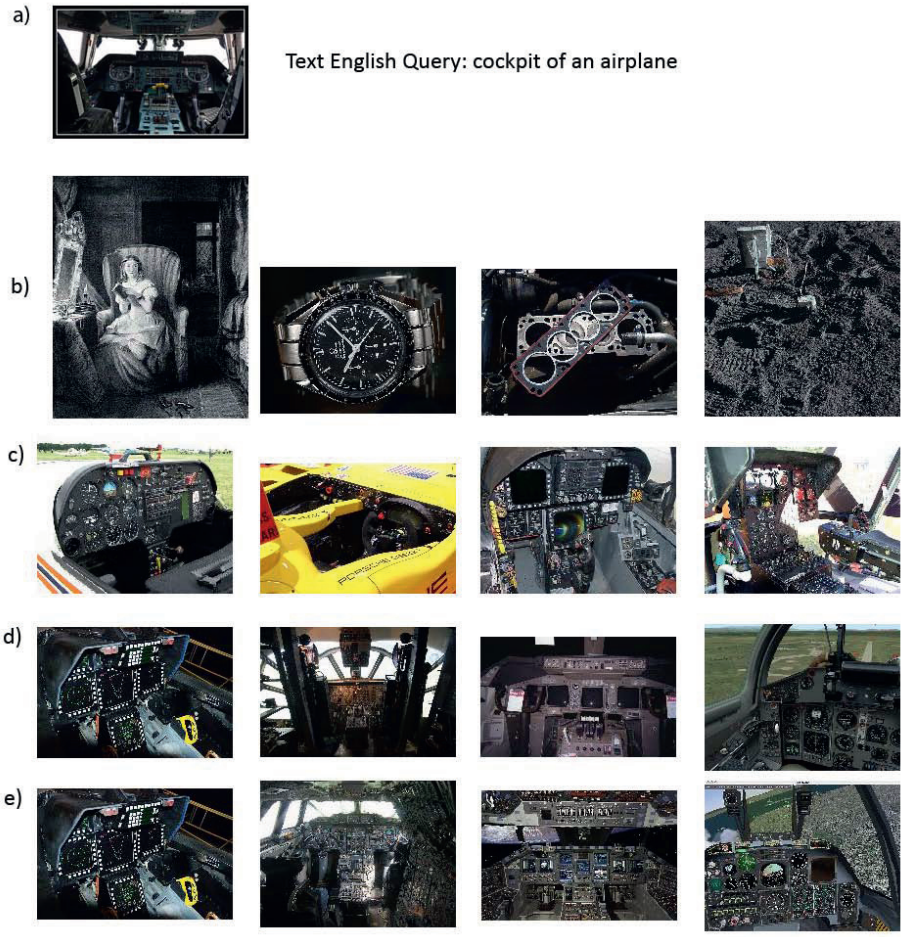

Fig. 2. Retrieval results: (a) query, (b) image-only, (c) text-only, (d) $K = 25$, (e) $\theta = 0.8$

In [17], the authors proposed also a two-stage image retrieval system with external information requirements: the first stage is text-based with automatic query expansion, whereas the second exploits the visual properties of the query to improve the results of the text search. In order to visually re-rank the top-1000 images, they employed a visual model (a set of images which depicts each topic) using Web images. To describe the visual content of the images, several methods using global or local features were employed. Experimental results demonstrated that visual re-ranking improves the retrieval performance significantly in MAP, P@10 and P@20. We have confirmed that visual re-ranking of top-ranked results improves early precision, though with a simpler setup without using external information.

Some other similar setups to the one we propose are these in [5] and [16]. In [5], the authors trained their system to perform automatic re-ranking on all results returned by text retrieval. The re-ranking method considered several aspects of both document and query (e.g. generality of the textual features, color amount from the visual features). Improved results were obtained only when the training set had been derived from the database which is searched. Our method re-ranks the results using only visual features; it does not require training and can be applied to any database. In [16], the authors re-rank the top-K results retrieved by text using visual information. The rank thresholds of 60 and 300 were tried and both resulted to a decrease in mean average precision compared to the text-only baseline, with the 300 performing worse. Our experiments have confirmed their result: static thresholds degrade MAP. They did not report early precision figures.

5 Conclusions and Directions for Further Research

We have experimented with two-stage image retrieval from a large multimodal database, by first using a text modality to rank the collection and then perform content-based image retrieval only on the top-K items. In view of previous literature, the biggest novelty of our method is that re-ranking is not applied to a preset number of top-K results, but K is calculated dynamically per query to optimize a predefined effectiveness measure. Additionally, the proposed method does not require any external information or training data. The choice between static or dynamic nature of rank-thresholds has turned out to make the difference between failure and success of the two-stage setup.

We have found that two-stage retrieval with dynamic thresholding is more effective and robust than static thresholding, practically insensitive to a wide range of reasonable choices for the measure under optimization, and beats significantly the text-only and several image-only baselines. A two-stage approach, irrespective of thresholding type, has also an obvious efficiency benefit: it cuts down greatly on expensive image operations. Although we have not measured running times, only the 0.02–0.05% of the items (on average) had to be scored at the expensive image stage for effective retrieval from the collection at hand. While for the dynamic method there is some overhead for estimating thresholds, this offsets only a small part of the efficiency gains.

There are a couple of interesting directions to pursue in the future. First, the idea can be generalized to *multi-stage* retrieval for multimodal databases, where rankings for the modalities are successively being thresholded and re-ranked according to a modality hierarchy. Second, although in Section 2 we merely argued on the unsuitability of fusion under the assumptions of the setup we considered, a future plan is to compare the effectiveness of two-stage against fusion. Irrespective of the outcome, fusion does not have the efficiency benefits of two-stage retrieval.

Acknowledgments

We thank Jaap Kamps for providing the code for the statistical significance testing.

References

1. Aly, M., Welinder, P., Munich, M.E., Perona, P.: Automatic discovery of image families: global vs. local features. In: ICIP, pp. 777–780. IEEE, Los Alamitos (2009)
2. Arampatzis, A., Kamps, J., Robertson, S.: Where to stop reading a ranked list: threshold optimization using truncated score distributions. In: SIGIR, pp. 524–531. ACM, New York (2009)
3. Arampatzis, A., Robertson, S., Kamps, J.: Score distributions in information retrieval. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 139–151. Springer, Heidelberg (2009)
4. Barthel, K.U.: Improved image retrieval using automatic image sorting and semi-automatic generation of image semantics. In: International Workshop on Image Analysis for Multimedia Interactive Services, pp. 227–230 (2008)
5. Berber, T., Alpkocak, A.: DEU at ImageCLEFMed 2009: Evaluating re-ranking and integrated retrieval systems. In: CLEF Working Notes (2009)
6. Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: SIGIR, pp. 25–32. ACM, New York (2004)
7. Chang, E., Goh, K., Sychay, G., Wu, G.: CBSA: content-based soft annotation for multimodal image retrieval using bayes point machines. IEEE Transactions on Circuits and Systems for Video Technology 13(1), 26–38 (2003)
8. Chatzichristofis, S.A., Boutalis, Y.S., Lux, M.: Selection of the proper compact composite descriptor for improving content-based image retrieval. In: SPPRA, pp. 134–140 (2009)
9. Chatzichristofis, S.A., Boutalis, Y.S., Lux, M.: SpCD—spatial color distribution descriptor. A fuzzy rule based compact composite descriptor appropriate for hand drawn color sketches retrieval. In: ICAART, pp. 58–63 (2010)
10. Chatzichristofis, S.A., Arampatzis, A.: Late fusion of compact composite descriptors for retrieval from heterogeneous image databases. In: SIGIR, pp. 825–826. ACM, New York (2010)
11. Kilinc, D., Alpkocak, A.: Deu at imageclef 2009 wikipediamm task: Experiments with expansion and reranking approaches. In: Working Notes of CLEF (2009)
12. van Leuken, R.H., Pueyo, L.G., Olivares, X., van Zwol, R.: Visual diversification of image search results. In: WWW, pp. 341–350. ACM, New York (2009)
13. Lewis, D.D.: Evaluating and optimizing autonomous text classification systems. In: SIGIR, pp. 246–254. ACM Press, New York (1995)
14. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 985–1002 (2008)
15. Li, X., Chen, L., Zhang, L., Lin, F., Ma, W.Y.: Image annotation by large-scale content-based image retrieval. In: ACM Multimedia, pp. 607–610. ACM, New York (2006)
16. Maillot, N., Chevallet, J.P., Lim, J.H.: Inter-media pseudo-relevance feedback application to imageclef 2006 photo retrieval. In: CLEF Working Notes (2006)
17. Myoupo, D., Popescu, A., Le Borgne, H., Moellic, P.: Multimodal image retrieval over a large database. In: Peters, C., Caputo, B., Gonzalo, J., Jones, G.J.F., Kalpathy-Cramer, J., Müller, H., Tsikrika, T. (eds.) CLEF 2009. LNCS, vol. 6242, pp. 177–184. Springer, Heidelberg (2010)
18. Popescu, A., Moëllic, P.A., Kanellos, I., Landais, R.: Lightweight web image reranking. In: ACM Multimedia, pp. 657–660. ACM, New York (2009)
19. Popescu, A., Tsikrika, T., Kludas, J.: Overview of the wikipedia retrieval task at imageclef 2010. In: CLEF (Notebook Papers/LABs/Workshops) (2010)
20. Robertson, S.E., Hull, D.A.: The TREC-9 filtering track final report. In: TREC (2000)
21. Zagoris, K., Arampatzis, A., Chatzichristofis, S.A.: www.mmretrieval.net: a multimodal search engine. In: SISAP, pp. 117–118. ACM, New York (2010)