QbS – Searching for Known Images using User-Drawn Sketches

Michael Springmann  Dietmar Kopp  Heiko Schuldt
Department of Computer Science
University of Basel, Switzerland
{michael.springmann|heiko.schuldt}@unibas.ch

ABSTRACT
With the increasingly growing size of digital image collections, known image search is gaining more and more importance. Especially when the objects in such collections do not possess appropriate metadata (e.g., tags, annotations), content-based image retrieval (CBIR) is a promising approach. However, the application of CBIR to known item search usually suffers from the unavailability of query images that are good enough to express the user’s information need.

In order to improve this situation, we propose the QbS system which provides an approach to content-based search in large image collections based on user-drawn sketches. By exploiting novel devices for human-computer interaction like interactive paper, tablet PCs, or graphic tablets, users are able to draw a sketch that reflects their information need and start a content-based search using this sketch. The QbS system provides query support and offers several invariances that allow the user-generated sketch to slightly deviate from the searched image in terms of rotation, translation, relative size, and/or unknown objects in the background. To illustrate the benefits of the approach, we show search results from the evaluation of QbS on the basis of the MIRFLICKR collection with 25,000 objects.

Categories and Subject Descriptors
H.3.3 [Information Storage And Retrieval]: Information Search and Retrieval—query formulation, retrieval models; H.5.2 [Information Interfaces and Presentation]: User Interfaces—input devices and strategies

General Terms
Algorithms, Human Factors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 2010 ACM 978-1-60558-815-5/10/03 ...$10.00.

Keywords
Content-based image retrieval, Query by sketch, Known item search.

1. INTRODUCTION
Over the last years, digital cameras and flash memory cards with capacities of several gigabytes have become increasingly popular. This has led to a very rapid growth of (personal) photo collections, both in terms of sheer size and numbers of objects. As a consequence, searching in these large collections has become an important challenge. Browsing through a collection, which used to be the most common approach, is no longer a satisfactory solution as it does not scale with the increasing collection sizes.

Usually, searching for images requires a high degree of manual activity from a user: either she has to properly organize her digital photos on the hard disk (using meaningful folder structures and/or file names that characterize the images’ content) or she has to annotate each photo with a set of metadata tags that precisely describe the image content. However, both tasks are very time consuming. In addition, these tasks are extremely difficult to achieve as future queries need to be anticipated already at the time a photo is tagged in order to make sure that it will actually be found.

By “crowdsourcing” the load of tagging images, e.g., using the ESP Game [11] or in the context of sharing photos in social networks like flickr1 or Facebook2, the workload can be distributed. However, the tags associated in the first case might not be specific enough to handle very precise queries whereas the tags associated with images in social networks are typically very subjective, do not follow an established terminology and are thus not always usable for searches [2]. Such tags can be sufficient for searching particular concepts in an image like a person you know by name, particular mountains, famous buildings, or any sunset — but it reaches its limits when a name is not known or also if the image composition is important, e.g., when a certain object should be placed in the upper-left corner or lower center. Moreover, photos shared with friends in a social network are usually small fractions of the overall collection a user has in her personal information space. Thus, a large number of photos is left without annotations anyways and looking for one particular image will very frequently terminate without success.

In such situations, letting the user draw a simple sketch

1http://www.flickr.com
2http://www.facebook.com
The QbS system addresses these two issues by i.) taking into account novel input devices very well suited for the creation of sketches \[10\], in particular tablet PCs, graphic tablets, and interactive paper \[8\]; ii.) tailoring CBIR algorithms to user-drawn sketches as query images. The latter uses a distance measure to calculate the similarity between sketches and the query sketch are extracted within the search process.

2. PROTOTYPE IMPLEMENTATION

2.1 Used Features and Similarity Measures

Content-based image retrieval usually does not directly compare the content of two images, but first transforms them into a feature space in which comparison takes place. Features of the all images in the user’s collection are extracted as soon as the images are added to the collection and are stored together with the images. The features of the query sketch are extracted within the search process.

2.1.1 Angular Radial Partitioning

Angular Radial Partitioning (ARP) \[3\] generates ‘edge maps’ from images depending on the choice of a parameter $\beta$ which specifies the edge threshold. Edge maps are then split into the number of angular and radial partitions as illustrated in Fig. 1. The feature vector finally consists of the number of detected edge pixels in each partition.

Since sketches are expected to contain only edges, there is no need to convert them into edge maps before they can be analyzed. The features of the sketch and the images from the collection are then compared based on the Manhattan distance. Notice, that this already provides robustness against minor modifications of the image w.r.t. the sketch as long as not too many edge pixels are missing or shifted from one partition to another. By scaling images to a common size, the approach becomes invariant to changes in scale of the image as a whole.

2.1.2 Optional Invariances

Additional robustness or invariances might be beneficial depending on the user’s information need, but unwanted in other situations since any invariance comes at the cost of losing expressiveness in the query. Hence, the user is able to specify for each search the invariances which should apply.

For instance, if the user is unsure if the sketch she drew matches size and aspect ratio well enough, she can select the scale checkbox below the sketch in Fig. 2 to compare the sketch also with subregions extracted at different scales from the images in the user’s collection. Distances are computed to all these versions, but only the best matching version is used in ranking an image from the collection.

Rotation invariance is achieved by performing a 1D discrete Fourier transformation \[3\] on the features in the collection as well as for the features of the sketch. This allows to search for images of arbitrary orientation compared to the sketch without the need of comparing several versions.

If the position of the sketched object inside the image is not known, translation invariance is achieved by searching subregions of the images and a bounding box of the image scaled to full size. Selection of the best version can be handled exactly as for scale invariance.

Finally, if images do not only contain the sketched objects alone or an almost monochromatic background like a blue sky or a white table, or if the user does not remember the background of the searched image in the collection, then the user can simply leave such areas empty and set the option to ignore the background. This will skip, during evaluation, all elements of the query vector that are zero, thus allow

\[3\] Image "Let your soul fly" by Dakotilla – part of the MIRFLICKR-25000 benchmark collection.
arbitrary content in images in areas left blank by the user in the sketch.

For rather diverse collections, it might be hard for the user to estimate which edges in an image will still be detected by the system and which ones are not strong enough or blurred too much to be found for a particular value of $\beta$. In our system, the user can draw the sketch independently of these details and enable an option that allows comparing the sketch not only with the features of the images taken with one single value for $\beta$, but an entire range from which the system selects the best-matching version of each image.

In order to give more control over the level of details, several combinations of angular and radial partitions are extracted. The user can specify for the search to either use few partitions which makes the approach more robust or she selects many partitions which better captures the details in her sketch. In comparison to previous attempts to querying image collections with user drawn sketches [6, 5, 1, 4], these options give the user a very fine-grained control over the comparison between her sketch and the images in the collection. Therefore, she can easily adapt the search to the parts of the image she remembers and also take into account her assessment of the quality of her sketch (how well she was able to draw the sketch).

2.2 Storage and Query Execution

Search is performed by applying a $k$-nearest neighbor sequential scan over the file containing the features. To reduce computational complexity, the early termination strategy proposed in [9] is applied. Thus, the time spent in reading the features from disk can easily exceed the time needed to compute similarity. To also reduce disk I/O, the files are compressed and features are cached between subsequent searches.

Since the features are rather compact — for one image with a specific value of $\beta$ only the number of angular partitions $\times$ the number of radial partitions individual values need to be stored — caching the features in memory is not a problem even for thousands of images on current hardware. To handle the possible combinations of features, a simple Least Recently Used (LRU) strategy is used to keep subsequent searches fast, allowing interactive modifications of the sketch before re-submitting a query. If query execution time should still be crucial, a high-dimensional index structure such as, for instance, the VA-File [12] can be used after modifications of distance evaluation to make sure that invariances introduced in 2.1.2 are supported by the index.

2.3 Result Presentation

The query results are presented to the user in the right hand area of the QbS window. If the user left clicks on a thumbnail, the original image is presented in the same size as the drawing area as shown in Fig. 3(b). After right click on the result, the edge map of the image is presented as depicted in Fig. 3(c) such that the user can easily compare the sketch with the retrieved results and modify the query parameters, e.g., refine the sketch or enable/disable options for invariances.

Because of fast query execution, loading images from disk can become the bottleneck in user experience. To be able to present results quickly, previously retrieved thumbnail images are held in an LRU cache (similar to the features), such that only slightly modified searches will display results almost instantly while newly retrieved items are loaded in a background thread from disk.

Notice that for finishing the search task successfully, it is beneficial but not necessary that the known item is returned as the first or one of the top-ranked results. In situations where no appropriate metadata for querying is known to the user, the only alternative would be browsing the entire collection — therefore the most important aspect must be that the searched item is among the query results and that the user can review even long result lists fast and easily.

3. EXPERIMENTS

For the search results presented in this paper we use the MIRFLICKR-25000 Image Collection\(^4\) as a benchmark collection.

\(^4\)http://press.liacs.nl/mirflickr/
lection with images selected based on interestingness by general public [7].

Execution of queries takes between 15 ms and 1 s on a regular tablet PC with Intel Core 2 Duo 1.86 GHz and 2 GB RAM if features are cached, and depends mainly on selected combination of invariances; with common settings usually returning results in less than 200 ms. If features are not found in cache, loading from hard disk takes approx. 500 ms.

We intentionally ignored all metadata like tags associated with the images for the purpose of showing only CBIR functionality, which otherwise can get hidden behind the quality of the metadata. Combining QbS with traditional metadata-oriented image search is straightforward, in particular since the features do not require any clustering. Therefore, QbS can seamlessly be applied to subsets of the entire collection (e.g., on the result set of a keyword query) without degrading the retrieval time.

4. CONCLUSION

CBIR provides a powerful search paradigm for large image collections but is usually not applicable when no adequate query image is available. Query by sketching allows to overcome this problem. The QbS system exploits novel input devices for the creation of sketches and is very robust as it provides the most important invariances in comparing sketches with images from the user’s image collection.

5. REFERENCES


Figure 3: Search for a flying jet (a-c) and apple (d)