Query Model-Based Content-Based Image Retrieval:
Similarity Definition, Application and Automation - ABSTRACT

Horst Eidenberger
Vienna University of Technology, Institute of Software Technology, Austria
-11/188, A-1040 Vienna, Austria
Phone: +43 1 4035158 14
eMail: hme@bibvb.ac.at

Advisor: Prof. Dr. Christian Breiteneder, Vienna University of Technology, Austria

1. Introduction

This abstract describes my doctoral thesis in the field of multimedia / content-based image retrieval (CBIR). CBIR aims at searching image libraries for specific image features (e.g., colors, textures, shapes). Querying is performed by comparing feature vectors (e.g., color histograms) of a search image with the feature vectors of all images in the database. To start a query the user selects search image, features and weights in order to indicate the importance of the selected features. This method is called the computer centric approach (CCA). Currently, the computer centric approach in CBIR suffers from several disadvantages:

- Bad results due to the semantic gap and the subjectivity of human perception: The first point stands for the difference between the high-level CBIR concepts usually presented to users and the low-level features actually employed. The latter addresses the fact that different persons (recipients) or the same person in different situations may judge visual content differently.

- Bad querying performance: Using (computational often very complex) distance functions for the comparison of feature vectors leads to bad, sometimes unacceptable response times.

- Complex interfaces: CBIR is very different from traditional text retrieval. CBIR interfaces tend to be complex and difficult to use. Additionally, average users are overtaxed by the requirement to select features and weights for a specific querying process.

Our work aims at reducing these problems. We developed a system with several simple, but robust features and use them in groups. These groups are called query models and will be discussed in Section 2. We further developed algorithms for the automatic generation of queries from search images and use iterative refinement to improve results. In order to improve the querying performance we developed an algorithm for the performance-optimized ordering of query features.

The remainder of this abstract is organized as follows: Section 2 describes the concepts of our approach to CBIR. Section 3 is dedicated to the implemented algorithms. Finally, in Section 4 we describe the results achieved by the various methods.

2. Our approach to CBIR

The basic idea of our approach is to use several simple features in combination instead of just a few, but more complex ones (e.g., color histogram, etc.) [3]. The features selected for a specific query are grouped into query models. A query model [1] consists of a set of layers. Each layer identifies a feature, its distance function, weight and threshold value. The threshold describes the maximum distance between an image and the search image. During our work we found that controlling the result set by thresholds instead of a number identifying the size of the result set increases the quality of results significantly ([1]).

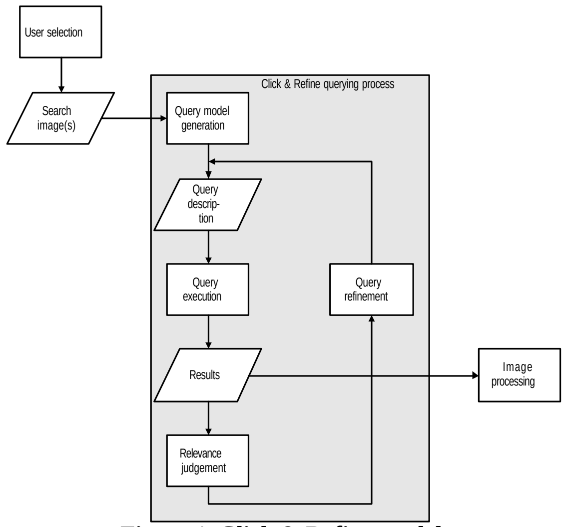

Figure 1. Click & Refine model.

Features utilized in a query model are dynamically chosen and usually cover only a subset of the features available. The query process is based on query models and follows our click & refine model (see Figure 1). In this model the user has to choose one or more search images to initiate a query. From the information in the search image(s) and expert knowledge the system derives a first query model and retrieves a first result. Then, in an iterative refinement process the result can be improved by relevance feedback provided by the user. The click & refine model has two major advantages: First, employing iterative refinement helps reducing the semantic gap. Second, the interface for our system becomes much easier than in traditional CBIR systems. All the user has to do is selecting examples for her/his query and rating results by her/his relevance judgement.

3. Implementation

We implemented our models in a CBIR system for a specific application domain, the retrieval of coats of arms. We implemented altogether 19 features, including a color histogram, symmetry features, etc. (see [1], [3], [4]). All distance functions (including Euclidean distance, city block distance, etc.) are standardized on the interval [0, 1]. For these features four algorithms were implemented to overcome the disadvantages of the computer centered approach:

1. A *weighting algorithm* for the automatic derivation of weights for all features in a query model [4]—For this purpose we clustered the global feature vectors (all feature vectors merged) of images in the test database. The weight of a feature for a specific search image is defined as the contribution of this feature to build the cluster of the search image.

2. An *ordering algorithm* for the performance-optimized ordering of features [5]—The ordering algorithm maintains a prognosis database and sorts query models before their execution according to their predicted number of returned images and the performance of the distance functions. This is a tricky task but allows an enormous increase of performance.

3. Two *generation algorithms* for the automatic generation of query models out of a search image or out of a group of search images and expert knowledge [2] to make the application of the click & refine model possible. The task of these two algorithms is to select features and suitable thresholds for one or more query models. Weighting and ordering is done by the algorithms above.

   The algorithm for the generation of a query model from one search image offers three methods for feature selection and three for threshold definition, which can be arbitrarily combined. Features may be selected by weight (importance), by properties or by a combination. Thresholds can be derived from the weight of a feature, the probable number of returned images or any linear combination of both methods [2].

   The second algorithm for the generation of query models (from a group of search images) performs an even more difficult task. By selecting a group of images the user may define similarity subjectively and asks to retrieve all images that are similar to this search group. Our algorithm solves this task by clustering the presented group of images and calculating a query model for each cluster of search images. If a cluster consists of only one image the algorithm described above is applied. Otherwise, the centroid of the cluster is utilized as the search image and the feature and thresholds are derived from mean and variance of the distances between every element of the cluster and all other elements [2].

These algorithms are implemented in our test environment as C-libraries. The CBIR system is based on IBM's QBIC (version 2; [6]). The features are implemented as QBIC feature classes. The QBIC query engine was replaced by our own query engine, which can handle query models and supports the click & refine model. The whole system is installed on a Linux-PC [3]. The test environment consists of 888 images of coats of arms.

4. Results

We tested all components of our system except the iterative refinement process. The components of the generation algorithms were evaluated by recall and precision and the ordering algorithm was tested by its performance. The best algorithm for query model derivation from one image has a precision of 68% and a recall of 94%. A human expert in our tests could increase this result to a precision of 91% with a recall of at least 83%. The algorithm for query generation from a group of search images achieves a recall of about 80% with a precision of at least 60%. During the tests we learned that results do not improve with the number of search images but depend rather on the search images chosen [2]. We conclude that the problem of generating queries out of groups of images is difficult to solve, even for coats of arms. However, we think that our algorithm provides a good entry point into an iterative retrieval process.

After these tests we investigated how the quality of our algorithms changes when a certain percentage (n%) of new images is added to the test database. We found out that our algorithms do not depend too much on which images were used to calculate the image clusters and the algorithm parameters.

The weighting algorithm was tested on its own. The testing process and the results cannot be displayed here due to lack of space. The derived weights lead to an ordering of the result set that was at least judged suitable in more than 80% of the cases [4]. The ordering algorithm was tested with more than 1 000 generated query models. We found that our ordering algorithm reduces the average response time in our CBIR system from 190.7 ms by 66% to 64.6 ms [5].

5. ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Dr. Christian Breiteneder (Vienna University of Technology, Austria) for his conscientious and patient guidance.

6. References

[1] Breiteneder, C., and Eidenberger, H., A Retrieval System for Coats of Arms, *Proc. of ISIMADE'99*, 1999.

[2] Breiteneder, C., and Eidenberger, H., Automatic Query Generation for Content-based Image Retrieval, *Proc. of IEEE Multimedia Conference*, 2000.

[3] Breiteneder, C., and Eidenberger, H., Content-based Image Retrieval of Coats of Arms, *Proc. of the 1999 International Workshop on Multimedia Signal Processing*, Helsingör, 1999.

[4] Breiteneder, C., Merkl, D., and Eidenberger, H., Merging Image Features by Self-organizing Maps in Content-based Image Retrieval, *Proc. of European Conference on Electronic Imaging and the Visual Arts*, Berlin, 1999.

[5] Breiteneder, C., and Eidenberger, H., Performance-optimized feature ordering for Content-based Image Retrieval, *X European Signal Processing Conference*, Tampere, 2000.

[6] Flickner, M., and et al., Query by Image and Video Content: The QBIC System, *IEEE Computer*, 1995.