AUTOMATIC QUERY GENERATION FOR CONTENT-BASED IMAGE RETRIEVAL

Christian Breiteneder
University of Vienna, Institute for Computer Science
and Business Informatics
Liebiggasse 4/3-4, A-1010 Vienna, Austria
Breiteneder@ifs.univie.ac.at

Horst Eidenberger
Ministry of Science and Transport,
Austrian Libraries Network
Garnisongasse 7/21, A-1090 Vienna, Austria
hme@bibvb.ac.at

ABSTRACT

We describe a subsystem of a content-based image retrieval (CBIR) environment that supports a user in the definition of image similarity. Out of a single image or a set of query images we refine a query model: a list of feature extraction functions with associated thresholds and weights. The subsystem aims at bridging the gap between a user's high-level concepts and the low-level visual features employed and at supporting both, the casual user and the expert. The paper investigates and evaluates several approaches for this purpose within a CBIR system for coats of arms. A user may edit any entry of the query model in order to optimize retrieval results by iteration.

1. INTRODUCTION

The objective of content-based image retrieval (CBIR) is to efficiently retrieve images that satisfy a user's criteria of similarity. In order to cover a wide range of similarity aspects CBIR systems usually exploit multiple features addressing different image properties. The use of multiple features confronts the user with several difficulties: First, it requires a deeper understanding of the feature functions implemented. Second, the user has to understand how these functions are to be combined and has to provide further specification by assigning thresholds and weights.

In many cases this problem of matching a user's high-level concepts with low-level features is demanding too much and users often refuse using such a system [10] [11]. Several CBIR systems therefore offer the possibility of presenting an example search image from which queries are automatically generated. The disadvantage of this approach is a general decrease in retrieval quality.

The approach presented in this paper tries to overcome this disadvantage by first, employing an iterative technique in which generated queries can be refined depending on retrieval results and second, allowing a user to present a set of query images in order to better define the user's criteria of similarity. This approach is novel for multi-feature CBIR systems. To the authors' knowledge there is no such functionality in any of the common commercial or experimental systems (QBIC [5], Virage [1], VisualSEEk [12], MARS [6], etc.).

Similarity in our CBIR system is defined by *query models* [2]. A query model is a list of tuples of the form: feature extraction function, distance function, threshold and weight. The size of the result set is determined by the thresholds of all elements of a query model and not - as common in other retrieval approaches - by an absolute number. In other words, every entry in a query model eliminates some images until the result set is computed.

For the generation of query models from the query images given two different approaches are investigated:

- Approach 1: derivation of a query model from a single search image to find *similar* images in the database.
- Approach 2: formulation of a set of suitable query models from a set of images to find all images in the database which belong to a certain *semantic group* (e.g., the group of family photos in a photo database)

After the user has presented query examples the retrieval system suggests suitable query models and runs a first query. Then – after examining the query result - the user can refine this search by adapting the used features, threshold values and weights to improve the quality of the result.

The remainder of the paper is organized as follows. Section 2 investigates several techniques for the generation of query models for both approaches. Section 3 presents evaluation results for these approaches within a CBIR environment for civic coats of arms. Concluding remarks are given in Section 4.

2. QUERY MODEL GENERATION

2.1 Model Generation following Approach 1

A method implementing the first approach has to solve the following three problems:

- *Feature selection*. Which features should be used in a query model?
- *Threshold definition*. What is the maximum distance between search image and a candidate image?
- *Weight definition*. What is the weight of each feature? For the definition of weights we use the algorithm presented in [3].

2.1.1 Feature Selection

There are two alternatives to solve the feature selection problem:

1. Use all features meeting the following condition:

$$w_f > g(m_w) \qquad (1)$$

where w_f is the weight of feature f, μ_w the mean over all weights and g a suitable linear function. This method is based on feature clustering with self-organizing maps (SOM; [8]) and is called the *SOM method* in this paper. The weight of a feature is its contribution to the cluster structure.

2. Use all features, which satisfy - for the search image - a certain condition ("striking properties"). For example, for a feature counting the number of color shades (described in [1]; the feature vector has only one element, f_0) we used condition (2). In other words, this feature is used in a query model if the number of color shades is less than 4 or greater than 10.

$$f_0 < 4 \vee f_0 > 10 \xrightarrow{?} use \quad feature \tag{2}$$

For each feature a suitable condition has to be defined. That's why we call this method the *condition method*. In an additional step we use the two methods in combination and employ all features selected by one or both of them.

2.1.2 Threshold Definition

There are again two alternatives for the definition of threshold values:

1. Setting thresholds in such a way that all features eliminate an equal proportion of the image database. For this purpose the prediction of the number of images, which a specific combination of feature and threshold would eliminate is required. This task is performed with the prediction algorithm presented in [4]. The algorithm employs a data structure storing for each image class, feature and distance value the number of similar images in the database. We call this method the *shared method*, since all features participate equally.

2. Deriving the threshold value from the feature weight: more important features should have lower threshold values to guarantee that returned images have very similar properties. We defined the threshold by equation (3),

$$t_f = 1 - g\left(\frac{w_f}{\sum_{i=0}^{F} w_i} \right) \tag{3}$$

where w_f is the weight of feature f, F is the number of features and g is a suitable linear function. In addition, we tested linear combinations. Figure 1 summarizes our algorithms for feature selection and threshold definition in approach 1.

2.2 Model Generation following Approach 2

The task of approach 2 is considerably more difficult because here we do not want to retrieve images most similar to a single search image but those belonging to a specific semantic group. Such a group is defined by the examples a user selects from the database. The algorithm addressing this goal consists of the following steps:

1. Dividing the example set into clusters. For this we used our SOM algorithm [3], which produces a natural clustering from an unsorted image database.

2. For each subset of images a query model is derived which should return only images belonging to the same semantic group and the same image cluster:

2.1 If there is only one example image in a sub-set we have a similar situation as described in approach 1. Consequently, we use the same methods to define a query model but stricter parameter values to retrieve only very similar images.

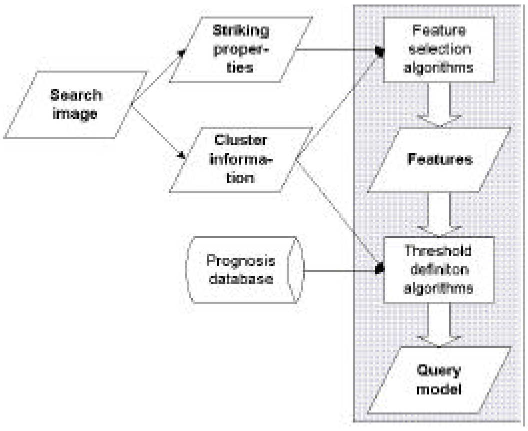

Figure 1. Feature selection and threshold definition in approach 1.

2.2 If there are two or more examples available in a sub-set, we compare - for each feature - all images by the features distance function and calculate the distance sum, the mean and the variance over all distance values. For the query we use the example with the minimum distance sum as the search image because this image can be considered to represent the center of the example set. For the query model all features satisfying the following condition are acceptable:

$$m_f < g(m_F) \tag{4}$$

Here μ_f is the mean over all distance values for feature f, μ_F is the global distance mean and g is a suitable linear function. In our test environment in which all distance functions are normalized to the interval [0,1] we used heuristics to identify equation (5) as a suitable feature condition.

$$m_f < 0.1 m_F + 0.05 \tag{5}$$

Finally, to complete the query model, it is necessary for each selected feature to define proper weights and threshold values. For the weights we use again the method presented in [3]. The thresholds were derived from the distance means and variances for all features:

$$t_f = g(m_f, s_f) \tag{6}$$

Again, by heuristics, the linear function given by equation (7) turned out to be suitable for g:

$$t_f = 1.5 m_f + 0.01 s_f + 0.1 \tag{7}$$

However, the method does not guarantee that a query model using these thresholds does indeed retrieve suitable images. If the query model is too prohibitive and therefore the result set empty, we repeat the query with a backup query model. This model has the same appearance as the original model but uses the threshold calculation method of approach 1 (with more strict parameters). Figure 2 shows the data flow in approach 2.

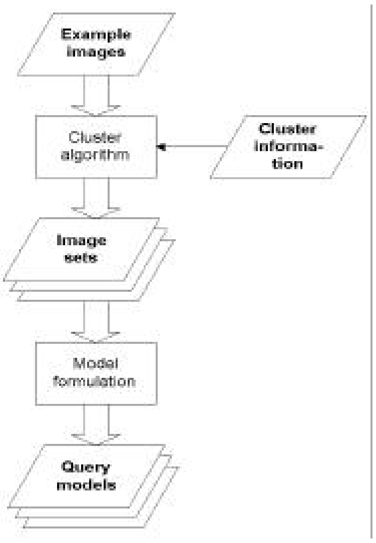

Figure 2. Query model generation in approach 2.

Our test environment is based on IBM's QBIC system (version 2; [5]) and was extended by web interfaces for searching by query models and group definition, a query engine which can handle query models and some C/C++-libraries implementing the algorithms for model generation, weight definition, performance optimization, etc. The image database comprises 444 pictures of German civic arms, for which 19 features (color, shape, specific features for heraldry, etc.) were developed that use three C/C++-libraries for vectorization, object recognition, etc. Heraldry-specific features include seal prints, the segmentation of arms, symmetries, etc. Clustering was done using Kohonen's SOM-PAK software [8]. To verify the conditions in the feature selection phase of approach 1 we used and adapted the free GNU test command [7].

3. EVALUATION

This section describes tests and results for the two approaches. We used recall and precision to compare various methods and tried to focus on recall while trying to keep precision reasonably good.

3.1 Tests and Results for Approach 1

For approach 1 we first tested the quality of the basic methods for feature selection and threshold definition. To test the feature selection methods we made several queries for each feature selection method and a set of threshold definition methods. Additionally, we compared the performance of our algorithms with the case of simply using all available features for each query. Besides the rather bad query computation performance of the latter method it turned out that recall values of this method are much lower than the ones of our algorithms. Figure 3 shows recall and precision for the feature computation methods.

The best recall was produced by the condition method (79%). It is considerably higher than the recall for the second basic method, the SOM method (56%). The reason why these values are rather poor is that they are averaged over all threshold definition methods. We will see later on, that the best combination of basic methods produces quite reasonable results. Using all features results in a very low recall value (25%) because here similarity is defined globally and not focusing on the given image class.

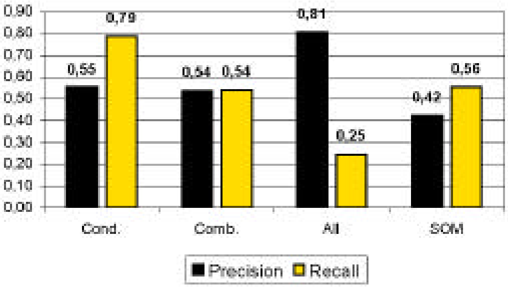

Figure 3. Performance of feature selection methods.

Next we investigated the performance of the different threshold definition methods. Again, we evaluated each method with every available feature selection method and calculated the mean for recall and precision. Figure 4 shows the results of this process. The recall of the shared method is much higher (74%) than the one of the other basic method (41%). Surprisingly, the combined method improves the recall by 3%. Obviously, the importance of a feature alone is not sufficient to derive a good threshold value. However, it may represent a useful contribution to the shared method.

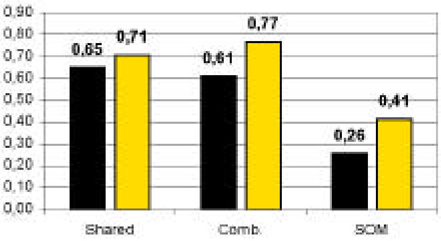

Figure 4. Performance of threshold definition methods.

After testing the basic methods all possible combinations were checked in order to identify the best algorithm for automatic query model generation. Additionally, we compared the results of the generated models to the results a human expert can achieve. Figure 5 shows the resulting recall and precision values.

The best combinations are the two best basic methods: feature selection by striking properties and threshold definition by the shared method with additional cluster information. It has a recall value of 94% with a precision of 68%. The second and third best methods (condition / shared method, all features / combined thresholds) produce both better precision values (75% vs. 100%) by much smaller recall values (80% vs. 38%).

Human experts familiar with image class and query environment reached a recall of 83% with a precision of 91%. The lower recall (compared to the best generation algorithm) can be explained by the way the expert tests were done: we used the best generated query model and tried to improve the precision by adapting the threshold values without significantly dropping the recall.

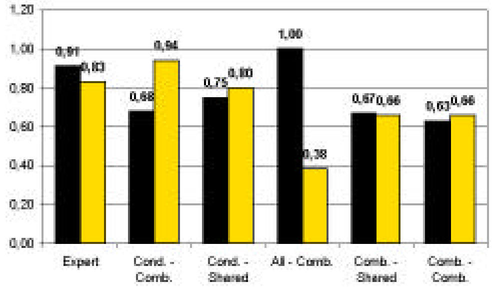

Figure 5. Performance of generated query models.

3.2 Tests and Results for Approach 2

For approach 2 we mainly tested the development of recall and precision when we increased the number of examples. We used the semantic group of Bavarian coats of arms for testing (see figure 6). These arms have a field division of two or three regions where the top region shows blue and white lozenges, the Bavarian national emblem. Object layout, colors, and field division depend on the history of the bearing community and therefore vary from image to image.

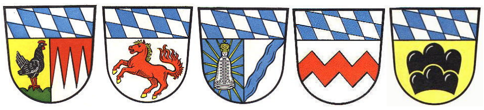

Figure 6. Examples of Bavarian arms.

In our tests (see figure 7) we found, that using more examples does not necessarily improve the quality of the result set; in some cases precision was even reduced. In general, using more examples increased recall while precision remained constant. The performance of an algorithm for approach 2 depends very much on *which* examples are chosen to describe the semantic group. It can be seen from the diagram below that in one query example we reached a recall of nearly 100 percent with two examples and only 82 percent with three examples.

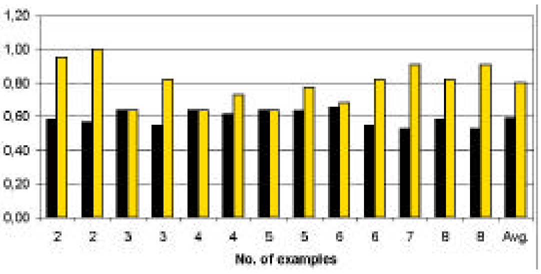

Figure 7. Performance of approach 2.

The algorithm presented is able to reach a recall value of 80 - 90% and to keep precision on a level of about 60%. Compared to earlier tests when trying to retrieve the members of a semantic group with a single query model (38% recall, 51% precision) an improvement of 29% for recall and 22% for precision was gained. We draw the conclusion that the retrieval of semantic groups is not a problem easy to solve.

4. CONCLUSION

The interactive approach presented in this paper allows the user to define similarity queries by a set of images. Out of these images query models are generated that may be iteratively refined depending on the retrieval result. Query models consist of a layered list of feature extraction functions with associated distance functions, thresholds and weights. We investigated, tested and compared several methods for their generation. The best methods developed result in a recall value of 94% with a precision of 68%.

Acknowledgment: The authors are thankful to Simon Gibbs for comments and proof-reading.

5. BIBLIOGRAPHY

[1] Bach, J., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R., Shu, C., "The Virage image search engine: An open framework for image management", SPIE Storage and Retrieval for Image and Video Databases, 1996.

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

[3] Breiteneder, C., Merkl, D., 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.

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

[5] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P., “Query by Image and Video Content: The QBIC System”, IEEE Computer, 1995.

[6] Huang, T., Mehrotra, S., Ramchandran, K.: Multimedia Analysis and Retrieval System (MARS) Project, Data Processing Clinic, 1996.

[7] Homepage of the GNU project: http://www.gnu.org/

[8] Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J., SOM-PAK: The Self-Organizing Map Program Package, Helsinki, 1995.

[9] Pentland, A., Picard, R. W., Sclaroff, S., Photobook: Content-Based Manipulation of Image Databases, SPIE Storage and Retrieval Image and Video Databases II, 1994

[10] Rui, Y., Huang, T., Chang, S., Image Retrieval: Past, Present and Future, International Symposium on Multimedia Information Processing, Taiwan, 1997.

[11] Rui, Y., Huang T. S., Ortega, M. and Mehrotra S., Relevance Feedback: A Power Tool in Interactive Content-Based Image Retrieval, IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Segmentation, Description, and Retrieval of Video Content, pp. 644-655, Vol. 8, No. 5, 1998.

[12] Smith, J. R., Chang, S., VisualSEEk: a fully automated content-based image query system, ACM Multimedia, 1996.