A Retrieval System for Coats of Arms

Christian Breiteneder
University of Vienna, Institute for Applied Computer Science and Information Systems,
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
Horst.Eidenberger@bibvb.ac.at

Abstract

The paper describes a project aiming at content-based retrieval of coats of arms to facilitate information finding. The retrieval system should be usable by experts and by casual users. In order to make use of the specific nature of arms about 20 different features have been developed to allow an expert to fine-tune an image search. For the casual user different features are grouped into feature lists that are called query models. These models correspond to image clusters in the database. The paper discusses the customization of features for the retrieval of coats of arms and their grouping into query models. Experiments and test results show a good performance of query models.

keywords: content-based image retrieval, image databases, coats of arms.

1. Introduction

The continuously increasing number of digital image databases pushes the demand for customized solutions of content-based image retrieval (CBIR) for a particular scope and interest. The development of a CBIR system for a new domain requires the selection, adoption and integration of existing solutions and the development of new ones. The designer of such a system is confronted with issues as what image features are best suited, how should they be extracted and compared, and what techniques should be employed for proper searching. In addition, the types of prospective users have an important impact on search and similarity features: Expert users may be trained in using the system in order to gain sufficient expertise in selecting and controlling suitable similarity features. Casual users may require a very small set of robust features that cover most purposes. However, such features are difficult to identify and to obtain. Finally, the particular application domain may add specific requirements and demand uncommon solutions.

The application domain discussed in this paper is the retrieval of images of civic coats of arms. Especially in European countries coats of arms have a long tradition reaching back to early medieval times. Later on, besides aristocratic families, arms were used to identify regions, cities, states and special social groups, e.g. craft guilds. Knowledge about arms was high in former times and people would be able to read them. This knowledge has been lost and is now only available from books or heraldry experts.

Huge libraries employ such experts to handle enquiries on coats of arms. Typically, library users get in touch with these experts, describe the coats of arms or show an image and ask for its origin or more general information about the arms. The expert uses his specific knowledge to identify the arms in question or to find similar arms in books on heraldry in order to know more about their semantics. This process is time consuming and sometimes ends with failure. An arms database with an interface for CBIR would very much improve the situation: The search process would take considerably less time, errors would be minimized and search results would be presented in a compact form.

The project presented aims at developing a CBIR system for coats of arms supporting both, the needs of the expert and the casual user. The expert is provided with a rich set of features that can be controlled by parameters while the casual user may just present an image (not necessarily from the database) and ask for similar images to be retrieved. For the latter purpose the project attempts to select a small set of developed features that describe best the specific nature of the image in question. These features are grouped into so called query models in order to optimally correspond to clusters of images in the database.

The paper describes the first major steps in the development of a CBIR system: the selection, implementation and testing of suitable image features and their grouping into query models. Chapter 2 starts with an explanation of the characteristics of arms and their advantage for CBIR.Two subsections give an overview over the various features developed and the idea and computation of query models. Chapter 3 outlines the test environment and Chapter 4 summarizes the testing process and the gained results. Chapter 5 gives a short outlook on future work.

2. Coats of Arms Retrieval

Books on arms published in the last decades very often contain an index that is sorted by the various elements occurring in arms. However, there is no general agreement on the method of index production and different fields, for example, aristocratic and civic arms, show huge differences in the methods applied and in the elements listed. Heraldry experts have expressed strong doubts that even with computer support there will ever be a general systematic index for coats of arms [8].

In this situation CBIR seems to be an appropriate and more promising approach. In addition, arms retrieval offers certain advantages in comparison to more general approaches or other application domains. First, arms have a synthetic origin and use only a few specific colors and shades, called tinctures in heraldry. Second, most arms have a clear structure and simple composition. They show often symmetries and shields are composed according to a relatively small set of rules. Finally, arms contain only a limited variety of objects, called charges in heraldry, as e.g., dragons, lions or Christian symbols. There is even a distinction between the types of charges (proper and common) that we are not exploiting at the moment.

Over the centuries heraldry has developed its own terminology (see e.g., [5]) which in general we try to avoid in this paper. However, some of these terms will be used especially, if they are easily understood from the context. Some of the main elements of coats of arms needed for further discussion are depicted in Figure 1. These elements will also be used for the explanation of some of the features utilized.

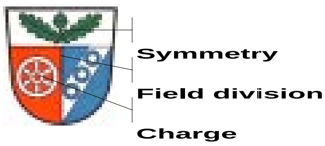

Figure 1 Main elements of coats of arms images

2.1 Features

For the retrieval of coats of arms various features addressing different elements and characteristics were developed. The underlying idea of the feature development was that different features are used in common. This is different to approaches relying mainly on color [10], texture [2][6] or shape [7]. The features developed and used may be grouped into 5 categories:

- Systematic features—among these are features for the detection of field divisions, the seal print (important for old arms which are often kept in this form only) or the complexity of arms.
- Object features—to extract typical charges or the object layout of arms images.
- Symmetry features—e.g., vertical symmetry.
- Color features—we use a color histogram for the typical occurring colors: Or (gold, yellow), Argent (silver, white), Gules (red), Azure (blue), Sable (black) and Vert (green). Purple was left out because it is rarely used in images in the database (German Civic Arms). In addition, we developed features to count the number of tinctures respectively color shades in images.
- Combined features—a typical example is a feature that identifies the field division of arms and calculates a color histogram for each field.

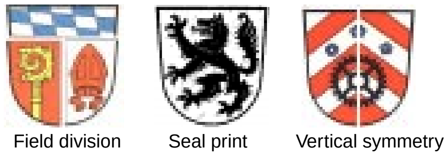

Figure 2 Examples of the 3 features discussed

In the following we pick three features as examples from the 19 developed so far and describe in more detail how they work: the segmentation feature or field division, as it is called in heraldry, the feature for finding similar seal prints and - as an example for the group of symmetry features - vertical symmetry. Figure 2 shows a typical example for each feature (all three pictures are from our test database and show the arms of Füssen, Wolfhagen and Bielefeld (all Germany)).

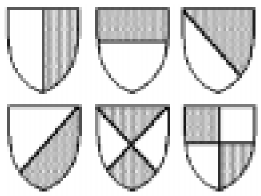

Figure 3 Some allowed layouts for field division of shields

Heraldry defines several rules for the composition of “valid” arms that are used by the various features. For example, one set of rules concerns the field division of the shield. Some of the allowed layouts are shown in Figure 3 (from [5], modified). The purpose of the segmentation feature is to detect the type of field division of an image. For this purpose the following algorithm was implemented:

- Detection of edges in the input image and merging of neighboring edges.
- Evaluation of identified edges: check, if they have the correct length, the correct starting point and the correct angle to be an element of a field division.

- The type of field division is derived from the number, orientation and position of valid edges identified. The feature value is an integer which represents the various types of segmentation.

The distance function for this feature compares the field division for two images and returns 0 if they are equal, 0.5 if the are similar (e.g, the first and the last shield in Figure 3) or 1 else. The current version of this feature distinguishes 22 different types of field divisions and detects the correct segmentation for 86 percent of the images in the test database.

Heraldry has developed standard black and white textures for the substitution of seals tinctures in case color representation or printing is not possible. The feature for seal prints is based on these tincture substitutes.

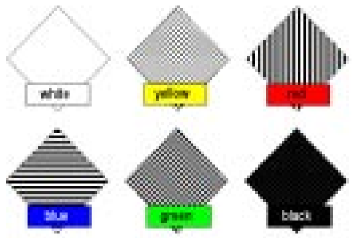

Figure 4 Common tincture substitutes

The feature calculation function maps subareas of the input image into one pixel of the feature matrix which is a two-dimensional matrix of boolean values. The size of the feature matrix is constant and therefore the size of the subarea depends on the overall size of the input image. For each subarea the most frequent tincture is determined and represented in the feature matrix as one of the textures shown above. This is done by simple rules for every color; for example, if the most frequent color in the subarea $R(x_1,y_2,x_2,y_2)$ of the input image is red then the corresponding pixel P(x,y) of the feature matrix is set if x is an even number. Applying these rules for each subarea of an image results in a seal print of the input image with a standard size. The distance function for this feature simply overlays two feature maps and counts matching pixels. The more matching pixels found the smaller is the distance value returned.

The feature for horizontal symmetry cuts the input images into two halves, mirrors the second one, overlays the first with the second and counts the pixels with similar colors. The feature value is the number of equal pixels in two images. The distance function returns the Euclidean distance of the two feature values. This concept is used for all symmetry features. The concept describes whether two images have the same type of symmetry and also similarity of symmetry of the images in question.

2.2 Query Models

Since coats of arms follow certain rules in their composition, coloring, and objects occurring, images following the same set of rules form image clusters. Using this fact and in order to make the retrieval as easy as possible for the casual user we try to select these features for a similarity query that match best to images of these clusters. For this purpose we have developed simple, very specific and robust features that should be used in combination. Currently, we select features manually and group them into a query model intended to deliver images of a specific cluster.

We define a query model as a list of weighted features. A feature entry in this list consists of a feature extraction function, a distance function and a parameter controlling the size of the result set. We are experimenting with two different types of this parameter:

- the number of images in the result set and
- a threshold for the maximum distance between the query object and the objects being retrieved.

The first method - the standard method in QBIC (see section 3) - is easier to handle but - as shown in section 4 - the threshold method improves the recall for most queries significantly. The following table represents a simplistic example of a query model consisting of two features:

No	Feature function	Distance function	Max. threshold
1	Colour histogram	Euclidean	0.5
2	vertical symmetry	Boolean	0

Table 1: Examples of query models

In the first step of the example the color histogram of every image is compared to the color histogram of the query object. For images with a distance value less than 0.5 it is determined if they show a vertical symmetry. These images are returned as the query result. The result set is ordered by the weighted sum of the distance values (position value) for each image. The position value for each database object is defined by the equation

$$\textit{Position value}_{Object} = \sum_{i=1}^{F} w_i d_i$$

where F is the number of features, w_i the weight of feature i and d_i the distance value between the query object and the database object for feature i. This evaluation method assumes that all distance functions are standardized for the same range.

Besides improving the recall value the use of thresholds has another important advantage. The result set stays the same even when the order of features in a query model changes which is not the case for using the size of the result set as parameter. We can therefore order the features by their evaluation speed and use the fastest features first. Features with lower performance have then to be calculated for fewer images. We will incorporate this performance-optimized ranking into automatically generated query models in the future.

3. The Retrieval Environment

This section describes the environment in which experiments and tests were performed: the QBIC system, our extensions, the web interface and the used images.

For the tests described in section 4 we used the QBIC system developed by IBM [1][3]. Among its advantages are an easy to use C++-API and input filters for many image formats. For our purpose we extended QBIC by a practical web interface using perl as CGI-scripting language, a search engine for query models, a possibility to control the size of the result set by thresholds (similar to QbQBE) and some C-libraries for vectorization [9], object evaluation and so on. The features were programmed as C++-classes. The architecture of the test environment is shown in Figure 5.

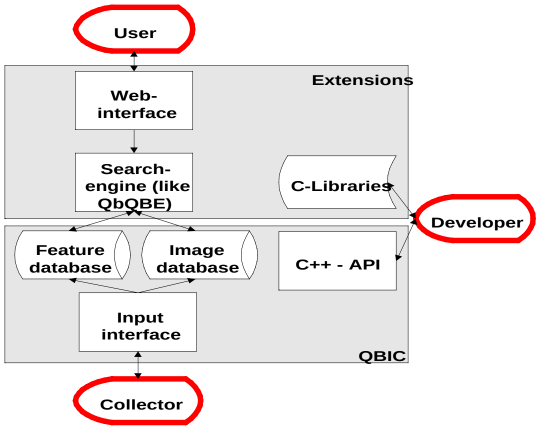

Figure 5 Architecture of the retrieval environment

The web-interface consists of a query-section for the definition of query models, a picture-section presenting the retrieval results and a result table with statistic information (returned images per feature, performance, etc.). Figure 6 shows the web-interface (rendered by Netscape Navigator 4.5) with the three important elements: statistics for the query issued, results of the query and the query model definition section. In this example all images in the database with a Bavarian top, which have a T-like field segmentation and more than three percent red pixels are looked for. It can be seen from the statistics table (top of page) that there are 23 images with T-form, ten with T-form and a Bavarian top and six with all three selection criteria in

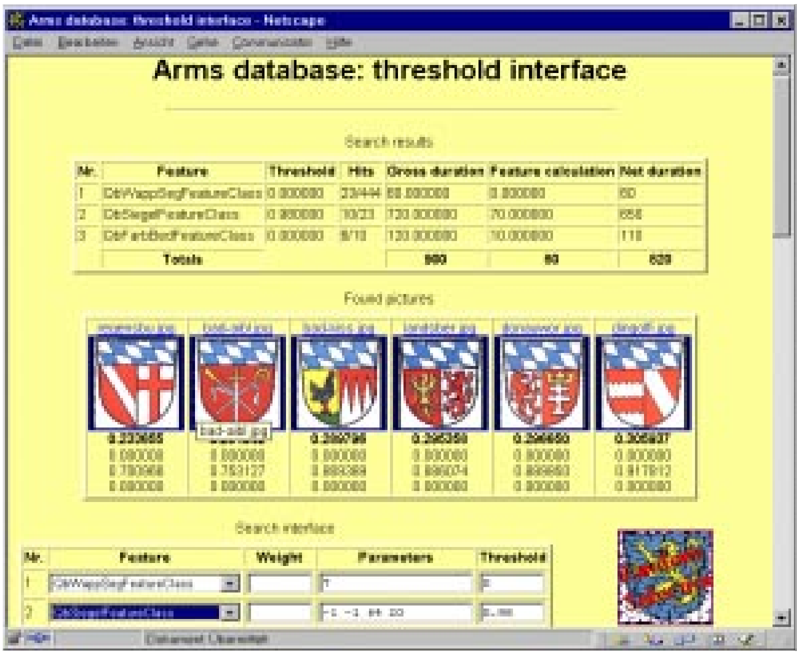

Figure 6 Web interface example

the database. These images are presented - together with their distance values to the query image - in the second table.

The coats of arms for our test database (444 pictures) were taken from a heraldry server in the Netherlands [4]: most of them are German civic arms and show only the shield. Each picture was rated (segmentation, etc.) and groups of similar arms were formed (e. g., Bavarian arms).

4. Tests and Results

This section describes the testing process for our arms database: what queries we used, the query models derived and the results in terms of precision and recall we gained. In the discussion the following terms will be used:

- Efficiency—the ability of a feature to cluster the image database.
- Query model—a list of features. A feature is a combination of a feature calculation function, a distance function and an identifier for the size of the result set.
- Search question—a sentence describing the result set (e.g., “all green images with a white border”).

No	Query	Used query model
1.1	Symmetric, red images	X-axis symmetry, tinctures histogram
1.2	Images with a vertical division in two parts and few colours.	Field division, number of used colour gradations
1.3	Images with a big central image in white or yellow and a blue background	Object layout, tinctures histogram
2.1	Images with no field division, a big black charge and 3 colours.	Field division, seal print, number of tinctures
2.2	Colourful images without any symmetry containing complex charges	Number of tinctures, type of symmetry, image complexity
2.3	Simple images with few colours and without any field division.	Image complexity, number of tinctures, field division
2.4	Shields with Bavarian flag on top and without golden elements.	Seal print, tinctures histogram with expression evaluation

Table 2: Search queries and corresponding query models

The evaluation process was performed in three steps:

- Formulation of suitable search queries. For the moment, we tried to phrase our search questions on a more technical level, enthusiastic questions as for example, a query for all arms which politically belong to Bavaria can hardly be answered by our current system.
- Derivation of a query model for each search question. Every feature developed was used at least once and each query model contains at least two features.
- Testing by using the web-interface and computation of precision and recall for each model.

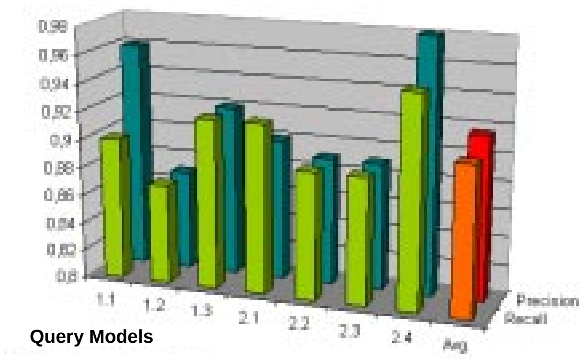

Figure 7 Test results for query models

Table 2 summarizes search queries and the derived query models. For these search queries and query models several tests were performed with different query images. The average precision and recall values were calculated over all tests. Figure 7 presents the results for all tests where thresholds are used to define the size of the

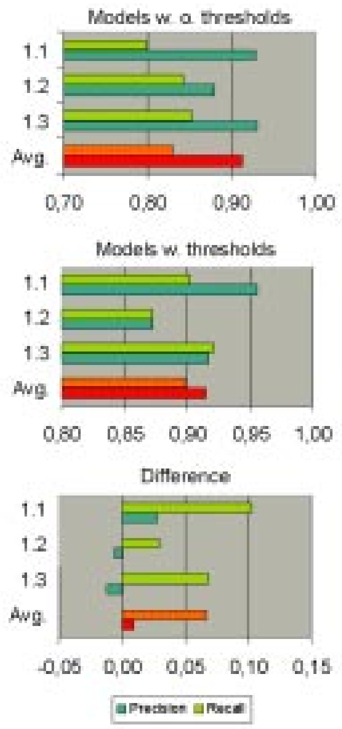

Figure 8 Tests with and without thresholds

result set: The minimum average precision and recall for every question was 87 percent, the average over all tests shows a precision value of 92 and a recall value of 91 percent. In addition to tests on the quality of features and query models we also evaluated the improvement of recall and precision due to using thresholds. It can be seen from Figure 8 that using thresholds increases the recall considerably. We used the search questions 1.1 to 1.3 for this purpose and got the results depicted in Figure 8. The use of thresholds hardly affected the precision of query models (sometimes the precision was even reduced) but improved the recall by an average of seven percent. It seems to be that using thresholds does not “cut off” suiting images with a slightly greater distance if they exceed the number of images to return. Thresholds handle this problem much more flexible and should be seen - besides the feature calculation function and the distance function - as another important component to control similarity.

5. Future Research

Currently, we are tuning feature functions by adding new control parameters that are not discussed in this paper.

At the moment, the problem of ordering features in a (manually defined) query model was solved by using the features with faster distance functions first. For a more sophisticated solution to this problem, an optimization model was developed, which also takes into account the contribution of a feature to cluster the image database. It will be one of the next steps to integrate this model into the current environment.

One of the main goals of the project is the automatic generation of query models. For this purpose we plan to use a self-organizing net to identify image clusters. In a second step the contribution of the various features to describe these clusters will be calculated. This contribution factor will be used to rank the images in the result set.

When experimenting with the retrieval system we sometimes made the following observation: The result set of images included images with a small distance value but were surprising for the experimenter. For example, the features used in the query model would focus on symmetry, color distribution and objects included, but some of the images in the result set had additional colors that appeared to be so prominent that an experimenter would have estimated a larger distance value. So the question to be answered is whether or not an automatic generation of query models will solve or at least reduce this problem or human perception has also to be taken in account.

6. Conclusion

The paper outlines a project for CBIR of coats of arms. During the project the characteristics of arms were analyzed in order to develop and implement specific features for segmentation, object layout, etc. The underlying idea of the feature development was that they are used in common. Such combinations of features are called query models. Different models are evaluated by recall and precision values and different parameters describing the size of the result set tested. The implementation uses IBM’s QBIC system as kernel and runs under LINUX. Results are promising but leave enough space for future work.

7. References

[1] 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, Vol. 28, No. 3, 1995, pp. 23-30.

[2] Gorkani, M.M. and Picard, R.W., Texture Orientation for Sorting Photos ‘at a glance’. In *Proc. 12th Int. Conf. on Pattern Recognition* (October 1994), pp. A459-A464.

[3] *IBM QBIC – Homepage*: http://wwwqbic.almaden.ibm.com/

[4] *International Civic Arms*: http://www.bng.nl/ngw/int/dld/germany.htm

[5] *Introduction to arms*: http://www.sca.org.au/lochac/scribes/hrld_int.html

[6] Ma, W. Y., and Manjunath, B. S., Texture-based pattern retrieval from image databases. *Multimedia Tools and Applications* 2, 1 (January 1996), 35-51.

[7] Mehrotra, R. and Gary, J.E., Similar-shape retrieval in shape data management. *IEEE Computer* 28, 9 (September 1995), 57-62.

[8] Neudecker, O., *Coats of Arms Encyclopedia* (in German), Bechtermünz, 1991.

[9] Steinböck, E., *Extraction of essential structures from grey-scaled bitmaps as a preliminary stage for object-recognition* (in German), master thesis Technical University of Vienna, 1988.

[10] Strickler, M. and Orengo, M. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases III*, Vol. 2420 of SPIE Proceedings Series, 1995.