Content Based Image Retrieval by Combining Features and Query-By-Sketch

Raoul Pascal Pein
Department of Computer Science
Hamburg University of Applied Sciences, Germany

Zhongyu (Joan) Lu
School of Computing and Engineering
University of Huddersfield, UK

Abstract

This paper reports an approach to improve content-based image retrieval systems. Most current systems are based on a single technique for feature extraction and similarity search. Each technique has its advantages and drawbacks concerning the result quality. Usually they cover one or two certain features of the image, e.g. histograms or shape information. To overcome these restrictions a flexible framework is proposed, capable of combining several different features in a single retrieval system. This system allows an administrator to build a repository managing different feature vectors. A user searching through this repository defines and weights these features according to his needs in the query. It concludes that a combined retrieval can be used much more widely than a highly specialized one and the use of query-by-sketch or -example combined with semantic information (e.g. keywords) could enhance the result quality.

Keywords: Content-based image retrieval (CBIR), feature vectors; query image; combined retrieval; improved result quality

1 Introduction

With rapid increasing volume of image repository, efficient access/retrieval becomes a huge challenge for computing science and other disciplines [3]. Current Internet search technology is largely on the text-based search rather than on the image based search. For the pixel-based images, the solution would be expected from a different approach, because pictures do not contain repeating symbols, but consist of unique patterns of pixels. It follows that the search mechanisms for images are different from that of common text search [3].

To solve the problems mentioned above, a standard framework has been proposed as Content based Image Retrieval - CBIR [10]. The framework contains an image repository, indexed features and a user interface for formulating queries [10]. The purpose to use indexing structure is to find relevant matches within a repository to achieve efficient search.

Currently numerous studies have been reported in this area [3, 8, 9, 10, 11]. Eakins and Graham developed a 3 layered model that differentiates between primitives (e.g. colours, histograms, shapes), entities (e.g. objects, individuals) and abstract entities (e.g. moods, a complex scenery) [3]. A well-known system is Query By Image Content - QBIC, developed by IBM [9]. However, these systems are often explorative and not yet widespread in the Internet, e.g. services providing *query-by-sketch QBS* [8, 11] cannot be found easily. But some open photo data base Flickr [4] allows people to share images and let them be annotated by random visitors. This approach works fine with keywords and categories but currently lacks a content based search. Basically most systems either provide support for keywords or content. It follows that a study in the combination of content-based image search techniques is necessary.

The main objective of this research is to develop a search tool capable of finding images based on a simple user drawn sketch. To enhance the search capabilities, different additional parameters will be considered. Combining multiple methods (strings, colour information, etc.) is expected to generate a high quality ranking for the result set. The focus of the search tool lies in improving the quality of the generated results rather than the retrieval speed. To accelerate the process especially for large scale applications, multi-dimensional data structures like the R-Tree [5, 6] is going to be used.

2 Methods Employed

The basic concept is to combine several separate similarity features in a single system, i.e. all features need to be comparable against each other, independent from their characteristics. Otherwise the system only generates incoherent result sets, one for each feature. Therefore, all similarities of each feature are mapped to a linear range $[-1.0, 1.0]$ expressing the similarity/differences between 2 images. The value -1 stands for the complete opposite, 1 stands for identity. This simplification allows merging different similarity values into a combined ranking r_x for image x compared to the query:

$$r_x = \frac{1}{\sum_{f=1}^{n} w^f} * \sum_{f=1}^{n} w^f * r_x^f \tag{1}$$

where w^f is the weight/importance of feature f. r_x^f is the partial ranking for image x using feature f. A newly developed indexing method to extract image features is based on a research of Faruq A et al [1]. A histogram is represented by 12 expressive stochastic moments.

$$V_I = [\mu_R, \mu_G, \mu_B, \sigma_R^2, \sigma_G^2, \sigma_B^2, S_R, S_G, S_B, \rho_{RG}, \rho_{RB}, \rho_{GB}][1] \tag{2}$$

The similarity s between two histograms is defined by the distance between their two feature vectors. It is calculated by the formula

$$s = E_{QI} = \frac{2 * (V^Q \bullet V^I)}{V^Q \bullet V^Q + V^I \bullet V^I}[1] \tag{3}$$

where V^Q is the query vector and V^I is an indexed vector from a repository image.

To add spatial information to this histogram, the image is split into a quad tree. In the prototype, a tree depth of 3 is chosen, resulting in 16 sub images. For each sub image a separate histogram is calculated. The similarity (s) of a complete image is composed of the similarities of its sub images. For each sub image the similarity is defined by the equation above (see equation 3). The average value is further developed as follows:

$$s = \frac{1}{n} * \sum_{i=1}^{n} s_i = \frac{1}{n} * \sum_{i=1}^{n} \frac{2 * (V_i^Q \bullet V_i^I)}{V_i^Q \bullet V_i^Q + V_i^I \bullet V_i^I} \tag{4}$$

where s_i is the similarity of a single sub image and n the total amount of sub images. As s lies between -1 and 1, this value can be interpreted as ranking r without further mapping.

For the prototype, three feature vector modules have been developed: First the *histogram* [1] ("Stochastic"), second the *enhanced version with quad tree* ("Quad Stochastic"); and finally a simple *keyword* ("Keyword") search. The system allows users to adjust the weights w^f of each module to meet the users needs. Additionally, basic browsing is supported.

The histogram modules take query images and the keyword module accepts a search string, requiring an individual query form described in the module itself. Every module contains a function to calculate similarities and an individual function to extract the specific feature from an image. The indexing data is stored in a database.

Finding an image is then performed in several steps. First, the user builds a query consisting of an image and/or additional information like keywords. The weights for each feature can also be adjusted if wanted. Then the system generates a combined ranking using equation 3. To keep the final result set in a realistic size, only the highest ranked images are presented to the user. As there is no absolute right or wrong in the content based modules, the limit can be set manually. The keyword module differentiates much less, so it is imaginable to draw a hard line between hits and rejects.

3 Results

The test database contains information of 1709 images. To evaluate the efficiency of each implemented module, several queries are executed with a set of images. For each query image all other relevant images in the repository are picked manually. The generated result sets are then compared to determine if all expected results achieved high ranks. In addition each result set is reviewed for all other images with an acceptable similarity to the query.

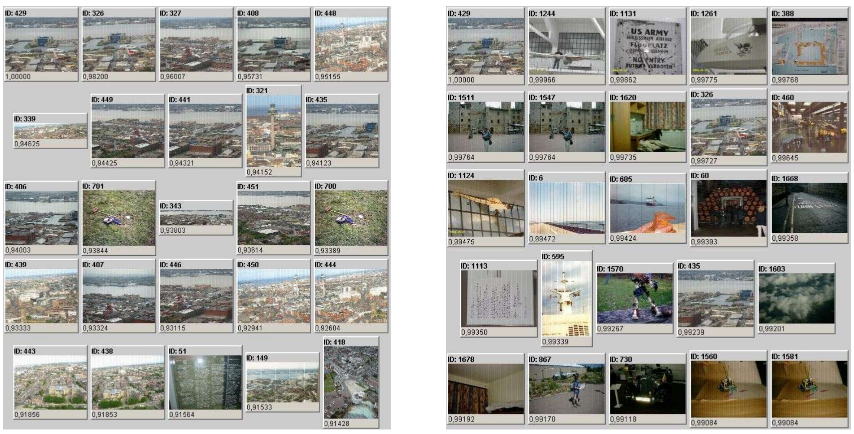

Figure 1: Results for example query with *Quad Stochastic* (left) and *Stochastic* (right) module

An exemplary result set is represented in figure 1. Both content-based modules, i.e. Stochastic and Quad Stochastic, generated a reasonable ranking. The image identical to the query is located at the highest position on the top left. It is a bird view, containing many grey tones. In this particular case, acceptable similarity is determined by the context, i.e. other bird views.

A closer look reveals differences in the quality. The "Quad Stochastic" module put most of the manually picked images to top positions. The simpler "Stochastic" module found more grey images with a different context (fig. 2).

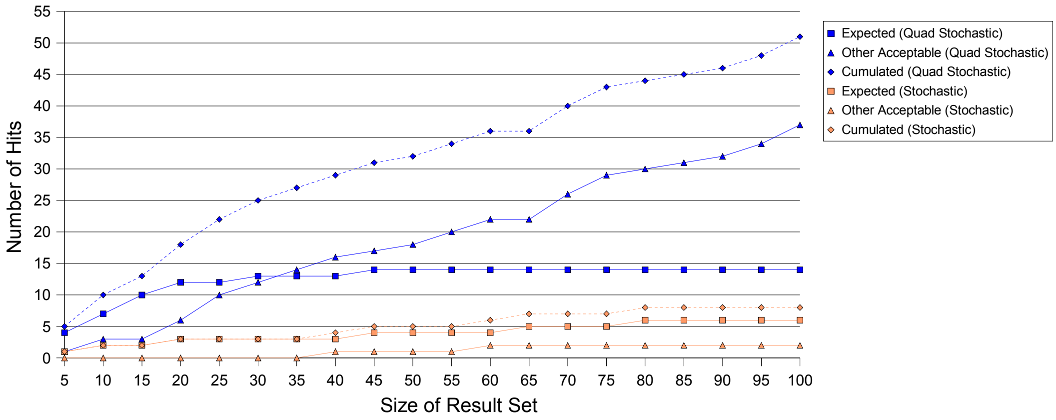

Figure 2: Amount of correct hits in the result set

The second part to be evaluated is the efficiency of the combined ranking. The prototype combines three different result sets by using equation 1. As the *Keyword* component cannot handle query images, an additional query keyword is set. The query image shows a scene in Liverpool, therefore the query "liverpool" is passed to the engine.

This keyword already stands for a simple multi modal query. The search engine now has two independent views to calculate the final ranking. The weighted ranking is displayed in figure 3 listing the detailed ranking for the first 125 images. To refine the result set, the weigths w^f for each of the three modules has been adjusted manually.

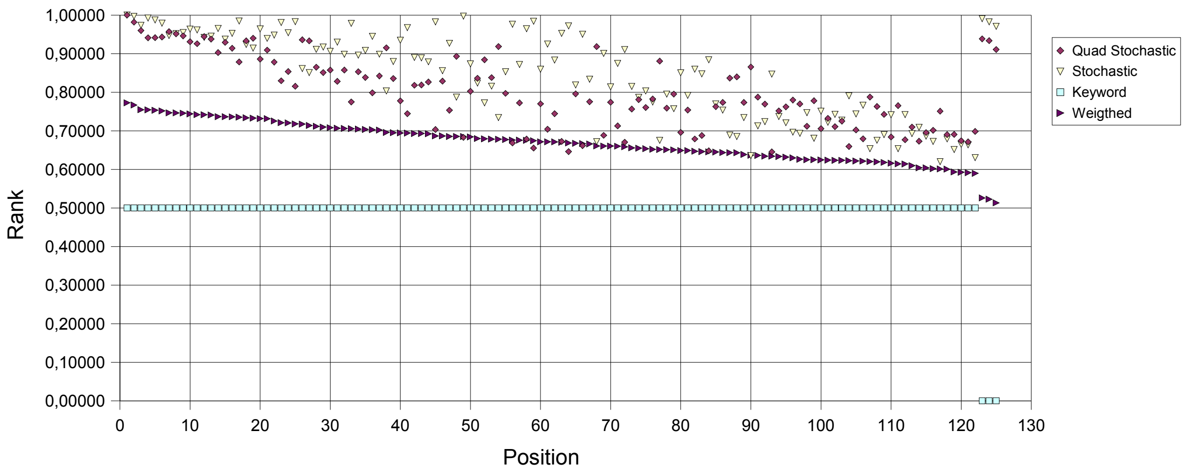

Figure 3: Combined and weigthed ranking

4 Discussion and Analysis

Figure 2 plots the amount of correct hits in the result set for both modules. An eye-catching result is the different slope. While the cumulated hits of the Quad Stochastic module rise steadily up to about 50 hits among the first 100 results, the Stochastic module does not even find 10 acceptable images. Further the Quad Stochastic module put most expected images among the first 30 positions. From now on the influence of the other acceptable hits surpasses the expected results which remain at the same level.

The comparison of the two result sets indicates the high impact of the chosen module on the result. Both modules - *Stochastic* and *Quad Stochastic* - are based on the same compressed histogram [1]. Adding spatial information by generating multiple histograms for each image causes significant differences compared to the simpler model.

The *Stochastic* module focuses on the amount of each colour of the image, ignoring all other information. It is a quite powerful tool to find images with similar lighting, rotated or translated views. The major drawback is the poor ability to reject false positives.

A user in search for all images of the same bird view series should prefer the *Quad Stochastic* module. This module exploits information about several different image regions. It enables the system to compare images in a way that matches the human perception quite well. Obviously this advantage goes to expense of a larger dataset and the lack of retrieving rotated, translated or cropped images.

A way to overcome the drawbacks of each single module is the combination of their strengths. Figure 3 shows the effect of combined retrieval. In this case, especially the *Keyword* module is a highly efficient filter to reject false positives. At the lower end (rank 123-125) some images had a very high similarity (> 0.9) when only considering the content based modules. All 122 top ranked images all contain the specified keyword, while the similarity drops from 1.0 to about 0.6. All of the first 24 images show houses of Liverpool from above, while the best straightforward search (fig. 1, left) also contains images of grass patches and an information panel.

Looking at the current prototype, there are many possible extensions to be added. A couple of further ideas are listed below. Many of these are far beyond the scope of this project, but they seem to be beneficial additions.

- Support for shape based search may be added
- A semantic search could be added, similar to the Ontogator project [7]. This requires a huge administration effort.
- The concept of ImageScape [2] to arrange semantic icons on a canvas could be added. Spatial distribution of semantic content seems to be a very expressive way to formulate queries.
- Providing an open repository [4] in the Internet could lessen the effort of manual annotation. Every user could be allowed to add new images and edit annotations. Such a repository hopefully grows with the time and the workload of annotating is distributed to several people.
- Researching relevance for other data types (e.g. audio files)

5 Conclusions

This paper shows a possible way to develop a flexible image retrieval system. The prototype design is a trade off between extensibility and high performance with an emphasis on extensibility. It is assumed that a combined retrieval can be used much more widely than a highly specialized one.

The prototype is in general a framework for image retrieval, already offering basic functionality to be used.

- The Retrieval Tool offers multiple query types and allows keywords as well as the import of external images.
- A new indexing model has been developed, supporting query by sketch. The model exploits the spatial distribution of colours to compare images.
- New feature vector modules can be written and integrated quite easily. This might be useful for testing newly developed models.
- Multiple indexing techniques can be combined and individually weighted to improve the result quality.
- The user interface for formulating queries is extensible. Special requirements of a new feature vector can be supported by programming a new QueryPanel which is added to the basic query form.

References

[1] Faruq A. Al-Omari and Mohammad A. Al-Jarrah. Query by image and video content: a colored-based stochastic model approach. *Data and Knowledge Engineering*, 2004.

[2] Jean Marie Buijs and Michael Lew. Visual learning of simple semantics in imagescape. In D. P. Huijsmans and A. W. M. Smeulders, editors, *Visual Information and Information Systems*, pages 131–138, June 1999.

[3] J.P. Eakins and M.E. Graham. Content-based Image Retrieval. A Report to the JISC Technology Applications Programme. Technical report, University of Northumbria at Newcastle, January 1999. Available from: `http://www.unn.ac.uk/iidr/VISOR`.

[4] Flickr - photo sharing [online]. 2006. Available from: `http://www.flickr.com/`.

[5] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *ACM SIGMOD Conf. on the Management of Data*, pages 47–57, 1984.

[6] P.W. Huang, P.L. Lin, and H.Y. Lin. Optimizing storage utilization in r-tree dynamic index structure for spatial databases. *The Journal of Systems and Software*, 55:291–299, 2001.

[7] Eero Hyvönen, Samppa Saarela, and Kim Viljanen. Intelligent Image Retrieval and Browsing Using Semantic Web Techniques A Case Study. Technical report, Helsinki Institute for Information Technology (HIIT) / University of Helsinki, September 2003.

[8] Michael Lew, Kim Lempinen, and Nies Huijmans. Webcrawling using sketches. In *2nd International Conference on Visual Information Systems*, pages 77–84, December 1997.

[9] W. Niblack, X. Zhu, J. Hafner, T. Breuel, D. Ponceleón, D. Petkovic, M. Flickner, E. Upfal, S. Nin, S. Sull, B. Dom, B.-L. Yeo, S. Srinivasan, D. Zivkovic, and M. Penner. Updates to the qbic system. *Retrieval for Image and Video Databases VI*, 3312:150–161, 1998.

[10] Remco C. Veltkamp and Mirela Tanase. Content-Based Image Retrieval Systems: A Survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, October 2002.

[11] James Ze Wang, Gio Wiederhold, Oscar Firschein, and Sha Xin Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. In *Fourth Forum on Research and Technology Advances in Digital Libraries*, pages 13–24, May 1997.