2009 Second International Workshop on Similarity Search and Applications

Text-Based and Content-Based Image Retrieval on Flickr: DEMO

Juan Manuel Barrios, Diego Díaz-Espinoza, and Benjamin Bustos
Department of Computer Science
University of Chile
Santiago, Chile
Email: jbarrios,ddiaz,bebustos@dcc.uchile.cl

***Abstract*—We present an image retrieval system based on a combined search of text and content. The idea is to use the text present in title, description, and tags of the images for improving the results obtained with a standard content-based search. The system contains two different user interfaces: a sidebar for the browser designed for end users, where the user must enter the Flickr URL that is visiting and the system retrieves similar images from the collection, and an advanced search designed for experienced users, where the distance functions and weights can be customized.**

***Keywords*-Image databases; Information retrieval;**

I. Introduction

Text-based search provides results with semantic similarity, while content-based search provides results with visual similarity. Due to the independence between these approaches, is likely that their combination could improve the performance of a search system by benefiting of both approaches. In the present work, we present an image retrieval system based on a combined search of text and content.

II. System overview

The present image search system has been implemented using Java 1.6, C++, and PostgreSQL. The set of image object were taken from Flickr web site[1] using the SAPIR collection [3].

In the offline phase, the images are downloaded from Flickr using the URL provided by the SAPIR collection. The content-based descriptors extracted from the images were: Color Histogram $3 \times 3 \times 3$ using RGB color space (a $27d$ vector), Gabor Wavelet (a $48d$ vector), Efficient Color Descriptor (ECD) 8×1 using RGB color space (a $32d$ vector), ECD 8×1 using HSV color space (a $32d$ vector), and Edge Local 4×4 (a $80d$ vector). The Color Histogram and Gabor Wavelet descriptors were implemented in C++ with the OpenCV library, and the other ones were implemented in Java.

The text-based descriptors (title, description, and tags) were extracted from the SAPIR collection. The feature vectors were calculated using the vectorial model and the *tf-idf* weighing [1]. Six feature vectors were created for each image, three for the text using the Porter stemming algorithm [4] and three without stemming. A wordlist, stoplist, and the reverse file for the text features (with and without stemming) were also calculated and stored in the PostgreSQL database.

In the online phase, the user enters the query image, a search text, and a weighed distance function for each available feature. The distance functions can be metric (like Euclidean distance) or non-metric (like DPF and cosine distance). Then, the system performs a k-NN search using a weighed combination of distances, normalized by the maximum distance of a feature to the origin.

All the text-based and content-based features (up to 11 vectors for an image) are stored in a PostgreSQL database for efficient retrieval of a small subset, and in a binary file designed for efficient linear scan. Currently, the system contains more than 115,000 images in the collection and the binary file size is about 130 MB.

The system contains two different Graphical User Interfaces: a browser sidebar (intended for end users), and an advanced search (indented for experienced users).

III. Browser sidebar

The interface for end users consists in a sidebar for the browser[2] (see Figure 1). When the user is browsing images on Flickr site and wants to perform a search, he must copy the URL of the image that is viewing and paste it into the URL input text of the sidebar. The resulting set of objects are displayed to the user sorted by relevance, measured as the distance of each element in the collection to the query object. The user can click on a thumbnail for accessing the published image on Flickr web site.

IV. Advanced search

In the advanced search[3] (see Figure 2) the user can enter the query image by different means: uploading an image file, entering the URL of a public image, entering a Flickr URL that displays an image (this is the only way in which the query image will contain text-based and content-based

[1] http://www.flickr.com/

[2] For adding a sidebar in Firefox, create a bookmark to the URL http://prisma.dcc.uchile.cl/ImageSearch/sidebar.jsp, edit the properties of the bookmark, and mark the "Load this bookmark in the sidebar" checkbox.

[3] http://prisma.dcc.uchile.cl/ImageSearch/index.jsp

978-0-7695-3765-8/09 $25.00 © 2009 IEEE
DOI 10.1109/SISAP.2009.30

Figure 1. The sidebar installed in Firefox.

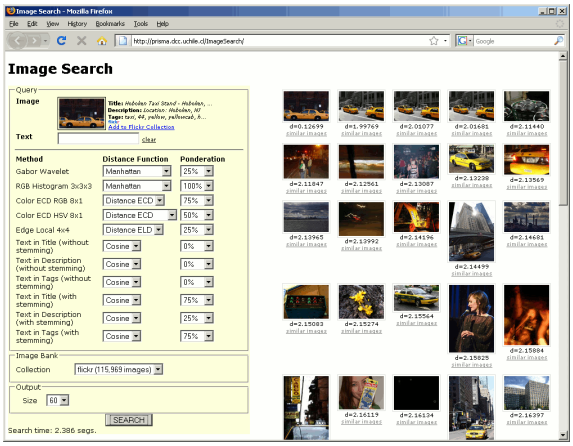

Figure 2. The advanced search.

features), or selecting one of the previous query images (useful for reusing query images). Then, the user may enter a search text. The words in the search text restricts the universe of objects for the similarity search. If the user did not enter a search text, the text-based features of the query image, if there is any, are used as search text. If the query image does not contain text-based features, the search will be performed over the whole collection.

Once entered the query image and text, the user can select the features for the comparison, the distance function for each feature, and the weight of each one in the final distance. Then, the system performs a k-NN search using the normalized weighed combination of distances.

The resulting set of objects are displayed to the user, and for each result the calculated distance to the query object is shown. The system displays a "similar images" link for each result, allowing to start a new search but using the selected object as a query object, and a link to the published image on Flickr web site.

V. Runtime performance

The system is installed on a desktop computer with an AMD Athlon XP 1800+ and 1 GB RAM running CentOS 5.2 with Sun Java 1.6, Jakarta Tomcat 6.0 and PostgreSQL 8.1 installed. The memory usage of the VM is about 90 MB (this includes the memory used by Tomcat and our system).

When the user does not enter any search text and the query object does not contain text-based features, a linear search over all the collection is performed. In this case, the search time depends mainly on the number of compared features and the distance functions used. Using all the features available in the system, the search time of the linear scan is about 20 seconds. Using only the RGB Histogram $3 \times 3 \times 3$ and Euclidean distance, the search time of a linear scan is about 1 second. It is worth noting that the images features are retrieved from a file during the search and are not stored in main memory.

When the user enters any search text or the query contains text-based features, the inverted file is used for retrieving the set of images that contain them. In this case, on average, the search time is reduced to less than a second.

VI. Conclusions

In this work, we presented a system that allows a combined search of text-based and content-based features on a sample of more than 115,000 images of Flickr obtained from the SAPIR collection.

We are evaluating the inclusion of the M^3-tree [2] into the system as an indexing structure for dynamic combinations of metric distances.

Acknowledgment

This work was partially supported by FONDECYT (Chile) Project 11070037.

References

[1] R. A. Baeza-Yates and B. Ribeiro-Neto. "Modern Information Retrieval". Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[2] B. Bustos and T. Skopal. "Dynamic similarity search in multi-metric spaces". In *Proc. 8th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'06)*, pages 137–146. ACM Press, 2006.

[3] F. Falchi, M. Kacimi, Y. Mass, F. Rabitti, and P. Zezula. "SAPIR: Scalable and distributed image searching". In *SAMT (Posters and Demos)*, vol. 300 of *CEUR Workshop Proceedings*, pages 11–12, Genoa, Italy, November 2007.

[4] M. F. Porter. "An algorithm for suffix stripping". *Program*, 14(3):130-137, July 1980. http://tartarus.org/~martin/PorterStemmer/