

Eva - An Evaluation Tool for Comparing Descriptors in Content-Based Image Retrieval Tasks

Otávio A. B. Penatti
Institute of Computing
University of Campinas
Campinas, SP, Brazil
penatti@ic.unicamp.br

Ricardo da S. Torres
Institute of Computing
University of Campinas
Campinas, SP, Brazil
rtorres@ic.unicamp.br

ABSTRACT

This paper presents Eva, a tool for evaluating image descriptors for content-based image retrieval. Eva integrates the most common stages of an image retrieval process and provides functionalities to facilitate the comparison of image descriptors in the context of content-based image retrieval. Eva supports the management of image descriptors and image collections and creates a standardized environment to run comparative experiments using them.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Search Process; H.3.4 [Systems and Software]: Performance evaluation (efficiency and effectiveness)—*tool*

General Terms

Experimentation, Standardization

Keywords

evaluation tool, image descriptors, content-based image retrieval

1. INTRODUCTION

Due to the great amount of images and videos available nowadays, content-based image retrieval (CBIR) has become an important approach for retrieving multimedia information. CBIR is an approach that uses image properties, such as color, texture and shape of objects for indexing and retrieving images. This creates the possibility of comparing images by considering their low-level properties. One advantage of this approach is that the image retrieval process is free from images textual annotations, which are usually subjective and arduous to create.

A CBIR system relies on the task of determining similarities among images. Image descriptors are used for this task.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.
Copyright 2010 ACM 978-1-60558-815-5/10/03 ...\$10.00.

The image descriptor is responsible for extracting image features and for comparing the images based on their extracted features.

It is very important to know which descriptors are more adequate for a given CBIR system. Many descriptors are application (collection) dependent. Consequently, it is important to conduct comparative experiments using different descriptors. However, comparing them is an arduous task. Several experiments may be executed, several routines and scripts need to be prepared and this process is error-prone and can lead to no standardized results.

We have developed a tool to facilitate the experimental comparison among image descriptors in the context of content-based image retrieval. This tool, named Eva, integrates the most important stages in an image retrieval process and provides a standardized environment for the comparative evaluation of image descriptors. Eva makes it easy to manage image descriptors, to manage image collections and to run comparative experiments using them, providing easy access to the results, like efficiency and effectiveness performance measures. It is important to mention that our tool is currently being used by researchers from Brazilian universities and has already been used in comparative studies [4].

The research community has already presented some attempts to standardize the performance comparison in the CBIR context [1, 3, 5], creating, for example, image collections, test suites, and new performance measures. However, Eva faces the standardization problem under a different point of view. Eva creates an environment to perform comparative experiments in an organized and standardized way.

In the next section we present the functionalities, the architecture and implementation aspects of Eva. Section 3 concludes the paper.

2. Eva

Eva has been implemented using PHP, Python and C languages and it has a Web interface that makes it accessible by any browser.

2.1 Architecture

Figure 1 shows the tool architecture. It has an interface to access the tool core modules. The core modules are responsible for managing image descriptors, managing image collections, running experiments and viewing the experiments' results. Eva has a repository that contains image descriptors, image collections, and data related to experiments.

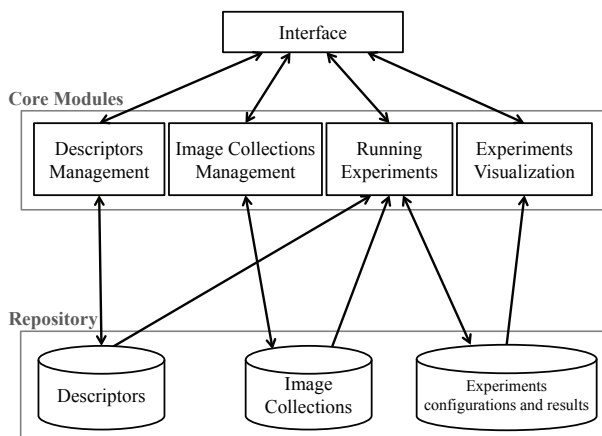


Figure 1: Eva’s architecture.

To manage all the information necessary, Eva uses a relational database, whose entity-relationship model is presented in Figure 2. The relational database stores information about image descriptors, image collections, experiments configurations, and results from experiments. Examples of experiments’ results data include time required for extracting feature vectors, time for distance computations, effectiveness values from user effectiveness evaluations, and distances between images.

Eva creates an environment to manage image descriptors, where it is possible to insert and delete descriptors used or developed by any researcher. Eva requires that a descriptor follows an specific implementation interface. This makes it possible to use a descriptor as a plugin to the tool (see Section 2.2 for more details).

Eva also creates an environment to manage image collections. Image collections can contain images in several different formats, like JPG, PNG, GIF, PPM, and the collection can be categorized or not.

2.2 Defining descriptors

A standard interface is defined for image descriptors. A descriptor must have the following two functions with their headings in C language:

- void Extraction(char *img_path, char *fv_path);
- double Distance(char *fv1_path, char *fv2_path);

The *Extraction* function is responsible for extracting the feature vector from the input image. The path of the input image is the first parameter and the second parameter is the path where the feature vector will be stored. Both parameters are paths from the file system and are adjusted by Eva when it calls this function. The *Extraction* function is responsible for opening the input image and for creating the output feature vector file.

The *Distance* function is responsible for comparing two feature vectors. Both parameters are paths of the feature vectors. The *Distance* function must open the feature vector files and return the distance between them according to the distance function used by the descriptor. The return value must be of double type.

When the descriptor’s implementation is ready, it must be compiled to generate the plugin. The compilation instruc-

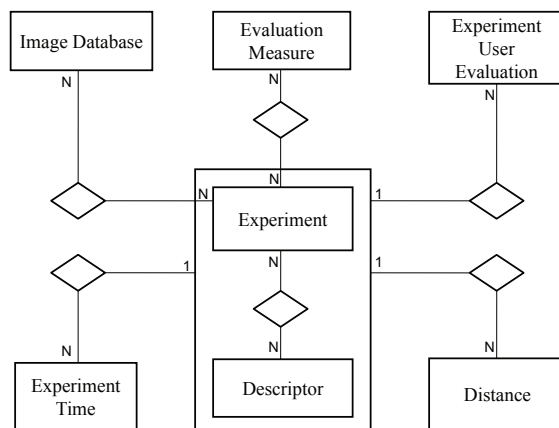


Figure 2: Entity-relationship model for the database used by Eva.

tions given by Eva generate the plugin file and this file must be submitted to Eva. When the file is uploaded, Eva verifies if the plugin follows the implementation instructions, testing, for instance, if the *Extraction* function exists. If the implementation is correct, the user may insert metadata information, such as the name, the authors, and the type of the descriptor (color, texture or shape).

2.3 Running experiments

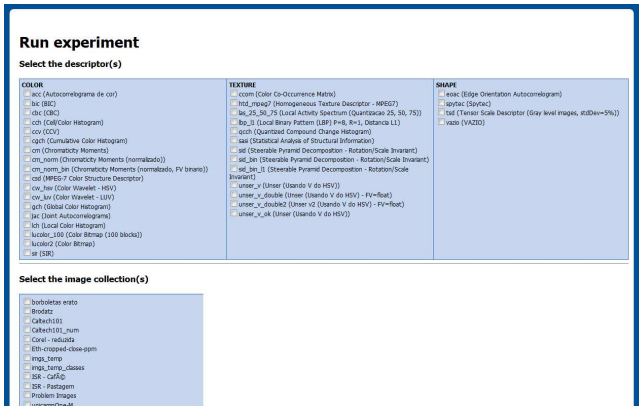
The main functionality of Eva is to run experiments. To run an experiment, the user needs only to select which descriptors and which image collections will be used. The comparison process is completely automatic. When preparing an experiment, the user has some optional parameters to configure. For example, it is possible to use a list of pre-defined query images. When this list is used, it is possible to conduct cross-validation experiments.

When running an experiment, Eva provides an interface to show the progress of each step of the experiment. When the experiment finishes, Eva provides many facilities. One of them, is an interface to view the images retrieved considering a query image and an evaluated descriptor. This interface makes it easy to compare the retrieved images by each descriptor for each query image. Eva also computes the average times required to extract image feature vectors and the average time necessary to compare them.

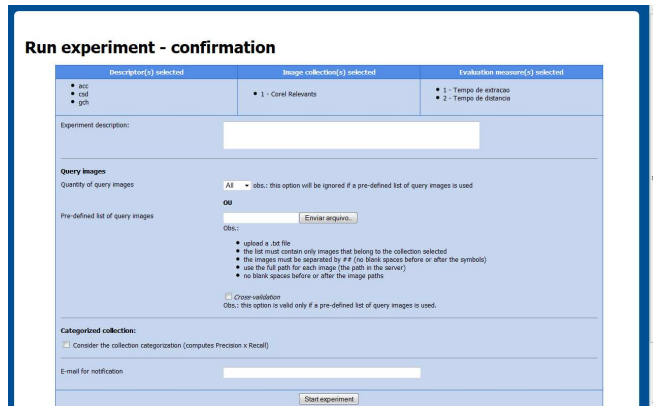
When the experiment uses a pre-defined list of query images, Eva also provides an interface to be used in a subjective (with real users) evaluation of the effectiveness of the descriptors. This subjective analysis is important when using databases with no pre-defined categorization, for example.

2.3.1 Experiment setup

Using Eva’s interface, the user selects the descriptors to be compared and the collections to be used. The user can also submit a pre-defined list of query images and can choose which effectiveness measures (e.g., Precision×Recall curves) should be computed. This last option is available only if the image collection is categorized. If a pre-defined list of query images is used, the user can also choose to use or not cross-validation. Figure 3 shows the interfaces to configure an experiment.



(a)



(b)

Figure 3: Eva’s experiment setup interfaces.

2.3.2 Experiment process

An experiment is basically composed by two stages: extraction of images feature vectors and distance computations.

In the beginning of an experiment session, the module responsible for features extraction is started. This module locates all the images of the selected collections and applies the descriptors on them. The *Extraction* function of each descriptor selected is used in this module. The elapsed time of each use of the *Extraction* function is computed. When all the images are processed, the module computes the average and the standard deviation of the elapsed extraction time for each descriptor.

After the extraction stage, the module responsible for distance computations is started. For each descriptor used, the module locates every feature vector file. The module verifies if a pre-defined list of query images is used. If no query list is used, every feature vector is compared to every feature vector. If a query list is used, the feature vectors are compared only to those of the query images. For every comparison to be computed, the module calls the plugin’s *Distance* function passing the paths of the feature vectors and stores the returned distance value into the tool database. Every use of the *Distance* function has its elapsed time measured. When the computations finish, the module computes the average and the standard deviation of the elapsed distance times of each descriptor.

A log file is used to record every step of all the modules. In case of errors, this log file can be analyzed.

2.3.3 Effectiveness and efficiency evaluation

When an experiment finishes, Eva provides interfaces to visualize the experiment results. One interface shows the times computed for each descriptor as well as the experiment configuration. Another option is to view the log file, the feature vector files and the effectiveness measures computed. Figure 4 shows the interface to view the experiment details.

There is also a button to access an interface that visualizes the retrieved images for each query image. This interface allows the direct comparison of the retrieved images by different descriptors. Figure 5 shows an example of this interface. A combo box lets the user switch among the de-

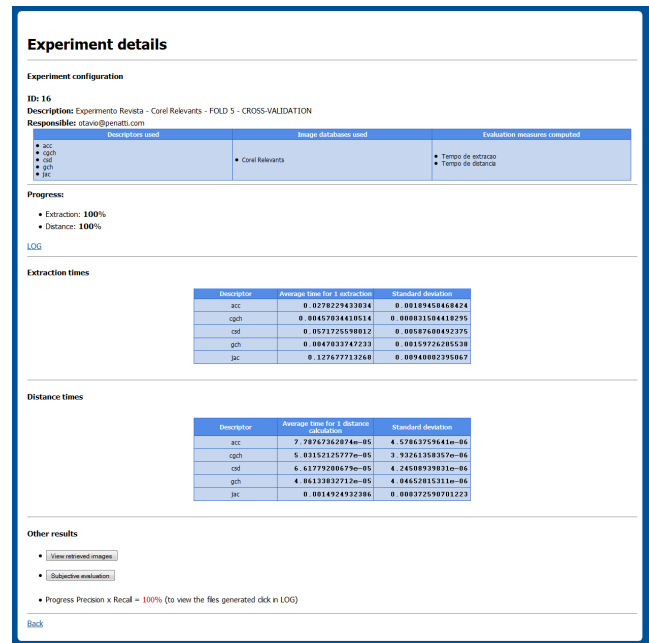


Figure 4: Eva’s interface to view the experiment details.

scriptors used in the experiment. If a list of pre-defined query images was used in the experiment, another combo box lets the user switch among the different queries used. If no query list was used, every image in the list of retrieved images can be used as query image.

2.3.4 Subjective evaluation

When a pre-defined list of query images is used, Eva provides a way to conduct a subjective analysis of the descriptors effectiveness by creating an interface for a real user evaluation. This interface is similar to the interface used to view the retrieved images. This kind of evaluation is important when the image collection is not categorized.

In [2] a user-oriented evaluation is used for digital libraries. The effectiveness of structured and non-structured queries in

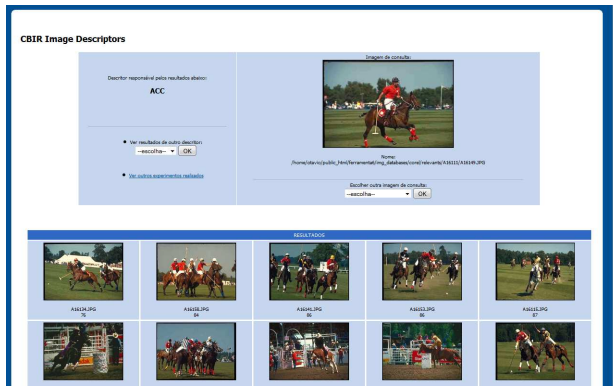


Figure 5: Eva’s interface to view the retrieved images.

digital libraries is evaluated by real users. A similar evaluation is realized by our tool to assess the effectiveness of the image descriptors.

Figure 6 shows an example of the interface for the subjective evaluation. The interface shows to the user the query image and a set of retrieved images. Users are asked to indicate which of those images they consider similar to the query image.

The set of images showed to the user is created as follows. For each descriptor used in the experiment, a list containing the 30 most relevant images is obtained for a given query image. The lists of each descriptor are combined, eliminating duplicates. Every image in the list has a reference to the descriptor(s) that retrieved it and to the rank in its original list. The final list is shuffled and showed to the user.

The query image is highlighted at the top of the page and the shuffled list is showed below it. The user has no information about which descriptor retrieved each image. After the user has indicated the images he or she considered similar to the query image, he or she clicks on a button to finish the evaluation of that query image. The process is repeated until all query images are evaluated.

When the user finishes the evaluation, effectiveness measures are computed. The effectiveness measures currently computed by Eva are: P_{10} , P_{20} , and P_{30} . These measures stand for precision values and they indicate the percentage of images marked as similar among the top 10, 20, and 30 results, respectively.

These measures are calculated for each descriptor, for each query image and for each user. Therefore, it is possible to calculate the descriptors’ effectiveness for each query image independently.

An example evaluation interface can be accessed at www.lis.ic.unicamp.br/~otavio/cbir_eval/experiment503.html.

3. CONCLUSIONS

Evaluating image descriptors comparatively is very important as descriptors are crucial for CBIR systems. However, the comparative process is error-prone and can lead to problems in standardization. To solve these problems, we presented Eva, which is a tool for evaluating image descriptors in the context of content-based image retrieval. Eva provides a standardized environment for the evaluation of image descriptors integrating the most important stages in

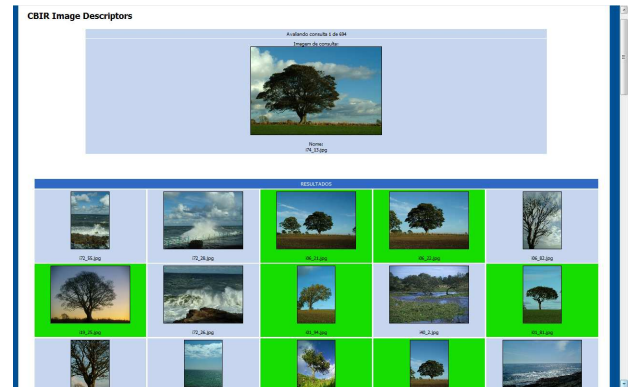


Figure 6: Eva’s interface for subjective evaluation.

an image retrieval process. Eva makes it easy to manage image descriptors, to manage image collections, and to run comparative experiments, providing easy access to efficiency and effectiveness measures.

4. ACKNOWLEDGMENTS

Authors are grateful to FAPESP (grants 2006/59525-1, 2007/52015-0 and 2009/10554-8), CAPES, CNPq, and Microsoft Research for financial support. We also thank Rodrigo Senra for valuable help during the design and implementation of the Eva tool.

5. REFERENCES

- [1] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.
- [2] M. A. Gonçalves, E. A. Fox, A. Krowne, P. Calado, A. H. F. Laender, A. S. da Silva, and B. A. Ribeiro-Neto. The effectiveness of automatically structured queries in digital libraries. *Joint Conference on Digital Libraries*, 0:98–107, 2004.
- [3] H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval: overview and proposals. *Pattern Recognition Letters*, 22(5):593 – 601, 2001.
- [4] O. A. B. Penatti and R. d. S. Torres. Color descriptors for web image retrieval: A comparative study. *XXI Brazilian Symposium on Computer Graphics and Image Processing*, pages 163–170, October 2008.
- [5] J. Vogel and B. Schiele. Performance evaluation and optimization for content-based image retrieval. *Pattern Recognition*, 39(5):897 – 909, 2006.